

**УЧЕБНИК
ДЛЯ ВУЗОВ**

ПИТЕР

СТАНДАРТ ТРЕТЬЕГО ПОКОЛЕНИЯ



Н. В. Макарова В. Б. Волков

Информатика

**для
БАКАЛАВРОВ**

**РЕКОМЕНДОВАНО
УЧЕБНО-МЕТОДИЧЕСКИМ ОБЪЕДИНЕНИЕМ**

Информатика



Наталья Владимировна Макарова — заслуженный работник высшей школы РФ, профессор, доктор педагогических наук, кандидат технических наук, заведующая кафедрой информационных систем и технологий. Внесла большой вклад в становление и развитие школьной и вузовской информатики. Автор более 300 печатных работ, среди которых порядка 40 книг. Широкое признание в России завоевал рекомендованный Минобрнауки РФ учебно-методический комплект из 15 учебников и учебных пособий по школьному курсу информатики (5–11 классы), а также двухтомный учебник по информатике, который на протяжении многих лет являлся основным учебником по этому предмету в вузах нетехнического профиля. Имеет большой опыт преподавания для различных категорий учащихся. Сферы профессионального интереса — информационные системы и технологии в экономике и менеджменте, управление проектами, моделирование, методика преподавания.



Владимир Борисович Волков — старший преподаватель кафедры информатики и математики МГГУ, сертифицированный специалист Borland и AltLinux, эксперт в области SAP BusinessObjects Business Intelligence. Ведущий обучающих курсов по Delphi, CBuilder, JavaBuidler, технологиям разработки ПО Microsoft (Visual Studio), Perl, Bash, OpenOffice, операционной системе Linux. Автор и соавтор ряда книг по информационным технологиям и программированию.

В учебнике в соответствии с требованиями государственных образовательных стандартов представлен материал по основным направлениям информатики. Рассмотрены следующие разделы информатики: информатика как интегрирующая область деятельности; современные технологические средства и программный инструментарий новых информационных технологий (системное и прикладное программное обеспечение, инструментарий создания программных продуктов).

Рекомендовано Учебно-методическим объединением по университетскому политехническому образованию в качестве учебника для студентов высших учебных заведений, обучающихся по направлениям подготовки бакалавров «Системный анализ и управление» и «Экономика и управление». Учебник также может быть использован для подготовки студентов следующих направлений: гуманитарные науки, социальные науки, естественные науки, культуры и искусство, образование и педагогика, здравоохранение.



Заказ книг:

197198, Санкт-Петербург, а/я 127
тел.: (812) 703-73-74, postbook@piter.com

61093, Харьков-93, а/я 9130
тел.: (057) 758-41-45, 751-10-02, piter@kharkov.piter.com

www.piter.com — вся информация о книгах и веб-магазин

ISBN: 978-5-496-00001-7



9 785496 000017



СТАНДАРТ ТРЕТЬЕГО ПОКОЛЕНИЯ

Н. В. Макарова, В. Б. Волков

Информатика

Рекомендовано Учебно-методическим объединением
по университетскому политехническому образованию
в качестве учебника для студентов высших учебных
заведений, обучающихся по направлениям подготовки
бакалавров «Системный анализ и управление»
и «Экономика и управление»



Москва · Санкт-Петербург · Нижний Новгород · Воронеж
Ростов-на-Дону · Екатеринбург · Самара · Новосибирск
Киев · Харьков · Минск

2011

ББК 32.973.233я7
УДК 004.3(075)
М15

Макарова Н. В., Волков В. Б.

М15 Информатика: Учебник для вузов. — СПб.: Питер, 2011. — 576 с.: ил.

ISBN 978-5-496-00001-7

В учебнике в соответствии с требованиями государственных образовательных стандартов в трех разделах представлен материал по основным направлениям информатики. В разделе «Информатика как область интеграции знаний» дается представление о роли информации в развитии общества, об управлении знаниями, о логических основах построения компьютера, о методах и средствах моделирования, об основах построения баз данных и информационных системах, об информационной безопасности, о менеджменте информационной сферы. В разделе «Техническая база информатики» приведена информация об аппаратной части компьютера, о компьютерных сетях, рассматривается история и тенденции развития компьютерных систем. В разделе «Алгоритмическое и программное обеспечение информатики» излагаются основы теории алгоритмов и технологии программирования, приводится классификация программного обеспечения и характеристики разных классов программных продуктов.

Рекомендован Учебно-методическим объединением по университетскому политехническому образованию в качестве учебника для студентов высших учебных заведений, обучающихся по направлениям подготовки бакалавров «Системный анализ и управление» и «Экономика и управление».

Учебник также может быть использован для подготовки студентов следующих направлений: гуманитарные науки, социальные науки, естественные науки, культура и искусство, образование и педагогика, здравоохранение.

ББК 32.973.233я7
УДК 004.3(075)

Все права защищены. Никакая часть данной книги не может быть воспроизведена в какой бы то ни было форме без письменного разрешения владельцев авторских прав.

Информация, содержащаяся в данной книге, получена из источников, рассматриваемых издательством как надежные. Тем не менее, имея в виду возможные человеческие или технические ошибки, издательство не может гарантировать абсолютную точность и полноту приводимых сведений и не несет ответственности за возможные ошибки, связанные с использованием книги.

ISBN 978-5-496-00001-7

© Макарова Н. В., 2011
© Волков В. Б., 2011
© ООО Издательство «Питер», 2011

Оглавление

Предисловие	13
ЧАСТЬ ПЕРВАЯ. ИНФОРМАТИКА КАК ОБЛАСТЬ ИНТЕГРАЦИИ ЗНАНИЙ	15
Глава 1. Представление об информации	17
1.1. Понятие информации	17
1.1.1. Определение информации	17
1.1.2. Информация и данные	18
1.1.3. Адекватность информации	19
1.2. Меры информации	20
1.2.1. Синтаксическая мера информации	20
1.2.2. Семантическая мера информации	22
1.2.3. Прагматическая мера информации	23
1.3. Качество информации	25
1.4. Информационные процессы	27
1.5. Кодирование при передаче и хранении информации	29
1.6. Основы классификации и структурирования информации	31
1.6.1. Иерархическая система классификации	33
1.6.2. Фасетная система классификации	35
1.6.3. Deskрипторная система классификации	36
1.6.4. Классификация информации по разным признакам	37
1.7. Кодирование при классификации информации	41
1.7.1. Общие понятия	41
1.7.2. Классификационное кодирование	41
1.7.3. Регистрационное кодирование	43
Вопросы для самопроверки	43
Литература	44
Глава 2. Роль информации в развитии общества	45
2.1. Информатизация общества	45
2.1.1. Представление об информационном обществе	45
2.1.2. Роль информатизации в развитии общества	50
2.1.3. Опыт информатизации и перспективные идеи	53
2.1.4. Роль средств массовой информации	54
2.1.5. Об информационной культуре	55
2.2. Информационный потенциал общества	57
2.2.1. Информационные ресурсы	57
2.2.2. Информационные продукты и услуги	59
2.2.3. Рынок информационных продуктов и услуг	62

2.3. Информатика — предмет и задачи	70
2.3.1. Появление и развитие информатики	70
2.3.2. Структура информатики	72
Вопросы для самопроверки	74
Литература	75
Глава 3. Теоретические основы управления знаниями	76
3.1. Управление знаниями	76
3.1.1. Общее представление об управлении знаниями	76
3.1.2. Модель жизненного цикла управления знаниями	79
3.2. Данные, информация и знания	79
3.3. Модели представления знаний	81
3.3.1. Семантические сети	82
3.3.2. Фреймы	84
3.3.3. Формальные логические модели	85
3.4. Системы, основанные на знаниях	86
3.4.1. Экспертные системы	88
3.4.2. Нейронные сети	91
3.4.3. Системы, основанные на прецедентах	96
3.4.4. Системы, построенные на генетических алгоритмах	98
3.4.5. Интеллектуальные агенты	99
3.4.6. Системы добычи данных	100
3.5. Инженерия знаний	101
3.5.1. Получение знаний	102
3.5.2. Жизненный цикл и методология	105
Вопросы для самопроверки	106
Литература	107
Глава 4. Логические основы информатики	108
4.1. Представление о высказываниях и логических операциях	108
4.1.1. Понятие высказывания	108
4.1.2. Соглашения о языке алгебры высказываний	110
4.1.3. Логические операции над высказываниями	110
4.2. Алгебра логики	113
4.2.1. Понятие формулы алгебры логики	113
4.2.2. Равносильные формулы алгебры логики	115
4.2.3. Основные равносильности	116
4.2.4. Равносильности, выражающие одни логические операции через другие	116
4.2.5. Равносильности, выражающие основные законы алгебры логики	117
4.2.6. Решение логических задач методами алгебры логики	117
4.2.7. Булева алгебра	119
4.3. Построение коммутационных схем на основе алгебры логики	121
Вопросы для самопроверки	124
Литература	124
Глава 5. Информационные системы и технологии	125
5.1. Основные сведения об информационных системах	126
5.1.1. Понятие информационной системы	126
5.1.2. Этапы развития информационных систем	126
5.1.3. Процессы в информационной системе	127
5.1.4. Результаты внедрения информационных систем	128
5.1.5. Роль структуры управления в информационной системе	128
5.1.6. Персонал и прочие элементы организации	131
5.1.7. Примеры информационных систем	132

5.2. Структура и классификация информационных систем	133
5.2.1. Структура информационной системы	133
5.2.2. Классификация информационных систем по признаку структурированности задач	138
5.2.3. Классификация информационных систем по функциональному признаку и уровням управления	141
5.2.4. Прочие варианты классификации информационных систем	147
5.3. Основные сведения об информационных технологиях	149
5.3.1. Понятие информационной технологии	149
5.3.2. Инструментарий информационной технологии	150
5.3.3. Соотношение между информационными технологиями и системами	151
5.3.4. Составляющие информационной технологии	152
5.3.5. Использование информационных технологий	153
5.4. Виды информационных технологий	155
5.4.1. Информационная технология обработки данных	155
5.4.2. Информационная технология управления	157
5.4.3. Автоматизация офиса	159
5.4.4. Информационная технология поддержки принятия решений	161
5.4.5. Информационная технология экспертных систем	164
Вопросы для самопроверки	166
Литература	167
Глава 6. Теория баз данных	169
6.1. Общие понятия	169
6.1.1. Компоненты среды функционирования СУБД	171
6.1.2. Классификация СУБД	173
6.1.3. Функции СУБД	176
6.2. Модели данных	178
6.2.1. Классификация моделей данных	178
6.2.2. Термины и определения	180
6.2.3. Модель «сущность-связь»	181
6.3. Реляционные базы данных	185
6.3.1. Реляционная модель данных	185
6.3.2. Правила Кодда	186
6.3.3. Ключи и связи	188
6.3.4. Ссылочная целостность	190
6.3.5. Нормализация данных	190
6.3.6. Язык SQL	192
6.3.7. Преимущества и недостатки реляционной модели	194
6.4. Постреляционные модели и базы данных	196
6.4.1. Основные понятия OODM	197
6.4.2. Преимущества OODM	197
6.4.3. Недостатки OODM	198
6.5. Проектирование баз данных	198
Вопросы для самопроверки	200
Литература	201
Глава 7. Основы моделирования	202
7.1. Представление о назначении и особенностях моделирования	202
7.1.1. Понятия модели и моделирования	202
7.1.2. Компьютерное моделирование	204
7.1.3. Параметры модели	205
7.2. Классификация моделей	207
7.2.1. Классификация моделей по назначению	207
7.2.2. Классификация моделей по уровню моделирования	208

7.2.3. Классификация моделей по принадлежности к иерархическому уровню	209
7.2.4. Классификация моделей по характеру взаимоотношений со средой	210
7.2.5. Классификация моделей по способу представления свойств объекта	210
7.2.6. Классификация моделей по причинной обусловленности	211
7.2.7. Классификация моделей по отношению ко времени	212
7.2.8. Классификация моделей по сфере применения	212
7.2.9. Классификация моделей по методологии применения	213
7.2.10. Классификация моделей по способу представления	214
7.3. Основные этапы компьютерного моделирования	216
7.3.1. Постановка задачи и анализ объекта моделирования	217
7.3.2. Разработка (формализация и синтез) модели	218
7.3.3. Проведение компьютерного эксперимента	218
7.3.4. Анализ результатов моделирования	219
7.4. Основы имитационного моделирования	219
7.4.1. Особенности имитационного моделирования	220
7.4.2. Подходы к имитационному моделированию	221
7.4.3. Этапы имитационного моделирования	222
7.5. Программные среды моделирования	224
7.5.1. Характеристика систем имитационного моделирования	224
7.5.2. AnyLogic — универсальная среда имитационного моделирования	227
7.5.3. Моделирование бизнес-процессов	228
Вопросы для самопроверки	232
Литература	234
Глава 8. Информационная безопасность	235
8.1. Основные положения, понятия и определения	236
8.2. Виды угроз в информационной сфере	237
8.3. Внутренние и внешние источники угроз	239
8.4. Организационно-правовое обеспечение информационной безопасности	240
8.5. Государственная система правового обеспечения защиты информации в Российской Федерации	241
8.6. Угрозы в информационных системах	243
8.6.1. Нарушения конфиденциальности	243
8.6.2. Нарушения достоверности	244
8.6.3. Нарушения целостности	244
8.6.4. Нарушения доступности	244
8.6.5. Классификация угроз информации	244
8.7. Защита информации в информационных системах	246
8.7.1. Требования к защите информации	246
8.7.2. Способы и средства защиты информации	249
8.7.3. Системы защиты информации	251
8.8. Государственные стандарты по информационной безопасности	253
8.8.1. Стандарты общего назначения	253
8.8.2. Стандарты по криптографической защите	256
8.8.3. Стандарты информационной безопасности в кредитно-финансовой сфере	256
Вопросы для самопроверки	257
Литература	257
Глава 9. Менеджмент информационной сферы	259
9.1. Становление информационного менеджмента	260
9.2. Основные понятия информационного менеджмента	261
9.3. Международные и российские стандарты в сфере информационного менеджмента	265
9.3.1. Стандарт ITIL	265
9.3.2. Стандарт CoBIT	268

9.3.3. Стандарт MOF	269
9.3.4. Стандарт ISO 20000:2005 и его отечественная адаптация — ГОСТ Р ИСО/МЭК 20000	270
9.3.5. Стандарт ISO/IEC 38500:2008	272
9.4. Информационный менеджмент как управление информационно-технологическими услугами	273
9.5. Информационный менеджмент как товар (аутсорсинг)	275
Вопросы для самопроверки	277
Литература	278

ЧАСТЬ ВТОРАЯ. ТЕХНИЧЕСКАЯ БАЗА ИНФОРМАТИКИ 279

Глава 10. Аппаратная часть компьютера	281
10.1. Обобщенная структура компьютера	281
10.2. Устройства ввода	284
10.3. Устройства вывода	288
10.3.1. Мониторы	288
10.3.2. Принтеры	291
10.3.3. Устройства вывода звука	294
10.3.4. Другие устройства вывода	295
10.4. Материнская плата	296
10.5. Память компьютера	296
10.5.1. Внутрипроцессорная память	297
10.5.2. Оперативная память	299
10.5.3. Постоянная память	300
10.5.4. Дисковая память и флэш-память	300
10.6. Центральный процессор	301
10.6.1. Классификация процессоров	301
10.6.2. Работа центрального процессора	302
10.6.3. Структура центрального процессора	303
10.6.4. Пути повышения производительности центрального процессора	305
10.7. Системный блок	306
Вопросы для самопроверки	308
Литература	309

Глава 11. Представление данных в компьютере	310
11.1. Форма представления данных в компьютере	310
11.1.1. Общее представление	310
11.1.2. Числовые данные	312
11.1.3. Символьные данные	314
11.1.4. Мультимедийные данные	317
11.1.5. Служебные данные	320
11.2. Системы счисления	320
11.2.1. Определение и классификация	320
11.2.2. Перевод из одной Р-ичной системы счисления в другую	323
11.2.3. Выполнение арифметических операций	326
Вопросы для самопроверки	328
Литература	329

Глава 12. История, состояние и тенденции развития компьютеров и вычислительных систем 330

12.1. История развития компьютеров	330
12.1.1. Механические устройства для вычислений	331
12.1.2. Электронные программируемые устройства	332

12.2. Классификация компьютеров и вычислительных систем	337
12.2.1. Классификация по принципу действия	337
12.2.2. Классификация по вычислительной мощности и габаритам	338
12.2.3. Классификация по способу применения	339
12.3. Настоящее и будущее компьютеров	342
12.3.1. Микропроцессоры	342
12.3.2. Суперкомпьютеры	346
12.3.3. Серверные компьютеры	350
12.3.4. Персональные компьютеры	352
12.3.5. Тенденции развития компьютеров	355
Вопросы для самопроверки	358
Литература	359
Глава 13. Основы построения компьютерных сетей	360
13.1. История появления и развития компьютерных сетей	361
13.2. Развитие сетевых межкомпьютерных коммуникаций в России	365
13.3. Классификация компьютерных сетей	366
13.3.1. Классификация по технологии передачи данных	367
13.3.2. Классификация по типу коммутации между узлами	367
13.3.3. Классификация по среде передачи данных	368
13.3.4. Классификация по территориальному охвату	369
13.3.5. Классификация по скорости передачи данных	370
13.3.6. Классификация по иерархической организации	371
13.4. Топология компьютерных сетей	371
13.4.1. Полносвязная топология	371
13.4.2. Общая шина	372
13.4.3. Звезда	372
13.4.4. Кольцо	373
13.4.5. Дерево	373
13.4.6. Смешанная топология	374
13.5. Физическая реализация среды передачи данных	375
13.5.1. Разделяемые среды передачи	375
13.5.2. Сетевые адаптеры	378
13.5.3. Концентраторы	379
13.5.4. Маршрутизаторы	379
13.6. Модели и протоколы компьютерных сетей	379
13.6.1. Общее представление	379
13.6.2. Стек протоколов TCP/IP	380
13.6.3. Сетевая модель OSI	382
13.7. Феномены века коммуникаций	382
13.7.1. Военно-техническая область	382
13.7.2. Финансовая сфера	383
13.7.3. Бизнес	383
13.7.4. География и навигация	384
13.7.5. Наука	384
13.7.6. Знание	384
13.7.7. Образование	384
13.7.8. Технологии	385
13.7.9. Общество	385
13.7.10. Здоровье и экология	386
Вопросы для самопроверки	387
Литература	387

ЧАСТЬ ТРЕТЬЯ. АЛГОРИТМИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАТИКИ	389
Глава 14. Основы теории алгоритмов	391
14.1. Представление об алгоритмах	391
14.1.1. Понятие алгоритма	391
14.1.2. Формализация понятия алгоритма посредством машины Поста	393
14.1.3. Формализация понятия алгоритма посредством машины Тьюринга	395
14.1.4. Современная теория алгоритмов	397
14.2. Способы записи алгоритмов	399
14.2.1. Словесный способ представления алгоритма	399
14.2.2. Графический способ записи алгоритма	400
14.2.3. Представление алгоритма с помощью диаграммы Нэсси—Шнейдермана	402
14.2.4. Представление алгоритма с помощью псевдокодов	404
14.2.5. Программный способ представления алгоритмов	405
14.3. Базовые алгоритмические конструкции	406
14.4. Представление и обработка данных разного типа	409
14.4.1. Общее представление о типах данных	409
14.4.2. Базовые типы данных	410
14.4.3. Представление и обработка данных в виде структур (массив, запись)	412
14.4.4. Представление и обработка данных в виде символьных цепочек	413
14.4.5. Представление и обработка данных в виде одно- и двухсвязных списков	415
14.4.6. Представление и обработка данных в виде деревьев	417
14.4.7. Представление и обработка данных в виде графов	420
14.5. Алгоритмы сортировки и поиска	425
14.5.1. Сортировка	425
14.5.2. Поиск	428
Вопросы для самопроверки	429
Литература	431
Глава 15. Классификация и тенденции развития программного обеспечения	432
15.1. Классификация программного обеспечения по степени взаимодействия с аппаратной частью компьютера	433
15.2. Классификация программного обеспечения по виду лицензирования	435
15.3. Прочие классификации	438
15.3.1. Классификация программного обеспечения по платформе назначения	438
15.3.2. Классификация программного обеспечения по способу взаимодействия с пользователем	439
15.3.3. Классификация программного обеспечения по отношению к сети	440
15.3.4. Классификация программного обеспечения по способу установки	440
15.3.5. Классификация программного обеспечения по целям разработки	440
15.3.6. Классификация программного обеспечения по аппаратной платформе	441
15.3.7. Классификация программного обеспечения по степени опасности	441
15.4. Промежуточное программное обеспечение	441
15.5. Программное обеспечение процесса разработки программного обеспечения	442
15.6. Области применения прикладного программного обеспечения	443
15.6.1. Офисные приложения	444
15.6.2. Программы управления проектами	448
15.6.3. Клиентские программы для работы с сервисами Интернета	448
15.6.4. Программы для научных исследований и расчетов	453
15.6.5. Образовательные программы	454
15.6.6. Программы для организации работы учебных заведений	456
15.6.7. Мультимедийные программы	458
15.6.8. Бухгалтерские программы	458
15.6.9. Программы для финансовых расчетов и прогнозирования	458

15.6.10. Программы для технического проектирования	459
15.6.11. Программы для бизнеса	459
15.7. Перспективы развития программного обеспечения	460
15.7.1. Ориентация на взаимодействие	460
15.7.2. Интеграция с сетью	460
15.7.3. Компьютеры без операционной системы	461
15.7.4. Распределенность	461
15.7.5. Автономизация и самообучение, повышение интеллектуальности программного обеспечения	461
15.7.6. Интеграция с бытовой техникой	462
15.7.7. Новые виды пользовательского интерфейса	462
15.7.8. Социальные вызовы развития программного обеспечения	462
Вопросы для самопроверки	463
Литература	464
Глава 16. Системное программное обеспечение компьютера	465
16.1. Состав системного программного обеспечения	466
16.1.1. BIOS	466
16.1.2. Ядро операционной системы	467
16.1.3. Драйверы устройств	468
16.1.4. Службы операционной системы	468
16.1.5. Системные оболочки	468
16.1.6. Инструменты администрирования	469
16.1.7. Инструменты системного программирования	469
16.2. Классификация операционных систем	470
16.3. Сферы применения операционных систем	474
16.4. Представление о структуре операционной системы	476
16.4.1. Операционная система в виде набора функций	476
16.4.2. Системные службы операционной системы	479
16.4.3. Представление операционной системы в виде слоев	480
16.5. Операционные системы семейства Windows	481
16.5.1. Подсемейство Windows 3x	481
16.5.2. Подсемейство Windows 9x	481
16.5.3. Подсемейство Windows NT/2000/XP/2003	482
16.6. Операционная система Windows XP	483
16.6.1. Графический пользовательский интерфейс	483
16.6.2. Файловые системы	484
16.6.3. Работа с локальной сетью	484
16.6.4. Работа с Интернетом	486
16.6.5. Мультимедийные возможности	488
16.6.6. Обеспечение безопасности	488
16.6.7. Средства коллективной работы	490
16.6.8. Средства настройки и администрирования	491
16.6.9. Приложения из состава ОС	491
16.7. Операционная система Alt Linux	491
16.7.1. История появления свободной операционной системы Linux	491
16.7.2. Графический пользовательский интерфейс	492
16.7.3. Файловые системы	494
16.7.4. Работа с сетью	495
16.7.5. Работа с Интернетом	497
16.7.6. Мультимедийные возможности	497
16.7.7. Обеспечение безопасности	498
16.7.8. Средства коллективной работы	499
16.7.9. Средства настройки и администрирования	499
16.7.10. Приложения из состава операционной системы	500

Вопросы для самопроверки	500
Литература	501
Глава 17. Офисное программное обеспечение	502
17.1. Текстовый процессор	503
17.1.1. Общее представление о функциональности	503
17.1.2. Сравнительная характеристика текстовых процессоров Microsoft Word, OpenOffice.org Writer и Abiword	508
17.2. Табличный процессор	511
17.2.1. Общее представление о функциональности	511
17.2.2. Дополнительные возможности табличного процессора	513
17.2.3. Сравнительная характеристика табличных процессоров Microsoft Excel, OpenOffice.org Calc и Gnumeric	516
17.3. Органайзер	517
17.3.1. Общие принципы использования программных инструментов организации деятельности	517
17.3.2. Сравнительная характеристика органайзеров Microsoft Outlook, doOrganizer, KOrganizer и Mozilla Sunbird	518
17.4. Система электронного документооборота	520
17.4.1. Назначение и функциональность	520
17.4.2. Сравнительная характеристика систем электронного документооборота DIRECTUM, Digital Design, ЕВФРАТ-Документооборот	521
Вопросы для самопроверки	523
Литература	523
Глава 18. Средства мультимедиа	524
18.1. Представление о мультимедиа	524
18.2. Звук	526
18.2.1. Разновидности звуковых данных в компьютере	526
18.2.2. Сжатие звука	528
18.2.3. Устройства для получения и воспроизведения звука	528
18.2.4. Программные продукты для обработки и воспроизведения звука	530
18.3. Изображения	535
18.3.1. Цифровые форматы изображений	535
18.3.2. Сжатие изображений	536
18.3.2. Получение изображений	539
18.3.3. Программы для работы с изображениями	539
18.4. Видео	543
18.4.1. Кодирование видеосигналов	543
18.4.2. Сжатие видео	543
18.4.3. Программы для обработки и воспроизведения видео	544
18.5. Мультимедиа-презентации	546
18.5.1. Возможные типы мультимедиа-презентаций	546
18.5.2. Программные технологии создания мультимедиа-презентаций	547
18.5.3. Демонстрация мультимедиа-презентаций	550
Вопросы для самопроверки	550
Литература	551
Глава 19. Технологии и инструменты программирования	552
19.1. Основные понятия и классификация языков программирования	553
19.1.1. Основные понятия	553
19.1.2. Классификация языков программирования	557
19.2. Краткая история языков программирования	559
19.2.1. Первый этап — машинные коды	559

19.2.2. Второй этап — языки высокого уровня	560
19.2.3. Третий этап — структурное программирование	560
19.2.4. Четвертый этап — модульное программирование	561
19.2.5. Пятый этап — объектно-ориентированный подход	561
19.2.6. Шестой этап — компонентный подход	561
19.2.7. Седьмой этап — архитектура, управляемая моделью	562
19.3. Концепция объектно-ориентированного программирования	562
19.3.1. Абстракция	562
19.3.2. Наследование	563
19.3.3. Полиморфизм	564
19.3.4. Инкапсуляция	564
19.4. Инструментальные средства и среды разработки программного обеспечения ..	565
19.5. Жизненный цикл программного обеспечения ..	568
19.5.1. Анализ требований и разработка спецификаций	569
19.5.2. Проектирование	570
19.5.3. Кодирование и тестирование	571
Вопросы для самопроверки	572
Литература	573

Предисловие

Прошло относительно немного времени, как в школах и вузах страны начали изучать информатику. Сейчас она претендует на звание базовой дисциплины в системе высшего образования и в комплексе с другими классическими дисциплинами (математикой, физикой, химией, естествознанием, биологией, историей) призвана создавать фундамент профессионального образования в вузе.

Стремительное развитие компьютерных технологий, рост объемов обрабатываемой информации и проникновение информационных технологий практически во все области жизни сделали изучение информатики в учебных заведениях. Наступило время, когда профессионал — юрист, инженер, экономист, социолог, журналист — уже с трудом справляется с потоками информации. Специалисту, для того чтобы на должном уровне выполнять свои обязанности, необходимы инструментарий и методика его применения для обработки информации. Это сравнимо с использованием средств передвижения: теоретически человек может пешком преодолеть любое расстояние, но современный темп жизни просто невыносим без применения автомобиля, поезда, самолета и т. д. То же самое происходит и в области, связанной с обработкой информации: теоретически человек сам может переработать без компьютера любую информацию, но сделает это эффективнее, если овладеет знаниями и умениями, которыми располагает информатика.

Однако темпы самоопределения информатики как науки, ее размежевание с теми дисциплинами и науками, на базе которых она рождалась, были значительно более медленными, чем темпы развития техники и технологии. Это привело к тому, что сегодня нет единого, общепринятого определения и взгляда на информатику как науку — ни в России, ни за рубежом. Именно сегодня этот взгляд вырабатывается, идет рефлексивный процесс самоопределения.

Разные взгляды на область знаний, названную Информатикой, и на ее границы определили и разные подходы к формированию содержания курса в вузе. В настоящее время существует несколько примерных программ по дисциплине «Информатика», и каждое учебное заведение, ориентируясь на одну из них, проводит обучение в соответствии с потребностями своей предметной области.

В связи с этим перед авторами учебника встала непростая задача — определить, что и в каком объеме нужно знать студенту по базовой дисциплине «Информатика».

Решение задачи велось в двух направлениях:

- отбор содержания и выделение наиболее значимых тем и аспектов, которые могли бы составить ядро дисциплины;
- разработка методики изложения и структурирования материала.

Известно мнение широкого круга специалистов по компьютерной технике, информационным системам и технологиям, что информатика — это, скорее, дисциплина практического направления, где студент приобретает навыки работы на персональном компьютере в наиболее распространенных программных средах, и вопрос состоит только в том, какие из них выбрать. Если ориентироваться на такой подход, то, учитывая высокую динамику замены одних программных продуктов и технических средств другими, вряд ли когда-нибудь удастся своевременно создать учебник по этой дисциплине.

По мнению авторов, *информатика служит прежде всего для формирования мировоззрения в информационной сфере и определенного уровня информационной культуры*, то есть умения целенаправленно работать с информацией, профессионально используя для ее получения, обработки и передачи компьютерную информационную технологию и соответствующие ей технические и программные средства.

Учебник по информатике содержит базовые теоретические и практические сведения, которыми должен владеть любой профессионал независимо от рода его деятельности. Учебник состоит из трех частей, отражающих следующие аспекты информатики:

- *информатика как область интеграции знаний* о роли информации в развитии общества, об управлении знаниями, о логических основах построения компьютера и обработки информации, о методах и средствах моделирования информационных объектов, об основах построения баз данных и информационных системах, об информационной безопасности и новом направлении — менеджменте информационной сферы;
- *техническая база информатики*, где формируется представление об аппаратной части компьютера и о формах представления в нем данных, излагаются основы построения компьютерных сетей, приводится история, состояние и тенденции развития компьютерных систем;
- *алгоритмическое и программное обеспечение информатики*, где излагаются основы теории алгоритмов, технологии и инструменты программирования, приводится классификация программного обеспечения, рассматриваются характеристики и особенности разных классов программных продуктов.

Цель учебника:

- дать целостное представление об информатике и ее роли в развитии общества;
- раскрыть суть и возможности технических и программных средств информатики;
- сформировать понимание того, с какой целью и каким образом можно использовать информационные системы и технологии.

Как работать с учебником?

Учебник построен таким образом, чтобы активизировать познавательную деятельность студента. Каждая глава учебника написана как автономная часть, и при ее изучении необязательно знать содержание предыдущих глав. Вы можете читать учебник с любой интересующей вас темы. Глава начинается с формулировки цели изучения. Далее перечислены основные темы предмета, обсуждаемые в данной главе. Чтобы обучение было эффективным, студент должен, прочитав главу, хорошо понять смысл этих тем. Для уверенности в том, что изучение главы прошло успешно и освоил весь материал, постарайтесь ответить на все вопросы для самопроверки, которые приведены в конце каждой главы. Отвечая на них, обращайтесь вновь к учебнику.

Завершается каждая глава списком рекомендованной к изучению литературы. Чтение этих книг поможет освоить и расширить знания по изучаемым проблемам информатики. А знать ее крайне необходимо, так как человек, режиссирующий профессионально работать в избранной отрасли, должен систематически обрабатывать информацию, применяя информационные технологии. Надеемся, что в реализации этих задач сыграет свою роль предлагаемый учебник.

От всей души желаем вам успехов!

Часть первая

Информатика как область интеграции знаний

Глава 1. Представление об информации	17
Глава 2. Роль информации в развитии общества	45
Глава 3. Теоретические основы управления знаниями	76
Глава 4. Логические основы информатики	108
Глава 5. Информационные системы и технологии	125
Глава 6. Теория баз данных	169
Глава 7. Основы моделирования	202
Глава 8. Информационная безопасность	235
Глава 9. Менеджмент информационной сферы	259

Глава 1 Представление об информации

- 1.1. Понятие информации
- 1.2. Меры информации
- 1.3. Качество информации
- 1.4. Информационные процессы
- 1.5. Кодирование при передаче и хранении информации
- 1.6. Основы классификации и структурирования информации
- 1.7. Кодирование при классификации информации

В этой главе излагаются основные сведения об информации, дается определение понятию информации, описываются основные свойства информации, поясняется, в чем различие между понятиями информации и данных. Большое место в главе отводится вопросам измерения количества информации и способам оценки ее качества. Информационные процессы, составляющие неотъемлемую часть современной деятельности человека, рассматриваются в этой главе совместно с вопросами кодирования информации.

1.1. Понятие информации

1.1.1. Определение информации

Термин *«информация»* происходит от латинского *«informatio»*, что означает *«разъяснение»*, *«осведомление»*, *«изложение»*.

Существует множество определений информации. Так, один из основоположников современной теории информации, Роберт Винер, определял информацию так: «Информация есть информация, а не материя или энергия».

Такое определение через отрицание кажется достаточно полным и универсальным, но применить его в качестве инструмента для построения научной методологии практически невозможно.

В то же время и современной технике получили широкое распространение методологические подходы, позволяющие применять понятие информации и предложенные инструменты для исследования процессов, происходящих в технических системах, экономике, социуме, в живой и неживой природе.

Наиболее известный среди таких подходов — математическая теория Клода Шеннона, позволяющая вероятностно обосновать надежность передачи сигналов по линии связи. В подходе Шеннона информация — это мера снижения неопределенности системы.

Существует также термодинамический (энергетический) подход, рассматривающий информацию как способ уменьшения энтропии системы.

Советским математиком Колмогоровым был предложен алгоритмический подход, позволяющий оценить информацию по сложности алгоритма, необходимого для ее обработки. Все эти подходы тесно связывали понятие информации со сферой применения.

С позиции материалистической философии информация есть отражение реального мира с помощью сведений (сообщений). Сообщение — это форма представления информации в виде речи, текста, изображения, цифровых данных, графиков, таблиц и т. п. В широком смысле информация — это общенаучное понятие, включающее в себя обмен сведениями между людьми, обмен сигналами между живой и неживой природой, людьми и устройствами.

Информация — это сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии, которые уменьшают степень неопределенности и неполноты имеющихся о них знаний.

1.1.2. Информация и данные

Информатика рассматривает информацию как концептуально связанные между собой сведения, понятия, изменяющие наши представления о явлении или объекте окружающего мира. Наряду с информацией в информатике часто употребляется понятие данных. Покажем, в чем их различие.

Данные могут рассматриваться как признаки или записанные наблюдения, которые по каким-то причинам не используются, а только хранятся. В том случае, если появляется возможность использовать эти данные для уменьшения неопределенности знаний о чем-либо, данные превращаются в информацию.

Данные — это информация, закодированная определенным образом с целью передачи, обработки, хранения, поиска или извлечения.

Пример. Напишите на листе бумаги десять номеров телефонов в виде последовательности десяти чисел и покажите их вашему другу. Он воспримет эти цифры как данные, так как они не предоставляют ему никаких сведений. Затем против каждого номера укажите название фирмы и род деятельности. Для вашего друга непонятные цифры обретут определенность и превратятся из данных в информацию, которую он в дальнейшем мог бы использовать.

1.1.3. Адекватность информации

При работе с информацией всегда имеются ее *источник* и *потребитель* (получатель). Пути и процессы, обеспечивающие передачу сообщений от источника информации к ее потребителю, называются *информационными коммуникациями*.

Для потребителя информации очень важной характеристикой является ее адекватность.

Адекватность информации — это определенный уровень соответствия создаваемого с помощью полученной информации образа реальному объекту, процессу, явлению и т. п.

В реальной жизни вряд ли возможна ситуация, когда вы сможете рассчитывать на полную адекватность информации. Всегда присутствует некоторая степень неопределенности. От степени адекватности информации реальному состоянию объекта или процесса зависит правильность принятия решений человеком.

Адекватность информации может выражаться в трех формах: семантической, синтаксической, прагматической.

Синтаксическая адекватность отображает формально-структурные характеристики информации и не затрагивает ее смыслового содержания. На синтаксическом уровне учитываются тип носителя и способ представления информации, скорость передачи и обработки, размеры кодов представления информации, надежность и точность преобразования этих кодов и т. п. Информацию, рассматриваемую только с синтаксических позиций, обычно называют данными, так в этой форме смысловая сторона не имеет значения. Эта форма способствует восприятию внешних структурных характеристик, то есть синтаксической стороны информации.

Семантическая (смысловая) адекватность определяет степень соответствия образа объекта и самого объекта. Семантический аспект предполагает учет смыслового содержания информации. На этом уровне анализируются те сведения, которые отражает информация, рассматриваются смысловые связи. В информатике устанавливаются смысловые связи между кодами представления информации. Эта форма служит для формирования понятий и представлений, выявления смысла, содержания информации, для ее обобщения.

Прагматическая (потребительская) адекватность отражает отношение информации и ее потребителя, соответствие информации цели управления, которая на ее основе реализуется. Проявляются прагматические свойства информации только при наличии единства информации (объекта), пользователя и цели управления. Прагматический аспект рассмотрения связан с ценностью, полезностью использования информации при выработке потребителем решения для достижения своей

цели. С этой точки зрения анализируются потребительские свойства информации. Эта форма адекватности непосредственно связана с практическим использованием информации, с соответствием ее целевой функции деятельности системы.

1.2. Меры информации

Для измерения информации вводятся два параметра: количество информации I и объем данных V_d .

Эти параметры имеют разные выражения и интерпретацию в зависимости от рассматриваемой формы адекватности. Каждой форме адекватности соответствует своя мера количества информации и объема данных (рис. 1.1).

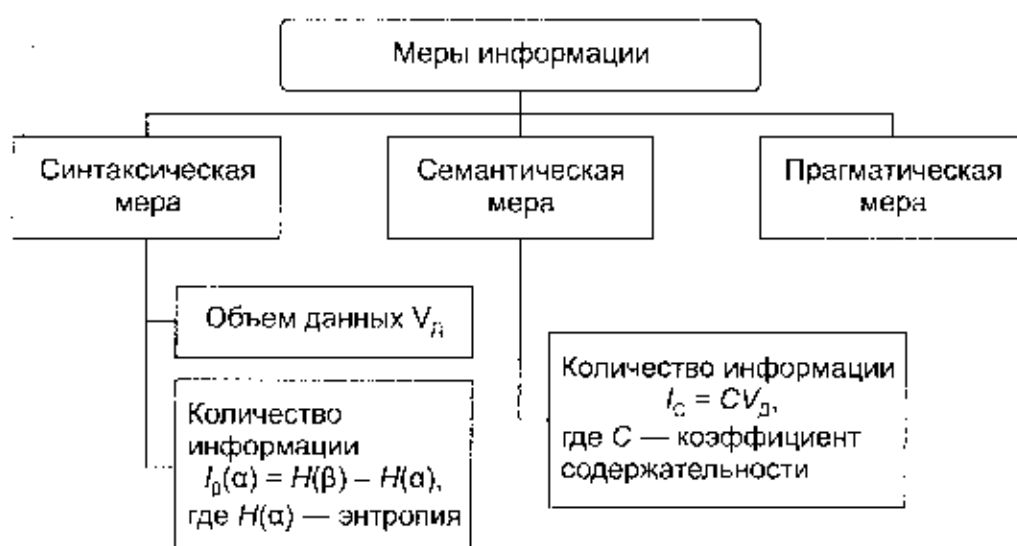


Рис. 1.1. Меры информации

1.2.1. Синтаксическая мера информации

Синтаксическая мера количества информации оперирует обезличенной информацией, не выражающей смыслового отношения к объекту.

Объем данных V_d в сообщении измеряется количеством символов (разрядов) в этом сообщении. В различных системах счисления один разряд имеет различный вес и соответственно меняется единица измерения данных:

- в двоичной системе счисления единица измерения — бит (bit, или binary digit, — двоичный разряд);
- в десятичной системе счисления единица измерения — дит (десятичный разряд).

ПРИМЕЧАНИЕ

В современных ЭВМ наряду с минимальной единицей измерения данных — битом — широко используется укрупненная единица измерения — байт, равная 8 бит.

Пример. Сообщение в двоичной системе в виде восьмиразрядного двоичного кода 10111011 имеет объем данных $V_d = 8$ бит.

Сообщение в десятичной системе в виде шестиразрядного числа 275903 имеет объем данных $V_d = 6$ диг.

Количество информации I на синтаксическом уровне невозможно определить без рассмотрения понятия неопределенности состояния системы (энтропии системы). Действительно, получение информации о какой-либо системе всегда связано с изменением степени неосведомленности получателя о состоянии этой системы. Рассмотрим это понятие.

Пусть до получения информации потребитель имеет некоторые предварительные (априорные) сведения о системе α . Мерой его неосведомленности о системе является функция $H(\alpha)$, которая в то же время служит и мерой неопределенности состояния системы.

После получения некоторого сообщения β получатель приобретает некоторую дополнительную информацию $I_\beta(\alpha)$, уменьшающую его априорную неосведомленность так, что апостериорная (после получения сообщения β) неопределенность состояния системы становится равной $H_\beta(\alpha)$.

Тогда количество информации $I_\beta(\alpha)$ о системе, полученной в сообщении β , определится так:

$$I_\beta(\alpha) = H(\alpha) - H_\beta(\alpha),$$

то есть количество информации измеряется изменением (уменьшением) неопределенности состояния системы.

Если конечная неопределенность $H_\beta(\alpha)$ обратится в нуль, то первоначальное неполное знание заменится полным знанием и количество информации:

$$I_\beta(\alpha) = H(\alpha)$$

Иными словами, энтропия системы $H(\alpha)$ может рассматриваться как мера недостающей информации.

Энтропия системы $H(\alpha)$, имеющая N возможных состояний, согласно формуле Шеннона, равна:

$$H(\alpha) = - \sum_{i=1}^N P_i \log P_i.$$

Здесь P_i — вероятность того, что система находится в i -м состоянии.

Для случая, когда все состояния системы равновероятны, то есть их вероятности равны, сс $P_i = \frac{1}{N}$ энтропия определяется соотношением

$$H(\alpha) = - \sum_{i=1}^N \frac{1}{N} \log \frac{1}{N}.$$

Часто информация кодируется числовыми кодами в той или иной системе счисления, особенно это актуально при представлении информации в компьютере. Естественно, что одно и то же количество разрядов в разных системах счисления может передать разное количество состояний отображаемого объекта, что можно представить в виде соотношения

$$N = m^n.$$

Здесь N – количество всевозможных отображаемых состояний; m – основание системы счисления (разнообразие символов, применяемых в алфавите); n – количество разрядов (символов) в сообщении.

Пример. По каналу связи передается n разрядное сообщение, использующее m различных символов. Так как количество всевозможных кодовых комбинаций будет $N = m^n$, то при равной вероятности появления любой из них количество информации, приобретенной абонентом в результате получения сообщения, окажется равным

$$I = \log N = n \log m.$$

Это формула Хартли.

Если в качестве основания логарифма принять m , то $I = n$. В данном случае количество информации (при условии полного априорного незнания абонентом содержания сообщения) будет равно объему данных, полученных по каналу связи ($I = V_d$). Для неравновероятных состояний системы всегда $I < V_d = n$.

Наиболее часто используются двоичные и десятичные логарифмы. Единицами измерения в этих случаях являются *бит* и *дит* соответственно.

Коэффициент (степень) информативности (законоичность) сообщения определяется отношением количества информации к объему данных:

$$Y = \frac{I}{V_d}, \text{ причем } 0 < Y < 1.$$

С увеличением Y уменьшаются объемы работы по преобразованию информации (данных) в системе. Поэтому стремятся к повышению информативности, для чего разрабатываются специальные методы оптимального кодирования информации.

1.2.2. Семантическая мера информации

Для измерения смыслового содержания информации, то есть ее количества на семантическом уровне, наибольшее признание получила тезаурусная мера, которая связывает семантические свойства информации со способностью пользователя принимать поступившее сообщение. Для этого используется понятие тезауруса пользователя.

Тезаурус — это совокупность сведений, которыми располагает пользователь или система.

В зависимости от соотношений между смысловым содержанием информации S и тезаурусом пользователя S_p изменяется количество семантической информации I_c воспринимаемой пользователем и включаемой им в дальнейшем в свой тезаурус. Характер такой зависимости показан на рис. 1.2. Рассмотрим два предельных случая, когда количество семантической информации I_c равно 0:

- при $S_p \sim 0$ пользователь не воспринимает и не понимает поступающую информацию;
- при $S_p \rightarrow \infty$ пользователь все знает, и поступающая информация ему не нужна.

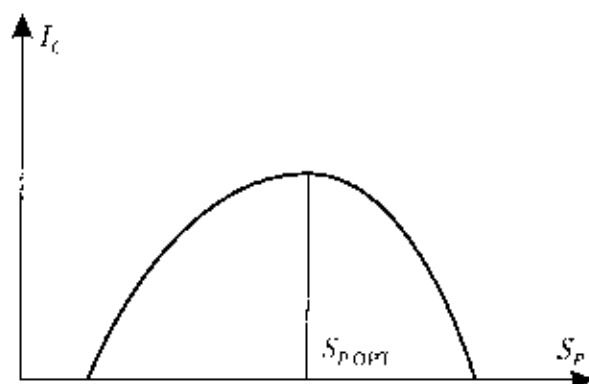


Рис. 1.2. Зависимость количества информации, воспринимаемой пользователем, от его тезауруса $I_c = f(S_p)$

Максимальное количество семантической информации I_c потребитель приобретает при согласовании ее смыслового содержания S со своим тезаурусом S_p ($S_p = S_{p\text{ опт}}$), когда поступающая информация понятна пользователю и несет ему ранее неизвестные (отсутствующие в его тезаурусе) сведения.

Следовательно, количество семантической информации в сообщении, то есть количество новых знаний, получаемых пользователем, является величиной относительной. Одно и то же сообщение может иметь смысловое содержание для компетентного пользователя и быть бессмысленным (семантический шум) для пользователя некомпетентного.

При оценке семантического (содержательного) аспекта информации необходимо стремиться к согласованию величин S и S_p .

Относительной мерой количества семантической информации может служить коэффициент содержательности C , который определяется как отношение количества семантической информации к ее объему:

$$C = \frac{I_c}{V_d}.$$

1.2.3. Прагматическая мера информации

Прагматическая мера определяет полезность (ценность) информации для достижения пользователем поставленной цели. Эта мера также величина относительная, обусловленная особенностями применения этой информации в той или иной

системе. Ценность информации целесообразно измерять в тех же самых единицах (или близких к ним), в которых измеряется целевая функция.

Пример. В экономической системе прагматические свойства (ценность) информации можно определить приростом экономического эффекта функционирования, достигнутым благодаря использованию этой информации для управления системой:

$$I_{\pi\beta(\gamma)} = H\left(\frac{\gamma}{\beta}\right) - H(\gamma).$$

Здесь $I_{\pi\beta(\gamma)}$ — ценность информационного сообщения β для системы управления γ ; $H(\gamma)$ — априорный ожидаемый экономический эффект функционирования системы управления γ ; $H\left(\frac{\gamma}{\beta}\right)$ — ожидаемый эффект функционирования системы γ при условии, что для управления будет использована информация, содержащаяся в сообщении β .

Для сопоставления введенные меры информации представим в табл. 1.1.

Кроме трех перечисленных иногда выделяется еще один класс информационных мер, структурный, в который включают геометрическую, комбинаторную и адаптивную меры.

Адаптивная мера (мера Хартли) уже была рассмотрена в составе синтаксической меры.

Геометрическая мера предполагает измерение параметра геометрической модели информационного сообщения (длины, площади, объема) в дискретных единицах. Например, геометрической моделью информации может быть линия единичной длины, квадрат или куб. Максимально возможное количество информации в заданных структурах определяет информационную емкость модели (систему), которая описывается как сумма дискретных значений по всем измерениям.

В *комбинаторной мере* количество информации определяется как число комбинаций элементов (символов). Возможное количество информации совпадает с числом возможных сочетаний, перестановок или размещений элементов. Комбинирование символов в словах, состоящих только из 0 и 1, меняет значение слов.

Таблица 1.1. Единицы измерения информации и примеры

Мера информации	Единицы измерения	Примеры (для компьютерной области)
Синтаксическая	Степень уменьшения неопределенности (шенноновский подход)	Вероятность события
	Единицы представления информации (компьютерный подход)	Бит, байт, Кбайт и т. д.
Семантическая	Тезаурус	Пакет прикладных программ, персональный компьютер, компьютерные сети и т. д.
	Экономические показатели	Рентабельность, производительность, коэффициент амортизации и т. д.

Мера информации	Единицы измерения	Примеры (для компьютерной области)
Прагматическая	Ценность использования	Емкость памяти, производительность компьютера, скорость передачи данных и т. д. Денежное выражение Время обработки информации и принятия решений

1.3. Качество информации

Под *качеством* информации понимается такая совокупность свойств, которая обуславливает ее способность удовлетворять определенные потребности людей. Основными потребительскими показателями качества информации являются: репрезентативность, содержательность, достаточность, доступность, актуальность, своевременность, точность, достоверность, устойчивость.

Репрезентативность информации связана с правильностью ее отбора и формирования в целях адекватного отражения свойств объекта. Важнейшее значение здесь имеют

- правильность концепции, на базе которой сформулировано исходное понятие;
- обоснованность отбора существенных признаков и связей отображаемого явления.

Нарушение репрезентативности информации приводит нередко к существенным ее погрешностям.

Содержательность информации отражает семантическую емкость, равную отношению количества семантической информации в сообщении к объему обрабатываемых данных:

$$C = \frac{I_c}{V_d}.$$

С увеличением содержательности информации растет семантическая пропускная способность информационной системы, так как для получения одних и тех же сведений требуется преобразовать меньший объем данных.

Наряду с *коэффициентом содержательности* C , отражающим семантический аспект, можно использовать и *коэффициент информативности* Y , характеризующийся отношением количества синтаксической информации (по Шеннону) к объему данных:

$$Y = \frac{I}{V_d}.$$

Достаточность (полнота) информации означает, что ее состав (набор показателей) минимален, но достаточен для принятия правильного решения. Понятие полноты информации связано с ее смысловым содержанием (семантикой) и прагматикой. Как неполная, то есть недостаточная для принятия правильного решения, так и избыточная информация снижает эффективность принимаемых пользователем решений.

Доступность информации восприятию пользователя обеспечивается выполнением соответствующих процедур ее получения и преобразования. Например, в информационной системе информация преобразуется к доступной и удобной для восприятия пользователем форме. Это достигается, в частности, и путем согласования ее семантической формы с тезаурусом пользователя.

Актуальность информации определяется степенью сохранения ценности информации для управления на момент использования и зависит от динамики изменения ее характеристик, а также от интервала времени, прошедшего с момента возникновения данной информации.

Своевременность информации означает ее поступление не позже заранее назначенного момента времени, согласованного с временем решения поставленной задачи.

Точность информации определяется степенью близости получаемой информации к реальному состоянию объекта, процесса, явления и т. п. Для информации, отображаемой цифровым кодом, известны четыре классификационных понятия точности:

- формальная точность измеряется значением единицы младшего разряда числа;
- реальная точность определяется значением единицы последнего разряда числа, верность которого гарантируется;
- максимальная точность представляет собой точность, которую можно получить в конкретных условиях функционирования системы;
- необходимая точность определяется функциональным назначением показателя.

Достоверность информации определяется ее свойством отражать реально существующие объекты с необходимой точностью. Измеряется достоверность информации доверительной вероятностью необходимой точности, то есть вероятностью того, что отображаемое информацией значение параметра отличается от истинного значения этого параметра в пределах необходимой точности.

Устойчивость информации отражает ее способность реагировать на изменения исходных данных без нарушения необходимой точности. Устойчивость информации, как и репрезентативность, обусловлена выбранной методикой ее отбора и формирования.

В заключение следует отметить, что такие параметры качества информации, как репрезентативность, содержательность, достаточность, доступность, устойчивость, целиком определяются на методическом уровне разработки информационных систем. Параметры актуальности, своевременности, точности и достоверности также в большей степени обуславливаются на методическом уровне, однако на их величину существенно влияет еще и характер функционирования системы, в первую очередь ее надежность. При этом параметры актуальности и точности жестко связаны, соответственно, с параметрами своевременности и достоверности.

1.4. Информационные процессы

Процессы, связанные с поиском, хранением, передачей, обработкой и использованием информации, называются *информационными*.

Поиск информации — это процесс извлечения хранимой информации.

Методы поиска информации:

- ☐ непосредственное *наблюдение*;
- ☐ *общение* со специалистами по интересующему вас вопросу;
- ☐ *чтение* соответствующей литературы;
- ☐ *просмотр* видео, телепрограмм;
- ☐ *прослушивание* радиопередач, аудиокассет;
- ☐ *работа в библиотеках и архивах*;
- ☐ *запрос к информационным системам*, в том числе базам и банкам компьютерных данных;
- ☐ *поиск информации в Интернете*;
- ☐ другие методы.

Сбор информации — это деятельность субъекта, в ходе которой он получает информацию об интересующем его объекте.

Хранение информации — это процесс поддержания исходной информации в виде, обеспечивающем выдачу данных по запросам конечных пользователей в установленные сроки.

Способ хранения информации зависит от ее носителя (книга — библиотека, картина — музей, фотография — альбом). ЭВМ может рассматриваться как устройство для компактного хранения информации с возможностью быстрого доступа к ней.

Передача (обмен) информацией — это процесс, в ходе которого передатчик (источник) передает информацию, а получатель (приемник) ее принимает.

В процессе передачи информации обязательно участвуют *источник* и *приемник* информации. Между источником и приемником действует канал передачи информации — канал связи. *Канал связи* — совокупность технических устройств, обеспечивающих передачу сигнала от источника к получателю. Кодировующее устройство — устройство, предназначенное для преобразования исходного сообщения источника к виду, удобному для передачи. Декодировующее устройство — устройство для преобразования кодированного сообщения в исходное (рис. 1.3). Деятельность людей всегда связана с передачей информации. В процессе передачи информация может теряться и искажаться, в качестве примеров можно привести искажение звука в телефоне, атмосферные помехи в радио, искажение или затемнение изображения в телевидении, ошибки при передаче в телеграфе.



Рис. 1.3. Передача информации по каналу связи

Каналы передачи сообщений характеризуются пропускной способностью и помехозащищенностью. Каналы передачи данных делятся на симплексные (с передачей информации в одну сторону, например телевидение) и дуплексные (по которым можно передавать информацию в оба направления, например телефон, телеграф). По каналу могут одновременно передаваться несколько сообщений. Каждое из этих сообщений выделяется (отделяется от других) с помощью специальных фильтров. Например, возможна фильтрация по частоте передаваемых сообщений, как это делается в радиоканалах. Пропускная способность канала определяется максимальным количеством символов, передаваемых по нему в отсутствие помех. Эта характеристика зависит от физических свойств канала. Для повышения помехозащищенности канала используются специальные методы передачи сообщений, уменьшающие влияние шумов. Например, вводят лишние символы. Эти символы не имеют реального содержания, но используются для контроля правильности сообщения при получении. С точки зрения теории информации избыточно все то, что делает литературный язык красочным, гибким, богатым оттенками, многоплановым, многозначным.

Обработка информации — это упорядоченный процесс ее преобразования в соответствии с алгоритмом решения задачи или с другими формальными правилами.

После решения задачи обработки информации результат должен быть выдан конечным пользователям в требуемом виде. Эта операция реализуется в ходе решения задачи выдачи информации. Выдача информации, как правило, производится с помощью внешних устройств ЭВМ в виде текстов, таблиц.

Защита информации в более узком смысле понимается как предотвращение доступа к информации лицам, не имеющим соответствующего разрешения (несанкционированный, нелегальный доступ), непредумышленного или недозволенного использования, изменения или разрушения информации.

Защита информации (в широком смысле) — это комплекс организационных, правовых и технических мер по предотвращению угроз информационной безопасности и устранению их последствий.

Наиболее эффективным средством организации информационных процессов является *информационная система*, оснащенная средствами ввода, поиска, размещения, обработки и выдачи информации. Наличие таких средств — главная особенность информационных систем, отличающих их от простых скоплений информационных материалов. Например, личная библиотека, в которой может ориентироваться только ее владелец, информационной системой не является.

В публичных же библиотеках порядок размещения книг всегда строго определен. Благодаря ему, поиск и выдача книг, а также размещение новых поступлений реализованы в виде стандартных, формализованных процедур.

1.5. Кодирование при передаче и хранении информации

Путем кодирования производится отображение одного набора знаков в другой набор знаков. Отображаемый набор знаков называется исходным алфавитом, а набор знаков, который используется для отображения, — кодовым алфавитом. Для построения кода применяются как отдельные символы кодового алфавита, так и их комбинации.

Совокупность символов кодового алфавита, применяемых для кодирования одного символа или одной комбинации символов исходного алфавита, называется кодом символа.

Взаимосвязь символов или комбинаций символов исходного алфавита с их кодовыми комбинациями составляет таблицу кодов (таблицу соответствия).

Для получения исходных символов по кодам символов применяют процедуру декодирования. При этом обязательно соблюдение условия: код должен быть однозначным, то есть одному исходному знаку должен соответствовать точно один код и наоборот.

Кодирование — это представление сигнала в определенной форме.

Декодирование — это получение исходных символов по кодам символов.

Виды кодирования зависят от поставленной цели.

- ❑ **Кодирование по образцу** — каждый знак дискретного сигнала представляется знаком или набором знаков того алфавита, в котором выполняется кодирование. Кодирование по образцу используется, например, для ввода информации в компьютер с целью ее внутреннего представления.

Пример. Для перевода символов, вводимых с клавиатуры, в числовой код, хранящийся в памяти компьютера, используется кодовая таблица ASCII (American Standard Code for Information Interchange — американский стандартный код для обмена информацией), в которой каждому символу алфавита, а также множеству специальных управляющих команд соответствует числовой код.

- ❑ **Криптографическое кодирование, или шифрование**, используется тогда, когда нужно защитить информацию от несанкционированного доступа. Существует два основных широко применяющихся сегодня способа криптографического кодирования: симметричное кодирование с закрытым ключом и асимметричное кодирование с открытым ключом. При симметричном кодировании с закрытым

ключом для кодирования и декодирования данных применяется один и тот же ключ. Этот ключ должен быть по безопасным каналам доставлен стороне, осуществляющей декодирование, что делает шифрование с симметричным ключом уязвимым. Напротив, при шифровании с асимметричным ключом сторона, осуществляющая декодирование, публикует так называемый *открытый ключ* (public key), который применяется для кодирования сообщений, а декодирование осуществляется другим — *закрытым ключом* (private key), известным только принимающей стороне. Такая схема делает асимметричный способ кодирования высоконадежным. По этой причине в последнее время он приобрел массовую популярность.

Пример. Во множестве шпионских фильмов-боевиков основным вопросом при захвате агента противника было получение ключей к шифрам. Получение ключа давало возможность прочесть все перехваченные ранее сообщения и сразу получить множество полезной информации. Но эта возможность достижима только тогда, когда речь идет о симметричных ключах. Получение публичного асимметричного ключа в этом смысле не дает никаких преимуществ, поскольку открытый ключ позволяет кодировать сообщения, но не может применяться для их декодирования.

□ *Оптимальное кодирование* служит для устранения избыточности данных путем снижения среднего числа символов кодового алфавита, предназначенных для представления одного исходного символа. Оптимальное кодирование обычно используется в архиваторах. В большинстве современных программ, предназначенных для оптимального кодирования (сжатия) данных, применяются одни и те же базовые алгоритмы:

○ *Статистическое кодирование* базируется на предварительном вычислении частоты повторения одних и тех же кодов в сообщении и составлении таблицы кодирования, позволяющей оптимальным образом расположить результирующий код. Таблица кодирования в этом случае должна стать неотъемлемой частью закодированных данных (методы Хаффмена, Шэннона–Фэно, арифметическое кодирование).

○ *Словарное кодирование* основано на нахождении повторяющихся последовательностей символов в сообщении (слов) и замене каждого повторяющегося слова ссылкой на его первое вхождение. В этом случае вместе с закодированным сообщением необходимо хранить (передавать) словарь. К алгоритмам словарного кодирования относятся LZ77, LZ78, LZSS, LZW.

Во многих современных программах-архиваторах применяется последовательное (двухшаговое) использование методов статистического и словарного кодирования, позволяющее достигнуть наилучшей степени сжатия данных.

□ *Помехозащищенное кодирование* служит для передачи данных по каналам связи и учитывает возможность возникновения помех и связанного с этим искажения или утраты части данных. При помехозащищенном кодировании избыточность не снижается, а, наоборот, увеличивается, тем самым обеспечивается возмож-

ность определения факта потери или искажения информации. Существует два основных типа помехозащищенного кодирования:

- *Кодирование с фиксацией ошибок* обладает сравнительно небольшой избыточностью, позволяющей зафиксировать на приемной стороне факт появления в сообщении ошибки. В случае обнаружения ошибки сообщение передается повторно.
- *Кодирование с коррекцией ошибок* еще более избыточно, что позволяет не только обнаружить возникшую при передаче по каналу связи ошибку, но и исправить ее. Недостатком этого вида кодирования является значительное (примерно в 3 раза) увеличение объема передаваемых данных. По этой причине оно используется гораздо реже, чем кодирование с фиксацией ошибок.

1.6. Основы классификации и структурирования информации

Важным понятием при работе с информацией является классификация объектов. Под объектом понимается любой предмет, процесс, явление материального или нематериального свойства. Система классификации позволяет сгруппировать объекты и выделить определенные классы, которые будут характеризоваться рядом общих свойств. Классификация объектов — это процедура группировки на качественном уровне, направленная на выделение однородных свойств. Применительно к информации как к объекту классификации выделенные классы называют информационными объектами.

Классификация — это система распределения объектов (предметов, явлений, процессов, понятий) по классам в соответствии с определенным признаком.

Пример. Всю информацию об университете можно классифицировать по многочисленным информационным объектам, которые будут характеризоваться общими свойствами:

- *информация о студентах* — в виде информационного объекта «Студент»;
- *информация о преподавателях* — в виде информационного объекта «Преподаватель»;
- *информация о факультетах* — в виде информационного объекта «Факультет» и т. п.

Свойства информационного объекта определяются информационными параметрами, называемыми *реквизитами*. Реквизиты представляются либо числовыми данными (например, вес, стоимость, год), либо признаками (например, цвет, марка машины, фамилия).

Реквизит — это логически неделимый информационный элемент, описывающий определенное свойство объекта, процесса, явления.

Пример. Информация о каждом студенте в отделе кадров университета систематизирована и представлена посредством одинаковых реквизитов:

- фамилия, имя, отчество;
- пол;
- год рождения;
- место рождения;
- адрес проживания;
- факультет, где проходит обучение студент, и т. д.

Все перечисленные реквизиты характеризуют свойства информационного объекта «Студент».

Кроме выявления общих свойств информационного объекта, классификация нужна для разработки правил (алгоритмов) и процедур обработки информации, представленной совокупностью реквизитов.

Пример. Алгоритм обработки информационных объектов библиотечного фонда позволяет получить информацию обо всех книгах по определенной тематике, об авторах, абонентах и т. д.

Алгоритм обработки информационных объектов фирмы позволяет получить информацию об объемах продаж, прибыли, заказчиках, видах производимой продукции и т. д.

Алгоритмы обработки в том и другом случае преследуют разные цели, обрабатывают разную информацию, реализуются разными способами.

В любой стране разработаны и применяются государственные, отраслевые и региональные классификаторы. Например, классифицированы отрасли промышленности, оборудование, профессии, единицы измерения, статьи затрат и т. д.

Классификатор — это систематизированный свод наименований и кодов классификационных группировок.

При классификации широко используются понятия «*классификационный признак*» и «*значение классификационного признака*», которые позволяют установить степень сходства или различия объектов. Возможен подход к классификации с объединением этих двух понятий в одно, названное *признаком классификации*. Синонимом признака классификации является основание деления.

Пример. В качестве признака классификации выбирается возраст, который состоит из трех значений: до 20 лет, от 20 до 30 лет, свыше 30 лет.

Можно в качестве признаков классификации использовать возраст до 20 лет, возраст от 20 до 30 лет, возраст свыше 30 лет.

Логическим продолжением задачи классификации является задача *кластеризации*, которую можно воспринимать как автоматическую классификацию. Главное отличие кластеризации в том, что перечень классов (групп) и их характеристики четко не заданы и определяются в процессе работы алгоритма.

Разработаны три метода классификации объектов: *иерархический*, *фасетный*, *дескрипторный*. Эти методы различаются разной стратегией применения классификационных признаков.

1.6.1. Иерархическая система классификации

Иерархическая система классификации (рис. 1.4) строится следующим образом:

- исходное множество элементов составляет 0-й уровень и делится в зависимости от выбранного классификационного признака на классы (группировки), которые образуют 1-й уровень;
- каждый класс 1-го уровня в соответствии с характерным классификационным признаком делится на подклассы, которые образуют 2-й уровень;
- каждый класс 2-го уровня аналогично делится на группы, которые образуют 3-й уровень, и т. д.

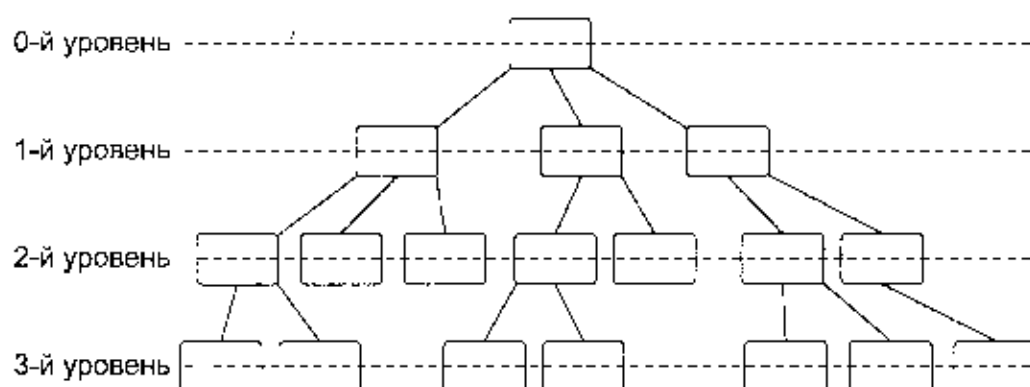


Рис. 1.4. Иерархическая система классификации

С учетом достаточно жесткой процедуры построения структуры классификации необходимо перед началом работы определить ее цель, то есть выяснить, какими свойствами должны обладать объединяемые в классы объекты. Эти свойства принимаются в дальнейшем за признаки классификации.

ВНИМАНИЕ

В иерархической системе классификации из-за жесткой структуры особое внимание следует уделить выбору классификационных признаков.

В иерархической системе классификации каждый объект на любом уровне должен быть отнесен к одному классу, который характеризуется конкретным значением выбранного классификационного признака. Для последующей группировки в каждом новом классе необходимо задать свои классификационные признаки и их значения. Таким образом, выбор классификационных признаков будет зависеть от

семантического содержания того класса, для которого необходима группировка на последующем уровне иерархии.

Количество уровней классификации, соответствующее числу признаков, выбранных в качестве основания деления, характеризует глубину классификации.

Достоинства иерархической системы классификации:

- простота построения;
- использование независимых классификационных признаков в различных ветвях иерархической структуры.

Недостатки иерархической системы классификации:

- жесткая структура, которая приводит к сложности внесения изменений, так как приходится перераспределять все классификационные группировки;
- невозможность группировать объекты по заранее непредусмотренным сочетаниям признаков.

Пример. Поставлена задача — создать иерархическую систему классификации для информационного объекта «Факультет», которая позволила бы классифицировать информацию обо всех студентах по следующим классификационным признакам: факультет, на котором учится студент, возрастной состав студентов, пол студента, для женщин — наличие детей.

Система классификации, представленная на рис. 1.5, будет иметь следующие уровни:

- 0-й уровень. Информационный объект «Факультет».
- 1-й уровень. Выбирается классификационный признак — название факультета, что позволяет выделить несколько классов с разными названиями факультетов, в которых хранится информация обо всех студентах.
- 2-й уровень. Выбирается классификационный признак — возраст, который имеет три градации: до 20 лет, от 20 до 30 лет, свыше 30 лет. По каждому факультету выделяются три возрастных подкласса студентов.
- 3-й уровень. Выбирается классификационный признак — пол. Каждый подкласс 2-го уровня разбивается на две группы. Таким образом, информация о студентах каждого факультета в каждом возрастном подклассе разделяется на две группы: мужчины и женщины.
- 4-й уровень. Выбирается классификационный признак — наличие детей у женщин: есть, нет.

Созданная иерархическая система классификации имеет глубину классификации, равную четырем.



Рис. 1.5. Пример иерархической системы классификации для информационного объекта «Факультет»

1.6.2. Фасетная система классификации

Фасетная система классификации в отличие от иерархической позволяет выбирать признаки классификации независимо как друг от друга, так и от семантического содержания классифицируемого объекта. Признаки классификации называются *фасетами* (facet — рамка). Каждый фасет (Φ_i) содержит совокупность однородных значений данного классификационного признака. Причем значения в фасете могут располагаться в произвольном порядке, хотя предпочтительнее их упорядочение.

Пример. Фасет *цвет* содержит значения: красный, белый, зеленый, черный, желтый. Фасет *специальность* содержит названия специальностей. Фасет *образование* содержит значения: среднее, среднее специальное, высшее.

Схема построения фасетной системы классификации в виде таблицы представлена на рис. 1.6. Названия столбцов соответствуют выделенным классификационным признакам (фасетам), обозначенным $\Phi_1, \Phi_2, \dots, \Phi_i, \dots, \Phi_n$ (например: цвет, размер одежды, вес и т. д.). Произведена нумерация строк таблицы. В каждой клетке таблицы хранится конкретное значение фасета. Например, фасет *цвет*, обозначенный как Φ_2 , содержит значения красный, белый, зеленый, черный и желтый.

Процедура классификации состоит в присвоении каждому объекту соответствующих значений из фасетов. При этом могут быть использованы не все фасеты. Для каждого объекта задается конкретная группировка фасетов структурной формулой, в которой отражается их порядок следования:

$$K_s = (\Phi_1, \Phi_2, \dots, \Phi_i, \dots, \Phi_n)$$

Здесь Φ_i — i -й фасет; n — количество фасетов.

		Фасеты						
		Φ_1	Φ_2	Φ_3	...	Φ_i	...	Φ_n
Значения фасетов	1	●	●	●		●		●
	2	●	●	●		●		●
	...	●		●		●		●
	k	●				●		

Рис. 1.6. Фасетная система классификации

При построении фасетной системы классификации необходимо, чтобы значения, используемые в различных фасетах, не повторялись. Фасетную систему легко можно модифицировать, внося изменения в конкретные значения любого фасета.

Достоинства фасетной системы классификации:

- возможность достижения большой емкости классификации, то есть использования большого количества признаков классификации и их значений для создания группировок;

- возможность простой модификации всей системы классификации без изменения структуры существующих группировок.

Недостатком фасетной системы классификации является сложность ее построения, так как необходимо учитывать все многообразие классификационных признаков.

Пример. Вернемся к примеру из подраздела «Иерархическая система классификации» и для сопоставления разработаем фасетную систему классификации. Сгруппируем и представим в виде таблицы (табл. 1.1) все классификационные признаки по фасетам:

- *название факультета* — фасет с пятью названиями факультетов;
- *возраст* — фасет с тремя возрастными группами;
- *пол* — фасет с двумя градациями;
- *дети* — фасет с двумя градациями.

Структурную формулу любого класса можно представить в виде:

$$K_i = (\text{Факультет}, \text{Возраст}, \text{Пол}, \text{Дети}).$$

Присвоив конкретные значения каждому фасету, получим следующие классы:

- $K_1 = (\text{Радиотехнический факультет}, \text{возраст до 20 лет}, \text{мужчина}, \text{есть дети})$;
- $K_2 = (\text{Коммерческий факультет}, \text{возраст от 20 до 30 лет}, \text{мужчина}, \text{детей нет})$;
- $K_3 = (\text{Математический факультет}, \text{возраст до 20 лет}, \text{женщина}, \text{детей нет})$ и т. д.

Таблица 1.1. Пример фасетной системы классификации для информационного объекта «Факультет»

Название факультета	Возраст	Пол	Дети
Радиотехнический	До 20 лет	М	Есть
Машиностроительный	20–30 лет	Ж	Нет
Коммерческий	Свыше 30 лет		
Информационные системы			
Математический			

1.6.3. Дескрипторная система классификации

Для организации поиска информации, для ведения тезаурусов (словарей) эффективно используется дескрипторная (описательная) система классификации, язык которой приближается к естественному языку описания информационных объектов. Особенно широко она применяется в библиотечной системе поиска.

Суть дескрипторного метода классификации заключается в следующем:

1. Отбирается совокупность *ключевых слов* или словосочетаний, описывающих определенную предметную область или совокупность однородных объектов. Причем среди ключевых слов могут быть синонимы.

2. Выбранные ключевые слова и словосочетания подвергаются *нормализации*, то есть из совокупности синонимов выбирается один или несколько наиболее употребимых.
3. Создается словарь *дескрипторов*, то есть словарь ключевых слов и словосочетаний, отобранных в результате процедуры нормализации.

Пример. В качестве объекта классификации рассмотрим успеваемость студентов. Ключевыми словами могут быть выбраны: оценка, экзамен, зачет, преподаватель, студент, семестр, название предмета. Здесь нет синонимов, поэтому указанные ключевые слова можно использовать как словарь дескрипторов.

В качестве предметной области выбираем учебную деятельность в высшем учебном заведении. Ключевыми словами могут быть выбраны: студент, обучаемый, учащийся, преподаватель, учитель, педагог, лектор, ассистент, доцент, профессор, коллега, факультет, подразделение университета, аудитория, комната, лекция, практическое занятие, занятие и т. д. Среди указанных ключевых слов встречаются синонимы, например: студент, обучаемый, учащийся; преподаватель, учитель, педагог; факультет, подразделение университета и т. д. После нормализации словарь дескрипторов будет состоять из следующих слов: студент, преподаватель, лектор, ассистент, доцент, профессор, факультет, аудитория, лекция, практическое занятие и т. д.

Между дескрипторами устанавливаются связи, которые позволяют расширить область поиска информации. Связи могут быть трех видов:

- *синонимические связи* указывают некоторую совокупность ключевых слов как синонимов;
- *родо-видовые связи* отражают включение некоторого класса объектов в более представительный класс;
- *ассоциативные связи* соединяют дескрипторы, обладающие общими свойствами.

Пример. Синонимическая связь: студент – учащийся – обучаемый.

Родо-видовая связь: университет – факультет – кафедра.

Ассоциативная связь: студент – экзамен – профессор – аудитория.

1.6.4. Классификация информации по разным признакам

Любая классификация всегда относительна. Один и тот же объект может быть классифицирован по разным признакам или критериям. Часто встречаются ситуации, когда в зависимости от условий внешней среды объект может быть отнесен к разным классификационным группировкам. Эти рассуждения особенно актуальны при классификации видов информации без учета ее предметной ориентации, так как она часто может быть использована в разных условиях, разными потребителями, для разных целей.

В табл. 1.2 приведена одна из схем классификации циркулирующей в организации (фирме) информации. В основу классификации положено пять наиболее

общих признаков: место возникновения, стадия обработки, способ отображения, стабильность, функция управления.

Таблица 1.2. Классификация информации, циркулирующей в организации

Информация				
По месту возникновения	По стабильности	По стадии обработки	По способу отображения	По функции управления
Входная	Переменная	Первичная	Текстовая	Планивая
Выходная	Постоянная	Вторичная	Графическая	Нормативно-справочная
Внутренняя		Промежуточная		Учетная
Внешняя		Результатная		Оперативная

Место возникновения

По признаку места возникновения информацию можно разделить на входную, выходную, внутреннюю, внешнюю.

Входная информация — это информация, поступающая в фирму или ее подразделения. *Выходная информация* — это информация, поступающая из фирмы в другую фирму, организацию (подразделение).

Одна и та же информация может являться входной для одной фирмы, а для другой, ее вырабатывающей, выходной. По отношению к объекту управления (фирма или ее подразделение: цех, отдел, лаборатория) может быть определена информация как внутренняя, так и внешняя.

Внутренняя информация возникает внутри объекта, *внешняя* — за пределами объекта.

Пример. Содержание указа правительства об изменении уровня взимаемых налогов для фирмы является, с одной стороны, внешней информацией, с другой — входной. Сведения фирмы, подаваемые в налоговую инспекцию о размере отчислений в госбюджет, являются, с одной стороны, выходной информацией, с другой — внешней по отношению к налоговой инспекции.

Стадия обработки

По стадии обработки информация может быть первичной, вторичной, промежуточной, результатной.

Первичная информация — это информация, которая возникает непосредственно в процессе деятельности объекта и регистрируется на начальной стадии. *Вторичная информация* — это информация, которая получается в результате обработки первичной информации; она может быть промежуточной и результатной. *Промежуточная информация* используется в качестве исходных данных для последующих расчетов. *Результатная информация* получается в процессе обработки первичной и промежуточной информации и служит для выработки управленческих решений.

Пример. В художественном цехе, где производится роспись чашек, в конце каждой смены регистрируется общее количество произведенной продукции и количество каждым работником расписанных чашек. Это первичная информация. В конце каждого месяца мастер подводит итоги первичной информации. Это, с одной стороны, вторичная промежуточная информация, а с другой — результатная. Итоговые данные поступают в бухгалтерию, где производится расчет заработной платы каждого работника в зависимости от его выработки. Полученные расчетные данные — результатная информация.

Способ отображения

По способу отображения информация подразделяется на текстовую и графическую.

Текстовая информация — это совокупность алфавитных, цифровых и специальных символов, с помощью которых информация представляется на физическом носителе (бумага, изображение на экране дисплея). *Графическая информация* — это различного рода графики, диаграммы, схемы, рисунки и т. д.

Стабильность

По стабильности информация может быть переменной (текущей) и постоянной (условно постоянной).

Переменная информация отражает фактические количественные и качественные характеристики производственно-хозяйственной деятельности фирмы. Она может меняться для каждого случая как по назначению, так и по количеству. Например, количество произведенной продукции за смену, еженедельные затраты на доставку сырья, количество исправных станков и т. п. *Постоянная (условно-постоянная) информация* — это неизменная и многократно используемая в течение длительного периода времени информация. Постоянная информация может быть справочной, нормативной, плановой:

- *постоянная справочная информация* включает в себя описание постоянных свойств объекта в виде устойчивых длительное время признаков (например: табельный номер служащего, профессия работника, номер цеха и т. п.);
- *постоянная нормативная информация* содержит местные, отраслевые и общегосударственные нормативы (например: размер налога на прибыль, стандарт на качество продуктов определенного вида, размер минимальной оплаты труда, тарифная сетка оплаты государственным служащим);
- *постоянная плановая информация* содержит многократно используемые в фирме плановые показатели (например: план выпуска телевизоров, план подготовки специалистов определенной квалификации).

Функция управления

По функциям управления обычно классифицируют экономическую информацию, при этом выделяют следующие группы: плановая, нормативно-справочная, учетная и оперативная (текущая).

Плановая информация – информация о параметрах объекта управления на будущий период. На эту информацию идет ориентация всей деятельности фирмы.

Пример. К плановой информации фирмы могут быть отнесены такие показатели, как план выпуска продукции, планируемая прибыль от реализации, ожидаемый спрос на продукцию и т. д.

Нормативно-справочная информация представляет собой различные нормативные и справочные данные. Ее обновление происходит достаточно редко.

Пример. Нормативно-справочной информацией на предприятии являются:

- время, предназначенное для изготовления типовой детали (нормы трудоемкости);
- среднедневная оплата рабочего по разряду;
- оклад служащего;
- адрес поставщика или покупателя и т. д.

Учетная информация – это информация, которая характеризует деятельность фирмы за определенный прошлый период времени. На основании этой информации могут быть проведены следующие действия: скорректирована плановая информация, сделан анализ хозяйственной деятельности фирмы, приняты решения по более эффективному управлению работами и пр. На практике в качестве учетной информации могут выступать информация бухгалтерского учета, статистическая информация и информация оперативного учета.

Пример. Учетной информацией являются: количество проданной продукции за определенный период времени; среднесуточная загрузка или простой станков и т. п.

Оперативная (текущая) информация – это информация, используемая в оперативном управлении и характеризующая производственные процессы в текущий (данный) период времени. К оперативной информации предъявляются серьезные требования по скорости поступления и обработки, а также по степени ее достоверности. От того, насколько быстро и качественно проводится ее обработка, во многом зависит успех фирмы на рынке.

Пример. К оперативной информации относят:

- количество изготовленных деталей за час, смену, день;
- количество проданной продукции за день или определенный час;
- объем сырья от поставщика на начало рабочего дня и т. д.

1.7. Кодирование при классификации информации

1.7.1. Общие понятия

Система кодирования применяется для замены названия объекта условным обозначением (кодом) в целях обеспечения удобной и более эффективной обработки информации.

Система кодирования — это совокупность правил кодового обозначения объектов.

Код строится на базе алфавита, состоящего из букв, цифр и других символов. Код характеризуется длиной и структурой:

- ❑ *длина* -- число позиций в коде;
- ❑ *структура* — порядок расположения в коде символов, используемых для обозначения классификационного признака.

Процедура присвоения объекту кодового обозначения называется кодированием. Можно выделить две группы методов, используемых в системе кодирования (рис. 1.7):

- ❑ *классификационная система кодирования* ориентирована на предварительную классификацию объектов на основе либо иерархической, либо фасетной системы;
- ❑ *регистрационная система кодирования* предварительной классификации объектов не требует.



Рис. 1.7. Система кодирования, использующая разные методы

1.7.2. Классификационное кодирование

Классификационное кодирование применяется после проведения классификации объектов. Различают последовательное и параллельное кодирование.

Последовательное кодирование используется для иерархической классификационной структуры. Суть метода заключается в следующем: сначала записывается код старшей группировки 1-го уровня, затем код группировки 2-го уровня, затем код группировки 3-го уровня и т. д. В результате получается кодовая комбинация, каждый разряд которой содержит информацию о специфике выделенной группы

на каждом уровне иерархической структуры. Последовательная система кодирования обладает теми же достоинствами и недостатками, что и иерархическая система классификации.

Пример. Проведем кодирование информации, классифицированной с помощью иерархической схемы (см. рис. 1.5). Количество кодовых группировок определяется глубиной классификации и в данном случае равно 4. Прежде чем начать кодирование, необходимо определиться с алфавитом, то есть выбрать используемые символы. Для большей наглядности выберем десятичную систему счисления — 10 арабских цифр. Анализ схемы на рис. 1.5 показывает, что длина кода определяется четырьмя десятичными разрядами, а кодирование группировки на каждом уровне можно делать путем последовательной нумерации слева направо. В общем виде код можно записать как XXXX, где X — значение десятичного разряда. Рассмотрим структуру кода, начиная со старшего разряда:

- 1-й (старший) разряд выделен для классификационного признака «название факультета» и имеет следующие значения: 1 — коммерческий; 2 — информационные системы; 3 — название следующего факультета и т. д.
- 2-й разряд выделен для классификационного признака «возраст» и имеет следующие значения: 1 — до 20 лет; 2 — от 20 до 30 лет; 3 — свыше 30 лет;
- 3-й разряд выделен для классификационного признака «пол» и имеет следующие значения: 1 — мужчины; 2 — женщины;
- 4-й разряд выделен для классификационного признака «наличие детей у женщин» и имеет следующие значения: 1 — есть дети; 2 — нет детей; 0 выделяется для мужчин, так как для них подобной информации не требуется.

Принятая система кодирования позволяет легко расшифровать любой код группировки, например:

- 1310 — студенты коммерческого факультета, свыше 30 лет, мужчины;
- 2221 — студенты факультета информационных систем, от 20 до 30 лет, женщины, имеющие детей.

Параллельное кодирование используется для фасетной системы классификации. Суть метода заключается в следующем: все фасеты кодируются независимо друг от друга; для значений каждого фасета выделяется определенное количество разрядов кода. Параллельная система кодирования обладает теми же достоинствами и недостатками, что и фасетная система классификации.

Пример. Проведем кодирование информации, классифицированной с помощью фасетной схемы (см. рис. 1.6). Количество кодовых группировок определяется количеством фасетов и равно 4. Выберем десятичную систему счисления в качестве алфавита кодировки, что позволит для значений фасетов выделить один разряд и иметь длину кода, равную 4. В отличие от последовательного кодирования для иерархической системы классификации в данном методе не имеет значения порядок кодировки фасетов. В общем виде код можно записать как XXXX, где X — значение десятичного разряда. Рассмотрим структуру кода, начиная со старшего разряда:

- 1-й (старший) разряд выделен для фасета «пол» и имеет следующие значения: 1 — мужчины; 2 — женщины;

- 2-й разряд выделен для фасета «наличие детей у женщин» и имеет следующие значения: 1 — есть дети, 2 — нет детей; 0 выделяется для мужчин, так как для них подобной информации не требуется;
- 3-й разряд выделен для фасета «возраст» и имеет следующие значения: 1 — до 20 лет; 2 — от 20 до 30 лет; 3 — свыше 30 лет;
- 4-й разряд выделен для фасета «название факультета» и имеет следующие значения: 1 — радиотехнический, 2 — машиностроительный, 3 — коммерческий, 4 — информационные системы, 5 — математический и т. д.

Принятая система кодирования позволяет легко расшифровать любой код группировки, например:

- 2135 — женщины в возрасте свыше 30 лет, имеющие детей и являющиеся студентами математического факультета;
- 1021 — мужчины возраста от 20 до 30 лет, являющиеся студентами радиотехнического факультета.

1.7.3. Регистрационное кодирование

Регистрационное кодирование используется для однозначной идентификации объектов и не требует предварительной классификации объектов. Различают порядковую и серийно-порядковую систему.

Порядковая система кодирования предполагает последовательную нумерацию объектов числами натурального ряда. Этот порядок может быть случайным или определяться после предварительного упорядочения объектов, например, по алфавиту. Этот метод применяется в том случае, когда количество объектов невелико, например, при кодировании названий факультетов университета или студентов в учебной группе.

Серийно-порядковая система кодирования предусматривает предварительное выделение групп объектов, которые составляют серию, затем в каждой серии производится порядковая нумерация объектов. Каждая серия, в свою очередь, тоже получает порядковую нумерацию. По своей сути серийно-порядковая система является смешанной: классифицирующей и идентифицирующей. Применяется она тогда, когда количество групп невелико.

Пример. Все студенты одного факультета разбиваются на учебные группы (в данной терминологии — серии), для которых используется порядковая нумерация. Внутри каждой группы производится упорядочение фамилий студентов по алфавиту и каждому студенту присваивается номер.

Вопросы для самопроверки

1. Дайте определение термину «информация». Какие подходы в определении этого термина вам известны?
2. Чем информация отличается от данных? Как связаны между собой эти два понятия?

3. Что такое «адекватность информации»? Какие виды адекватности вы знаете? Кратко опишите их.
4. Каким образом измеряют информацию? Какие меры информации вам известны?
5. Как связаны между собой количество информации и мера неопределенности состояния системы?
6. Чем определяется семантическая мера информации? Что такое «тезаурус»? Каким отношением связаны между собой объем тезауруса и объем воспринимаемой информации?
7. Что такое «прагматическая мера информации»?
8. Какими понятиями определяется качество информации?
9. Дайте определение информационным процессам.
10. Что такое «информационная система»?
11. Нарисуйте схему передачи информации и поясните назначение ее элементов.
12. Что такое «кодирование» и «декодирование»?
13. Когда применяется кодирование по образцу?
14. Какие типы шифрования вам известны?
15. В каких случаях применяют помехозащищенное кодирование?
16. Что такое «классификация»? Что такое «реквизит», «классификатор»?
17. Какие методы классификации вам известны?
18. В чем состоит фасетный принцип классификации?
19. В каких случаях применяется дескрипторный метод классификации?
20. По каким признакам можно классифицировать информацию?
21. Какие системы кодирования информации применяются при классификации?

Литература

1. Вернер М. Основы кодирования. М.: Техносфера, 2004.
2. Зверев Г.Н. Теоретическая информатика и ее основания (в 2 томах). М.: Физматлит, 2007.
3. Кудряшов Б.Д. Теория информации: Учебник для ВУЗов. СПб.: Питер, 2009.

Глава 2 Роль информации в развитии общества

- 2.1. Информатизация общества
- 2.2. Информационный потенциал общества
- 2.3. Информатика — предмет и задачи

Информатизация обеспечит переход общества от индустриального этапа развития к информационному. Информационный рынок предоставит потребителям все необходимые информационные продукты и услуги, а их производство обеспечит индустрия информатики, часто называемая информационной индустрией. Особую роль информатизация общества должна сыграть во взаимоотношениях личности и государства, бизнеса и государства, государства и науки. В данной главе рассказывается о роли информатики как науки, об информатизации общества и развитии его информационного потенциала.

2.1. Информатизация общества

2.1.1. Представление об информационном обществе

Роль и значение информационных революций

В истории развития цивилизации произошло несколько информационных революций — преобразований общественных отношений из-за кардинальных изменений в сфере обработки информации. Следствием подобных преобразований являлось приобретение человеческим обществом нового качества.

Первая информационная революция связана с появлением языка и членораздельной человеческой речи. Ведь именно развитие языка оказало колоссальное влияние

на развитие сознания людей, а его использование в их практической деятельности стало информационной основой появления первых технологий, то есть знания и навыков рациональной организации этой деятельности.

В первобытном обществе использовались и распространялись только «живые знания», носителями которых являлись живые люди — старейшины, жрецы, шаманы. В этих условиях процессы накопления и распространения знания в обществе осуществлялись чрезвычайно медленно, а сохранение уже накопленных знаний было недостаточно надежным. Со смертью их носителей многие знания утрачивались и должны были формироваться заново. На это уходили столетия.

Ситуация коренным образом изменилась, когда люди научились *отчуждать знания* и фиксировать их на материальных носителях в виде рисунков, чертежей, условных знаков, многие из которых сохранились до настоящего времени. Это и привело ко второй информационной революции.

Вторая информационная революция связана с изобретением письменности. Это изобретение позволило не только обеспечить *сохранность* уже накопленных человеческим обществом знаний, но и повысить *достоверность* этих знаний, создать условия для их существенно более широкого, чем ранее, распространения. Это был крупнейший шаг в развитии цивилизации, последствия которого мы ощущаем до настоящего времени. Ведь именно с изобретением письменности стало возможным развитие науки и культуры в современном понимании этих терминов.

Существенным образом изменилась и *информационная среда общества*, стали возможными новые виды информационных коммуникаций между людьми посредством обмена письменными сообщениями. Появились исторические летописи, поэзия и литература, зародились элементы того нового и своеобразного явления, которое мы сегодня называем *информационной культурой*. Новый смысл приобрело и понятие образования.

Третья информационная революция началась в эпоху Возрождения и связана с изобретением *книгопечатания*, которое следует признать одной из первых эффективных *информационных технологий*. Широкое внедрение этого изобретения в социальную практику привело к первому информационному взрыву. Произошел своеобразный рост количества используемых в обществе информационных документов, а самое главное — началось более широкое распространение информации, научных знаний и информационной культуры. Появились первые библиотеки печатных книг, сначала частного характера, а затем и публичные. Печатная книга стала главным хранителем и источником знаний.

Своего апогея третья информационная революция достигла, по-видимому, с появлением *печатных средств массовой информации*: газет, журналов, рекламных объявлений, информационных справочников и т. п.

Четвертая информационная революция началась в XIX в. и продолжалась 1-ю половину XX в. Тогда были изобретены и стали все более широко распространяться такие новые средства информационной коммуникации, как *радио, телефон и телевидение*. Эти средства имеются сегодня в развитых странах практически в каждой семье, оказывая значительное воздействие на формирование общественного сознания. Благодаря этим средствам люди уже не испытывают чувства одиночества и изолированности от окружающего их общества. Ведь они сегодня

подключены к общему информационному пространству не только своей страны, но и значительной части нашей планеты.

Мы уже не мыслим своего существования в отрыве от *мирового информационного пространства*. А это очень важный аспект формирования сознания человека, который сегодня ощущает себя гражданином не только своей страны, но и, в определенной степени, всего мира. Таким образом, *информационная революция изменяет общественное сознание всего человечества*, делает его все более глобальным.

Пятая информационная революция началась в 50-е гг. XX в., то есть с того времени, когда в социальной практике стали использоваться средства *цифровой вычислительной техники*. Применение этих средств для обработки научной, экономической и социальной информации кардинальным образом изменило возможности человека по активизации и эффективному использованию *информационных ресурсов*.

Этот период характеризуют три фундаментальные инновации:

- переход от механических и электрических средств преобразования информации к электронным;
- миниатюризация всех узлов, устройств, приборов, машин;
- создание программно-управляемых устройств и процессов.

Для получения более целостного представления об этом периоде целесообразно познакомиться с приведенной далее справкой о смене поколений электронно-вычислительных машин (ЭВМ) и сопоставить эти сведения с этапами в области обработки и передачи информации.

СПРАВКА О СМЕНЕ ПОКОЛЕНИЙ ЭВМ

1-е поколение (начало 50-х гг.). Элементная база — электронные лампы. ЭВМ отличались большими габаритами, значительным потреблением энергии, малым быстродействием, низкой надежностью, программированием в кодах.

2-е поколение (с конца 50-х гг.). Элементная база — полупроводниковые элементы. Улучшились по сравнению с ЭВМ предыдущего поколения все технические характеристики. Для программирования использовались алгоритмические языки.

3-е поколение (начало 60-х гг.). Элементная база — интегральные схемы, многослойный печатный монтаж. Характеризуется резким уменьшением габаритов ЭВМ, повышением их надежности, увеличением производительности. Становится возможным доступ с удаленных терминалов.

4-е поколение (с середины 70-х гг.). Элементная база — микропроцессоры, большие интегральные схемы. Улучшились технические характеристики. Начался массовый выпуск персональных компьютеров. Направления развития: мощные микропроцессорные вычислительные системы с высокой производительностью, создание дешевых микроЭВМ.

5-е поколение (с середины 80-х гг.). Началась разработка интеллектуальных компьютеров, пока не увенчавшаяся успехом. Характеризуется внедрением во все сферы компьютерных сетей и их объединения, использованием распределенной обработки данных, повсеместным применением компьютерных информационных технологий.

Последняя информационная революция выдвигает на первый план новую отрасль — информационную индустрию, связанную с появлением технических средств, методов, технологий для производства новых знаний. Важнейшими составляющими информационной индустрии становятся все виды информационных технологий, особенно телекоммуникации. Современная информационная технология опирается на достижения в области компьютерной техники и средств связи.

Информационная технология (ИТ) — это процесс, использующий совокупность средств и методов сбора, обработки и передачи данных (первичной информации) для получения информации нового качества о состоянии объекта, процесса или явления.

Телекоммуникации — это дистанционная передача данных на базе компьютерных сетей и современных технических средств связи.

Усложнение индустриального производства, социальной, экономической и политической жизни, изменение динамики процессов во всех сферах деятельности человека привели, с одной стороны, к росту потребностей в знаниях, а с другой — к созданию новых средств и способов удовлетворения этих потребностей.

Бурное развитие компьютерной техники и информационных технологий послужило толчком к развитию общества, построенного на использовании различной информации и получившего название информационного общества.

Об особенностях информационного общества

В информационном обществе процесс компьютеризации даст людям доступ к надежным источникам информации, избавит их от рутинной работы, обеспечит высокий уровень автоматизации обработки информации в производственной и социальной сферах. Развитие общества в первую очередь будет определяться производством информационного, а не материального продукта. Материальный же продукт станет более информационно емким, что означает увеличение доли инноваций, дизайна и маркетинга в его стоимости.

В информационном обществе изменятся не только производство, но и весь уклад жизни, система ценностей, возрастет значимость культурного досуга по отношению к материальным ценностям. По сравнению с индустриальным обществом, где все направлено на производство и потребление товаров, в информационном обществе увеличивается доля умственного труда, приоритетными становятся интеллект и знания. От человека потребуются способность к творчеству, возрастет спрос на знания.

Материальной и технологической базой информационного общества станут различного рода системы на основе компьютерной техники и компьютерных сетей, информационной технологии, телекоммуникационной связи.

Информационное общество — это общество, в котором большинство работающих занято производством, хранением, переработкой и реализацией информации и знаний.

В реальной практике развития науки и техники передовых стран в начале XXI в. постепенно приобретает зримые очертания созданная теоретиками картина

информационного общества. Прогнозируется превращение всего мирового пространства в единое компьютеризированное и информационное сообщество людей, проживающих в электронных квартирах и коттеджах. Любое жилище оснастится всевозможными электронными приборами и компьютеризированными устройствами. Деятельность людей будет сосредоточена главным образом на обработке информации, а материальное производство и производство энергии останется для машин.

Уже опубликован ряд фактических материалов, свидетельствующих, что это не утопия, а неизбежная реальность недалекого будущего.

Пример. Исследования, проведенные в США, показали, что количество работающих, вовлеченных в удаленную работу (от ежедневной до эпизодической), за последние десять лет увеличилось примерно в 3 раза. При этом среди тех, кто работает удаленно эпизодически, выше всех оказалась доля работников с доходами выше среднего. Это произошло из-за того, что все больше фирм использует так называемые мобильные офисы, позволяющие не прерывать работу при выходе из здания фирмы.

При переходе к информационному обществу возникает новая индустрия переработки информации на базе компьютерных и телекоммуникационных информационных технологий.

Ряд ученых выделяют характерные черты информационного общества:

- будет решена проблема информационного кризиса, то есть разрешено противоречие между информационной лавиной и информационным голодом;
- будет обеспечен приоритет информации по сравнению с другими ресурсами;
- главной формой развития станет информационная экономика;
- в основу общества будут заложены автоматизированные генерация, хранение, обработка и использование знаний с помощью новейшей информационной техники и технологий;
- информационная технология приобретет глобальный характер, охватывая все сферы социальной деятельности человека;
- сформируется информационное единство всей человеческой цивилизации;
- с помощью средств информатики будет реализован свободный доступ каждого человека к информационным ресурсам всей цивилизации;
- будут реализованы гуманистические принципы управления обществом и воздействия на окружающую среду.

Кроме положительных моментов прогнозируются и опасные тенденции:

- все большее влияние на общество средств массовой информации;
- возможное разрушительное вторжение информационных технологии в частную жизнь людей и организаций;
- проблема отбора качественной и достоверной информации;
- возможные трудности многих людей в отношении адаптации к среде информационного общества, опасность разрыва между «информационной элитой»

(людьми, занимающимися разработкой информационных технологий) и потребителями.

Ближе всех на пути к информационному обществу стоят страны с развитой информационной индустрией, к числу которых следует отнести США, Японию, Англию, страны Западной Европы. В этих странах уже давно одно из направлений государственной политики связано с инвестициями и поддержкой инноваций в информационную индустрию, в развитие компьютерных систем и телекоммуникаций. В последние годы в нашей стране также предпринимаются определенные усилия по созданию современного информационного общества, все больше внимания этим вопросам уделяет российское правительство, парламент и президент.

2.1.2. Роль информатизации в развитии общества

Процесс информатизации общества является закономерным глобальным процессом развития цивилизации, который обусловлен целым рядом объективных факторов. Важнейшими из них являются:

- быстро возрастающая сложность искусственно создаваемой человеком среды своего обитания — техносферы, которая все больше снижает надежность и устойчивость этой среды;
- истощение природных ресурсов планеты и обусловленная этим необходимость отказа от господствующей в настоящее время парадигмы экстенсивного развития цивилизации;
- возрастание экологической опасности и необходимость поиска решения самой актуальной и сложной проблемы современности — проблемы выживания человечества как биологического вида.

Глобальная информатизация общества активно содействует развитию новых геополитических процессов, наиболее важными из которых являются:

- *глобализация экономики*, проявляющаяся в создании транснациональных корпораций, международного разделения труда и международных рынков сбыта продукции;
- *глобализация науки*, проявляющаяся в создании распределенных международных творческих коллективов ученых, которые работают над общими научными проектами, и в интенсификации процессов международного обмена научной информацией;
- *глобализация образования*, проявляющаяся в развитии систем дистанционного образования, создании открытых территориально-распределенных университетов, колледжей, центров повышения профессиональной квалификации;
- *глобализация культуры*, характерными признаками которой является создание электронных библиотек и электронных версий картинных галерей, художественных и музейных экспозиций, а также наиболее ценных объектов архитектуры и строительства.

Деятельность отдельных людей, групп, коллективов и организаций сейчас все в большей степени начинает зависеть от их информированности и способности

эффективно использовать имеющуюся информацию. Прежде чем предпринять какие-то действия, необходимо провести большую работу по сбору и переработке информации, ее осмыслению и анализу. Отыскание рациональных решений в любой сфере требует обработки больших объемов информации, что подчас невозможно без привлечения специальных технических средств.

Возрастание объема информации особенно стало заметно в середине XX в. Лавинообразный поток информации хлынул на человека, не давая ему возможности воспринять эту информацию в полной мере. В ежедневно появляющемся новом потоке информации ориентироваться становилось все труднее. Подчас выгоднее стало создавать новый материальный или интеллектуальный продукт, нежели вести розыск аналога, сделанного ранее. Возникновение больших потоков информации обуславливается:

- чрезвычайно быстрым ростом числа документов, отчетов, диссертаций, докладов и т. п., в которых излагаются результаты научных исследований и опытно-конструкторских работ;
- постоянно увеличивающимся числом периодических изданий по разным областям человеческой деятельности;
- появлением разнообразных данных (метеорологических, геофизических, медицинских, экономических и др.), записываемых обычно на магнитных лентах и поэтому не попадающих в сферу действия систем коммуникаций.

Как результат — наступает информационный кризис (взрыв), который имеет следующие проявления:

- появляются противоречия между ограниченными возможностями человека по восприятию и переработке информации и существующими мощными потоками и массивами хранящейся информации. Так, например, общая сумма знаний менялась вначале очень медленно, но уже с 1900 г. она удваивалась каждые 50 лет, к 1950 г. удвоение происходило каждые 10 лет, к 1970 г. — уже каждые 5 лет, с 1990 г. — ежегодно;
- существует большое количество избыточной информации, которая затрудняет восприятие полезной для потребителя информации;
- возникают определенные экономические, политические и другие социальные барьеры, которые препятствуют распространению информации. Например, по причине соблюдения секретности часто необходимой информацией не могут воспользоваться работники разных ведомств.

Эти причины породили весьма парадоксальную ситуацию — в мире накоплен громадный информационный потенциал, но люди не могут им воспользоваться в полном объеме в силу ограниченности своих возможностей. Информационный кризис поставил общество перед необходимостью поиска путей выхода из создавшегося положения. Внедрение ЭВМ, современных средств переработки и передачи информации в различные сферы деятельности послужило началом нового эволюционного процесса, называемого *информатизацией*, в развитии человеческого общества, находящегося на этапе индустриального развития.

Информатизация общества — это организованный социально-экономический и научно-технический процесс создания оптимальных условий для удовлетворения информационных потребностей и реализации прав граждан, органов государственной власти, органов местного самоуправления, организаций, общественных объединений на основе формирования и использования информационных ресурсов.

Рассмотрим этот процесс более подробно.

История развития информатизации началась в США с 60-х гг., затем с 70-х гг. — в Японии и с конца 70-х — в Западной Европе.

Современное материальное производство и другие сферы деятельности все больше нуждаются в информационном обслуживании, переработке огромного количества информации. Универсальным техническим средством обработки любой информации является компьютер, который играет роль усилителя интеллектуальных возможностей человека и общества в целом, а коммуникационные средства, использующие компьютеры, служат для связи и передачи информации. Появление и развитие компьютеров — это необходимая составляющая процесса информатизации общества.

Информатизация общества является одной из закономерностей современного социального прогресса. Этот термин все настойчивее вытесняет широко используемый до недавнего времени термин «компьютеризация общества». При внешней схожести этих понятий они имеют существенное различие.

При *компьютеризации общества* основное внимание уделяется развитию и внедрению компьютеров, обеспечивающих оперативное получение результатов переработки информации и ее накопление.

При *информатизации общества* основное внимание уделяется комплексу мер, направленных на полное использование достоверного, исчерпывающего и своевременного знания во всех видах человеческой деятельности.

Таким образом, «информатизация общества» является более широким понятием, чем «компьютеризация общества», и направлена на скорейшее овладение информацией для удовлетворения своих потребностей. В понятии «информатизация общества» акцент надо делать не столько на технических средствах, сколько на сущности и цели социально-технического прогресса. Компьютеры являются базовой технической составляющей процесса информатизации общества.

Информатизация на базе внедрения компьютерных и телекоммуникационных технологий является реакцией общества на потребность в существенном увеличении производительности труда в информационном секторе общественного производства, где сосредоточено более половины трудоспособного населения. Так, например, в информационной сфере США занято более 60 % трудоспособного населения, в СНГ — около 40 %.

Пример. Согласно экспертным оценкам, в США вклад информационного сектора (информационные системы и телекоммуникации, средства массовой информации) в ВВП в следующем десятилетии удвоится, что приведет к созданию 3–5 млн новых рабочих мест главным образом в малых и средних инновационных фирмах. Только вокруг Интернета уже возникло примерно 400 тыс. рабочих мест. Япония,

в свою очередь, надеется создать в секторе мультимедиа в ближайшие 15 лет 2,5 млн рабочих мест, страны ЕС – 6 млн.

Опыт США опровергает широко распространенное мнение, что инновационные процессы снижают качество и ухудшают содержание труда, разделяя рабочую силу на своего рода рабочую аристократию – «тех, кто знает» – и основную массу работников низкой квалификации. В действительности в последнее десятилетие значительная часть новых рабочих мест была создана в третичном секторе, где работники выполняют функции менеджеров, специалистов, консультантов и мелких предпринимателей, нередко работая неполный день. Например, на предприятиях, производящих программное обеспечение, с высокой долей специалистов в числе занятых за пять лет было создано 250 тыс. новых рабочих мест, в то время как в обрабатывающей промышленности их количество сократилось на 116 тыс. В секторе финансовых услуг за последние десять лет было ликвидировано 80 тыс. рабочих мест, связанных с оказанием традиционных услуг, но появилось 500 тыс. рабочих мест в таких новых областях, как управление ценными бумагами и акциями, обслуживание кредитных карт.

2.1.3. Опыт информатизации и перспективные идеи

В настоящее время все страны мира в той или иной степени осуществляют процесс информатизации. Неправильно выбранная стратегия информатизации или ее недостаточные динамизм и мобильность могут привести к существенным, а подчас драматическим изменениям во всех сферах жизни страны. Как известно, первая страна, которая начала информатизацию, – это США. Другие промышленно развитые страны мира, поняв перспективность и неизбежность этого направления, достаточно быстро сориентировались и стали наращивать темпы внедрения компьютеров и средств телекоммуникаций. В настоящее время вся деловая и политическая пресса США полна бесконечных дискуссий о потере рынков сбыта этой страной в компьютерной, телекоммуникационной и микроэлектронной областях из-за вытеснения США другими развитыми странами (Японией, Германией и др.).

Постоянная тенденция к сокращению доли востребованных потребителями американских товаров на внутреннем рынке привела к многочисленным проблемам в экономике США. Для выхода из создавшегося положения были приняты меры по интенсификации информатизации всех сторон деятельности американского общества, а именно:

- увеличение инвестирования в новые исследовательские разработки;
- улучшение качества образования;
- развитие международного сотрудничества на стадии разработки продукта;
- повышение качества рабочей силы и ряд других мероприятий.

В результате США не только сохранили положительное сальдо в процессе торговли технологиями, но и увеличили его с 2000 по 2007 г. в полтора раза. К сожалению, Россия за это же время из небольшого положительного сальдо ушла в глубоко отрицательную область (–796 млн долларов США).

Этот опыт важно учесть при разработке государственной политики информатизации нашей страны, так как, производя информационные технологии, можно

получить все преимущества и условия для развития других высоких технологий и экономики. В большинстве развитых стран понимают, что без чрезвычайных усилий отставание в области информационных и коммуникационных технологий может стать необратимым для их развития в целом. Руководители некоторых стран «третьего мира» с нарастающей тревогой наблюдают за все большим отставанием их от промышленно развитых стран, осуществляющих информатизацию. Это может привести к тому, что страна будет восприниматься как сырьевой придаток сообщества информационно и промышленно развитых стран. Это в полной мере относится и к России.

В любой стране независимо от уровня ее развития понимают в той или иной мере неизбежность и необходимость претворения в жизнь идей информатизации общества. Многие страны имеют национальные программы информатизации с учетом местных особенностей и условий. Однако при создании и внедрении таких программ следует опираться на опыт передовых стран, учитывать их успехи и неудачи, отражать в этих программах существующие и перспективные тенденции информатизации.

Для успешной реализации программы информатизации желательно следовать общим для всего мирового сообщества принципам, среди которых нужно отметить:

- необходимость замены экономической структуры, основанной на тяжелой промышленности, структурой, базирующейся на наукоемких отраслях;
- признание приоритетного характера информационного сектора, создание новой инфраструктуры и сектора услуг, способных поддержать национальную экономику, в качестве основы успешного экономического развития;
- широкое использование достижений мировой науки и техники;
- вложение значительных финансовых средств в информатизацию, как государственную, так и частную.

Результатом процесса информатизации является создание информационного общества, где манипулируют не материальными объектами, а символами, идеями, образами, интеллектом, знаниями. Если рассматривать человечество в целом, то оно в настоящее время переходит от индустриального общества к информационному.

Для каждой страны ее движение от индустриального этапа развития к информационному определяется степенью информатизации общества.

2.1.4. Роль средств массовой информации

Одной из отличительных особенностей жизни в современном обществе является стремительное развитие средств массовой информации (газеты, журналы, кино, телевидение, радио). Поставленные современными научно-техническими разработками на качественно новый уровень и объединенные средствами связи в мировые информационно-коммуникационные сети, они оказывают чрезвычайно сильное влияние на психологию громадной массы людей во всем мире. Особенно сильно и отчетливо это обнаруживается в наиболее развитых странах Западной Ев-

ропы, США, Японии, Великобритании. С помощью средств массовой информации достигается манипулирование общественным мнением, создаются необходимые психологические предпосылки для формирования политических решений в различных сферах деятельности.

Развитию средств массовой информации во многом способствует процесс информатизации общества. Появление новых технических средств, информационных технологий, телекоммуникаций и др. обеспечивает своевременный сбор, накопление, оперативную обработку и передачу информации в любую точку мирового пространства. Как следствие, становится возможным принятие оперативных решений и целенаправленных воздействий на общество. Это одна из причин, вследствие которых правительства наиболее передовых стран в последние годы стали уделять большое внимание развитию информационной сферы производства. Наряду с позитивным влиянием информатизации общества на средства массовой информации существует и негативное. Так, ряд ученых во многих странах заявляют, что технический прогресс в сфере массовых коммуникаций служит в некоторых случаях социальному регрессу общества, так как порой разрушает веками создаваемые социальные коммуникационные связи.

В свою очередь, и средства массовой информации могут оказывать влияние на процесс информатизации общества, рекламируя новые информационные продукты и услуги, формируя общественное мнение о приоритетности этого процесса по сравнению с другими, о первостепенной важности проводимых мероприятий по его интенсификации, о роли информационной сферы в модели будущего информационного общества.

2.1.5. Об информационной культуре

В период перехода к информационному обществу необходимо подготовить человека не только к решению описанных проблем, но и к быстрому восприятию и обработке больших объемов информации, к овладению современными средствами, методами и технологией работы. Кроме того, новые условия работы порождают зависимость информированности одного человека от информации, приобретенной другими людьми. Поэтому уже недостаточно уметь самостоятельно осваивать и накапливать информацию, надо научиться такой технологии работы с информацией, когда подготавливаются и принимаются решения на основе коллективного знания. Это говорит о том, что человек должен иметь определенный уровень культуры обращения с информацией. Для отражения этого факта было введено понятие информационной культуры.

Информационная культура — это умение целенаправленно работать с информацией и использовать для ее получения, обработки и передачи компьютерную информационную технологию, современные технические средства и методы.

Приведем еще одно определение информационной культуры: «Информационная культура в узком смысле — это уровень достигнутого в развитии информационного общения людей, а также характеристика информационной сферы жизнедея-

тельности людей, в которой мы можем отметить степень достигнутого, количество и качество созданного, тенденции развития, степень прогнозирования будущего».

Для свободной ориентации в информационном потоке человек должен обладать информационной культурой как одной из составляющих общей культуры. Информационная культура связана с социальной природой человека. Она является продуктом разнообразных творческих способностей человека и проявляется в следующих аспектах:

- ┐ в конкретных навыках по использованию технических устройств (от телефона до персонального компьютера и компьютерных сетей);
- ┐ в способности применять в своей деятельности компьютерную информационную технологию, базовой составляющей которой являются многочисленные программные продукты;
- ┐ в умении извлекать информацию из различных источников: как из периодической печати, так и из электронных средств, представлять ее в понятном виде, эффективно использовать;
- ┐ во владении основами аналитической переработки информации;
- ┐ в умении работать с различной информацией;
- ┐ в знании особенностей информационных потоков в своей области деятельности.

Информационная культура вбирает в себя знания из тех наук, которые способствуют ее развитию и приспособлению к конкретному виду деятельности (информатика, теория информации, математика, теория проектирования баз данных и ряд других дисциплин). Неотъемлемой частью информационной культуры являются знание новой информационной технологии и умение ее применять как для автоматизации рутинных операций, так и в неординарных ситуациях, требующих нетрадиционного творческого подхода.

В информационном обществе необходимо начать овладевать информационной культурой с детства, сначала с помощью электронных игрушек, а затем привлекая персональный компьютер. Для высших учебных заведений социальным заказом информационного общества следует считать обеспечение уровня информационной культуры студента, необходимого для работы в конкретной сфере деятельности. В процессе привития информационной культуры студенту в вузе наряду с изучением теоретических дисциплин информационного направления много времени необходимо уделить компьютерным информационным технологиям, являющимся базовыми составляющими будущей сферы деятельности. Причем качество обучения должно определяться степенью закрепленных устойчивых навыков работы в среде базовых информационных технологий при решении типовых задач выбранной сферы деятельности.

В информационном обществе центр тяжести приходится на общественное производство, где существенно повышаются требования к уровню подготовки всех его участников. Поэтому в программе информатизации следует особое внимание уделить информатизации образования как направления, связанного с приобретением и развитием информационной культуры человека. Это, в свою очередь, ставит образование в положение «объекта» информации, где требуется так изме-

нить содержание подготовки, чтобы обеспечить будущему специалисту не только общеобразовательные и профессиональные знания в области информатики, но и необходимый уровень информационной культуры. Повсеместное внедрение персонального компьютера во все сферы народного хозяйства, новые его возможности по организации «дружественной» программной среды, ориентированной на пользователя, применение телекоммуникационной связи, обеспечивающей новые условия для совместной работы специалистов, применение информационных технологий для самой разнообразной деятельности, постоянно растущая потребность в специалистах, способных ее осуществлять, ставят перед государством проблему о пересмотра всей системы подготовки на современных технологических принципах. В нашей стране решение этой проблемы находится на начальной стадии, поэтому целесообразно учесть опыт наиболее развитых стран, к числу которых относятся США, Япония, Англия, Германия, Франция, где этот процесс уже получил значительное развитие.

2.2. Информационный потенциал общества

2.2.1. Информационные ресурсы

Ресурс — это запасы, источники чего-нибудь.

Такая трактовка приведена в словаре русского языка С. И. Ожегова. В индустриальном обществе, где большая часть усилий направлена на материальное производство, известно несколько основных видов ресурсов, ставших уже классическими экономическими категориями:

- ❑ *материальные ресурсы* — совокупность предметов труда, предназначенных для использования в процессе производства общественного продукта, например: сырье, материалы, топливо, энергия, полуфабрикаты, детали и т. д.;
- ❑ *природные ресурсы* — объекты, процессы, условия природы, используемые обществом для удовлетворения материальных и духовных потребностей людей;
- ❑ *трудовые ресурсы* — люди, обладающие общеобразовательными и профессиональными знаниями для работы в обществе;
- ❑ *финансовые ресурсы* — денежные средства, находящиеся в распоряжении государственной или коммерческой структуры;
- ❑ *энергетические ресурсы* — носители энергии, например: уголь, нефть, нефтепродукты, газ, гидроэнергия, электроэнергия и т. д.

В информационном обществе акцент внимания и значимости смещается с традиционных видов ресурсов на информационный ресурс, который, хотя всегда существовал, не рассматривался ни как экономическая, ни как иная категория: никто специально о нем не говорил и тем более не вводил никаких определений.

Одним из ключевых понятий при информатизации общества стало понятие информационных ресурсов, толкование и обсуждение которого велось с того момента, когда начали говорить о переходе к информационному обществу. Этому вопросу посвящено довольно много публикаций, в которых отразились как разные мнения и определения, так и разные научные школы, рассматривающие эти понятия.

С принятием Федерального закона «Об информации, информатизации и защите информации» большая часть неопределенности была снята. Руководствуясь не научной стороной этого вопроса, а скорее прагматической позицией потребителя информации, целесообразно воспользоваться тем определением, которое приведено в этом законе. Тем более нельзя не учитывать тот факт, что юридическое толкование во всех случаях является для пользователя информации опорой при защите его прав.

Информационные ресурсы — это отдельные документы и отдельные массивы документов, документы и массивы документов в информационных системах (библиотеках, архивах, фондах, банках данных, других информационных системах).

Надо понимать, что документы и массивы информации, о которых говорится в этом законе, не существуют сами по себе. В них в разных формах представлены знания, которыми обладали люди, создававшие их. Таким образом, информационные ресурсы — это знания, подготовленные людьми для социального использования и зафиксированные на материальном носителе.

Информационные ресурсы общества, если их понимать как знания, отчуждены от тех людей, которые их накапливали, обобщали, анализировали, создавали и т. п. Эти знания материализовались в виде документов, баз данных, баз знаний, алгоритмов, компьютерных программ, а также произведений искусства, литературы, науки.

В настоящее время не разработана методология количественной и качественной оценки информационных ресурсов, а также прогнозирования потребностей общества в них. Это снижает эффективность информации, накапливаемой в виде информационных ресурсов, и увеличивает продолжительность переходного периода от индустриального к информационному обществу. Кроме того, неизвестно, какой объем трудовых ресурсов должен быть задействован в сфере производства и распространения информационных ресурсов в информационном обществе. Несомненно, в будущем эти проблемы будут решены.

Информационные ресурсы страны, региона, организации должны рассматриваться как стратегические ресурсы, аналогичные по значимости запасам сырья, энергии, ископаемых и прочим ресурсам.

Развитие мировых информационных ресурсов позволило:

- превратить деятельность по оказанию информационных услуг в глобальную человеческую деятельность;
- сформировать мировой и внутригосударственный рынок информационных услуг;
- образовать всевозможные базы данных ресурсов регионов и государств, к которым возможен сравнительно недорогой доступ;

- повысить обоснованность и оперативность принимаемых решений в фирмах, банках, биржах, промышленности, торговле и др. за счет своевременного использования необходимой информации.

2.2.2. Информационные продукты и услуги

Информационные ресурсы являются базой для создания информационных продуктов. Любой информационный продукт отражает информационную модель его производителя и воплощает его собственное представление о конкретной предметной области, для которой он создан. Информационный продукт, являясь результатом интеллектуальной деятельности человека, должен быть зафиксирован на материальном носителе любого физического свойства в виде документов, статей, обзоров, программ, книг и т. д.

Информационный продукт — это совокупность данных, сформированная производителем для распространения в вещественной или неведественной форме.

Информационный продукт может распространяться такими же способами, как и любой другой материальный продукт, с помощью услуг.

Услуга — это вид деятельности, направленный на удовлетворение человеческой потребности, в процессе которой не создается новый продукт, но изменяется качество существующего.

Информационная услуга — это услуга, ориентированная на удовлетворение информационных потребностей человека путем предоставления информационных продуктов.

В узком смысле информационная услуга часто воспринимается как услуга, получаемая с помощью компьютеров, хотя на самом деле это понятие намного шире.

При предоставлении услуги заключается соглашение (договор) между двумя сторонами — предоставляющей и потребляющей услугу. В договоре указываются срок ее использования и соответствующее этому вознаграждение.

Перечень услуг определяется объемом, качеством, предметной ориентацией по сфере использования информационных ресурсов и создаваемых на их основе информационных продуктов.

Пример. Библиотеки являются местом сосредоточения значительной части информационных ресурсов страны. Перечислим основные виды информационных услуг, оказываемых библиотечной сферой:

- предоставление полных текстов документов, а также справок по их описанию и местонахождению;
- выдача результатов библиографического поиска и аналитической переработки информации (справки, указатели, дайджесты, обзоры и пр.);
- получение результатов фактографического поиска и аналитической переработки информации (справки, таблицы, фирменные досье);

- организация научно-технической пропаганды и рекламной деятельности (выставки новых поступлений, научно-технические семинары, конференции и т. п.);
- выдача результатов информационного исследования (аналитические справки и обзоры, отчеты, рубрикаторы перспективных направлений, конъюнктурные справки и т. д.).

Информационные услуги возникают только при наличии баз данных в компьютерном или некомпьютерном варианте. Под *базой данных* понимают совокупность связанных данных, правила организации которых основаны на общих принципах описания данных, хранения данных и манипулирования данными.

Базы данных являются источником и своего рода полуфабрикатом при подготовке информационных услуг соответствующими службами. Базы данных, хотя они так и не назывались, существовали и до компьютерного периода в библиотеках, архивах, фондах, справочных бюро и других подобных организациях. В них содержатся всевозможные сведения о событиях, явлениях, объектах, процессах, публикациях и т. п.

С появлением компьютеров существенно увеличиваются объемы хранимых баз данных и соответственно расширяется круг информационных услуг.

Рассмотрим классификацию баз данных с позиций их использования для систематизации информационных услуг и продуктов.

Базы данных принято разделять на библиографические и небиблиографические.

Библиографические базы данных содержат вторичную информацию о документах, включая рефераты и аннотации.

Небиблиографические базы данных имеют множество видов:

- справочные базы данных содержат информацию о различных объектах и явлениях, например: адреса, расписания движения, телефоны магазинов и т. п.;
- базы данных полного текста содержат первичную информацию, например: статьи, журналы, брошюры и т. п.;
- числовые базы данных содержат количественные характеристики и параметры объектов и явлений, например: химические и физические данные, статистические и демографические данные и т. п.;
- текстово-числовые базы данных содержат описания объектов и их характеристики, например, по промышленной продукции, фирмам, странам и т. п.;
- финансовые базы данных содержат финансовую информацию, предоставляемую банками, биржами, фирмами и т. п.;
- юридические базы данных содержат правовые документы по отраслям, регионам, странам.

Исходя из возможных видов информационных продуктов, баз данных и ресурсов классификация информационных услуг представлена на рис. 2.1.

Выпуск информационных изданий означает подготовку печатной продукции: библиографических и других указателей, реферативных сборников, обзорных изданий, справочных изданий.

Информационные издания подготавливаются практически всеми видами информационных служб, органов и систем. Эти издания содержат вторичную информацию, которая создается на основе работы с базами данных, причем предоставление возможности такой работы также является услугой.



Рис. 2.1. Основные виды информационных услуг

Ретроспективный поиск информации – это целенаправленный по заявке пользователя поиск информации в базе данных и пересылка результатов либо по почте в виде распечаток, либо по электронной почте в виде файлов.

Предоставление первоисточника является традиционной услугой библиотечных служб. Эта услуга предусматривает выдачу не только первоисточников, но и их копий, полученных с помощью устройств различного принципа действия.

Традиционные услуги научно-технической информации осуществляются по предварительному заказу и включают в себя:

- подготовку обзоров в виде рукописей;
- подготовку переводов текстов.

Дистанционный доступ к удаленным базам данных организуется в компьютерной сети в диалоговом режиме. Популярность услуг дистанционного доступа к базам данных нарастает быстрыми темпами и опережает все виды других услуг благодаря

- все большему числу пользователей, овладевших информационной технологией работы в коммуникационной среде компьютерных сетей;
- высокой оперативности предоставления услуг;
- возможности отказа от собственных информационных систем.

Традиционно основными пользователями услуг дистанционного доступа к базам данных являются организации. Однако за последние годы наметилась тенденция к существенному увеличению числа индивидуальных пользователей.

В основном эти услуги предоставляются специальными организациями, называемыми вычислительными центрами коллективного пользования, располагающими мощными ЭВМ с внешней памятью более сотен гигабайтов и лазерными принтерами. Дистанционный доступ к базам данных может быть предоставлен по подписке на основе абонементной платы или по договорам. Схема оплаты может быть разная, но в основном это почасовая оплата, зависящая от объема получаемой информации.

Услуги дистанционного доступа к базам данных можно классифицировать следующим образом:

- ❑ *непосредственный доступ* к базам данных может быть организован с локального места пользователя только при условии его обученности работе в коммуникационной среде, в противном случае следует воспользоваться услугами, предоставляемыми специальными организациями;
- ❑ *косвенный доступ* включает в себя организацию обучения пользователей, выпуск бюллетеня новостей, организацию справочной службы, организацию встреч с пользователем для выяснения интересующих его вопросов, рассылку вопросников пользователям;
- ❑ *услуга загрузки (downloading)* позволяет получить на свой персональный компьютер результаты поиска в центральной базе данных для дальнейшего использования в качестве персональной базы данных;
- ❑ *регулярный поиск* предусматривает регулярное проведение поиска в массивах одной или нескольких центральных баз данных и предоставление результатов поиска на терминал пользователю в удобное для него время.

В сферу подготовки и оказания информационных услуг входят:

- ❑ связь (телефонная, телекоммуникационная), служащая для предоставления информационных услуг в форме передачи данных;
- ❑ обработка данных в вычислительных центрах;
- ❑ программное обеспечение;
- ❑ разработка информационных систем;
- ❑ разработка информационных технологий.

2.2.3. Рынок информационных продуктов и услуг

Назначение рынка

Как и при использовании традиционных видов ресурсов и продуктов, люди должны знать: где находятся информационные ресурсы, сколько они стоят, кто ими владеет, кто в них нуждается, насколько они доступны.

Ответы на эти вопросы можно получить, если существует рынок информационных продуктов и услуг.

Рынок информационных продуктов и услуг (информационный рынок) — это система экономических, правовых и организационных отношений по торговле продуктами интеллектуального труда на коммерческой основе.

Информационный рынок характеризуется определенной номенклатурой продуктов и услуг, условиями и механизмами их предоставления, ценами. В отличие от торговли обычными товарами, имеющими материально-вещественную форму, здесь в качестве предмета продажи или обмена выступают информационные системы, информационные технологии, лицензии, патенты, товарные знаки, ноу-хау, инженерно-технические услуги, различного рода информация и прочие виды информационных ресурсов.

Основным источником информации для информационного обслуживания в современном обществе являются базы данных. Они интегрируют в себе поставщиков и потребителей информационных услуг, связи и отношения между ними, порядок и условия продажи и покупки информационных услуг.

Поставщиками информационных продуктов и услуг могут быть:

- центры, где создаются и хранятся базы данных, а также производятся постоянное накопление и редактирование информации;
- центры, распределяющие информацию на основе разных баз данных;
- службы телекоммуникации и передачи данных;
- специальные службы, куда стекается информация по конкретной сфере деятельности для ее анализа, обобщения, прогнозирования, например: консалтинговые фирмы, банки, биржи;
- коммерческие фирмы;
- информационные брокеры.

Потребителями информационных продуктов и услуг могут быть различные юридические и физические лица.

ПРИМЕЧАНИЕ

Основную информацию об участниках и состоянии информационного рынка России еще недавно можно было получать из печатных изданий, таких как справочники «Вся компьютерная Москва», «Кто есть кто на компьютерном рынке», «Кто есть кто на компьютерном рынке России», или из журнала «Информационные ресурсы России». Однако состояние информационного рынка характеризуется большим динамизмом: рынок непрерывно растет, растет аудитория Интернета, подавляющее количество участников информационного рынка имеют свои интернет-представительства и все большее их число присутствует только в Интернете. По этой причине актуальную информацию о состоянии информационного рынка России и его участниках можно получить только в Интернете, используя такие ресурсы, как поисковые системы Google и Yandex, специализированные разделы тематических каталогов Rambler и Апорт, а также специальные сайты-справочники, такие как Yell.ru.

История развития рынка информационных услуг

С середины 50-х гг. началось формирование устойчивого рынка информационных услуг. Основными поставщиками информационных услуг являлись информационные службы академических, профессиональных и научно-технических обществ, государственных учреждений, учебных заведений. Основные потребители — ученые и специалисты в области науки и техники.

С начала 60-х гг. параллельно с рынком информационных услуг начал формироваться рынок услуг электронной обработки и передачи информации.

С середины 60-х до середины 70-х гг. в результате широкого внедрения компьютерной техники важнейшим видом информационных услуг стали базы данных, содержащие разные виды информации по всевозможным отраслям знаний.

Начиная с середины 70-х гг. с созданием национальных и глобальных сетей передачи данных ведущим видом информационных услуг стал диалоговый поиск информации в удаленных от пользователя базах данных.

Начиная с 80-х гг. информационная индустрия приобретает все больший удельный вес и влияние на экономическую и социальную жизнь общества.

СПРАВКА

До середины 80-х гг. лидирующее место на рынке информационных услуг со значительным отрывом от других стран занимали США. Причем государственная политика была сориентирована на повышение роли рынка и сокращение роли государства. С середины 80-х гг. Япония и страны Западной Европы догнали США во многом благодаря смешанному характеру экономики всей индустрии информации с преобладанием государственного сектора.

В настоящее время на информационном рынке наметилось отставание США от темпов роста информационных услуг в Японии и других странах, что побудило правительство США принять ряд мер и скорректировать государственную политику в сторону увеличения государственного регулирования и дополнительного финансирования системы научно-технических коммуникаций.

Структура рынка информационных продуктов и услуг

Совокупность средств, методов и условий, позволяющих использовать информационные ресурсы, составляет *информационный потенциал* общества. Это не только весь индустриально-технологический комплекс производства современных средств и методов обработки и передачи информации, но и сеть научно-исследовательских, учебных, административных, коммерческих и других организаций, обеспечивающих информационное обслуживание на базе современной информационной технологии.

В настоящее время в России быстрыми темпами идет формирование рынка информационных продуктов и услуг, имеющего несколько важнейших составляющих.

- *Техническая и технологическая составляющая* — это современное информационное оборудование, мощные компьютеры, развитая компьютерная сеть и соответствующие им технологии переработки информации.

Пример. В настоящее время в России получают распространение современные мировые технические достижения благодаря возможности работы в глобальной компьютерной сети (Интернете), позволившей вывести информационные ресурсы России на мировой рынок. По данным статистического подразделения популярной поисковой системы Rambler (<http://rismetrika.rambler.ru/>) количество пользователей Интернета с 2002 по 2010 г. выросло почти в 5 раз и составило более 35 млн человек.

- **Нормативно-правовая составляющая** — это юридические документы: законы, указы, постановления, которые обеспечивают цивилизованные отношения на информационном рынке.

Пример. К числу таких законов относятся законы «Об информации, информатизации и защите информации», «Об авторском праве и смежных правах», «О правовой охране программ для ЭВМ и баз данных», «О правовой охране топологий интегральных схем».

- **Информационная составляющая** — это справочно-навигационные средства и структуры, помогающие находить нужную информацию.
- **Организационная составляющая** — это элементы государственного регулирования взаимодействия производителей и распространителей информационных продуктов и услуг.

Рассматриваемый далее пример демонстрирует роль государства в регулировании рыночной экономики в период нестабильности. В нашей стране, претерпевающей серьезные экономические изменения, организационный фактор государственной политики становится особенно актуальным. Следовательно, формирование информационного рынка и решение всех сопутствующих этому процессу проблем наше государство во многом должно взять на себя.

Пример. Трудности, возникшие в последнее время в экономике США, породили ряд проблем, в решении которых возросла плановая и стабилизирующая роль государства как регулятора экономических, технологических и социальных отношений:

- во-первых, американские предприниматели, переставшие быть монополистами на мировом рынке и встретившие жесткую конкуренцию со стороны Японии, Германии и других стран, начинают искать поддержку у государства;
- во-вторых, государство оказалось вынужденным перераспределять общественные фонды с тем, чтобы помочь увеличению занятости, которая последние годы в связи с внедрением новых информационных технологий (НИТ) постоянно уменьшается;
- в-третьих, в связи с сокращением военно-промышленного комплекса государство вынуждено направить поток высвобождающихся высококвалифицированных специалистов в наукоемкие технологии, где требуются огромные затраты на их разработку и реализацию.

Россия последовательно продвигается вперед по пути формирования пакета государственных информационных услуг. Благодаря личному вкладу и интересу

к данным вопросам президента и правительства России за последние десять лет в этой сфере произошли значительные изменения: органы власти практически всех субъектов Российской Федерации имеют свои представительства в Интернете на региональном уровне и большинство — на районном и муниципальном уровнях. Это обеспечивает возможность гражданам получать дополнительную информацию о работе властных структур, обращаться к органам государственного управления через Интернет, способствует формированию открытого информационного общества.

Пример. В конце 2009 г. был открыт портал <http://www.gosuslugi.ru/>, который является единой точкой доступа граждан и организаций к информации о государственных услугах, предоставляемых органами исполнительной власти Российской Федерации, а также о возможности получения этих услуг. В перспективе этот портал должен не только обеспечить граждан информацией об услугах, но и позволить получать услуги государственных органов непосредственно в режиме подключения.

В нашей стране в силу переходных процессов в экономике и начавшейся информатизации общества пока отсутствует единое мнение относительно инфраструктуры информационного рынка. Мы предлагаем вам следующее толкование этого понятия.

Инфраструктура информационного рынка — это совокупность секторов, каждый из которых объединяет группу людей или организаций, предлагающих однородные информационные продукты и услуги.

Инфраструктуру информационного рынка можно представить пятью секторами:

- научно-техническая продукция в виде проектных, технологических, методических разработок по разным отраслям;
- объекты художественной культуры в виде текстовой, визуальной и аудиопроductии;
- услуги образования — все виды обучений;
- управленческие данные и сообщения: политическая и хозяйственная информация, статистические данные, данные о рыночной ситуации, рекламные сообщения, оценки и рекомендации по принятию решений;
- бытовая информация: сообщения общего характера, сведения о потребительском рынке, сведения о рынке труда.

В этой инфраструктуре отсутствуют три основных аспекта, характерные для информационного общества:

- технический — вся аппаратно-техническая база информатики;
- программный — все программные продукты информатики;
- коммуникационный — все виды компьютерных сетей и их инфраструктура передачи информации, все виды телефонной и факсимильной связи.

Кроме того, существует мнение относительно разделения на секторы рынка дистанционных информационных услуг:

- высокомонополизированный профессиональный рынок предоставляет пользователю информацию по его сфере деятельности;
- услуги поддержки управленческих решений позволяют использовать в фирме информацию о состоянии внешних по отношению к ней систем;
- домашний рынок предоставляет услуги бытового характера, например, для электронных платежных операций и покупок.

Уточненная, с учетом этих мнений, инфраструктура информационного рынка представлена на рис. 2.2.

Выделим пять секторов рынка информационных продуктов и услуг.

- 1-й сектор — *деловая информация* — состоит из следующих частей:
 - биржевая и финансовая информация — котировки ценных бумаг, валютные курсы, учетные ставки, рынок товаров и капиталов, инвестиции, цены. Поставщиками являются специальные службы биржевой и финансовой информации, брокерские компании, банки;
 - статистическая информация — ряды динамики, прогнозные модели и оценки по экономической, социальной, демографической областям. Поставщиками являются государственные службы, компании, консалтинговые фирмы;
 - коммерческая информация по компаниям, фирмам, корпорациям, направлениям работы и их продукции, ценам; о финансовом состоянии, связях, сделках, руководителях, деловых новостях в области экономики и бизнеса. Поставщиками являются специальные информационные службы.
- 2-й сектор — *информация для специалистов* — содержит следующие части:
 - профессиональная информация — специальные данные и информация для юристов, врачей, фармацевтов, преподавателей, инженеров, геологов, метеорологов и т. д.;
 - научно-техническая информация — документальная, библиографическая, реферативная, справочная информация в области естественных, технических, общественных наук, по отраслям производства и сферам человеческой деятельности;
 - доступ к первоисточникам — организация доступа к источникам информации через библиотеки и специальные службы, возможности приобретения первоисточников, их получения по межбиблиотечному абонементу в различных формах.
- 3-й сектор — *потребительская информация* — состоит из следующих частей:
 - новости и литература — информация служб новостей и агентств прессы, электронные журналы, справочники, энциклопедии;
 - потребительская информация — расписания транспорта, резервирование билетов и мест в гостиницах, заказ товаров и услуг, банковские операции и т. п.;
 - развлекательная информация — игры, телетекст, видеотекст.



Рис. 2.2. Секторы информационного рынка

- 4-й сектор *услуги образования* — включает в себя все формы и ступени образования: дошкольное, школьное, специальное, среднепрофессиональное, высшее, повышение квалификации и переподготовку. Информационная продукция может быть представлена в компьютерном или некомпьютерном виде: учебники, методические разработки, практикумы, развивающие компьютерные игры, компьютерные обучающие и контролирующие системы, методики обучения и пр.
- 5-й сектор — *обеспечивающие информационные системы и средства* — состоит из следующих частей:
 - программные продукты — программные комплексы с разной ориентацией — от профессионала до неопытного пользователя компьютера: системное программное обеспечение, программы общей ориентации, прикладное программное обеспечение по реализации функций в конкретной области принадлежности, по решению задач типовыми математическими методами и др.;
 - технические средства — компьютеры, телекоммуникационное оборудование, оргтехника, сопутствующие материалы и комплектующие;
 - разработка и сопровождение информационных систем и технологий — обследование организации в целях выявления информационных потоков, разработка концептуальных информационных моделей, разработка структуры программного комплекса, создание и сопровождение баз данных;
 - консультирование по различным аспектам информационной индустрии — какую приобретать информационную технику, какое программное обеспечение необходимо для реализации профессиональной деятельности, нужна ли информационная система и какая, на базе какой информационной технологии лучше организовать свою деятельность и т. д.;
 - подготовка источников информации — создание баз данных по заданной теме, области, явлению и т. п.

ПРИМЕЧАНИЕ

В каждом секторе может быть организован любой вид доступа:

- непосредственный к хранилищу информации на бумажных носителях;
- дистанционный доступ к удаленным или находящимся в данном помещении компьютерным базам данных.

Информационный рынок, несмотря на разные концепции и мнения относительно его инфраструктуры, существует и развивается, а значит, можно говорить о бизнесе информационных продуктов, услуг, под которым понимается не только торговля и посредничество, но и производство.

Функции информационного бизнеса:

- управление финансами и ведение учета;
- управление кадрами;
- материально-техническое снабжение;
- организация производства;
- маркетинговые исследования;
- лизинговые операции;
- консультационное обслуживание;
- страхование имущества и информации;
- организация службы информационной безопасности;
- сервисное обслуживание.

Правовое регулирование на информационном рынке

Развитие рыночных отношений в информационной деятельности поставило вопрос о защите информации как объекта интеллектуальной собственности и имущественных прав на нее. В Российской Федерации, как уже отмечалось, принят ряд указов, постановлений, законов:

- «Об информации, информатизации и защите информации»;
- «Об авторском праве и смежных правах»;
- «О правовой охране программ для ЭВМ и баз данных»;
- «О правовой охране топологий интегральных схем».

Рассмотрим основные положения закона «Об информации, информатизации и защите информации», который является базовым юридическим документом, открывающим путь к принятию дополнительных нормативных законодательных актов, падающих на успешное развитие информационного общества. С его помощью удалось частично решить ряд вопросов правового регулирования на информационном рынке, касающихся защиты прав и свобод личности от угроз и ущерба, связанных с искажением, порчей, уничтожением «персональной» информации.

Закон состоит из 25 статей, сгруппированных по пяти главам:

- Общие положения.
- Информационные ресурсы.

- Пользование информационными ресурсами.
- Информатизация, информационные системы, технологии и средства их обеспечения.
- Защита информации и прав субъектов в области информационных процессов и информатизации.

В законе определены цели и основные направления государственной политики в сфере информатизации. Информатизация определяется как важное новое стратегическое направление деятельности государства. Указано, что государство должно заниматься формированием и реализацией единой государственной научно-технической и промышленной политики в сфере информатизации.

Закон создает условия для включения России в международный информационный обмен, предотвращает бесхозность отношения к информационным ресурсам и информатизации, обеспечивает информационную безопасность и права юридических и физических лиц на информацию. В нем определяются комплексное решение проблемы организации информационных ресурсов и правовые положения по их использованию, а также предлагается рассматривать информационные ресурсы в двух аспектах:

- как материальный продукт, который можно покупать и продавать;
- как интеллектуальный продукт, на который распространяется право интеллектуальной собственности, авторское право.

Закон закладывает юридические основы гарантий прав граждан на информацию. Он направлен на урегулирование важнейшего вопроса экономической реформы — формы, права и механизма реализации собственности на накопленные информационные ресурсы и технологические достижения. Кроме того, он обеспечивает защиту собственности в сфере информационных систем и технологий, что способствует формированию цивилизованного рынка информационных ресурсов, услуг, систем, технологий, средств их обеспечения.

Ввод в действие закона, обеспечение выполнения его положений гарантируют, что государство получит значительную экономию средств и необходимые условия для более устойчивого развития экономики и построения демократического общества в России.

2.3. Информатика — предмет и задачи

2.3.1. Появление и развитие информатики

Термин *информатика* возник в 60-х гг. во Франции для названия области, занимающейся автоматизированной обработкой информации с помощью электронных вычислительных машин. Французский термин *informatique* (информатика) образован путем слияния слов *information* (информация) и *automatique* (автоматика) и означает буквально информационную автоматику, или автоматизированную переработку информации. В англоязычных странах этому термину соответствует синоним *computer science* (наука о компьютерной технике).

Выделение информатики как самостоятельной области человеческой деятельности и первую очередь связано с развитием компьютерной техники. Причем основная заслуга в этом принадлежит микропроцессорной технике, появление которой в середине 70-х гг. послужило началом второй электронной революции. С этого времени элементной базой вычислительной машины стали интегральные схемы и микропроцессоры, а область, связанная с созданием и использованием компьютеров, получила мощный импульс в своем развитии. Термин «информатика» приобретает новый смысл и используется не только в отношении компьютерной техники, но и связывается с процессами передачи и обработки информации.

В нашей стране подобная трактовка термина «информатика» утвердилась с момента принятия в 1983 г. на сессии годовичного собрания Академии наук СССР решения об организации нового отделения информатики, вычислительной техники и автоматизации. Информатика трактовалась как «комплексная научная и инженерная дисциплина, изучающая все аспекты разработки, проектирования, создания, оценки, функционирования основанных на ЭВМ систем переработки информации, их применения и воздействия на различные области социальной практики».

Информатика в таком понимании нацелена на разработку общих методологических принципов построения информационных моделей. Поэтому методы информатики применимы всюду, где существует возможность описания объекта, явления, процесса и т. п. с помощью информационных моделей.

Существует множество определений информатики, что связано с многогранностью ее функций, возможностей, средств и методов. Обобщая опубликованные в литературе по информатике определения этого термина, предлагаем такую трактовку.

Информатика — это область человеческой деятельности, связанная с процессами преобразования информации с помощью компьютеров и их взаимодействием со средой применения.

Часто возникает путаница в понятиях «информатика» и «кибернетика». Попытаемся разъяснить их сходство и различие.

Основная концепция, заложенная Н. Винером в кибернетику, связана с разработкой теории управления сложными динамическими системами в разных областях человеческой деятельности. Кибернетика существует независимо от наличия или отсутствия компьютеров.

Кибернетика — это наука об общих принципах управления в различных системах: технических, биологических, социальных и др.

Информатика занимается изучением процессов преобразования и создания новой информации более широко, практически не решая, в отличие от кибернетики, задачи управления различными объектами. Поэтому может сложиться впечатление об информатике как о более емкой дисциплине, чем кибернетика. Однако информатика не занимается решением проблем, не связанных с использованием компьютерной техники, что, несомненно, сужает ее, казалось бы, обобщающий характер. Между этими двумя дисциплинами провести четкую границу не представляется

возможным в связи с ее размытостью и неопределенностью, хотя существует довольно распространенное мнение о том, что информатика является одним из направлений кибернетики.

Информатика появилась благодаря развитию компьютерной техники, базируется на ней и совершенно немыслима без нее. Кибернетика же развивается сама по себе, строя различные модели управления объектами, хотя и очень активно использует все достижения компьютерной техники. Кибернетика и информатика внешне очень похожие дисциплины и различаются, скорее всего, в расстановке акцентов:

- в информатике акцент делается на свойствах информации и аппаратно-программных средствах ее обработки;
- в кибернетике акцент делается на разработке концепций и построении моделей объектов с использованием, в частности, информационного подхода.

2.3.2. Структура информатики

Информатика в широком смысле представляет собой единство разнообразных отраслей науки, техники и производства, связанных с переработкой информации главным образом с помощью компьютеров и телекоммуникационных средств связи во всех сферах человеческой деятельности.

Информатику в узком смысле можно представить как состоящую из трех взаимосвязанных частей — технических (hardware), программных (software) и алгоритмических средств (brainware). В свою очередь, информатику в целом и каждую ее часть обычно рассматривают с разных позиций (рис. 2.3): как производственную отрасль, как фундаментальную науку, как прикладную дисциплину.

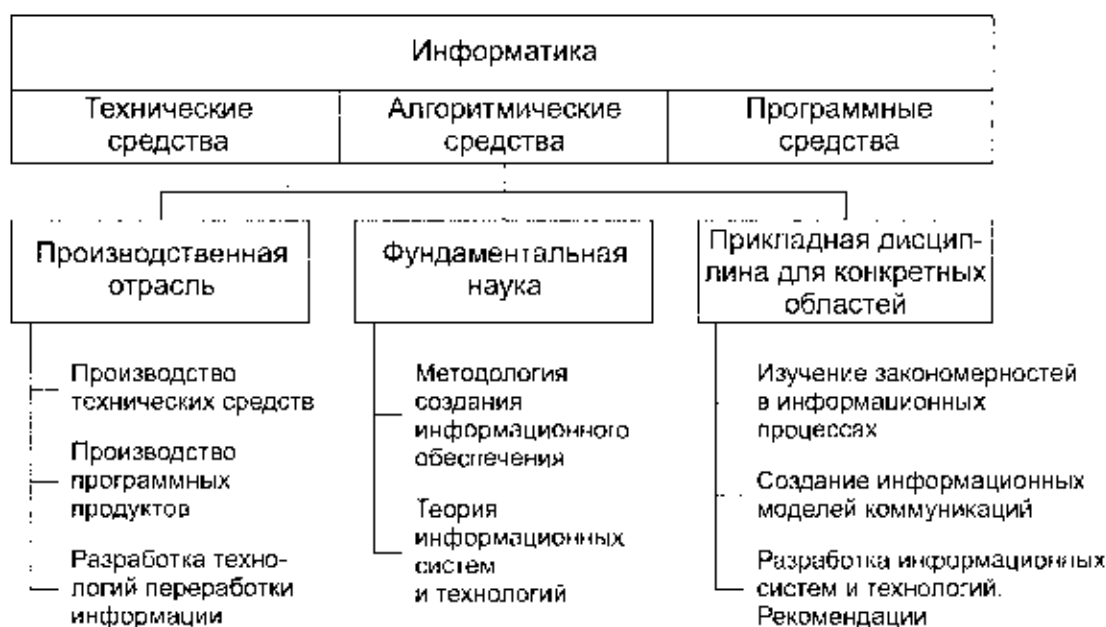


Рис. 2.3. Структура информатики как отрасли, науки, прикладной дисциплины

Информатика как *производственная отрасль* состоит из однородной совокупности предприятий разных форм хозяйствования, где занимаются производством компьютерной техники, программных продуктов и разработкой современной технологии переработки информации. Специфика и значение информатики как

отрасли производства состоят в том, что от нее во многом зависит рост производительности труда в других отраслях народного хозяйства. Более того, для нормального развития этих отраслей производительность труда в самой информатике должна возрастать более высокими темпами, так как в современном обществе информация все чаще выступает как предмет конечного потребления: людям необходима информация о событиях, происходящих в мире, о предметах и явлениях, относящихся к их профессиональной деятельности, о развитии науки и самого общества. Дальнейший рост производительности труда и уровня благосостояния возможен лишь на основе использования новых интеллектуальных средств и человеко-машинных интерфейсов, ориентированных на прием и обработку больших объемов мультимедийной информации (текст, графика, видеоизображение, звук, анимация). При отсутствии достаточных темпов увеличения производительности труда в информатике может произойти существенное замедление роста производительности труда во всем народном хозяйстве. В настоящее время около 50 % всех рабочих мест в мире поддерживается средствами обработки информации.

Информатика как *фундаментальная наука* занимается разработкой методологии создания информационного обеспечения процессов управления любыми объектами на базе компьютерных информационных систем. Существует мнение о том, что одна из главных задач этой науки — выяснение, что такое информационные системы, какое место они занимают, какую должны иметь структуру, как функционировать, какие общие закономерности им свойственны. В Европе можно выделить следующие основные научные направления в области информатики: разработка сетевой структуры, компьютерно-интегрированные производства, экономическая и медицинская информатика, информатика социального страхования и окружающей среды, профессиональные информационные системы.

Цель фундаментальных исследований в информатике — получение обобщенных знаний о любых информационных системах, выявление общих закономерностей их построения и функционирования.

Информатика как *прикладная дисциплина* занимается:

- изучением закономерностей в информационных процессах (накопление, переработка, распространение информации);
- созданием информационных моделей коммуникаций в различных областях человеческой деятельности;
- разработкой информационных систем и технологий в конкретных областях и выработкой рекомендаций относительно их жизненного цикла: этапов проектирования и разработки, производства, функционирования и т. д.

Главная функция информатики заключается в разработке методов и средств преобразования информации и их использовании с целью организации технологического процесса переработки информации.

Задачи информатики состоят в следующем:

- исследование информационных процессов любой природы;
- разработка информационной техники и создание новейшей технологии переработки информации на базе полученных результатов исследования информационных процессов;

- решение научных и инженерных проблем создания, внедрения и обеспечения эффективного использования компьютерной техники и технологии во всех сферах общественной жизни.

Информатика существует не сама по себе, а является комплексной научно-технической дисциплиной, призванной создавать новые информационные техники и технологии для решения проблем в других областях. Она предоставляет методы и средства исследования другим областям, даже таким, где считается невозможным применение количественных методов из-за неформализуемости процессов и явлений. Особенно следует выделить в информатике методы математического моделирования и методы распознавания образов, практическая реализация которых стала возможной благодаря достижениям компьютерной техники.

Комплекс индустрии информатики станет ведущим в информационном обществе. Тенденция ко все большей информированности в обществе в существенной степени зависит от прогресса информатики как единства науки, техники и производства.

Вопросы для самопроверки

1. Расскажите об информационных революциях в истории развития цивилизации.
2. Сопоставьте процессы, происходящие в истории развития ЭВМ, с последней информационной революцией.
3. Определите суть информационных технологий и телекоммуникаций.
4. Как вы себе представляете информационное общество?
5. В чем проявляется информационный кризис?
6. В чем состоит процесс информатизации?
7. В чем различие процессов компьютеризации и информатизации?
8. Дайте определение информационной культуре. Как она проявляется?
9. Чем определяется информационный потенциал общества?
10. Расскажите о видах ресурсов.
11. Охарактеризуйте информационный ресурс, информационный продукт, информационную услугу. Приведите примеры.
12. Как вы понимаете базу данных?
13. Расскажите о классификации основных видов информационных услуг.
14. Что такое «рынок информационных услуг и продуктов»? Каковы составляющие рынка информационных услуг и продуктов?
15. Расскажите о пяти секторах рынка информационных услуг и продуктов.
16. В чем заключается правовое регулирование на информационном рынке?
17. Как и для чего появилась информатика?
18. Расскажите об информатике как об отрасли, как о науке, как о прикладной дисциплине. Опишите цели и задачи.

Литература

1. Введение в информационный бизнес: Учебное пособие / Под ред. В. П. Тихомирова, А. В. Хоропилова. М.: Финансы и статистика, 1996.
2. Майоров С. И. Информационный бизнес: коммерческое распространение и маркетинг. М.: Финансы и статистика, 1993.
3. Мелюхин И. С. Рынок электронных информационных продуктов и услуг в России: состояние и тенденции развития / ПТИ. – Серия 1. ВИНТИ. 1994. № 2.
4. Тамбовцев В. Л. Пятый рынок: экономические проблемы производства информации. М.: Изд-во МГУ, 1993.

Глава 3 Теоретические основы управления знаниями

- 3.1. Управление знаниями
- 3.2. Данные, информация и знания
- 3.3. Модели представления знаний
- 3.4. Системы, основанные на знаниях
- 3.5. Инженерия знаний

Управление знаний представляет собой одно из быстро развивающихся направлений информатики. Все современные достижения компьютерной индустрии, начиная от микрочипов, встроенных в бытовую технику, и заканчивая распределенными информационными системами суперкомпьютеров, так или иначе связаны с процессом получения, хранения, переработки и применения знаний. В этой главе даны теоретические основы и описаны основные технологии, позволяющие управлять знаниями.

3.1. Управление знаниями

3.1.1. Общее представление об управлении знаниями

Несмотря на то что само определение знаний гораздо старше и шире, чем определение информации, в компьютерно-кибернетическую эпоху информации уделяется большое внимание. В конце двадцатого и в начале двадцать первого века управление знаниями становится все более и более актуальным.

Во-первых, сформировался запрос к достижениям в этой области со стороны бизнеса. Так, руководители бизнеса в самых разных отраслях стали осознавать, что накопление все больших массивов все более детализированных данных и превращение этих данных во все более мощные информационные потоки в информационных процессах зачастую не стимулируют, а отягощают бизнес-процессы. Возникла потребность не только в информации, но и в стандартизированном умении этой информацией пользоваться. Нужны методы и средства по извлечению, формализации, хранению и использованию знаний.

Во-вторых, компьютеры и компьютерные сети достигли достаточно больших вычислительных мощностей, чтобы работать в режиме, схожем с режимом работы мозга человека, то есть в режиме реального времени осуществлять печеткие вычисления, обучать большие нейронные сети, осуществлять самообучение.

В-третьих, в науке были открыты методы получения и формализации знаний такого рода, которые еще в середине прошлого века считались уникальными и неповторимыми. Сразу несколько направлений в психологии, из которых наиболее популярно и известно нейролингвистическое программирование (Neuro-Linguistic Programming, NLP), доказали, что есть возможность переводить человеческий опыт в самых разнообразных сферах деятельности на язык формальных описаний, а затем, используя подобные описания, обучать этому опыту и навыкам работы с ним других людей. Как оказалось, все, от искусства продаж до навыков гипноза, от самой глубокой психофизической саморегуляции до умения продуктивно и творчески мыслить, может быть извлечено из мозга эксперта, превращено в передаваемые знания и усвоено другими людьми.

Такое сочетание потребности и возможностей неизбежно ведет к появлению новой парадигмы обработки информации в компьютерной индустрии. Эта парадигма не замедлила появиться под названием «управление знаниями» (knowledge management) и ныне приобретает все большую популярность и широкое распространение.

Что такое «управление знаниями», что можно и нужно относить к нему, что включить в это понятие, что исключить — на этот счет существует множество точек зрения и моделей. На рис. 3.1 представлена четырехслойная модель предметной области управления знаниями.

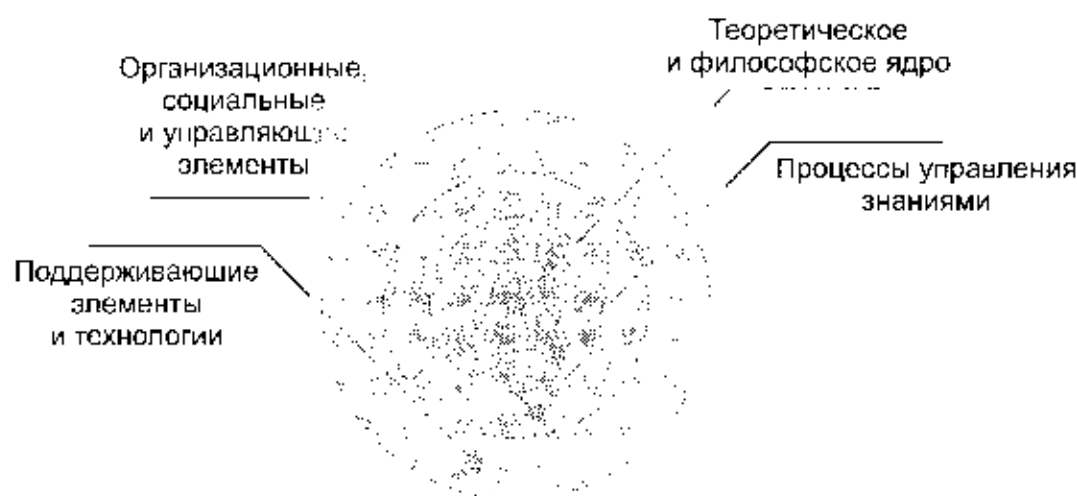


Рис. 3.1. Четырехслойная модель предметной области управления знаниями

Теоретическое и философское ядро управления знаниями составляют как философские, так и естественнонаучные труды, исследования и произведения как древних (в большей степени), так и современных мыслителей, которые старались определить само понятие «знания», классифицировать и упорядочить виды и типы знаний, способы работы со знаниями. К философско-теоретическому ядру можно отнести труды таких ученых, как Аристотель, Платон, Маршалл, Маймонид, Гегель, Декарт, Кант, Поппер, Риль, Хайдеггер, Гадамер, Милский и другие.

К процессам управления знаниями относятся:

- *извлечение знаний*, включая создание, исследование, накопление, проверку полезности и применимости знаний;
- *организация знаний*, включая моделирование, классификацию, калибровку и интеграцию знаний;
- *поставка знаний*, включая распространение, техническую поддержку, совместное и повторное использование знаний.

Организационные, социальные и управляющие элементы составляют третий слой четырехслойной модели. К этому слою относятся такие понятия, как знания и память организации, корпоративная культура, конкурентные преимущества, взаимодействие внутри и вне организации, интеллектуальный капитал, стратегии использования и передачи знаний, обеспечение сохранности, надежности и приватности знания, измерение знаний, использование знаний в социальных сообществах и сетях, мотивация к использованию и созданию знаний у сообществ и индивидов, а также архитектура, интеграция и жизненный цикл системы управления знаниями.

Поддерживающие элементы и технологии находятся во внешнем слое, и только они непосредственно связаны с технической стороной организации и управления знаниями. К этим элементам и технологиям можно отнести:

- сетевую инфраструктуру, мобильную и кабельную;
- хранилища данных, как структурированные и интеллектуальные, так и неструктурированные индексированные (то есть в чистом виде базы данных);
- приложения, обеспечивающие семантическое и онтологическое структурирование знаний;
- приложения, обеспечивающие безопасность, разделение доступа и шифрование данных;
- системы мета-знаний (экспертные системы, системы искусственного интеллекта), обеспечивающие частичную или полную автоматизацию процесса управления знаниями;
- корпоративные распределенные системы и порталы, обеспечивающие взаимодействие людей и знаний;
- технологии извлечения и представления знаний;
- программные агенты, обеспечивающие автоматизированное накопление знаний;

Существуют и многие другие технологии, оформленные непосредственно в виде приложений пользователя или приложений среднего звена, а также аппаратное обеспечение этих технологий.

3.1.2. Модель жизненного цикла управления знаниями

Модель жизненного цикла управления знаниями в организации показана на рис. 3.2.

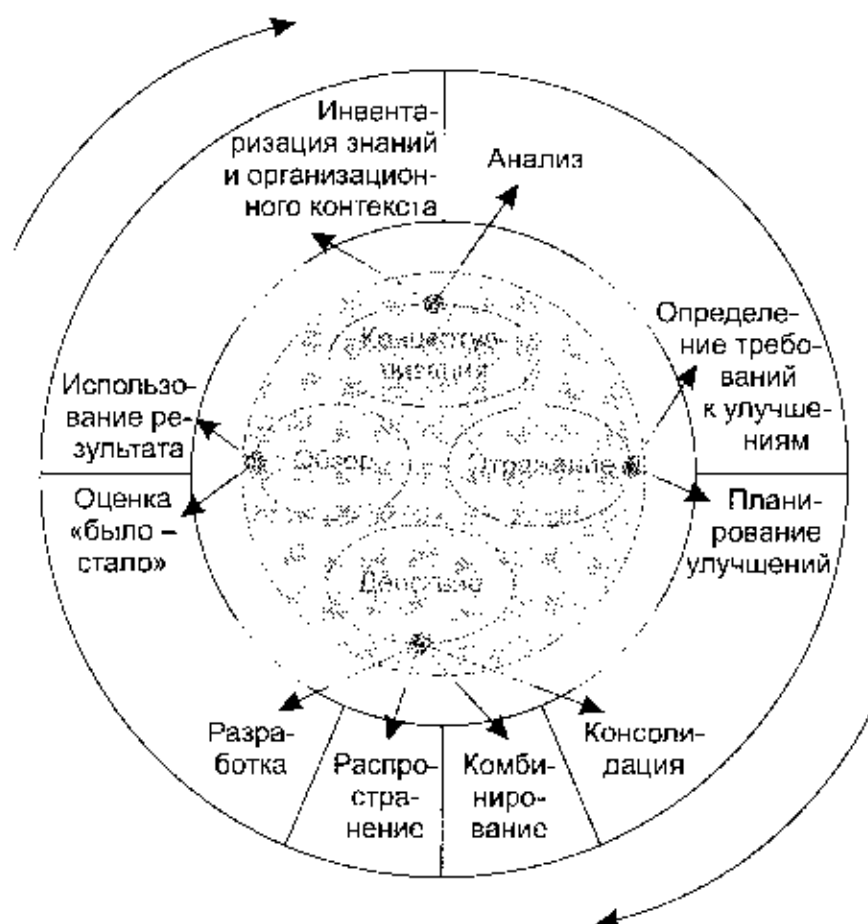


Рис. 3.2. Модель жизненного цикла управления знаниями

Жизненный цикл управления знаниями напоминает жизненный цикл разработки программного обеспечения. Различие состоит в том, что при каждой итерации жизненного цикла программного обеспечения происходит улучшение функциональных возможностей программы, а при каждой итерации жизненного цикла управления знаниями происходит увеличение количества и качества знания в организации.

3.2. Данные, информация и знания

Традиционно возникает вопрос: что же такое «знания» и чем они отличаются от обычных данных, десятилетиями обрабатываемых компьютером? Можно предложить несколько рабочих определений, в рамках которых это становится очевидным. Как отмечено в главе 1, под данными понимается закодированная определенным способом информация.

При обработке на компьютере данные трансформируются, условно проходя несколько этапов.

1. Получение результатов измерений и наблюдений, фиксируемых на материальных носителях информации (таблицы, протоколы, справочники).

2. Разработка моделей (структур) данных в виде диаграмм, графиков, функций.
3. Ввод данных в компьютер на языке описания данных.
4. Создание и поддержка базы данных на машинных носителях.

Знания связаны с данными, основываются на них, но представляют собой результат мыслительной деятельности человека, обобщают его опыт, полученный в ходе выполнения какой-либо практической деятельности. Они получаются эмпирическим путем.

Знания — это выявленные закономерности предметной области (принципы, связи, законы), позволяющие решать задачи в этой области.

Часто под знаниями понимают *структурированные данные, данные о данных, или метаданные*.

Пример. В компьютере хранятся данные, представляющие собой число $-5,0$. В процессе извлечения, представления и осмысления этих данных они, в зависимости от описываемой предметной области, превращаются, например, в информацию о температуре воздуха на улице. В этом случае знанием становится закон (понимание), определяющий то, что при такой температуре необходимо надеть теплую одежду.

При обработке на компьютере знания трансформируются аналогично этапам преобразования данных.

1. Добыча (извлечение) знаний и их фиксация на материальных носителях информации (таблицы, протоколы, учебники, методические пособия).
2. Разработка поля знаний — условного описания основных объектов предметной области, их атрибутов, связывающих их закономерностей.
3. Описание знаний на языках представления знаний в соответствии с выбранной моделью (продукционные языки, семантические сети, фреймы — см. далее).
4. Создание и поддержка базы знаний на машинных носителях.

Для хранения данных используются базы данных (для них характерны большой объем и относительно небольшая удельная стоимость информации), для хранения знаний — базы знаний (небольшого объема, но исключительно дорогие информационные массивы). **База знаний** — основа любой интеллектуальной системы.

Создание баз данных может быть в значительной степени автоматизировано, а информационные системы, основанные на базах данных, часто могут быть полностью автоматическими.

Пример. Процесс покупки товаров в магазине самообслуживания может быть автоматизирован полностью. Покупатель сам выбирает нужные ему товары, при выходе датчики считывают информацию с электронных меток и на табло отображается информация о счете. В случае, когда платеж производится при помощи электронной банковской карты, участие продавца может быть исключено полностью: после платежа специальное устройство дистанционно деактивирует метки, обеспечив покупателю беспрепятственный выход из магазина.

В этом случае процесс сбора данных совершается автоматически, а процесс принятия решения о совершении покупки поддается полной алгоритмизации.

В отличие от баз данных базы знаний требуют участия высококвалифицированного персонала на всех этапах работы с ними, как во время создания, так и во время использования в информационных системах.

Знания могут быть классифицированы по следующим категориям:

- *поверхностные знания* — это знания о видимых взаимосвязях между отдельными событиями и фактами в предметной области;
- *глубинные знания* представляют собой абстракции, аналогии, схемы, отображающие структуру и процессы в предметной области.

Современные экспертные системы работают в основном с поверхностными знаниями. Это связано с тем, что на данный момент нет адекватных моделей, позволяющих работать с глубинными знаниями.

Близко к предыдущему подразделение знаний на эксплицитные и тацитные:

- *эксплицитные знания* — это формализованные (описанные формальным языком) знания, предназначенные для передачи и отчужденные от носителя знаний;
- *тацитные знания* — это неформализованные (не поддающиеся описанию) знания, являющиеся результатом личного опыта и интуиции, неотчуждаемые от носителя знаний.

Основной задачей аналитика, занимающегося созданием базы знаний, является перевод знаний из тацитной в эксплицитную форму.

Кроме того, знания можно разделить на *процедурные* и *декларативные*. Исторически первичными были процедурные знания, то есть знания, «растворенные» в алгоритмах. Они управляли данными. Для их изменения требовалось изменять программы. Однако с развитием искусственного интеллекта приоритет данных постепенно изменялся, и все большая часть знаний сосредоточивалась в структурах данных (таблицы, списки, абстрактные типы данных).

Сегодня знания приобрели чисто декларативную форму, то есть знаниями считаются предложения, записанные на языках представления знаний, приближенных к естественному языку, и понятные неспециалистам.

3.3. Модели представления знаний

Существуют десятки моделей (или языков) представления знаний для различных предметных областей. Большинство из них может быть сведено к следующим классам:

- продукционные модели;
- семантические сети;
- фреймы;
- формальные логические модели.

Продукционная модель — модель, основанная на правилах, представляющих знания в виде предложений типа: *если* (условие), *то* (действие). База знаний при использовании продукционной модели представляет собой совокупность такого рода правил, записанных на формальном языке.

Под *условием* понимается некоторое предложение-образец, по которому осуществляется поиск в базе знаний, а под *действием* — операции, выполняемые при успешном исходе поиска. Действия могут быть промежуточными, выступающими далее как условия, и терминальными, или целевыми, завершающими работу системы.

Программа, управляющая перебором правил, называется машиной вывода. Чаще всего вывод бывает *прямым* (от данных к поиску цели) или *обратным* (от цели для ее подтверждения — к данным). Данные — это исходные факты, на основании которых запускается машина вывода — программа, перебирающая правила из базы.

Пример. Имеется фрагмент базы знаний из двух правил, первого (П1) и второго (П2):

П1: Если «котировки на акции падают» и «состояние рынка нестабильное», то «покупать».

П2: Если «биржевые индексы падают», то «состояние рынка нестабильное».

Предположим, в систему поступили данные — «котировки на акции падают» и «биржевые индексы падают».

Прямой вывод — исходи из данных, получить ответ.

Первый проход.

Шаг 1. Пробуем П1, не работает (не хватает данных «состояние нестабильное»).

Шаг 2. Пробуем П2, работает, в базу поступает факт «состояние нестабильное».

Второй проход.

Шаг 3. Пробуем П1, работает, активируется цель «покупать», которая и выступает как совет, который дает ЭС.

Обратный вывод — подтвердить выбранную цель при помощи имеющихся правил и данных.

Первый проход.

Шаг 1. Цель — «покупать»: пробуем П1 — данных «состояние нестабильное» нет, они становятся новой целью, и ищется правило, где цель в правой части.

Шаг 2. Цель «состояние нестабильное»: правило П2 подтверждает цель и активирует ее.

Второй проход.

Шаг 3. Пробуем П1, подтверждается искомая цель.

Продукционная модель чаще всего применяется в промышленных экспертных системах. Она привлекает разработчиков своей наглядностью, высокой степенью модульности, легкостью внесения дополнений и изменений, простотой механизма логического вывода.

3.3.1. Семантические сети

Термин «*семантическая*» означает «смысловая», а сама семантика — это наука, устанавливающая отношения между символами и объектами, которые они обозначают, то есть наука, определяющая смысл знаков.

Семантическая сеть — это ориентированный граф, вершины которого — понятия, а дуги — отношения между ними.

Понятия — это абстрактные или конкретные объекты, а **отношения** — связи типа «часть — целое», «производит», «влияет», «любит» и т. д.

Можно ввести несколько классификаций семантических сетей. Например, по количеству типов отношений:

- однородные — с единственным типом отношений;
- неоднородные — с различными типами отношений.

По типам отношений:

- бипарные, в которых отношения связывают два объекта;
- n -арные, в которых есть специальные отношения, связывающие более двух понятий.

Наиболее часто в семантических сетях используются следующие отношения:

- связи типа «часть — целое» («класс — подкласс», «элемент — множество» и т. п.);
- функциональные связи (определяемые обычно глаголами «производит», «влияет», «любит»...);
- количественные связи («больше», «меньше», «равно»...);
- пространственные связи («далеко от», «близко от», «за», «под», «над»...);
- временные связи («раньше», «позже», «в течение»...);
- атрибутивные связи («иметь свойство», «иметь значение»...);
- логические связи («и», «или», «не») и др.

Проблема поиска решения в базе знаний типа семантической сети сводится к задаче поиска фрагмента сети (подсети), соответствующей поставленному вопросу.

Пример. На рис. 3.3 изображена семантическая сеть. В качестве вершин используются понятия: Подразделение, ОГЭ (отдел главного энергетика), Лифт, Отис, Подъемный механизм, Двигатель, Человек, Вместимость.

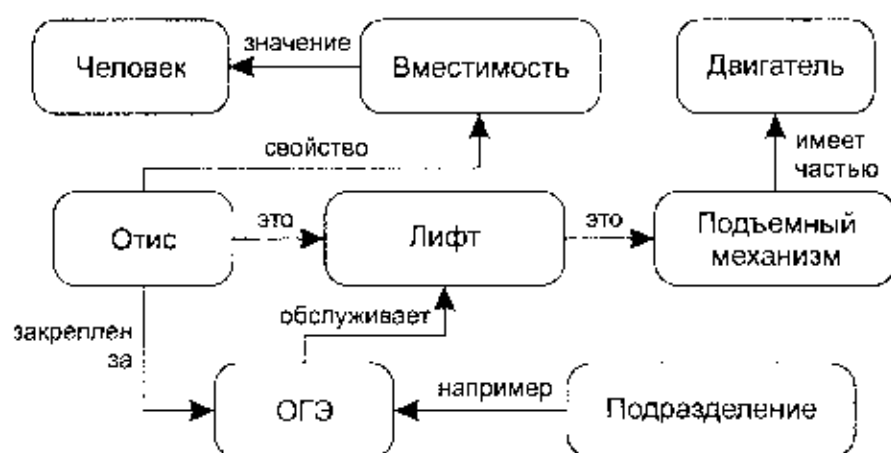


Рис. 3.3. Семантическая сеть

Основное преимущество этой модели — в ее соответствии современным представлениям об организации долговременной памяти человека. Недостаток модели — сложность поиска вывода.

3.3.2. Фреймы

Фрейм (от англ. frame — каркас, или рамка) предложен М. Минским в 70-е гг. как структура знаний, предназначенная для восприятия пространственных сцен. Эта модель, как и семантическая сеть, имеет глубокое психологическое обоснование.

Фрейм — это абстрактная структура, формализованная модель, описывающая факты, события, объекты или ситуации.

Свойства, характеризующие описываемый объект или ситуацию, в случае фрейм-модели именуются *слотами*. Каждый слот имеет название, и ему может быть присвоено значение.

Структуру фрейма можно представить так:

ИМЯ ФРЕЙМА:

(имя 1-го слота: значение 1-го слота).

(имя 2-го слота: значение 2-го слота).

(имя N-го слота: значение N-го слота).

Ту же запись можно представить в виде таблицы, дополнив двумя столбцами:

Имя фрейма			
Имя слота	Тип слота	Значение слота	Присоединенная процедура

В таблице дополнительные столбцы предназначены для описания типа слота и возможного присоединения к тому или иному слоту специальных процедур, что допускается в теории фреймов. В качестве значения слота может выступать имя другого фрейма — так образуют сети фреймов.

Различают *фреймы-образцы*, или *прототипы*, хранящиеся в базе знаний, и *фреймы-экземпляры*, которые создаются для отображения реальных ситуаций на основе поступающих данных. У фреймов-образцов все слоты или часть их не имеют значений, у фреймов-экземпляров все слоты заполнены значениями, описывающими конкретный объект, сценарий, ситуацию или роль.

Модель фрейма является достаточно универсальной, поскольку позволяет отобразить все многообразие знаний о мире через

- ☐ *фреймы-структуры* (обозначают объекты и понятия, такие как заем, залог, вексель);
- ☐ *фреймы роли* (менеджер, кассир, клиент);
- ☐ *фреймы-сценарии* (банкротство, собрание акционеров, празднование именин);
- ☐ *фреймы-ситуации* (тревога, авария, рабочий режим устройства) и др.

Важнейшим в теории фреймов является наследование свойств, заимствованное из теории семантических сетей. И во фреймах, и в семантических сетях наследование происходит по АКО-связям (от англ. A Kind Of — это). Слот АКО указывает на фрейм более высокого уровня иерархии, откуда неявно наследуются (то есть переносятся) значения аналогичных слотов.

Пример. В сети фреймов на рис. 3.4 понятие «программист» наследует свойства фреймов «инженер» и «служащий», которые находятся на более высоком уровне иерархии. Так, на вопрос: «Есть ли у инженера табельный номер?» — следует ответ: «Да», так как этим свойством обладают все служащие, что указано во фрейме «служащий». Таким образом, иерархическая связь между фреймами осуществляется через значение, указанное в слоте АКО.



Рис. 3.4. Сеть фреймов

Основным преимуществом фреймов в качестве модели представления знаний является способность отражать концептуальную основу организации памяти человека, можно также отметить ее гибкость и наглядность.

3.3.3. Формальные логические модели

Традиционно в представлении знаний выделяют *формальные логические модели*, основанные на классическом исчислении предикатов 1 порядка. Предикат — это то, что утверждается или отрицается о субъекте суждения.

Формальная логическая модель — это предметная область или задача описываемая в виде набора аксиом.

Пример. Мы можем обозначить предикатом РАВНО(x, y) отношение равенства ($x = y$), где x и y принадлежат множеству вещественных чисел. В этом случае предикат РАВНО будет принимать значение ИСТИНА для всех равных значений x и y .

Связывая между собой предикаты при помощи символов математической логики, можно описывать сложные логические высказывания на формальном языке. Для того, чтобы на основе исчисления предикатов создавать компьютерные модели, разработаны специальные языки логического программирования (одним из них является *prolog*).

Описание этих моделей опустим, так как исчисление предикатов I порядка в промышленных экспертных системах практически не используется. Эта логическая модель применима в основном в исследовательских «игрушечных» системах, так как предъявляет очень высокие требования и ограничения к предметной области.

В промышленных же экспертных системах используются различные ее модификации и расширения, изложение которых выходит за рамки темы данного учебника.

3.4. Системы, основанные на знаниях

Живые существа в зависимости от положения на эволюционной лестнице обладают большими или меньшими интеллектуальными способностями. Интеллектуальные возможности пчелы, птицы, дельфина и человека различаются во много раз. Тем не менее про каждое из этих существ можно сказать, что оно имеет возможность индивидуального выбора и принятия собственного решения. Это и есть один из признаков интеллекта.

В приложениях, построенных на основе искусственного интеллекта, делается попытка воспроизвести поведение живых существ средствами компьютерных систем. При этом инженер по знаниям решает довольно сложную задачу: он должен встроить в компьютерные программы поведение, свойственное живым существам.

Используя ряд методологий, которые включают в себя экспертные системы, нейронные сети, системы, основанные на прецедентах, генетические алгоритмы, интеллектуальные агенты и добычу данных, мы можем заставить компьютерные системы действовать сходно с живыми существами, воспроизводя небольшое количество аспектов интеллектуального поведения, таких как

- принятие решений, диагностика, планирование систем и ситуаций с использованием экспертных систем или нейронных сетей;
- прогнозирование развития сложных систем с помощью генетических алгоритмов;
- изучение предыдущего опыта (иногда единственного примера) и принятие на его основе решения по текущей ситуации средствами систем, основанных на прецедентах;
- распознавание рукописного ввода или графического изображения путем моделирования в нейронных сетях;
- установление причинно-следственных связей между наборами данных с помощью механизмов извлечения данных (тем самым можно идентифицировать причинно-следственные связи в реальном мире, из которого эти данные поступили);

- симуляция самостоятельного поведения компьютерной программы, реализуемая с помощью технологии интеллектуальных агентов.

Например, юридические информационные системы могут предлагать подходящие решения, основанные на примерах из прошлого опыта, используя механизм прецедентной аргументации (такие системы будут работать эффективно при прецедентной правовой системе, в которой решения принимаются с учетом прошлых судебных решений по сходным делам).

Системы, основанные на знаниях, — это компьютерные программы, спроектированные таким образом, чтобы воспроизводить работу экспертов в заданных областях знания.

Есть семь основных типов систем, основанных на знаниях.

1. *Экспертные системы* моделируют процесс принятия решения, свойственный человеческому мозгу. Они используются для того, чтобы действовать подобно экспертам-людям, помогая руководителям или специалистам принять решение, основываясь на экспертных знаниях. Типичные области применения экспертных систем — планирование, составление расписаний, диагностика. Хотя экспертные системы служат для того, чтобы воспроизвести процесс принятия решений, свойственный человеку, алгоритмы и правила, на основе которых принимаются решения, являются статическими. Это значит, что экспертная система не может самостоятельно модифицировать логику своей работы и не является самообучающейся.
2. *Нейронные сети* моделируют работу человеческого мозга на биологическом уровне (на уровне поведения отдельных клеток мозга, нейронов). Это означает, что нейронные сети обладают встроенной способностью к самообучению: они могут учиться распознавать шаблоны, а затем на основе шаблонов распознавать образы. Благодаря способности к самообучению, нейронные сети могут быть в некоторой степени использованы для решения задач прогнозирования на основе прошлого опыта (например, для прогнозирования суточного распределения затрат электроэнергии в энергосистеме).
3. *Системы, основанные на прецедентах*, моделируют человеческую способность принимать решение по аналогии с уже имеющимися прецедентами. Такого рода системы часто используют в службах технической поддержки, где схожие проблемные ситуации повторяются многократно у разных клиентов.
4. *Системы, построенные на генетических алгоритмах*, как явствует из названия, моделируют процесс развития биологических организмов на клеточном уровне. Такого рода алгоритмы предназначены для поиска одного лучшего из множества приемлемых вариантов решения проблем с большим количеством определяющих параметров. Поскольку биологический организм в процессе развития непрерывно решает задачу оптимизации всех своих параметров и установления равновесия с окружающей средой, постольку системы, основанные на генетических алгоритмах, могут применяться для решения задач оптимизации сложных многофакторных систем и прогнозирования вариантов их развития.

5. *Интеллектуальные агенты* — это программы, для которых задана конечная цель, но они могут самостоятельно выбирать пути достижения этой цели, гибко изменяя алгоритм своего поведения. Большинство интеллектуальных агентов являются фоновыми программами, ведущими черновую работу, и выходят на передний план только тогда, когда нужно передать пользователю результат. В качестве примеров интеллектуальных агентов можно назвать интернет-роботы поисковых систем, которые непрерывно просеивают миллионы страниц различных сайтов, чтобы поддерживать данные поисковой машины в актуальном состоянии.
6. *Системы добычи данных*. Термин «добыча данных» используют для того, чтобы описать процесс обнаружения знаний путем нахождения ранее неизвестных взаимосвязей между данными. В базах, в которых хранятся большие массивы данных, можно выявить неявные закономерности, имеющие практическую пользу. Например, анализируя продажи в супермаркете, можно выяснить, что летом в один и тот же временной интервал резко возрастает продажа лимонада, мороженого и семечек. Эти товары надо расположить в торговом зале рядом и обеспечить их бесперебойное поступление. Такие меры позволяют значительно увеличить продажу этих продуктов.

3.4.1. Экспертные системы

Экспертные системы должны обладать некоторыми специфическими характеристиками: для своей работы экспертная система использует знания, которые она должна уметь сохранять в базе знаний, извлекать и обрабатывать определенным образом для решения проблемы. Другими словами, экспертная система должна полностью заменять эксперта-человека в какой-либо специфической области деятельности.

Экспертная система — это реализованный при помощи компьютерных технологий и основанный на знаниях программный продукт, главной функцией которого является способность предложить обоснованную рекомендацию по пути решения проблемы или сформулировать само решение.

Экспертные системы создаются при помощи языков программирования, поддерживающих программирование, основанное на правилах. Обычно в качестве таких языков выступают Prolog и Lisp.

Два требования являются особенно важными для экспертных систем:

- *Достаточно узкая предметная область*. Чем уже предметная область, в которой реализуются функции экспертной системы, тем легче создать для нее набор правил и тем эффективнее выданное экспертное заключение.
- *Возможность обратного вывода* (см. ранее пример продукционной модели) позволяет отследить, на основе каких исходных данных вырабатывалась экспертная рекомендация.

Основными элементами экспертной системы являются (рис. 3.5):

- модуль хранения знаний (база знаний), в котором знания сохраняются в различных форматах представления;

- механизм логического вывода, который использует базу знаний для решения проблем;
- интерфейс пользователя, через который пользователь применяет экспертную систему, формулирует проблему и получает ее решение;
- механизм получения знаний, при помощи которого формируются базы знаний различного назначения (для различных предметных областей).

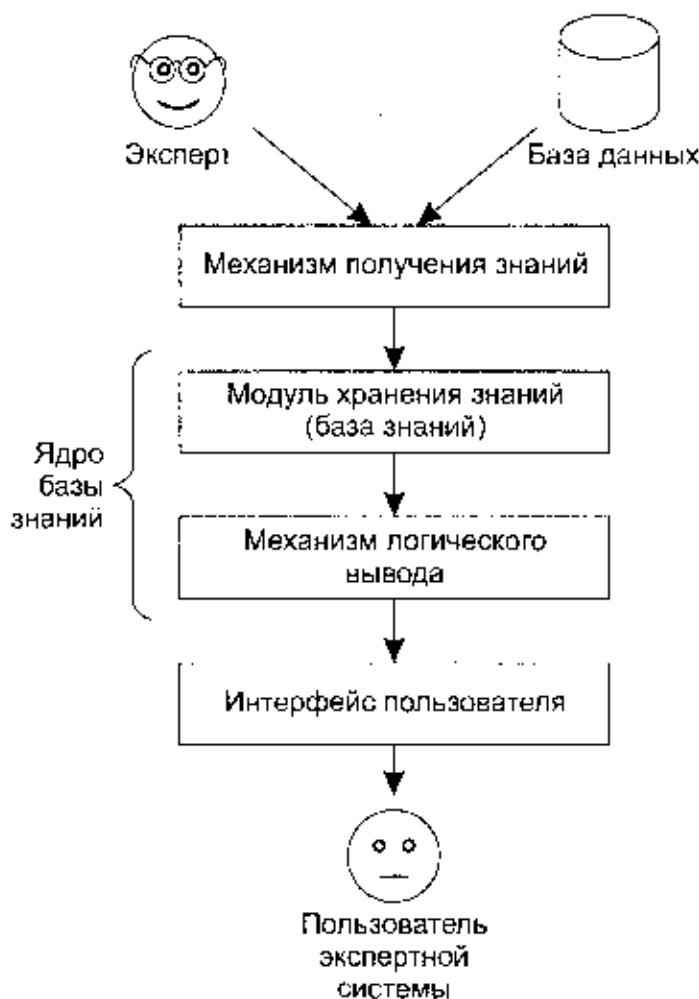


Рис. 3.5. Элементы экспертной системы

Достоинства экспертных систем можно отнести:

- процесс принятия решения происходит на основе заранее установленных правил;
- раздельное использование базы знаний и механизма логического вывода позволяет раздельно модифицировать то и другое;
- возможность узнать, как было получено решение;
- быстрота получения решения;
- наличие стандартной процедуры вывода;
- возможность многократного решения типовых задач с освобождением от этой работы экспертов-людей;
- расширение возможностей пользователей до возможностей опытных экспертов в определенной области.

Недостатки экспертных систем:

- ориентация на узкую область знаний, на решение проблем с ограниченной спецификой;
- потребность в экспертах для приобретения знаний;
- частая потребность в эксперте для окончательного заключения;
- высокая степень формализации, неспособность применять в рассуждениях «простой здравый смысл»;
- высокая стоимость разработки и обслуживания;
- высокая степень механистичности выводов, отсутствие «творческих способностей»;
- потребность в постоянном пополнении базы знаний в соответствии с изменениями в экспертной области;
- трудоемкость получения знаний от экспертов и особые требования к инженеру по знаниям (процесс перевода интуиции эксперта в символы и выражения формальной логики может занимать много времени и быть очень дорогим).

Экспертные системы применяются там, где

- решаемые проблемы настолько важны, что применение экспертной системы помогает сэкономить время и деньги или то и другое;
- правила экспертизы стабильны и хорошо известны (иными словами, есть доступные эксперты, от которых достаточно легко получить знания, что позволяет построить экспертную систему, предназначенную для многократного и длительного применения);
- создать и содержать экспертную систему дешевле, чем содержать в штате экспертов-людей;
- есть много потребителей, нуждающихся именно в такой экспертной системе (в этом случае однажды созданная экспертная система может быть многократно воспроизведена и применена разными потребителями);
- легче и дешевле создать экспертную систему, чем обучать персонал новой области деятельности;
- решение легко формализуется и принимается на основе устойчивых правил (без привнесения интуиции или плохо формализуемых знаний).

Различные по стоимости, мощности, объему базы знаний и аппаратной реализации экспертные системы используются для принятия решений, получения дополнительных сведений, повышения надежности или снижения стоимости решения проблем в разных отраслях науки, техники, экономики, медицины и т. д. На рис. 3.6 приведена классификация экспертных систем по разным признакам.

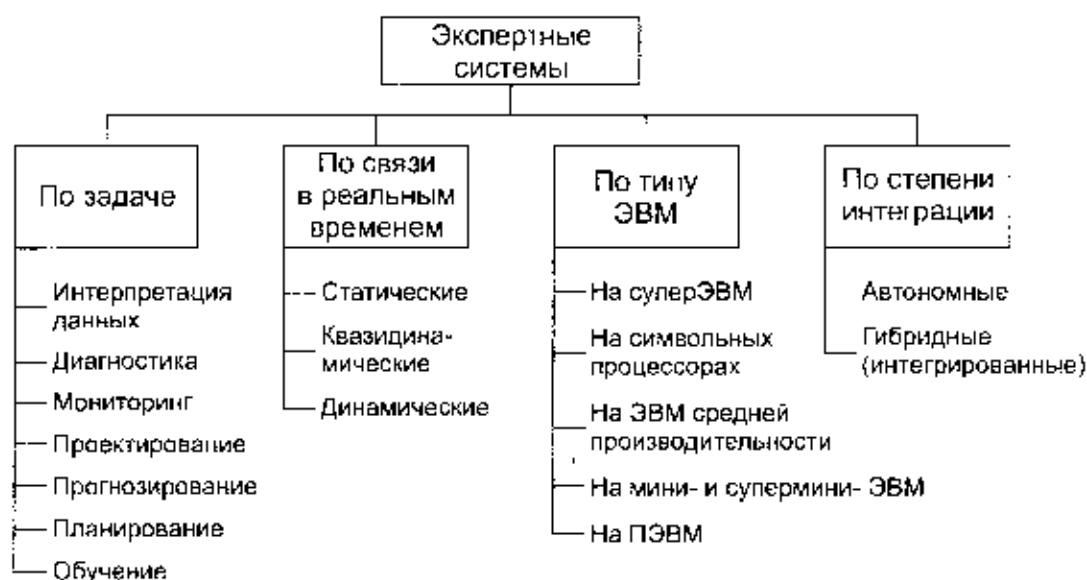


Рис. 3.6. Классификация экспертных систем

3.4.2. Нейронные сети

Искусственные нейронные сети представляют собой один из интереснейших с точки зрения информатики случаев систем, основанных на знаниях. Искусственные нейронные сети возникли из многочисленных попыток воспроизвести при помощи математики и компьютера абстрактную модель совместной работы клеток головного мозга.

Нейрон, основная строительная единица биологического первичного вещества, представляет собой клетку, в которой множество входных сигналов обрабатывается по определенному алгоритму, и затем, как результат этой обработки, формируется выходной сигнал (рис. 3.7).

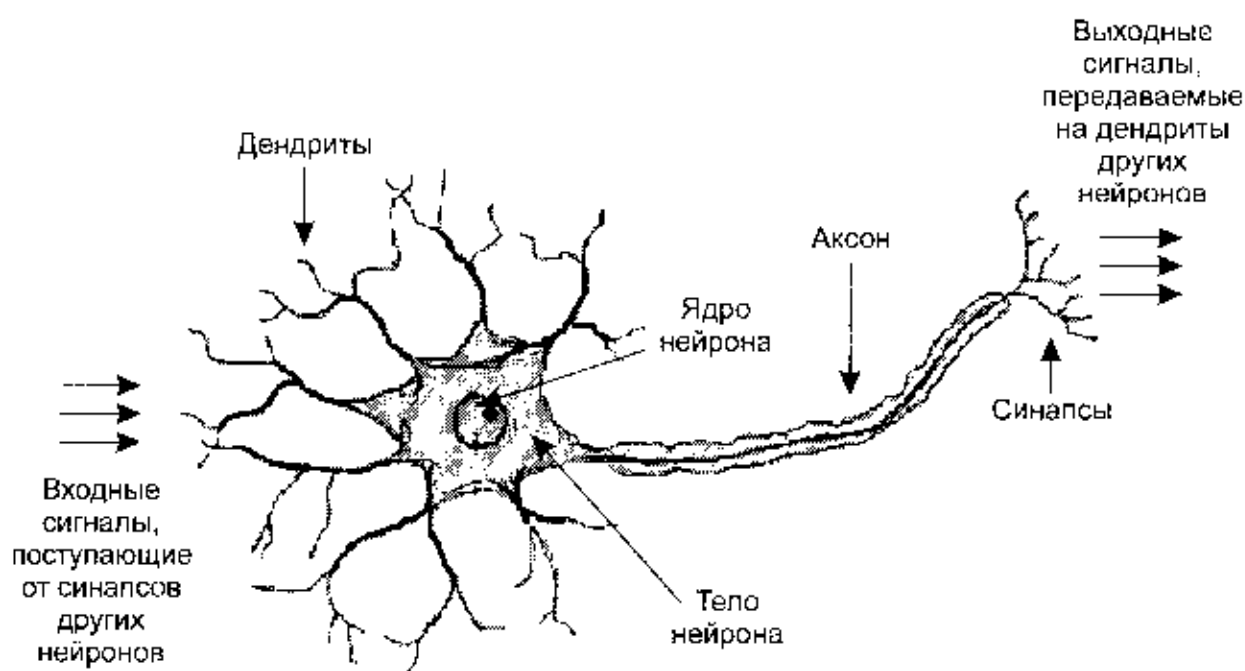


Рис. 3.7. Биологический нейрон

Абстрагируя принцип работы биологического нейрона, математики построили математическую модель, отражающую основные принципы работы этой клетки (рис. 3.8).

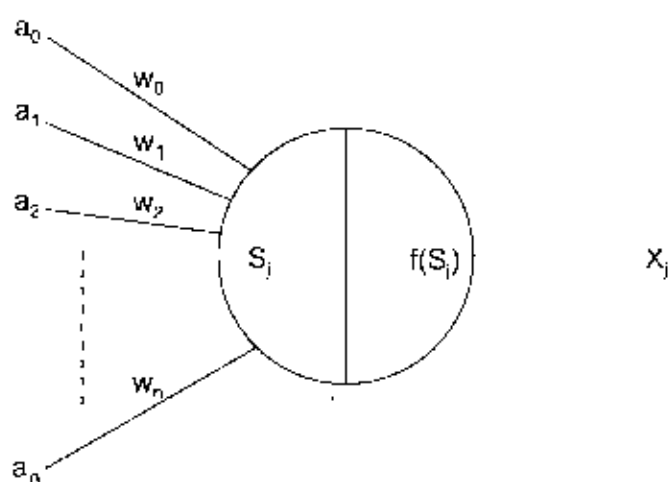


Рис. 3.8. Классическая модель искусственного нейрона

В этой модели сигналы, поступающие на входы $a_0...a_n$, умножаются на весовые коэффициенты $w_0...w_n$ и на основе функции обработки $f(S_j)$ вырабатывается выходной сигнал X_j . В простейшем случае входные сигналы, помноженные на весовые коэффициенты, суммируются, и, когда значение суммы становится больше определенного порога, генерируется выходной сигнал. В других случаях зависимость выходного сигнала от суммы входных может иметь более сложный характер, описываемый заданной математической функцией.

Физически один искусственный нейрон представляет собой простой микропроцессор с возможностью программирования функции $f(S_j)$, установки весовых коэффициентов и с небольшим объемом встроенной памяти. Для обработки знаний отдельный искусственный нейрон применен быть не может. Но когда множество искусственных нейронов соединяются между собой в искусственную нейронную сеть (подобно тому, как биологические нейроны соединены в мозгу человека), у них появляется возможность обрабатывать и накапливать знания. Знания в искусственной нейронной сети накапливаются в виде значений весовых коэффициентов.

Таким образом, можно сформулировать определение.

Искусственная нейронная сеть — это математическая модель вместе с ее аппаратной и программной реализацией, построенная по принципу действия биологической нейронной сети и предназначенная для установления взаимосвязности данных.

Исходя из этого определения, *необходимыми элементами* искусственной нейронной сети являются:

- математическая модель;
- искусственные нейроны (реализованные аппаратно или программно);
- программно-топологическая реализация математической модели за счет задания соединений между нейронами и обрабатывающей (активационной) функции $f(S_j)$.

Искусственная нейронная сеть в общем случае характеризуется следующими параметрами:

- *Адаптивная обучаемость.* В контексте искусственной нейронной сети обучаемость означает, что сеть может усваивать различные варианты поведения в зависимости от того, какие данные поступают на ее вход. Вместо того чтобы диктовать сети, как она должна реагировать на каждую порцию данных (как это было бы в случае обычного программирования), сеть сама находит сходства и различия в поступающих данных. По мере поступления новых данных обучение продолжается, и поведение сети изменяется.
- *Самоорганизация.* По мере того как данные поступают на вход сети, сеть имеет возможность изменять весовые коэффициенты тех или иных соединений. Тем самым, по мере поступления новых данных, практически изменяется структура сети. Эффективность самоорганизации сети зависит от начальной структуры соединений и выбранного в качестве математической модели алгоритма обучения (треппинга).
- *Устойчивость к ошибкам.* Искусственная нейронная сеть умеет выделять из потока данных важные свойства и усиливать их, при этом слабо реагируя на случайные, искаженные или совершенно новые данные (свойства). Таким образом данные, не несущие в себе повторяющихся закономерностей (другими словами, помехи), просто отбрасываются нейронной сетью, обеспечивая тем самым устойчивость к ошибкам.
- *Работа в режиме реального времени и параллельная обработка информации.* Эти преимущества нейронных сетей проявляются только в промышленном исполнении (когда каждый нейрон действительно представляет собой отдельный процессор) и не могут быть получены при программной эмуляции нейронов.

Основными признаками для классификации искусственных нейронных сетей являются архитектура и связанный с ней алгоритм обучения. По этим признакам искусственные нейронные сети можно классифицировать так, как это показано на рис. 3.9.

В этой классификации можно отметить несколько моментов.

- *Обучение с учителем* — класс искусственных нейронных сетей, для которого заранее известен диапазон выходных данных. В случае, когда после преобразования входных данных и получения выходных данных последние не укладываются в заранее заданные диапазон значений, нейронная сеть получает сигналы обратной связи и корректирует свою структуру с целью уменьшения ошибки.

Пример. При распознавании изображений оператор (человек или электронное устройство) сообщает нейронной сети, правильно или нет она распознала изображение.

- *Обучение без учителя* — класс искусственных нейронных сетей, для которого диапазон выходных значений не задан и обучение сети проводится только на основании закономерностей, обнаруженных во входных сигналах.



Рис. 3.9. Классификация нейронных сетей

- ❑ *Многослойный перцептрон* – искусственная нейронная сеть, состоящая из
 - входного слоя нейронов, на которые поступают входные сигналы;
 - выходного слоя нейронов, передающего выходные сигналы на интерфейс пользователя;
 - скрытых слоев нейронов, расположенных между входным и выходным слоями;
 - механизма (или алгоритма) обратного распространения, обеспечивающего при наступлении ошибки (несовпадении выходного сигнала с шаблоном) последовательную корректировку всех весовых коэффициентов связей, начиная с ближнего к выходному скрытого слоя и заканчивая входным слоем, с целью устранения этой ошибки.
- ❑ *Сеть с радиально-базисной функцией* – искусственная нейронная сеть, имеющая, кроме входного и выходного, один скрытый слой и использующая в качестве активационной функции нейронов скрытого слоя радиально-базисную функцию (РБФ), которая в общем виде выглядит так:

$$f(x) = \varphi\left(\frac{x^2}{\sigma^2}\right).$$

- ❑ *Байесовская сеть* – искусственная нейронная сеть, использующая в качестве математической модели сеть Байеса, связывающую между собой множество переменных (весовых коэффициентов) и их вероятностных зависимостей (активационных функций).

- *Самоорганизующиеся карты Кохонена* – искусственные нейронные сети, осуществляющие последовательное (по мере обучения) группирование сходных данных в плоской системе координат таким образом, что к завершению обучения узлы, содержащие сходные данные, располагаются геометрически в непосредственной близости друг от друга. Данный подход позволяет эффективно выделить главные данные, подавив случайные шумы и ошибки.
- *Сети на основе адаптивной теории резонанса (ART-сети)* были созданы специально для решения задач классификации. В них заложено сразу два дихотомически противоположных принципа: во-первых, сеть должна самомодифицироваться в ответ на каждый входной сигнал; во-вторых, сеть должна сохранять знания, а значит, быть стабильной. Решение заключается в нахождении точки равновесия между требованиями пластичности и стабильности сети.

К достоинствам искусственных нейронных сетей можно отнести:

- возможность решения задач, которые не решаются никакими другими методами;
- самообучение;
- получение результатов в режиме реального времени;
- «креативность»;
- создание новых знаний внутри сети в процессе переработки входных данных.

Недостатки искусственных нейронных сетей:

- сравнительная дороговизна аппаратной реализации;
- трудность тиражирования накопленных знаний;
- для больших сетей невозможность заранее даже приблизительно оценить время обучения сети.

Искусственные нейронные сети хорошо подходят для решения задач:

- с большими массивами входных данных;
- с неизвестным алгоритмом, но большим количеством конкретных примеров решения;
- с большим количеством шумов;
- с недостаточностью или, наоборот, избыточностью данных.

Искусственные нейронные сети могут эффективно решать задачи распознавания изображений, классификации, оптимизации или прогнозирования. Более конкретно такими задачами являются:

- распознавание лиц, голосов, отпечатков пальцев;
- анализ рентгенограмм;
- обнаружение отклонений в ЭКГ;
- обработка звуковых сигналов (разделение, идентификация, локализация, устранение шума, интерпретация);
- обработка радарных сигналов (распознавание целей, идентификация и локализация источников);

- проверка достоверности подписей;
- оценка риска займов;
- прогнозирование изменений экономических показателей и т. п.

3.4.3. Системы, основанные на прецедентах

Когда речь идет о системах, основанных на прецедентах, имеется в виду накопленный ранее опыт решения проблем, сходных с возникшей. В базе знаний системы хранятся не правила, а прецеденты. *Прецедент* состоит из двух важных составляющих:

- хорошо сформулированного описания проблемы (или набора проблем);
- описания решения этой проблемы.

Система, основанная на прецедентах, — это система с базой знаний, в которой в качестве знаний выступают прецеденты.

Таким образом, система, основанная на прецедентах, наиболее полно соответствует понятию *знаний организации*. Она накапливает знания о методах, способах и путях решения часто встречающихся проблем и решения стандартных задач.

Обобщенная схема работы системы, основанной на прецедентах, показана на рис. 3.10.

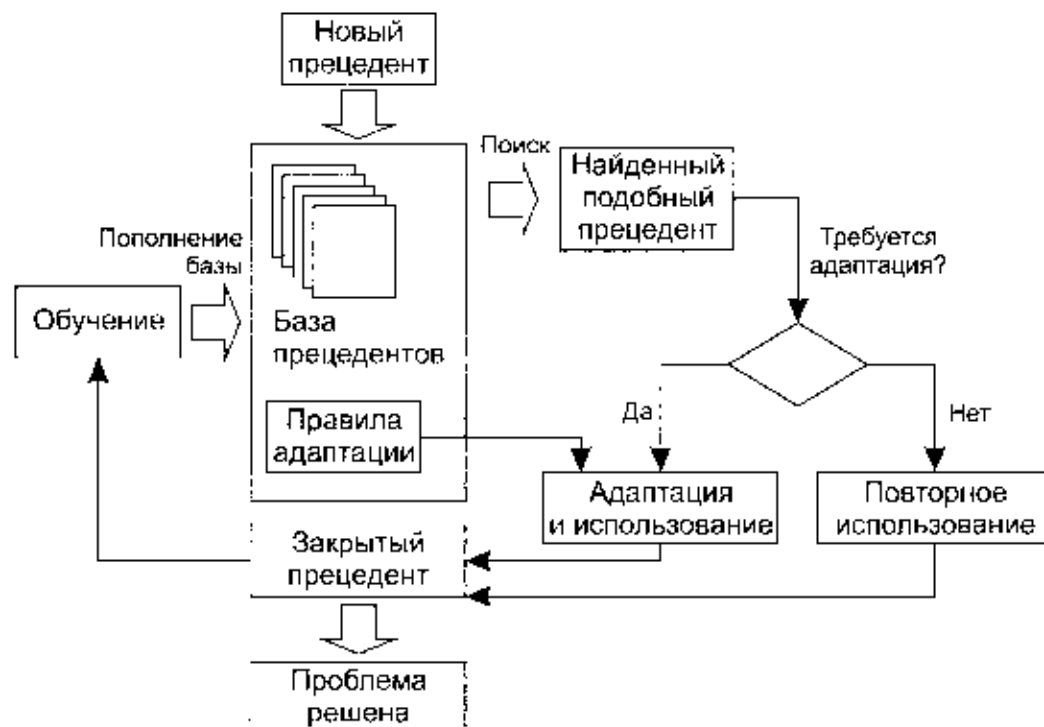


Рис. 3.10. Схема работы системы, основанной на прецедентах

Алгоритм работы предложенной схемы следующий:

1. На первом этапе возникает проблемная ситуация (техническая или производственная задача, требующая решения).
2. Заполняется диагностическая форма, то есть происходит описание ситуации по формальным признакам.

3. По набору формальных признаков производится поиск схожей или идентичной ситуации в базе знаний:

- если идентичная ситуация найдена, то алгоритм ее решения предлагается в качестве решения текущей проблемы;
- если найдена ситуация, отклоняющаяся по формальным признакам от идентичной ниже порогового уровня (который вычисляется как сумма формальных признаков, помноженная на весовые коэффициенты признаков), то исследуется возможность адаптации прецедента из базы знаний к текущему прецеденту с использованием правил базы знаний;
- если схожая ситуация не найдена или адаптация невозможна, к решению проблемы подключается эксперт в предметной области, после чего решение проблемы формализуется и новый прецедент вносится в базу знаний;
- если в процессе поиска находится несколько соответствий описанию входной ситуации, необходимо уточнить набор входных параметров, чтобы выявить наиболее подходящий прецедент.

В качестве достоинств систем, основанных на прецедентах, можно отметить:

- сокращение трудоемкости приобретения новых знаний, поскольку описывать прецеденты проще, чем программировать правила;
- обучение системы происходит на основе добавления новых прецедентов, а не изменения или добавления новых правил.

Недостатки систем, основанных на прецедентах:

- необходимость строго отслеживать правильность оформления прецедентов, в противном случае может произойти «засорение» базы знаний;
- необходимость разработки методов поиска, однозначно идентифицирующих подходящие прецеденты.

Наиболее часто системы, основанные на прецедентах, используются в тех областях, для которых характерны повторяющиеся проблемные ситуации:

- службы технической поддержки, сервисные подразделения;
- диагностика в медицинских учреждениях;
- организация юридической помощи (особенно в странах, где действует прецедентное законодательство);
- контрольные органы;
- системы обеспечения качества и т. д.

В общем случае рекомендуется применять системы, основанные на прецедентах, когда

- трудно произвести детальную декомпозицию ситуации с целью выработать правила ее решения, то есть тогда, когда не может быть применена экспертная система, основанная на правилах;
- трудно понять и описать суть возникающих проблем, но легко накопить примеры их разрешения.

3.4.4. Системы, построенные на генетических алгоритмах

Под алгоритмом обычно понимают более или менее сложную последовательность действий, которые необходимо выполнить для решения текущей проблемы или задачи. При этом подразумевается, что данная последовательность является статической, то есть формируется до начала решения задачи и остается неизменной в ходе решения. Статичность алгоритма накладывает ограничения на поиски путей решения.

Генетический алгоритм — это одна из моделей машинного самообучения, основанная на абстрагировании моделей биологической эволюции.

Так же как и в биологической эволюции, для применения генетического алгоритма внутри компьютера программным путем создаются *популяции особей*, представленные *хромосомами*.

При реализации генетического алгоритма в процессе используются следующие составляющие:

- генетическое представление потенциального решения проблемы;
- способ создания начальной популяции потенциального решения;
- эволюционная функция, которая играет роль среды обитания, сортируя решения согласно их адекватности;
- генетические операторы, которые изменяют сочетания генов потомства;
- значения различных параметров, которые использует генетический алгоритм.

Эволюция решений при использовании генетического алгоритма базируется на трех основных процессах:

- *Воспроизведение*. На базе родительских шаблонов создается новое поколение кода, которое имеет более высокий адаптационный рейтинг, то есть в большей мере соответствует решению проблемы.
- *Скрещивание*. Строки кода в случайном порядке разбиваются пополам и создаются новые строки путем обмена половинками старых строк. Этот процесс имитирует биологический процесс, происходящий при создании хромосом потомства из хромосом родителей.
- *Мутация*. В случайном порядке изменяется один символ (одна цифра) в исходной строке без воспроизведения или перекрещивания.

Процесс применения генетического алгоритма является циклическим и выполняется до тех пор, пока не найдется приемлемое решение проблемы:

1. Вычисляется уровень адаптации всех членов популяции.
2. Над каждым членом популяции выполняются операции воспроизведения, генерации и мутации.
3. Старая популяция уничтожается.

Генетические алгоритмы применяют для решения сложных задач с большим количеством данных и параметров, которым должно соответствовать решение:

- составление расписаний;
- решение логистических задач;
- создание сложных многослойных топологий (например, проектирование топологии микросхем) и т. д.

3.4.5. Интеллектуальные агенты

Существует множество рутинных процессов, не поддающихся строгой алгоритмизации. При достаточно точной формулировке конечной цели или задачи процесс ее решения может потребовать от исполнителя определенной гибкости и адаптивности. Речь идет о таких процессах, как длительный поиск в больших массивах информации, расположенных в динамически изменяющейся системе (например, в Интернете), уборка квартиры или очистка участка улицы от снега, то есть работа, которую приходится выполнять человеку, но которую желательно порекомендовать автомату.

Для выполнения такого рода работ были созданы интеллектуальные агенты.

Интеллектуальные агенты — это некоторые сущности, такие как компьютерные программы, или роботы, которые выполняют поставленные задачи от лица человека и обладают определенными уникальными характеристиками, позволяющими им работать в динамически изменяющемся окружении.

Для того чтобы классифицироваться как интеллектуальный агент, сущность должна обладать следующей функциональностью:

- воспринимать свое окружение;
- понимать конечную цель или назначение;
- взаимодействовать с окружением;
- реагировать на изменения в окружении;
- проявлять признаки интеллекта, то есть уметь принимать решение и, возможно, обучаться на основе приобретенного опыта;
- принимать автономные решения, то есть, обладая знанием конечной цели или предназначения, самостоятельно решать, какие действия нужно выполнить для достижения цели.

Общая архитектура интеллектуального агента представлена на рис. 3.11.

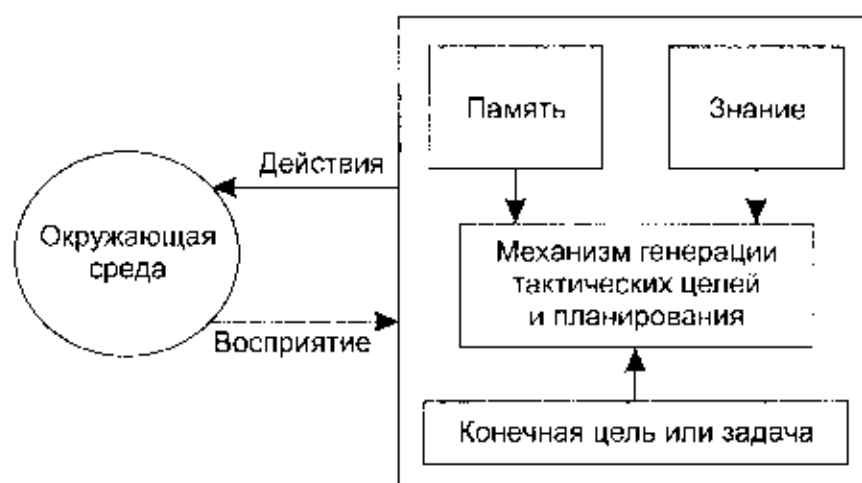


Рис. 3.11. Общая архитектура интеллектуального агента

Интеллектуальные агенты все чаще применяются в самых разных областях деятельности: в поисковых и промышленных роботах, в беспилотных летательных аппаратах, в исследовательских роботах, работающих на дне океана, и т. д.

3.4.6. Системы добычи данных

Базы данных в их классическом понимании, а также большие хранилища данных вне баз данных (например, лог-файлы крупных веб-серверов) могут быть источником информации, которая в явном виде в них не отображена. Например, анализируя лог-файлы популярного образовательного интернет-портала, можно сделать вывод об успешности информатизации школ в том или ином регионе страны.

Под добычей данных (выявлением знаний) понимают автоматическое извлечение скрытых (присутствующих в неявной форме) данных из наборов данных. Добычей данных занимаются специализированные системы. Скрытые данные, добываемые при помощи инструментов добычи данных, могут служить основой для создания баз знаний экспертных систем различного типа. Сами инструменты добычи данных, в свою очередь, зачастую являются инструментами, основанными на знаниях. В зависимости от выбранной модели добычи данных, в качестве инструментов анализа и выявления скрытых данных (знаний) могут быть использованы:

- искусственные нейронные сети;
- деревья решений;
- генетические алгоритмы;
- индукционные алгоритмы.

Источниками данных для выявления знаний могут служить любые массивы накопленных данных, например:

- медицинские и анкетные данные от государственной переписи населения до базы данных персонала и клиентов крупной фирмы;
- сохраненные данные видео- и фотонаблюдения;
- данные спутниковой съемки;
- данные на цифровых посетителях;
- текстовые отчеты и записки;
- данные Всемирной паутины (несмотря на свою неструктурированную и гетерогенную природу, Интернет сосредотачивает в себе самый большой объем данных, который когда-либо существовал).

Для выявления знаний используют два основных типа моделей:

- *описательная модель* выявляет и описывает общие свойства и закономерности в существующих данных;
- *прогнозирующая модель* пытается предсказать тенденции в той или иной области, основываясь на добытых знаниях.

Функциональность систем добычи данных может включать в себя следующие компоненты:

- *Информационные характеристики* объединяют в себе общие особенности объектов какого-либо класса.
- *Дифференцирование данных*. Продуктом дифференцирования данных является набор правил, описывающих разницу основных особенностей объектов целевого

и контрастного классов. Например, чтобы лучше организовывать рекламные компании, полезно увидеть, чем отличаются клиенты, арендующие большие площади, от клиентов, арендующих малые площади.

- *Анализ связности данных* служит для извлечения сведений о данных, которые часто встречаются вместе.
- *Классификационный анализ* обеспечивает разделение данных на классы по некоторым заранее известным характеристикам. Для этого обычно применяется нейронная сеть с учителем, когда в качестве выходного шаблона задаются требуемые характеристики класса.
- *Предсказание данных*. На базе шаблонов, уже имеющихся в наборе данных, можно предсказать вероятность появления данных. Это может пригодиться для предсказания поведения потребителей на основе их прошлого поведения.
- *Кластеризация* обеспечивает разделение данных на классы с заранее неизвестными параметрами, характеристики которых извлекаются из самих данных. При кластеризации основания для классификации ищутся в самих данных. Для кластеризации применяются нейронные сети без учителя, такие как самоорганизующиеся карты Кохонена.
- *Анализ выбросов* позволяет отбирать данные, которые не подпадают ни под один шаблон или класс. Иногда эти данные являются шумом, но иногда могут представлять собой важные знания.
- *Эволюционный анализ* служит для моделирования изменения наборов данных во времени.
- *Анализ отклонений* позволяет сравнить эталонные и реальные значения данных и выявить причины, по которым данные отклоняются от эталона.

Системы добычи данных можно классифицировать по

- типу источника данных (распределенные данные, временные ряды, данные мультимедиа, текстовые источники);
- используемой модели данных (реляционные данные, объектно-ориентированные базы данных, хранилища данных);
- способу добычи данных (дифференциация, классификация, кластеризация, и т. д.);
- технологии реализации (нейронные сети, генетические алгоритмы, статистические методы, и т. д.).

3.5. Инженерия знаний

Хотя термины «*инженерия знаний*» и «*управление знаниями*» часто используют как взаимозаменяемые, на самом деле необходимо провести демаркационную линию между двумя этими понятиями. Управление знаниями в большей мере связано с функцией руководства, администрирования и контроля. Менеджер знаний чаще

всего является управляющим информационного подразделения в организации, отвечая за определение потребности организации в знаниях. Основной вопрос, на который отвечает менеджмент знаний: «зачем?».

В *менеджмент знаний* входят следующие функции:

- определение потребности организации в знаниях;
- определение роли знаний в структуре управления работой организации;
- определение типа и объема знаний, необходимых различным подразделениям организации;
- выявление приоритетов важности знаний и времени, когда они необходимы.

В отличие от менеджмента, инженерия знаний подразумевает ответ на вопрос «как?».

Инженер по знаниям обеспечивает технологию решения задач и обеспечения потребностей, которые формулирует менеджер по знаниям.

Инженерия знаний выполняет следующие функции:

- обеспечивает технологический процесс сбора и хранения знаний;
- разрабатывает методологии и технологии извлечения знаний из хранилищ;
- отвечает за форму представления знаний конечному пользователю;
- поддерживает сбор, хранение, модификацию и целостность знаний.

Таким образом, если менеджер по знаниям может быть своего рода информационным управляющим, не имеющим прямого отношения к компьютерным технологиям, то инженер по знаниям обязательно должен быть компьютерным специалистом, способным решать сложные задачи разработки и сопровождения информационных систем и информационных технологий самого широкого спектра. Одновременно инженер по знаниям должен обладать глубокими знаниями психологии общения и развитыми социальными навыками, позволяющими обеспечить получение знаний от экспертов.

3.5.1. Получение знаний

Получение (извлечение, добыча, выявление, приобретение) знаний является главным процессом в инженерии знаний. Без этого процесса все остальные технологии становятся малоприменимыми.

Получение знания — это процесс передачи знания от эксперта или группы экспертов инженеру по знаниям для создания интеллектуальных систем.

Поскольку экспертная система или любая другая система, основанная на знаниях, должна содержать в себе в первую очередь знания экспертов-людей той или иной предметной области, процесс получения знаний в первую очередь предполагает вовлечение экспертов в диалог с инженером по знаниям. Хотя эксперты обычно с удовольствием говорят на темы, непосредственно затрагивающие близкую им предметную область, это не значит, что процесс получения знаний является легким.

Кроме получения знаний от экспертов, знания можно получать из других источников: технической литературы или литературы по предметной области, анкетных опросов, документации аналогичных систем, основанных на знаниях. Однако стержнем системы, основанной на знаниях, являются знания экспертов, связанных с конкретной задачей в конкретной организации. Для получения знаний от экспертов широко используются различные способы собеседования (интервью).

Выделяют четыре главных типа интервью:

- *Неструктурированное интервью.* Такого рода интервью представляет собой собеседование без четкого плана. В этом случае инженер получает базовые представления о предметной области. Задача инженера в процессе проведения интервью — мягко, но настойчиво возвращать эксперта к обсуждению предметной области.
- *Структурированное интервью.* Этот тип интервью имеет четкий план и структуру и используется для получения подробных сведений о предметной области. Поскольку структурированные интервью обычно проводятся после нескольких неструктурированных, инженер по знаниям обычно уже ориентируется в предметной области. К этому интервью у него подготовлен список вопросов, на которые он хочет получить ответы. В основном интервью сосредотачивается на том, *почему* были предприняты те или иные действия, для того чтобы понять, *как* эксперт принимает решение. Структурированное интервью проходит в три этапа:
 - 1) инженер по знаниям излагает предстоящие темы и согласовывает их с экспертом. Задача этого этапа — мотивировать эксперта;
 - 2) инженер по знаниям задает эксперту вопросы, эксперт отвечает;
 - 3) инженер по знаниям еще раз проверяет, правильно ли он понял ответы, все ли вопросы были заданы, на все ли вопросы были получены правильные ответы. После чего он излагает свои выводы эксперту и добивается его одобрения.
- *Случай из прошлого.* В этом интервью инженер по знаниям просит эксперта вспомнить конкретный случай из прошлой практики, когда эксперт применял свои навыки в предметной области. Берется локальная ситуация, возможно, небольшая по масштабам, и разбирается очень подробно. Этот тип интервью направлен в первую очередь на то, чтобы понять, как именно происходит мыслительный процесс у эксперта, на основании каких неявных мыслей принимается решение, и требует особых психологических навыков от инженера по знаниям. Цель интервью — постараться перевести в эксплицитные как можно больше тацитных (неявных) знаний эксперта.
- *Мысли вслух.* В этом типе интервью так же, как и в случае из прошлого, разбирается конкретная ситуация (из реальной жизни или смоделированная), но эксперта просят все рассуждения делать явным образом, вслух. Цель интервью такая же, как и в предыдущем — выявление неявных знаний.

Кроме четырех базовых типов интервью в работе инженера по знаниям могут применяться и другие типы бесед с экспертами.

- *Учебное интервью.* Эксперта просят подготовить презентацию для инженера по знаниям, чтобы ознакомить его со специфической областью.

- *Двадцать вопросов.* Инженер по знаниям подготавливает опросный лист из вопросов, на которые надо ответить только «да» или «нет».
- *Пусковое интервью.* Инженер по знаниям готовит материал, который может резко повысить активность эксперта. Такая подготовка делается на основе предыдущих бесед (например, отмечается, что эксперт был очень эмоционален в обсуждении какого-то вопроса). Этот материал включается в интервью и все внимание сосредоточивается на нем.
- *Интервью обратного обучения.* Инженер по знаниям делает попытку обучить эксперта в обсуждаемой предметной области. Эксперт обеспечивает обратную связь.
- *Репертуарная решетка.* Объекты в предметной области помещаются в заголовки столбцов, характеристики объектов – в названия строк, затем на пересечении строк и столбцов ставятся крестики (методология работы с репертуарной решеткой достаточно объемна и в данной книге не приводится).

Есть несколько советов, которые могут помочь в ходе проведения интервью.

Во-первых, избегайте двусмысленности. Всякого рода сравнения типа «более быстрый», «лучше», «больше» требуют уточнения: больше чего? лучше по сравнению с чем?

Во-вторых, надо принимать во внимание, что эксперт в своих рассуждениях часто опускает ключевые части процесса. Для того чтобы инженер по знаниям понял его рассуждения, эксперт часто опускает ключевые, но сложные для понимания технические детали.

В-третьих, то, что эксперт считает интуицией, часто является частью прошлого опыта, который может быть переведен в явную форму. Однако эксперт может настаивать на том, что он просто «почувствовал», что именно такое решение будет правильным.

Вопросы, которые полезно задавать в процессе интервью:

1. Можете ли вы дать мне краткий обзор выбранной темы?
2. Можете описать последний случай, с которым вы столкнулись на практике?
3. Какие факты или гипотезы приходят вам в голову, когда вы думаете о проблеме?
4. Какие вещи вы предпочитаете узнать перед тем, как начать размышлять о проблеме?
5. Расскажите, как именно вы к этому пришли?
6. И что вы делаете затем?
7. Как это связано с ...?
8. Как (почему, когда, зачем) вы делаете это?
9. Вы можете описать другими словами, что вы подразумеваете под ...?

В завершение интервью крайне важно сделать краткий обзор полученной информации и оповестить эксперта о возможности или необходимости дальнейших интервью.

3.5.2. Жизненный цикл и методология

При проектировании и разработке систем, основанных на знаниях, стандартную методологию разработки информационных систем трудно применить в полном объеме. Специфика систем, основанных на знаниях, требует применения особых подходов и инструментов.

В стандартной методологии жизненный цикл разработки информационных систем обычно составляет 6 этапов, объединенных в процессе разработки моделями водопада или спирали:

1. Анализ (сбор требований).
2. Дизайн (моделирование).
3. Реализация (программирование).
4. Тестирование (испытание).
5. Развергивание (установка).
6. Поддержка (обслуживание).

При попытке применить эту методологию к созданию систем, основанных на знаниях, инженер по знаниям может столкнуться со следующими трудностями:

- *Проблемы спецификации требований.* На этапе сбора требований еще не существует точной информации о том, какие знания будут содержаться в системе и как именно они будут обрабатываться. Поскольку модель водопада полностью построена на подробной спецификации требований при анализе, ее реализация оказывается невозможной.
- *Проблемы макетирования.* Циклический процесс разработки, который начинается с создания макета и заканчивается разработкой полноценной системы, казалось бы, больше соответствует системам, основанным на знаниях. Однако этот процесс также затруднен: требования к выходным знаниям, к процессу принятия решений, распыленные на начальных этапах, должны конкретизироваться от цикла к циклу. Но по мере выявления явных знаний у экспертов эти требования обычно претерпевают разительные изменения. По этой причине зачастую каждый цикл представляет собой не доработку существующего макета, а создание нового, при этом количество циклов и сроки выполнения работ невозможно предсказать даже приблизительно.

Такого рода затруднения требуют применения новых подходов и новых методологий проектирования при создании систем, основанных на знаниях.

Архитектура общего плана подразумевает разделение одной большой системы, основанной на знаниях, на ряд связанных подсистем и спецификацию однотипного интерфейса между ними. Таким образом, сложная задача создания большой системы разбивается на ряд подзадач, объем работ и время выполнения которых можно с достаточной степенью достоверности оценить.

Название архитектуры произошло от способа проектирования, когда несколько человек усаживаются за один планшет (или доску) и совместно рисуют на ней. При этом каждый хорошо видит часть работы, которую выполняют другие.

Архитектура планшета подходит не для всех проектов. Для того чтобы проект можно было разрабатывать, используя эту архитектуру, он должен допускать возможность декомпозиции на ряд связанных, но в достаточной мере изолированных подзадач.

Пример. Создание экспертной системы строительства дома на участке.

Такого рода экспертная система легко разбивается на ряд подсистем: организация садового участка, кухни, гаража, спальни, комнат общего пользования и интегрирующей системы знаний.

Все включенные экспертные системы могут разрабатываться независимо друг от друга, иметь независимые базы знаний и модули представления знаний при наличии однотипного интерфейса с интегрирующей системой.

Кроме методологии общего планшета при разработке систем, основанных на знаниях, применяются следующие методологии:

- *Метод решения проблем* основывается на накоплении и создании библиотеки последовательности действий, необходимых для решения проблемы, и набора знаний, позволяющего эти действия совершить. Достоинство метода — возможность многократного использования. Недостаток — необходимость создания библиотеки знаний и методов.
- *Методология моделирования получения знаний* сочетает в себе черты методологии архитектуры планшета (деление проблемы на составные части) и метода решения проблем (накопления методов и знаний).

Вопросы для самопроверки

1. Как вы понимаете термин «управление знаниями»?
2. Какие процессы включаются в управление знаниями?
3. Из каких этапов состоит жизненный цикл управления знаниями?
4. Чем отличаются друг от друга понятия «данные», «информация» и «знания»?
5. Как развивались информационные системы?
6. Какие способы классификации знаний вы знаете?
7. Какие модели представления знаний вам известны?
8. Приведите пример продукционной модели представления знаний.
9. Дайте определение семантической сети.
10. Что такое «фрейм»? Что такое «слот»?
11. Перечислите признаки интеллектуального поведения.
12. Какие типы систем, основанных на знаниях, вам известны?
13. Дайте определение экспертной системы.
14. Из каких основных элементов состоит экспертная система?
15. Каковы достоинства и недостатки экспертных систем?

16. В чем суть методологии построения баз данных?
17. Что такое «искусственная нейронная сеть»?
18. Перечислите характеристики нейронной сети?
19. Какие типы нейронных сетей вам известны?
20. Каков алгоритм работы системы, основанной на прецедентах?
21. В каком виде знание представлено в искусственной нейронной сети? В экспертной системе? В системе, основанной на прецедентах?
22. Что такое «генетический алгоритм» и для чего он применяется?
23. Что такое интеллектуальный агент?
24. Что может быть источниками данных в процессе добычи данных?
25. На основе каких функций производится добыча данных?
26. В чем различие инженерии знаний и управления знаниями?
27. Что такое «получение знаний»? Какую роль этот процесс играет в инженерии знаний?
28. Перечислите четыре основных типа интервью.
29. Почему стандартная методология разработки информационных систем часто не подходит для инженерии знаний?
30. Какие специфические методологии применяются в инженерии знаний?

Литература

1. Башмаков А. И., Башмаков И. А. Интеллектуальные информационные технологии. М.: Издательство МГТУ им. Баумана, 2005.
2. Гаврилова Т. А., Хорошевский В. Ф. Базы знаний интеллектуальных систем. СПб.: Питер, 2000.
3. Джанетто Карен, Уилер Эми. Управление знаниями. М.: Хорошая книга. 2005.
4. Емельянова Н. З., Партыка Т. Л., Попов И. И. Основы построения автоматизированных информационных систем. М.: ФОРУМ-ИФРА-М, 2007.

Глава 4 Логические основы информатики

- 4.1. Представление о высказываниях и логических операциях
- 4.2. Алгебра логики
- 4.3. Построение коммутационных схем на основе алгебры логики

Алгебра логики (алгебра высказываний) и основы математической логики играют важную роль в информатике. Математическая логика присутствует в различных разделах информатики: в виде двоичной логики, на которой основана работа цифровых компьютеров; в виде специальной алгебры логики, лежащей в основе математической модели реляционных баз данных; в виде правил, определяющих функционирование алгоритмов и программ, работу интеллектуальных и экспертных систем. Таким образом, логика для информатики является фундаментальным предметом, изучение которого даст ключ к множеству других разделов.

4.1. Представление о высказываниях и логических операциях

4.1.1. Понятие высказывания

Основным понятием математической логики является понятие простого высказывания.

Высказывание — это повествовательное предложение, утверждающее что-либо о чем-либо, о котором можно сказать, истинно оно или ложно в данных условиях. Логическими значениями высказываний являются «истина» и «ложь».

В алгебре логики все высказывания рассматриваются только с точки зрения их логического значения. Каждое высказывание либо истинно, либо ложно, и ни одно высказывание не может быть одновременно истинным и ложным.

Пример. Волга впадает в Каспийское море. Значение высказывания — «истина».

Лондон — столица Франции. Значение высказывания — «ложно».

Карась не рыба. Значение высказывания — «ложно».

Число 6 делится на 2 и на 3. Значение высказывания — «истина».

Если юноша окончил среднюю школу, то он получает аттестат зрелости. Значение высказывания — «истина».

Предложения «Вперед, гардемарины!» или «Какова сейчас температура воздуха за окном?» не являются высказываниями, поскольку не несут в себе однозначных сведений об истинности или ложности. Таким образом, высказыванием обычно являются повествовательные (но не вопросительные и не восклицательные) предложения.

Высказывание, представляющее собой одно утверждение, принято называть *простым*, или *элементарным*. Примерами элементарных высказываний являются первое и второе высказывание в приведенном примере.

Высказывания, которые получаются из элементарных с помощью грамматических связок «не», «и», «или», «если», «то», «тогда и только тогда», принято называть *сложными*, или *составными*. В приведенном примере третье высказывание получается из простого высказывания «Карась — рыба» путем добавления отрицания «не»; четвертое высказывание образовано из элементарных высказываний «Число 6 делится на 2», «Число 6 делится на 3», соединенных союзом «и». Пятое высказывание получается из простых высказываний «Юноша окончил среднюю школу» и «Юноша получает аттестат зрелости» путем добавления грамматической связки «если ..., то ...». Аналогично, сложные высказывания могут быть получены из простых высказываний путем добавления грамматических связок «или», «тогда и только тогда».

В дальнейшем элементарные высказывания мы будем обозначать малыми буквами латинского алфавита; истинное значение высказывания цифрой 1, а ложное значение — цифрой 0. Например, если высказывание a истинно, то будет справедлива запись $a = 1$, если высказывание a ложно, то $a = 0$.

Всякая точная наука, в данном случае математическая логика, абстрагируется от многих побочных явлений в изучаемых ею объектах и рассматривает в некоторой мере идеализированную картину. Аналогично и в других науках, например, геометрия рассматривает точки, лишенные геометрических размеров, и линии, лишенные толщины.

При изучении логики высказываний предполагается, что все простые высказывания, входящие в рассмотрение, обладают одним из двух свойств — являются истинными либо ложными. Математические утверждения обладают этим свойством, и так как до сих пор математическая логика изучала в первую очередь логику математических доказательств, то такая абстракция особенно оправдана.

4.1.2. Соглашения о языке алгебры высказываний

Используются различные обозначения (нотации) как для самих высказываний, так и для операций алгебры высказываний (алгебры логики). Возможные варианты сведены в табл. 4.1.

Таблица 4.1. Обозначения в алгебре высказываний

Понятие	Возможные обозначения	Обозначения, принятые в данной книге
Высказывание	Строчные и прописные буквы латинского алфавита: $a, b, c, \dots, z, A, B, C, \dots, Z$; прописные буквы русского алфавита: $A, B, B, \dots, Я$	Строчные буквы латинского алфавита: $a, b, c, \dots, z$
Истинность	Прописная или строчная русская буква И (и); слово истина; прописная или строчная латинская буква Т (t); слово true; цифра 1	Цифра 1
Ложность	Прописная или строчная русская буква Л (л); слово ложь; прописная или строчная латинская буква F (f); слово false; цифра 0	Цифра 0
Отрицание, опровержение, инверсия	Символ $\neg$ или $\bar{}$	Символ подчеркивания $\bar{}$
Конъюнкция, логическое «и»	Символ $\&$, $\wedge$ или $\bullet$. Кроме того, иногда знак между двумя высказываниями просто опускают	Символ $\wedge$
Дизъюнкция, логическое «или»	Символ $\vee$	Символ $\vee$
Импликация	Символ $\supset$ или $\rightarrow$	Символ $\supset$
Эквивалентность	Символ $\sim$, $\leftrightarrow$ или $\leftrightarrow$	Символ $\leftrightarrow$
Равносильность	Символ $\equiv$	Символ $\equiv$

4.1.3. Логические операции над высказываниями

Над высказываниями можно выполнять следующие логические операции: отрицание, конъюнкция, дизъюнкция, импликация и эквивалентность.

Операция *отрицания* высказывания x обозначается $\bar{x}$ и читается «не x » или «неверно, что x ».

Отрицание высказывания x — это новое высказывание, которое является истинным, если высказывание x ложно, и ложным, если высказывание x истинно.

Логическое значение высказывания $\bar{x}$ можно описать с помощью следующей таблицы:

x	$\bar{x}$
1	0
0	1

Таблицы подобного вида принято называть *таблицами истинности*.

Пусть x — высказывание. Так как $\bar{x}$ также является высказыванием, можно образовать отрицание высказывания $\bar{x}$, то есть высказывание $\bar{\bar{x}}$, которое называется двойным отрицанием высказывания x . Ясно, что логические значения $\bar{\bar{x}}$ и x совпадают.

Пример. Пусть имеется высказывание x «река Волга впадает в Каспийское море». Тогда высказывание «Неверно, что река Волга впадает в Каспийское море» будет отрицанием, то есть $\bar{x}$, а высказывание «Неверно, что река Волга не впадает в Каспийское море» — двойным отрицанием, то есть $\bar{\bar{x}}$.

Операция *конъюнкции* высказываний x и y обозначается символом $\wedge$, а выражение $x \wedge y$ читается « x и y ». Высказывания x и y называются членами конъюнкции.

Конъюнкция (логическое умножение) двух высказываний x и y — это новое высказывание, которое считается истинным, если оба высказывания, x и y , истинны, и ложным, если хотя бы одно из них ложно.

Логические значения операции конъюнкции описываются следующей таблицей истинности:

x	y	$x \wedge y$
1	1	1
1	0	0
0	1	0
0	0	0

Пример. Если мы обозначим высказывание «6 делится на 2» как x , а «6 делится на 3» как y , их конъюнкцией будет высказывание «6 делится на 2 и 6 делится на 3», которое записывается как $x \wedge y$ и является истинным.

Из определения операции конъюнкции видно, что союз «и» в алгебре логики употребляется в том же смысле, что и в повседневной речи. Но в обычной речи не принято соединять союзом «и» два высказывания, далеких друг от друга по содержанию, а в алгебре логики рассматриваются конъюнкции любых двух высказываний.

Из определения операции конъюнкции и отрицания ясно, что высказывание $x \wedge \bar{x}$ всегда ложно.

Операция *дизъюнкции* высказываний x и y обозначается символом $\vee$, а выражение $x \vee y$ читается как « x или y ». Высказывания x и y называются членами дизъюнкции.

Дизъюнкция (логическое сложение) двух высказываний, x и y , — это новое высказывание, которое считается истинным, если хотя бы одно из высказываний, x и y , истинно, и ложным, если они оба ложны.

Логические значения операции дизъюнкции описываются следующей таблицей истинности:

x	y	$x \vee y$
1	1	1
1	0	1
0	1	1
0	0	0

Пример. Обозначим высказывание «В треугольнике DFE угол D острый» как x , а высказывание «В треугольнике DFE угол E острый» как y . Тогда дизъюнкция $x \vee y$ этих высказываний «В треугольнике DFE угол D или угол E острый» истинна, так как обязательно истинно хотя бы одно из высказываний.

В повседневной речи союз «или» употребляется в различном смысле: исключающем и не исключающем. В алгебре логики союз «или» всегда употребляется в не исключающем смысле.

Из определения операции дизъюнкции и отрицания ясно, что высказывание $x \vee \bar{x}$ всегда истинно.

Операция *импликации* высказываний x и y обозначается символом $\supset$, а выражение $x \supset y$ читается как «если x , то y ». Высказывание x называют условием, или посылкой, высказывание y — следствием, или заключением, а высказывание $x \supset y$ — следованием, или импликацией.

Импликация двух высказываний, x и y , — это новое высказывание, которое считается ложным, если x истинно, а y ложно, и истинным во всех остальных случаях.

Логические значения операции импликации описываются следующей таблицей истинности:

x	y	$x \supset y$
1	1	1
1	0	0
0	1	1
0	0	1

Пример. Обозначив высказывание «Число 12 делится на 6» как x , а высказывание «Число 12 делится на 3» как y , мы получим импликацию $x \supset y$, которая отражает высказывание «Если число 12 делится на 6, то оно делится на 3» и является истинным.

Употребление слов «если..., то...» в алгебре логики отличается от употребления их в обыденной речи, где мы, как правило, считаем, что если высказывание x ложно, то высказывание «Если x , то y » не имеет смысла. Кроме того, строя предложение

«Если x , то y », мы всегда подразумеваем, что предложение y вытекает из предложения x . Употребление слов «если..., то...» в математической логике не требует этого, поскольку в ней смысл высказываний не рассматривается.

Операция *эквивалентности* высказываний x и y обозначается символом $\leftrightarrow$, а выражение $x \leftrightarrow y$ читается «для того чтобы x , необходимо и достаточно, чтобы y » или « x тогда и только тогда, когда y ». Высказывания x и y называются членами эквивалентности.

Эквивалентность двух высказываний, x и y , — это новое высказывание, которое считается истинным, когда оба высказывания (x и y) либо одновременно истинны, либо одновременно ложны, и ложным во всех остальных случаях.

Логические значения операции эквивалентности описываются следующей таблицей истинности:

x	y	$x \leftrightarrow y$
1	1	1
1	0	0
0	1	0
0	0	1

Пример. Обозначив высказывание «Треугольник SPQ с вершиной S и основанием PQ равнобедренный» как x , а высказывание «В треугольнике SPQ с вершиной S и основанием PQ $\angle P = \angle Q$ » как y , мы можем записать высказывание «Треугольник SPQ с вершиной S и основанием PQ равнобедренный тогда и только тогда, когда $\angle P = \angle Q$ » в форме эквивалентности $x \leftrightarrow y$. Эквивалентность является истинной, так как высказывания либо одновременно истинны, либо одновременно ложны.

Эквивалентность играет важную роль в математических доказательствах. Известно, что значительное количество теорем формулируется в форме необходимых и достаточных условий, то есть в форме эквивалентности. В этом случае, зная об истинности или ложности одного из двух членов эквивалентности и доказав истинность самой эквивалентности, мы приходим к заключению об истинности или ложности второго члена эквивалентности.

4.2. Алгебра логики

4.2.1. Понятие формулы алгебры логики

С помощью логических операций над высказываниями можно строить сложные высказывания. При этом порядок выполнения операций определяется скобками. Например, из трех высказываний, x , y , z , можно построить два высказывания:

$$(x \wedge y) \vee \bar{z} \quad \text{и} \quad x \supset (\overline{y \vee (x \wedge z)}).$$

Первое высказывание есть дизъюнкция конъюнкции x, y и отрицания высказывания z . Второе высказывание есть импликация, посылкой которой является высказывание x , а заключением — отрицание дизъюнкции высказывания y и конъюнкции высказываний x, z .

Формула алгебры логики — это сложное высказывание, которое может быть получено из элементарных высказываний посредством применения логических операций отрицания, конъюнкции, дизъюнкции, импликации и эквивалентности.

Формулы алгебры логики будем обозначать прописными буквами латинского алфавита $A, B, C, \dots$.

Буквенные обозначения формул алгебры логики используются в основном в определениях, когда надо обозначить некоторые общие для формул закономерности.

Для того чтобы логическими высказываниями, сформулированными на естественном языке, можно было оперировать при помощи алгебры логики, их необходимо формализовать, то есть перевести с естественного языка на язык символов алгебры логики. Для этого рекомендуется следующая процедура:

1. Если высказывание простое, то ему ставится в соответствие элементарная формула.
2. Если высказывание составное, то для составления соответствующей формулы нужно:
 - 1) выделить все элементарные высказывания и логические связки, образующие данное составное высказывание;
 - 2) заменить их соответствующими символами (различные элементарные высказывания обозначаются различными символами);
 - 3) расставить скобки в соответствии со смыслом данного высказывания.

Правила записи формулы:

Скобки можно опускать, придерживаясь следующего порядка действий:

- конъюнкция выполняется раньше, чем все остальные операции;
- дизъюнкция выполняется раньше, чем импликация и эквивалентность.

Если над формулой стоит знак отрицания, то скобки опускаются.

Количество открывающих скобок должно быть равно количеству закрывающих.

Пример. $(x \wedge y) \vee \bar{z} \equiv x \wedge y \vee \bar{z}$ — мы опустили скобки, в которые была взята операция дизъюнкции.

$x \supset (y \vee (x \wedge z)) \equiv x \supset y \vee (x \wedge z)$ — мы опустили скобки, в которые было заключена формула со знаком отрицания над ней.

Логическое значение формулы алгебры логики полностью определяется результатом логических операций над значениями входящих в нее элементарных высказываний. Например, логическим значением формулы $\overline{x \wedge y} \vee \bar{z}$ в случае, если $x = 1, y = 1, z = 0$, будет истина, то есть $\overline{x \wedge y} \vee \bar{z} = 1$.

Таблица истинности позволяет полностью описать все возможные значения любой формулы в зависимости от значений входящих в нее элементарных высказываний.

Очевидно, что если в формуле n элементарных высказываний, то она принимает 2^n значений, состоящих из нулей и единиц, и результирующая таблица содержит 2^n строк.

Для формулы таблица истинности имеет вид

x	y	$\bar{x}$	$\bar{y}$	$\bar{x} \vee y$	$x \vee \bar{y}$	$\bar{x} \vee y \supset x \vee \bar{y}$
1	1	0	0	1	0	0
1	0	0	1	0	1	1
0	1	1	0	1	0	0
0	0	1	1	1	0	0

4.2.2. Равносильные формулы алгебры логики

Равносильные формулы — это две формулы алгебры логики, A и B , если они принимают одинаковые логические значения на любом наборе значений.

Равносильность формул будем обозначать знаком $\equiv$, тогда запись означает, что формулы A и B равносильны.

Пример. Равносильные формулы:

$$\begin{aligned}\bar{\bar{x}} &\equiv x, \\ x \vee x &\equiv x, \\ (x \wedge \bar{x}) \vee y &\equiv y.\end{aligned}$$

Тавтологией, или **тождественно истинной**, называется формула, принимающая значение 1 при всех значениях входящих в нее переменных.

Пример. Тождественно истинные формулы:

$$\begin{aligned}x \vee \bar{x}, \\ x \supset (x \supset y).\end{aligned}$$

Тождественно ложной называется формула, принимающая значение 0 при всех значениях входящих в нее переменных.

Пример. Тождественно ложная формула: $x \wedge \bar{x}$.

Между понятиями равносильности и эквивалентности существует следующая связь: если формулы A и B равносильны, то формула $A \leftrightarrow B$ — тавтология, и обратно, если формула $A \leftrightarrow B$ — тавтология, то формулы A и B равносильны.

Важнейшие равносильности алгебры логики можно разбить на три группы:

- основные равносильности;
- равносильности, выражающие одни логические операции через другие;
- равносильности, выражающие основные законы алгебры логики.

Используя приведенные далее равносильности, можно часть формулы или формулу полностью заменить равносильной ей формулой. Такие преобразования формул называются *равносильными*.

Равносильные преобразования используются для доказательства равносильностей, для приведения формул к заданному виду, для упрощения формул.

Формула A считается проще равносильной ей формулы B , если она содержит меньше символов, меньше логических операций. При этом обычно операции эквивалентности и импликации заменяются операциями дизъюнкции и конъюнкции, а отрицание относят к элементарным высказываниям.

4.2.3. Основные равносильности

1. $x \wedge x = x$.
2. $x \vee x = x$.
3. $x \wedge 1 = x$.
4. $x \vee 1 = 1$.
5. $x \wedge 0 = 0$.
6. $x \vee 0 = x$.
7. $x \wedge \bar{x} = 0$ — закон противоречия.
8. $x \vee \bar{x} = 1$ — закон исключения третьего.
9. $\bar{\bar{x}} = x$ — закон снятия двойного отрицания.
10. $x \wedge (y \vee x) = x$.
11. $x \vee (y \wedge x) = x$.

4.2.4. Равносильности, выражающие одни логические операции через другие

1. $(x = y) = (x \supset y) (y \supset x)$.
2. $x \supset y = \bar{x} \vee y$.
3. $\overline{x \wedge y} = \bar{x} \vee \bar{y}$.
4. $\overline{x \vee y} = \bar{x} \wedge \bar{y}$.
5. $x \wedge y = \overline{\bar{x} \vee \bar{y}}$.
6. $x \vee y = \overline{\bar{x} \wedge \bar{y}}$.

Из равносильностей этой группы следует, что всякую формулу алгебры логики можно заменить равносильной ей формулой, содержащей только две логические операции: конъюнкцию и отрицание или дизъюнкцию и отрицание.

Дальнейшее исключение логических операций невозможно. Так, если мы будем использовать только конъюнкцию, то уже такая формула, как отрицание x , не может быть выражена с помощью операции конъюнкции.

Однако существуют операции, с помощью которых может быть выражена любая из пяти логических операций, которыми мы пользуемся. Такой операцией является, например, операция Штрих Шеффера. Эта операция обозначается символом $|$ и определяется следующей таблицей истинности:

x	y	$x y$
1	1	0
1	0	1
0	1	1
0	0	1

Очевидно, имеют место равносильности:

1. $\bar{x} \equiv x | x$.
2. $x \wedge y \equiv (x | y) | (x | y)$.

Из этих двух равносильностей следует, что всякая формула алгебры логики может быть заменена равносильной формулой, содержащей только операцию Штрих Шеффера.

Отметим также, что $x \vee y \equiv \overline{x | y}$.

4.2.5. Равносильности, выражающие основные законы алгебры логики

1. $x \wedge y \equiv y \wedge x$ — коммутативность конъюнкции.
2. $x \vee y \equiv y \vee x$ — коммутативность дизъюнкции.
3. $x \wedge (y \wedge z) \equiv (x \wedge y) \wedge z$ — ассоциативность конъюнкции.
4. $x \vee (y \vee z) \equiv (x \vee y) \vee z$ — ассоциативность дизъюнкции.
5. $x \wedge (y \vee z) \equiv (x \wedge y) \vee (x \wedge z)$ — дистрибутивность конъюнкции относительно дизъюнкции.
6. $x \vee (y \wedge z) \equiv (x \vee y) \wedge (x \vee z)$ — дистрибутивность дизъюнкции относительно конъюнкции.

4.2.6. Решение логических задач методами алгебры логики

Суть применения методов алгебры логики к решению логических задач состоит в том, что конкретные условия логической задачи необходимо представить в виде формул алгебры логики. В дальнейшем путем равносильных преобразований полученную формулу упрощают. Полученная в результате преобразований упрощенная формула, как правило, приводит к ответу на вопрос задачи.

Пример. Пытаясь вспомнить победителей прошлогоднего турнира, пять бывших зрителей турнира заявили:

1. Антон был вторым, а Борис — пятым.
2. Виктор был вторым, а Денис — третьим.
3. Григорий был первым, а Борис — третьим.
4. Антон был третьим, а Евгений — шестым.
5. Виктор был третьим, а Евгений — четвертым.

Впоследствии выяснилось, что каждый зритель ошибся в одном из двух своих высказываний. Каково было истинное распределение мест в турнире?

Обозначим участника первой буквой его имени, нижний индекс около буквы (цифра) будет обозначать номер места, которое он занял в турнире, то есть в выражении X_y имеем, что X — это участник турнира, а y — номер места, которое он занял в турнире. Обозначим буквой L истинное распределение мест в турнире ($L \equiv 1$).

Так как в паре высказываний каждого зрителя одно истинно, а второе ложно, то дизъюнкции этих высказываний будут истинными:

$$A_2 \vee B_5 \equiv 1,$$

$$B_2 \vee D_3 \equiv 1,$$

$$G \vee B_3 \equiv 1,$$

$$A_3 \vee E_6 \equiv 1,$$

$$B_3 \vee E_4 \equiv 1.$$

Тогда будет истинной формула

$$L \equiv (A_2 \vee B_5) \wedge (B_2 \vee D_3) \wedge (G \vee B_3) \wedge (A_3 \vee E_6) \wedge (B_3 \vee E_4).$$

Исходя из данных условия, можно начать преобразования. Поскольку одно из входящих в дизъюнкции высказываний обязательно истинно, а второе обязательно ложно, то за отправную точку в преобразованиях можно принять пару $G \vee B_3 \equiv 1$. В этой паре высказывание G является истинным, а B_3 — ложным, поскольку на первое место нет других претендентов. Тогда, используя равносильность $x \vee 0 \equiv x$, получаем $G \vee 0 \equiv G$, и формула приводится к виду

$$L \equiv (A_2 \vee B_5) \wedge (B_2 \vee D_3) \wedge G \wedge (A_3 \vee E_6) \wedge (B_3 \vee E_4).$$

Поскольку вариант B_3 оказался ложным, а других претендентов на пятое место нет, то истинным будет вариант B_5 . Тогда в первой по счету паре, применив ту же равносильность, что и на предыдущем шаге, мы также оставляем одно высказывание, и основная формула принимает вид

$$L \equiv B_5 \wedge (B_2 \vee D_3) \wedge G \wedge (A_3 \vee E_6) \wedge (B_3 \vee E_4).$$

Продолжая эти равносильные преобразования, приводим формулу к виду

$$L \equiv B_5 \wedge B_2 \wedge G \wedge A_3 \wedge E_4.$$

Отсюда следует, что $B_5 \equiv 1$, $B_2 \equiv 1$, $G \equiv 1$, $A_3 \equiv 1$, $E_4 \equiv 1$, то есть первым был Григорий, вторым — Виктор и т. д. Это и является ответом на вопрос, поставленный в задаче.

4.2.7. Булева алгебра

Без булевой алгебры было бы невозможно создание таких устройств, как современные компьютеры, в состав центральных процессоров которых входят миллиарды электронных переключателей.

Рассмотрим непустое множество M элементов любой природы $\{x, y, z, \dots\}$, в котором определены отношение равенства ($=$) и три операции: сложение ($+$), умножение ($\cdot$) и отрицание ($\bar{}$), подчиняющиеся следующим законам:

□ *Коммутативные законы:*

$$x + y = y + x;$$

$$x \cdot y = y \cdot x.$$

□ *Ассоциативные законы:*

$$x + (y + z) = (x + y) + z;$$

$$x \cdot (y \cdot z) = (x \cdot y) \cdot z.$$

□ *Дистрибутивные законы:*

$$((x + y) \cdot z) = (x \cdot z) + (y \cdot z);$$

$$((x \cdot y) + z) = (x + z) \cdot (y + z).$$

□ *Законы идемпотентности:*

$$x + x = x;$$

$$x \cdot x = x.$$

□ *Закон двойного отрицания:*

$$\overline{\overline{x}} = x.$$

□ *Законы де-Моргана:*

$$\overline{x + y} = \overline{x} \cdot \overline{y};$$

$$\overline{x \cdot y} = \overline{x} + \overline{y}.$$

□ *Законы поглощения:*

$$x \cdot (y + x) = x;$$

$$x + (y \cdot x) = x.$$

Такое множество M называют *булевой алгеброй*.

Если под основными элементами $x, y, z, \dots$ подразумевать высказывания, под операциями сложения, умножения и отрицания булевой алгебры — соответственно, дизъюнкцию, конъюнкцию и отрицание алгебры высказываний, а знак равенства рассматривать как знак равносильности, то, как следует из равносильностей алгебры высказываний, все аксиомы булевой алгебры выполняются.

В тех случаях, когда для некоторой системы аксиом удастся подобрать конкретные объекты и конкретные соотношения между ними так, что все аксиомы выполняются, говорят, что найдена *интерпретация*, или *модель*, данной системы аксиом.

Значит, алгебра логики является интерпретацией булевой алгебры. Булева алгебра имеет и другие интерпретации. Например, если под основными элементами $x, y, z, \dots$ множества M подразумевать множества, под операциями булевой

алгебры — соответственно, объединение, пересечение и дополнение множеств, а под знаком равенства — знак равенства множеств, то мы приходим к алгебре множеств. Нетрудно убедиться, что и в алгебре множеств все аксиомы булевой алгебры выполняются.

Функция Були (или функция алгебры логики) n переменных — это функция n переменных, в которой и каждая переменная, и сама функция может принимать только одно из двух значений, 0 или 1.

Ясно, что тождественно истинные и тождественно ложные формулы алгебры логики представляют собой постоянные функции, а две равносильные формулы выражают одну и ту же функцию.

Каково количество функций n переменных? Очевидно, каждую функцию алгебры логики (как и формулу алгебры логики) можно задать с помощью таблицы истинности, которая будет содержать 2^n строк. Следовательно, каждая функция n переменных принимает 2^n значений, состоящих из нулей и единиц. Таким образом, функция n переменных полностью определяется набором значений из нулей и единиц длины 2^n . Общее же количество наборов, состоящих из нулей и единиц длины 2^n равно 2^{2^n} . Значит, количество различных функций алгебры логики n переменных равно 2^{2^n} .

В частности, различных функций одной переменной четыре, а различных функций двух переменных — шестнадцать. Выпишем все функции алгебры логики одной и двух переменных.

Рассмотрим таблицу истинности для различных функций одной переменной. Она, очевидно, имеет вид

x	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
1	1	1	0	0
0	1	0	1	0

Из этой таблицы следует, что две функции одной переменной будут постоянными: $f_1(x) = 1$ и $f_4(x) = 0$, а $f_2(x) = x$ и $f_3(x) = \bar{x}$.

Таблица истинности для всевозможных функций двух переменных имеет вид

x	y	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}
1	1	1	1	1	1	0	1	1	0	0	0	1	0	0	0	1	0
1	0	1	1	1	0	1	1	0	0	1	1	0	0	0	1	0	0
0	1	1	1	0	1	1	0	0	1	1	0	1	0	1	0	0	0
0	0	1	0	1	1	1	0	1	1	0	1	0	1	0	0	0	0

Ясно, что аналитические выражения этих функций могут быть записаны следующим образом:

$f_1 \equiv 1;$	$f_5 \equiv \overline{x \wedge y};$	$f_9 \equiv \overline{x \leftrightarrow y};$	$f_{13} \equiv \overline{y \supset x};$
$f_2 \equiv x \vee y;$	$f_6 \equiv x;$	$f_{10} \equiv \overline{y};$	$f_{14} \equiv \overline{x \supset y};$
$f_3 \equiv y \supset x;$	$f_7 \equiv x \leftrightarrow y;$	$f_{11} \equiv y;$	$f_{15} \equiv x \wedge y;$
$f_4 \equiv x \supset y;$	$f_8 \equiv \overline{x};$	$f_{12} \equiv \overline{x \vee y};$	$f_{16} \equiv 0.$

4.3. Построение коммутационных схем на основе алгебры логики

Среди технических средств автоматизации значительное место занимают устройства релейно-контактного действия. Они широко используются в технике автоматического управления, в электронно-вычислительной технике, и т. д.

Эти устройства (их в общем случае называют переключательными, или коммутационными, схемами) содержат сотни реле, полупроводниковых элементов и других переключающих элементов. Описание и конструирование таких схем в силу их большого объема представляет трудную задачу.

Еще в 1910 году физик Н. С. Эренфест указал на возможность применения аппарата алгебры логики при исследовании релейно-контактных схем. Однако его идеи начали реализовываться значительно позже, когда создание общей теории конструирования таких схем стало остро необходимым.

Использование алгебры логики в конструировании коммутационных схем оказалось возможным в связи с тем, что каждой схеме можно поставить в соответствие некоторую формулу алгебры логики и каждая формула алгебры логики реализуется с помощью некоторой схемы. Это обстоятельство помогает выявить возможности упрощения заданной схемы, изучая соответствующую формулу, а упрощение формулы затем реализовать как упрощение схемы.

В то же время еще до построения схемы можно заранее описать с помощью формул те функции, которые схема должна выполнять. Рассмотрим, как устанавливается связь между формулами алгебры логики и переключательными схемами.

Под переключательной схемой понимается схематическое изображение некоторого устройства, состоящего из

- переключателей;
- соединительных проводников;
- входов в схему и выходов из нее.

Переключателями могут быть электромеханические устройства (выключатели, переключатели, кнопки), электромагнитные реле, полупроводниковые элементы и т. п., а входами и выходами — клеммы, на которые подается электрическое напряжение.

Коммутационной схемой принимается в расчет только два состояния каждого переключателя, которые называются «замкнутым» и «разомкнутым».

Рассмотрим схему переключения, состоящую из источника питания и электрической лампочки (рис. 4.1).

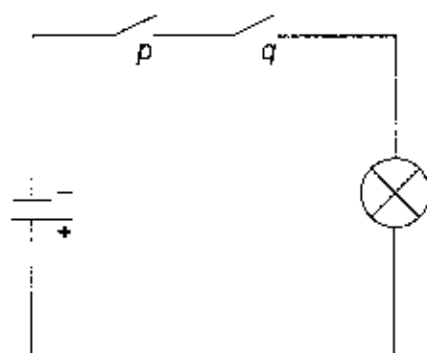


Рис. 4.1. Схема с последовательным соединением переключателей

Присвоим значение 1 переключателям p и q , если они замкнуты (то есть электрический ток проходит через них). В противоположной ситуации присвоим им значение 0. Присвоим значение 1 схеме, когда лампочка светится (то есть электрический ток через нее проходит). Заметим, что при последовательном соединении элементов цепи p и q , как это имеет место на приведенной схеме, лампочка загорается и значение схемы становится равным 1 только в том случае, когда оба переключателя замкнуты, то есть когда и p , и q имеют значение 1. Таким образом, схема соответствует высказыванию $p \wedge q$. Такое расположение переключателей называется логическим элементом p и q , или схемой логического умножения. Этот логический элемент обозначается символом, изображенным на рис. 4.2.

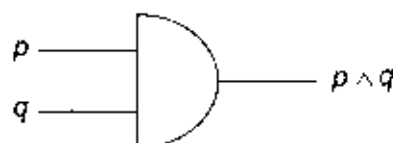


Рис. 4.2. Элемент логического умножения

Рассмотрим схему переключения, показанную на рис. 4.3, где переключатели p и q соединены параллельно.

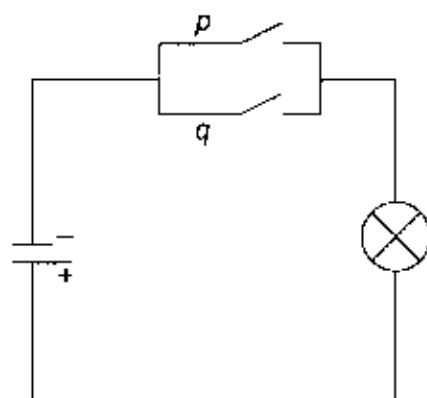


Рис. 4.3. Схема с параллельным соединением переключателей

Отметим, что лампочка загорается и значение схемы становится равным 1, когда один из двух переключателей (p или q) замкнут, то есть либо $p = 1$, либо $q = 1$ (либо оба они равны 1). Эта схема соответствует высказыванию $p \vee q$. Такое

расположение переключателей называется логическим элементом p или q , или схемой *логического сложения*. Этот логический элемент обозначается символом, изображенным на рис. 4.4.

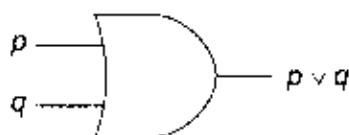


Рис. 4.4. Элемент логического сложения

Предположим, имеется схема с одним переключателем p , который обладает таким свойством, что лампочка загорается тогда и только тогда, когда переключатель разомкнут. Следовательно, схема имеет значение 1, когда p равно 0, и значение 0, когда p равно 1. Эта схема соответствует $\bar{p}$, а соответствующий логический элемент называется *логическим элементом не*, или *инвертором* (рис. 4.5).



Рис. 4.5. Инвертор

Пример. Схема на рис. 4.6 содержит логический элемент p и q , за которым следует инвертор, так что схема соответствует выражению $\overline{p \wedge q}$. Заметим, что инвертор отрицает всю предшествующую ему схему.

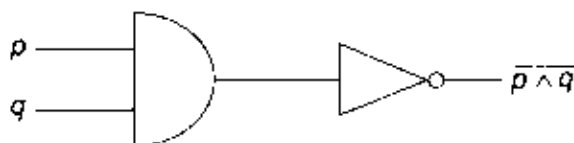


Рис. 4.6. Логическая схема $\overline{p \wedge q}$

Пример. Схема на рис. 4.7 содержит соединение логического элемента p или q с логическим элементом *не* r посредством логической схемы умножения. Следовательно, она соответствует выражению $(p \vee q) \wedge \bar{r}$.

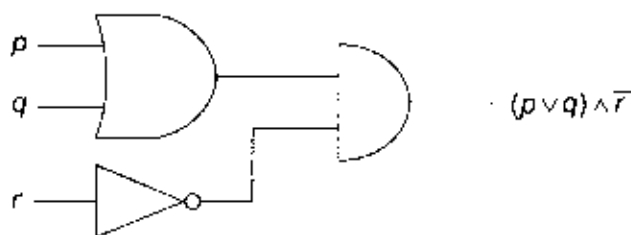
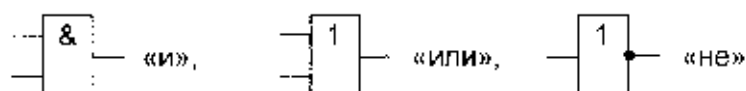


Рис. 4.7. Логическая схема $(p \vee q) \wedge \bar{r}$

Из трех логических элементов, соответствующих выражениям сложения, умножения и отрицания в булевой алгебре, можно строить электронные логические схемы любой сложности.

На практике часто встречаются обозначения, принятые для цифровых микросхем:



Вопросы для самопроверки

1. Почему изучение логических основ так важно для понимания информатики?
2. Дайте определение понятию простого высказывания. Приведите примеры.
3. Является ли предложение «Который сейчас час?» простым высказыванием с точки зрения математической логики?
4. Какие логические операции над высказываниями вам известны?
5. Дайте определение конъюнкции и дизъюнкции, приведите примеры.
6. Дайте определение импликации и эквивалентности, приведите примеры.
7. Дайте определение формулы алгебры логики.
8. Какие формулы алгебры логики называются равносильными?
9. Приведите формулы законов идемпотентности.
10. Приведите формулы законов поглощения.
11. Как записывается и чем примечательна логическая операция Итрих Шеффера?
12. Какими формулами записываются законы идемпотентности в булевой алгебре?
13. Дайте определение функции Буля.
14. Что такое «переключательная (коммутационная) схема»?
15. Нарисуйте коммутационную схему, реализующую логическую операцию И.
16. Нарисуйте коммутационную схему, реализующую логическую операцию ИЛИ.
17. Чем важна булева алгебра для информатики и компьютерной техники?

Литература

1. Азиев Ю. А., Тюрин С. Ф. Дискретная математика и математическая логика. М.: Финансы и статистика, 2006.
2. Вережагин Н. К., Плиско И. К., Успенский В. А. Вводный курс математической логики. М.: Физматлит, 2004.
3. Новиков Ф. А. Дискретная математика для программистов. СПб.: Питер, 2003.

Глава 5 Информационные системы и технологии

- 5.1. Основные сведения об информационных системах
- 5.2. Структура и классификация информационных систем
- 5.3. Основные сведения об информационных технологиях
- 5.4. Виды информационных технологий

В прошлом информация считалась сферой бюрократической работы и ограниченным инструментом для принятия решений. Сегодня информацию рассматривают как один из основных ресурсов развития общества, а информационные системы (ИС) и технологии (ИТ) — как средство повышения производительности и эффективности работы людей.

Наиболее широко информационные системы и технологии используются в производственной, управленческой и финансовой деятельности, хотя относительно необходимости их внедрения и активного применения начались подвижки в сознании людей, запятых и в других сферах. Это определило угол зрения, под которым будут рассмотрены основные области их применения. Главное внимание уделяется рассмотрению информационных систем и технологий с позиций использования их возможностей для повышения эффективности труда работников информационной сферы производства и поддержки принятия решений в организациях (фирмах).

5.1. Основные сведения об информационных системах

5.1.1. Понятие информационной системы

Под *системой* понимают любой объект, который одновременно рассматривается и как единое целое, и как объединенная в интересах достижения поставленных целей совокупность разнородных элементов. Системы значительно различаются между собой как по составу, так и по достигаемым целям.

Информационные системы обеспечивают сбор, хранение, обработку, поиск и выдачу информации, необходимой в процессе принятия решений в любой области деятельности. Они помогают анализировать проблемы и создавать новые продукты.

Информационная система — это взаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации в интересах достижения поставленной цели.

Современное понимание информационной системы предполагает использование персонального компьютера в качестве основного технического средства переработки информации. В крупных организациях наряду с персональным компьютером в состав технической базы информационной системы может входить мэйнфрейм или суперЭВМ. Кроме того, техническое воплощение информационной системы само по себе ничего не будет значить, если не учтена роль человека, для которого предназначена производимая информация и без которого невозможно ее получение и представление.

ВНИМАНИЕ

Под организацией мы будем понимать сообщество людей, объединенных общими целями и использующих общие материальные и финансовые средства для производства материальных и информационных продуктов и услуг. Помимо слова «организация» в тексте на равных правах будет употребляться слово «фирма».

5.1.2. Этапы развития информационных систем

История развития информационных систем и цели их использования в разные периоды представлены в табл. 5.1.

Зарождение основ информационных систем можно отнести в конце 50-х гг. XX столетия, когда обработка счетов и расчет зарплаты производились на электро-механических бухгалтерских счетных машинах. Это приводило к уменьшению объема рутинных операций и некоторому сокращению затрат и времени на подготовку бумажных документов.

Теоретическое и практическое развитие информационных системы получили начиная с 70-х гг. XX столетия, когда для разных сфер деятельности, и в первую очередь для производственной сферы, стали разрабатываться и внедряться авто-

матризованные системы управления (АСУ). Информация, полученная с их помощью, стала применяться для периодической отчетности. В качестве технической базы в этих системах использовались большие и малые ЭВМ, способные выполнять множество функций.

Таблица 5.1. Изменение подхода к использованию информационных систем

Период времени	Концепция использования информации	Вид информационных систем	Цель использования
1955–1970 гг.	Бумажный поток расчетных документов	Информационные системы обработки расчетных документов на электромеханических бухгалтерских машинах	Повышение скорости обработки документов, а также упрощение процедур обработки счетов и расчета зарплат
1970–1980 гг.	Основная помощь в подготовке отчетов	Управленческие информационные системы для производственной сферы	Ускорение процесса подготовки отчетности
1980–2000 гг.	Управленческий контроль реализации (продаж)	Системы поддержки принятия решений и системы для высшего звена управления	Выработка наиболее рационального решения
2000 гг. по настоящее время	Стратегический ресурс, обеспечивающий конкурентное преимущество	Стратегические информационные системы и автоматизированные офисы	Выживание и процветание организации

С 80-х гг. XX столетия информационные системы начинают широко использоваться в качестве средств управленческого контроля, поддерживающего и ускоряющего процесс принятия решений.

К началу 2000 гг. концепция использования информационных систем вновь изменилась. Они стали стратегическим источником информации и до настоящего времени продолжают использоваться на всех уровнях организации любого профиля. Информационные системы, предоставляя вовремя нужную информацию, помогают организации достичь успеха в своей деятельности, создавать новые товары и услуги, находить новые рынки сбыта, обеспечивать себе достойных партнеров, организовывать выпуск продукции по низкой цене и многое другое.

5.1.3. Процессы в информационной системе

Процессы, обеспечивающие работу информационной системы любого назначения, условно можно представить в виде схемы, состоящей из блоков (рис. 5.1):

- ввод информации из внешних или внутренних источников;
- обработка входной информации и представление ее в удобном виде;
- вывод информации для представления потребителям или передачи в другую систему;
- обеспечение обратной связи, то есть коррекция входной информации людьми данной организации.

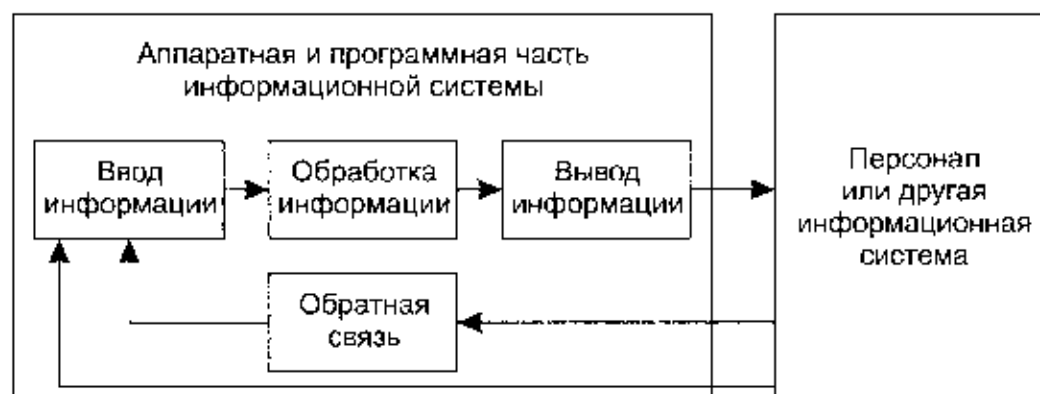


Рис. 5.1. Процессы в информационной системе

Хотя в настоящее время об информационной системе сложилось мнение как о системе, реализованной с помощью компьютерной техники, в общем случае ее можно понимать и в некомпьютерном варианте.

5.1.4. Результаты внедрения информационных систем

Внедрение информационных систем может способствовать:

- ☐ получению более рациональных вариантов решения управленческих задач за счет внедрения математических методов и интеллектуальных систем и т. д.;
- ☐ освобождению работников от рутинной работы за счет ее автоматизации;
- ☐ обеспечению достоверности информации;
- ☐ замене бумажных носителей данных на магнитные диски или ленты, что приводит к более рациональной организации переработки информации на компьютере и снижению объемов документов на бумаге;
- ☐ совершенствованию структуры потоков информации и системы документооборота в фирме;
- ☐ снижению затрат на производство продуктов и услуг;
- ☐ предоставлению потребителям уникальных услуг;
- ☐ отысканию новых рыночных ниш;
- ☐ привязке к фирме покупателей и поставщиков за счет предоставления им разных скидок и услуг.

5.1.5. Роль структуры управления в информационной системе

Построение информационной системы должно начинаться с анализа структуры управления организацией.

Координация работы всех подразделений организации осуществляется через органы управления разного уровня. Под *управлением* понимают достижение по-

ставленной цели при условии реализации следующих функций: организационной, плановой, учетной, анализа, контрольной, стимулирования. Рассмотрим содержание управленческих функций.

Организационная функция заключается в разработке организационной структуры и комплекса нормативных документов: штатного расписания с указанием подчиненности, ответственности, сферы компетенции, прав, обязанностей и т. п.

Планирование (плановая функция) состоит в разработке и реализации планов по решению поставленных задач.

Учетная функция заключается в разработке или использовании уже готовых форм и методов учета показателей деятельности фирмы, включая бухгалтерский учет, финансовый учет, управленческий учет и т. п.

Анализ, или аналитическая функция, связывается с изучением итогов выполнения планов и заказов, определением влияющих факторов, выявлением резервов, изучением тенденций развития и т. д.

Контрольная функция чаще всего осуществляется менеджером и касается контроля за выполнением планов, расходом материальных ресурсов, использованием финансовых средств и т. п.

Функция *стимулирования*, или *мотивационная* функция, предполагает разработку и применение различных методов стимулирования труда подчиненных работников:

- финансовые стимулы — зарплата, премия, акции, повышение в должности и т. п.;
- психологические стимулы — благодарности, грамоты, звания, степени, доски почета и т. п.

В последние годы в сфере управления все активнее стало применяться понятие принятия решения и связанные с этим понятием системы, методы, средства поддержки принятия решений.

Принятие решения — это акт выбора целенаправленного воздействия на объект управления, основанный на анализе ситуации, определении цели, разработке программы достижения этой цели.

Структура управления любой организации традиционно делится на три уровня: операционный, функциональный и стратегический.

Уровни управления (вид управленческой деятельности) определяются сложностью решаемых задач. Чем сложнее задача, тем более высокий уровень управления требуется для ее решения. При этом следует понимать, что более простых задач, требующих немедленного (оперативного) решения, возникает значительно большее количество, а значит, и уровень управления для них нужен другой — более низкий, где решения принимаются оперативно.

На рис. 5.2 представлены три уровня управления, которые соотносятся с такими факторами, как степень возрастания власти, ответственности, сложности решаемых задач, а также динамика принятия решений.

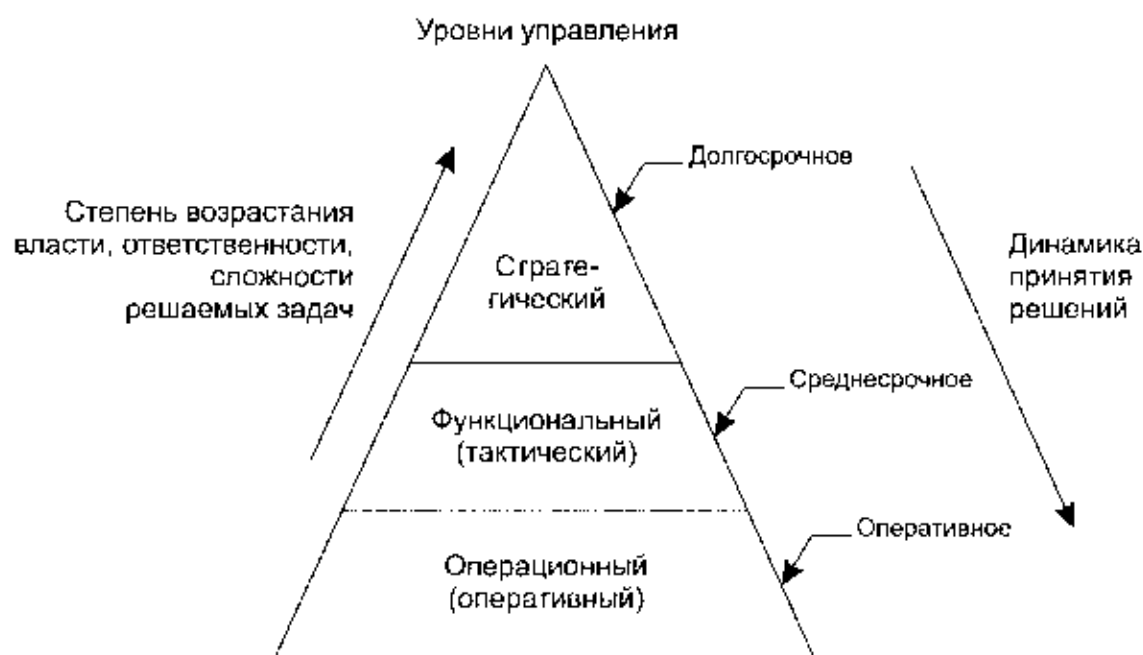


Рис. 5.2. Пирамида уровней управления, отражающая возрастание власти, ответственности, сложности и динамику принятия решений

Операционный (оперативный) уровень управления обеспечивает решение многократно повторяющихся задач и операций, а также быстрое реагирование на изменения входной текущей информации. На этом уровне достаточно велики как объем выполняемых операций, так и динамика принятия управленческих решений. Этот уровень управления часто называют *оперативным* из-за необходимости быстрого реагирования на изменение ситуации. На уровне оперативного (операционного) управления большой объем занимают учетные задачи.

Пример. В качестве примера можно привести учет затрат времени, сырья и материалов при выполнении отдельных производственных операций, бухгалтерский учет и т. д.

Функциональный (тактический) уровень управления обеспечивает решение задач, требующих предварительного анализа информации, подготовленной на первом уровне. На этом уровне большое значение приобретает такая функция управления, как анализ. Объем решаемых задач уменьшается, но возрастает их сложность. Управление связано с некоторой задержкой от момента поступления информации до принятия решений и их реализации, а также от момента реализации решений до получения реакции на них.

Пример. На основании анализа статистических данных по спросу на продукцию, о ценах конкурентов и пр. прогнозируется прибыль и разрабатывается план выпуска продукции на ближайший период (неделю, месяц, квартал). Результаты принимаемых управленческих решений проявляются спустя некоторое время.

Стратегический уровень обеспечивает выработку управленческих решений, направленных на достижение долгосрочных стратегических целей организации.

Поскольку результаты принимаемых решений проявляются спустя длительное время, особое значение на этом уровне имеет такая функция управления, как стратегическое планирование. Прочие функции управления на этом уровне в настоящее время разработаны недостаточно полно. Часто стратегический уровень управления называют *стратегическим*, или *долгосрочным, планированием*. Правильность принятого на этом уровне решения может быть подтверждена спустя достаточно длительное время. Могут пройти месяцы или годы. Ответственность за принятие управленческих решений чрезвычайно велика и определяется не только результатами анализа с использованием математического и специального аппарата, но и профессиональной интуицией менеджеров.

Пример. На основании анализа финансового состояния фирмы принимаются решения об увеличении (уменьшении, снятии с продажи) выпуска продукции, о привлечении дополнительных работников или об их сокращении.

5.1.6. Персонал и прочие элементы организации

Персонал организации — это сотрудники разной степени квалификации и уровней управления, начиная от секретарей, выполняющих простейшие типовые операции обработки, и заканчивая специалистами и менеджерами, принимающими стратегические решения. На рис. 5.3 показано соответствие разных уровней квалификации персонала уровням управления.



Рис. 5.3. Квалификация персонала по уровням управления

- На верхнем, стратегическом, уровне управления находятся менеджеры высшего звена руководства организации (глава фирмы и его заместители). Основная их задача — стратегическое планирование деятельности фирмы на рынке и координация внутрифирменной тактики управления.

- На среднем, функциональном, уровне находятся менеджеры среднего звена и специалисты (начальники служб, отделов, цехов, начальник смены, участка, научные сотрудники и т. п.). Их основная задача — тактическое управление фирмой при выполнении основных функций в заданной сфере деятельности.
- На нижнем, операционном, уровне находятся исполнители и менеджеры низшего звена (бригадиры, инженеры, ответственные исполнители, мастера, нормировщики, техники, лаборанты и т. п.). Их основная задача — оперативное реагирование на изменение ситуации.

На всех уровнях управления работают как менеджеры, осуществляющие только общие функции, так и менеджеры-специалисты, которые реализуют функции управления в сфере своей компетенции.

Пример. Главный инженер организации (менеджер-специалист) передал часть своих функций менеджерам среднего уровня, например, главному энергетiku, главному механику, главному электрику, оставив за собой общие функции управления этими службами, не вмешиваясь в их деятельность на оперативном уровне.

Стандартные процедуры (регламенты) в организации — это точно определенные правила выполнения заданий в различных ситуациях. Они охватывают все стороны функционирования организации, начиная от технологических операций по составлению документов на производимую продукцию и заканчивая разбором жалоб потребителей.

Субкультура любой организации — совокупность представлений, принципов, типов поведения. Особую роль играет важная ее составляющая — информационная культура специалиста. Это также должно найти отражение в информационной системе.

Пример. В фирме, предоставляющей туристические услуги, принято следующее правило — клиент обслуживается в порядке очередности. Значит, и информационная система должна обрабатывать и выдавать информацию, анализируя время поступления заявки клиента.

Существует взаимозависимость между стратегией, правилами, процедурами организации и аппаратной, программной, телекоммуникационной частями информационной системы. Поэтому очень важно на этапе внедрения и проектирования информационных систем активное участие менеджеров, определяющих предлагаемый круг проблем, задач и функций своей предметной области.

5.1.7. Примеры информационных систем

Информационные системы по отысканию рыночных ниш. При покупке товаров в некоторых фирмах информационная система регистрирует данные о покупателе, что позволяет:

- определять группы покупателей, их состав и запросы, а затем ориентироваться в своей стратегии на наиболее многочисленную группу;
- посылать потенциальным покупателям различные предложения, рекламу, напоминания;
- предоставлять постоянным покупателям товары и услуги в кредит, со скидкой, с отсрочкой платежей.

Информационные системы, ускоряющие потоки товаров. Предположим, фирма специализируется на поставках продуктов в определенное учреждение, например в больницу. Как известно, иметь большие запасы продуктов на складах фирмы очень невыгодно, а не иметь их невозможно. Для того чтобы найти оптимальное решение этой проблемы, фирма устанавливает терминалы в обслуживаемом учреждении и подключает их к информационной системе. Заказчик прямо с терминала вводит свои пожелания по предоставляемому ему каталогу. Эти данные поступают в информационную систему по учету заказов.

Менеджеры, делая выборки по поступившим заказам, принимают оперативные управленческие решения по доставке заказчику нужного товара за короткий промежуток времени. Таким образом экономятся огромные деньги на хранение товаров, ускоряется и упрощается регулирование потока товаров, отслеживаются потребности покупателей.

Информационные системы по снижению издержек производства. Эти информационные системы, отслеживая все фазы производственного процесса, способствуют улучшению управления и контроля, более рациональному планированию и использованию персонала и, как следствие, снижению себестоимости производимой продукции и предоставляемых услуг.

Пример. Информационная система, установленная в фирме по сдаче автомашин внаем, отслеживает местонахождение, стоимость и техническое состояние парка прокатных машин. Это позволяет минимизировать потери от простоя и пустого прогона каждой автомашины, перераспределяя предложения согласно спросу.

5.2. Структура и классификация информационных систем

5.2.1. Структура информационной системы

Структуру информационной системы составляет совокупность отдельных ее частей, называемых подсистемами.

Общую структуру информационной системы можно рассматривать как совокупность подсистем независимо от сферы применения. В этом случае говорят о *структурном признаке классификации*, а подсистемы называют *обеспечивающими*.

Таким образом, структура любой информационной системы может быть представлена совокупностью обеспечивающих подсистем (рис. 5.4).



Рис. 5.4. Структура информационной системы как совокупности обеспечивающих подсистем

Обеспечивающие подсистемы — это совокупность необходимых для функционирования информационной системы подсистем.

Среди обеспечивающих подсистем обычно выделяют информационное, техническое, математическое, программное, организационное и правовое обеспечение.

Информационное обеспечение

Информационное обеспечение — это совокупность единой системы классификации и кодирования информации, унифицированных систем документации, схем информационных потоков, циркулирующих в организации, а также методологии построения баз данных.

Назначение подсистемы информационного обеспечения состоит в своевременном формировании и выдаче достоверной информации для принятия управленческих решений.

Унифицированные системы документации создаются на государственном, республиканском, отраслевом и региональном уровнях. Главная цель — обеспечение сопоставимости показателей различных сфер общественного производства. Разработаны стандарты, где устанавливаются требования:

- ☐ к унифицированным системам документации;
- ☐ к унифицированным формам документов различных уровней управления;
- ☐ к составу и структуре реквизитов и показателей;
- ☐ к порядку внедрения, ведения и регистрации унифицированных форм документов.

Однако несмотря на существование унифицированной системы документации, при обследовании большинства организаций постоянно выявляется целый комплекс типичных недостатков:

- ☐ чрезвычайно большой объем документов для ручной обработки;
- ☐ одни и те же показатели часто дублируются в разных документах;
- ☐ работа с большим количеством документов отвлекает специалистов от решения непосредственных задач;
- ☐ имеются показатели, которые создаются, но не используются, и др.

Устранение указанных недостатков является одной из задач, требующих решения при создании информационного обеспечения.

Схемы информационных потоков отражают маршруты движения информации и ее объемы, места возникновения первичной и места использования результатной информации. За счет анализа структуры подобных схем можно выработать меры по совершенствованию всей системы управления.

Пример. В качестве примера простейшей схемы потоков данных можно привести схему, где отражены все этапы прохождения служебной записки или записи в базе данных о приеме на работу сотрудника — от момента ее создания до выхода приказа о его зачислении на работу.

Построение схем информационных потоков, позволяющих выявить объемы информации и провести ее детальный анализ, обеспечивает:

- исключение дублирующей и неиспользуемой информации;
- классификацию и рациональное представление информации.

При этом подробно должны рассматриваться вопросы взаимосвязи движения информации по уровням управления (см. рис. 3.2). Следует выявить, какие показатели необходимы для принятия управленческих решений, а какие нет. К каждому исполнителю должна поступать только та информация, которая используется.

Методология построения баз данных базируется на теоретических основах их проектирования. Для понимания концепции методологии приведем основные ее идеи в виде двух последовательно реализуемых на практике этапов.

1. Обследование всех функциональных подразделений организации с целью:
 - понять специфику и структуру ее деятельности;
 - построить схему информационных потоков;
 - проанализировать существующую систему документооборота;
 - определить информационные объекты и соответствующий состав реквизитов (параметров, характеристик), описывающих их свойства и назначение.
2. Построение концептуальной информационно-логической модели данных для обследованной на первом этапе сферы деятельности организации. В этой модели должны быть установлены и оптимизированы все связи между объектами и их реквизитами. Информационно-логическая модель является фундаментом, на котором будет создана база данных.

Для создания информационного обеспечения необходимо:

- ясное понимание целей, задач, функций всей системы управления организацией;
- выявление движения информации от момента возникновения и до ее использования на различных уровнях управления, представленной для анализа в виде схем информационных потоков;
- совершенствование системы документооборота;
- наличие и использование системы классификации и кодирования;

- владение методологией создания концептуальных информационно-логических моделей, отражающих взаимосвязь информации;
- создание массивов информации на машинных носителях, что требует наличия современного технического обеспечения.

Техническое обеспечение

Техническое обеспечение — это комплекс технических средств, предназначенных для работы информационной системы, а также соответствующая документация на эти средства и технологические процессы.

Комплекс технических средств составляют:

- компьютеры любых моделей;
- устройства сбора, накопления, обработки, передачи и вывода информации;
- устройства передачи данных и линии связи;
- оргтехника и устройства автоматического съема информации;
- эксплуатационные материалы и др.

Документацией оформляются предварительный выбор технических средств, организация их эксплуатации, технологический процесс обработки данных, технологическое оснащение. Документацию можно условно разделить на три группы:

- общесистемная, включающая государственные и отраслевые стандарты по техническому обеспечению;
- специализированная, содержащая комплекс методик по всем этапам разработки технического обеспечения;
- нормативно-справочная, используемая при выполнении расчетов по техническому обеспечению.

К настоящему времени сложились две основные формы организации технического обеспечения (формы использования технических средств): централизованная и частично или полностью децентрализованная.

Централизованное техническое обеспечение базируется на использовании в информационной системе больших ЭВМ и вычислительных центров.

Децентрализация технических средств предполагает реализацию функциональных подсистем на персональных компьютерах непосредственно на рабочих местах.

Перспективным подходом следует считать, по-видимому, *частично децентрализованный* подход — организацию технического обеспечения на базе распределенных сетей, состоящих из персональных компьютеров и большой ЭВМ для хранения баз данных, общих для любых функциональных подсистем.

Математическое и программное обеспечение

Математическое и программное обеспечение — это совокупность математических методов, моделей, алгоритмов и программ, необходимых для достижения целей и решения задач информационной системы, а также для нормального функционирования комплекса технических средств.

К средствам математического обеспечения относятся:

- средства моделирования процессов управления;
- готовые решения типовых задач управления;
- методы математического программирования, математической статистики, теории массового обслуживания и др.

В состав программного обеспечения входят общесистемные и специальные программные продукты, а также техническая документация.

К общесистемному программному обеспечению относятся комплексы программ, ориентированных на пользователей и предназначенных для решения типовых задач обработки информации. Они служат для расширения функциональных возможностей компьютеров, контроля и управления процессом обработки данных.

Специальное программное обеспечение представляет собой совокупность программ, разработанных при создании конкретной информационной системы. В его состав входят пакеты прикладных программ (ППП), реализующие разработанные модели разной степени адекватности, отражающие функционирование реального объекта.

Техническая документация на разработку программных средств должна содержать описание задач, задание на алгоритмизацию, экономико-математическую модель задач, контрольные примеры.

Организационное обеспечение

Организационное обеспечение — это совокупность методов и средств, регламентирующих взаимодействие работников с техническими средствами и между собой в процессе разработки и эксплуатации информационной системы.

Организационное обеспечение реализует следующие функции:

- анализ существующей системы управления организацией, где будет использоваться ИС, и выявление задач, подлежащих автоматизации;
- подготовка задач к решению на компьютере, включая техническое задание на проектирование ИС и технико-экономическое обоснование ее эффективности;
- разработка управленческих решений по составу и структуре организации, методологии решения задач, направленных на повышение эффективности системы управления.

Организационное обеспечение создается по результатам предпроектного обследования на первом этапе построения баз данных, с целями которого вы познакомились при рассмотрении информационного обеспечения.

Правовое обеспечение

Правовое обеспечение — это совокупность правовых норм, определяющих создание, юридический статус и функционирование информационных систем, регламентирующих порядок получения, преобразования и использования информации.

Главной целью правового обеспечения является укрепление законности.

В состав правового обеспечения входят законы, указы, постановления государственных органов власти, приказы, инструкции и другие нормативные документы министерств, ведомств, организаций, местных органов власти. В правовом обеспечении можно выделить общую часть, регулирующую функционирование любой информационной системы, и локальную часть, регулирующую функционирование конкретной системы.

Правовое обеспечение этапов разработки информационной системы включает в себя нормативные акты, связанные с договорными отношениями разработчика и заказчика и правовым регулированием отклонений от договора.

Правовое обеспечение этапов функционирования информационной системы включает в себя:

- статус информационной системы;
- права, обязанности и ответственность персонала;
- правовые положения отдельных видов процесса управления;
- порядок создания и использования информации и др.

5.2.2. Классификация информационных систем по признаку структурированности задач

Понятие структурированности задач

При создании или при классификации информационных систем неизбежно возникают проблемы, связанные с формальным — математическим и алгоритмическим — описанием решаемых задач. От степени формализации во многом зависят эффективность работы всей системы и уровень автоматизации, определяемый степенью участия человека в принятии решения на основе получаемой информации.

Чем точнее математическое описание задачи, тем шире возможности компьютерной обработки данных и тем меньше степень участия человека в процессе ее решения. Это и определяет степень автоматизации задачи.

Различают *три типа задач*, для которых создаются информационные системы: структурированные (формализуемые), неструктурированные (неформализуемые) и частично структурированные.

Структурированная (формализуемая) задача — это задача, где известны все ее элементы и взаимосвязи между ними.

Неструктурированная (неформализуемая) задача — это задача, в которой невозможно выделить элементы и установить между ними связи.

В *структурированной задаче* удастся выразить ее содержание в форме математической модели, имеющей точный алгоритм решения. Подобные задачи обычно приходится решать многократно, и они носят рутинный характер. Целью использования информационной системы для решения структурированных задач является полная автоматизация их решения, то есть сведение роли человека к нулю.

Решение *неструктурированных задач* из-за невозможности создания математического описания и разработки алгоритма связано с большими трудностями. Возможности использования здесь информационной системы невелики. Решение в таких случаях принимается человеком из эвристических соображений на основе своего опыта и, возможно, косвенной информации из разных источников.

Пример. Попробуйте формализовать взаимоотношения в вашей студенческой группе. Наверное, вряд ли вы сможете это сделать. Это связано с тем, что для данной задачи существенны психологический и социальный факторы, которые очень сложно описать алгоритмически.

Заметим, что в практике работы любой организации существует сравнительно немного полностью структурированных или совершенно неструктурированных задач. О большинстве задач можно сказать, что известна лишь часть их элементов и связей между ними. Такие задачи называются *частично структурированными*. В этих условиях можно создать информационную систему. Получаемая в ней информация анализируется человеком, который играет определяющую роль. Такие информационные системы являются автоматизированными, так как в их функционировании принимает участие человек.

Типы информационных систем для решения частично структурированных задач

Информационные системы, используемые для решения частично структурированных задач, подразделяются на два вида (рис. 5.5):

- Информационные системы, создающие управленческие отчеты и ориентированные главным образом на обработку данных (поиск, сортировку, агрегирование, фильтрацию). Используя сведения, содержащиеся в этих отчетах, управляющий принимает решение.
- Информационные системы, разрабатывающие возможные альтернативы решения. Принятие решения при этом сводится к выбору одной из предложенных альтернатив.

Информационные системы, создающие *управленческие отчеты*, обеспечивают информационную поддержку пользователя, то есть предоставляют доступ к информации в базе данных и ее частичную обработку. Процедуры манипулирования данными в информационной системе должны обеспечивать следующие возможности:

- составление комбинаций данных, получаемых из различных источников;
- быстрое добавление или исключение того или иного источника данных и автоматическое переключение источников при поиске данных;
- управление данными с использованием возможностей систем управления базами данных;
- логическая независимость данных этого типа от других баз данных, входящих в подсистему информационного обеспечения;
- автоматическое отслеживание потока информации для наполнения баз данных.

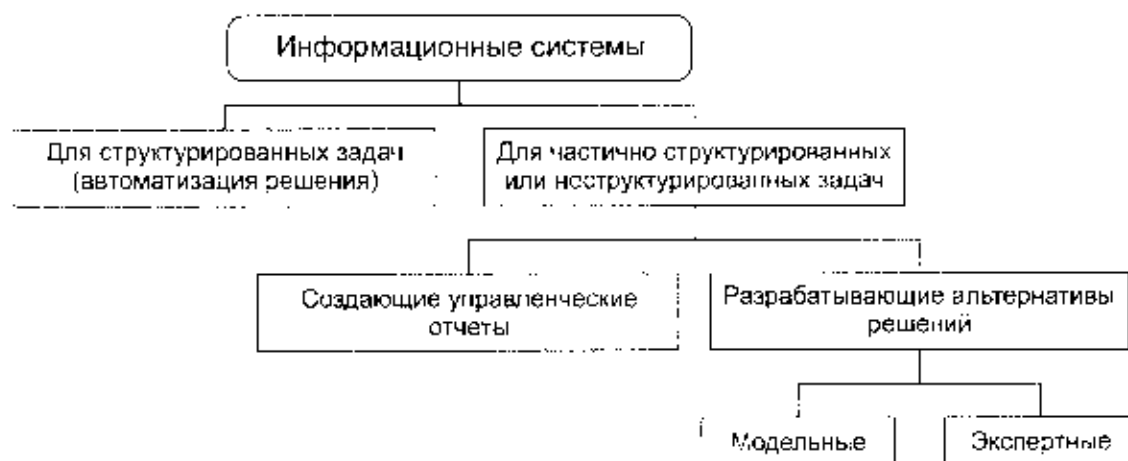


Рис. 5.5. Классификация информационных систем по признаку структурированности решаемых задач

Информационные системы, разрабатывающие *альтернативы решений*, могут быть модельными или экспертными.

Модельные информационные системы предоставляют пользователю математические, статистические, финансовые и другие модели, применение которых облегчает выработку и оценку альтернатив решения. Пользователь может получить недостающую ему для принятия решения информацию путем диалога с моделью в процессе ее исследования.

Основными функциями модельной информационной системы являются:

- возможность работы в среде типовых математических моделей, включая решение основных задач моделирования типа «как сделать, чтобы?», «что будет, если?», анализ чувствительности и др.;
- достаточно быстрая и адекватная интерпретация результатов моделирования;
- оперативная подготовка и корректировка входных параметров и ограничений модели;
- возможность графического отображения динамики модели;
- возможность объяснения пользователю необходимых шагов формирования и работы модели.

Экспертные информационные системы обеспечивают выработку и оценку возможных альтернатив пользователем за счет создания экспертных систем, связанных с обработкой знаний. Экспертная поддержка принимаемых пользователем решений реализуется на двух уровнях.

Работа первого уровня экспертной поддержки исходит из концепции «типовых управленческих решений», в соответствии с которой часто возникающие в процессе управления проблемные ситуации можно свести к некоторым однородным классам управленческих решений, то есть к некоторому типовому набору альтернатив. Для реализации экспертной поддержки на этом уровне создается информационный фонд хранения и анализа типовых альтернатив.

Если возникающая проблемная ситуация не ассоциируется с имеющимися классами типовых альтернатив, в работу должен вступать второй уровень экспертной поддержки управленческих решений. Этот уровень генерирует альтернативы на

базе имеющихся в информационном фонде данных, правил преобразования и процедур оценки синтезированных альтернатив.

5.2.3. Классификация информационных систем по функциональному признаку и уровням управления

Понятие функционального признака

Функциональный признак определяет назначение подсистемы, а также ее основные цели, задачи и функции. Структура информационной системы может быть представлена как совокупность ее функциональных подсистем, а функциональный признак может быть использован при классификации информационных систем.

В хозяйственной практике производственных и коммерческих объектов типовыми видами деятельности, которые определяют функциональный признак классификации информационных систем, являются: производственная, маркетинговая, финансовая, кадровая.

Производственная деятельность связана с непосредственным выпуском продукции и направлена на создание и внедрение в производство научно-технических новшеств.

Маркетинговая деятельность включает в себя:

- анализ рынка производителей и потребителей выпускаемой продукции, анализ продаж;
- организацию рекламной кампании по продвижению продукции;
- рациональную организацию материально-технического снабжения.

Финансовая деятельность связана с организацией контроля и анализа финансовых ресурсов фирмы на основе бухгалтерской, статистической, оперативной информации.

Кадровая деятельность направлена на подбор и расстановку необходимых фирме специалистов, а также ведение служебной документации по различным аспектам.

Указанные направления деятельности определили типовой набор информационных систем:

- производственные системы;
- системы маркетинга;
- финансовые и учетные системы;
- системы кадров (человеческих ресурсов);
- прочие типы, выполняющие вспомогательные функции в зависимости от специфики деятельности фирмы.

В крупных фирмах основная информационная система функционального назначения может состоять из нескольких подсистем для выполнения подфункций. Например, производственная информационная система имеет следующие подсистемы: управления запасами, управления производственным процессом, компьютерного инжиниринга и т. д.

Для лучшего понимания функционального назначения информационных систем в табл. 5.2 приведены типовые задачи, решаемые по каждому рассмотренному виду информационных систем.

Таблица 5.2. Функции информационных систем

Система маркетинга	Производственные системы	Финансовые и учетные системы	Система кадров (человеческих ресурсов)	Прочие системы, например ИС руководства
Исследование рынка и прогнозирование продаж	Планирование объемов работ и разработка календарных планов	Управление портфелем заказов	Анализ и прогнозирование потребности в трудовых ресурсах	Контроль за деятельностью фирмы
Управление продажами	Оперативный контроль и управление производством	Управление кредитной политикой	Ведение архивов записей о персонале	Выявление оперативных проблем
Рекомендации по производству новой продукции	Анализ работы оборудования	Разработка финансового плана	Анализ и планирование подготовки кадров	Анализ управленческих и стратегических ситуаций
Анализ и установление цены	Участие в формировании заказов поставщикам	Финансовый анализ и прогнозирование		Обеспечение процесса выработки стратегических решений
Учет заказов	Управление запасами	Контроль бюджета, бухгалтерский учет и расчет зарплаты		

Типы информационных систем

Тип информационной системы зависит от того, чьи интересы она обслуживает и на каком уровне управления.

На рис. 5.6 показан один из возможных вариантов классификации информационных систем по функциональному признаку с учетом уровней управления и уровней квалификации персонала (см. рис. 5.2 и 5.3).

Из рисунка видно, что чем выше по значимости уровень управления, тем меньше объем работ, выполняемых специалистом и менеджером с помощью информационной системы. Однако при этом возрастают сложность и интеллектуальные возможности информационной системы, ее роль в принятии менеджером решений. Любой уровень управления нуждается в информации из всех функциональных систем, но в разных объемах и с разной степенью обобщения.

Основание пирамиды составляют информационные системы, с помощью которых сотрудники-исполнители занимаются операционной обработкой данных, а менеджеры высшего звена — оперативным управлением. Наверху пирамиды на уровне стратегического управления информационные системы изменяют свою роль и становятся стратегическими, поддерживающими деятельность менеджеров высшего звена по принятию решений в условиях плохой структурированности поставленных задач.

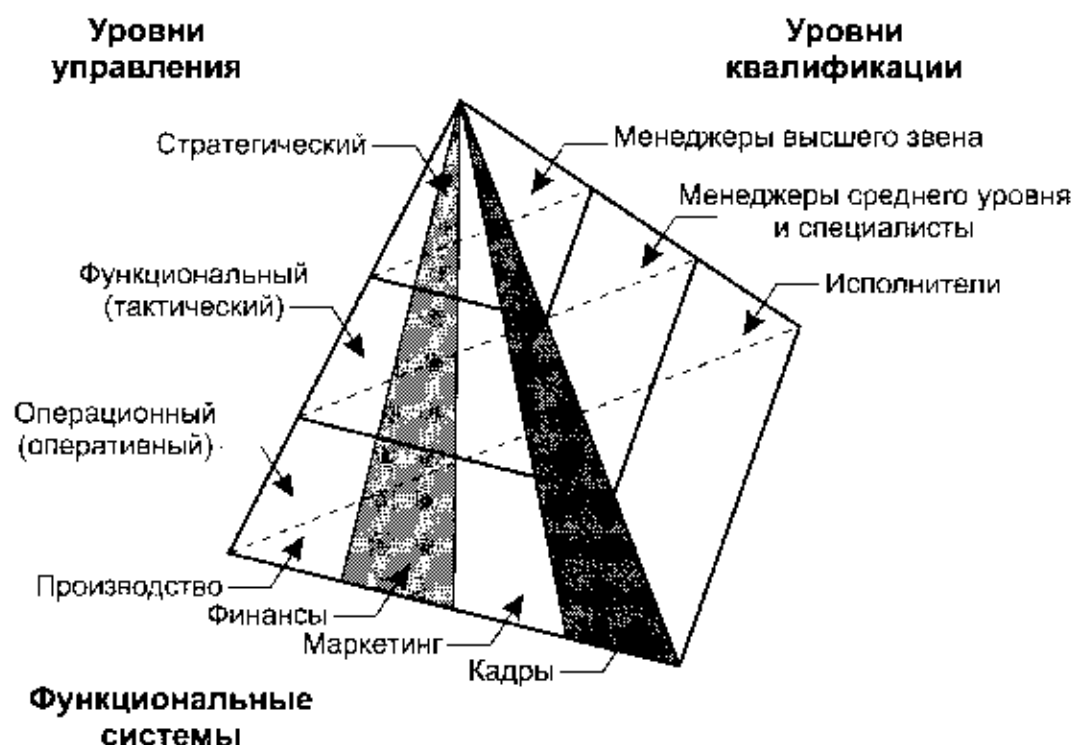


Рис. 5.6. Типы информационных систем в зависимости от функционального признака с учетом уровней управления и квалификации персонала

Информационные системы операционного (оперативного) уровня

Информационная система операционного уровня поддерживает специалистов-исполнителей, обрабатывая данные о сделках и событиях (счета, накладные, зарплата, кредиты, поток сырья и материалов). Назначение ИС на этом уровне — отвечать на запросы о текущем состоянии и отслеживать поток сделок в фирме, что соответствует оперативному управлению. Чтобы с этим справиться, информационная система должна быть легкодоступной, непрерывно действующей и предоставлять точную информацию.

Задачи, цели и источники информации на операционном уровне заранее определены и в высокой степени структурированы. Решение запрограммировано в соответствии с заданным алгоритмом.

Информационная система операционного уровня является связующим звеном между фирмой и внешней средой. Если система работает плохо, то организация либо не получает информации извне, либо не выдает информацию. Кроме того, система — это основной поставщик информации для остальных типов информационных систем в организации, так как содержит и оперативную, и архивную информацию.

Отключение этой ИС привело бы к необратимым негативным последствиям.

Пример. Информационные системы операционного уровня:

- банковских депозитов;
- обработки заказов;
- регистрации авиабилетов;
- выплаты зарплат и т. д.

Информационные системы для специалистов

Информационные системы этого уровня помогают специалистам, работающим с данными, повышают продуктивность и производительность работы инженеров и проектировщиков. Задача подобных информационных систем — интеграция новых сведений в организацию и помощь в обработке бумажных документов.

В этом классе информационных систем можно выделить две группы:

- информационные системы офисной автоматизации;
- информационные системы обработки знаний.

Информационные системы *офисной автоматизации* вследствие своей простоты и многопрофильности активно используются работниками любого организационного уровня. Наиболее часто их применяют работники средней квалификации: бухгалтеры, секретари, клерки. Основная цель — обработка данных, повышение эффективности работы, упрощение канцелярского труда.

ИС офисной автоматизации связывают воедино работников информационной сферы в разных регионах и помогают поддерживать связь с покупателями, заказчиками и другими организациями. Их деятельность в основном охватывает управление документацией, коммуникации, составление расписаний и т. д. Эти системы выполняют следующие функции:

- обработка текстов на компьютерах с помощью различных текстовых процессоров;
- производство высококачественной печатной продукции;
- архивация документов;
- ведение деловой информации с использованием электронных календарей и электронных записных книжек;
- поддержка электронной и аудиопочты;
- ведение видео- и телеконференций.

Информационные системы *обработки знаний*, в том числе экспертные системы, вбирают в себя знания, необходимые инженерам, юристам, ученым при разработке или создании нового продукта. Их работа заключается в создании новой информации и нового знания. Так, например, существующие специализированные рабочие станции по инженерному и научному проектированию позволяют обеспечить высокий уровень технических разработок.

Информационные системы для менеджеров среднего звена

Информационные системы уровня менеджмента используются работниками среднего управленческого звена для мониторинга (постоянного слежения), контроля, принятия решений и администрирования. Основные функции этих информационных систем:

- сравнение текущих показателей с прошлыми;
- составление периодических отчетов за определенное время, а не выдача отчетов по текущим событиям, как на оперативном уровне;
- обеспечение доступа к архивной информации и т. д.

На этом уровне можно выделить два типа информационных систем: управленческие (для менеджмента) и системы поддержки принятия решений.

Управленческие ИС имеют крайне небольшие аналитические возможности. Они обслуживают управленцев, которые нуждаются в ежедневной, еженедельной информации о состоянии дел. Основное их назначение состоит в отслеживании ежедневных операций в фирме и периодическом формировании строго структурированных сводных типовых отчетов. Информация поступает из информационной системы операционного уровня.

Характеристики управленческих информационных систем:

- решение структурированных и частично структурированных задач на уровне контроля за операциями;
- ориентация на контроль, отчетность и принятие решений по оперативной обстановке;
- опора на существующие данные и их потоки внутри организации;
- слабые аналитические возможности и негибкая структура.

Системы поддержки принятия решений обслуживают частично структурированные задачи, результаты которых трудно спрогнозировать заранее. Они имеют более мощный аналитический аппарат с несколькими моделями. Информацию получают из управленческих и операционных информационных систем. Используют эти системы все, кому необходимо принимать решение: менеджеры, специалисты, аналитики и пр. Например, их рекомендации могут пригодиться при принятии решения покупать или взять оборудование в аренду и пр.

Характеристики систем поддержки принятия решений:

- ориентация на решение проблем, развитие которых трудно прогнозировать;
- оснащение сложными инструментальными средствами моделирования и анализа;
- простота смены решаемых задач и входных данных;
- гибкость и простота адаптации к частым изменениям условий;
- максимальная ориентация на пользователя.

Стратегические информационные системы

Развитие и успех любой организации (фирмы) во многом определяются принятой в ней стратегией. Под *стратегией* понимается набор методов и средств решения перспективных долгосрочных задач.

Необходимость выработки и реализации стратегии фирмы привела к появлению нового типа информационных систем — стратегических.

Стратегическая информационная система — компьютерная информационная система, обеспечивающая поддержку принятия решений по реализации стратегических перспективных целей развития организации.

Известны ситуации, когда новое качество информационных систем заставляло изменять не только структуру, но и профиль фирм, содействуя их процветанию.

Для принятия стратегических решений необходимо понимать взаимосвязь фирмы с ее внешним окружением. На рис. 5.7 показано воздействие на фирму внешних факторов:

- ❑ конкурентов, проводящих на рынке свою политику;
- ❑ покупателей, обладающих разными возможностями по приобретению товаров и услуг;
- ❑ поставщиков, которые проводят свою ценовую политику.

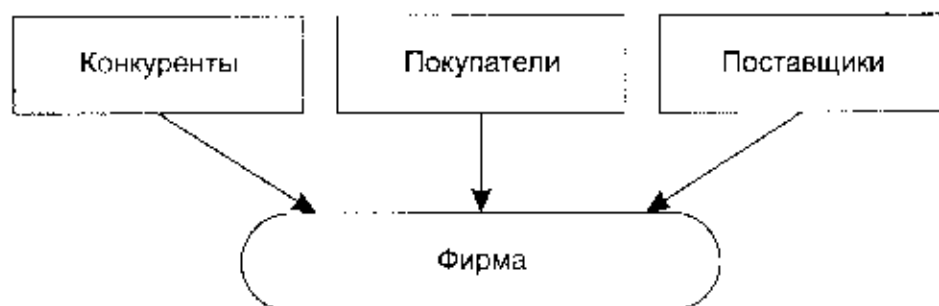


Рис. 5.7. Внешние факторы, воздействующие на деятельность фирмы

Фирма может обеспечить себе конкурентное преимущество, если будет учитывать эти факторы и придерживаться следующих стратегий:

- ❑ создание новых товаров и услуг, которые выгодно отличаются от аналогичных;
- ❑ отыскание рынков, где товары и услуги фирмы обладают рядом отличительных признаков по сравнению с уже имеющимися там аналогами;
- ❑ создание таких связей, которые закрепляют покупателей и поставщиков за данной фирмой и делают невыгодным обращение к другой;
- ❑ снижение стоимости продукции без ущерба качеству.

Информационные системы стратегического уровня помогают высшему звену управленцев решать неструктурированные задачи, подобные ранее описанным, осуществлять долгосрочное планирование. Основная задача — сравнение происходящих во внешнем окружении изменений с существующим потенциалом фирмы. Они призваны создать общую среду компьютерной и телекоммуникационной поддержки решений в неожиданно возникающих ситуациях.

Информационные системы в фирме

В любой фирме желательно иметь несколько локальных ИС разного назначения, которые взаимодействуют между собой и поддерживают управленческие решения на всех уровнях. На рис. 5.8 показан один из таких вариантов. Между локальными ИС организуются связи различного характера и назначения. Одни локальные ИС могут быть связаны с большим количеством работающих в фирме систем и иметь выход во внешнюю среду, другие — только с одной или несколькими родственными. Современный подход к организации связи основан на применении локальных внутрифирменных компьютерных сетей с выходом на аналогичную ИС другой фирмы или подразделения корпорации. При этом пользуются ресурсами региональных и глобальных сетей.

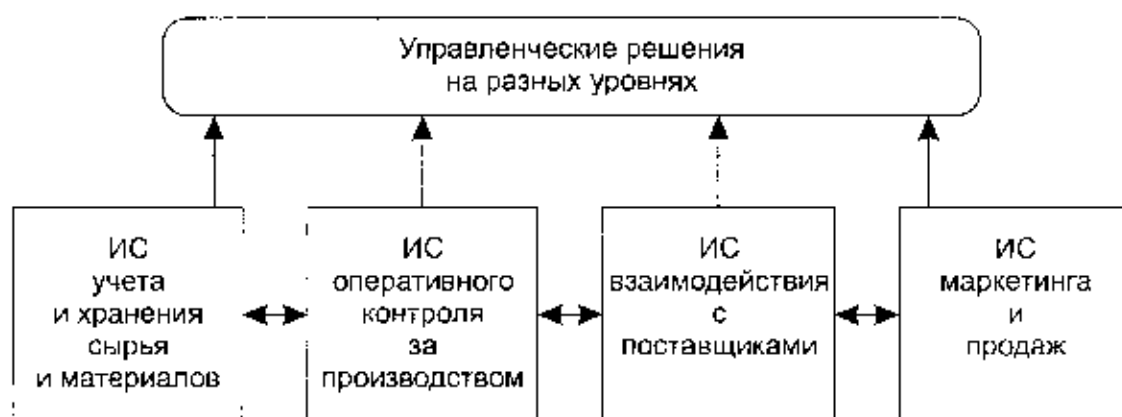


Рис. 5.8. Примеры информационных систем, поддерживающих деятельность фирмы

На основе интеграции ИС разного назначения с помощью компьютерных сетей в фирме создаются корпоративные ИС. Подобные ИС предоставляют пользователю возможность работать как с общефирменной базой данных, так и с локальными базами данных.

Рассмотрим роль в фирме корпоративной ИС при формировании стоимости выпускаемой продукции.

Информационные системы в фирме, поддерживая все стадии выпуска продукции, могут предоставлять информацию разной степени подробности для анализа, в результате которого выявляются этапы, где происходит сверхнормативное увеличение стоимости продукции. В этом случае может быть выбрана стратегия уменьшения стоимости продукции. Результаты принимаемых мер, в свою очередь, отразятся в информационной системе. Снова можно будет использовать полученную информацию для анализа. И так до тех пор, пока не будет достигнута поставленная цель.

Пример. Фирма может резко сократить издержки, связанные с хранением сырья и полуфабрикатов, договорившись с поставщиками о ежедневных поставках. Сведения о произведенных поставках учитываются информационной системой, которая предоставляет информацию для принятия решений на соответствующем уровне управления.

5.2.4. Прочие варианты классификации информационных систем

Классификация по степени автоматизации

В зависимости от степени автоматизации информационных процессов в системе управления фирмой информационные системы определяются как ручные, автоматические, автоматизированные (рис. 5.9).

Ручные ИС характеризуются отсутствием современных технических средств переработки информации и выполнением всех операций человеком. Например, о деятельности менеджера в фирме, где отсутствуют компьютеры, можно говорить, что он работает с ручной ИС.

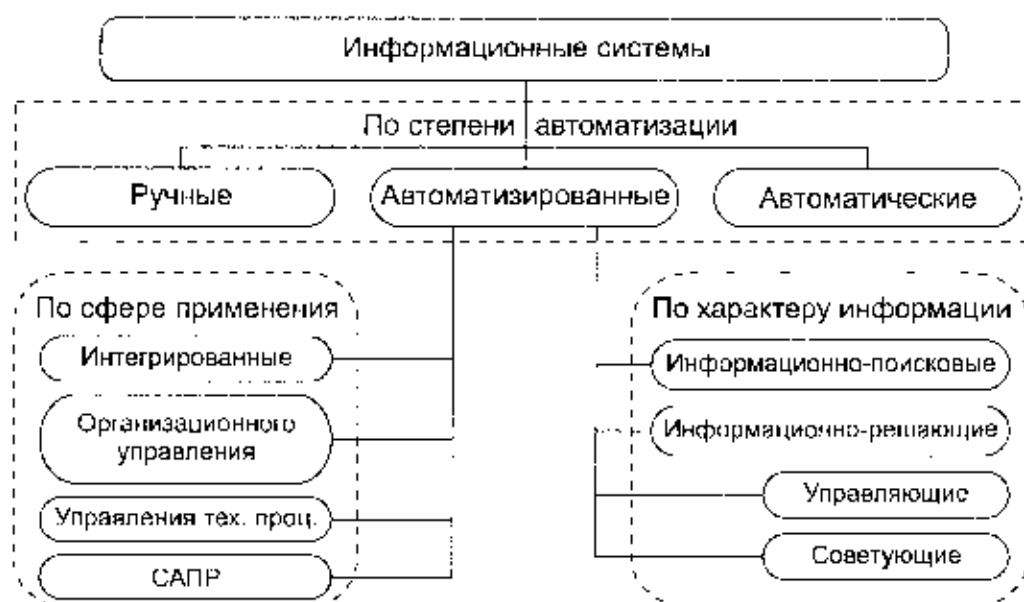


Рис. 5.9. Классификация информационных систем по разным признакам

Автоматические ИС выполняют все операции по переработке информации без участия человека.

Автоматизированные ИС предполагают участие в процессе обработки информации и человека, и технических средств, причем главная роль отводится компьютеру. В современном толковании в термин «информационная система» обязательно вкладывается понятие автоматизируемой системы.

Автоматизированные ИС, учитывая высокую степень их вовлеченности в организацию процессов управления, имеют различные модификации и могут быть классифицированы, например, по характеру использования информации и по сфере применения.

Пример. Роль бухгалтера в информационной системе по расчету заработной платы заключается в задании исходных данных. Информационная система обрабатывает их по заранее известному алгоритму с выдачей результатной информации в виде ведомости, напечатанной на принтере.

Классификация по характеру использования информации

Информационно-поисковые системы обеспечивают ввод, систематизацию, хранение и выдачу информации по запросу пользователя без сложных преобразований данных (см. рис. 5.9). Например, такие системы функционируют в библиотеках, в железнодорожных и авиакассах продажи билетов.

Информационно-решающие системы осуществляют все операции переработки информации по определенному алгоритму. Среди них можно провести классификацию по степени воздействия выработанной результатной информации на процесс принятия решений и выделить два класса: управляющие и советующие.

Управляющие ИС вырабатывают информацию, на основании которой человек принимает решение. Для этих систем характерны задачи расчетного характера и об-

работка больших объемов данных. Примером могут служить система оперативного планирования выпуска продукции, система бухгалтерского учета.

Советующие ИС вырабатывают информацию, которая принимается человеком к сведению и не превращается немедленно в серию конкретных действий. Эти системы обладают более высокой степенью интеллекта, так как для них характерна обработка знаний, а не данных.

Пример. Существуют медицинские информационные системы для постановки диагноза больного и определения предполагаемой процедуры лечения. Врач при работе с подобной системой может принять к сведению полученную информацию, но предложить иное по сравнению с рекомендуемым решение.

Классификация по сфере применения

Информационные системы организационного управления предназначены для автоматизации функций управленческого персонала (см. рис. 5.9). К этому классу относятся информационные системы управления как промышленными фирмами, так и непромышленными объектами: гостиницами, банками, торговыми фирмами и др.

Основными функциями подобных систем являются: оперативный контроль и регулирование, оперативный учет и анализ, перспективное и оперативное планирование, бухгалтерский учет, управление сбытом и снабжением и другие экономические и организационные задачи.

ИС управления технологическими процессами (ТП) служат для автоматизации функций производственного персонала. Они широко используются при организации поточных линий, изготовлении микросхем, сборке, поддержании технологического процесса в металлургической и машиностроительной промышленности.

ИС автоматизированного проектирования (САПР) предназначены для автоматизации функций инженеров-проектировщиков, конструкторов, архитекторов, дизайнеров при создании новой техники или технологии. Основными функциями подобных систем являются: инженерные расчеты, подготовка графической документации (чертежей, схем, планов), разработка проектной документации, моделирование проектируемых объектов.

Интегрированные (корпоративные) ИС служат для автоматизации всех функций фирмы и охватывают весь цикл работ от проектирования до сбыта продукции.

5.3. Основные сведения об информационных технологиях

5.3.1. Понятие информационной технологии

Технология в переводе с греческого (techné) означает искусство, мастерство, умение. Под *технологией материального производства* понимают совокупность средств и методов обработки, изготовления, изменения состояния, свойств, формы

сырья или материала. Процесс, определяемый технологией, изменяет качество или первоначальное состояние материи в целях получения материального продукта (рис. 5.10).

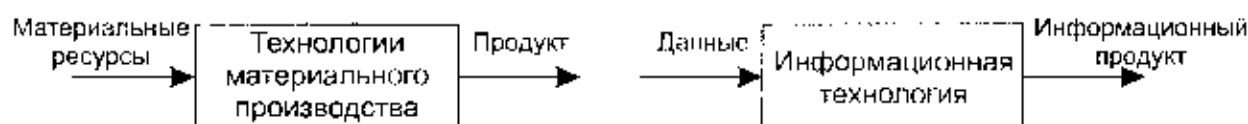


Рис. 5.10. Информационная технология как аналог технологии переработки материальных ресурсов

Информация является одним из ценнейших ресурсов общества наряду с традиционными видами материальных ресурсов, поэтому справедливо следующее определение:

Информационная технология — это совокупность средств и методов сбора, обработки и передачи данных (первичной информации) для получения информации нового качества о состоянии объекта, процесса или явления (информационного продукта).

Известно, что, применяя разные технологии к одному и тому же материальному ресурсу, можно получить разные изделия, продукты. То же самое справедливо и для технологии переработки информации.

Пример. Для выполнения контрольной работы по математике каждый студент применяет свою технологию переработки первоначальной информации (исходных данных задач). Информационный продукт (результаты решения задач) будет зависеть от технологии решения, которую выберет студент. Обычно используется ручная информационная технология. Если же воспользоваться компьютерной информационной технологией, способной решать подобные задачи, то информационный продукт приобретет уже иное качество.

Для сравнения в табл. 5.3 приведены основные компоненты обоих видов технологий.

Таблица 5.3. Сопоставление основных компонентов технологий для производства продуктов

Материальные компоненты	Информационные компоненты
Подготовка сырья и материалов	Сбор данных или первичной информации
Производство материального продукта	Обработка данных и получение результатной информации
Сбыт произведенных продуктов потребителям	Передача результатной информации пользователю для принятия на ее основе решений

5.3.2. Инструментарий информационной технологии

Реализация технологического процесса материального производства осуществляется с помощью различных технических средств, к которым относятся: оборудование, станки, инструменты, конвейерные линии и т. п.

По аналогии и для информационной технологии должно быть нечто подобное. Такими техническими средствами производства информации является аппаратное, программное и математическое обеспечение этого процесса. С их помощью производится переработка первичной информации в информацию нового качества. Выделим отдельно из этих средств программные продукты и назовем их инструментарием, а для большей четкости его можно конкретизировать, назвав программным инструментарием информационной технологии. Определим это понятие.

Инструментарий информационной технологии — это один или несколько взаимосвязанных программных продуктов для определенного типа компьютера, технология работы в котором позволяет достичь поставленную пользователем цель.

В качестве инструментария можно использовать следующие распространенные виды программных продуктов для персонального компьютера: текстовый процессор (редактор), настольные издательские системы, электронные таблицы, системы управления базами данных, электронные записные книжки, электронные календари, информационные системы функционального назначения (финансовые, бухгалтерские, для маркетинга и пр.), экспертные системы и т. д.

5.3.3. Соотношение между информационными технологиями и системами

Информационная технология тесно связана с информационными системами, которые являются для нее основной средой. На первый взгляд может показаться, что введенные в учебнике определения информационной технологии и системы очень похожи между собой. Однако это не так.

Информационная технология является совокупностью процессов и методик, состоящей из четко регламентированных правил выполнения операций, действий, этапов разной степени сложности над данными, хранящимися в компьютерах. Основная цель информационной технологии — в результате целенаправленных действий по переработке первичной информации получить необходимую для пользователя информацию.

Информационная система — это среда, составляющими элементами которой являются компьютеры, компьютерные сети, программные продукты, базы данных, люди, различного рода технические и программные средства связи и т. д. Основное назначение информационной системы — организация хранения и передачи информации. Информационная система представляет собой человеко-компьютерную систему обработки информации.

Реализация функций информационной системы невозможна без знания ориентированной на нее информационной технологии. Информационная технология может существовать и вне сферы информационной системы.

Пример. Информационная технология работы в среде текстового процессора Microsoft Word, который не является информационной системой.

Информационная технология мультимедиа, где с помощью телекоммуникационной связи осуществляются передача и обработка на компьютере изображения и звука.

Таким образом, информационная технология является более емким понятием, отражающим современное представление о процессах преобразования информации в информационном обществе. Обобщая все сказанное, предлагаем несколько более узкие, нежели введенные ранее, определения информационной системы и технологии, реализованных средствами компьютерной техники.

Информационная технология — это совокупность четко определенных целенаправленных действий персонала по переработке информации на компьютере.
Информационная система — это человек-компьютерная среда поддержки принятия решений и производства информационных продуктов, использующая компьютерную информационную технологию.

5.3.4. Составляющие информационной технологии

Используемые в производственной сфере такие технологические понятия, как «норма», «норматив», технологический процесс, технологическая операция и т. п., могут применяться и в информационной технологии. Прежде чем разрабатывать эти понятия в любой технологии, в том числе в информационной, всегда следует начинать с определения цели. Затем нужно попытаться провести структурирование всех предполагаемых действий, приводящих к намеченной цели, и выбрать необходимый программный инструментарий.

На рис. 5.11 технологический процесс переработки информации представлен в виде иерархической структуры по уровням.

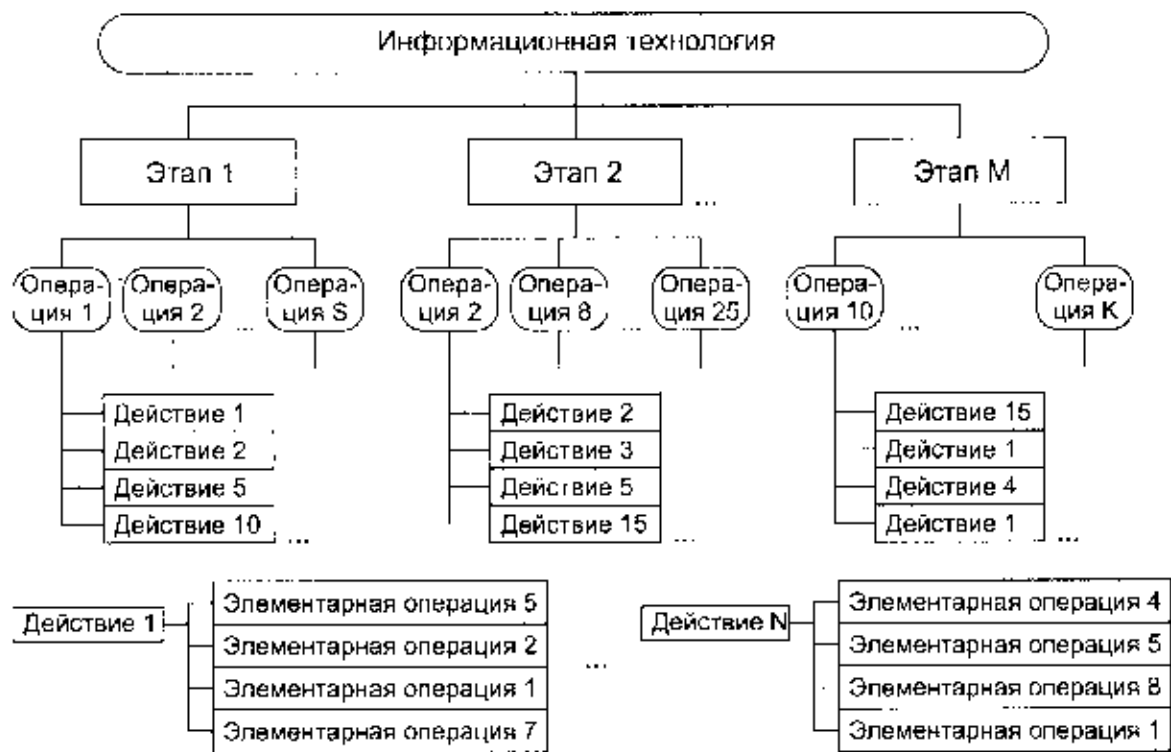


Рис. 5.11. Представление информационной технологии в виде иерархической структуры, состоящей из этапов, действий, операций

1-й уровень — *этапы*, на которых реализуются сравнительно длительные технологические процессы, состоящие из операций и действий последующих уровней.

2-й уровень — *операции*, в результате выполнения которых будет создан конкретный объект в выбранной на 1-м уровне программной среде.

3-й уровень — *действия* — совокупность стандартных для каждой программной среды приемов работы, приводящих к достижению поставленной в соответствующей операции цели. Каждое действие изменяет содержание экрана.

4-й уровень — *элементарные операции* по взаимодействию с устройствами ввода и вывода информации.

Пример. Технология создания шаблона формы документа в среде текстового процессора Microsoft Word состоит из нескольких этапов, в том числе этапа создания постоянной части формы в виде кадра, куда затем помещается рисунок. В свою очередь, этап создания кадра разбивается на несколько операций: создание кадра, настройка кадра, внедрение в кадр рисунка. Операция внедрения в кадр рисунка состоит из следующих действий: установка курсора в кадр, выполнение команды вставки, задание параметров. Элементарная операция подразумевает ввод команды, щелчок правой кнопки мыши, выбор пункта меню и т. п.

Информационная технология, как и любая другая, должна отвечать следующим требованиям:

- обеспечивать высокую степень расчленения всего процесса обработки информации на этапы (фазы), операции, действия;
- включать весь набор элементов, необходимых для достижения поставленной цели;
- иметь регулярный характер, то есть этапы, действия, операции технологического процесса должны быть стандартизированы и унифицированы, что позволит более эффективно осуществлять целенаправленное управление информационными процессами.

5.3.5. Использование информационных технологий

Централизованная обработка информации на ЭВМ вычислительных центров была первой исторически сложившейся технологией. Создавались крупные вычислительные центры (ВЦ) коллективного пользования, оснащенные большими ЭВМ (в нашей стране — ЭВМ ЕС). Применение таких ЭВМ позволяло обрабатывать большие массивы входной информации и получать на этой основе различные виды информационной продукции, которая затем передавалась пользователям. Такой технологический процесс был обусловлен недостаточным оснащением вычислительной техникой предприятий и организаций в 60–70-х гг.

Достоинства методологии централизованной технологии:

- возможность обращения пользователя к большим массивам информации в виде баз данных и к информационной продукции широкой номенклатуры;
- сравнительная легкость внедрения методологических решений по развитию и совершенствованию информационной технологии благодаря централизованному их принятию.

Недостатки такой методологии очевидны:

- ограниченная ответственность низшего персонала, который не способствует оперативному получению информации пользователем, тем самым препятствуя правильности выработки управленческих решений;
- ограничение возможностей пользователя в процессе получения и применения информации.

Децентрализованная обработка информации связана с появлением в 80-х гг. персональных компьютеров и развитием средств телекоммуникаций. Она весьма существенно потеснила предыдущую технологию, поскольку дает пользователю широкие возможности в работе с информацией и не ограничивает его инициатив.

Достоинствами такой методологии являются:

- гибкость структуры, обеспечивающая простор инициативам пользователя;
- усиление ответственности низшего звена сотрудников;
- уменьшение потребности в центральном компьютере и, соответственно, контроле со стороны вычислительного центра;
- более полная реализация творческого потенциала пользователя благодаря применению средств компьютерной связи.

Однако эта методология имеет свои недостатки:

- сложность стандартизации из-за большого числа уникальных разработок;
- психологическое неприятие пользователями рекомендуемых вычислительным центром стандартов и готовых программных продуктов;
- неравномерность развития уровня информационной технологии на локальных местах, что в первую очередь определяется уровнем квалификации конкретного работника.

Описанные достоинства и недостатки централизованной и децентрализованной информационных технологий привели к необходимости придерживаться линии разумного применения обоих подходов. Такой подход назовем *рациональной методологией*. Покажем, как в этом случае должны распределяться обязанности:

- вычислительный центр должен отвечать за выработку общей стратегии использования информационной технологии, помогать пользователям как в работе, так и в обучении, устанавливать стандарты и определять политику применения программных и технических средств;
- персонал, использующий информационную технологию, должен придерживаться указаний вычислительного центра, осуществлять разработку своих локальных систем и технологий в соответствии с общим планом организации.

Рациональная методология использования информационной технологии позволяет достичь большей гибкости, поддерживать общие стандарты, обеспечить совместимость информационных локальных продуктов, снизить дублирование деятельности и др.

5.4. Виды информационных технологий

5.4.1. Информационная технология обработки данных

Характеристика и назначение

Информационная технология обработки данных предназначена для решения хорошо структурированных задач, по которым имеются необходимые входные данные и известны алгоритмы и другие стандартные процедуры их обработки. Эта технология применяется на уровне операционной (исполнительской) деятельности (см. рис. 5.3) персонала невысокой квалификации в целях автоматизации некоторых рутинных постоянно повторяющихся операций управленческого труда. Поэтому внедрение информационных технологий и систем на этом уровне существенно повышает производительность труда персонала, освобождает его от рутинных операций, возможно, даже ведет к необходимости сокращения численности работников.

На уровне операционной деятельности решаются следующие задачи:

- обработка данных об операциях, производимых фирмой;
- создание периодических контрольных отчетов о состоянии дел в фирме;
- получение ответов на всевозможные текущие запросы и оформление их в виде бумажных документов или отчетов.

Пример. Примеры рутинных операций:

- операция проверки на соответствие нормативу уровня запасов указанных товаров на складе (при уменьшении уровня запаса выдается заказ поставщику с указанием потребного количества товара и сроков поставки);
- операция продажи товаров фирмой, в результате которой формируется выходной документ для покупателя в виде чека или квитанции.

Пример контрольного отчета: ежедневный отчет о поступлениях и выдачах наличных средств банком, формируемый в целях контроля баланса наличных средств.

Пример запроса: запрос к базе данных по кадрам, который позволяет получить данные о требованиях, предъявляемых к кандидатам на занятие определенной должности.

Существует несколько особенностей, связанных с обработкой данных, отличающих данную технологию от всех прочих:

- решение необходимых фирме задач по обработке данных. Каждой фирме предписано законом иметь и хранить данные о своей деятельности, которые можно использовать как средство обеспечения и поддержания контроля на фирме. Поэтому в любой фирме обязательно должна быть информационная система обработки данных и разработана соответствующая информационная технология;
- решение только хорошо структурированных задач, для которых можно разработать алгоритм;

- ❑ выполнение стандартных процедур обработки. Существующие стандарты определяют типовые процедуры обработки данных и предписывают их соблюдение организациям всех видов;
- ❑ выполнение основного объема работ в автоматическом режиме с минимальным участием человека;
- ❑ использование детализированных данных. Записи о деятельности фирмы имеют детальный (подробный) характер, допускающий проведение ревизий. В процессе ревизии деятельность фирмы проверяется хронологически от начала периода к его концу и от конца к началу;
- ❑ акцент на хронологию событий;
- ❑ требование минимальной помощи в решении проблем со стороны специалистов других уровней.

Основные компоненты

Представим основные компоненты информационной технологии обработки данных и приведем их характеристики (рис. 5.12).

- ❑ *Сбор данных.* Каждое действие фирмы фиксируется в виде соответствующих записей.
- ❑ *Обработка данных.* Для создания из поступающих данных информации, отражающей деятельность фирмы, используются следующие типовые операции:
 - классификация или группировка;
 - сортировка, с помощью которой упорядочивается последовательность записей;
 - вычисления, включающие в себя арифметические и логические операции;
 - укрупнение или агрегирование, служащее для уменьшения количества данных и реализуемое в форме расчетов итоговых или средних значений.
- ❑ *Хранение данных.* Для хранения данных создаются базы данных.
- ❑ *Создание отчетов (документов).* Отчеты предназначены для эффективного использования хранимых данных.

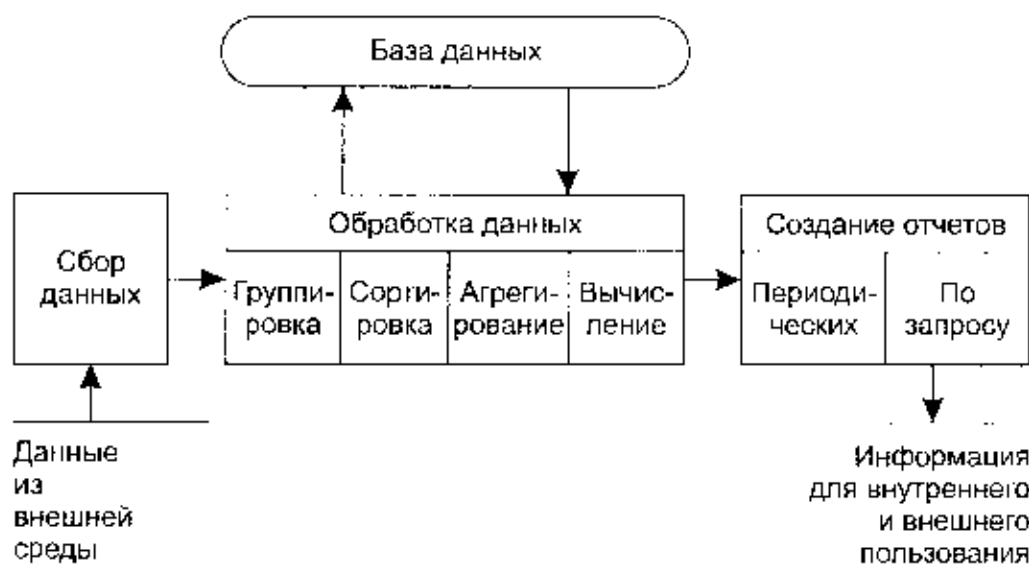


Рис. 5.12. Основные компоненты информационной технологии обработки данных

5.4.2. Информационная технология управления

Характеристика и назначение

Целью информационной технологии управления является удовлетворение информационных потребностей всех без исключения сотрудников фирмы, имеющих дело с принятием решений. Она может быть полезна на любом уровне управления.

Эта технология ориентирована на работу в среде информационной системы управления и используется при худшей структурированности решаемых задач, если их сравнивать с задачами, решаемыми с помощью информационной технологии обработки данных.

ИС управления идеально подходят для удовлетворения сходных информационных потребностей работников различных функциональных подсистем (подразделений) или уровней управления фирмой. Поставляемая ими информация содержит сведения о прошлом, настоящем и вероятном будущем фирмы. Эта информация имеет вид регулярных или специальных управленческих отчетов.

Для принятия решений на уровне управленческого контроля информация должна быть представлена в агрегированном виде так, чтобы просматривались тенденции изменения данных, причины возникших отклонений и возможные решения. На этом этапе решаются следующие задачи обработки данных:

- оценка планируемого состояния объекта управления;
- оценка отклонений от планируемого состояния;
- выявление причин отклонений;
- анализ возможных решений и действий.

Информационная технология управления направлена на создание различных видов отчетов.

Регулярные отчеты создаются в соответствии с установленным графиком, определяющим время их создания, например месячный анализ продаж компании.

Специальные отчеты создаются по запросам управленцев или при возникновении в компании каких-то незапланированных ситуаций.

Те и другие виды отчетов могут иметь форму суммирующих, сравнительных и чрезвычайных отчетов.

В *суммирующих* отчетах данные объединены в отдельные группы, отсортированы и представлены в виде промежуточных и окончательных итогов по отдельным полям.

Сравнительные отчеты содержат данные, полученные из различных источников или классифицированные по различным признакам, и используются для сравнения.

Чрезвычайные отчеты содержат данные исключительного (чрезвычайного) характера.

Использование отчетов с целью управления оказывается особенно эффективным при реализации так называемого управления по отклонениям.

Управление по отклонениям предполагает, что главным содержанием получаемых менеджером данных должны являться отклонения состояния хозяйственной деятельности фирмы от некоторых установленных стандартов (например,

от ее запланированного состояния). При использовании на фирме принципов управления по отклонениям к создаваемым отчетам предъявляются следующие требования:

- отчет должен создаваться только тогда, когда отклонение произошло;
- сведения в отчете должны быть отсортированы по значению критического для данного отклонения показателя;
- все отклонения желательно показать вместе, чтобы менеджер мог уловить существующую между ними связь;
- в отчете необходимо показать количественное отклонение от нормы.

Основные компоненты

Основные компоненты информационной технологии управления показаны на рис. 5.13.

Входная информация поступает из систем операционного уровня. Выходная информация формируется в виде управленческих отчетов в удобном для принятия решения виде.

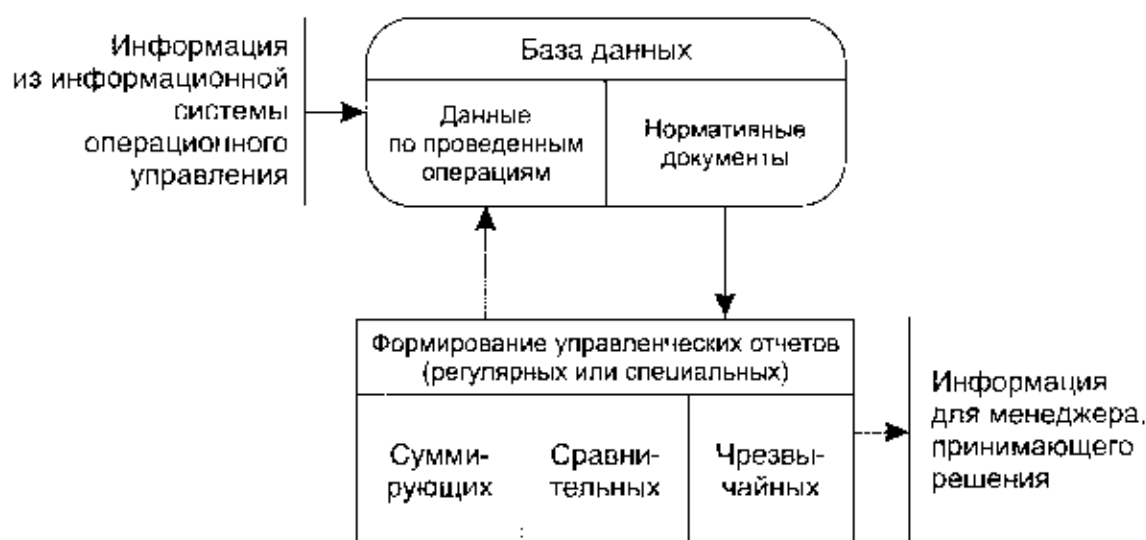


Рис. 5.13. Основные компоненты информационной технологии управления

Содержимое базы данных при помощи соответствующего программного обеспечения преобразуется в периодические и специальные отчеты, поступающие к специалистам, участвующим в принятии решений в организации. База данных, используемая для получения указанной информации, должна состоять из двух элементов:

- данных, накапливаемых на основе оценки операций, проводимых фирмой;
- планов, стандартов, бюджетов и других нормативных документов, определяющих планируемое состояние объекта управления (подразделения фирмы).

5.4.3. Автоматизация офиса

Характеристика и назначение

Исторически автоматизация началась на производстве и затем распространилась на офис, причем первоначально целью была лишь автоматизация рутинной секретарской работы. По мере развития средств коммуникаций автоматизация офисных технологий заинтересовала специалистов и управленцев, которые увидели в ней возможность повысить производительность своего труда.

Автоматизация офиса призвана не заменить существующую традиционную систему коммуникации персонала (с ее совещаниями, телефонными звонками и приказами), а лишь дополнить ее (рис. 5.14). При совместном использовании, обе эти системы должны быть нацелены на рациональную автоматизацию управленческого труда и наилучшее обеспечение управленцев информацией.

Автоматизированный офис привлекателен для менеджеров всех уровней управления в фирме не только потому, что поддерживает внутрифирменную связь персонала, но и потому, что предоставляет им новые средства коммуникации с внешним окружением.

Информационная технология автоматизированного офиса — организация и поддержка коммуникационных процессов как внутри организации, так и с внешней средой на базе компьютерных сетей и других современных средств передачи и обработки информации.



Рис. 5.14. Основные компоненты автоматизации офиса

Офисные автоматизированные технологии используются управленцами, специалистами, секретарями и конторскими служащими, особенно они привлекательны для группового решения проблем. Они позволяют повысить производительность труда секретарей и конторских работников и дают им возможность справляться с возрастающим объемом работ. Однако это преимущество является второстепенным по сравнению с возможностью решения многих проблем благодаря автоматизации офиса. Улучшение принимаемых менеджерами решений в результате их более совершенной коммуникации способно обеспечить экономический рост фирмы.

В настоящее время известно несколько десятков программных продуктов для компьютеров и некомпьютерных технических средств, обеспечивающих технологию автоматизации офиса: текстовый процессор, табличный процессор, электронная почта, электронный календарь, аудиопочта, компьютерные и телеконференции, видеотекст, хранение изображений, а также специализированные программы управленческой деятельности для ведения документов, контроля за исполнением приказов и т. д.

Также широко используются некомпьютерные средства: аудио- и видеоконференции, факсимильная связь, ксерокс и другие средства оргтехники.

Основные компоненты

Обязательным компонентом любой технологии является *база данных*. В автоматизированном офисе база данных концентрирует в себе данные о производственной системе фирмы так же, как в технологии обработки данных на операционном уровне. Информация в базу данных может также поступать из внешнего окружения фирмы. Специалисты должны владеть основными технологическими операциями по работе в среде баз данных.

Пример. В базе данных собираются сведения о ежедневных продажах, передаваемые торговыми агентами фирмы на главный компьютер, или сведения о еженедельных поставках сырья. В базу данных могут ежедневно по электронной почте поступать с биржи сведения о курсе валют или котировках ценных бумаг, в том числе и акций этой фирмы, которые ежедневно корректируются в соответствующем массиве базы данных.

Информация из базы данных поступает на вход компьютерных приложений (программ), таких как текстовый процессор, табличный процессор, электронная почта, компьютерные конференции и пр. Любое компьютерное приложение автоматизированного офиса обеспечивает работникам связь друг с другом и с другими фирмами.

Полученная из баз данных информация может быть использована и в некомпьютерных технических средствах для передачи, тиражирования, хранения.

Для создания и обработки текстовых документов предназначен *текстовый процессор*, а *электронная почта* (e-mail) дает возможность пользователю получать, хранить сообщения и отправлять их своим партнерам по сети.

Мультимедийная почта служит для передачи сообщений голосом в сочетании с движущимся (или статичным) сообщением и текстом.

Функции современных программных сред *табличных процессоров* позволяют выполнять многочисленные операции над данными, представленными в табличной форме, в том числе ввод, хранение, обработку (в том числе организация сложных вычислений и моделирования) и вывод информации в виде отчетов.

Электронный календарь (органайзер) предоставляет еще одну возможность использовать сетевой вариант компьютера для хранения и манипулирования рабочим расписанием управленцев и других работников организации. Современные электронные календари – это полноценные персональные информационные менеджеры (ПИМ), позволяющие осуществлять персональное планирование.

Компьютерные конференции, или *телеконференции*, используют компьютерные сети для обмена информацией между участниками группы, решающей определенную проблему. В случае, когда организуется групповая «встреча», при которой на экране компьютера каждый из участников может видеть видеобразное изображение и слышать то, что говорит другой участник, речь идет о *мультимедиаконференции*.

В любой фирме необходимо длительное время хранить большое количество документов. Их количество может быть так велико, что хранение даже в форме файлов вызывает серьезные проблемы. Поэтому возникла идея хранить не сам документ, а его образ (*изображение*), причем хранить в цифровой форме.

Факсимильная связь основана на использовании факс-аппарата, способного читать документ на одном конце коммуникационного канала и воспроизводить его изображение на другом.

Факсимильная связь вносит свой вклад в принятие решений за счет быстрой и легкой рассылки документов участникам группы, решающей определенную проблему, независимо от их географического положения.

5.4.4. Информационная технология поддержки принятия решений

Характеристика и назначение

Системы поддержки принятия решений и соответствующая им информационная технология появились усилиями в основном американских ученых в конце 70-х – начале 80-х гг., чему способствовали широкое распространение персональных компьютеров, стандартных пакетов прикладных программ, а также успехи в создании систем искусственного интеллекта.

Главной особенностью *информационной технологии поддержки принятия решений* является качественно новый метод организации взаимодействия человека и компьютера. Выработка решения, что является основной целью этой технологии, происходит в результате итерационного процесса, в котором участвуют (рис. 5.15):

- система поддержки принятия решений в качестве вычислительного звена и объекта управления;
- человек как управляющее звено, задающее входные данные и оценивающее полученный результат вычислений на компьютере.



Рис. 5.15. Информационная технология поддержки принятия решений как итерационный процесс

Окончание итерационного процесса происходит по воле человека. В этом случае можно говорить о способности информационной системы совместно с пользователем создавать новую информацию для принятия решений.

Дополнительно к этой особенности информационной технологии поддержки принятия решений можно указать еще ряд ее отличительных характеристик:

- ориентация на решение плохо структурированных (формализованных) задач;
- сочетание традиционных методов доступа и обработки компьютерных данных с возможностями математических моделей и методами решения задач на их основе;
- направленность на непрофессионального пользователя компьютера;
- высокая степень адаптивности, обеспечивающая возможность приспособляться к особенностям имеющегося технического и программного обеспечения, а также требованиям пользователя.

Основные компоненты

Рассмотрим структуру системы поддержки принятия решений, а также функции составляющих ее блоков, которые определяют основные технологические операции (рис. 5.16).

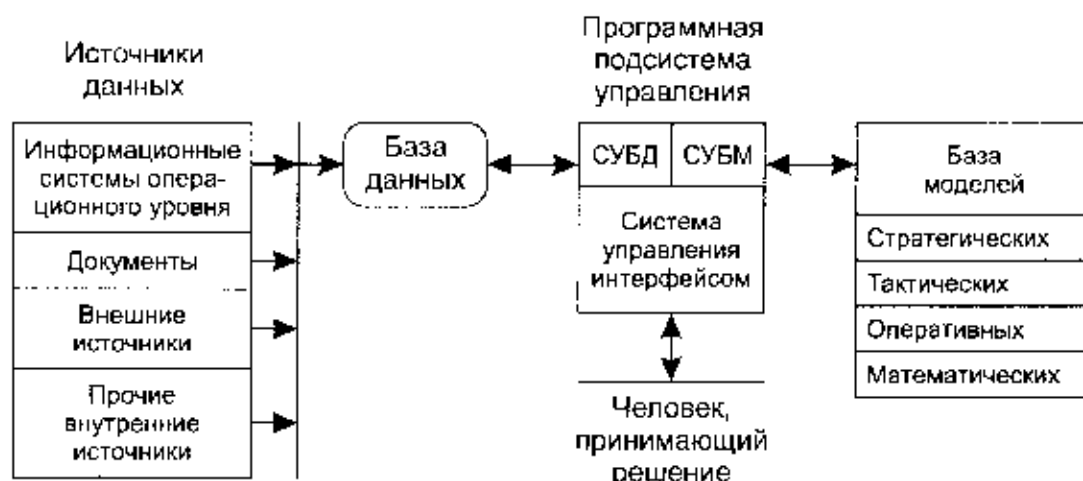


Рис. 5.16. Основные компоненты информационной технологии поддержки принятия решений

В состав системы поддержки принятия решений входят три главных компонента: база данных, база моделей и программная подсистема, которая состоит из

системы управления базой данных (СУБД), системы управления базой моделей (СУБМ) и системы управления интерфейсом между пользователем и компьютером.

База данных играет в информационной технологии поддержки принятия решений важную роль. Данные могут использоваться непосредственно пользователем для расчетов при помощи математических моделей. Рассмотрим источники данных и их особенности.

Система управления данными должна обладать следующими возможностями:

- составление комбинаций данных, получаемых из различных источников, посредством использования процедур агрегирования и фильтрации;
- быстрое прибавление или исключение того или иного источника данных;
- построение логической структуры данных в терминах пользователя;
- использование неофициальных данных и манипулирование ими для экспериментальной проверки рабочих альтернатив пользователя;
- обеспечение полной логической независимости этой базы данных от других операционных баз данных, функционирующих в рамках фирмы.

Целью создания *моделей* являются описание и оптимизация некоторого объекта или процесса. Использование моделей обеспечивает проведение анализа в системах поддержки принятия решений. Модели, базируясь на математической интерпретации проблемы, при помощи определенных алгоритмов способствуют нахождению информации, полезной для принятия правильных решений.

Существует множество типов моделей и способов их классификации, например, по цели использования, области возможных приложений, способу оценки переменных и т. п.

Эффективность и гибкость информационной технологии во многом зависят от характеристик *интерфейса* системы поддержки принятия решений. Интерфейс определяет: язык пользователя; язык сообщений компьютера, организующий диалог на экране дисплея; знания пользователя.

Язык пользователя — это те действия, которые пользователь выполняет в отношении системы, применяя клавиатуру, электронные карандаши, пишущие на экране, джойстик, мышь; подавая команды голосом, и т. п. Наиболее простой формой языка пользователя является создание форм входных и выходных документов.

Язык сообщений — это то, что пользователь видит на экране дисплея (символы, графика, цвет), данные, полученные на принтере, звуковые выходные сигналы и т. п.

Долгое время единственной реализацией языка сообщений был отпечатанный или выведенный на экран дисплея *отчет*, или *сообщение*. Теперь появилось новое средство представления выходных данных — *машинная графика*. Она даст возможность создавать на экране и бумаге цветные графические изображения в трехмерном виде. Еще одна реализация языка сообщений — машинная *анимация* (или *мультипликация*), позволяющая выводить сообщений в виде движущихся изображений.

Пример. Система поддержки принятия решений, предназначенная для обслуживания клиентов в банке, с помощью мультипликационных моделей может

реально проанализировать различные варианты организации обслуживания в зависимости от потока посетителей, допустимой длины очереди, количества пунктов обслуживания и т. п.

В ближайшие годы следует ожидать использования в качестве языка сообщений человеческого голоса. Сейчас эта форма применяется в системе поддержки принятия решений сферы финансов, где в процессе генерации чрезвычайных отчетов голосом поясняются причины исключительности той или иной позиции.

Знания пользователя — это то, что пользователь должен знать, работая с системой. К ним относятся не только план действий, сформированный в сознании пользователя, но и учебники, инструкции, справочные данные, выдаваемые компьютером.

5.4.5. Информационная технология экспертных систем

Характеристика и назначение

Наибольший прогресс среди компьютерных информационных систем отмечен в области разработки *экспертных систем*, основанных на использовании искусственного интеллекта. Экспертные системы дают возможность менеджеру или специалисту получать консультации экспертов по любым проблемам, о которых этими системами накоплены знания.

Под *искусственным интеллектом* обычно понимают способности компьютерных систем к таким действиям, которые назывались бы интеллектуальными, если бы исходили от человека. Чаще всего здесь имеются в виду способности, связанные с человеческим мышлением. Работы в области искусственного интеллекта не ограничиваются экспертными системами. Они также включают в себя создание роботов, систем, моделирующих нервную систему человека, его слух, зрение, обоняние, способность к обучению.

Решение специальных задач требует специальных знаний. Однако не каждая компания может себе позволить держать в своем штате экспертов по всем связанным с ее работой проблемам или даже приглашать их каждый раз, когда проблема возникла. Главная идея применения технологии экспертных систем заключается в том, чтобы получить от эксперта его знания и, загрузив их в память компьютера, использовать всякий раз, когда в этом возникнет необходимость.

Сходство информационных технологий, используемых в экспертных системах и системах поддержки принятия решений, состоит в том, что обе они обеспечивают высокий уровень поддержки принятия решений. Однако имеется три существенных различия. Первое связано с тем, что решение проблемы в рамках систем поддержки принятия решений отражает уровень понимания этой проблемы пользователем и его возможности получить и осмыслить решение. Технология экспертных систем, наоборот, предлагает пользователю принять решение, превосходящее его возможности. Второе отличие указанных технологий выражается в способности экспертных систем пояснять свои рассуждения в процессе получения решения. Очень часто эти пояснения оказываются более важными для пользователя, чем само решение. Третье отличие связано с новым компонентом информационной технологии — знаниями.

Основные компоненты

Основными компонентами информационной технологии, используемой в экспертной системе, являются: интерфейс пользователя, база знаний, интерпретатор, модуль создания системы (рис. 5.17).

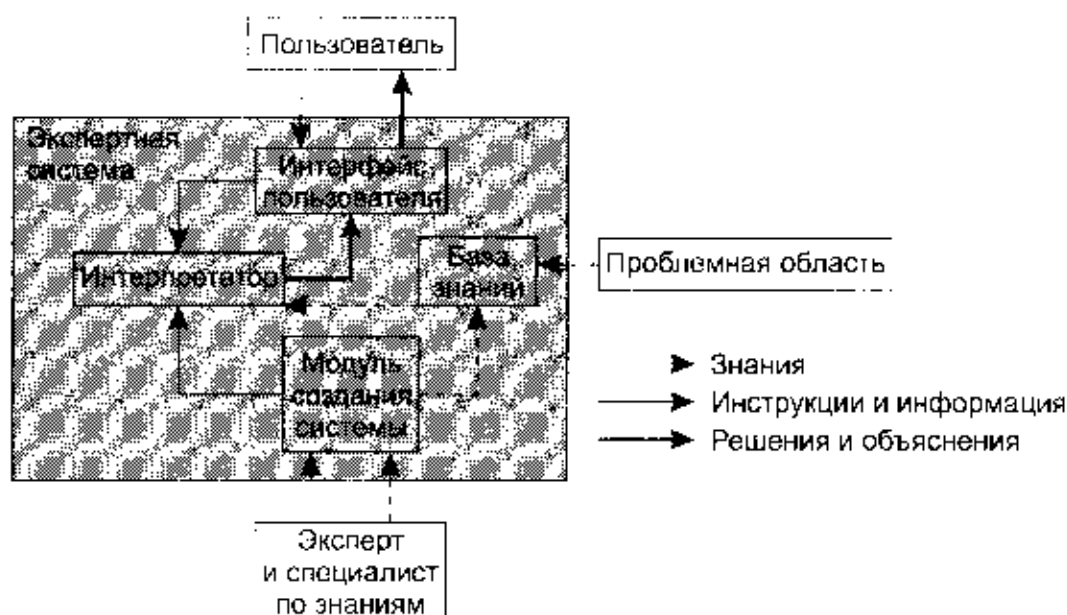


Рис. 5.17. Основные компоненты информационной технологии экспертных систем

Менеджер (специалист) использует *интерфейс пользователя* для ввода информации и команд в экспертную систему и получения выходной информации из нее. Команды включают в себя параметры, направляющие процесс обработки знаний. Информация обычно выдается в форме значений, присваиваемых определенным переменным.

Технология экспертных систем предусматривает возможность получать в качестве выходной информации не только решение, но и необходимые объяснения. Различают два вида объяснений:

- объяснения, выдаваемые по запросам. Пользователь в любой момент может потребовать от экспертной системы объяснения своих действий;
- объяснения полученного решения проблемы. После получения решения пользователь может потребовать объяснений того, как оно было получено. Система должна пояснить каждый шаг своих рассуждений, ведущих к решению задачи.

Хотя технология работы с экспертной системой достаточно сложна, пользовательский интерфейс этих систем является дружественным и обычно не вызывает трудностей при ведении диалога.

База знаний содержит факты, описывающие проблемную область, а также логическую взаимосвязь этих фактов. Центральное место в базе знаний принадлежит правилам. *Правило* определяет, что следует делать в данной конкретной ситуации, и состоит из двух частей: условия, которое может выполняться или нет, и действия, которое следует произвести, если условие выполняется.

Все используемые в экспертной системе правила образуют *систему правил*, которая даже для сравнительно простой экспертной системы может содержать несколько тысяч правил.

Интерпретатор — это часть экспертной системы, производящая в определенном порядке обработку знаний (мышление), находящихся в базе знаний. Технология работы интерпретатора сводится к последовательному рассмотрению совокупности правил (правило за правилом). Если условие, содержащееся в правиле, соблюдается, выполняется определенное действие, и пользователю предоставляется вариант решения его проблемы.

Модуль создания системы служит для создания набора (иерархии) правил. Существует два подхода, которые могут быть положены в основу работы модуля создания системы: использование алгоритмических языков программирования и использование оболочек экспертных систем.

Для представления базы знаний специально разработаны языки Lisp и Prolog, хотя можно задействовать и любой другой известный алгоритмический язык.

Оболочка экспертных систем представляет собой готовую программную среду, которая может быть приспособлена к решению определенной проблемы путем создания соответствующей базы знаний. В большинстве случаев использование оболочек позволяет создавать экспертные системы быстрее и легче в сравнении с программированием.

Вопросы для самопроверки

1. Что такое «информационная система»?
2. Как вы понимаете информационную технологию?
3. В чем состоит разница между компьютерами и информационными системами?
4. Как можно представить процессы, происходящие в информационной системе?
5. Как развивались информационные системы?
6. Почему информационные системы являются стратегическим средством развития фирмы и в чем состоит их вклад?
7. Расскажите о пирамиде уровней управления в фирме.
8. Почему при создании информационной системы следует учитывать влияние внешней среды?
9. Приведите примеры информационных систем, поддерживающих деятельность фирмы.
10. Какие задачи стоят при создании информационной системы?
11. Расскажите об основных функциональных информационных системах.
12. Приведите примеры информационных систем, обеспечивающих эффективность работы.
13. Как вы представляете структуру информационной системы?
14. Расскажите об информационном, техническом, программном и математическом обеспечении, об организационном и правовом обеспечении.
15. Для чего нужна схема информационных потоков?

16. В чем суть методологии построения баз данных?
17. Почему при разработке информационной системы важным фактором является структурированность задач?
18. Как структурированность задач влияет на классификацию информационных систем?
19. Каковы особенности информационных систем, создающих управленческие отчеты?
20. Каковы особенности и виды информационных систем, разрабатывающих альтернативы решений?
21. В чем суть функционального признака при классификации информационных систем?
22. Что такое «признак уровней управления при классификации систем»?
23. Расскажите о пирамиде информационных систем в фирме, где используется функциональный признак классификации.
24. Каковы роль и функции ИС оперативного уровня, ИС для специалистов, для менеджеров среднего звена, стратегических ИС?
25. Приведите классификацию информационных систем по характеру использования информации, по степени автоматизации, по сфере применения.
26. В чем сходство и в чем различие информационной технологии и технологии материального производства?
27. Отобразите информационную технологию в виде иерархической структуры и приведите примеры ее составляющих.
28. Изложите требования, которым должна отвечать информационная технология.
29. Что такое «инструментарий информационной технологии»?
30. Как соотносятся информационная технология и информационная система?
31. Охарактеризуйте методологию использования информационной технологии.
32. Дайте общее представление об информационных технологиях обработки данных, управления, автоматизации офиса, поддержки принятия решений. Назовите их основные компоненты.
33. Расскажите о компьютерных и некомпьютерных офисных технологиях.

Литература

1. Баицаков А. И., Баицакова И. А. Интеллектуальные информационные технологии. М.: Издательство МГТУ им. Баумана, 2005.
2. Гагарина Л. Г., Киселев Д. В., Федотова Е. Л. Разработка и эксплуатация автоматизированных информационных систем. М.: ФОРУМ-ИФРА-М, 2007.
3. Гвоздева В. А., Лаврентьева И. Ю. Основы построения автоматизированных информационных систем. М.: ФОРУМ-ИФРА-М, 2007.

4. Горлопанов В. В., Яловецкий В. И. Информационные технологии в органах государственной власти. Аналитический обзор. М.: Издательство МАГМУ, 2007.
5. Григорьев М. Н., Сергеев В. И., Уваров С. А. Логистика: информационные системы и технологии. М.: Альфа-пресс, 2008.
6. Желена Милан. Информационные технологии в бизнесе. СПб.: Питер, 2002.
7. Советов Б. Я., Цехановский В. В. Информационные технологии. М.: Высшая школа, 2006.

Глава 6 Теория баз данных

- 6.1. Общие понятия
- 6.2. Модели данных
- 6.3. Реляционные базы данных
- 6.4. Постреляционные модели и базы данных
- 6.5. Проектирование баз данных

Современные информационные системы, основанные на концепции интеграции данных, характеризуются огромными объемами хранимых данных, сложной организацией, необходимостью удовлетворять разнообразные требования многочисленных пользователей.

Данная глава призвана сформировать представление о базах данных (БД), в ней рассказывается о возможностях систем управления базами данных (СУБД) и их использовании. Основные функциональные возможности и технологические операции при работе с СУБД рассматриваются без привязки к конкретному типу программного продукта. Знания, полученные при изучении данной главы, являются базовыми при практическом знакомстве с любым новым видом СУБД.

6.1. Общие понятия

Целью любой информационной системы является обработка данных об объектах реального мира. В широком смысле база данных — это совокупность сведений о конкретных объектах реального мира в какой-либо предметной области. Под *предметной областью* принято понимать часть реального мира, например, предприятие, вуз и т. д., подлежащую изучению с целью организации управления и автоматизации.

Рассмотрим несколько определений термина «база данных» (database).

База данных (БД) — это именованная совокупность данных, отражающая состояние объектов и их отношений в предметной области.

База данных — это набор постоянно хранимых данных, используемых прикладными программными системами предприятия.

База данных — это реализация схемы и модели данных на физическом уровне.

База данных — это объект управления в банке данных. База данных описывает состояние объектов предметной области на определенный момент времени совокупностью предложений на некотором формализованном языке.

База данных — это совокупность хранящихся вместе данных при наличии такой минимальной избыточности, которая допускает их использование оптимальным образом несколькими приложениями.

База данных — это совокупность материалов (статей, расчетов, нормативных актов, судебных решений и иных подобных материалов), систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью компьютера.

Все эти определения не являются противоречивыми или взаимоисключающими. Скорее, они представляют разные точки зрения авторов на одно и то же понятие. Сложность определения заключается в том, что компьютерные базы данных за свою не очень длинную историю прошли несколько этапов развития, от файловых систем, хранящих в себе «сырые» (неупорядоченные) данные, до реляционных СУБД, содержимым которых являются данные, обладающие поведением (объекты). Остановимся на еще одном определении.

База данных — это информационная модель предметной области в виде совокупности данных, хранимых в памяти компьютера и связанных между собой по правилам, которые определяют их общие принципы описания, хранения и манипулирования.

Под *информационной моделью* понимают информацию об объекте, отображенную и структурированную в соответствии с заданной целью.

Модель данных — это описание методов представления и обработки данных в СУБД, в том числе методов определения типов и логических структур в базе данных, методов манипулирования данными и методов определения и поддержки целостности базы данных.

Исторически первые базы данных создавались на основе файловых систем, и вся ответственность за работу с ними возлагалась на прикладное программное обеспечение, использовавшее эти базы. Файловые базы данных сегодня практически не применяются. В современной технологии баз данных предполагается, что создание базы данных, ее поддержка и обеспечение доступа пользователей к ней осуществляются централизованно с помощью специального программного инструментария — системы управления базами данных.

Система управления базами данных (СУБД) — это комплекс языковых и программных средств, предназначенный для создания, ведения и применения базы данных пользователями.

Кроме базы данных и программного обеспечения, обеспечивающего основную функциональность СУБД, в состав современных серверов баз данных входят всевозможные средства разработки и механизмы взаимодействия с пользователем на высоком уровне (генераторы отчетов, конструкторы таблиц, построители запросов и форм). Эти средства разработки, сами являясь *приложениями* пользователя, позволяют создавать *приложения*, функционирующие как часть СУБД (например, формы и отчеты MS Access или веб-публикации в Oracle и MS SQL Server).

На рис. 6.1 представлена схема, в которой определены основные термины, используемые при обсуждении СУБД.

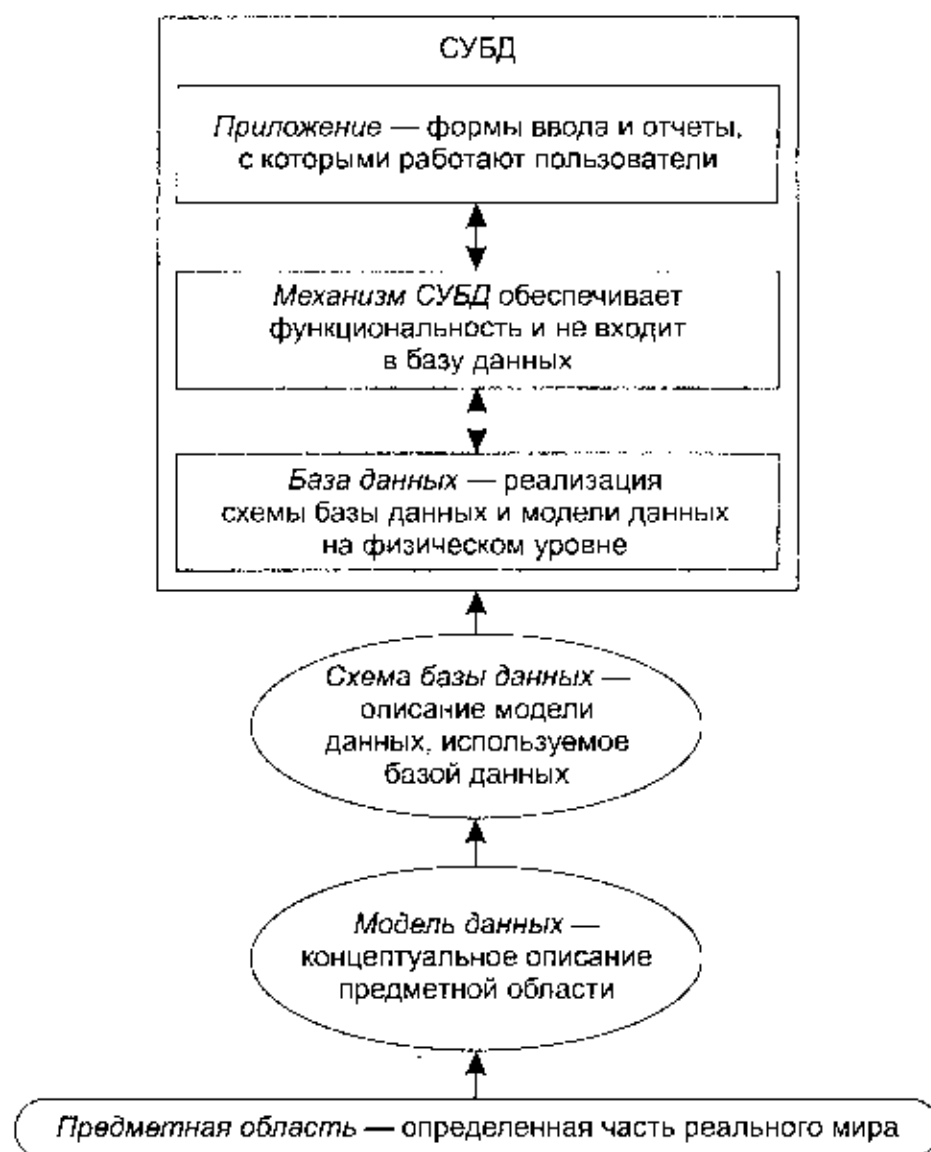


Рис. 6.1. Терминология СУБД

6.1.1. Компоненты среды функционирования СУБД

СУБД представляет собой комплекс программных средств, в работе которого принимает участие множество людей, как обслуживающих эти программы, так и использующих результат их работы. На рис. 6.2 представлены основные компоненты СУБД.

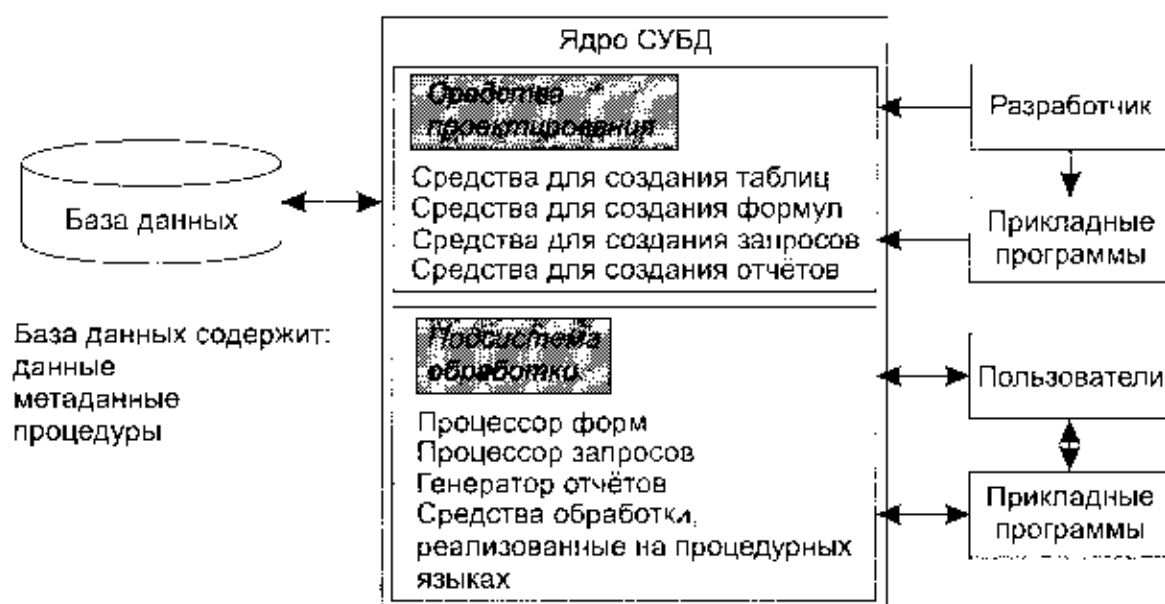


Рис. 6.2. Компоненты среды функционирования СУБД

Программное обеспечение

К программному обеспечению относятся все компьютерные программы, используемые в работе системы управления базами данных. Для выполнения всех функций СУБД требуется программное обеспечение трех видов: системное программное обеспечение, программное обеспечение СУБД, а также прикладные программы и утилиты. Поскольку программное обеспечение СУБД функционально располагается между системным и приложениями пользователя, его относят к разряду промежуточного (middleware) программного обеспечения.

Системное программное обеспечение управляет всеми компонентами оборудования и обеспечивает доступ к нему всех остальных приложений, работающих на компьютере. Примеры системного программного обеспечения: Windows, Linux, UNIX, MVS, MacOS, OpenSolaris и др.

Подсистема обработки СУБД управляет базой данных, реализуя функции СУБД.

Средства проектирования СУБД предназначены для получения доступа к данным и манипулирования ими в среде СУБД. Прикладные программы (приложения пользователя) в большинстве случаев служат для представления данных, хранящихся в БД, в виде отчетов и таблиц.

Люди

Сюда относятся все пользователи системы управления базой данных. Если взять за основу функциональные обязанности, то в системе управления базами данных можно выделить шесть основных групп пользователей: системные администраторы, администраторы баз данных, системные аналитики, проектировщики баз данных, программисты и конечные пользователи.

Системные администраторы несут ответственность и обеспечивают надежное функционирование системного программного обеспечения.

Администраторы баз данных (Data Base Administrator, DBA) управляют работой СУБД, обеспечивают функционирование СУБД, создают учетные записи пользователей СУБД, назначают права, ограничивают доступ, выполняют различные процедуры, связанные с обеспечением безопасности и надежности хранения данных.

Системные аналитики выполняют работу по сбору, систематизации и уточнению требований к структуре данных, приложениям и отчетам.

Проектировщики базы данных (системные архитекторы) проектируют структуру БД.

Программисты разрабатывают прикладное программное обеспечение. Они проектируют и создают формы ввода и отображения данных, отчеты и процедуры, с помощью которых конечные пользователи получают доступ к данным и возможность манипулирования ими.

Конечные пользователи применяют прикладные программы с целью выполнения ежедневных операций, например, в компании — это продавцы, заведующие складами, работники бухгалтерии, руководители и управляющие. Конечные пользователи высшего руководящего звена применяют информацию, полученную из базы данных, для решения тактических и стратегических задач предприятия.

База данных

База данных включает в себя данные, метаданные и процедуры.

Данные. Под терминами «данные», «информация» или «сведения» в данном контексте понимается весь фактический материал, хранящийся в базе данных. Данные являются необработанным сырьем, которое подлежит соответствующему структурированию. Принятие решения о том, какую информацию поместить в БД, каким образом ее упорядочить и структурировать, является важнейшей частью работы системных архитекторов (проектировщиков) базы данных.

Метаданные составляют содержимое системного каталога базы данных и представляют собой сведения об именах и структуре таблиц, именах и правах пользователей, наименовании и типах ограничений, о процедурах, функциях и других объектах базы данных.

Процедуры являются важным компонентом системы. Они устанавливают стандарты ведения коммерческой, технологической и производственно-технической деятельности в рамках предприятия и в отношениях с клиентами. Процедуры также используются для организации наблюдения и аудита как за вводимой в БД информацией, так и за информацией, порождаемой на основе извлекаемых данных.

6.1.2. Классификация СУБД

Классификация по типу принятой модели данных

Классификацию баз данных по модели данных иллюстрирует рис. 6.3.

Иерархические базы данных основаны на иерархической модели данных, в которой связь между объектами базы данных образует перевернутое дерево. При такой модели каждый нижележащий элемент иерархии соединен только с одним расположенным выше элементом

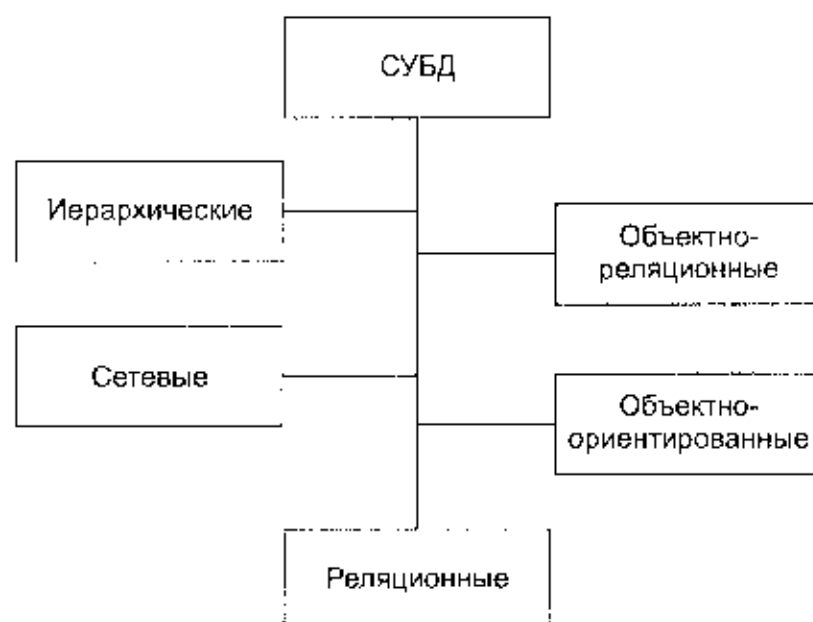


Рис. 6.3. Классификация баз данных по модели данных

Сетевые базы данных основаны на сетевой модели данных, в которой связи между объектами данных могут быть установлены в произвольном порядке.

Реляционные базы данных основаны на реляционной модели данных, в которой каждая единица данных в базе данных однозначно определяется именем таблицы (называемой отношением), идентификатором записи (кортежа) и именем поля.

Объектно-реляционные базы данных содержат объектно-ориентированные механизмы построения структур данных (как минимум, механизмы последования и поддержки методов) в виде расширений языка и программных надстроек над ядром СУБД.

Объектно-ориентированные базы данных определяют как новое поколение баз данных, основанное на сочетании трех принципов: реляционной модели, стандартов на описание объектов и принципов объектно-ориентированного программирования.

Классификация по архитектуре

Классификацию баз данных по архитектуре иллюстрирует рис. 6.4.



Рис. 6.4. Классификация баз данных по архитектуре

В *локальных* базах данных все данные и объекты СУБД находятся на одном компьютере.

В *распределенных* базах данных различные части данных (группы таблиц, таблицы и даже фрагменты таблиц) и объекты СУБД могут находиться на разных компьютерах.

Пример. В качестве примера можно привести сложное производство (или сеть супермаркетов), разные части которого находятся в разных городах. Каждое предприятие накапливает «свои» данные. Необходимо, чтобы каждое из предприятий имело доступ к одним и тем же данным, как своим, так и данным других предприятий. Решением данной проблемы может быть создание одной локальной базы данных на одном компьютере с механизмом удаленного доступа. Однако это решение не рационально, поскольку быстрый доступ к данным будут получать клиентские компьютеры только того предприятия, на котором находится СУБД. Другим решением данной проблемы может быть создание на каждом предприятии своей копии СУБД. В этом случае возникает затруднение с синхронизацией данных между копиями (особенно в масштабах нашей страны, где в Хабаровске может быть разгар рабочего дня, а в Москве — глубокая ночь). Распределенная СУБД в этом случае обеспечивает механизм хранения данных в разных базах данных таким образом, что при обращении совокупность разных баз данных выглядит как одна база. Тогда часто используемые данные («свои» данные) находятся в той части базы данных, которая расположена на предприятии. А при необходимости обратиться к «чужим» данным, СУБД делает запрос к удаленной СУБД и получает данные оттуда. Совокупность разных баз данных на разных компьютерах с точки зрения клиента выглядит как одна база данных.

Классификация по способу доступа к БД

Классификацию баз данных по способу доступа иллюстрирует рис. 6.5.

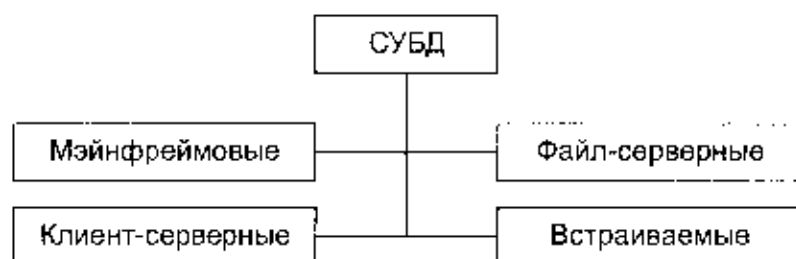


Рис. 6.5. Классификация баз данных по способу доступа

В *мэйнфреймовых* базах данных пользовательское рабочее место представляет собой текстовый или графический терминал, а вся информация обрабатывается на том же компьютере, где находится СУБД.

В *файл-серверных* СУБД файлы данных располагаются централизованно на файл-сервере, а ядро СУБД находится на каждом клиентском компьютере. Доступ к данным осуществляется через локальную сеть. Синхронизация чтений и обновлений осуществляется посредством файловых блокировок. Преимуществом этой архитектуры является низкая нагрузка на ЦП сервера, а недостатком — высокая нагрузка локальной сети.

Клиент-серверные СУБД состоят из клиентской части (которая входит в состав прикладной программы) и сервера. Клиент-серверные СУБД, в отличие от файл-серверных, обеспечивают разграничение доступа между пользователями и мало загружают сеть и клиентские машины. Сервер является внешней по отношению к клиенту программой, и при необходимости его можно заменить другим. Недостаток клиент-серверных СУБД состоит в самом факте существования сервера (что

плохо для локальных программ — в них удобнее встраиваемые СУБД) и больших вычислительных ресурсах, потребляемых сервером.

Встраиваемая СУБД представляет собой программную библиотеку, которая позволяет унифицированным образом хранить большие объемы данных на локальной машине. Доступ к данным может происходить посредством запросов на языке SQL либо путем вызова функций библиотеки из приложения пользователя. Встраиваемые СУБД быстрее обычных клиент-серверных и не требуют развертывания сервера.

Классификация по скорости обработки информации

Классификацию баз данных по скорости обработки информации иллюстрирует рис. 6.6.

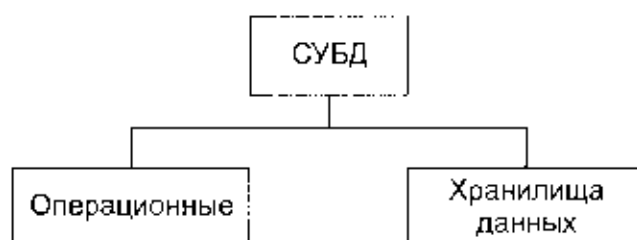


Рис. 6.6. Классификация баз данных по скорости обработки информации

Операционные (operational), или *рабочие* (production), базы данных обладают высокими скоростями реакции на запрос, извлечения и представления информации.

Хранилища данных и *многомерные хранилища данных* (data warehouse, OLAP) — это базы данных с очень большим объемом информации, подготовка представления которой занимает значительный объем времени.

6.1.3. Функции СУБД

Абстракция данных, управление словарем данных. Функционирование СУБД предусматривает, что определения элементов данных и их отношений (метаданные) хранятся в *словаре данных* (data dictionary). В свою очередь любые программы получают доступ к данным посредством СУБД. Для поиска необходимых структур данных и их отношений СУБД использует словарь данных, помогая избежать кодирования таких сложных взаимосвязей в каждой программе. Вдобавок любые изменения, которые делаются в структуре базы данных, автоматически регистрируются в словаре данных, что также освобождает программиста от необходимости модифицировать программы доступа к изменившимся структурам данных. СУБД обеспечивает абстракцию данных, тем самым устраняя в системе структурную зависимость и зависимость по данным.

Управление хранением данных. СУБД создает сложные структуры, необходимые для хранения данных, освобождая программистов от определения и программирования физических свойств данных. Современные СУБД обеспечивают хранение

не только данных, но и связанных с данными экранных форм, схем отчетов, правил проверки данных, кода процедур, систем обработки мультимедиа, форматов изображений, и т. п.

Преобразование и представление данных. СУБД берет на себя задачу структурирования вводимых данных, преобразуя их в форму, удобную для хранения. Поэтому СУБД и в данном случае избавляет человека от рутинной работы по преобразованию логического формата данных в физический формат. Обеспечивая независимость данных, СУБД преобразует логические запросы в команды, определяющие их физическое местоположение и извлечение. Таким образом, СУБД обеспечивает *программную независимость и абстракцию данных*.

Управление безопасностью. СУБД создает систему безопасности, которая обеспечивает защиту пользователя и конфиденциальность данных внутри БД. Правила безопасности устанавливают, какие пользователи могут получить доступ к базе данных, к каким элементам данных пользователь может получить доступ, какие операции с данными (чтение, добавление, удаление или изменение) может выполнять пользователь.

Управление многопользовательским доступом. СУБД создаст сложные структуры, обеспечивающие доступ к данным нескольким пользователям одновременно. Для того чтобы обеспечить целостность и непротиворечивость данных, в СУБД применяются сложные алгоритмы, гарантирующие, что несколько пользователей могут получить одновременный доступ к базе данных без риска нарушить ее целостность.

Управление резервным копированием и восстановлением. В СУБД имеются процедуры резервного копирования и восстановления данных, обеспечивающие их безопасность и целостность. Современные СУБД содержат специальные утилиты, с помощью которых администраторы базы данных могут выполнять регулярные и экстренные процедуры резервного копирования и восстановления данных. Восстановление данных производится после повреждения БД, например, в случае появления сбойного сектора на жестком диске или после аварийного отключения питания. Такая возможность необходима для обеспечения целостности данных.

Управление целостностью данных. В СУБД предусмотрены правила, обеспечивающие целостность данных, что позволяет минимизировать избыточность данных и гарантировать их непротиворечивость. Для обеспечения целостности данных используются их связи, которые хранятся в словаре данных.

Поддержка языка доступа к данным и интерфейсов прикладного программирования. СУБД обеспечивает доступ к данным при помощи языка запросов. *Язык запросов* — это непроцедурный язык, то есть он предоставляет пользователю возможность определить, что необходимо выполнить, не указывая, как это сделать. В состав языка запросов СУБД входят два основных компонента: *язык определения данных* (Data Definition Language, *DDL*) и *язык манипулирования данными* (Data Manipulation Language, *DML*). DDL определяет структуры, в которых размещаются данные, а DML позволяет конечным пользователям извлекать данные из

БД. СУБД также предоставляет программистам доступ к данным из процедурных языков третьего поколения, таких как COBOL, C, PASCAL и др. В составе СУБД имеются административные утилиты, ориентированные на администраторов и проектировщиков базы данных и предназначенные для внедрения, текущего контроля и обслуживания базы данных.

Интерфейсы взаимодействия с базой данных. Текущее поколение СУБД обеспечивает специальные программы взаимодействия, разработанные для того, чтобы база данных могла принимать запросы конечных пользователей в сетевом окружении. Фактически, возможности взаимодействия конечных пользователей с базой данных являются неотъемлемой составляющей современных СУБД. Например, СУБД предоставляет функции взаимодействия для получения доступа к базе данных, используя в качестве внешнего интерфейса интернет-браузер (Mozilla Firefox, Opera или Internet Explorer). В подобной среде взаимодействие может осуществляться несколькими способами:

- конечный пользователь может получать ответы на запросы, заводя экранные формы с помощью выбранного им браузера;
- средствами СУБД можно автоматизировать публикацию форм отчетов в Интернете посредством веб-форматирования, что позволяет просматривать отчеты в любом браузере и др.

6.2. Модели данных

6.2.1. Классификация моделей данных

Центральным понятием в области баз данных является понятие модели данных. Термин «*модель*» используется в нескольких значениях. В предыдущем разделе уже было дано определение модели данных. Другим его значением является описание на разных уровнях абстракции схемы конкретной базы данных, предназначенной для работы в определенных условиях. Несмотря на то что термины одинаковы, во втором случае подразумевается моделирование с точки зрения разработчиков информационной системы (базы данных).

Для того чтобы спроектировать и реализовать реляционную базу данных, состоящую из трех таблиц, нет необходимости прибегать к специальным технологиям и приемам, такого рода работу можно выполнять непосредственно при помощи соответствующих SQL-выражений. Но когда речь идет о базе данных для информационной системы предприятия, такое «прямое» проектирование становится не только утомительным и трудоемким, но во многих случаях просто невозможным. Для того чтобы облегчить работу проектировщиков, в процесс проектирования включается этап моделирования данных. На этом этапе структуры данных сначала представляются в виде графических схем и диаграмм, облегчающих общее пони-

мание связей и взаимодействия объектов базы данных, а также способствующих установлению большего соответствия бизнес-процессов предприятия, бизнес-правил информационной системы и структуры данных, а затем уже реализуют базу данных в виде набора реляционных таблиц и объектов.

На рис. 6.7 представлена классификация моделей данных в соответствии с трехуровневой архитектурой, предложенной в 1975 г. Комитетом планирования стандартов и норм (Standards Planning and Requirements Committee, SPARC) Национального института стандартизации США (American National Standard Institute, ANSI), ANSI/X3/SPARC (рис. 6.8). Так, модели данных, обозначенные на рис. 6.7 как физические, соответствуют первому уровню архитектуры ANSI/X3/SPARC, даталогические модели можно отнести ко второму, внутреннему уровню архитектуры, а инфологические модели соответствуют концептуальному уровню архитектуры, изображенной на рис. 6.8



Рис. 6.7. Классификация моделей данных

Поскольку практически все современные базы данных построены на основе реляционной модели данных, для моделирования данных чаще всего применяется модель «сущность-связь», лучшим образом позволяющая моделировать схемы реляционных баз данных. Однако прежде чем рассмотреть основные принципы применения этой модели, ознакомимся с базовыми терминами и определениями, принятыми при описании структур данных.

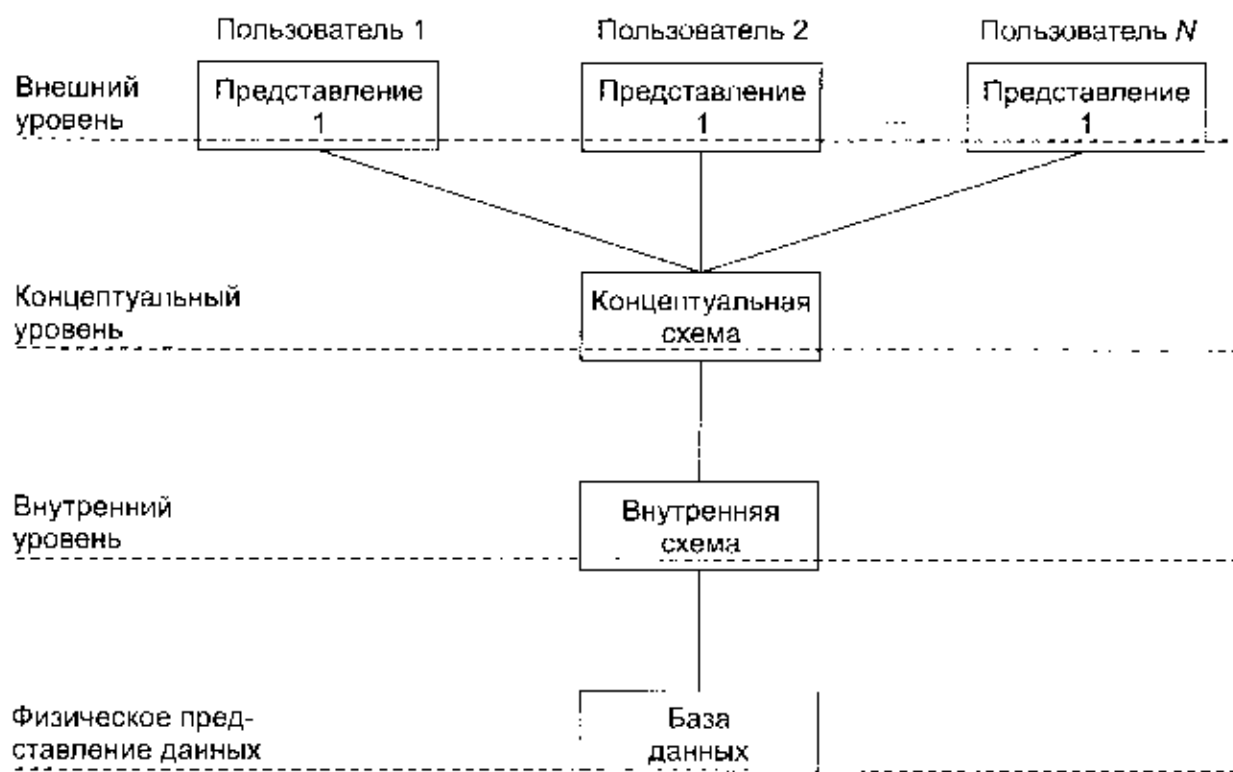


Рис. 6.8. Трехуровневая модель представления данных ANSI-SPARC

6.2.2. Термины и определения

Элемент данных определяет тип данных. Понятие элемента данных применяется как при концептуальном моделировании, так и в ходе создания физической модели данных. На этапе концептуального моделирования элемент данных — это элемент *абстрактного типа данных*, а во время создания физической модели данных это уже элемент *базового типа* конкретной СУБД.

Пример. Во время концептуального моделирования элементу данных, в котором нужно сохранить строку, присваивается абстрактный тип `String`. Если эта концептуальная модель будет реализована на СУБД MS SQL Server, то элемент абстрактного типа `String` преобразуется к базовому типу строковых данных, принятому в MS SQL Server, то есть к типу `varchar`. При реализации этого же элемента на СУБД Oracle он получит тип `char`, являющийся базовым типом для строки в Oracle.

Понятие *записи* в базах данных близко к этому понятию в языках программирования, но не во всем совпадает с ним.

Запись — это уникально идентифицируемая единица независимого хранения в системе баз данных.

Схема записи — это внутренняя структура записи, включающая в себя сведения о количестве полей и типе каждого поля записи.

Схема (тип) записи определяет связную последовательность *полей* — позиций в структурах хранения записей. Внутренняя структура каждого поля определяется типом данных, заданным в объявлении каждой записи.

Для уникальной идентификации записей одно или более полей записи должны быть объявлены явно как *ключ записи*. Значениями полей являются конкретные данные (числа, символьные строки, слова и пр.).

6.2.3. Модель «сущность-связь»

Команда разработчиков анализирует требования и строит пользовательскую модель данных, или модель требований к данным. Эта модель является представлением требований пользователя к структуре и связям объектов, которые должны храниться в базе данных. Для создания пользовательской модели данных команда разработчиков задействует средства, которые называются моделью «сущность-связь» и семантической объектной моделью. Эти средства состоят из языковых и изобразительных стандартов для представления пользовательской модели данных. Их роль в разработке баз данных подобна той роли, которую исполняют алгоритмы и псевдокод в программировании.

В этом разделе описывается и иллюстрируется использование модели «сущность-связь» (Entity-Relationship Model, MER), или *ER-модели*, введенной Питером Ченом (Peter Chen) в 1976 г. Модель «сущность-связь» вошла в состав множества CASE-инструментов, которые также внесли свой вклад в ее эволюцию. На сегодняшний день не существует единого общепринятого стандарта для модели «сущность-связь», но есть набор общих конструкций, которые лежат в основе большинства вариантов этой модели. Символы, применяемые для графического представления модели «сущность-связь», весьма различны.

Термины модели «сущность-связь»

Сущность (entity) — это именованный класс однотипных объектов.

Поскольку термином *сущность* обозначают класс объектов, можно также встретить определение *класса сущностей*, которое является синонимичным термину сущность. Сущность определяет собой некоторый тип сложной структуры данных, то есть наличие определенных полей (атрибутов), их имена и элементарные типы данных, к которым они принадлежат.

Пример. Пример структурно-графического определения класса АВТОР изображен на рис. 6.9.

АВТОР	
ID	<i>int</i>
Фамилия	<i>string</i>
Имя	<i>string</i>
Отчество	<i>string</i>

Рис. 6.9. Определение класса сущностей АВТОР

В этом примере АВТОР — это наименование сущности, в то время как ID, Фамилия, Имя и Отчество являются атрибутами сущности, которым присвоены типы соответствующие хранимым данным.

Атрибуты — именованные характеристики, отображающие свойства данной сущности.

Пример. Класс сущностей АВТОР описывается атрибутами ID, Фамилия, Имя, Отчество.

В модели «сущность-связь» предполагается, что все экземпляры данного класса сущностей имеют одинаковые атрибуты.

Экземпляр сущности — это конкретный представитель данного класса сущностей.

В экземпляре сущности, в отличие от класса, каждый атрибут содержит данные, характеризующие конкретный объект.

Пример. Пример нескольких экземпляров сущности класса АВТОР приведен на рис. 6.10.

1	Пушкин	Александр	Сергеевич
2	Гоголь	Николай	Васильевич
...			
22	Чехов	Антон	Павлович

Рис. 6.10. Несколько экземпляров сущностей класса АВТОР

Как видно из рисунка, все экземпляры имеют одинаковые атрибуты, но разные данные (значения) каждого атрибута.

Идентификаторы (identifiers) — это атрибуты, с помощью которых экземпляры именуются или идентифицируются.

Идентификатор экземпляра сущности состоит из одного или более атрибутов сущности. Идентификатор может быть уникальным (unique) либо неуникальным (nonunique). Если идентификатор является уникальным, его значение будет указывать на один и только на один экземпляр сущности. Если идентификатор является неуникальным, его значение будет указывать на некоторое множество экземпляров. Синонимом термину «идентификатор» является термин «ключ».

Пример. В классе сущностей АВТОР идентификатором является атрибут ID. В случае, если мы в качестве идентификатора приняли бы атрибут Фамилия, этот идентификатор был бы неуникальным. Например, он указывал бы на группу сущностей со значением Толстой у данного атрибута.

Связи (relationships) — это взаимоотношения сущностей.

Модель «сущность-связь» включает в себя классы связей и экземпляры связей. Классы связей определяют взаимоотношения между классами сущностей, а экземпляры связи — взаимоотношения между экземплярами сущностей. У связей могут быть атрибуты.

Класс связей может затрагивать несколько классов сущностей. Число классов сущностей, участвующих в связи, называется *степенью связи* (relationship degree). Изображенная на рис. 6.11 связь ПРОДАВЕЦ-ЗАКАЗ имеет степень 2, поскольку в ней участвуют два класса сущностей: ПРОДАВЕЦ и ЗАКАЗ. Связь РОДИТЕЛЬ имеет степень 3, так как в ней участвуют три класса сущностей: МАТЬ, ОТЕЦ и РЕБЕНОК. Связи степени 2 весьма распространены, их часто называют еще *бинарными связями* (binary relationships).

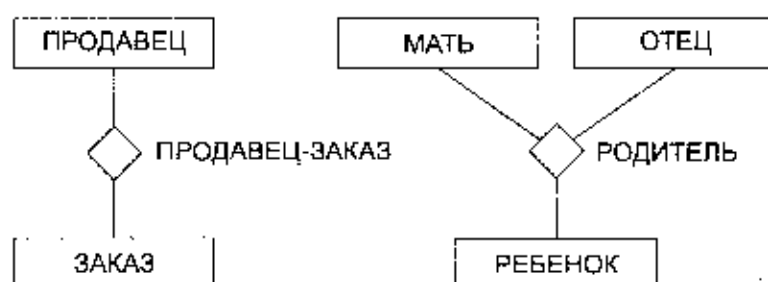


Рис. 6.11. Типы связей

Типы бинарных связей

На рис. 6.12 показаны три типа бинарных связей.

В связи 1:1 (*один к одному*) одиночный экземпляр сущности одного типа связан с одиночным экземпляром сущности другого типа. Связь СЛУЖЕБНЫЙ-АВТОМОБИЛЬ связывает одиночную сущность класса СОТРУДНИК с одиночной сущностью класса АВТОМОБИЛЬ. В соответствии с этой диаграммой, ни за одним сотрудником не закреплено более одного автомобиля и ни один автомобиль не закреплен более чем за одним сотрудником.

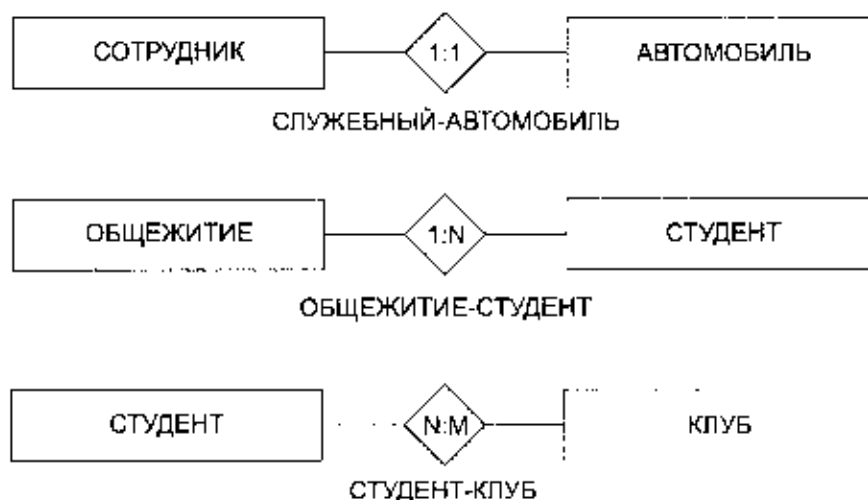


Рис. 6.12. Три типа бинарных связей

В связи типа $1:N$ (*один к N*, или *один ко многим*) экземпляр сущности одного типа связан со многими экземплярами сущности другого типа. В нашем примере таким типом связи является связь ОБЩЕЖИТИЕ-СТУДЕНТ, где единичный экземпляр сущности класса ОБЩЕЖИТИЕ связан со многими экземплярами сущности класса СТУДЕНТ, то есть в общежитии может проживать много студентов, но каждый студент живет только в одном общежитии.

Позиции, в которой стоят символы 1 и N , имеют значение. Единица стоит на той стороне связи, где располагается ОБЩЕЖИТИЕ, а N — на той, где располагается СТУДЕНТ. Если бы символы 1 и N располагались наоборот ($N:1$), получилось бы, что в общежитии живет один студент, причем каждый студент живет в нескольких общежитиях. Это, разумеется, не так.

Третьим типом бинарной связи является связь $N:M$ (читается *N к M*, или *многие ко многим*). В нашем примере это связь СТУДЕНТ-КЛУБ, связывающая экземпляры сущностей класса СТУДЕНТ с экземплярами сущностей класса КЛУБ. Один студент может быть членом нескольких клубов, а в одном клубе может состоять много студентов.

Числа внутри ромба, символизирующего связь, обозначают максимальное количество сущностей на каждой стороне связи. Эти ограничения называются *максимальными кардинальными числами*, а совокупность из двух таких ограничений для обеих сторон связи называется *максимальной кардинальностью* связи.

Преимущества ER-моделирования

Исключительная концептуальная простота. ER-модель дает очень простое и наглядное представление об основных логических объектах БД и существующих между ними связях, поэтому использование такой модели значительно упрощает разработку и организацию сложных баз данных.

Наглядное представление. ER-модель дает проектировщикам баз данных, программистам и конечным пользователям простое наглядное представление о данных и связях между ними. Поэтому ER-модель является чрезвычайно эффективным средством, интегрирующим различные представления о данных в единую рабочую среду.

Хорошая интеграция с реляционной моделью данных. ER-модель хорошо интегрируется с реляционной моделью БД. Такая интеграция помогает эффективно структурировать процесс проектирования реляционных БД.

Недостатки ER-моделирования

Недостаточные возможности представления ограничений. С помощью ER-модели легко изобразить ограничения, имеющие непосредственное отношение к связности. Например, ограничение «автор может работать над несколькими книгами, или более чем над одной книгой, но не более чем над тремя книгами» легко изображаются средствами ER-модели. Однако некоторые ограничения, например, «оплата автору может варьироваться в диапазоне от 4 до 25 %» или «автор не имеет права работать больше 10 часов подряд», в ER-модели представить невозможно, и они должны обрабатываться на уровне приложений.

Ограниченные возможности представления отношений. Связи представляются как нечто, происходящее *между* сущностями. Поэтому связи между атрибутами внутри сущностей не могут быть представлены средствами ER-модели.

Отсутствие языка манипулирования данными. Сторонники реляционной модели обычно указывают на отсутствие команд манипулирования данными в ER-модели. Отсутствие таких команд делает ER-модель «неполной».

Утеря информационного наполнения. ER-модель сильно «переполнится», если в ней отобразить все ее атрибуты. Поэтому проектировщики баз данных обычно избегают полного отображения атрибутов, таким образом, сужая информационное наполнение ER-модели.

ВНИМАНИЕ

Все сказанное о недостатках ER-модели верно лишь в отношении теоретической модели. В современных средствах моделирования данных (таких как программа ErWin Data Modeler) применяется расширенная ER-модель, практически лишенная всех перечисленных недостатков.

6.3. Реляционные базы данных

6.3.1. Реляционная модель данных

Наиболее популярным сейчас типом баз данных являются так называемые реляционные базы данных. *Реляционная модель данных* была предложена Е. Ф. Коддом (E. F. Codd), известным исследователем в области баз данных, в 1969 г., когда он был сотрудником IBM.

Реляционная база данных представляет собой хранилище данных, содержащее набор двумерных таблиц. Набор средств для управления подобным хранилищем называется *реляционной системой управления базами данных* (РСУБД). РСУБД может содержать утилиты, приложения, сервисы, библиотеки, средства создания приложений, и др.

Любая таблица реляционной базы данных состоит из строк (называемых также записями) и столбцов (называемых также полями). В данном разделе используются обе пары терминов.

Строки таблицы содержат сведения (факты) об однотипных объектах: документах, людях и пр. На пересечении столбца и строки находятся конкретные значения содержащихся в таблице данных.

Данные в таблицах удовлетворяют следующим принципам:

- каждое значение в ячейке, находящейся на пересечении строки и столбца, должно быть атомарным, то есть не расчленимым на несколько значений;
- значения данных в одном и том же столбце должны принадлежать одному и тому же типу, доступному для использования в данной СУБД;
- каждая запись в таблице уникальна, то есть в таблице не существует двух записей с полностью совпадающим набором значений ее полей;
- каждое поле имеет уникальное имя;

- последовательность полей в таблице несущественна;
- последовательность записей в таблице несущественна.

Несмотря на то что строки таблиц считаются неупорядоченными, любая система управления базами данных позволяет сортировать строки и столбцы. Так как последовательность столбцов в таблице несущественна, обращение к ним производится по имени, и эти имена для данной таблицы уникальны. Для того чтобы база данных (или система управления базой данных) была полностью реляционной, она должна соответствовать 12 правилам Кодда.

6.3.2. Правила Кодда

Автор реляционной модели данных, Е. Ф. Кодд, в 1985 г. опубликовал статью, в которой сформулировал 12 правил реляционной базы данных.

1. *Правило информации.*

Вся информация реляционной базы данных на логическом уровне в полной мере представляется только одним способом — значениями в таблицах.

ПОЯСНЕНИЕ

Не только данные, но и отношения между таблицами или ограничения должны быть представлены в виде значений в таблицах.

2. *Правило гарантированного доступа.*

Гарантируется, что каждый элемент данных (атомарное значение) реляционной базы данных логически доступен через комбинацию из имени таблицы, значения первичного ключа и имени столбца.

ПОЯСНЕНИЕ

В наборе из 12 правил нет явного указания, что каждая строка в таблице должна иметь уникальный первичный ключ. Однако правило 2 неявным образом говорит об этом: ни одна таблица не может соответствовать правилу гарантированного доступа, если в ней нет первичного ключа.

3. *Правило систематической трактовки значений null.*

Значения null (заметим, что они отличаются как от пустой строки символов, так и от строки пробелов) поддерживаются в полностью реляционной СУБД для представления отсутствующей информации систематически и независимо от типа данных.

ПОЯСНЕНИЕ

Во-первых, база данных должна поддерживать значения null для обозначения того, что атомарные данные являются неизвестными (null — это не 0, не пустая строка и не пробел); во-вторых, в базе данных должно быть общее правило, определяющее, как обрабатывать значения null (например, каким образом трактовать их при выполнении запросов с выборкой по полю, содержащему такие значения).

4. Правило наличия динамического оперативного каталога на основе реляционной модели.

Описание базы данных представляется на логическом уровне так же, как обычные данные, значит, уполномоченные пользователи могут применять для обращения к этому описанию тот же самый реляционный язык, что и для работы с обычными данными.

5. Правило наличия исчерпывающего подязыка данных.

Реляционная система может поддерживать несколько языков и различных режимов работы с устройствами ввода-вывода, однако должен существовать по меньшей мере один язык, операторы которого точно отражают реляционную модель с вполне определенными синтаксическими конструкциями в виде строк символов и с исчерпывающей поддержкой следующего:

- описания данных;
- описания представлений;
- манипуляций данными;
- ограничений целостности;
- границ транзакций (начала, завершения и отката).

6. Правило обновления представлений.

Все представления, обновляемые теоретически, должны обновляться практически.

7. Правило ввода, обновления и удаления данных на высоком уровне.

Возможность работы с базовым или производным отношением в качестве одного операнда применима не только к считыванию, но и к вводу, обновлению и удалению данных.

ПОЯСНЕНИЕ

Это правило определяет возможность в одной команде выполнить операцию сразу над несколькими строками данных.

8. Правило физической независимости данных.

Прикладные программы и терминальные операции остаются логически неповрежденными при любом изменении области хранения информации или механизма доступа.

ПОЯСНЕНИЕ

Это правило подразумевает возможность переноса физических данных из одного каталога (раздела, диска, компьютера) в другой без изменения способа взаимодействия пользователей и пользовательских приложений с таблицами базы данных.

9. Правило логической независимости данных.

Прикладные программы и терминальные операции остаются логически неповрежденными при внесении в базовые таблицы любого рода изменений, со-

храняющих информацию и теоретически допускающих возможность оставить информацию неповрежденной.

ПОЯСНЕНИЕ

Добавление таблицы к схеме базы данных или столбца к схеме таблицы не должно оказать никакого влияния на уже работающие прикладные программы, поскольку такое действие не уничтожает (а сохраняет) информацию. В случае удаления таблицы или столбца сохранение информации, а значит, и работоспособность прикладных программ не гарантируется.

10. *Правило независимости целостности.*

При помощи подязыка реляционных данных должна существовать возможность описания ограничений целостности, специфичных для конкретной реляционной базы данных, и сохранения их в каталоге, а не в прикладных программах.

Для этого должны соблюдаться как минимум два ограничения:

- **сущностная целостность:** ни один из компонентов первичного ключа не может принимать значения null;
- **ссылочная целостность:** для каждого значения внешнего ключа, отличного от других и не являющегося значением null, в реляционной базе должно существовать соответствующее значение первичного ключа из того же домена.

11. *Правило независимости распределения.*

В реляционной СУБД распределение независимо.

ПОЯСНЕНИЕ

Распределенная база данных с точки зрения пользователя не должна отличаться от централизованной.

12. *Правило соблюдения правил (запрет обхода правил).*

Если в реляционной системе имеется язык низкого уровня («одна запись за раз»), его нельзя использовать для нарушения или пропуска правил или ограничений целостности, установленных в реляционном языке более высокого уровня («несколько записей за раз»).

6.3.3. Ключи и связи

Рассмотрим стандартную для современных информационных систем задачу построения базы данных, которая содержит в себе информацию о клиентах и сделанных ими заказах (покупках).

Самое первое решение: каждый раз, когда клиент делает заказ, делать об этом одну запись в базу данных. Схема этой записи могла бы выглядеть так, как показано на рис. 6.13.

Клиенты и заказы
Номер заказа
Имя клиента
Адрес заказа
Адрес отгрузки
Контактный адрес
Номер счета
Дата заказа
Дата отгрузки
Дата оплаты
Товар 1
Товар 2
Товар 3
Товар 4

Рис. 6.13. Схема таблицы Клиенты и заказы

В этой таблице будет множество записей. Все поля в записях могут принять одинаковые значения (даже номер счета, поскольку один и тот же клиент может сделать два идентичных заказа, и их впишут в один счет). Значит, для того чтобы обеспечить уникальность каждой записи в таблице, необходимо иметь поле, в котором значения никогда не повторяются. Если в данной фирме принята сквозная уникальная нумерация заказов, мы можем в качестве уникального идентификатора выбрать номер заказа. В случае, если это не так, мы можем в качестве уникальной метки для записи добавить еще одно поле, значения в котором будут уникальными.

Первичный ключ — это поле, предназначенное для идентификации записи в таблице. Значения первичного ключа уникальны.

Проведя моделирование процесса добавления заказов в таблицу, мы можем понять, что такая схема записей является несовершенной: сколько бы позиций товаров мы не внесли, клиент все равно может заказать их больше. Если же мы в схеме предусмотрим заведомо избыточное количество позиций товаров, то получим совершенно нечитабельную запись и такую таблицу, в которой при каждом добавлении заказа будет дублироваться большое количество незаполненных полей. Избавиться от такого рода несовершенства можно за счет создания связи между таблицами (в нашем случае — связи *один ко многим*).

Произведя разделение схемы таблицы на две, одна из которых содержит сведения о сделанном заказе, а другая — о каждом заказанном товаре, мы свяжем эти таблицы между собой при помощи связи *один ко многим* (на рис. 6.14 это линия с цифрой 1 около одного конца и значком бесконечности около другого).

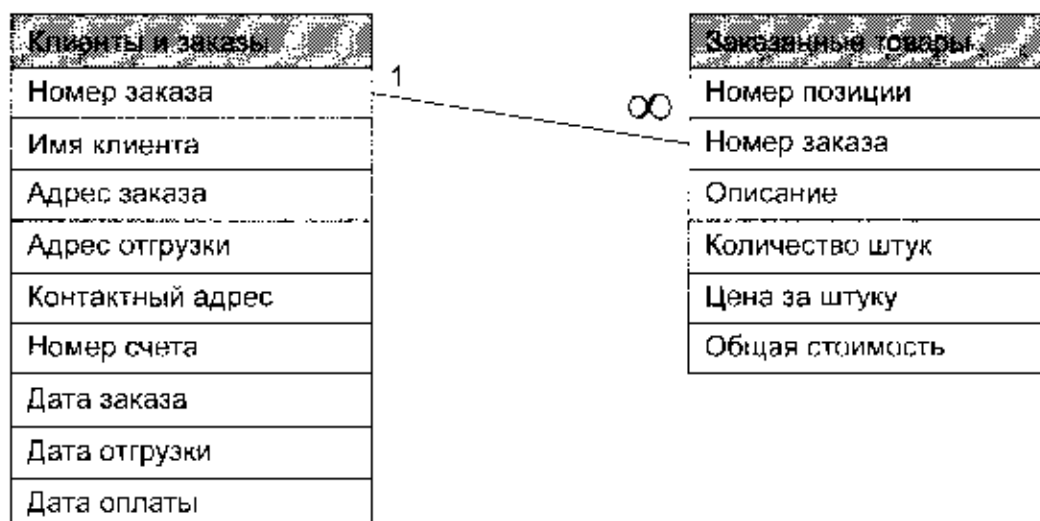


Рис. 6.14. Связь один ко многим

Связь *один ко многим* связывает один оформленный заказ с набором товаров, входящих в заказ, через поле *Номер заказа*. При этом в таблице *Клиенты и заказы* это поле является первичным ключом. В таблице же *Заказанные товары* это поле выступает в роли *внешнего ключа*, то есть ключа, через который осуществляется связь между набором товаров и конкретным заказом. Выбрав в таблице *Заказанные товары* все записи, в которых значения внешнего ключа совпадают, можно получить набор записей, относящихся к одному заказу. Первичным ключом таблицы *Заказанные товары* является поле *Номер позиции*.

6.3.4. Ссылочная целостность

Как уже отмечалось, первичный ключ любой таблицы должен содержать уникальные непустые значения для данной таблицы. Это утверждение является одним из правил *ссылочной целостности* (referential integrity). Практически все современные СУБД контролируют уникальность первичных ключей. При попытке присвоить первичному ключу значение, уже имеющееся в другой записи, СУБД генерирует диагностическое сообщение, обычно содержащее словосочетание *primary key violation* (нарушение первичного ключа). Если две таблицы связаны соотношением *один ко многим*, внешний ключ второй таблицы должен содержать только те значения, которые уже имеются среди значений первичного ключа первой таблицы. Современные СУБД отслеживают попытки удалить запись в первой таблице, если во второй таблице есть связанные с ней записи, препятствуя этому и генерируя диагностические сообщения. Если бы этого не происходило, со временем вторая таблица переполнилась бы записями, не связанными ни с одной записью в первой таблице (а значит, попросту бесполезными).

6.3.5. Нормализация данных

Нормализация представляет собой процесс приведения таблиц к виду, позволяющему осуществлять непротиворечивое и корректное редактирование данных.

Первая нормальная форма

Вернемся к таблице Клиенты и заказы. Структура этой таблицы показана на рис. 6.15.

Клиенты и заказы
Номер заказа
Имя клиента
Адрес заказа
Адрес отгрузки
Контактный адрес
Номер счета
Дата заказа
Дата отгрузки
Дата оплаты

Рис. 6.15. Таблица Клиенты и заказы в первой нормальной форме

Чтобы таблица соответствовала *первой нормальной форме*, все значения ее полей должны быть атомарными и все записи — уникальными.

Таблица Клиенты и заказы соответствует этим требованиям, как и любая другая реляционная таблица. Однако она, несмотря на выделение из нее таблицы с заказанными товарами, остается в значительной степени избыточной. На самом деле каждый раз, добавляя в таблицу очередной заказ от одного и того же клиента, мы многократно дублируем множество сведений. Это, в свою очередь, может привести к некоторым негативным последствиям:

- до тех пор, пока клиент не сделал хотя бы один заказ, сведения о нем не появятся в базе данных;
- при удалении последней записи о заказанном товаре из базы исчезают все сведения о клиенте;
- при изменении клиентом адреса в базе могут оказаться две записи с разными адресами одного и того же клиента.

Некоторые из этих проблем могут быть решены путем приведения таблиц данных ко второй нормальной форме.

Вторая нормальная форма

Реляционная таблица находится во *второй нормальной форме*, если она находится в первой нормальной форме и ее неключевые поля полностью зависят от всего первичного ключа.

Получить вторую нормальную форму можно, разделив таблицу Клиенты и заказы еще на несколько таблиц. При этом поля разлосятся таким образом, чтобы полностью зависеть от первичного ключа соответствующей таблицы (рис. 6.16).

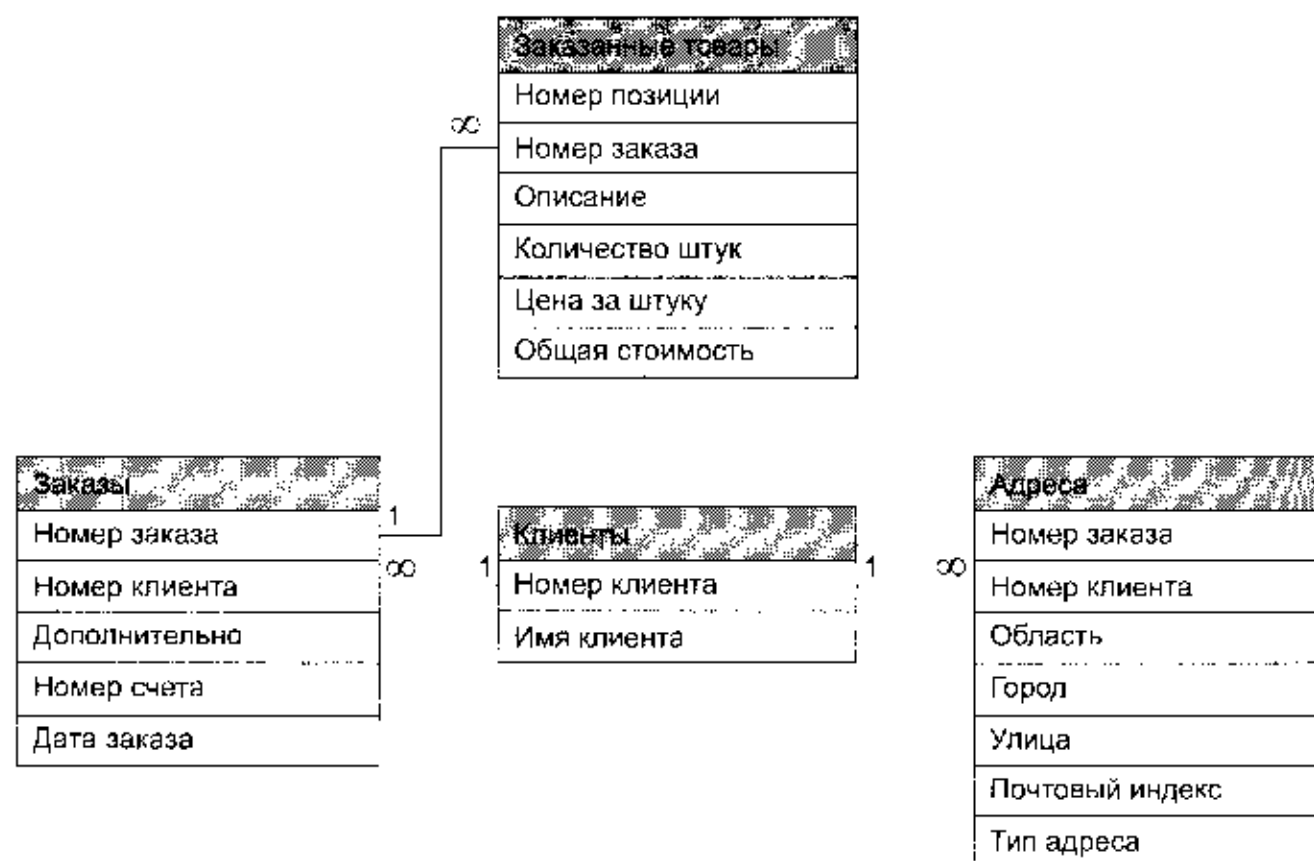


Рис. 6.16. Таблица Клиенты и заказы во второй нормальной форме

Третья нормальная форма

Итак, таблица Клиенты и заказы превратилась у нас в группу связанных таблиц. Благодаря упорядоченности связей в схеме появилась логика: в центре теперь клиент, с которым связаны его заказы и адреса. В свою очередь, с заказами связаны заказанные товары. Но и эта схема все еще не соответствует третьей нормальной форме. Происходит это потому, что в таблице Заказанные товары есть поле Общая стоимость. Это поле получено в результате произведения значений полей Количество штук и Цена за штуку.

База данных находится в *третьей нормальной форме*, если она находится во второй нормальной форме и все поля ее сущностей не зависят друг от друга¹.

Таким образом, чтобы привести полученную группу объектов к третьей нормальной форме, нужно удалить из таблицы Заказанные товары поле Общая стоимость.

6.3.6. Язык SQL

Математическая реляционная модель данных предполагает, что для манипуляций с данными и сущностями должна использоваться одна из специальных алгебр — реляционная. Для реализации операций реляционной алгебры в СУБД был разработан специальный язык SQL (Structured Query Language — структурированный язык запросов). SQL является декларативным языком программирования. В отличие от процедурных языков программирования, SQL определяет де

¹ Существуют и другие нормальные формы с более строгими условиями.

последовательность процедур, а условия выполнения запроса. При этом запросы могут быть объемом с простую программу на процедурном языке программирования (от одной строки до нескольких страниц).

Язык SQL содержит несколько групп операторов.

Операторы манипулирования данными

Операторы манипулирования данными позволяют производить извлечение, добавление, изменение и удаление данных. В группу входят следующие операторы:

- **SELECT** – оператор извлечения данных.

Пример:

```
SELECT 'Номер заказа', 'Номер счета', 'Дата заказа'  
FROM 'Заказы' WHERE 'Номер клиента' = 20
```

Этот запрос обращается к таблице Заказы, и выбирает из нее все записи, принадлежащие клиенту с номером 20, организуя эти записи в три столбца: Номер заказа, Номер счета, Дата заказа.

- **INSERT** — оператор добавления данных.

Пример:

```
INSERT INTO 'Заказы' VALUES (1420, 20, 'Ускоренная доставка',  
1221, 15.03.2010)
```

Этот запрос обращается к таблице Заказы и добавляет к ней запись со значениями полей, перечисленными в скобках после оператора VALUES.

- **UPDATE** – оператор изменения данных.

Пример:

```
UPDATE 'Заказы' SET 'Номер счета' = 1432 WHERE 'Номер заказа' = 1220
```

Этот запрос обращается к таблице Заказы и в записи с номером заказа 1220 изменяет значение поля Номер счета на 1432.

- **DELETE** — оператор удаления данных.

Пример:

```
DELETE FROM 'Заказы' WHERE 'Номер клиента' = 20
```

Из таблицы Заказы удаляются строки с номером клиента 20 (то есть все заказы данного клиента).

Операторы определения данных

Операторы определения данных позволяют манипулировать схемой базы данных, то есть создавать объекты базы данных (таблицы, представления, триггеры, хранимые процедуры, индексы, ключи, ограничения, и т. д.).

- **CREATE (TABLE, VIEW, TRIGGER, INDEX, STORED PROC)** — создаст указанный объект базы данных.

- **DROP (TABLE, VIEW, TRIGGER, INDEX, STORED PROC)** — уничтожает указанный объект базы данных.

- **ALTER (TABLE, VIEW, TRIGGER, INDEX, STORED PROC)** — изменяет структуру указанного объекта базы данных.

Операторы управления данными

К операторам управления данными относятся операторы назначения прав доступа и управления транзакциями, а также операторы создания баз данных и изменения структуры данных.

□ **GRANT** — предоставляет привилегии пользователям.

Пример:

```
GRANT SELECT ON 'Заказы' TO NEKTO
```

Разрешает пользователю NEKTO выполнять запросы на выборку данных к таблице Заказы. При этом пользователь не получает привилегий, позволяющих ему изменять данные в этой таблице или в структуре таблицы.

□ **REVOKE** — аннулирует привилегии.

Пример:

```
REVOKE ALL ON 'Заказы' FROM NEKTO
```

Лишает пользователя NEKTO всех привилегий в таблице Заказы.

□ **COMMIT** — подтверждает транзакцию.

□ **ROLLBACK** — отменяет транзакцию.

□ **CREATE DATABASE** — создает базу данных.

□ **ALTER DATABASE** — изменяет базу данных.

□ **DROP DATABASE** — уничтожает базу данных.

В данном разделе перечислены только основные операторы языка SQL. Даже в стандартной реализации их гораздо больше. Однако ограничения, накладываемые декларативным характером языка, вынуждают производителей СУБД создавать расширенные версии SQL. Примерами таких расширенных версий могут служить Transact SQL для MS SQL Server или PL/SQL для ORACLE. В этих расширениях в язык включены операторы для организации ветвлений, циклов, и т. д.

6.3.7. Преимущества и недостатки реляционной модели

Структурная независимость. Поскольку в реляционной модели баз данных навигационная схема доступа к данным не используется, маршрут доступа к данным не имеет значения для проектировщиков, программистов и конечных пользователей реляционной базы данных. Изменения в структуре реляционной БД не влияют на доступ к данным со стороны СУБД. Поэтому в реляционной модели базы данных достигается структурная независимость, не свойственная сетевым и иерархическим моделям. Структурная независимость имеет место, когда изменения в структуре БД не влияют на возможности доступа к данным со стороны СУБД. В отличие от реляционной базы данных, любые изменения в древовидной структуре иерархической базы данных или во множествах баз сетевой БД влияют на маршруты доступа к данным, что требует изменения всех прикладных программ.

Концептуальная простота. Реляционная модель на концептуальном уровне проста для понимания. Поскольку реляционная модель позволяет полностью избавиться от подробностей физического хранения данных, пользователи могут полностью сосредоточиться на логическом представлении базы данных, то есть уделять большее внимание естественному представлению о хранении.

Простота проектирования, реализации, управления и использования. Поскольку в реляционной модели достигаются и независимость по данным, и структурная независимость, становится проще проектировать базу и управлять ее содержанием.

Нерегламентированные запросы. Одним из главных факторов, позволивших реляционным базам данных занять доминирующее положение на рынке, была возможность применять гибкий и унифицированный механизм создания запросов. Для большей части программного обеспечения реляционных БД стандартным языком запросов является язык SQL. SQL относится к так называемым языкам четвертого поколения (4GL), которые дают пользователям возможность определить, *что* делать, не указывая, *как* именно это делать. В РСУБД язык SQL применяется при трансляции запроса пользователя в специальный код, необходимый для извлечения запрошенной информации. Следовательно, запросы в реляционной базе данных требуют меньшего программирования, чем в любой другой базе или в файловой системе.

Любое SQL-приложение реляционной БД состоит из трех частей: интерфейса пользователя, набора таблиц внутри БД и *SQL-машины* (SQL-engine). Интерфейс включает в себя систему меню, команды запросов и генераторы отчетов. В основном такой интерфейс дает возможность конечному пользователю взаимодействовать с данными. С помощью генераторов приложений, которые сегодня являются стандартными средствами, входящими в состав многих РСУБД, пользователь может разрабатывать собственные интерфейсы.

В значительной степени скрытая от конечного пользователя SQL-машина выполняет большую работу. Внутри РСУБД SQL-машина служит для создания структуры таблиц, обслуживания словаря данных и системного каталога, обеспечения доступа к таблицам БД, а также для трансляции запроса пользователя в формат, пригодный для обработки компьютером.

Мощная система управления базой данных. Хорошая РСУБД является более сложной частью программного обеспечения, нежели СУБД иерархических и сетевых баз данных. Это связано с тем, что она решает гораздо больше (и значительно более сложных) задач как для проектировщиков, так и для пользователей. РСУБД полностью скрывает физический уровень сложности системы от проектировщика и конечного пользователя.

Несмотря на существенные преимущества реляционной модели перед иерархической или сетевой, она имеет некоторые недостатки. Один из них – это существенные требования к оборудованию и системному программному обеспечению. На самом деле это не столько недостаток, сколько вполне справедливая плата за те преимущества, которые получают пользователи, программисты и администраторы.

6.4. Постреляционные модели и базы данных

По мере того как практические задачи приобретают все более сложный и комплексный характер, а информационные системы все больше склонны проявлять интеллектуальное поведение при взаимодействии с окружающим миром, необходимы более совершенные модели данных. Такие модели должны точнее описывать окружающий мир: уже не в терминах наборов данных, а в терминах объектов, обладающих свойствами, состояниями и поведением.

Первой из таких моделей стала *семантическая модель данных* (Semantic Data Model, *SDM*), разработанная М. Хаммером и Д. Маклеодом в 1981 г. SDM позволяет моделировать как данные, так и их отношения в единой структуре, называемой *объектом*. Поскольку основной структурой модели является объект, модель SDM получила название *объектно-ориентированной модели данных* (Object-Oriented Data Model, *OODM*). В свою очередь, OODM стала основой для создания объектно-ориентированной модели базы данных (*OODBM*), управление которой осуществляется с помощью системы управления объектно-ориентированной базой данных (*ООСУБД* или *OODBMS*).

Объект может включать в себя все данные о себе, связи с другими объектами и операции, которые могут выполняться с объектом (или выполняться объектом). Таким образом, объекты в усовершенствованной модели SDM стали похожи на объекты, создаваемые объектно-ориентированными языками. Это, в свою очередь, обеспечило возможность сближения и тесной интеграции различных групп разработчиков, как проектировщиков и архитекторов базы данных, так и программистов, разрабатывающих пользовательские программы.

Рисунок 6.17 иллюстрирует такую методологию работы, когда различные группы разработчиков, вовлеченные в процесс обеспечения работы ООСУБД, используют общий репозиторий (хранилище) классов для решения различных задач на различных этапах работы над проектом.

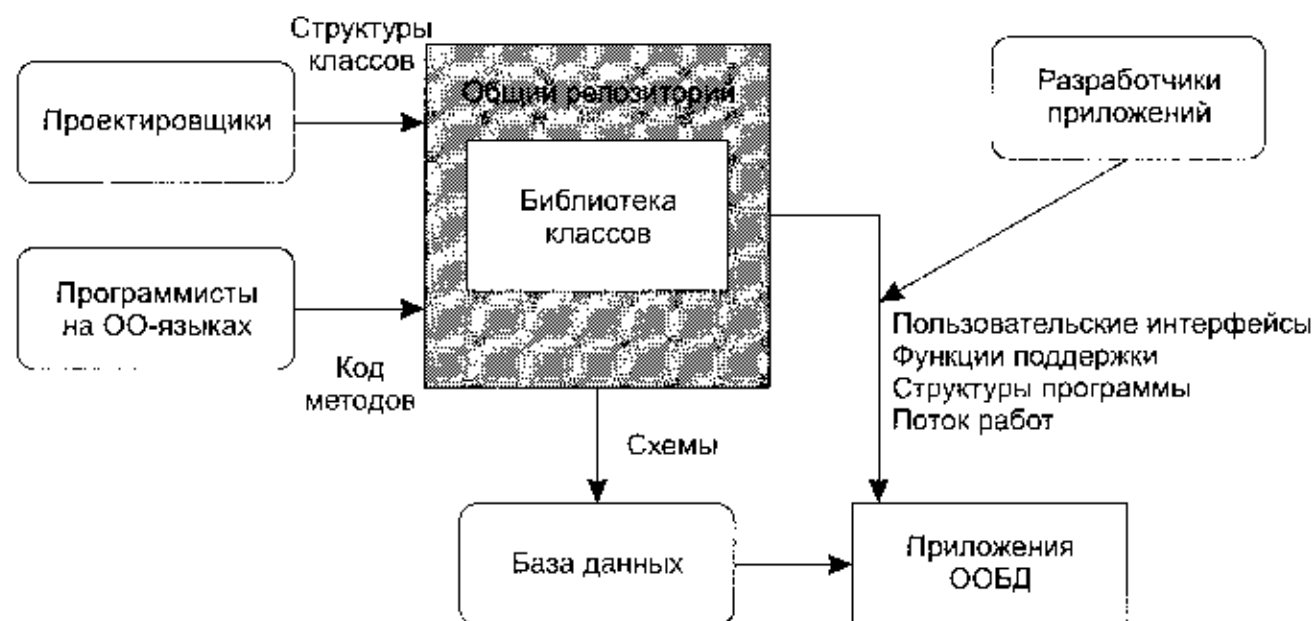


Рис. 6.17. Методология совместной разработки в парадигме ООБД

6.4.1. Основные понятия OODM

Объекты модели данных являются абстракцией сущностей и событий реального мира. В общих чертах любой объект может рассматриваться как эквивалент сущности ER-модели. Точнее, любой объект представляет собой только один экземпляр сущности.

Атрибуты описывают свойства объекта. Например, в объект PERSON (персона) включены атрибуты Name (имя), Social Security Number (номер социального страхования) и Date of Birth (дата рождения).

Объекты, которые совместно используют одни и те же характеристики, группируются в классы. *Класс* представляет собой абстрактное представление схожих объектов со структурой совместного доступа (*атрибутами*) и поведением (*методами*). В общем случае класс напоминает набор сущностей ER-модели. Однако класс отличается от набора сущностей тем, что содержит набор процедур, называемых методами. *Метод класса* представляет собой некоторое действие, например, поиск, изменение данных или распечатку данных. Иначе говоря, методы эквивалентны процедурам в традиционных языках программирования. В терминах объектно-ориентированного подхода методы определяют *поведение* объекта.

Классы организованы в *иерархию классов*. Иерархия классов похожа на перевернутое дерево, в ней каждый класс имеет только одного предка (явное сходство с иерархической моделью).

Наследование — это возможность объекта внутри иерархии классов наследовать атрибуты и методы классов, структурно расположенных выше него.

6.4.2. Преимущества OODM

Объектно-ориентированная модель имеет несколько важнейших преимуществ перед ER-моделью.

Добавление семантического наполнения делает модель более значимой и информационно-насыщенной.

Семантическое наполнение включено во внешнее представление. Как и ER-диаграммы, объектно-ориентированная модель представляет отношения в наглядной форме. Однако в наглядное представление объектно-ориентированной модели включается семантическое наполнение, что упрощает визуализацию сложных отношений между объектами.

Целостность базы данных. Так же как и иерархическая, объектно-ориентированная модель использует наследование для защиты целостности базы данных. Однако объекты OODM содержат большее количество типов связей, а сами связи являются более сложными.

Структурная независимость и независимость по данным. Автономия объекта объектно-ориентированной модели гарантирует структурную независимость и независимость по данным.

6.4.3. Недостатки OODM

Несмотря на то что OODM является безусловным шагом вперед по сравнению с реляционной моделью, подавляющее число СУБД на сегодняшний день продолжают оставаться реляционными. Причин этому множество, и не последнюю роль сыграли недостатки, свойственные OODM.

Отсутствие должного уровня стандартизации. Для OODM пока не существует стандартов объектно-ориентированной модели. Различные стандарты, связанные с OODM, находятся в активной разработке, работа над ними продолжается, однако они не являются столь распространенными, как стандарты для РСУБД. Происходит сближение стандартов для РСУБД с ООСУБД, например, в стандарте SQL3 и более поздних. Особенно проблематичным является отсутствие стандартного метода доступа к данным. Этот недочет создает проблемы при доступе к данным из различных источников (различные поставщики поддерживают различные методы доступа к данным, как правило, несовместимые).

Сложная навигация доступа к данным. Метод доступа к данным напоминает стиль навигации в иерархической и сетевой моделях.

Трудность изучения. Недостаток стандартизации и трудности, вызванные навигационным стилем доступа к данным, приводят к затруднениям в изучении объектно-ориентированной модели, даже большим, чем при изучении сетевой модели. Несмотря на то что мы с легкостью используем объекты — перетаскиваем объекты на экране дисплея, не заботясь и не задумываясь о тех процессах, которые обеспечивают это действие, — моделирование данных и реализация объектно-ориентированных баз данных — это совершенно иные действия.

Объекты сложны, и тот факт, что они могут иметь большое семантическое наполнение, делает их трудными для проектирования и реализации. Работа с OODM больше похожа на объектно-ориентированное программирование, чем на проектирование данных. Это приводит к тому, что конечные пользователи считают объектно-ориентированные системы трудными для понимания и применения.

Ресурсоемкость. Объектно-ориентированные системы сложнее, чем реляционные модели. Реализация такой модели требует солидных затрат на приобретение оборудования и операционной системы. Сложность конфигурации и повышенные системные требования приводят к замедлению выполнения запросов и транзакций.

6.5. Проектирование баз данных

Проектирование баз данных, как и проектирование информационных систем, состоит из нескольких этапов. Одним из важных этапов проектирования является создание диаграмм «сущность-связь». Для этого необходимо обозначить сущности, добавить к ним атрибуты, установить ключи и объединить сущности при помощи связей. Все это можно проделывать вручную, просто рисуя соответствующие диаграммы на бумаге. В те времена, когда была предложена концепция «сущность-

связь», так и делали, однако при наличии компьютера с развитым графическим интерфейсом рисунки на бумаге отошли в прошлое. Довольно быстро был создан целый класс программных продуктов, позволяющий не только выполнять моделирование в парадигме «сущность-связь», но и генерировать на основе созданных моделей схему базы данных для практически любых распространенных серверов баз данных. Одной из таких программ является ErWin Data Modeler (рис. 6.18).

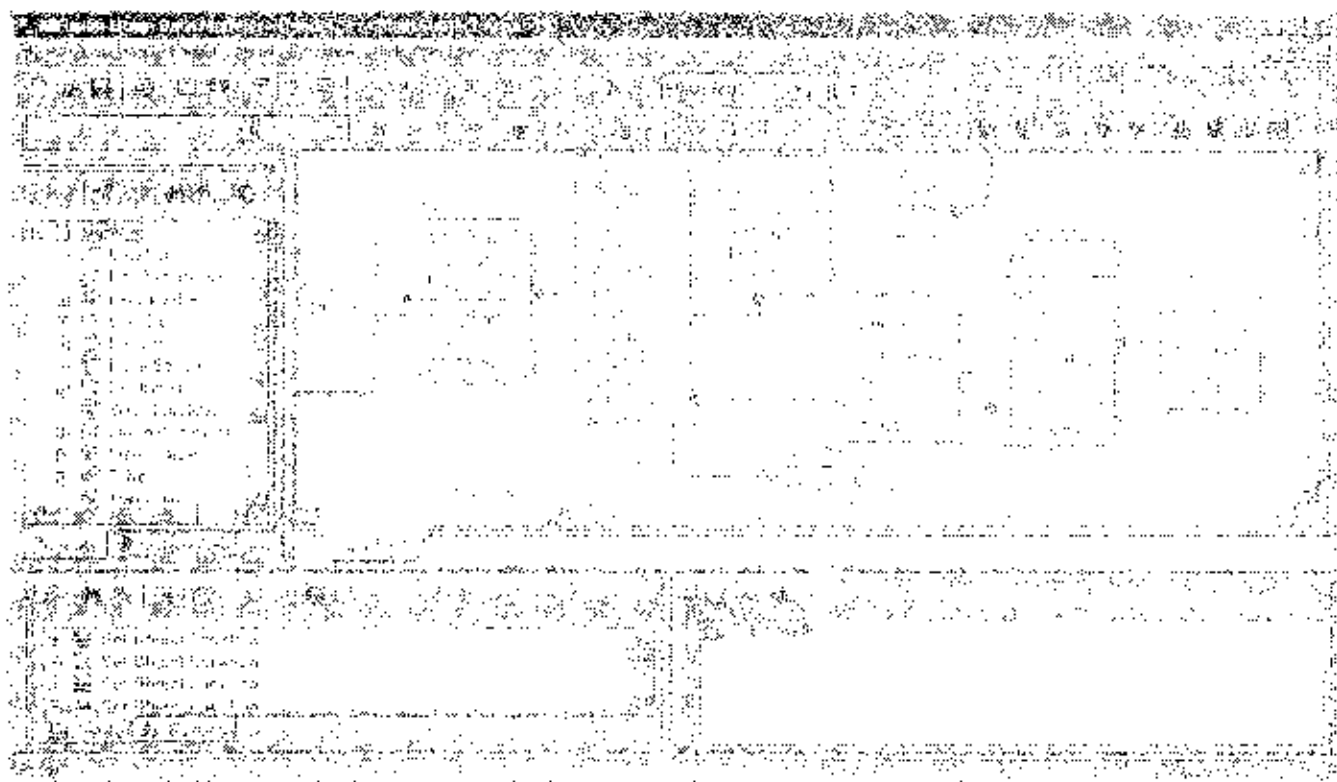


Рис. 6.18. Окно программы ErWin Data Modeler

При помощи этой программы можно быстро и легко создавать сущности, присваивать им атрибуты, ключи, создавать между ними связи. После того как схема создана, можно выбрать конкретный сервер баз данных и построить схему базы данных. При этом ErWin Data Modeler переводит созданную логическую модель на физический уровень представления и генерирует схему в соответствии с набором типов данных на конкретном выбранном сервере. Этот процесс носит название *прямого проектирования* (forward engineering).

Интересной особенностью ErWin Data Modeler является также возможность осуществить обратную операцию: извлечь схему из выбранного сервера баз данных и отобразить ее в качестве своей модели. Извлеченную схему можно затем изменить и снова осуществить прямую генерацию.

Надо отметить, что современные серверы баз данных часто оснащаются средствами моделирования модели «сущность-связь» или другими средствами создания схем баз данных.

Вопросы для самопроверки

1. Что такое «предметная область»?
2. Какие определения понятия «база данных» вам известны? Приведите минимум два.
3. Что такое «СУБД»?
4. Из каких компонентов состоит СУБД?
5. Каковы основные группы пользователей СУБД?
6. Что такое «модель данных»?
7. Какие виды классификаций СУБД вам известны?
8. Как СУБД классифицируются в зависимости от принятой модели данных?
9. Что такое «распределенная СУБД»?
10. Чем клиент-серверная СУБД отличается от встраиваемой?
11. Какие функции выполняет СУБД?
12. Какие модели данных вам известны?
13. Из каких уровней состоит трехуровневая модель представления данных ANSI-SPARC?
14. Что такое «элемент данных»?
15. Что такое «запись»? Что такое «схема записи»?
16. Зачем нужно ключевое поле?
17. Что такое «сущность» в модели «сущность-связь»? Приведите пример.
18. Что такое «класс сущностей»?
19. Что такое «атрибуты и идентификаторы сущностей»?
20. Что такое «связи сущностей», что они определяют? Какие типы связей вам известны?
21. Чем отличается связь *один ко многим* от связи *многие ко многим*? Приведите примеры.
22. Каковы преимущества и недостатки ER-моделирования?
23. Что такое «реляционная модель данных»? Кем она была предложена?
24. Назовите по крайней мере две системы, применяющиеся для непрерывного моделирования?
25. Каким условиям должны удовлетворять данные в реляционных таблицах?
26. Перечислите хотя бы 9 из 12 правил Кодда.
27. Что такое «первичный ключ»?
28. Что такое «внешний ключ», какова его роль в установлении связи между таблицами?
29. Что такое «ссылочная целостность»? Каковы последствия несоблюдения ссылочной целостности для базы данных?

30. В чем заключаются первая, вторая и третья нормальные формы?
31. Какие группы операторов языка SQL вам известны?
32. Какие операторы входят в группу операторов манипулирования данными? Каково их назначение?
33. Перечислите преимущества реляционной модели.
34. Каковы причины возникновения постреляционных моделей СУБД?
35. Чем OODM отличается от ER-модели?
36. Перечислите преимущества и недостатки OODM.

Литература

1. *Кренке Д.* Теория и практика построения баз данных. СПб.: Питер, 2003.
2. *Кузин А. В., Левонисова В. С.* Базы данных. М.: Академия, 2008.
3. *Марков А. С., Лисовский К. Ю.* Базы данных: введение в теорию и методологию. М.: Финансы и статистика, 2006.
4. *Райордан Р.* Основы реляционных баз данных. М.: Русская редакция, 2001.

Глава 7 Основы моделирования

- 7.1. Представление о назначении и особенностях моделирования
- 7.2. Классификация моделей
- 7.3. Основные этапы компьютерного моделирования
- 7.4. Основы имитационного моделирования
- 7.5. Программные среды моделирования

В этой главе дано определение понятию моделирования, описаны способы построения, назначение и области применения различных моделей. Особое внимание уделено компьютерному моделированию: описаны основные этапы создания и использования моделей на компьютере, а также современные программные средства, позволяющие осуществлять компьютерное моделирование в различных отраслях науки и техники.

7.1. Представление о назначении и особенностях моделирования

7.1.1. Понятия модели и моделирования

Моделирование в научных исследованиях стало применяться еще в глубокой древности и постепенно захватывало все новые области научных знаний: техническое конструирование, строительство и архитектуру, астрономию, физику, химию, биологию и, наконец, общественные науки. Постепенно стала осознаваться роль моделирования как универсального метода научного познания. Большие успехи и признание практически во всех отраслях современной науки получило моде-

лирование в XX в. Термин «модель» широко используется в различных сферах человеческой деятельности и имеет множество смысловых значений.

Модель — это описание или объект-заместитель объекта-оригинала, обеспечивающий изучение выбранных свойств оригинала в условиях, когда использование оригинала по тем или иным причинам невозможно.

Под *моделированием* понимается процесс построения, изучения и применения моделей. Моделирование является одной из форм отражения действительности. Моделирование тесно связано с такими категориями, как абстракция, аналогия, гипотеза и др. Процесс моделирования обязательно включает в себя и построение абстракций, и умозаключения по аналогии, и разработку научных гипотез. Главная особенность моделирования состоит в опосредованном познании с помощью объектов-заместителей. Модель выступает как своеобразный инструмент познания, который исследователь ставит между собой и объектом и с помощью которого изучает интересующий его объект. Понятие модели широко используется не только в науке и технике, но и в искусстве, и в повседневной жизни.

Возможности моделирования, то есть переноса результатов, полученных в ходе построения и исследования модели, на оригинал, основаны на том, что модель в определенном смысле отображает (воспроизводит, моделирует, описывает, имитирует) некоторые интересующие исследователя свойства объекта. Применительно к естественным и техническим наукам принято различать следующие виды моделирования:

- *Концептуальное моделирование.* При таком моделировании совокупность уже известных фактов или представлений относительно исследуемого объекта или системы истолковывается с помощью некоторых специальных знаков, символов, операций над ними или с помощью естественного или искусственного языка.
- *Физическое моделирование.* В этом случае модель и моделируемый объект представляют собой реальные объекты или процессы единой или различной физической природы, причем между процессами в объекте-оригинале и модели имеют место некоторые соотношения подобия, вытекающие из схожести физических явлений.
- *Структурно-функциональное моделирование.* Моделями являются схемы (блок-схемы), графики, чертежи, диаграммы, таблицы, рисунки, дополненные специальными правилами их объединения и преобразования.
- *Математическое (логико-математическое) моделирование.* Моделирование, включая построение модели, осуществляется средствами математики и логики.
- *Имитационное (программное) моделирование.* Логико-математическая модель исследуемого объекта представляет собой алгоритм функционирования системы, реализованный в виде программного комплекса.

Перечисленные виды моделирования не являются взаимоисключающими и могут применяться при исследовании сложных объектов либо одновременно, либо в некоторой комбинации. Кроме того, в определенном смысле концептуальное

и структурно-функциональное моделирование перазличимы, так как блок-схемы, конечно же, вполне можно считать специальными знаками с установленными операциями над ними.

7.1.2. Компьютерное моделирование

Традиционно под моделированием на компьютере понималось лишь имитационное моделирование. К настоящему времени компьютер используется практически для всех видов моделей за исключением физического моделирования. Например, при математическом моделировании выполнение одного из основных этапов – построения математических моделей по экспериментальным данным – в настоящее время просто немыслимо без компьютера. В последние годы, благодаря развитию графического интерфейса и графических пакетов, широкое развитие получило компьютерное, структурно-функциональное моделирование. Положено начало использованию компьютера даже при концептуальном моделировании, например, с целью построения систем искусственного интеллекта.

Таким образом, понятие «компьютерное моделирование» значительно шире традиционного понятия «моделирование на ЭВМ» и нуждается в уточнении, учитывающем сегодняшние реалии.

В настоящее время под *компьютерной моделью* чаще всего понимают следующее:

- условный образ объекта или некоторой системы объектов (или процессов), описанный с помощью взаимосвязанных компьютерных таблиц, блок-схем, диаграмм, графиков, рисунков, анимационных фрагментов, гипертекстов и т. д. и отображающий структуру и взаимосвязи между элементами объекта. Компьютерные модели такого вида мы будем называть структурно-функциональными;
- отдельную программу, совокупность программ, программный комплекс, позволяющий с помощью последовательности вычислений и графического отображения их результатов воспроизводить (имитировать) процессы функционирования объекта, системы объектов при условии воздействия на объект различных факторов. Такие модели определяют как имитационные.

Компьютерное моделирование является методом решения задачи анализа или синтеза сложной системы на основе использования ее компьютерной модели.

Суть компьютерного моделирования заключена в получении количественных и качественных результатов по имеющейся модели. Качественные выводы, получаемые по результатам анализа, позволяют обнаружить неизвестные ранее свойства сложной системы: ее структуру, динамику развития, устойчивость, целостность и др. Количественные выводы используются для прогноза будущих или объяснения прошлых значений переменных исследуемого объекта.

Основные области применения компьютера при моделировании:

- вспомогательное средство для решения задач;
- постановка и решение новых задач, не решаемых традиционными методами, алгоритмами, технологиями;
- разработка компьютерных обучающих и моделирующих сред;

- получение новых знаний в ходе моделирования;
- «обучение» разработанных моделей (самообучающиеся модели).

Компьютерное моделирование становится новым инструментом, методом научного познания, новой технологией еще и из-за возрастающей потребности в переходе от исследования линейных математических моделей систем к более сложным и плохо формализуемым системам.

Предметом компьютерного моделирования могут быть любые *сложные системы*, например, экономическое подразделение фирмы или банка, промышленное предприятие, информационно-вычислительная сеть, технологический процесс, любой реальный объект или процесс и т. д. Цели компьютерного моделирования могут быть различными. Наиболее часто моделирование является центральной процедурой системного анализа, причем под системным анализом мы далее понимаем совокупность методологических средств, используемых для подготовки и принятия решений экономического, организационного, социального, технического и иного характера.

Компьютерная модель сложной системы должна по возможности отображать все основные факторы и взаимосвязи, характеризующие реальные ситуации, критерии и ограничения. Модель должна быть достаточно универсальной, чтобы описывать близкие по назначению объекты, и в то же время достаточно простой, чтобы обеспечивать выполнение необходимых исследований с разумными затратами.

7.1.3. Параметры модели

Основные параметры, на основе которых оценивается модель, представлены на рис. 7.1.

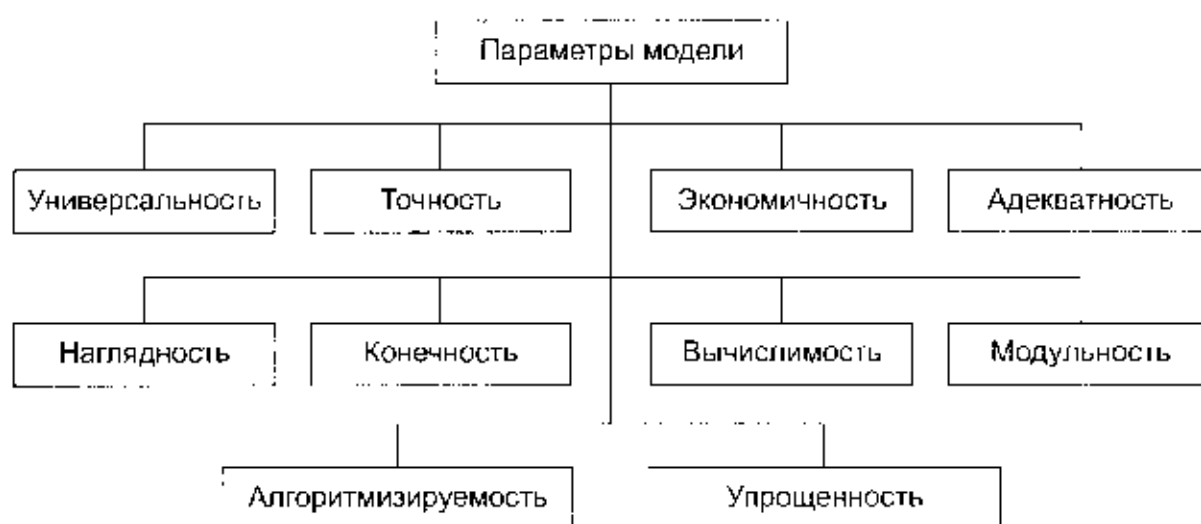


Рис. 7.1. Параметры модели

Универсальность модели характеризует полноту отражения в ней свойств реального объекта, поскольку модель отражают не все, а лишь некоторые его свойства.

Точность модели оценивается степенью совпадения (погрешностью) значений выходных параметров реального объекта и значений тех же параметров, рассчи-

танных с помощью модели. Пусть отражаемые в математической модели свойства объекта представлены вектором выходных параметров

$$Y = (y_1, y_2, \dots, y_n).$$

Здесь y_i — истинное значение i -го параметра. Обозначим через y_{im} i -й параметр, рассчитанный с помощью модели. Тогда относительная погрешность математической модели по i -му параметру будет:

$$E_i = \frac{|y_{im} - y_i|}{y_i}.$$

По этой формуле рассчитываются погрешности для каждого выходного параметра, в результате получается вектор погрешностей

$$E = (E_1, E_2, \dots, E_n).$$

Адекватность модели является мерой совпадения функциональных характеристик модели с функциональными характеристиками моделируемого объекта.

Экономичность модели характеризуется затратами вычислительных ресурсов на ее реализацию. Если работа с моделью осуществляется вручную, то ее экономичность определяется затратами времени разработчика. При автоматизированном проектировании модели — затратами машинного времени и памяти компьютера, а также затратами времени разработчика. Часто для оценки экономичности непосредственно компьютерной модели используют другие величины:

- среднее количество операций, выполняемых при одном обращении к модели;
- размерность системы уравнений в математической модели;
- количество используемых в модели внутренних параметров и т. д.

Вычислимость определяется как возможность ручного или компьютерного исследования качественных и количественных закономерностей функционирования объекта (системы).

Модульность показывает соответствие конструкций модели структурным составляющим объекта (системы).

Алгоритмизируемость характеризует возможность разработки соответствующих алгоритма и программы, реализующей математическую модель на ЭВМ.

Наглядность отражает удобство визуального восприятия модели.

Конечность показывает отображение оригинала лишь в конечном числе его отношений.

Упрощенность (приблизительность) говорит об отражении только существенных сторон объекта (системы).

Следует отметить, что одна группа параметров модели (упрощенность, конечность, экономичность) обычно вступает в противоречие с такими параметрами, как наглядность, адекватность и точность. Разрешение этого противоречия с целью получения оптимальной модели является одной из задач моделирования.

7.2. Классификация моделей

Существует множество способов классифицировать модели. Большой выбор способов классификации обусловлен тем, что моделирование применяется практически во всех областях деятельности человека. Под понятие моделирования попадает широкий диапазон человеческих действий и артефактов. Само человеческое мышление представляет собой непрерывное моделирование окружающего мира.

В этом разделе представлены разнообразные подходы к классификации моделей с разных точек зрения.

7.2.1. Классификация моделей по назначению

Классификацию моделей по назначению иллюстрирует рис. 7.2.



Рис. 7.2. Классификация моделей по назначению

Познавательная модель является формой организации и представления знаний, средством объединения новых и старых знаний. Познавательная модель, как правило, с максимально возможной точностью отображает реальность и изменяется в соответствии с изменением реальности. Является теоретической моделью.

Пример. Математическое моделирование мирового океана с целью изучения изменения течений и рельефа океанского дна. Разрабатывается теория, согласно этой теории строится модель. Если поведение модели плохо согласуется с процессами в реальном объекте, уточнению подлежат теория и построенная на ее основе модель.

Прагматическая модель является средством организации практических действий, рабочего представления целей системы для ее управления. Реальность подстраивается под некоторую прагматическую модель (как правило, прикладную).

Пример. Выбор модели финансового регулирования в стране. Если выбрана монетаристская модель, то все процессы финансово-валютного регулирования стараются согласовать с этой моделью. Если процессы, происходящие в финансовой сфере страны, не отвечают параметрам модели, то производятся действия, изменяющие процессы таким образом, чтобы они соответствовали с выбранной моделью.

Инструментальная модель является средством построения, исследования и (или) использования *прагматических* и (или) *познавательных* моделей.

Пример. После построения теоретической математической модели мирового океана она оформляется в виде компьютерной модели на языке программирования. Таким образом, инструментальная модель оказывается моделью модели, средством инструментальной реализации познавательной или прагматической модели.

7.2.2. Классификация моделей по уровню моделирования

Классификацию моделей по уровню моделирования иллюстрирует рис. 7.3.

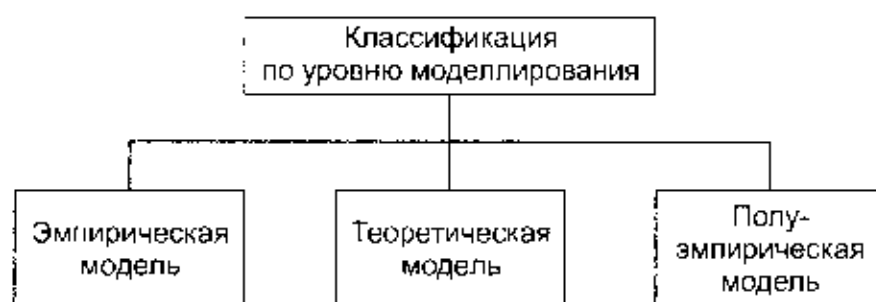


Рис. 7.3. Классификация моделей по уровню моделирования

Эмпирическая модель построена на основе установленных опытным путем зависимостей между входными и выходными параметрами модели. Эмпирические модели создаются в тех случаях, когда явление или процесс невозможно описать при помощи математических формул, поскольку о внутреннем устройстве объекта или механизме процесса ничего не известно либо внутренние зависимости являются слишком сложными для построения математического описания.

Пример. Все модели процессов, происходящих в человеческом обществе — социальных, экономических, финансовых, политических, — строятся эмпирически.

Теоретическая модель построена на основе математически описанных зависимостей между входными и выходными параметрами модели. В этом случае все внутренние механизмы явления известны настолько, чтобы можно было с достаточной точностью описать их с помощью математического аппарата.

Пример. Компьютерная модель простого физического процесса: растягивания идеальной пружины под действием груза (идеальный маятник).

Полуэмпирическая модель построена на основе аппроксимаций эмпирических зависимостей при помощи математических функций с удовлетворяющей за-

задам моделирования точностью. В случае полужемпирической модели объект моделирования (прототип) достаточно сложен, и внутренние механизмы его функционирования не могут быть в точности описаны при помощи математических функций. Однако опыт наблюдения за объектом позволяет установить закономерности между входными и выходными параметрами, которые можно с достаточной точностью описать (аппроксимировать) при помощи математических функций.

Пример. Компьютерная модель процесса обмена веществ в биологической клетке.

7.2.3. Классификация моделей по принадлежности к иерархическому уровню

Классификацию моделей по принадлежности к иерархическому уровню иллюстрирует рис. 7.4.



Рис. 7.4. Классификация по иерархическим уровням моделирования

Модель микроуровня отображает объекты или процессы самого нижнего, неделимого на составные части уровня в иерархической структуре. Модели микроуровня создаются как составные части модели макроуровня с целью более точного воспроизведения моделируемого прототипа.

Пример. Модель технологического процесса на предприятии.

Модель макроуровня отображает объекты или процессы среднего или высшего звена в иерархической структуре.

Пример. Модель работы сборочного цеха или предприятия.

Модель метауровня отображает процессы или объекты, взаимодействующие с прототипом модели макроуровня. Цель моделирования на метауровне - более точное воспроизведение среды (входных параметров) модели макроуровня.

Пример. Модель функционирования предприятия во взаимосвязи с государственными органами, поставщиками, потребителями, общественностью и окружающей средой.

7.2.4. Классификация моделей по характеру взаимоотношений со средой

Классификацию моделей по характеру взаимоотношений со средой иллюстрирует рис. 7.5.

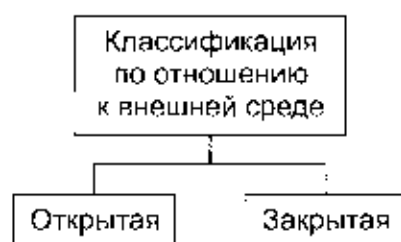


Рис. 7.5. Классификация по характеру отношения к внешней среде

Открытая модель осуществляет непрерывный энергоинформационный и вещественный обмен со средой.

Пример. Действующая модель водяной мельницы в уменьшенном масштабе.

Закрытая модель имеет слабую связь с внешней средой или вообще ее не имеет.

Пример. Компьютерная модель движения колеса по наклонной поверхности в отсутствие силы трения.

7.2.5. Классификация моделей по способу представления свойств объекта

Классификацию моделей по способу представления свойств объекта иллюстрирует рис. 7.6.

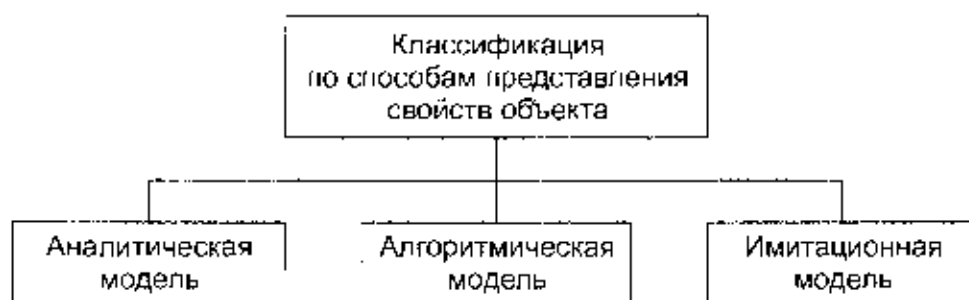


Рис. 7.6. Классификация по способу представления свойств объекта

Аналитическая модель полностью определяется через совокупность математических функций.

Пример. Модель работы часового механизма может быть полностью описана аналитически как совокупность функций угла поворота от времени.

Алгоритмическая модель описывается алгоритмом или комплексом алгоритмов, определяющим ее функционирование и развитие.

Пример. Типичным случаем алгоритмического моделирования являются продукционные экспертные системы, моделирующие поведение эксперта при принятии решений в той или иной предметной области при помощи набора алгоритмов (правил).

Имитационная модель строится для испытания, изучения или воспроизведения возможных путей развития и поведения объекта путем варьирования некоторых или всех параметров модели. Название «имитационная» модель получила, поскольку позволяет имитировать поведение реальных сложных систем без детального описания внутреннего механизма этого поведения.

В случае математической имитационной модели сложная система представляется в виде совокупности элементов, часть из которых может быть описана аналитически (функциональными зависимостями), а часть представляет собой «черные ящики», функционирование которых аппроксимируется вероятностными зависимостями.

Имитационные модели могут быть не только математическими, они могут реализовываться самыми разными способами, в том числе с помощью макетов или путем игрового моделирования.

Пример. Игровая реконструкция знаменитых военных сражений (например, Бородинской битвы) является очевидным примером имитационного моделирования. По части известных фактов и описаний процессов в ходе имитации может быть реконструирована картина сражения, близкая к реальным историческим событиям.

7.2.6. Классификация моделей по причинной обусловленности

Классификацию моделей по причинной обусловленности иллюстрирует рис. 7.7. *Детерминированная модель* позволяет однозначно определять набор выходных параметров для каждой допустимой совокупности входных параметров.

Пример. Модель функционирования лифтового механизма является детерминированной.

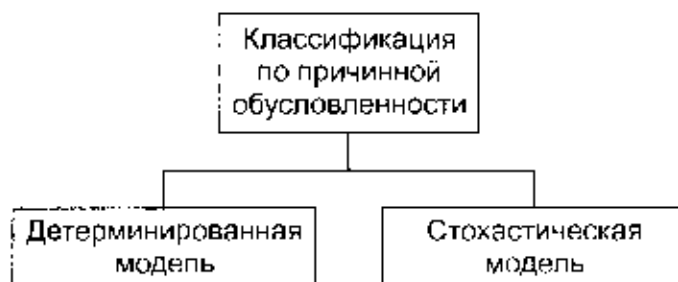


Рис. 7.7. Классификация по причинной обусловленности

Недетерминированная, или стохастическая (вероятностная), модель предполагает вероятностную природу входных параметров так же, как и вероятностную природу функций (или алгоритмов) их обработки. Таким образом набор выходных параметров в стохастической модели приобретает вероятностный характер.

Пример. Модель земной атмосферы, которая строится с целью формирования долгосрочного прогноза погоды и предупреждения стихийных бедствий, носит стохастический характер.

7.2.7. Классификация моделей по отношению ко времени

Классификацию моделей по отношению ко времени иллюстрирует рис. 7.8.



Рис. 7.8. Классификация по отношению ко времени

Динамическая модель в явной форме использует время в качестве одного из входных параметров. Обычно динамическая модель может быть «проиграна» во времени с некоторым масштабированием (замедлением или ускорением).

Пример. Модель развития колонии простейших микроорганизмов.

Статическая модель определяет модель, у которой параметр времени в явной форме среди входных параметров не присутствует. Статические модели обычно используют для отыскания граничных или оптимальных значений тех или иных параметров.

Пример. Модель воздушного судна для обдува в аэродинамической трубе.

7.2.8. Классификация моделей по сфере применения

Классификацию моделей по сфере применения иллюстрирует рис. 7.9.



Рис. 7.9. Классификация по сфере применения

Разделение моделей по сферам применения вызвано не столько особенностью самих моделей (принципы моделирования остаются одинаковыми независимо от области применения модели), сколько спецификой сбора и подготовки исходного материала для моделирования и специфическими особенностями описания предметной области.

7.2.9. Классификация моделей по методологии применения

Классификацию моделей по методологии применения иллюстрирует рис. 7.10.

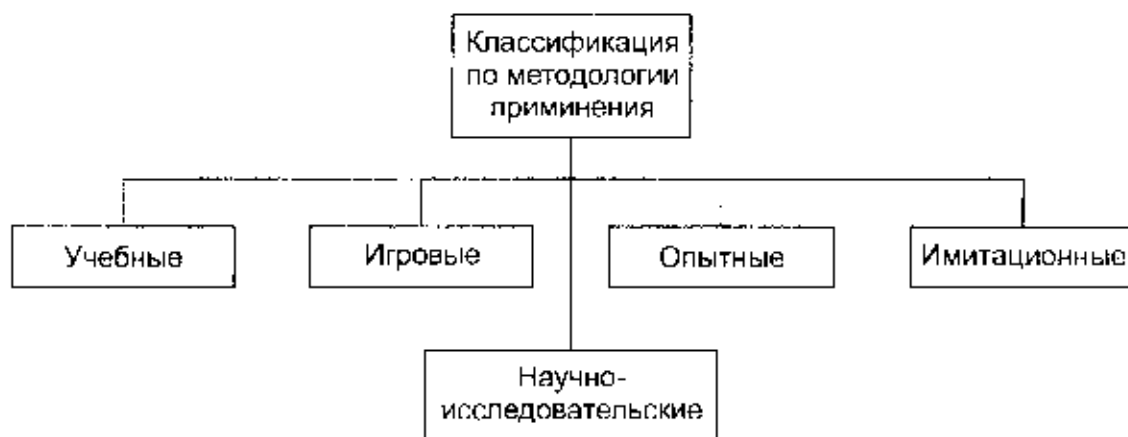


Рис. 7.10. Классификация по методологии применения

Учебная модель создается для поддержки учебного процесса. Учебные модели обычно частично воспроизводят функциональность объекта или детали процесса, которые невозможно наблюдать и изучать при рабочем функционировании объекта моделирования.

Пример. Модель пищеварительного тракта человека, модель электрической системы автомобиля, модель клетки биологической ткани.

Игровая модель в игровой форме или ситуации воспроизводит процессы, происходящие в сложной системе. Игровые модели чаще всего разрабатываются для тренинга навыков и умений. Игровая модель может строиться спонтанно или организованно.

Пример. Детская игра, воспроизводящая в игровой форме семейные отношения, деловая игра, направленная на выявление конфликтных ситуаций на предприятии и нахождение путей их разрешения.

Научно-исследовательская модель строится для изучения явлений, которые невозможно произвольно повторить в живой природе.

Пример. Компьютерная модель фрагмента земной коры, построенная для изучения способов прогнозирования землетрясений.

Опытная модель строится с целью воспроизведения свойств искусственного объекта и изучения его поведения в различных условиях. Опытная модель в некоторых случаях может быть сложнее и дороже, чем объект моделирования.

Пример. Опытная модель микропроцессорного устройства, построенная путем компьютерного моделирования. Такая модель может в целом обойтись дороже и сложнее, чем создание одного кристалла микропроцессора, но оправдывает себя, поскольку позволяет предотвратить ошибки в устройстве, которое будет изготовлено в количестве несколько миллионов штук.

Имитационная модель служит для имитации поведения или процессов в сложных системах. Определение и пример имитационной модели уже были приведены ранее в этом разделе.

7.2.10. Классификация моделей по способу представления

Классификацию моделей по способу представления иллюстрирует рис. 7.11.

Материальная модель по своей физической структуре, форме, энергетическим характеристикам воспроизводит моделируемый объект. Для материальной модели характерно непосредственное, в материальной, а не информационной форме, воспроизведение тех или иных особенностей прототипа.

Пример. Скульптура, уменьшенная модель поезда, модель ветряной мельницы, увеличенная модель клетки биологической ткани.



Рис. 7.11. Классификация по способу представления

Информационная модель представляет собой модель, в которой в качестве механизма создания модели выступает информация. Информационные модели могут быть *неформализованными* (например, мысленная модель или абстрактная живопись) и *формализованными* (то есть воплощенными в форме символов, высказываний, рисунков или чертежей, значение которых оговорено).

В свою очередь, формализованная модель может быть *компьютерной* и *некомпьютерной*.

Пример. Мысленное представление модели электрической машины является неформализованной информационной моделью. Мысленные эксперименты с такими моделями — известный факт из биографии знаменитого физика и изобретателя Никола Тесла. Однако мысленное представление модели электрической машины не может быть использовано при ее серийном производстве, поэтому мысленная модель формализуется, переводится на язык понятных другим людям символов или рисунков (чертежей). Таким образом создается формализованная модель. Формализованная модель, созданная при помощи компьютера, является компьютерной. Формализованная модель, созданная без участия компьютерной техники, является некомпьютерной.

ВНИМАНИЕ

Важно понимать разницу между материальной и информационной моделями. Носитель информации (бумага, на которой выполняется рисунок, или компьютер, моделирующий работу сложного механизма) всегда материален. Однако средства моделирования в информационной модели имеют нематериальную природу: образы, символы, речевые конструкции, алгоритмы, функциональные зависимости. Компьютерная модель не является материальной.

Количество приведенных способов классификации моделей показывает, насколько богат и разнообразен мир моделирования. Когда речь идет о компьютерных моделях, трудно отнести модель к одной какой-то группе, например, компьютерная игра оказывается одновременно и математической, и геометрической, и игровой, и цифровой (на определенном уровне представления), и аналоговой (на другом уровне), и дискретной, и непрерывной, и алгоритмической, и имитационной моделью. Невозможность однозначной классификации свойственна многим моделям, связанным с использованием при моделировании систем искусственного интеллекта, а также систем, основанных на знаниях, о которых мы говорили в предыдущей главе.

Кроме того, мир компьютерного моделирования быстро эволюционирует, вбирая в себя многие черты моделирования физического, и, зачастую, вытесняя физические модели. Если раньше архитектор создавал макеты зданий и размещал их на макете ландшафта, то сегодня и здание, и ландшафт легко создаются в программах трехмерного архитектурного моделирования (ArchiCAD). На компьютере строятся модели одежды и модели причесок, модели внешности человека, модели автомобилей. Моделируются как предметы физического мира, так и технологические или информационные процессы. Сами компьютерные программы также стали предметом моделирования, разработана целая технология программирования (объектно-ориентированное программирование), в основе которой лежит создание абстрактных информационных моделей объектов окружающего мира.

7.3. Основные этапы компьютерного моделирования

Необходимость моделирования определяется тем, что многие объекты и процессы непосредственно исследовать либо практически невозможно, либо это исследование требует много времени и средств.

Поскольку сегодня компьютерное моделирование успешно применяется в тех областях, где ранее было возможно применение только материальных моделей (моделирование зданий, объектов техники, одежды, и т. д.), в этом разделе речь пойдет об этапах построения и использования компьютерной модели.

Моделирование заключается в имитации изучаемого явления. Точность имитации определяется путем сравнения полученного при воспроизведении результата с его прототипом, объектом исследования, и оценки степени их сходства.

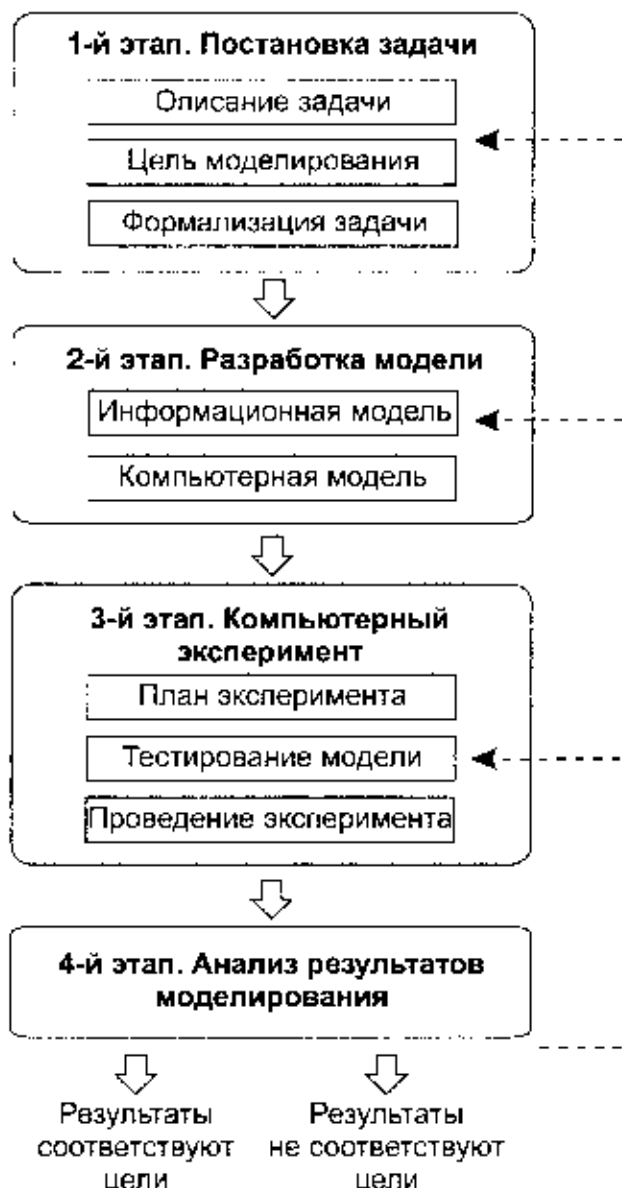


Рис. 7.12. Основные этапы компьютерного моделирования

В целом моделирование включает в себя четыре этапа (рис. 7.12):

1. Постановка задачи.
2. Разработка модели.
3. Компьютерный эксперимент.
4. Анализ результатов моделирования.

7.3.1. Постановка задачи и анализ объекта моделирования

Задача формулируется в общем виде, в форме словесных описаний, достаточных для того, чтобы определить цель моделирования. При этом модель и ее характеристики напрямую зависят от поставленной исследователем цели. Создание модели автомобиля может преследовать три разные цели.

Пример. Модель 1: разработать внешний дизайн автомобиля.

Модель 2: установить предельные прочностные характеристики автомобиля при движении по пересеченной местности.

Модель 3: разработать оптимальный режим подачи топлива в двигатель автомобиля в зависимости от температурного режима окружающей среды и режима движения.

Несмотря на то что во всех трех случаях материальный объект (прототип) для моделирования один и тот же, цели моделирования разные, соответственно, и вид модели определяется заданной целью.

Пример. Модель 1: трехмерная геометрическая модель автомобиля, отображающая текстуру и цвет покрытия, а также учитывающая дизайн осветительной системы.

Модель 2: модель автомобиля, имитирующая прочностные характеристики его основных узлов при движении в различных режимах с учетом максимальных нагрузок.

Модель 3: модель топливной системы автомобиля с возможностью изменения параметров подачи топлива и режимов работы двигателя.

После определения цели моделирования следует этап тщательного исследования модели и описания полученных при исследовании результатов на формальном языке. На этом этапе формируется возможно более полное описание объекта: выделяются его элементы, устанавливаются связи между ними, выделяются существенные для исследования характеристики, выявляются параметры, изменение которых влияет или может влиять на объект. На том же этапе формируются подлежащие последующей проверке гипотезы о закономерностях, присущих изучаемому объекту, о характере влияния на него изменения тех или иных параметров и связей между его элементами.

7.3.2. Разработка (формализация и синтез) модели

На основе анализа результатов, полученных на предыдущем этапе, формируется информационная модель. На данном этапе исходные предположения переводятся на однозначный (обычно математический) язык количественных отношений и устраняются нечеткие, неоднозначные высказывания или определения. Завершается этап получением информационной модели объекта. Информационная модель может быть представлена в виде таблиц с данными, набора математических отношений, диаграмм и графиков, логических высказываний и других формальных описаний. Для каждой из трех моделей автомобиля требуются следующие данные.

Пример. Модель 1: геометрические размеры автомобиля, список допустимых материалов и покрытий, требования к максимальным и минимальным размерам тех или иных узлов.

Модель 2: схема крепления механических узлов автомобиля, описание прочностных характеристик отдельных узлов и деталей в математической форме (аналитической или иной), таблица прочностных характеристик различных материалов.

Модель 3: описание в математической форме характеристик движения топлива по топливопроводу, таблица эмпирических замеров зависимости скорости подачи топлива от режимов работы двигателя и температуры внешней среды.

После того как все необходимые исходные данные собраны и формализованы, производится построение компьютерной модели.

Пример. Модель 1: построение компьютерной модели трехмерного дизайна автомобиля с возможностью вращения модели в любой плоскости, изменения освещения. Возможно построение вариантов дизайна с целью экспертной оценки наиболее подходящей модели.

Модель 2: построение имитационной модели работы механической части автомобиля во время движения по пересеченной местности. В имитационной модели такого рода большое количество составляющих модель блоков может быть описано аналитическими функциями (поскольку методика расчета прочностных характеристик узлов и механизмов из разных материалов детально проработана). Случайный характер будет носить изменение нагрузки во времени (имитирующее движение по ухабам).

Модель 3: построение модели взаимодействия двигателя и топливной системы автомобиля. В этом случае в имитационной модели будет множество элементов со стохастическим и аппроксимированным характером моделирования, поскольку вариации качества топлива, изменения вязкости масла, прохождения горюче-смазочных материалов по топливной системе, процесса сгорания горючего в двигателе можно оценить только приблизительно.

7.3.3. Проведение компьютерного эксперимента

В зависимости от того, какой тип моделирования был выбран в соответствии с целью моделирования, компьютерный эксперимент может носить как кратко-

временный, так и долговременный характер, а его результаты можно использовать непосредственно после получения или же они послужат основой для уточнения и корректировки модели.

Пример. Модель 1: ряд моделей, разработанных дизайнером, просматривается экспертной группой, и выбирается лучшая модель.

Модель 2: прочностная модель автомобиля «проигрывается» путем изменения входных параметров, имитирующих дорогу с разным рельефом.

Модель 3: модель топливной системы автомобиля «проигрывается» при разных режимах работы двигателя и различной температуре внешней среды.

7.3.4. Анализ результатов моделирования

Анализ результатов моделирования заключается в установлении адекватности модели объекту исследования, то есть в определении степени сходства модели с ее оригиналом. Успешный результат сравнения (оценки) исследуемого объекта с моделью свидетельствует о достаточной степени изученности объекта, о правильности принципов, положенных в основу моделирования, о правильности алгоритма, моделирующего объект, то есть о том, что созданная модель работоспособна. Такая модель может быть использована для дальнейших более глубоких исследований объекта в новых условиях, в которых реальный объект еще не изучался.

Чаше всего первые результаты моделирования не удовлетворяют предъявленным требованиям. Это означает, что в одной из перечисленных позиций (изученность объекта, исходные принципы, алгоритм) имеются ошибки и неточности. Это требует проведения дополнительных исследований и соответствующего изменения компьютерной программы моделирования.

7.4. Основы имитационного моделирования

Существует класс объектов, для которых по различным причинам не разработаны аналитические модели либо не разработаны методы решения полученной модели. Сложность процессов, происходящих в неидеальных системах, часто делает невозможной формализацию. В этом случае, когда необходимо провести исследование сложных систем, строится имитационная модель. Экспериментирование с такой моделью называют имитацией.

Пример. Работа системы подачи топлива и смазки в автомобиле даже в идеальном случае (постоянная температура, постоянные свойства топлива и смазки) с трудом может быть описана при помощи аналитических методов. Если же речь идет о реальной системе, где плотность и вязкость жидкостей неоднородны и меняются в зависимости от множества факторов, где есть изменения температуры, построение аналитической модели системы становится невозможным.

Суть имитационного моделирования состоит в следующем:

1. Система разбивается на большое количество функциональных блоков (декомпозиция).
2. Каждый блок заменяется моделью «черного ящика» с набором входов и выходов и функцией зависимости выходных параметров от входных.
3. В качестве функции преобразования может выступать аналитическое выражение, вероятностная функция или аппроксимирующая функция на основе экспериментальных данных.
4. Модели функциональных блоков объединяются в модель системы, которая имитирует поведение реальной сложной системы.

Основой имитационного моделирования являются процессы декомпозиции, определения функции каждого блока, преобразующей входные параметры во входные, и последующего построения модели, основанного на синтезе системы из полученных моделей функциональных блоков.

7.4.1. Особенности имитационного моделирования

Все имитационные модели представляют собой модели типа «черного ящика». Это означает, что они обеспечивают выдачу входного сигнала системы, если на ее взаимодействующие подсистемы поступает входной сигнал. Поэтому для получения необходимой информации или результатов необходимо осуществлять «прогон» имитационных моделей, а не «решать» их. Имитационные модели не способны формировать решение в том виде, в каком это имеет место в аналитических моделях, а могут служить лишь средством анализа поведения системы в условиях, которые определяются экспериментатором.

Поскольку необходимо и желательно приспособлять средство или метод к решению задачи, а не наоборот, возникает вопрос: в каких случаях имитационное моделирование件 полезно?

Исследователь должен рассмотреть целесообразность применения имитационного моделирования при наличии любого из приведенных ниже условий:

- ❑ Формализация модели невозможна либо еще не разработаны аналитические методы решения сформулированной математической модели. К этой категории относятся многие модели массового обслуживания, связанные с рассмотрением очередей.
- ❑ Аналитические методы имеются, но математические процедуры столь сложны и трудоемки, что имитационное моделирование позволяет получить более простое решение задачи.
- ❑ Аналитические решения существуют, но их реализация невозможна вследствие недостаточной математической подготовки имеющегося персонала. В этом случае следует сопоставить затраты на проектирование, испытания и работу на имитационной модели с затратами, связанными с приглашением специалистов со стороны.

- Кроме оценки определенных параметров, желательно осуществить на имитационной модели наблюдение за ходом процесса в течение определенного периода.
- ┐ Трудность постановки экспериментов и наблюдений в реальных условиях; соответствующим примером может служить изучение поведения космических кораблей в условиях межпланетных полетов.
- Для длительного действия систем или процессов может понадобиться сжатие временной шкалы. Имитационное моделирование даст возможность полностью контролировать время изучаемого процесса, поскольку явление может быть замедлено или ускорено по желанию. К этой категории относятся, например, проблемы изучения динамики развития населенных пунктов разного уровня.

7.4.2. Подходы к имитационному моделированию

Современное имитационное моделирование в зависимости от целей моделирования и решаемой задачи предполагает осуществление одного из трех подходов к моделированию:

- системная динамика;
- дискретно-событийное моделирование;
- агентное моделирование.

Системная динамика рассматривает систему как совокупность переменных, соединенных между собой петлями обратной связи. Петли обратной связи обладают нелинейными характеристиками передачи сигнала и могут быть усиливающими (положительная обратная связь) или ослабляющими (отрицательная обратная связь). Петли обратной связи также могут взаимодействовать между собой. Передача сигнала по петле обратной связи может осуществляться напрямую или с временной задержкой. Таким образом, система в целом имитирует нелинейное поведение, свойственное природным системам и процессам в них. Метод системной динамики направлен не на получение количественных результатов или детального анализа роли того или иного элемента системы, а на оценку поведения системы в целом: развивается она или деградирует, стремится к равновесию или демонстрирует неустойчивость, насколько равномерно протекает процесс развития в разных частях системы. Такой подход обычно используется для создания моделей макроуровня с объектами природного, социального или экономического плана (например, моделирование равновесия рынков или поведения атмосферы планеты). При моделировании в парадигме системной динамики часто в модель системы включается взаимодействие с внешней средой.

Дискретно-событийное моделирование описывает систему как иерархическую последовательность элементов и операций (получение, захват, разгрузка, регистрация, размещение, выдача). Переход от одной операции к другой инициируется событием (или порождает событие). Операции производятся над некоторыми

сущностями (документ, заявка, контейнер, счет). Сами по себе сущности пассивны, но обладают атрибутами, способными влиять на процесс выполнения операции. Дискретно-событийное моделирование обладает довольно высокой степенью абстракции: множество несущественных для рассматриваемого процесса деталей игнорируется. Дискретно-событийный подход применяют при моделировании бизнес-процессов, технологических процессов на производстве, в логистике.

Агентное моделирование описывает систему как совокупность сложных объектов (агентов), связанных определенными правилами. Агентное моделирование применяется для того, чтобы при рассмотрении системы перенести акцент с процессов, происходящих в системе, на поведение сложных элементов системы (агентов). В качестве агентов могут выступать люди (моделирование коллектива), предприятия и учреждения (моделирование рынка), сложные механизмы (моделирование технологических процессов). Если при дискретно-событийном моделировании все внимание сосредотачивается на операциях, а индивидуальные особенности совершающих операции элементов системы игнорируются, то при агентном моделировании различия между объектами системы становятся определяющими в поведении модели.

7.4.3. Этапы имитационного моделирования

Основные этапы имитационного моделирования совпадают со схемой, приведенной на рис. 7.12, но имеют свои особенности.

Постановка задачи

Приводится описание проблемы и формулируется цель моделирования. После того как сформулирована цель моделирования, может оказаться, что из-за сложности рассматриваемого объекта или процесса невозможно осуществить его формализацию. Тогда принимается решение проводить имитационное моделирование.

Разработка модели

Разработка модели подразумевает создание информационной модели и построение компьютерной модели.

□ *Создание информационной модели.*

Построение информационной модели начинается со *структурного анализа процессов*. Проводится формализация структуры сложного реального процесса путем разложения его на подпроцессы, выполняющие определенные функции и имеющие взаимные функциональные связи. Выявленные подпроцессы, в свою очередь, могут разделяться на другие функциональные подпроцессы.

Структура общего моделируемого процесса может быть представлена в виде графа, имеющего иерархическую многослойную структуру, в результате появляется формализованное изображение имитационной модели в графическом виде. Структурный анализ особенно эффективен при моделировании экономических процессов, где в отличие от технических многие составляющие подпроцессы

не имеют физической основы и протекают виртуально, поскольку оперируют информацией, деньгами и логикой (законами) их обработки.

Ключевыми пунктами структурного анализа процессов являются декомпозиция (правильное выделение составляющих подпроцессов) и полнота связей (все связи между подпроцессами должны быть учтены).

□ *Построение компьютерной модели.*

Графическое изображение имитационной модели, функции, выполняемые каждым подпроцессом, условия взаимодействия всех подпроцессов и особенности поведения моделируемого процесса (временная, пространственная, энергетическая или финансовая динамика) должны быть описаны на специальном языке для последующей трансляции. Для этого существуют различные способы:

- описание вручную на языке типа GPSS, Pilgrim, Visual Basic (последний очень прост, на нем можно запрограммировать элементарные модели, но он не подходит для разработки реальных моделей сложных экономических процессов, так как описание модели средствами Pilgrim компактнее аналогичной алгоритмической модели на Visual Basic в десятки-сотни раз);
- автоматизированное описание с помощью компьютерного графического конструктора во время проведения структурного анализа, то есть с очень незначительными затратами на программирование (такой конструктор, создающий описание модели, имеется, например, в составе системы моделирования в Pilgrim).

Трансляция обычно осуществляется в различных режимах:

- в режиме интерпретации (характерен для систем типа GPSS, ReThink);
- в режиме компиляции (характерен для системы Pilgrim).

Каждый режим имеет свои особенности. Режим интерпретации проще в реализации. Специальная универсальная программа-интерпретатор на основании формализованного описания модели запускает все имитирующие подпрограммы. Данный режим не приводит к получению отдельной моделирующей программы, которую можно было бы передать заказчику, так как необходима еще и среда моделирования.

Режим компиляции сложнее реализуется при создании моделирующей системы. Однако это не усложняет процесс разработки модели. В результате можно получить отдельную моделирующую программу, которая работает независимо от системы моделирования в виде отдельного программного продукта.

Верификация (тестирование) параметров модели выполняется в соответствии с гипотезой, на основании которой построена модель, с помощью специально выбранных тестовых примеров.

Компьютерный эксперимент

Проведение компьютерного эксперимента заключается в «проигрывании» имитационной модели путем задания закона изменения одного или нескольких входных параметров. При имитационном моделировании во множестве случаев

время в явном виде присутствует среди входных параметров, то есть большинство имитационных моделей являются динамическими.

Анализ результатов моделирования

Сложность имитационной модели обычно делает необходимым проведение нескольких циклов уточнения модели. По мере сравнения полученной модели с прототипом могут быть выявлены неучтенные связи или уточнены эмпирические зависимости.

7.5. Программные среды моделирования

7.5.1. Характеристика систем имитационного моделирования

Современные тенденции в области имитационного моделирования связаны с развитием проблемно-ориентированных систем, созданием встроенных средств для интеграции моделей в единый модельный комплекс. Технологический уровень современных систем моделирования характеризуется большим выбором базовых концепций формализации и структуризации моделируемых систем, развитыми графическими интерфейсами и анимационным выводом результатов. Имитационные системы обладают средствами передачи информации из баз данных и других систем или имеют доступ к процедурным языкам, что позволяет легко выполнять вычисления, связанные с планированием факторных экспериментов, автоматизированной оптимизацией и др.

Анализ рынка информационных технологий позволяет выявить основные тенденции в области современных систем моделирования. В качестве доминирующих базовых концепций формализации и структуризации в современных системах моделирования используются:

- *для дискретного моделирования* — системы, основанные на описании процессов (process description) или на сетевых концептах (network paradigms), такие как Extend, Arena, ProModel, Witness, Taylor, Gpss/H-Proof и др.;
- *для систем, ориентированных на непрерывное моделирование* — модели и методы системной динамики, такие как Powersim, Vensim, Dynamo, Stella, Ithink и др.

Причем в мощных системах с целью расширения их функциональности поддерживаются альтернативные концепции формализации. Так, например, в системе Powersim, Ithink встроен аппарат дискретного моделирования и, наоборот, в системах Extend, ProcessModel реализована поддержка непрерывного моделирования.

Большинство систем моделирования имеют удобный, легко интерпретируемый графический интерфейс, системные потоковые диаграммы или блок-схемы реализуются на идеографическом уровне, то есть рисуются, параметры моделей определяются через подменю. Для подготовленных пользователей сохраняются элементы программирования (на языках общего назначения или объектно-ориентированных)

отдельных элементов модели или специализированных блоков – так называемое авторское моделирование (например, в системе Extend существует встроенный язык Modl, предназначенный для создания специализированных блоков).

Имитационные системы становятся все более проблемно-ориентированными. Известны системы моделирования производственных систем различного назначения (TOMAC, SIRE и др.), медицинского обслуживания (MEDMODEL), телекоммуникаций (COMNET) и др. Для этого в проблемно-ориентированные системы моделирования включаются абстрактные элементы, языковые конструкции и наборы понятий, взятые непосредственно из предметной области исследований. Определенные преимущества имеют системы моделирования, декларирующие свою проблемную ориентацию, например пакет Rethink, ориентирующийся на реинжиниринг. Все это, конечно, влияет на доступность и привлекательность имитационного моделирования.

В современных системах моделирования появляется некоторый инструментарий для создания стратифицированных моделей. *Стратифицированные модели* представляют собой машинно-ориентированные понятия, предполагающие конструирование баз данных и знаний, над которыми определены вычислительные процессы решения задач системного анализа и принятия решения. Разработчики систем моделирования используют различные подходы для реализации стратифицированных моделей. Ряд программных продуктов, такие как AUTOMOD, ProModel, TAYLOR, WITNESS и др., поддерживает интеграцию моделей на основе создания вложенных структур. В системах Atena, Extend реализован подход к стратификации, основанный на построении иерархических многоуровневых структур. Наиболее перспективным является структурно-функциональный подход, реализованный, например, в системах моделирования Ithink, Rethink и базирующийся на методологии структурного анализа и проектирования. В такой технологии есть возможность для реализации нескольких уровней представления моделей: высокоуровневое представление в виде блок-схем, представление с использованием CASE-средств, а на нижнем уровне модели могут отображаться, например, потоковыми схемами и диаграммами.

Новая методология научного исследования в компьютерном моделировании, предполагающая организацию и проведение вычислительного эксперимента на имитационной модели, требует серьезной математической и информационной поддержки процесса системного моделирования, особенно в части вычислительных процедур, связанных с планированием эксперимента, оптимизацией, организацией работы с большим объемом данных в процедурах принятия решений. Многие системы моделирования оснащены средствами интеграции с другими программными средами, осуществляют доступ к процедурным языкам, связанным с кодом имитационной модели, для реализации специальных вычислений, доступа к базам данных (подход Simulation Data Base).

В более мощных пакетах осуществляется интеграция через дополнительное программное обеспечение со специализированными блоками различного назначения. Это могут быть блоки анализа входных данных, гибкие средства анализа чувствительности, позволяющие осуществлять многократные прогоны с различными входными данными (в системах GPSS/H-PROOF, ProModel и др.). Перспективно

создание систем моделирования с функционально широкими, ориентированными на специфику имитационного моделирования блоками оптимизации (в этом смысле показательны системы WITNESS, TAYLOR). Реализуемый в ряде систем многопользовательский режим, применение интерактивного распределенного моделирования, разработки в области взаимодействия имитационного моделирования с Интернетом, расширяют возможности имитационного моделирования, позволяя отрабатывать совместные или конкурирующие стратегии различным компаниям. Характеристики современных систем моделирования сведены в табл. 7.1.

Таблица 7.1. Характеристики современных систем моделирования

Моделирующая среда и поддержка						
Система моделирования	Производитель ПО	Приложения	Графическая конструкция ИМ	Авторское моделирование, программирование моделей	Анимация (в реал. времени)	Поддержка анализа результатов
ARENA	System Modeling Corporation	Производство, анализ бизнес-процессов, дискретное моделирование	Блок-схемы	Да	Да	Да
EXTEND	Imagine That, Inc	Стратегическое планирование, бизнес-моделирование	Компоновочные блоки, непрерывные и дискретные модели	Да (язык Modll)	Да	Анализ чувствительности
GPSS/H-PROOF	Wolverine Software Corporation	Приложения общего назначения, производство, транспорт и др.	Блок-схемы	Да	Да	ANOVA
IT HINK ANALYST	High Performance System, Inc	Управление финансовыми потоками, реинжиниринг предприятий, банков, инвестиционных компаний и др.	CASE-средства, потоковые диаграммы	Да	Да	Анализ чувствительности
PROCESS MODEL	PROMODEL Corporation	Общее производство, реинжиниринг	Блок-схемы, дискретное моделирование	Нет	Нет	Да
SIMUL8	Visual Thinking International	Универсальное средство имитации дискретных процессов	Нет	Объектно-ориентированное программирование	Да	Да
TAYLOR SIMULATION SOFTWARE	F&H Simulation Inc	Производство, стоимостный анализ	Блок-схемы, дискретное моделирование	Нет	Да	Да

Моделирующая среда и поддержка						
Система моделирования	Производитель ПО	Приложения	Графическая конструкция ИМ	Авторское моделирование, программирование моделей	Анимация (в реал. времени)	Поддержка анализа результатов
WITNESS	Lanner Group Inc	Бизнес-планирование, производство, финансы	Да	Да	Да	Да (блок оптимизации)
VENSIM	Ventana Systems	Модели системной динамики	Потоковые диаграммы	Нет	Да	Да
POWER-SIM	Powersim Co	Непрерывное моделирование	Потоковые диаграммы	Нет	Да	Нет
DYNAMO	Expectation Software	Модели системной динамики вычислительного типа	Блок-схемы	Нет	Нет	Нет

7.5.2. AnyLogic — универсальная среда имитационного моделирования

AnyLogic является уникальной средой имитационного моделирования, разработанной российской компанией «Экс Джей Текнолоджис». Уникальность этого программного продукта проявляется практически во всех аспектах:

- ❑ *Полная переносимость между платформами.* Среда AnyLogic написана на языке Java, что обеспечивает ее функционирование в любой операционной системе, на любой аппаратной платформе.
- ❑ *Независимость моделей.* После создания модели она может быть сгенерирована как отдельный Java-апплет и запущена вне среды AnyLogic.
- ❑ *Возможность написания собственных модулей и функций на языке Java.*
- ❑ *Визуальное моделирование.* Все модели можно «складывать» мышью на экране компьютера, что сводит к минимуму необходимость ручного ввода и программирования.
- ❑ *Объектно-ориентированный подход* обеспечивает многократное использование кода.
- ❑ *Возможность создания пользовательских библиотек* обеспечивает многократное использование моделей.
- ❑ *Возможность интерактивной анимации моделей.*
- ❑ *Поддержка всех трех подходов к имитационному моделированию:* системной динамики, дискретно-событийного моделирования и агентного моделирования.
- ❑ *Возможность совмещать внутри одной модели все три подхода.* Если перечисленные ранее пункты можно обнаружить в тех или иных системах имитаци-

онного моделирования, то возможность совмещения подходов внутри одной модели свойственна только среде AnyLogic, что выводит ее на первое место по функциональности в ряду систем имитационного моделирования.

AnyLogic обладает удобным и интуитивно понятным пользовательским интерфейсом. Во время «проигрывания» модели в любой момент можно получить значение любой переменной или характеристику петли обратной связи, просто щелкнув на ней мышью (рис. 7.13).

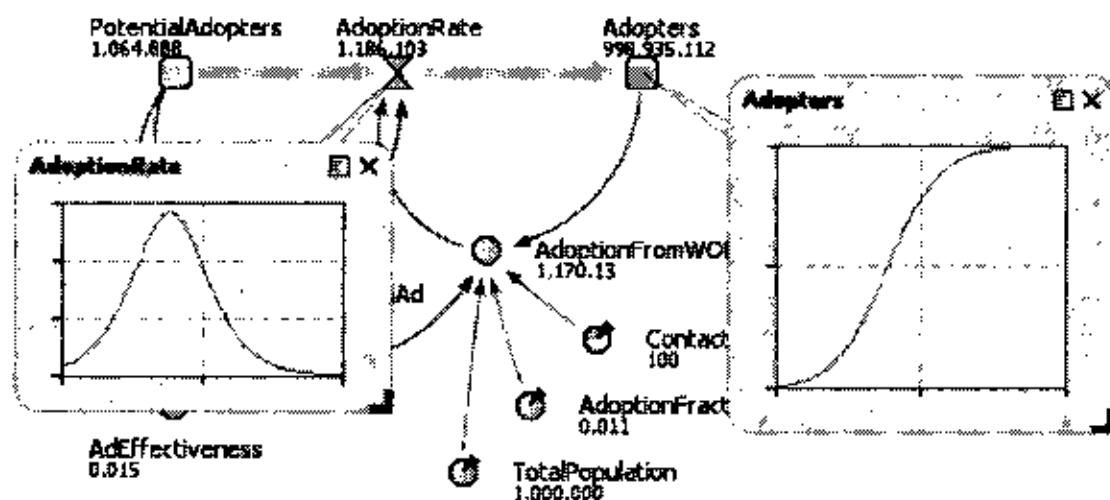


Рис. 7.13. Получение значения переменной во время прогона модели в среде AnyLogic

Также просто в среде AnyLogic создаются дискретно-событийные модели — для этого достаточно просто скомпоновать элементы на экране компьютера (рис. 7.14).

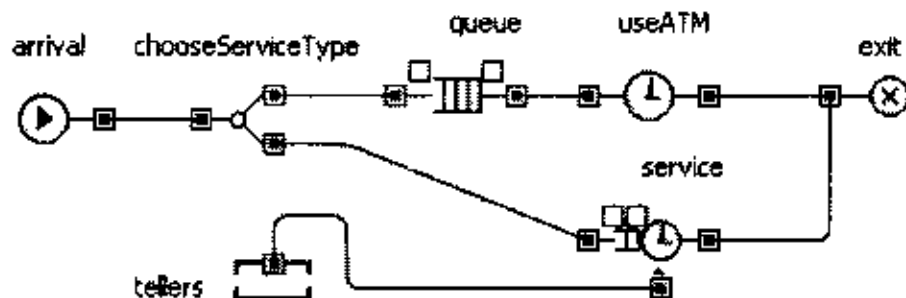


Рис. 7.14. Пример разработки дискретно-событийной модели в среде AnyLogic

В заключение надо сказать, что, являясь безусловным лидером в мире систем имитационного моделирования, AnyLogic кроме профессиональных имеет еще и учебные лицензии (для университетов и школ), которые в 15–20 раз дешевле.

7.5.3. Моделирование бизнес-процессов

Структурно-функциональное моделирование бизнес-процессов может быть одним из этапов построения имитационной модели или выступать в качестве важного инструмента в работе системного аналитика и системного архитектора при созда-

нии информационных систем широкого спектра. Моделирование бизнес-процессов применяют как при создании корпоративных стандартов, разработке технологий или оптимизации структуры организации, так и при создании корпоративных веб-сайтов или интернет-магазинов. Широкий спектр областей применения этого типа моделирования, а также сравнительная простота реализации породила множество программных продуктов, в том числе выпускаемых под свободной лицензией.

Среди свободного программного обеспечения, моделирующего бизнес-процессы, можно назвать такие программы, как Bonita BPM (модель бизнес-процесса, созданная в этой программе, показана на рис. 7.15) или ProcessMaker (моделирование в этой программе иллюстрирует рис. 7.16).

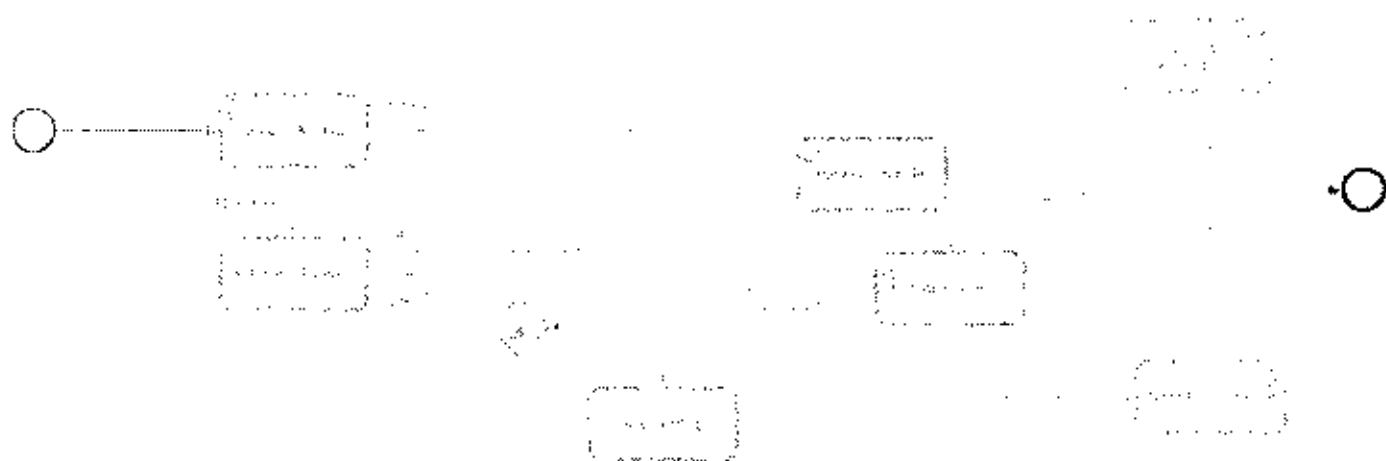


Рис. 7.15. Моделирование бизнес-процесса в среде Bonita BPM

Среди проприетарного программного обеспечения в России наибольшую известность и популярность имеет программный продукт AllFusion Process Modeler (более известный под своим старым названием BPWin). Этот продукт полностью ориентирован на моделирование по стандартам семейства IDEF.

При моделировании в Process Modeler прототипом всегда является процесс (например, процесс оформления заказа). Процесс обозначается прямоугольником, у которого три стороны — это входы в процесс (верх, левая сторона, нижняя сторона), а четвертая (правая) сторона является результатом работы процесса. При этом каждый из входов имеет свое назначение: слева в процесс подается то, что подлежит обработке (информация, материалы); сверху входят нормативные документы, шаблоны, управляющие воздействия; снизу к процессу подключается тот, кто осуществляет управление процессом (персонал, устройство, управляющая система). На выходе процесса формируется результат. Первая диаграмма, описывающая моделируемый процесс, всегда выглядит довольно просто (рис. 7.17).

После создания диаграммы процесса-прототипа начинается декомпозиция: исходный процесс разбивается на несколько составляющих его подпроцессов; чаще всего это последовательные процессы, в которых результат одного процесса подается на вход другого (рис. 7.18), но процессы могут быть и параллельными.

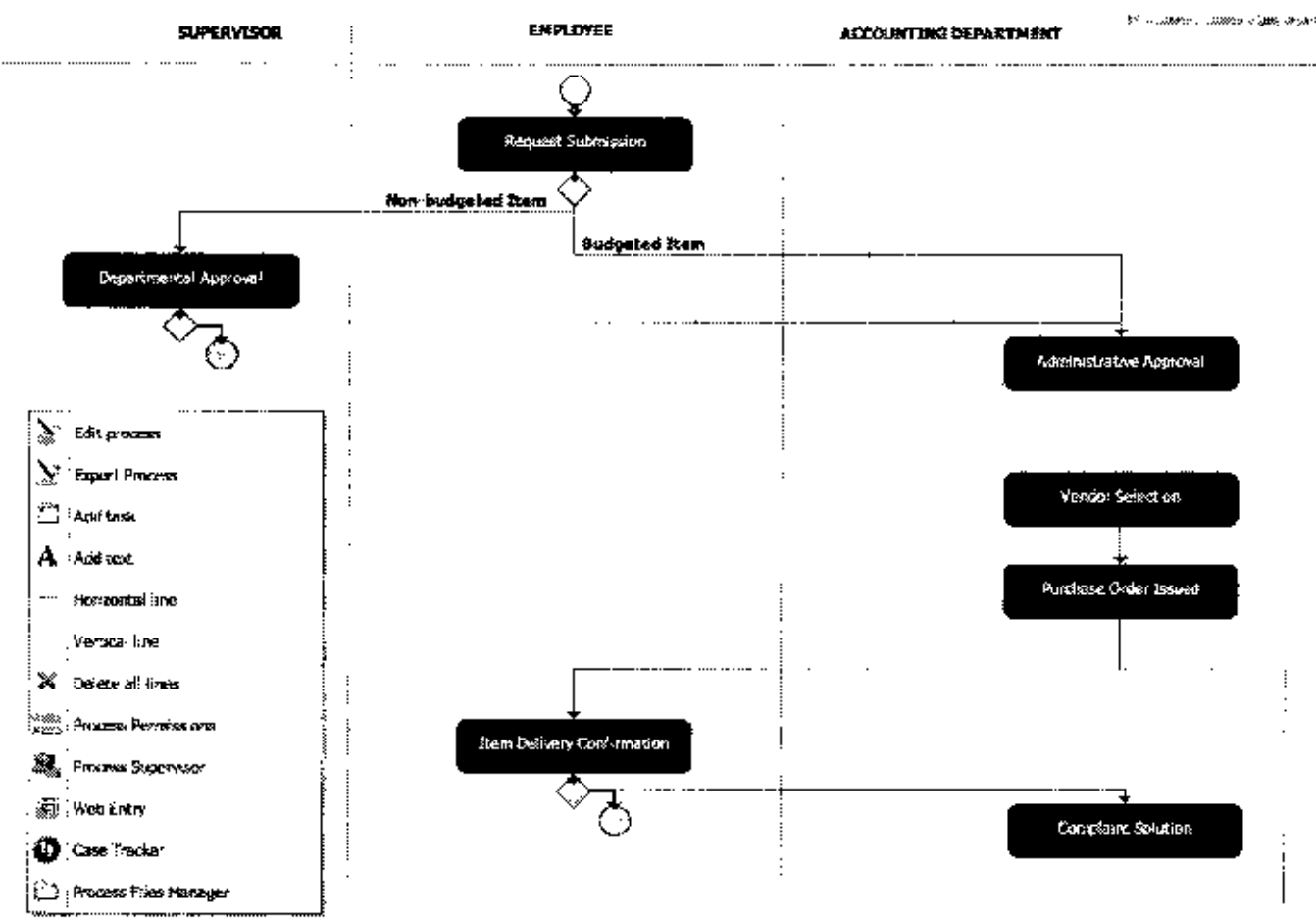


Рис. 7.16. Моделирование бизнес-процесса в среде ProcessMaker

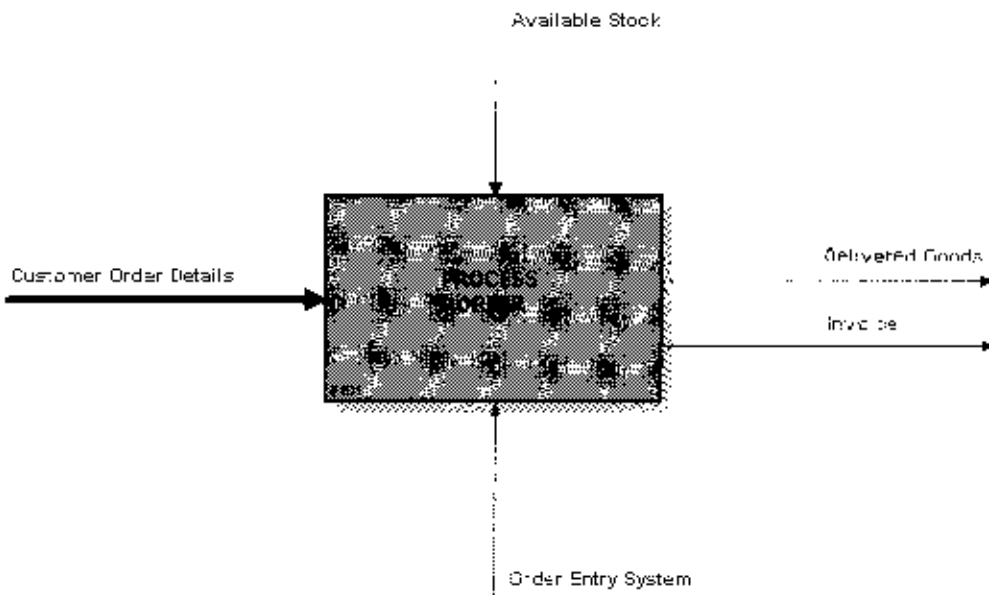


Рис. 7.17. Начальная стадия моделирования в Process Modeler

Каждый из декомпозированных процессов может снова быть декомпозирован. Декомпозиция продолжается до тех пор, пока не достигается результат, сформулированный целью моделирования: процессы становятся настолько элементарными, что могут быть описаны языком программирования или встроены в качестве эле-

ментов в имитационную модель. Рабочее окно AllFusion Process Modeler показано на рис. 7.19.

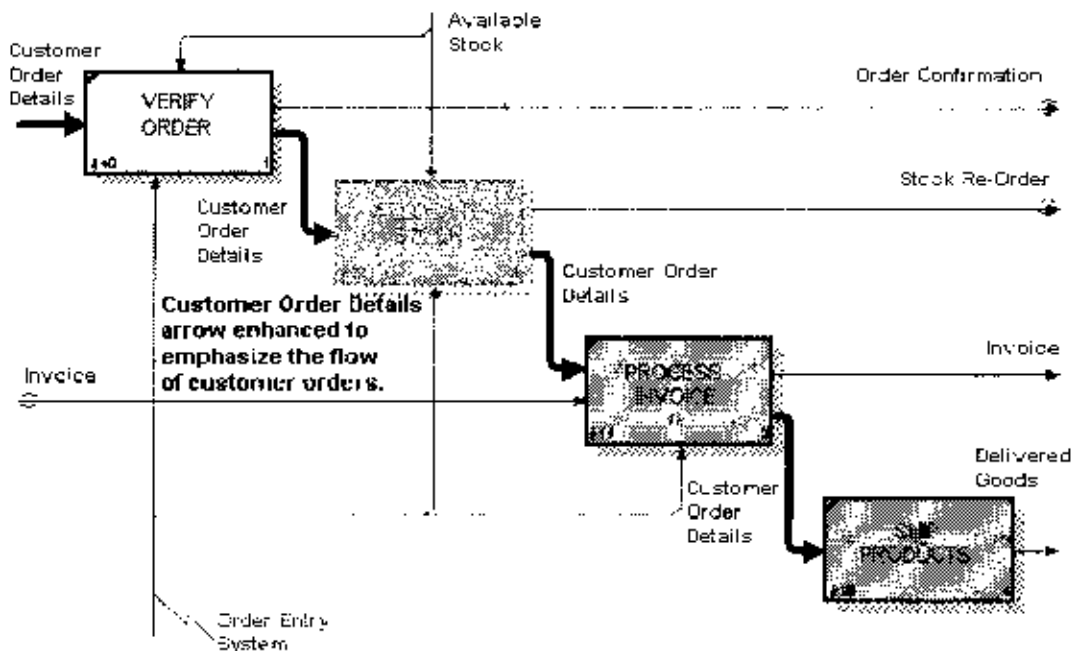


Рис. 7.18. Первый этап декомпозиции

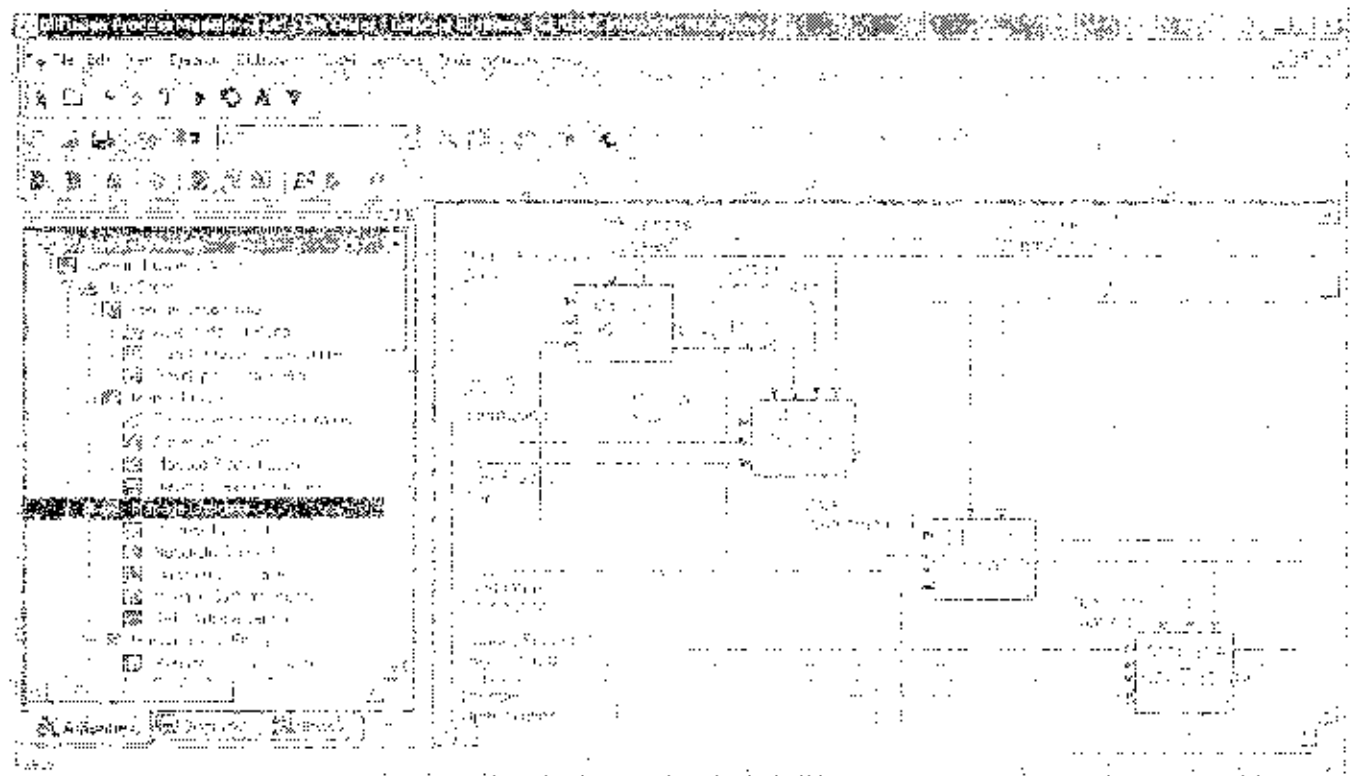


Рис. 7.19. Окно программы Process Modeler

Кроме AllFusion Process Modeler в России применяют отечественные разработки, позволяющие моделировать бизнес-процессы. Среда моделирования бизнес-процессов Орг-Мастер является современным программным продуктом,

оснащенным дополнительными сервисными возможностями по сравнению с All-Fusion Process Modeler:

- ❑ встроенный генератор отчетов (AllFusion Process Modeler работает с Crystal Report);
- ❑ режим улучшенного визуального представления диаграмм (параллельно диаграммам в формате IDEF создаются более удобные для чтения визуальные образы).

На рис. 7.20 приведена подобного рода «улучшенная» диаграмма, облегчающая понимание моделируемого процесса.

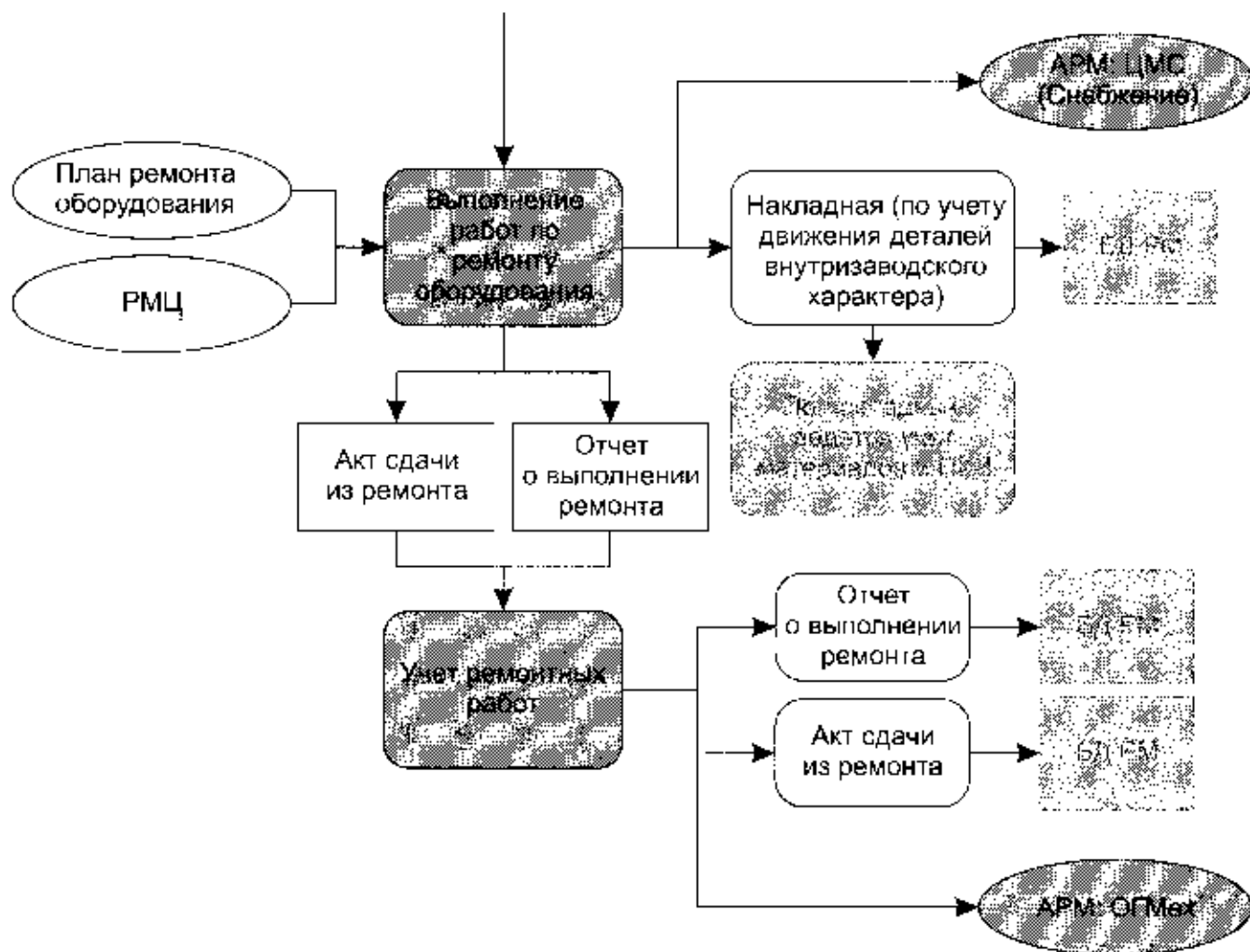


Рис. 7.20. Улучшенная визуализация в Орг-Мастер

Вопросы для самопроверки

1. Дайте определение модели.
2. Какие виды моделирования вы знаете?
3. Что такое «компьютерное моделирование»? «Компьютерная модель»?
4. Каковы основные функции компьютера при компьютерном моделировании?

5. Какие параметры модели вам известны?
6. Каким образом оценивается точность модели?
7. Чем точность модели отличается от адекватности?
8. Какие способы классификации моделей вам известны?
9. Чем прагматическая модель отличается от познавательной?
10. Прагматической или познавательной является модель бизнес-процессов в организации? Дайте объяснение.
11. Чем отличается теоретическая модель от эмпирической? Приведите примеры.
12. Что такое «аналитическая модель» и чем она отличается от алгоритмической?
13. Чем материальная модель отличается от информационной? Является ли чертеж детали материальной моделью? Является ли материальной компьютерная модель?
14. Является ли мысленное представление объекта моделью? Если да, то к какому типу моделей вы его отнесете?
15. Каковы основные этапы построения компьютерной модели?
16. Почему так важна формулировка цели моделирования?
17. В чем заключается анализ результатов моделирования?
18. Что такое «имитационное моделирование»?
19. Каковы особенности имитационного моделирования? В каких случаях необходимо применять имитационные модели?
20. Какие подходы к построению имитационных моделей вам известны?
21. Чем агентное моделирование отличается от дискретно-событийного?
22. Каковы особенности в проведении каждого этапа имитационного моделирования?
23. Назовите по крайней мере две системы, применяющиеся для дискретно-событийного моделирования.
24. Назовите по крайней мере две системы, применяющиеся для непрерывного моделирования.
25. В каких областях человеческой деятельности обычно применяется имитационное моделирование?
26. В чем заключается преимущество российского продукта для имитационного моделирования AnyLogic перед его зарубежными аналогами?
27. Что такое «моделирование бизнес-процессов»? Для чего оно применяется?
28. В чем заключается моделирование бизнес-процессов в соответствии со стандартами IDEF?
29. Какие системы бизнес-моделирования вам известны?
30. В чем заключается преимущество российского продукта для построения моделей бизнес-процессов Org-Master перед его зарубежными аналогами?

Литература

1. *Алямовский А. А.* SolidWorks. Компьютерное моделирование в инженерной практике. СПб.: БХВ, 2005.
2. *Власова Е. А., Дума Р. В., Емельянов А. А.* Имитационное моделирование экономических процессов. М.: Финансы и статистика, 2002.
3. *Дубейковский В. И.* Эффективное моделирование с AllFusion Process Modeler. М.: Диалог-МИФИ, 2007.
4. *Дьяконов В. П.* Matlab + Simulink в математике и моделировании. М.: Солон-пресс, 2005.

Глава 8 Информационная безопасность

- 8.1. Основные положения, понятия и определения
- 8.2. Виды угроз в информационной сфере
- 8.3. Внутренние и внешние источники угроз
- 8.4. Организационно-правовое обеспечение информационной безопасности
- 8.5. Государственная система правового обеспечения защиты информации в Российской Федерации
- 8.6. Угрозы в информационных системах
- 8.7. Защита информации в информационных системах
- 8.8. Государственные стандарты по информационной безопасности

По мере развития и усложнения средств, методов и форм автоматизации процессов обработки информации повышается зависимость общества от степени безопасности используемых им информационных технологий, оказывающих прямое воздействие на благополучие, а иногда и жизнь многих людей. Данная глава посвящена всестороннему рассмотрению вопросов безопасности в информационной сфере: откуда возникают угрозы безопасности и что является основанием для обеспечения надежной защиты от этих угроз.

8.1. Основные положения, понятия и определения

Сложность и объем информационных потоков в современном мире возрастает с каждым днем, информация оказывает непосредственное влияние на деятельность государственных органов, предприятий, организаций, жизнь частных лиц. От того как обеспечена защита циркулирующей в любом срезе общества информации, зависит его жизнедеятельность и безопасность людей. Поэтому вопросам защиты информации во все времена придавалось и придается огромное значение.

Информационная безопасность — это защищенность информационной среды общества посредством различных средств и методов.

Под информационной средой понимается совокупность информационных ресурсов и наличие соответствующей инфраструктуры их создания и использования. *Целью информационной безопасности* является предотвращение влияния неблагоприятных событий (угроз) или обеспечение минимального ущерба от них информационной среде.

Информационная безопасность определяется способностью государства, общества, личности

- обеспечивать защищенность информационных ресурсов для поддержания своей жизнедеятельности и развития;
- противостоять негативным информационным воздействиям на сознание и психику людей, а также на компьютерные сети и другие технические источники информации;
- вырабатывать личностные и групповые навыки безопасного поведения;
- поддерживать постоянную готовность к мерам информационного противодействия.

Информационная безопасность должна обеспечивать *конфиденциальность, точность, полноту и доступность* информации. Эффективная информационная безопасность должна основываться на *теории защиты информации*, в которой рассматриваются следующие направления:

- сбор, систематизация и анализ сведений о проблеме защиты информации;
- формирование на основе собранных сведений научно обоснованных прогнозов о возможности возникновения угроз;
- научно обоснованная постановка задачи защиты информации в современных условиях;
- разработка мероприятий по организации защиты информации;
- разработка методологии и инструментальной базы защиты информации.

В Российской Федерации был выработан ряд подходов к защите информации, закрепленный государственными документами и стандартами. Одним из

таким документов является «Доктрина информационной безопасности Российской Федерации». Согласно этому документу, государственная политика обеспечения информационной безопасности основывается на следующих основных принципах:

- соблюдении Конституции РФ, законодательства РФ, общепризнанных принципов и норм международного права при осуществлении деятельности по обеспечению информационной безопасности РФ;
- информированности общества о деятельности федеральных органов государственной власти и общественных объединений;
- правовом равенстве всех участников процесса информационного взаимодействия вне зависимости от их политического, социального и экономического статуса;
- приоритетном развитии отечественных информационных и телекоммуникационных технологий, производстве технических и программных средств в целях соблюдения жизненно важных интересов РФ.

8.2. Виды угроз в информационной сфере

Угрозы информационной безопасности РФ иллюстрирует рис. 8.1.

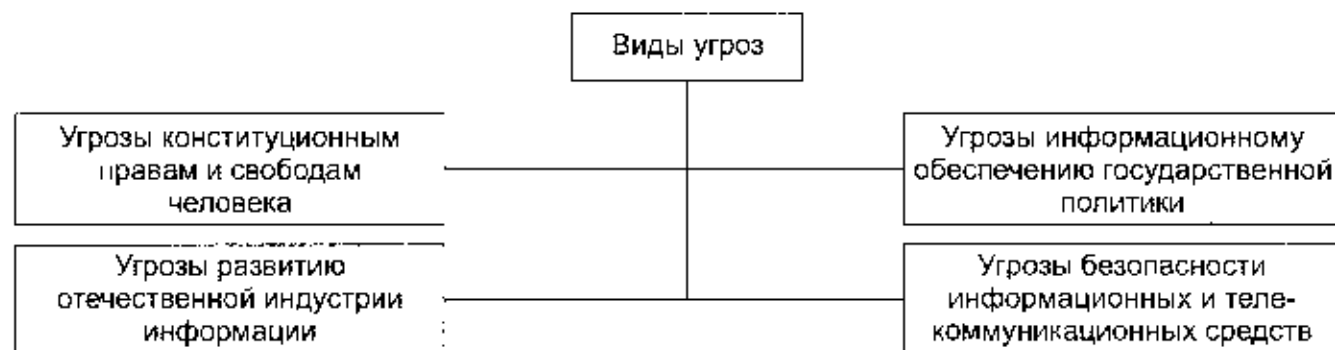


Рис. 8.1. Виды угроз информационной безопасности

Угрозами конституционным правам и свободам человека и гражданина в области духовной жизни и информационной деятельности могут являться:

- принятие федеральными органами государственной власти правовых актов, ущемляющих конституционные права и свободы граждан в области духовной жизни и информационной деятельности;
- нерациональное, чрезмерное ограничение доступа к общественно необходимой информации;
- противоправное применение специальных средств воздействия на индивидуальное, групповое и общественное сознание;
- неисполнение органами государственной власти, организациями и гражданами требований федерального законодательства, регулирующего отношения в информационной сфере;

- вытеснение российских информационных агентств, средств массовой информации с внутреннего информационного рынка;
- манипулирование информацией (дезинформация, сокрытие или искажение информации).

Угрозами информационному обеспечению государственной политики РФ могут являться:

- монополизация информационного рынка России отечественными и зарубежными компаниями;
- блокирование деятельности государственных средств массовой информации по информированию российской и зарубежной аудитории;
- дефицит квалифицированных кадров;
- отсутствие системы формирования и реализации государственной информационной политики.

Угрозами развитию отечественной индустрии информации, включая индустрию средств информатизации, телекоммуникации и связи, могут являться:

- противодействие доступу РФ к новейшим информационным технологиям;
- создание условий для технологической зависимости РФ в области современных информационных технологий;
- закупка органами государственной власти импортных средств информатизации, телекоммуникации и связи при наличии отечественных аналогов;
- вытеснение с отечественного рынка российских производителей средств информатизации, телекоммуникации и связи;
- отток за рубеж специалистов и правообладателей интеллектуальной собственности.

Угрозами безопасности информационных и телекоммуникационных средств и систем могут являться:

- нарушения технологии обработки информации;
- внедрение в аппаратные и программные изделия компонентов, реализующих не предусмотренные документацией функции;
- разработка и распространение программ, нарушающих нормальное функционирование информационных и информационно-телекоммуникационных систем;
- уничтожение или повреждение систем обработки информации;
- воздействие на системы защиты автоматизированных систем обработки и передачи информации;
- утечка информации по техническим каналам;
- внедрение электронных устройств для перехвата информации в технические средства обработки, хранения и передачи информации по каналам связи, а также в служебные помещения органов государственной власти, предприятий, учреждений и организаций независимо от формы собственности;

- ❑ уничтожение, повреждение, разрушение или хищение машинных и других носителей информации;
- ❑ перехват информации в сетях передачи данных и на линиях связи, дешифрование этой информации и навязывание ложной информации;
- ❑ использование несертифицированных отечественных и зарубежных информационных технологий при создании и развитии российской информационной инфраструктуры;
- ❑ несанкционированный доступ к информации в базах данных;
- ❑ нарушение законных ограничений на распространение информации.

8.3. Внутренние и внешние источники угроз

Источники угроз информационной безопасности РФ подразделяются на внешние и внутренние.

К внешним источникам угроз относятся:

- ❑ деятельность иностранных разведок в информационной сфере;
- ❑ ущемление интересов России в мировом информационном пространстве;
- ❑ обострение международной конкуренции за обладание информационными технологиями и ресурсами;
- ❑ деятельность международных террористических организаций;
- ❑ отставание РФ в сфере создания конкурентоспособных информационных технологий;
- ❑ разработка концепций информационных войн.

К внутренним источникам угроз относятся:

- ❑ критическое состояние отечественных отраслей промышленности;
- ❑ неблагоприятная криминогенная обстановка, рост количества преступлений в информационной сфере;
- ❑ плохая координация деятельности органов государственной власти в области обеспечения информационной безопасности РФ;
- ❑ отсутствие нормативной правовой базы в информационной сфере, а также слабая правоприменительная практика;
- ❑ неразвитость институтов гражданского общества;
- ❑ недостаточное финансирование мероприятий по обеспечению информационной безопасности РФ;
- ❑ недостаток квалифицированных кадров в области обеспечения информационной безопасности;
- ❑ отставание России от ведущих стран мира по уровню информатизации органов государственной власти.

8.4. Организационно-правовое обеспечение информационной безопасности

Организационно-правовое обеспечение информационной безопасности — это совокупность решений, законов, нормативов, регламентирующих организацию работ, создание и функционирование систем защиты информации.

Основные функции организационно-правовой базы следующие:

- разработка основных принципов отнесения конфиденциальных сведений к защищаемой информации;
- определение системы органов и должностных лиц, ответственных за обеспечение и регулирование информационной безопасности в стране;
- создание комплекса нормативно-правовых документов, регламентирующих обеспечение информационной безопасности;
- определение мер ответственности за нарушения правил защиты;
- определение порядка разрешения спорных и конфликтных ситуаций по вопросам защиты информации.

Созданием законодательной базы в области информационной безопасности каждое государство стремится защитить свои информационные ресурсы. Информационные ресурсы государства в самом первом приближении могут быть разделены на три большие группы:

- открытую информацию, на распространение и использование которой не имеется ограничений;
- запатентованную информацию, которая охраняется законодательством или международными соглашениями как объект интеллектуальной собственности;
- информацию, «защищаемую» ее владельцем с помощью механизмов защиты государственной, коммерческой или другой охраняемой тайны.

Одним из важнейших в системе защиты государственных секретов в любой стране является понятие государственной тайны. Определение этого понятия дано в Законе РФ «О государственной тайне»:

Государственная тайна — это защищаемые государством сведения в области его военной, внешнеполитической, экономической, разведывательной, контрразведывательной и оперативно-розыскной деятельности, распространение которых может нанести ущерб безопасности РФ.

В зависимости от вида, содержания и размеров ущерба можно выделить группы некоторых видов ущерба при утечке сведений, составляющих государственную тайну.

Политический ущерб может наступить при утечке сведений политического и внешнеполитического характера, о разведывательной деятельности спецслужб государства и др. В результате политического ущерба могут произойти серьезные

изменения в международной обстановке, утрата страной политических приоритетов в каких-то областях, ухудшение отношений с какой-либо страной и т. д.

Экономический ущерб может наступить при утечке сведений любого содержания: политического, экономического, военного, научно-технического и т. д. Экономический ущерб может быть выражен в денежном исчислении в виде прямых и косвенных потерь. Например, *прямые потери* могут наступить в результате утечки секретной информации о системах вооружения и, как следствие, потребовать крупных затрат на их замену. *Косвенные потери* чаще всего выражаются в виде размера упущенной выгоды, например, из-за срыва переговоров с иностранными фирмами, утраты приоритета в научном исследовании, и т. п.

Моральный ущерб наступает от утечки информации, вызвавшей противоправную государству пропагандистскую кампанию, подрывающую репутацию страны.

Засекречивать информацию имеют право органы власти, управления и должностные лица, наделенные соответствующими полномочиями. Осуществляется эта деятельность посредством совокупности организационно-правовых мер, регламентированных законами и другими нормативными актами на распространение и использование информации в интересах ее собственника.

8.5. Государственная система правового обеспечения защиты информации в Российской Федерации

Государственная система обеспечения информационной безопасности создается для решения следующих проблем, требующих законодательной поддержки:

- защита персональных данных;
- борьба с компьютерной преступностью, в первую очередь в финансовой сфере;
- защита коммерческой тайны и обеспечение благоприятных условий для предпринимательской деятельности;
- защита государственных секретов;
- создание системы взаимных финансовых расчетов в электронной форме с элементами цифровой подписи;
- обеспечение безопасности автоматизированных систем управления (АСУ) потенциально опасными производствами;
- страхование информации и информационных систем;
- сертификация и лицензирование в области безопасности, контроль безопасности информационных систем;
- организация взаимодействия в сфере защиты данных со странами-членами СНГ и другими государствами.

Общая организация и координация работ в стране по защите информации, обрабатываемой техническими средствами, осуществляется *Федеральной службой*

по техническому и экспортному контролю (ФСТЭК России). ФСТЭК России является федеральным органом исполнительной власти, осуществляющим реализацию государственной политики, организацию межведомственной координации и взаимодействия, специальные и контрольные функции в области государственной безопасности по следующим вопросам в области информационной безопасности:

- обеспечение безопасности информации в системах информационной и телекоммуникационной инфраструктуры;
- противодействие иностранным техническим разведкам на территории РФ;
- обеспечение защиты (некриптографическими методами) информации, отнесенной к государственной тайне, предотвращение ее утечки по техническим каналам и несанкционированного доступа к ней;
- защита информации при разработке, производстве, эксплуатации и утилизации неинформационных комплексов, систем и устройств.

Основными задачами в области обеспечения информационной безопасности для ФСТЭК России являются:

- реализация государственной политики в области обеспечения безопасности информации;
- организация деятельности государственной системы противодействия техническим разведкам;
- самостоятельное нормативно-правовое регулирование вопросов, связанных с безопасностью информации;
- обеспечение в пределах своей компетенции безопасности информации в ключевых системах информационной инфраструктуры государства;
- прогнозирование развития сил, средств и возможностей технических разведок, выявление угроз безопасности информации;
- осуществление центральным аппаратом ФСТЭК России организационно-технического обеспечения деятельности Межведомственной комиссии по защите государственной тайны.

Нормативные правовые акты и методические документы, изданные по вопросам деятельности ФСТЭК России, обязательны для исполнения аппаратами федеральных органов государственной власти и органов государственной власти субъектов РФ, федеральными органами исполнительной власти, органами исполнительной власти субъектов РФ, органами местного самоуправления и организациями.

Разработка правового обеспечения защиты информации идет по трем направлениям:

- защита прав личности на частную жизнь;
- защита государственных интересов;
- защита предпринимательской и финансовой деятельности.

Структура нормативной базы по вопросам информационной безопасности включает:

- Конституцию РФ;
- федеральные законы и законы РФ;
- кодексы РФ (уголовный, гражданский, об административных правонарушениях);
- постановления правительства РФ;
- ведомственные нормативные акты, ГОСТы, руководящие документы.

8.6. Угрозы в информационных системах

Угрозы информационной безопасности информационных систем можно подразделить на несколько групп в зависимости от того, на какой параметр информации воздействует угроза (рис. 8.2).

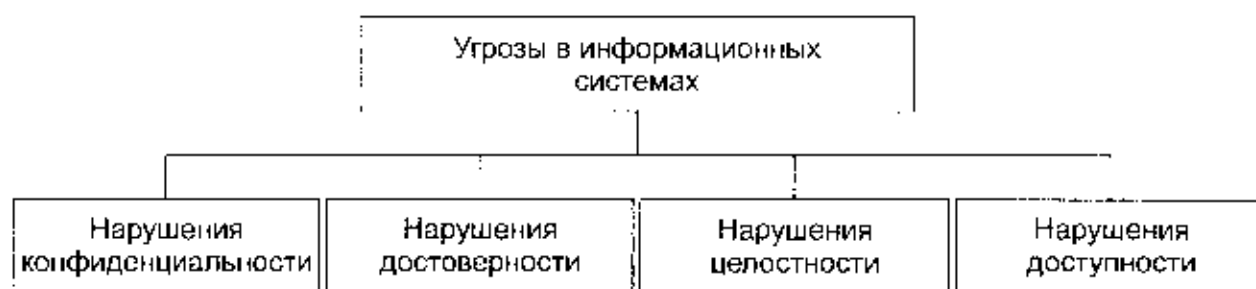


Рис. 8.2. Угрозы в информационных системах

8.6.1. Нарушения конфиденциальности

Нарушение конфиденциальности является одной из наиболее тяжелых с точки зрения ущерба угроз информационной безопасности. Если в случае нарушений достоверности, целостности и доступности информации факт угрозы обнаруживается немедленно или в течение сравнительно короткого промежутка времени, то в случае нарушения конфиденциальности деструктивное воздействие может продолжаться длительное время, нанося тяжелый материальный, моральный и политический ущерб.

Нарушение конфиденциальности наступает в результате следующих действий:

- *разглашение* — умышленные или неосторожные действия должностных лиц и граждан, приведшие к ознакомлению посторонних лиц с конфиденциальными сведениями;
- *утечка* — бесконтрольный выход информации за пределы круга лиц или организаций, которым она была доверена;
- *несанкционированный доступ* — противоправное преднамеренное овладение конфиденциальной информацией лицом, не имеющим на это права.

Пример. Сбор конфиденциальных (в том числе военных) сведений разведками иностранных государств или промышленный шпионаж, позволяющий похитить информацию, дающую конкурентные преимущества.

8.6.2. Нарушения достоверности

Нарушение достоверности — это вторая по тяжести последствий после нарушения конфиденциальности угроза. Нарушение достоверности информации происходит в результате фальсификации, подделки или мошенничества. Ущерб от этой угрозы может быть менее масштабным, чем от нарушения конфиденциальности, но при этом последствия могут быть катастрофическими.

Пример. Подмена кодов или координат цели в системе наведения стратегических ядерных ракет, изменение режимов синхронизации в глобальной энергетической системе, вызывающее разрушение генераторов и массовое отключение электроэнергии.

8.6.3. Нарушения целостности

Нарушение целостности информации обычно происходит в каналах передачи или в хранилищах информации и заключается в искажениях, внесении ошибок и потере части передаваемой или хранимой информации. Обычно нарушение целостности информации выявляется гораздо раньше, чем нарушение достоверности, поскольку целостность информации легче поддается автоматизированному (аппаратному) контролю. Последствия нарушения целостности информации обычно носят тактический характер, а тяжесть последствий зависит от ситуации.

Пример. Нарушение целостности информации в распределенной информационной системе управления предприятием может привести к финансовым потерям одной фирмы, но такое же нарушение в распределенной информационной системе противовоздушной обороны в момент ракетного нападения может быть фатальным.

8.6.4. Нарушения доступности

Нарушение доступности происходит вследствие блокирования, невозможного искажения или уничтожения информации. Последствия от нарушения доступности, так же как и в случае нарушения целостности, могут иметь разную тяжесть в зависимости от ситуации.

Пример. Стирание информации на магнитных носителях в банковской системе может иметь тяжелые финансовые и экономические последствия, в то время как физическое уничтожение библиотеки уникальных документов может быть культурной катастрофой.

8.6.5. Классификация угроз информации

Классификацию угроз иллюстрирует рис. 8.3.

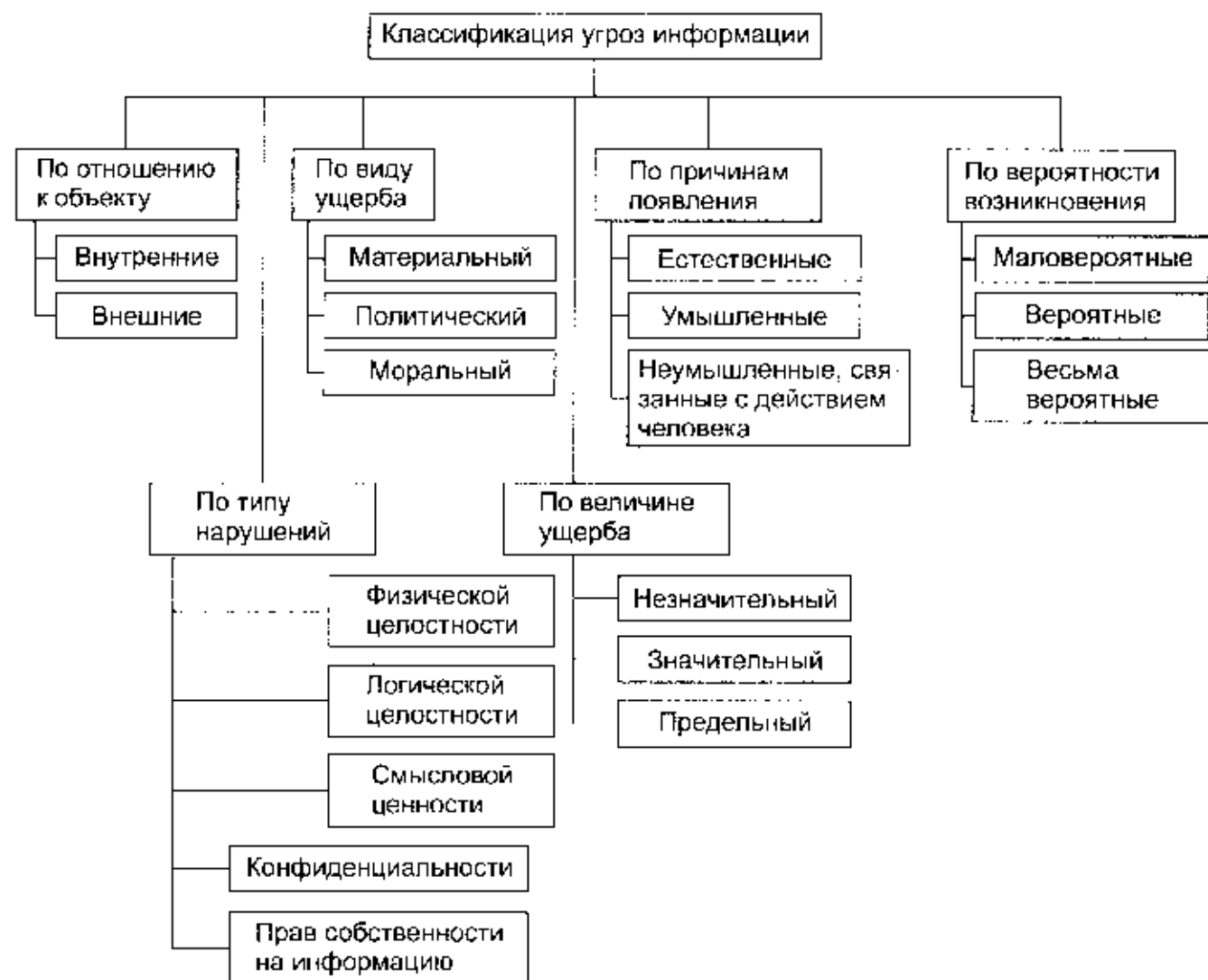


Рис. 8.3. Классификация угроз

Классификация угроз по отношению к объекту

Разделение угроз на внутренние и внешние зависит от того, где проводится граница между объектом, содержащим информацию, и внешним миром. Так, для персонального компьютера внешней угрозой могут быть действия пользователя локальной сети. Эта же угроза с точки зрения организации, которой принадлежит данная локальная сеть, будет внутренней.

Наиболее опасной и вероятной внутренней угрозой для любой информационной системы или системы охраны информации являются низкая квалификация, халатность или враждебные действия персонала (пользователей).

Классификация угроз по виду ущерба

Разделение ущерба на материальный и моральный в достаточной мере условно: тот и другой, в конце концов, ведут к материальным потерям, а размеры материаль-

ного ущерба, косвенно вызванного моральным ущербом, могут быть больше, чем прямые материальные потери.

Пример. Рассылка в локальной сети организации писем, содержащих провокационные сведения, приводит к созданию атмосферы разобщенности, недоверия, дезорганизованности в коллективе. Такая атмосфера может послужить почвой как для намеренных, так и для ненамеренных действий, приводящих к значительному материальному ущербу (разглашение конфиденциальных данных, техническая авария крупных размеров).

Классификация угроз по причинам появления

По причинам появления угрозы можно разделить на связанные с явлениями природы (стихийные бедствия или экспессы) и действиями человека. В свою очередь, человеческие действия, составляющие угрозу, могут совершаться на основе умысла (шпионаж, диверсии, кража или порча конфиденциальной информации) или без умысла (халатность, некомпетентность, результат плохого самочувствия).

Классификация угроз по вероятности возникновения

Вероятность возникновения того или иного вида угроз не является постоянной величиной и зависит от комплекса причин. В разных ситуациях возникновение одной и той же угрозы может иметь разную вероятность.

Пример. Угроза физического разрушения сетевой инфраструктуры или информационных хранилищ маловероятна в мирное время, однако во время вооруженных конфликтов она может стать весьма вероятной и даже основной.

Классификация угроз по типу нарушений

Под нарушениями в первую очередь понимаются нарушения целостности информации: физической (уничтожение, разрушение элементов), логической (разрушение логических связей), смысловой (навязывание ложной информации). Нарушение конфиденциальности выражается в разрушении защиты и несанкционированном овладении конфиденциальной информацией. Все чаще в последнее время в нашей стране обращают внимание на нарушение прав собственности на информацию (несанкционированное копирование, использование).

8.7. Защита информации в информационных системах

8.7.1. Требования к защите информации

Конкретные требования к защите, обусловленные спецификой автоматизированной обработки информации, определяются совокупностью следующих факторов:

- ☐ характером обрабатываемой информации;
- ☐ объемом обрабатываемой информации;

- продолжительностью пребывания информации в автоматизированной системе обработки информации;
- структурой автоматизированной системы обработки данных;
- видом защищаемой информации;
- технологией обработки информации;
- организацией информационно-вычислительного процесса в автоматизированной системе обработки данных;
- этапом жизненного цикла автоматизированной системы обработки данных.

Информация должна защищаться во всех структурных элементах информационной системы.

На рис. 8.4 представлена структурная схема автоматизированной системы обработки информации с мейнфреймовой структурой (одним центральным вычислителем и пользовательскими местами в виде терминалов). Именно такая система обеспечивает наиболее эффективную защиту информации. Если терминалами пользователя являются персональные компьютеры, необходимо сделать конструктивно невозможными бесконтрольное снятие и ввод информации через устройства ввода-вывода (например, физически заблокировать или удалить все разъемы подключения USB-устройств, порты ввода-вывода, устройства чтения гибких и лазерных дисков).

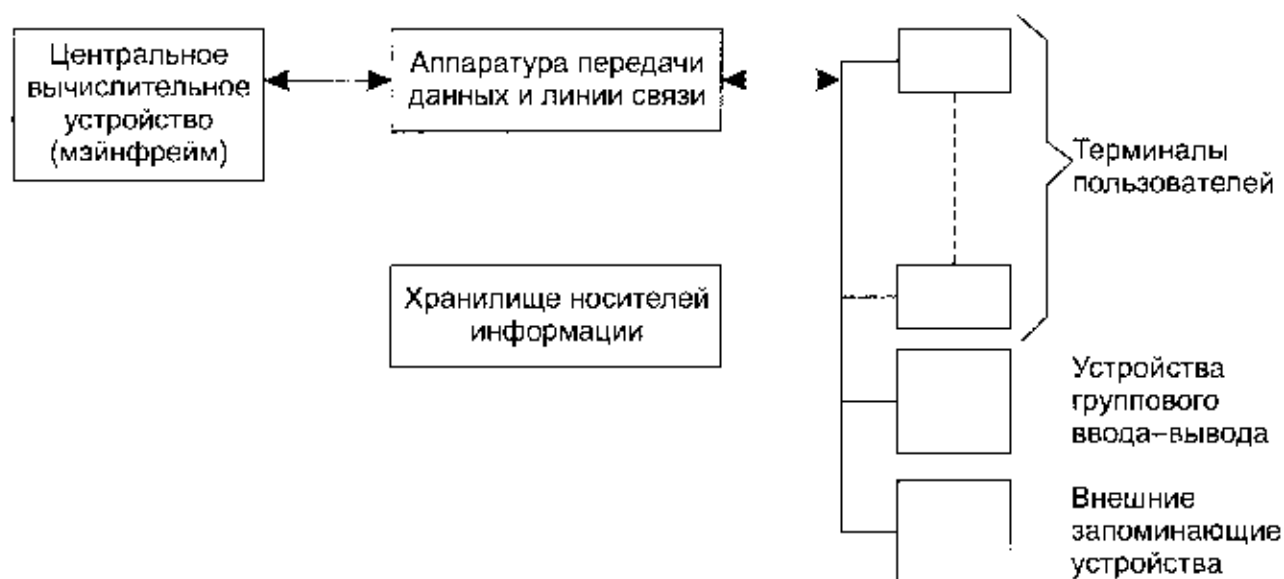


Рис. 8.4. Структурная схема информационной системы

Защита информации в терминалах пользователей:

- защищаемая информация может находиться только во время сеанса;
- должна быть исключена возможность просмотра отображаемой информации со стороны;
- информация, имеющая ограничительный гриф, должна выдаваться (отображаться) совместно с этим грифом.

Защита информации в устройствах группового ввода-вывода:

- в устройствах группового ввода-вывода информация может находиться только во время решения задач либо с нормированной длительностью хранения;

- устройства отображения и фиксации информации должны исключать возможность просмотра отображаемой информации со стороны;
- информация, имеющая ограничительный гриф, должна выдаваться (отображаться) совместно с этим грифом.

Защита информации в аппаратуре и линиях связи:

- в аппаратуре и линиях связи защищаемая информация должна находиться только в течение сеанса;
- линии связи, по которым защищаемая информация передается в явном виде, должны находиться под непрерывным контролем на все время сеанса передачи;
- перед началом каждого сеанса передачи должна осуществляться проверка адреса выдачи данных;
- при передаче большого объема защищаемой информации проверка адреса передачи должна производиться периодически (через заданные промежутки времени или после передачи заданного числа знаков сообщения);
- при наличии в составе аппаратуры связи процессоров и запоминающих устройств должна вестись регистрация данных обо всех сеансах передачи защищаемой информации.

Защита информации в центральном вычислителе:

- защищаемая информация в оперативном запоминающем устройстве может находиться только во время сеансов решения соответствующих задач, во внешних запоминающих устройствах — минимальное время, определяемое технологией решения соответствующей прикладной задачи в автоматизированной системе обработки данных;
- устройства отображения и фиксации информации должны исключать возможность просмотра отображаемой информации со стороны;
- информация, имеющая ограничительный гриф, должна выдаваться (отображаться) совместно с этим грифом;
- при обработке защищаемой информации должно осуществляться установление подлинности всех участвующих в обработке устройств и пользователей с ведением протоколов их работы;
- всякое обращение к защищаемой информации должно проверяться на санкционированность;
- при обмене защищаемой информацией, осуществляемом с использованием линий связи, должна осуществляться проверка адресов корреспондентов.

Защита информации во внешних запоминающих устройствах:

- сменные носители информации должны находиться на устройствах управления в течение минимального времени, определяемого технологией автоматизированной обработки информации;
- устройства управления внешними запоминающими устройствами, на которых установлены носители с защищаемой информацией, должны иметь замки, предупреждающие несанкционированное изъятие или замену носителя.

Защита информации в хранилище носителей:

- все носители, содержащие защищаемую информацию, должны иметь четкую и однозначную маркировку, которая, однако, не должна раскрывать содержания записанной на них информации;
- носители, содержащие защищаемую информацию, должны храниться таким образом, чтобы исключались возможности несанкционированного доступа к ним;
- при выдаче и приеме носителей должна осуществляться проверка личности получающего (сдающего) и его санкции на получение (сдачу) этих носителей.

Защита информации для всех устройств: для всех устройств должна быть предусмотрена возможность аварийного уничтожения информации.

8.7.2. Способы и средства защиты информации

Содержание способов обеспечения безопасности представлено на рис. 8.5.



Рис. 8.5. Методы защиты информации

Препятствие заключается в создании на пути возникновения или распространения угрозы барьера, не позволяющего угрозе принять опасные размеры.

Пример. Блокировки, не позволяющие техническому устройству или программе выйти за опасные границы; создание физических препятствий на пути злоумышленников, экранирование помещений и технических средств и т. п.

Управление (или регламентация) заключается в определении алгоритмов функционирования систем обработки информации, а также процедур и правил работы предприятий и учреждений, препятствующих возникновению угрозы.

Пример. Управление доступом на секретный объект, включающее в себя процедуры и правила, которые должен выполнять персонал объекта, а также алгоритмы работы механизмов и устройств слежения, фиксации и ограничения доступа.

Маскировка — преобразование информации или скрываемого объекта, вследствие которого снижается степень распознавания скрываемой информации или объекта и затрудняется доступ к ним.

Пример. Шифрование информации, дезинформация о месте нахождения конфиденциальной информации, создание легенд, намеренное внесение помех.

Принуждение – это такой метод защиты, при котором пользователи и персонал системы вынуждены соблюдать правила обработки, передачи и использования защищаемой информации под угрозой материальной, административной или уголовной ответственности.

Побуждение есть способ защиты информации, при котором пользователи и персонал объекта внутренне (благодаря материальным, моральным, этическим, психологическим и другим мотивам) побуждаются к соблюдению всех правил обработки информации.

Нападение – способ защиты, применяемый в активной фазе информационной войны. Цель – заставить противника сосредоточить усилия на защите, ослабив усилия на создание угроз.

Рассмотренные способы обеспечения защиты информации реализуются с применением различных средств (рис. 8.6).



Рис. 8.6. Средства защиты информации

Физико-технические средства делятся на физические, аппаратные и программные.

- ❑ **Физические средства** – механические, электрические, электромеханические и т. п. устройства и системы, которые функционируют автономно, создавая различного рода препятствия на пути угроз.
- ❑ **Аппаратные средства** – различные электронные и электронно-механические и т. п. устройства, встраиваемые в схемы аппаратуры системы обработки данных или сопрягаемые с ней специально для решения задач защиты информации. Например, для защиты от утечки по техническим каналам используются генераторы шума.

Физические и аппаратные средства объединяются в класс технических средств защиты информации.

- ❑ **Программные средства** – специальные пакеты программ или отдельные программы, включаемые в состав программного обеспечения автоматизированных систем с целью решения задач защиты информации. Это могут быть различные

программы по криптографическому преобразованию данных, контролю доступа, защите от вирусов и др.

Организационно-социальные средства делятся на организационные, законодательные и морально-этические.

- *Организационные средства* — специально предусматриваемые в технологии функционирования объекта организационно-технические мероприятия для решения задач защиты информации, осуществляемые в виде целенаправленной деятельности людей.
- *Законодательные средства* — нормативно-правовые акты, с помощью которых регламентируются права и обязанности всех лиц и подразделений, связанных с защитой информации, а также устанавливается ответственность за нарушение правил обработки информации, следствием чего может быть нарушение защищенности информации.
- *Морально-этические средства* защиты информации предполагают, прежде всего, воспитание сотрудника, допущенного к секретам, то есть проведение специальной работы, направленной на формирование у него системы определенных качеств, взглядов и убеждений (патриотизма, понимания важности и полезности защиты информации и для него лично), обучение сотрудника, осведомленного в сведениях, составляющих охраняемую тайну, правилам и методам защиты информации, привитие ему навыков работы с носителями секретной и конфиденциальной информации.

8.7.3. Системы защиты информации

С целью всесторонней защиты информации рекомендуется создать единую, целостную систему, являющуюся функционально самостоятельной подсистемой любого объекта обработки информации. Такое решение позволяет объединить все ресурсы, средства и методы, а также полноценно координировать мероприятия по защите информации.

Организационно система защиты информации (СЗИ) должна состоять из трех механизмов:

- обеспечения защиты информации;
- управления средствами защиты;
- общей организации работы системы.

Ядро системы защиты информации

Ядро системы защиты предназначено для объединения всех подсистем системы защиты информации в единую целостную систему и организации ее функционирования.

Ядро может включать в себя организационные и технические составляющие.

Организационная составляющая представляет собой совокупность специально выделенных для защиты информации сотрудников, выполняющих свои функции в соответствии с разработанными правилами, а также нормативную базу, регламентирующую выполнение этих функций.

Техническая составляющая обеспечивает техническую поддержку организационной составляющей и представляет собой совокупности технических средств отображения состояний элементов системы защиты информации, контроля доступа к ним, управления их включением и т. д. Чаще всего эти средства объединены в соответствующий пульт управления системой защиты информации.

Ядро системы защиты информации должно обеспечивать выполнение двух функций.

- Включение компонентов системы защиты информации в работу при поступлении запросов на обработку защищаемой информации и блокирование бесконтрольного доступа к ней. Для этого требуется:
 - оборудование объекта средствами охранной сигнализации;
 - хранение носителей защищаемой информации в отдельных хранилищах (документация, шифры, магнитные носители и т. д.);
 - включение блокирующих устройств, регулирующих доступ к элементам системы защиты информации при предъявлении соответствующих полномочий и средств сигнализации.
- Организация и обеспечение проверок правильности функционирования системы защиты информации. При этом реализуется проверка:
 - аппаратных средств — по тестовым программам и организационно;
 - физических средств — организационно (плановые проверки средств охранной сигнализации, сигнализации о повышении давления в кабелях и т. д.);
 - программных средств — по специальным контрольным суммам (на целостность) и по другим идентифицирующим признакам.

Ресурсы системы защиты информации

Ресурсы информационно-вычислительной системы, необходимые для создания и поддержания функционирования системы защиты информации, как и любой другой автоматизированной системы, объединяются в техническое, математическое, программное, информационное и лингвистическое обеспечение.

Техническое обеспечение — совокупность технических средств, необходимых для технической поддержки решения всех тех задач защиты информации, которые возникают в процессе функционирования системы защиты информации.

Математическое обеспечение — совокупность математических методов, моделей и алгоритмов, необходимых для оценки уровня защищенности информации и решения других задач защиты.

Программное обеспечение — совокупность программ, реализующих программные средства защиты, а также программ, необходимых для решения задач управления механизмами защиты. К ним должны быть отнесены также сервисные и вспомогательные программы системы защиты информации.

Информационное обеспечение — совокупность систем классификации и кодирования данных о защите информации, массивы данных системы защиты информации, а также входные и выходные документы системы защиты информации.

Лингвистическое обеспечение – совокупность языковых средств, необходимых для обеспечения взаимодействия компонентов системы защиты информации между собой, с компонентами объекта обработки информации, с внешней средой.

Организационное построение

Организационное построение системы защиты информации в самом общем случае может быть представлено совокупностью следующих рубежей защиты:

- территории, занимаемой системой защиты информации;
- зданий, расположенных на территории;
- помещений внутри здания, в которых расположены ресурсы системы защиты информации и защищаемая информация;
- ресурсов, используемых для обработки и хранения информации и самой защищаемой информации;
- линий связи, проходящих в пределах одного и того же здания;
- линий (каналов) связи, проходящих между различными зданиями, расположенными на одной и той же охраняемой территории;
- линий (каналов) связи, соединяющих системы защиты информации с другими объектами вне охраняемой территории.

8.8. Государственные стандарты по информационной безопасности

8.8.1. Стандарты общего назначения

ГОСТ Р ИСО/МЭК 13335-1-2006. Информационная технология. Методы и средства обеспечения безопасности. Часть 1. Концепция и модели менеджмента безопасности информационных и телекоммуникационных технологий. Information technology. Security techniques. Part 1. Concepts and models for information and communications technology security management.

ГОСТ Р ИСО/МЭК ТО 13335-3-2007. Информационная технология. Методы и средства обеспечения безопасности. Часть 3. Методы менеджмента безопасности информационных технологий. Information technology. Security techniques. Part 3. Techniques for the management of information technology security.

ГОСТ Р ИСО/МЭК ТО 13335-4-2007. Информационная технология. Методы и средства обеспечения безопасности. Часть 4. Выбор защитных мер. Information technology. Security techniques. Part 4. Selection of safeguards.

ГОСТ Р ИСО/МЭК ТО 13335-5-2006. Информационная технология. Методы и средства обеспечения безопасности. Часть 5. Руководство по менеджменту безопасности сети. Information technology. Security techniques. Part 5. Management guidance on network security.

ГОСТ Р ИСО/МЭК 13569-xxxx (проект 10.05.2007). Финансовые услуги. Рекомендации по информационной безопасности. Financial services — Information security Guidelines.

ГОСТ Р ИСО/МЭК 15026-2002. Информационная технология. Уровни целостности систем и программных средств. Information technology. System and software integrity levels.

ГОСТ Р ИСО/МЭК 15408-1-2002. Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 1. Введение и общая модель. Information technology. Security techniques. Evaluation criteria for IT security. Part 1. Introduction and general model.

ГОСТ Р ИСО/МЭК 15408-2-2002. Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 2. Функциональные требования безопасности. Information technology. Security techniques. Evaluation criteria for IT security. Part 2. Security functional requirements.

ГОСТ Р ИСО/МЭК 15408-3-2002. Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 3. Требования доверия к безопасности. Information technology. Security techniques. Evaluation criteria for IT security. Part 3. Security assurance requirements.

ГОСТ Р ИСО/МЭК 17799-2005. Информационная технология. Практические правила управления информационной безопасностью. Information technology. Code of practice for information security management.

ГОСТ Р ИСО/МЭК 18044-xxxx (проект). Информационная технология. Методы и средства обеспечения безопасности. Менеджмент инцидентов информационной безопасности. Information technology. Security techniques. Information security incident management.

ГОСТ Р ИСО/МЭК 18045-xxxx (проект). Информационная технология. Методы и средства обеспечения безопасности. Методология оценки безопасности информационных технологий. Information technology. Security techniques. Methodology for IT security evaluation.

ГОСТ Р ИСО/МЭК 19794-2-2005. Автоматическая идентификация. Идентификация биометрическая. Форматы обмена биометрическими данными. Часть 2. Данные изображения отпечатка пальца — контрольные точки. Automatic identification. Biometrics. Biometric data interchange formats. Part 2. Finger minutiae data.

ГОСТ Р ИСО/МЭК 19794-4-2006. Автоматическая идентификация. Идентификация биометрическая. Форматы обмена биометрическими данными. Часть 4. Данные изображения отпечатка пальца. Automatic identification. Biometrics. Biometric data interchange formats. Part 4. Finger image data.

ГОСТ Р ИСО/МЭК 19794-5-2006. Автоматическая идентификация. Идентификация биометрическая. Форматы обмена биометрическими данными. Часть

5. Данные изображения лица. Automatic identification. Biometrics. Biometric data interchange formats. Part 5. Face image data.

ГОСТ Р ИСО/МЭК 19794-6-2006. Автоматическая идентификация. Идентификация биометрическая. Форматы обмена биометрическими данными. Часть 6. Данные изображения радужной оболочки глаза. Automatic identification. Biometrics. Biometric data interchange formats. Part 6. Iris image data.

ГОСТ Р ИСО/МЭК 27001-2006. Информационная технология. Методы и средства обеспечения безопасности. Системы менеджмента информационной безопасности. Требования. Information technology. Security techniques. Information security management systems. Requirements.

Р 50.1.056. Техническая защита информации. Основные термины и определения.

ГОСТ Р 50739-95 Средства вычислительной техники. Защита от несанкционированного доступа к информации. Общие технические требования. Computers technique. Information protection against unauthorised access to information. General technical requirements.

ГОСТ Р 50922-2006. Защита информации. Основные термины и определения. Protection of information. Basic terms and definitions.

ГОСТ Р 50922-96. Защита информации. Основные термины и определения. Protection of information. Basic terms and definitions.

ГОСТ Р 51188-98. Защита информации. Испытания программных средств на наличие компьютерных вирусов. Типовое руководство. Information security. Software testing for the existence of computer viruses. The sample manual.

ГОСТ Р 51275-2006. Защита информации. Объект информатизации. Факторы, воздействующие на информацию. Общие положения. Protection of information. Object of informatisation. Factors influencing the information. General.

ГОСТ Р 51275-99. Защита информации. Объект информатизации. Факторы, воздействующие на информацию. Общие положения. Protection of information. Object of informatization. Factors influencing on information. General outlines.

ГОСТ 51583-2000. Порядок создания автоматизированных систем в защищенном исполнении.

ГОСТ Р 51624-2000. Автоматизированные системы в защищенном исполнении. Общие положения.

ГОСТ Р 51725.6-2002. Каталогизация продукции для федеральных государственных нужд. Сети телекоммуникационные и базы данных. Требования информационной безопасности. Catalogization of products for federal state needs. Telecommunication networks and data bases. Requirements of information security.

ГОСТ Р 51898-2002. Аспекты безопасности. Правила включения в стандарты. Safety aspects. Guidelines for their inclusion in standards.

ГОСТ Р 52069.0-2003. Защита информации. Система стандартов. Основные положения. Safety of information. System of standards. Basic principles.

ГОСТ Р 52447-2005. Защита информации. Техника защиты информации. Номенклатура показателей качества. Information protection. Information protection technology. Nomenclature of quality indices.

8.8.2. Стандарты по криптографической защите

ГОСТ 28147-89. Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования.

ГОСТ Р 34.10-2001. Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи. Information technology. Cryptographic data security. Formation and verification processes of [electronic] digital signature.

ГОСТ Р 34.11-94. Информационная технология. Криптографическая защита информации. Функция хэширования. Information technology. Cryptographic data security. Hashing function.

ГОСТ 34.311-95. Информационная технология. Криптографическая защита информации. Функция хэширования. Information technology. Cryptographic Data Security. Hashing function.

8.8.3. Стандарты информационной безопасности в кредитно-финансовой сфере

СТО БР ИББС-1.0-2006. Стандарт Банка России. Обеспечение информационной безопасности организаций банковской системы Российской Федерации. Общие положения.

СТО БР ИББС-1.1-2007. Стандарт Банка России. Обеспечение информационной безопасности организаций банковской системы Российской Федерации. Аудит информационной безопасности.

СТО БР ИББС-1.2-2007. Стандарт Банка России. Обеспечение информационной безопасности организаций банковской системы Российской Федерации. Методика оценки соответствия информационной безопасности организаций банковской системы Российской Федерации требованиям СТО БР ИББС-1.0-2006.

РС БР ИББС-2.0-2007. Рекомендации в области стандартизации Банка России. Обеспечение информационной безопасности организаций банковской системы Российской Федерации. Методические рекомендации по документации в области обеспечения информационной безопасности в соответствии с требованиями СТО БР ИББС-1.0.

РС БР ИББС-2.1-2007. Рекомендации в области стандартизации Банка России. Обеспечение информационной безопасности организаций банковской системы Российской Федерации. Руководство по самооценке соответствия информационной безопасности организаций банковской системы Российской Федерации требованиям СТО БР ИББС-1.0.

Вопросы для самопроверки

1. Дайте определение информационной безопасности.
2. Чем определяется информационная безопасность?
3. Каковы основные принципы государственной политики обеспечения информационной безопасности России?
4. Перечислите виды угроз в информационной сфере.
5. Что относится к внутренним и внешним источникам угроз информационной безопасности?
6. Дайте определение организационно-правовому обеспечению информационной безопасности.
7. Что такое «государственная тайна»?
8. Какие виды ущерба может нанести нарушение информационной безопасности?
9. Что такое «ФСТЭК» и каковы ее функции?
10. Какие вы знаете угрозы информационным системам?
11. В чем заключается нарушение конфиденциальности? Приведите пример.
12. В чем заключается нарушение достоверности? Приведите пример.
13. В чем заключается нарушение целостности? Приведите пример.
14. В чем заключается нарушение доступности? Приведите пример.
15. Каким образом можно классифицировать угрозы информации?
16. Какие факторы влияют на требования к защите информации в конкретной информационной системе?
17. Какие способы защиты информации вам известны?
18. При помощи каких средств реализуется защита информации?
19. Что такое «система защиты информации»?
20. Какие семь рубежей защиты в системе защиты информации вам известны?

Литература

1. Белов Е. Б., Лось В., Мещеряков Р. В., Шелудянов Д. А. Основы информационной безопасности. М.: Горячая линия-Телеком, 2006.
2. Галатенко В. А. Основы информационной безопасности. М.: Интернет-университет информационных технологий, 2006.
3. Галатенко В. А. Стандарты информационной безопасности. М.: Интернет-университет информационных технологий, 2006.
4. Девянин П. Н. Модели информационной безопасности компьютерных систем. М.: Издательский центр «Академия», 2005.

5. *Домарев В. В.* Безопасность информационных технологий. Методология создания систем защиты. К.: ТИД ДИА Софт, 2002.
6. *Куприянов А. И.* Основы защиты информации. М.: Издательский центр «Академия», 2008.
7. *Курбатов В., Скиба В.* Руководство по защите от внутренних угроз информационной безопасности. СПб.: Питер, 2008.

Глава 9 Менеджмент информационной сферы

- 9.1. Становление информационного менеджмента
- 9.2. Основные понятия информационного менеджмента
- 9.3. Международные и российские стандарты в сфере информации менеджмента
- 9.4. Информационный менеджмент как управление информационно-технологическими услугами
- 9.5. Информационный менеджмент как товар (аутсорсинг)

Наличие информационной системы в любой организации не просто дань моде, это реальная необходимость иметь средства, направленные на интенсификацию и обеспечение более продуктивной деятельности. При этом неизбежным становится решение множества вопросов, связанных с разработкой, внедрением и использованием в текущей деятельности компьютерных систем, технологий и служб. В настоящее время происходит осознание необходимости грамотного управления работой информационной системы организации на основе современных методов и средств менеджмента. Выделившееся в этой сфере направление, названное информационным менеджментом, как раз и направлено на решение основных вопросов информационной сферы, при этом оно опирается на разработанные международные стандарты. Глава посвящена управлению в сфере информационных технологий в самом широком понимании этого процесса.

9.1. Становление информационного менеджмента

Использование во всех сферах жизни информационных систем различного назначения породило ряд проблем, которые требовали соответствующих решений. Эти проблемы условно можно сгруппировать в три класса:

- необходимость внедрения;
- эффективность использования;
- продвижение и продажа.

Первый класс проблем связан с принятием решения о необходимости внедрения компьютерных информационных систем, технологий или служб в организации. Эти проблемы встают на первый план при возрастании объема и сложности информации. Так, для магазина с двумя продавцами на вопрос: «Нужны ли информационные технологии?» — ответом, возможно, будет «нет», а в большом магазине продажи компьютеров, компьютерных материалов и комплектующих, скорее всего, — «да». При существенном же увеличении площадей магазина и количества продавцов очевидно, что использование информационных систем в любом случае должно принести пользу.

В государственных структурах ориентация идет на интеграцию в существующую международную информационную среду, на удобство получения гражданами государственных услуг, на оперативность в случае, когда это связано с безопасностью. Вопросы же о том, в каком виде, каких масштабов и на какой технологической базе будут присутствовать информационные технологии (ИТ) в государственной машине, остаются, и на них нужно получить ответы.

Второй класс проблем связан с эффективностью использования компьютерных информационных систем, технологий и служб в организации. Эти проблемы, которые касаются эффективного применения уже имеющихся на предприятии или в учреждении информационных технологий и ресурсов, требуют серьезного анализа по следующим направлениям:

- конкурентное положение организации в области информационных технологий, ее конкурентные преимущества и отставания;
- эффективность использования информационных технологий и возможности ее повышения;
- эффективность вложения средств в оборудование и обучение персонала;
- возможность получения прямой или косвенной финансовой отдачи от вложений в информационные технологии;
- роль службы информационных технологий (ИТ-служба) в решении вопросов, напрямую не связанных с информационными технологиями (выявление новых рынков сбыта, создание новых видов товаров).

Третий класс проблем касается продвижения и продаж компьютерных информационных систем, технологий и служб в организации. Эти проблемы тесно связаны

с представлением об информации и информационных технологиях как о товаре. В этом классе можно выделить следующие направления, требующие своего решения:

- преодоление трудностей восприятия информационных технологий как товара;
- точное определение предмета купли-продажи в торговле информационными технологиями;
- нахождение способов оценки эффективности продаж информационных технологий;
- маркетинг информационных технологий (ИТ-маркетинг), выявления способов получения конкурентных преимуществ в условиях продажи одного и того же товара.

Такого рода проблемы в любой сфере, накапливаясь со временем, вызывают к жизни методологию их разрешения. В финансовой сфере такой методологией является финансовый менеджмент, в производственной сфере — производственный. Проблемы, возникшие в информационной сфере, не так давно привели к появлению методологии информационного менеджмента (ИТ-менеджмент).

9.2. Основные понятия информационного менеджмента

Информационная сфера — это, во-первых, сфера экономики, занятая производством, обработкой, хранением и распространением информации и знаний, во-вторых, совокупность информации, информационной инфраструктуры, субъектов, осуществляющих сбор, формирование, распространение и использование информации, а также системы регулирования возникающих при этом общественных отношений.

Информационный менеджмент — это специальная область менеджмента, охватывающая все аспекты проблемы управления в сфере создания и использования информационных ресурсов.

Поскольку информационные потоки пронизывают сегодня все сферы деятельности человека, для более точного определения необходимо ограничить понимание информационного менеджмента двумя значениями:

- управление информацией (информационными потоками и информационными ресурсами), то есть ИТ-менеджмент, — это автоматизированная технология обработки информации в определенной предметной области;
- управление с помощью информации, то есть ИТ-менеджмент, — это управленческая технология, менеджмент в собственном смысле этого слова.

Общественные отношения, возникающие в информационной сфере при осуществлении информационных процессов, называются *информационными*, в свою очередь, деятельность по осуществлению информационных процессов называется *информационной*.

Рассматривая информационные отношения, необходимо отметить, что, несмотря на все их разнообразие, они происходят между составными частями информа-

ционной сферы. Схематично информационную сферу можно представить в виде пяти областей (рис. 9.1).



Рис. 9.1. Структура информационной сферы

Хотя на схеме между областями проведены четкие границы, в жизни такое разграничение невозможно, поскольку функции областей информационной сферы часто пересекаются, а связи между областями носят характер сложный и взаимозависимый. Одни и те же действующие лица могут выступать в разных ролях и включаться в разные области.

- ❑ *Поиск, получение и потребление информации.* Это основная область, поэтому она расположена в центре информационной сферы. Она иницирует процессы во всех остальных областях информационной сферы и является причиной их возникновения. Кратко эту область можно обозначить словом «потребление».
- ❑ *Создание и распространение исходной и производной информации.* К этой области можно отнести все организации и всех частных лиц, тем или иным способом создающих как исходную информацию (научные факты, персональные данные и т. п., то есть по сути фактические данные), так и производную информацию (получившуюся в результате переработки исходной).
- ❑ *Формирование и подготовка информационных ресурсов, продуктов и услуг.* В этой области происходят упорядочивание, анализ и дополнительная обработка информации. В результате первичная и производная информация превращается в информационный ресурс (например, тематический сайт в Интернете),

продукт (примером может служить правовая система «Гарант») или услугу (например, систематизированный и периодический обзор состояния рынка в выбранной отрасли).

- ❑ *Создание и применение информационных систем и средств, их обеспечение.* Это сфера технологического обеспечения информационных процессов. С одной стороны, эта область может рассматриваться как вспомогательная (производство и потребление информации, оказание информационных услуг существовали и до появления информационных технологий в их современном виде), с другой стороны, эта область сама является мощным источником первичной и производной информации.
- ❑ *Создание и применение средств и механизмов информационной безопасности.* В этой области происходит разработка технических и технологических решений, стандартов и методологий, позволяющих обеспечить целостность, непротиворечивость и защиту принятой информации или информации, подпадающей под тот или иной тип ограничения доступности. Эта же область связана с вопросами защиты человека от опасной и вредоносной информации.

Что же собой представляет *сфера информационного менеджмента*?

- ❑ В узком смысле — это круг задач управления производственного и технологического характера в сфере основной деятельности организации, в той или иной мере использующих информационную систему и реализованные в ней информационные технологии.
- ❑ В широком смысле — это совокупность задач управления на всех этапах жизненного цикла организации, включающая в себя действия и операции как с информацией в различных ее формах и состояниях, так и с организацией в целом на основе информации.

Информационный менеджмент охватывает все аспекты проблемы менеджмента в сфере создания и использования информационных ресурсов.

Цель информационного менеджмента — повышение эффективности деятельности предприятия на основе использования информационных систем и технологий.

Предмет информационного менеджмента — процессы создания, эксплуатации и развития информационной системы предприятия.

Область профессиональной деятельности менеджера — обеспечение эффективного управления информационными ресурсами и информационными системами на уровне организации, обеспечение использования информации как стратегического ресурса, организация систем управления в отрасли информационного бизнеса, совершенствование управления в соответствии с тенденциями социально-экономического развития.

Объектами профессиональной деятельности менеджера являются различные информационные ресурсы и информационные системы организаций экономической, производственной и социальной сфер, информационные ресурсы и информационные системы подразделений систем управления государственных предприятий, акционерных обществ и частных фирм, а также различные организации в сфере информационного бизнеса. Профессиональная деятельность менеджера распространяется также на информационные ресурсы и информационные системы

научно-производственных объединений, научных, конструкторских и проектных организаций, органов государственного управления и социальной инфраструктуры народного хозяйства.

Сферу информационного менеджмента составляет совокупность всех задач управления, связанных с формированием и использованием информации во всех ее формах и состояниях для достижения поставленных перед предприятием целей. При этом должны решаться задачи определения ценности и эффективности использования не только собственно информации (данных и знаний), но и других ресурсов предприятия, в той или иной мере входящих в контакт с информацией: технологических, кадровых, финансовых и т. д. В этих задачах управления в той или иной мере используются информационные системы и реализованные в них информационные технологии.

Концепция информационного менеджмента объединяет следующие подходы:

- *экономический* — вопросы привлечения новой документированной информации рассматриваются исходя из соображений полезности и финансовых затрат;
- *аналитический* — основывается на анализе потребностей пользователей в информации и коммуникациях;
- *организационный* — информационные технологии рассматриваются в их влиянии на организационные аспекты;
- *системный* — обработка информации рассматривается на основе целостного, системно ориентированного, всеохватывающего процесса обработки информации в организации, при этом особое внимание уделяется оптимизации коммуникационных каналов, информации, материальных средств и других затрат, методов работы.

Основные направления информационного менеджмента:

- управление информационной системой на всех этапах ее жизненного цикла;
- стратегическое развитие ИС;
- ИС-маркетинг.

Задачи информационного менеджмента:

- формирование технологической среды информационной системы;
- развитие информационной системы и обеспечение ее обслуживания;
- планирование в среде информационной системы;
- формирование организационной структуры в области информатизации;
- использование и эксплуатация информационных систем;
- формирование инновационной политики и осуществление инновационных программ;
- управление персоналом в сфере информатизации;
- управление капиталовложениями в сфере информатизации;
- формирование и обеспечение комплексной защищенности информационных ресурсов.

Информационный менеджмент в организации решает стратегические, оперативные и административные задачи. К числу стратегических задач относятся: создание информационной инфраструктуры организации и управление информационными технологиями. Оперативные и административные задачи носят более узкий характер, ориентированный на выполнение работ по конкретным направлениям.

9.3. Международные и российские стандарты в сфере информационного менеджмента

Главной задачей информационного менеджмента является информационная поддержка основной деятельности организации.

Проникновение информационных систем и технологий во все области человеческой деятельности вызвали к жизни потребность в изучении и обобщении наиболее успешного опыта информационного менеджмента, выделения эффективных методологий, создания стандартов в этой области.

9.3.1. Стандарт ITIL

Назначение ITIL

Библиотека инфраструктуры информационных технологий (Information Technology Infrastructure Library, ITIL) в настоящее время фактически стала международным стандартом в сфере организации и управления информационными технологиями.

Библиотека ITIL изначально разрабатывалась Центральным агентством по вычислительной технике и телекоммуникациям Central Computer and Telecommunications Agency, CСТА) при правительстве Великобритании. В настоящее время владельцем библиотеки является британская правительственная организация OGC (The Office of Government Commerce), одно из направлений деятельности которой — повышение эффективности и рентабельности использования информационных технологий в государственных учреждениях Великобритании.

Значительная роль в развитии и популяризации ITIL принадлежит также некоммерческому профессиональному сообществу ITSMF (IT Service Management Forum), объединяющему крупные организации, в том числе компании-производители (Microsoft, SUN, HP и IBM), а также частных лиц, работающих в области ИТ.

ITIL включает в себя описание различных видов деятельности в сфере ИТ, таких как управление проектами, управление закупками, ИТ-сервис, и состоит из ряда практических руководств, предоставляющих информацию об эффективном и рациональном использовании различных ИТ-сервисов и предоставлении их потребителям.

Философия библиотеки ITIL основана на общих схемах обеспечения качества (total quality frameworks), предлагаемых Европейской организацией управления

качеством (European Foundation of Quality Management, EFQM) и стандартами серии ISO-9000. Эти системы качества поддерживаются путем предоставления стандартизированного описания процессов с учетом передового опыта ИТ-менеджмента.

Библиотека ITIL предлагает структурированное описание наиболее часто используемых ИТ-процессов, их целей и параметров, а также связей между отдельными ИТ-процессами, однако целью библиотеки ITIL не является предоставление описаний конкретных способов внедрения этих процессов.

Книги библиотеки ITIL

Первоначально библиотека ITIL состояла из нескольких комплектов книг, в каждом из которых описывалось отдельное направление в сфере информационной инфраструктуры. Основой ITIL считались десять книг, посвященных поддержке и предоставлению услуг.

Сейчас библиотека ITIL состоит из восьми томов:

- Service Support;
- Service Delivery;
- Planning to Implement Service Management;
- Application Management;
- ICT Infrastructure Management;
- Security Management;
- Software Asset Management;
- The Business Perspective: The IS View on Delivering Services to the Business.

Основные принципы ITIL

Организация деятельности ИТ-службы осуществляется с использованием процессного подхода. Задачей ИТ-службы является предоставление основному бизнесу полного набора информационных услуг.

ИТ-сервисы поставляются на основании соглашения об уровне предоставления сервисов (service level agreement), которое является согласованным и утвержденным документом.

Качество предоставления ИТ-сервиса является измеряемой величиной.

Преимущества ITIL

Преимущества ITIL заключаются в следующем:

- использование передового опыта и проверенных знаний;
- направленность деятельности ИТ на решение задач бизнеса;
- использование ИТ-служб поставщиками ИТ-услуг для бизнес-подразделений;
- регламентирование деятельности ИТ соглашением об уровне услуг;
- стандартизация работы ИТ-персонала;

- направленность на обеспечение оптимального качества ИТ-услуг для потребителей;
- использование подходов менеджмента качества в управлении ИТ-сервисами;
- возможность подтверждения стоимости ИТ-сервиса на основании соглашения об уровне обслуживания.

Библиотека ITIL постоянно пополняется и дорабатывается с учетом нового опыта и знаний, полученных в индустрии оказания ИТ-услуг. Передовые методы ITIL, позволяющие повысить эффективность управления ИТ-инфраструктурой, на сегодняшний день используются большинством крупных мировых компаний.

Процессы ITSM

Управление ИТ-услугами (IT Service Management, ITSM) объединяет 10 процессов, описанных в книгах Service Support и Service Delivery библиотеки ITIL (табл. 9.1).

Таблица 9.1. Управление ИТ-услугами

Процесс	Цель
Управление инцидентами (incident management)	Устранение инцидентов в предельно сжатые сроки. Инцидент — это любое событие (сбой, запрос на консультацию и т. п.), которое может привести к снижению качества предоставления услуги. Для успешного управления инцидентами необходимо создание диспетчерской службы (service desk), которая должна являться единой точкой контакта с пользователями и координировать вопросы устранения инцидентов
Управление проблемами (problem management)	Сокращение количества инцидентов за счет выявления и устранения причины их возникновения
Управление конфигурациями (configuration management)	Создание и поддержание в актуальном состоянии логической модели инфраструктуры компании
Управление изменениями (change management)	Проведение изменений с наименьшим риском возникновения инцидентов, вызванных изменениями
Управление релизами (release management)	Обеспечение работоспособности производственной среды при внедрении изменений
Управление уровнем сервиса (service level management)	Определение состава и уровня сервиса на основании требований потребителей и поставщиков ИТ-сервисов, контроль за достижением установленного уровня сервиса
Управление финансами (financial management for IT services)	Обеспечение стабильного финансирования всех прочих процессов
Управление мощностью (capacity management)	Исключение инцидентов, возникающих по причине недостаточной мощности ИТ-инфраструктуры компании, причем без неоправданных затрат на приобретение излишних, неиспользуемых мощностей
Управление непрерывностью (IT service continuity management)	Обеспечение гарантированного восстановления уровня ИТ-сервисов до величины, необходимой для продолжения бизнес-операций, за определенный промежуток времени в случае чрезвычайной ситуации (пожара, наводнения, отключения электроэнергии)
Управление доступностью (availability management)	Обеспечение согласованного уровня доступности сервиса, а также оценка текущей доступности сервиса и планирование действий, направленных на его дальнейшее улучшение

9.3.2. Стандарт CobiT

Стандарт CobiT (Control Objectives for Information and Related Technology — контрольные цели для технологии обработки информации и смежной технологии) представляет собой набор документов, в которых изложены международные стандарты управления, контроля и аудита информационных систем любого масштаба и сложности.

Открытый стандарт CobiT разрабатывается фондом ISACF (Information Systems Audit and Control Foundation — фонд аудита и контроля информационных систем) и поддерживается ассоциацией ISACA (Information Systems Audit and Control Association — ассоциации аудита и контроля информационных систем).

CobiT и ITIL

ITIL и CobiT являются открытыми стандартами и не зависят от конкретных производителей, платформ и технологий. Данные стандарты описывают часто используемые ИТ-процессы. Однако библиотека ITIL направлена в основном на управление ИТ-процессами, а стандарт CobiT в первую очередь предназначен для контроля и аудита информационных систем компаний.

Еще несколько принципиальных отличий:

- ❑ CobiT предоставляет инструменты управления ИТ-процессами более высокого уровня по сравнению с ITIL. В ITIL приводится подробное описание процедур, направленных на внедрение ИТ-процессов на уровне взаимодействия «директор по ИТ — руководители подразделений». Стандарт CobiT ориентирован на уровень взаимодействия «куратор ИТ от бизнеса — директор по ИТ».
- ❑ CobiT рекомендует стандартные механизмы управления и аудита в сфере ИТ, а стандарт ITIL основан на лучшем практическом опыте построения и совершенствования ИТ-процессов.
- ❑ Рекомендации ITIL в основном используются для управления процессом построения информационной системы, а рекомендации CobiT — для комплексного управления ИТ-процессами и для их аудита.

Структура CobiT

Объекты контроля CobiT строго структурированы, выделяется четыре базовые группы (домена): *Планирование и организация, Проектирование и внедрение, Эксплуатация и сопровождение, Мониторинг и оценка*. Домены подразделяются на 34 подгруппы, которые, в свою очередь, делятся на 318 объектов контроля.

Стандарт CobiT изложен в шести книгах:

- ❑ *Резюме для руководителя*. Краткое описание стандарта CobiT для топ-менеджеров организации, принимающих решения об использовании стандарта в организации.
- ❑ *Описание структуры*. В книге приводится подробное описание структуры стандарта, высокоуровневых целей контроля и комментариев, необходимые для эффективной работы со стандартом.

- ❑ *Объекты контроля.* Книга подробно описывает объекты контроля, предлагаемые стандартом CobIT.
- ❑ *Принципы управления.* Книга предназначена для руководителей ИТ-служб и рассматривает вопросы ИТ-менеджмента, включая определение целей и способов их достижения.
- ❑ *Принципы аудита.* В книге изложены правила проведения ИТ-аудита и даны рекомендации о том, у кого можно получить необходимую информацию, как ее проверить, какие вопросы задавать. Книга адресована внутренним и внешним ИТ-аудиторам, а также консультантам в сфере ИТ.
- ❑ *Набор инструментов внедрения стандарта.* Книга содержит практические советы по использованию стандарта в ИТ-управлении и ИТ-аудите. Ориентирована на внутренних и внешних ИТ-аудиторов, а также ИТ-консультантов.

Преимущества CobIT

Стандарт CobIT предлагает единый подход к сбору и анализу информации, подготовке выводов и заключений на всех этапах управления, контроля и аудита в сфере ИТ. CobIT позволяет использовать лучший опыт аудита и контроля в этой сфере с учетом существующих ИТ-процессов.

Перечислим преимущества модели процессов управления и аудита в сфере ИТ, построенной на основе CobIT.

- ❑ В основе модели лежат бизнес-процессы организации, методики и стандарты аудита производителей программно-аппаратных средств.
- ❑ Процесс — это действие, которое направлено на достижение результата при оптимальном использовании ресурсов и которое может корректироваться в ходе выполнения. Все ресурсы, задействованные в процессе, структурируются таким образом, чтобы максимально эффективно выполнять данный процесс.
- ❑ В подавляющем большинстве организаций процессы, а особенно их цели, стабильны и по сравнению с организационными объектами (сотрудники, отделы, департаменты и пр.) меняются не часто.
- ❑ Внедрение информационных технологий, как правило, не ограничивается рамками одного отдела или департамента, а затрагивает всю структуру организации и ИТ-службу. Прикладное программное обеспечение является неотъемлемым объектом контроля CobIT и стандартно оценивается в рамках единой структуры с применением единых метрик.

9.3.3. Стандарт MOF

Методология и соответствующий ей стандарт MOF (Microsoft® Operations Framework) разработаны корпорацией Microsoft.

MOF состоит из набора взаимосвязанных «рекомендованных практик», основополагающих принципов и процедур, которые вместе представляют собой полные руководства по обеспечению надежности решений и услуг в сфере ИТ. В составе MOF вы найдете инструкции в форме вопросов, помогающие определить, что

необходимо вашему ИТ-подразделению сегодня, а также мероприятия, которые позволят ему эффективно и результативно работать в будущем.

Инструкции в MOF охватывают все действия и процессы управления ИТ-услугами, включая планирование, разработку, использование, обслуживание и в конечном счете вывод из эксплуатации. В модели MOF эти действия и процессы упорядочены в виде функций управления ИТ-услугами (так называемых SMF-функций), которые группируются по этапам, отражающим жизненный цикл ИТ-услуги. Каждая SMF-функция относится к определенному этапу жизненного цикла и обладает уникальным набором целей и результатов, отвечающих предназначению этого этапа. Готовность ИТ-услуги к переходу на следующий этап определяет управленческий анализ, призванный убедиться, что цели были достигнуты надлежащим образом, а цели ИТ-услуги соответствуют целям организации.

Цель MOF заключается в предоставлении ИТ-подразделениям руководства, помогающих создавать, эксплуатировать и поддерживать ИТ-услуги, обеспечивая получение ожидаемых коммерческих преимуществ от конкретных инвестиций в ИТ с приемлемым уровнем риска.

Модель MOF предназначена для создания среды, в которой компания и ее ИТ-подразделение смогут совместно работать над совершенствованием деятельности и использовать при этом модель, описывающую процессы и стандартные процедуры, направленные на повышение эффективности и продуктивности ИТ-услуг. MOF предлагает логический подход к принятию решений и обмену информацией при планировании, развертывании и поддержке ИТ-услуг.

Описание модели MOF состоит из серии обзоров этапов (фаз жизненного цикла) и руководств по функциям сервисного управления (Service Management Function, SMF), в которых описываются действия по успешному управлению ИТ-услугами. Эти действия начинаются с оценки, предваряющей запуск новой или усовершенствованной услуги, проходят через оптимизацию существующей услуги и заканчиваются выводом устаревшей услуги из эксплуатации.

Руководства написаны для разных целевых аудиторий: ИТ-директоров, руководителей ИТ-подразделений и ИТ-специалистов.

- Обзорные руководства предназначены для ИТ-директоров, которым необходимо получить «общую картину» модели.
- Функциональные руководства, содержащие обзорную информацию и описание рабочих процессов, помогут руководителям ИТ-подразделений изучить стратегии предоставления ИТ-услуг.
- Функциональные руководства с подробным перечнем действий рассчитаны на ИТ-специалистов, занимающихся применением модели MOF на практике.

9.3.4. Стандарт ISO 20000:2005 и его отечественная адаптация — ГОСТ Р ИСО/МЭК 20000

Международный стандарт ISO/IEC 20000:2005, основанный на стандарте BS 15000:2002, разработанном британским институтом стандартов (British Standards Institution, BSI), является обобщением мирового опыта в организации

управления ИТ-сервисами и применим к любой организации вне зависимости от отрасли и размеров, в деятельности которой ИТ-сервисы играют важную роль.

Стандарт определяет требования и взаимосвязанные процессы, необходимые для создания и эффективного применения системы менеджмента. В нем предлагаются универсальные критерии, по которым можно объективно оценивать возможности компании при выполнении требований пользователей с учетом особенностей бизнеса.

Стандарт состоит из двух частей:

- ISO 20000-1:2005 «Information technology — Service management. Part 1: Specification»;
- ISO 20000-2:2005 «Information technology — Service management. Part 2: Code of Practice».

Первая часть представляет собой подробное описание требований к системе менеджмента ИТ-сервисов и ответственности за инициирование, выполнение и поддержку в организациях. Эта часть состоит из 10 разделов и 13 процессов, собранных в 5 ключевых групп:

- *Процессы предоставления сервисов (service delivery process)*. В эту группу входят управление уровнем сервисов, управление непрерывностью и доступностью, управление мощностями, отчетность по предоставлению сервисов, управление информационной безопасностью, ведение бюджета и учет затрат.
- *Процессы управления взаимодействием (relationship processes)*. Эта группа включает в себя управление взаимодействием с бизнесом, управление поставщиками.
- *Процессы разрешения (resolution processes)*. В этой группе разработчики стандарта фокусируются на инцидентах, которые удалось предотвратить или успешно разрешить. Она касается управления проблемами и инцидентами.
- *Процессы контроля (control processes)*. В данной группе рассматриваются процессы управления изменениями и конфигурациями.
- *Процессы управления релизами (release process)*. Здесь речь идет о выработке новых и коррекции уже имеющихся решений.

Вторая часть представляет собой практические рекомендации по процессам, требования к которым сформулированы в первой части. Она является руководством для аудиторов и компаний, намеренных пройти сертификацию, и содержит 10 разделов.

Оценка соответствия ИТ-сервисов требованиям ISO 20000-1:2005 позволяет представить объем нерезализованных требований. Кроме того, удастся запланировать их выполнение с учетом рекомендаций ISO 20000-2:2005, библиотеки ITIL или любой другой методологии. Внедрение ISO 20000-2:2005 или ITIL не является обязательным требованием для соответствия стандарту ISO 20000-1:2005, но использование практик, упоминаемых в ISO 20000-2:2005 или ITIL, делает этот процесс гораздо более простым и ясным.

Стандарт ISO 20000:2005 предлагает модель улучшения процессов — цикл PDCA (более известен как цикл Деминга):

- PLAN — спроектируйте или измените процесс для улучшения результатов;
- DO — осуществляйте выполнение;

- СЧЕСК — проведите измерение и подготовьте отчет о работе;
- АСТ — решите, какие изменения необходимы для улучшения процесса.

Кроме того, в стандарте выдвигаются требования к мере ответственности руководителей компании, предоставляющей ИТ-сервисы, а также к управлению документацией, к компетенции, осведомленности и подготовке персонала.

Чтобы обеспечить интегрированный подход к оказанию высококачественных и эффективных ИТ-услуг, стандарт ISO 20000:2005 предлагает переход к сервисной модели ИТ, который реализуется путем разработки и внедрения формальной системы управления ИТ-сервисами (СУИС). СУИС позволяет повысить уровень капитализации предприятия и способствует его признанию на международном уровне.

После внедрения СУИС предприятие вплотную подходит к сертификации. В ходе сертификации по стандарту ISO 20000 проверяются политика и цели компании, система оценки рисков ИТ-сервисов, используемые процедуры и соответствие требованиям стандарта.

Варианты сертификации:

- *Расширение области регистрации.* Компания после завершения сертификации по ISO 9001:2000 проверяется на соответствие требованиям ISO 20000 и, таким образом, расширяет область регистрации ISO 9001.
- *Получение сертификата на соответствие стандарту ISO 20000.* В данном случае компания проходит сертификацию на соответствие стандарту ISO 20000 отдельно. Представители сертифицирующей организации посещают предприятие с целью установить соответствие его системы управления ИТ-сервисами требованиям ISO 20000. При положительных результатах аудита выдается сертификат на систему, включающий в себя информацию о соответствии этой системы требованиям ISO 20000-1:2005.

Таким образом, ISO 20000:2005 предоставляет, во-первых, средства для оценки и измерения эффективности деятельности ИТ внутри организации. Во-вторых, стандарт направлен на реализацию задач, связанных с проведением тендеров при выборе внешнего поставщика ИТ-сервисов. Независимая сертификация по этому стандарту позволяет организациям продемонстрировать своим клиентам соответствие качества оказываемых услуг лучшим мировым практикам.

9.3.5. Стандарт ISO/IEC 38500:2008

При помощи стандарта ISO/IEC 38500:2008 любое учреждение может модернизировать рабочий процесс посредством эффективного ИТ-менеджмента. Стандарт предназначен в первую очередь высшему руководству организаций, чтобы они могли понимать и выполнять свои правовые, этические и нормативные обязательства в отношении ИТ.

Основная цель стандарта состоит в достижении эффективного, результативного и доступного применения ИТ любыми организациями и учреждениями. ISO/IEC 38500:2008 по ИТ-менеджменту предназначен для использования организациями

всех видов и размеров, от небольших коммерческих предприятий и некоммерческих организаций до крупных госучреждений.

Большинство организаций использует ИТ в качестве основного бизнес-средства. Все составляемые на будущее бизнес-планы не обходятся без ИТ-разработок. Неточная информация может помешать добиться конкурентоспособности или подвергнуть организацию риску несоблюдения законов. Для решения данных проблем был разработан и опубликован стандарт ISO/IEC 38500, в котором представлены подробные разъяснения и инструкции по ИТ-менеджменту.

Стандарт ISO/IEC 38500 способствует проведению политики развития деятельности организации в соответствии со всеми обязательствами (законодательством, общим правом, нормативными актами и контрактными соглашениями) относительно допустимого использования ИТ.

Посредством применения данного стандарта создается инфраструктура эффективного ИТ менеджмента, и тем самым оказывается реальная помощь в реализации организациями юридических, нормативно-правовых и моральных обязательств в сфере использования ИТ, соответствующих также и стандартам ГПН.

Структура стандарта ISO/IEC 38500 включает в себя определения, принципы и модели. В тексте стандарта описаны шесть наиболее эффективных принципов управления ИТ, в которых указаны основные аспекты принятия решений:

- ответственность;
- стратегия;
- приобретение;
- реализация;
- соответствие;
- человеческое поведение.

Посредством стандарта руководство может объективно оценить существующую ИТ-инфраструктуру своей организации, направить и проконтролировать использование информационных технологий и в итоге наладить четкую и продуктивную систему ИТ-менеджмента.

9.4. Информационный менеджмент как управление информационно- технологическими услугами

Информационные технологии в любой организации рассматриваются как комплекс услуг со своими финансово-экономическими показателями, своим набором стандартов и своими метриками, позволяющими измерить не только «что», но и «в каком объеме» делает ИТ-подразделение. Для этого было выпущено несколько стандартов и разработано несколько методологий, позволяющих эффективно управлять информационной сферой на основе цифр и измеримых показателей.

Из них может быть выделена методология CobiT, как специально предназначенная для менеджеров высшего звена и направленная на понимание руководством компаний, что и как делают в их бизнесе информационные технологии. Вместе с методологией, позволяющей руководству компании произвести аудит деятельности ИТ-подразделений и оценивать эту деятельность, был разработан ряд методик, позволяющих самому ИТ-подразделению правильно планировать и реализовывать набор ИТ-услуг, обеспечивающих эффективное, бесперебойное и экономически обоснованное функционирование бизнес-процессов предприятия.

Как только в сфере ИТ появились четкие методологии и стандарты, перечни функций и требований, схемы организации ИТ-служб, стало возможным создать основной инструмент для управления информационными технологиями, который получил обобщенное название *HelpDesk*, или *service desk*, и представляет собой информационную систему технической поддержки и решения проблем пользователей компьютеров, их аппаратного и программного обеспечения.

Крупные организации располагают сегодня сложной инфраструктурой информационных технологий, непрерывное функционирование всех элементов которой на должном уровне является обязательным условием выполнения организацией своих основных функций. Поддержка этой инфраструктуры в рабочем состоянии является одной из основных задач ИТ-служб предприятия. Системы HelpDesk позволяют ИТ-службам обеспечить качественное решение этой задачи.

Задачи, решаемые системой HelpDesk:

- ❑ реализация единой точки обращения к службе поддержки (удобный и понятный для пользователей механизм позволяет направлять запросы непосредственно в службу поддержки, минуя менее эффективные средства разрешения проблем);
- ❑ предоставление стандартного способа *регистрации и выдачи заданий специалистам*;
- ❑ обеспечение контроля за последовательностью исполнения работ, потраченным временем и ресурсами;
- ❑ назначение приоритета запросу в зависимости от его типа, конкретного пользователя или других обстоятельств;
- ❑ эскалация запросов и инцидентов, оповещение соответствующих администраторов;
- ❑ хранение базы знаний по прошлым запросам, позволяющей специалистам быстро разрешать проблемы, схожие с уже возникавшими;
- ❑ ведение отчетности по затратам времени и средств на выполнение запросов.

Отчетность по оказанию услуг может быть использована для формализации отношений между пользователями информационных систем предприятия и ИТ-службой. Ожидаемый уровень поддержки (время реакции на запросы и исполнения запросов, тип предоставляемых услуг) может быть сопоставлен и приведен в соответствие с финансированием и численностью ИС-подразделения.

При помощи отчетности системы HelpDesk могут также выявлять закономерности в потоке поступающих запросов, выделяя таким образом «узкие места» в инфраструктуре предприятия.

Системы HelpDesk могут интегрироваться со средствами учета компьютерного оборудования. Таким образом может осуществляться общий контроль за количеством и типами оборудования, причем всегда можно получить информацию о том, имеется ли в организации оборудование, отвечающее определенным требованиям (например, для замены вышедшего из строя).

В случае, когда на предприятии внедрена система ERP, HelpDesk обычно интегрируется в эту систему в качестве дополнительного модуля.

9.5. Информационный менеджмент как товар (аутсорсинг)

Как только ИТ-менеджмент стал рассматриваться в качестве услуги и у него появились признаки, позволяющие оценивать качество, количество и другие параметры получаемых за деньги выгод, все большее число фирм стало использовать в своей деятельности аутсорсинг информационных технологий, решений и услуг.

ИТ-аутсорсинг — это передача сторонней организации (аутсорсеру, или контактору) полностью или частично функций по обслуживанию информационных потребностей предприятия.

В качестве основания для привлечения в организацию внешнего поставщика ИТ услуг может рассматриваться несколько факторов.

- *Профессиональный фактор.* У аутсорсера имеются лучшие, чем у заказчика, специалисты. Такая ситуация может возникнуть, если, например, специалисты предприятия, вся информационная система которого построена с использованием операционной системы Windows, сталкиваются с необходимостью администрирования или выполнения работ на компьютере под управлением UNIX.
- *Производственно-технологический фактор.* Аутсорсер располагает необходимыми производственными мощностями. В этой сфере можно выделить такие области, как
 - размещение баз данных предприятий на внешних серверах аутсорсера, расположенных в специально оборудованных серверных центрах, способных выдержать стихийные бедствия, суточное отключение электроэнергии или вооруженное нападение, то есть предоставляющих клиентам повышенные гарантии сохранности информации;
 - выполнение аутсорсером работ по созданию сетевой инфраструктуры предприятия;
 - разработка аутсорсером корпоративной информационной системы со специализированной конфигурацией.
- *Финансово-административный фактор.* Аутсорсер имеет возможность управлять проектами или транзакциями так, что это получается быстрее, надежнее и выгоднее, чем у самого заказчика.

- *Географический фактор.* Некоторые виды работ в других регионах можно выполнить гораздо дешевле.

Помимо этих факторов, существенную роль играет также изменение стиля взаимодействия и ожиданий в случае заключения аутсорсингового соглашения по сравнению с общением с собственной ИТ-службой. В случае заключения контракта на предоставление аутсорсинговых услуг все меняется: бизнес попадает в знакомую и привычную ему среду заключения бизнес-договора, где все условия оговариваются заранее, четко прописывается набор и стоимость услуг, известны процесс и ожидаемый результат. Аутсорсер оказывает услуги, он подписан под контрактом, он отвечает за качество, количество, доступность, непрерывность и прочие характеристики ИТ-процессов. Даже если неприятность все-таки случится и оказание услуг окажется задержанным или сорванным, аутсорсер будет отвечать за это в установленном порядке и ему придется оплатить издержки и потери.

Таким образом, когда речь идет об аутсорсинге, в качестве приобретаемого продукта выступает сам факт удовлетворения потребности клиента, а не методы ее удовлетворения.

Помимо уже перечисленных и вполне прозрачных факторов, существует еще ряд аргументов, делающих аутсорсинговый подход к ИТ-менеджменту на предприятии более выгодным, чем содержание собственной полномасштабной ИТ-службы.

- *Мотивация.* Аутсорсер заинтересован в получении прибыли и, в частности, в том, чтобы ему доставалась часть экономии от снижения издержек клиента.
- *Масштабирование.* У аутсорсера, специализирующегося в определенной сфере, всегда есть несколько клиентов со схожими потребностями, что позволяет ему строить инфраструктуру управления стандартизированными способами, снижая при этом как стоимость решений, так и возможные риски.
- *Ориентация на клиента.* Аутсорсер обычно работает с клиентами, имеющими схожие потребности. Это дает ему возможность собирать статистику, изучать структуру и прогнозировать динамику потребностей каждого конкретного клиента.

Что происходит, когда ИТ-служба переводится под аутсорсинговое управление?

Аутсорсер, заинтересованный в получении прибыли, начинает искать способы снижения издержек и повышения эффективности собственно производства. В этих целях обычно

- прекращаются неэффективные разработки;
- закрываются инициативы местного уровня, не оказывающие положительного влияния на бизнес;
- увольняется лишний и неквалифицированный персонал и набирается более квалифицированный или более дешевый;
- укрупняются и более эффективно загружаются вычислительные ресурсы;
- более рационально используются имеющиеся площади;
- модернизируется инфраструктура;
- выстраиваются более эффективные процессы производства услуг.

Несмотря на очевидные преимущества аутсорсинга, покупка ИТ-услуг не всегда является выгодным или успешным решением. Это решение нужно принимать на основе анализа различных факторов. Инструментом, помогающим принять решение об аутсорсинге, является матрица ИТ-аутсорсинга (рис. 9.2).

На этом рисунке вертикальная ось показывает степень стратегической важности (СВ) данного ИТ-направления, продукта или услуги для предприятий. Горизонтальная ось служит для фиксации положения в этой сфере по сравнению с имеющимися на рынке предложениями. Между осями находится поле принятия решений. Как видно из матрицы, решение об аутсорсинге не всегда диктуется лишь соображениями экономии: стратегическая важность определенного ИТ-направления может быть настолько велика, что даже при очень плохом положении дел правильнее будет вложить дополнительные средства в развитие этого направления, чем отдавать его на внешнее обслуживание.



Рис. 9.2. Матрица ИТ-аутсорсинга

Вопросы для самопроверки

1. Опишите основные классы проблем, для решения которых требуется информационный менеджмент.
2. Дайте определение информационному менеджменту.
3. Из каких областей состоит информационная сфера?
4. Что является областью профессиональной деятельности ИТ-менеджера?
5. Что является предметом профессиональной деятельности ИТ-менеджера?
6. Перечислите основные задачи, которые должен решать ИТ-менеджмент.
7. Что такое «библиотека ITIL» и каково ее назначение?

8. Перечислите основные принципы ITIL.
9. Что такое «CobiT», в чем различия CobiT и ITIL?
10. Кто является разработчиком MOF, каково назначение этой методологии?
11. Какая модель положена в основу стандарта ГОСТ Р ИСО/МЭК 20000? Перечислите основные положения этой модели.
12. Какие принципы положены в основу стандарта ISO/IEC 38500:2008? Кто является основным пользователем этого стандарта?
13. Каковы назначение и основные задачи, решаемые системой HelpDesk?
14. Что такое «ИТ-аутсорсинг»?
15. Какие факторы влияют на решение о привлечении ИТ-аутсорсера?
16. Для чего используется матрица ИТ-аутсорсинга?

Литература

1. Аксенов Е., Альтшуллер И. Аутсорсинг: 10 заповедей и 21 инструмент. СПб.: Питер, 2008.
2. Бон, Ян Ван. ИТ-Сервис менеджмент, введение. М.: IT Expert, 2003.
3. Годин В. В., Корнеев И. К. Модульная система для менеджеров. Модуль 17: Управление информационными ресурсами. М.: Инфра-М, 2000.

Часть вторая

Техническая база информатики

Глава 10. Аппаратная часть компьютера	281
Глава 11. Представление данных в компьютере	310
Глава 12. История, состояние и тенденции развития компьютеров и вычислительных систем	330
Глава 13. Основы построения компьютерных сетей	360

Глава 10 Аппаратная часть компьютера

- 10.1. Обобщенная структура компьютера
- 10.2. Устройства ввода
- 10.3. Устройства вывода
- 10.4. Материнская плата
- 10.5. Память компьютера
- 10.6. Центральный процессор
- 10.7. Системный блок

В этой главе будет показано, что представляет собой компьютер, как функциональное и физическое устройство, из каких частей и компонентов состоит, и как функционирует. Кроме этого, будет рассмотрено, при помощи каких устройств компьютер общается с внешним миром (в том числе и с человеком), и где хранит информацию. Конец главы посвящен тому, какое будущее ожидает компьютерные архитектуры в ближайшее время.

10.1. Обобщенная структура компьютера

Несмотря на огромный диапазон размеров и вычислительных мощностей компьютеров, все они так или иначе имеют сходную функциональную структуру. В одних случаях эта структура может быть немного упрощена, в других — расширена, но в общем функции вычислительного устройства и необходимые для вычислений функциональные блоки остаются одними и теми же.

Обобщенная структура компьютера — это схема, которая определяет состав, порядок и принципы взаимодействия входящих в нее блоков.

На рис. 10.1 показана обобщенная структура компьютера, дающая представление об основных функциональных блоках, без которых не может обойтись ни один компьютер.

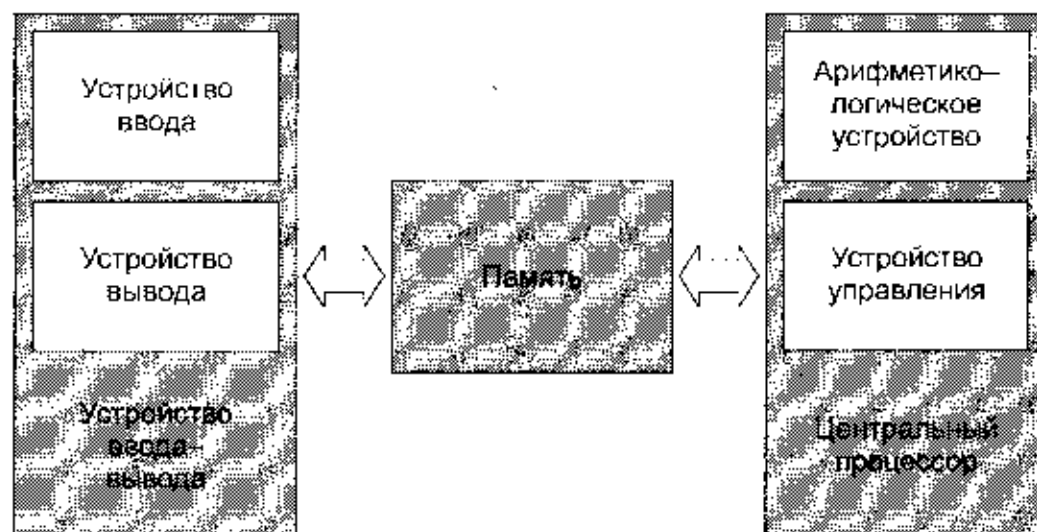


Рис. 10.1. Обобщенная структура компьютера

Для ввода данных в компьютер предназначается *устройство ввода*, для вывода — *устройство вывода*. Вместе эти устройства обычно составляют единый блок — *устройство ввода-вывода данных*.

Независимо от того, будут данные в компьютере храниться годы или секунды, компьютер должен иметь место для их хранения. Соответствующее устройство называется *памятью*.

В компьютере должно быть устройство, которое производит манипуляции над данными, от простейших арифметических операций до сложных статистических, финансовых или научных расчетов. Функционально это устройство состоит из двух блоков: один блок отвечает за производимые вычисления, другой — за правила, по которым данные извлекаются из памяти, над ними производятся операции и результаты этих операций снова возвращаются в память. Устройство, производящее вычисления, называется *арифметико-логическим устройством (АЛУ)*, а устройство, формирующее правила, по которым происходит обработка данных — *устройством управления (УУ)*. Вместе эти две функциональные единицы образуют единый блок, который получил название *центральный процессор*.

Связать между собой процессор, оперативную память, постоянную память (дисковые устройства), устройства ввода-вывода и другие компоненты компьютера напрямую невозможно конструктивно и технологически. Поэтому между всеми этими устройствами располагается *системная шина* — еще один весьма важный компонент, носителем которого является материнская плата. Системная шина представляет собой множество электрических проводников, конструктивно выполненных в виде тончайших проводящих дорожек, в несколько слоев пронизывающих

материнскую плату и соединяющих между собой все электронные компоненты компьютера (рис. 10.2).

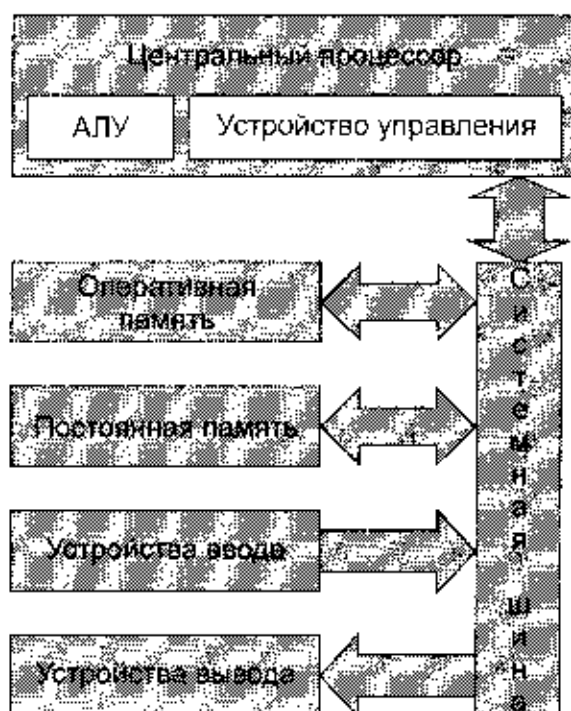


Рис. 10.2. Взаимодействие компонентов через системную шину

Реальные электрическая и компонентная схемы компьютера, конечно же, куда сложнее функциональной схемы. Прежде всего, они усложняются за счет того, что все эти устройства связаны между собой большим количеством соединений. Нужны проводники, чтобы передавать данные, чтобы указывать памяти адрес, из которого следует выбрать данные или в который их следует записать. Компьютер, как мы знаем, не работает без электропитания, значит, нужно устройство, преобразующее переменный ток электрической сети в постоянный низковольтный ток питания компьютера. Устройства ввода-вывода, обозначенные одним прямоугольником, на самом деле весьма разнообразны. Все эти и многие другие устройства должны быть где-то размещены и закреплены. Физически компьютер является собой куда более сложную систему, чем его функциональное описание.

В современном компьютере взаимодействие центрального процессора с устройствами ввода-вывода, а также с различными устройствами, присоединенными к общей шине, основано на механизме прерываний.

Механизм прерываний обеспечивает прерывание выполнения текущей программы при поступлении сигнала на соответствующий вход центрального процессора и выполнение процессором операций обслуживания устройства, вызвавшего прерывание.

При поступлении сигнала прерывания центральный процессор совершает следующие действия:

1. Запоминает адрес текущей инструкции выполняемой программы.
2. Получает вектор прерывания (вектор прерывания — запись в специальной таблице прерываний, связывающей номер прерывания и адрес подпрограммы,

выполняющей обработку прерывания. Получить вектор прерывания значит получить доступ к соответствующей записи в этой таблице).

3. Переходит по адресу, хранимому в соответствующей записи вектора прерывания.
4. Выполняет подпрограмму обработки прерывания.
5. После завершения подпрограммы обработки прерывания возвращается к выполнению прерванной программы.

Подпрограмму обработки прерывания называют *обработчиком прерывания*.

Прерывания обозначаются сокращением *IRQ* (Interrupt Request — запрос на прерывание) и номером. Прерывания имеют приоритет. Это значит, что прерывание с более высоким приоритетом может прервать работу программы обработки прерывания с более низким приоритетом. Прерывания управляются *контроллером прерываний*. Поскольку в компьютерах АТ в качестве контроллера прерывания использовалась микросхема i8259, количество входов в которой было меньше, чем необходимое количество прерываний, два контроллера собирались в единую схему путем каскадного подключения. Хотя в современных компьютерах применяются другие контроллеры прерываний, для сохранения обратной совместимости логическая структура номеров прерываний, номеров и адресов векторов прерываний и их назначение продолжают сохраняться в неизменном виде.

10.2. Устройства ввода

Компьютерный ввод можно подразделить на два основных типа — ввод данных и ввод команд.

Данные:

- необработанный текст (последовательность символов);
- числа;
- изображения;
- аудио данные;
- видео данные.

Команды:

- инструкции;
- программы;
- реакция пользователя.

Ввод информации в современный компьютер осуществляется различными путями и способами. Самое «старинное» устройство ввода — это клавиатура. Современная мультимедийная клавиатура с количеством клавиш больше сотни значительно отличается от своей 84-клавишной прародительницы, поскольку кроме клавиш, позволяющих осуществлять ввод чисел и букв, на ней можно отыскать и клавиши управления медиапроигрывателем, и клавиши управления браузером интернета, и много других интересных командных клавиш.

Клавиатура — это унифицированное устройство ввода со стандартным разъемом и последовательным интерфейсом связи с системной платой.

Последовательный интерфейс — это канал передачи данных между внешним устройством и компьютером, в котором данные передаются последовательно, один бит за другим.

Клавиатура подключается к компьютеру через разъем PS/2 или USB.

Клавиатура имеет внутренний *микроконтроллер*, способный определить факты нажатия и отпускания клавиш, при этом можно нажимать очередную клавишу, даже удерживая несколько ранее нажатых. При нажатии клавиши клавиатура передает идентифицирующий клавишу *скан-код*. При удержании клавиши в нажатом положении через некоторое время клавиатура начинает автоповтор передачи скан-кода нажатия этой клавиши. *Задержка автоповтора* (typematic delay) и *скорость автоповтора* (typematic rate) для современных клавиатур программируются.

Микроконтроллер (или контроллер) — это специализированная микросхема для управления различными устройствами.

Скан-код — это код, передаваемый при нажатии определенной клавиши от клавиатурного контроллера в системный блок компьютера.

Обмен данными между клавиатурой и процессором происходит через установленную на системной плате микросхему контроллера интерфейса клавиатуры. Каждое событие клавиатуры (нажатие или отпускание клавиши) порождает аппаратное прерывание и побуждает процессор считывать скан-код.

Современные клавиатуры поддерживают двусторонний обмен информацией с системной платой: от процессора в клавиатуру передаются команды задания параметров автоповтора, выбора таблицы скан-кодов, управления светодиодными индикаторами и запуска диагностического теста.

Типы клавиатур:

- простая проводная клавиатура без дополнительных клавиш;
- мультимедийная клавиатура имеет дополнительные зоны управления и клавиши для взаимодействия с различными устройствами воспроизведения мультимедийного контента;
- интернет клавиатура обладает дополнительными клавишами, облегчающими взаимодействие с программами просмотра Интернета;
- музыкальная клавиатура, предназначена для тех, кто увлекается созданием компьютерной музыки;
- беспроводная клавиатура не подключается к компьютеру проводом, вместо этого взаимодействие с системным блоком происходит по одному из беспроводных протоколов (IrDA, Wi-Fi или Bluetooth¹);
- встроенная клавиатура является неотъемлемым атрибутом ноутбуков;

¹ IrDA (Infrared Data Association — ассоциация разработчиков систем передачи данных) — стандарт на беспроводную передачу данных в инфракрасном (ИК) диапазоне частот. Wi-Fi (Wireless Fidelity — беспроводная точность, произносится как «вай-фай») и Bluetooth (в переводе с англ. — «синий зуб», произносится как «Блю Туз») являются стандартами (спецификациями) для сетей беспроводной передачи данных.

- экранная клавиатура отображается на экране, нажатие клавиш на такой клавиатуре осуществляется указывающим устройством;
- гибкая клавиатура выполнена в виде тонкого резинового коврика или пленки, ее можно свернуть в трубочку или сложить, как посовый платок;
- проекционная клавиатура позволяет проецировать изображения клавиш на стол или другую поверхность.

На рис. 10.3 показана современная мультимедийная интернет-клавиатура, где хорошо видно, что помимо основной группы клавиш в верхней части клавиатуры и справа присутствуют дополнительные группы клавиш, предназначенные для управления воспроизведением мультимедийного контента и навигации по интернет-страницам.

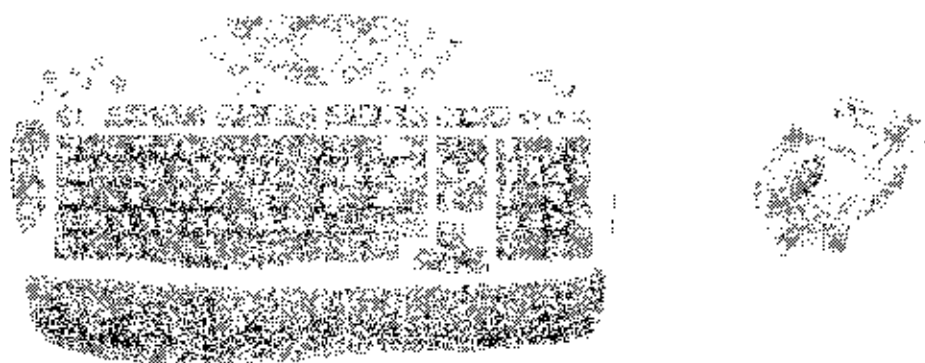


Рис. 10.3. Мультимедийная интернет-клавиатура и мультимедийная мышь

По мере развития компьютерной техники и программного обеспечения, особенно с появлением графических оконных оболочек, крайне актуальным стали указывающие устройства, позволяющие задавать определенные точки или объекты на экране, выбирать (путем щелчка), а также захватывать и перемещать экранные объекты (окна). Наиболее распространенным указывающим устройством сегодня является мышь. Типология мышей почти так же обширна, как типология клавиатур. Мыши бывают:

- двухкнопочные и трехкнопочные;
- проводные и беспроводные;
- механические и оптические;
- многофункциональные.

Так же, как и клавиатура, мышь передает информацию процессору через контроллер. Получив информацию о передвижении мыши, контроллер генерирует прерывание IRQ12, и процессор считывает данную информацию из контроллера.

Примером многофункциональной мыши может служить мышь, представленная на рис. 10.3. На ней помимо правой и левой кнопок имеется еще ряд элементов управления.

Кроме двух основных устройств ввода в современной компьютерной технике существует большое количество устройств, позволяющих вводить данные разного типа в компьютер.

□ *Указывающие устройства*

- Электропное перо выполнено в форме карандаша, внутри которого находится светочувствительная матрица, способная считывать информацию с экрана компьютера.
- Трекбол. Если в обыкновенной механической мыши шарик катается по коврику, то в случае трекбола шар довольно большого размера находится под рукой пользователя. Прокручивая шар в разных направлениях, пользователь перемещает указатель на экране.
- Тачпад — прямоугольная область, по которой пользователь может водить пальцем или осуществлять одинарные и двойные нажатия. Перемещения пальца приводят к перемещению указателя на экране, а одинарные и двойные нажатия воспринимаются как одинарный или двойной щелчок мышью. Тачпадами оснащаются, в основном, ноутбуки, но встречаются также клавиатуры для стационарных компьютеров со встроенным тачпадом.
- Pointing stick — выполнен в виде миниатюрного джойстика с поверхностью, похожей на твердый перепавый ластик. Это устройство располагается в центре клавиатуры (обычно на ноутбуках) и при надавливании на него перемещает указатель по экрану влево, вправо, вверх или вниз.
- Игровые джойстики могут быть самыми разными, от простейших рукояток, нагибающихся в четырех направлениях, до полных симуляторов кабины автомобиля или самолета.

□ *Устройства ввода графических данных*

- Сенсорными экранами мини-устройств оснащены по большей части ладонные компьютеры. Экран выполняет одновременно функции и мыши, и клавиатуры. Функция указывающего устройства выполняется путем прикосновений специальным пером (стилусом) к экрану, при этом имитируется как одинарный, так и двойной щелчок (вместо терминов click и double-click тут применяется термины tap и double-tap). Такой способ ввода информации позволяет обходиться без физической клавиатуры, которая заменяется экранным изображением. Кроме того, дополнительное программное обеспечение позволяет распознавать буквы, нарисованные на экране стилусом от руки.
- Сенсорными экранами больших размеров оснащаются планшетные компьютеры, а также профессиональные машины для компьютерного дизайна. Эти экраны предназначены в основном для ввода графического изображения путем рисования на экране стилусом.
- Скаперы позволяют вводить графическую информацию, считывая ее с бумажного листа.
- Цифровые фотоаппараты.

□ *Устройства ввода звуковых данных*

- Микрофон.

- Линейный вход получает сигнал с выхода звуковой аппаратуры (магнитофонов, проигрывателей, телеприемников и прочих источников звуковых сигналов).
- Midi-клавиатура.
- *Устройства ввода видеoinформации*
 - Цифровые и аналоговые видеокамеры со специальным видео-выходом.
 - Телеприемники.
 - Видеомагнитофоны.
 - Веб-камеры и камеры сотовых телефонов.
- *Устройства ввода радиосигнала* подключаются к высокочастотному входу для приема FM радио- и телевещания
- *Сканирующие устройства для ввода символьной, числовой и биометрической информации* (чтение штрих-кодов, отпечатков пальцев)
- *Устройства считывания данных от датчиков бытовой, технологической и телеметрической аппаратуры*

Разнообразие современных устройств, предназначенных для ввода информации в компьютер, очень велико. Выше перечислены не все устройства, поскольку непрерывно появляются все новые и новые устройства, совершенствующие взаимодействие компьютера и человека. Концепция «умных вещей» и «умных жилищ», появившаяся в конце прошлого века, ныне бурно развивается, и в соответствии с ней возникает все больше и больше каналов, по которым компьютеры могут воспринимать данные и взаимодействовать с человеком.

10.3. Устройства вывода

Существует главное устройство вывода данных из компьютера – монитор. Главенство монитора среди устройств вывода данных вполне объяснимо: человек воспринимает и осознает свыше 90 % всей входящей информации именно визуально. Таким образом, то устройство, которое обеспечивает генерацию изображения, всегда будет наиболее важным среди устройств вывода компьютера.

10.3.1. Мониторы

Мониторы в своем развитии прошли путь от микроскопических по сегодняшним меркам черно-белых (а вернее, черно-зеленых) экранов, работающих исключительно в текстовом режиме, до огромных плоских матриц, размеры которых заставляют отодвигаться от них подальше, иначе невозможно охватить взглядом всю их поверхность.

В настоящее время персональные компьютеры работают с тремя основными типами мониторов:

- мониторы с электрононо-лучевой трубкой;
- жидкокристаллические мониторы;
- мониторы с плазменной панелью.

Монитор с электронно-лучевой трубкой

Конструкция монитора с электронно-лучевой трубкой (Cathode-Ray Tube display, CRT), которая показана на рис. 10.4, абсолютно соответствует названию. Внутри этих мониторов находятся:

- катод, раскаляющий проводник так, чтобы вокруг него возникло облако электронов;
- трубка с тремя магнитными ловушками, создающими из электронов катода три хорошо сфокусированных луча;
- дисплей, покрытый светящимся под воздействием электронного луча слоем.

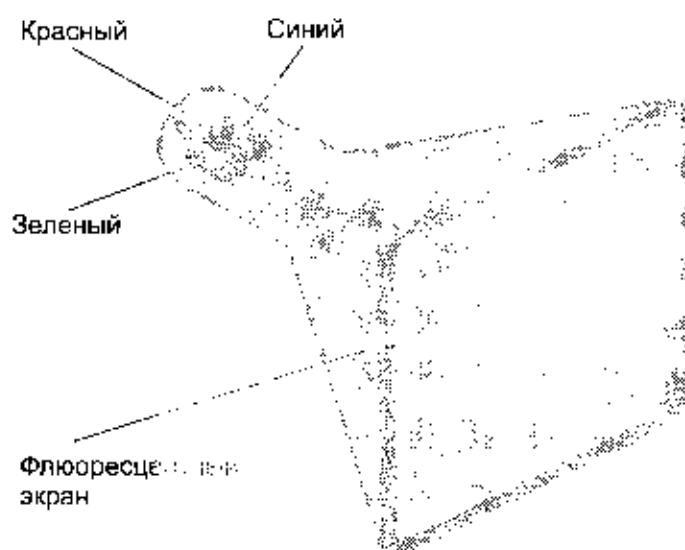


Рис. 10.4. Монитор с электронно-лучевой трубкой

Каждая точка экрана состоит из трех компонентов — красного, зеленого и синего, а каждый из лучей заставляет светиться с заданной яркостью один или несколько этих компонентов. В результате получается картинка, в которой смешение трех основных цветов создает всевозможные оттенки всех цветов.

Среди достоинств современных мониторов с электронно-лучевой трубкой можно отметить только два: они дешевле по сравнению с жидкокристаллическими и плазменными мониторами и могут менять в определенных пределах свое разрешение. Все остальные параметры при сравнении много хуже. Мониторы с электронно-лучевой трубкой занимают больше места, имеют больший вес, потребляют много электроэнергии, генерируют сильные электромагнитные поля, а четкость изображения у них хуже, чем у жидкокристаллических и плазменных мониторов. Эти мониторы постепенно заменяются жидкокристаллическими и плазменными.

Жидкокристаллические мониторы

Жидкокристаллический монитор (Liquid Crystal Display, LCD), или жидкокристаллический дисплей (ЖКД), представляет собой плоский экран, характеризующийся хорошей четкостью, фиксированным разрешением (количеством отображаемых точек), небольшим энергопотреблением, малыми габаритами (в глубину) и весом, к тому же он вполне доступен по цене. При более четком отображении ЖКД обладает куда меньшими показателями излучения от экрана и мощности электромагнитных полей.

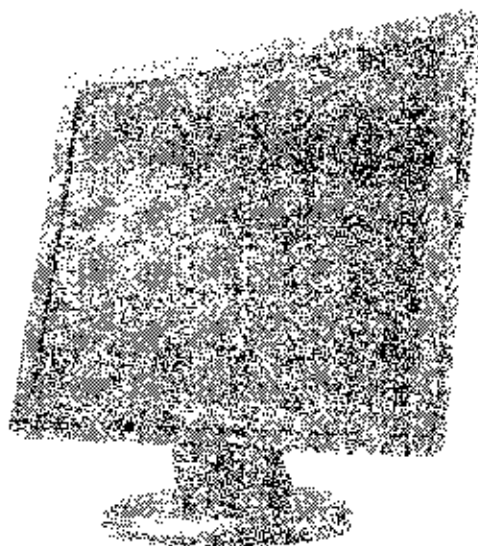


Рис. 10.5. Жидкокристаллический монитор

Принцип действия ЖКД основан на способности физической структуры, названной «жидкими кристаллами», менять свою светопроводность (вернее, цветопроводность) в зависимости от поданного на них напряжения. Таким образом, получая на входе определенный цветовой спектр, жидкие кристаллы, в зависимости от того, какое напряжение на них подается, пропускают только свет определенного цвета. Каждой точке экрана соответствует одна жидкокристаллическая ячейка, которая в результате подачи на нее электрического сигнала становится светофильтром и подсвечивается светом того или иного цвета.

Поскольку матрица ЖКД сформирована из заранее установленного количества ячеек, эти дисплеи имеют фиксированные размер и разрешение экрана. Если для настольных персональных компьютеров и рабочих станций вы можете выбирать между тремя типами дисплеев, то для наладонных устройств и ноутбуков ЖКД на сегодняшний день — единственное приемлемое решение.

Плазменные мониторы

Конструктивно плазменные мониторы, как и жидкокристаллические, имеют вид плоских панелей. Плазменные мониторы работают за счет свечения инертного газа в плазменном разряде. Каждая ячейка (точка экрана) плазменного монитора представляет собой сочетание трех миниатюрных люминесцентных ламп, каждая из которых может светиться своим цветом (красным, зеленым или синим). Для того чтобы плазменный разряд возник, необходимо подвести к электродам довольно высокое напряжение, что делает плазменные дисплеи весьма энергоемкими устройствами. Кроме этого, плазменные панели имеют ограничение по размеру отображаемой точки на экране. На самом деле, если вам необходимо в некотором объеме разместить три люминесцентные лампы, этот объем невозможно уменьшать до бесконечности. Для современных плазменных дисплеев предел составляет около половины миллиметра. Таким образом, с точки зрения четкости изображения для плазменных панелей «чем больше размер, тем лучше». Когда плазменные панели

применяются для представления изображений на большой площади, им нет равных как по четкости изображения, так и по насыщенности цвета.

Плазменные панели потребляют гораздо больше электроэнергии, чем жидкокристаллические, они имеют ограничение по размеру точки экрана и значительно дороже ЖКД. Их применение ограничено экранами телевизоров и мониторами особо крупного размера (скорее для коллективного, чем для индивидуального пользования).

Общая схема вывода изображения

Изображение через системную шину записывается в видеопамять, которая расположена на плате видеоадаптера. Затем видеоадаптер считывает изображение из видеопамяти, преобразуя его в видеосигнал, который и подается на вход монитора. Видеосигнал имеет стандартные параметры, поэтому мы можем подключать к компьютеру мониторы разных типов (с электронно-лучевой трубкой, жидкокристаллические, плазменные), используя один и тот же разъем на одном и том же видеоадаптере.

10.3.2. Принтеры

Первые устройства вывода на печать появились примерно в одно и то же время с устройствами вывода на экран (в 60-х годах прошлого века). На заре истории развития компьютеров принтеры играли более важную роль. Результаты компьютерных вычислений выводились на бумагу — никому не нужны были результаты вычислений на экране. С тех пор многое изменилось, и сейчас вполне можно себе представить персональный компьютер без принтера, но без монитора его представить невозможно. Однако если говорить о важности устройств вывода, то принтер стоит в очереди непосредственно после монитора. Сколь бы ни были красивы созданные изображения на экране, рано или поздно появляется необходимость получить изображение (будь то текст или рисунок), отпечатанное на бумаге.

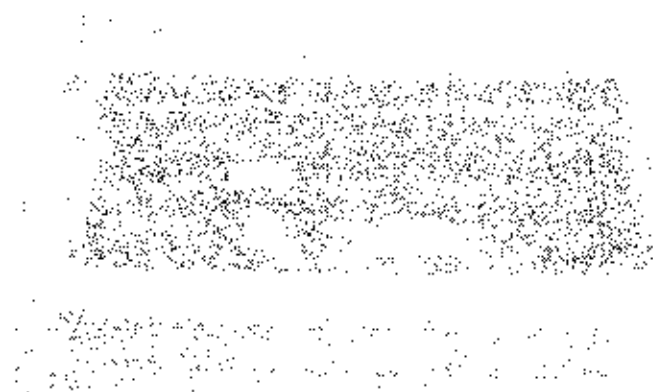
По конструкции и принципу действия принтеры можно подразделить на следующие группы:

- матричные принтеры (ныне почти не встречаются);
- струйные принтеры;
- лазерные принтеры.

Матричные принтеры

Принцип действия матричного принтера очень напоминает принцип действия печатной машинки: есть лента, покрытая красящим веществом, бумага и печатающая головка. Но если в печатной машинке для пропечатывания каждого символа был свой боек с гравировкой, то в матричном принтере символ формируется при помощи иголок, выстреливающих из печатного узла под действием электромагнитов. Иголки выдвигаются сквозь отверстия в прямоугольной матрице, откуда

и пошло название принтера. Таким образом, каждая буква собирается из точек, которые появляются при ударе иголки по ленте. Матричные принтеры прожили довольно долгую жизнь, и изрядную часть своего пути персональные компьютеры в прошлом столетии прошли под аккомпанемент стрекотания матричных принтеров. В сравнении с современными типами принтеров у матричных принтеров практически нет никаких плюсов: они громоздки, производят много шума, потребляют много электроэнергии, лента на них быстро изнашивается, да и остальные части механизма довольно часто выходят из строя. Но самое главное — они медленно печатают. Скорость печати среднего матричного принтера вполне сравнима со скоростью печати хорошей машинистки. Все эти факторы привели к тому, что с появлением струйных и лазерных принтеров матричные принтеры практически сразу ушли в прошлое.



10.6. Матричный принтер

Струйные принтеры

Первыми на смену матричным игольчатым принтерам пришли струйные принтеры. Их принцип действия основан на выбрасывании специальных чернил через сопла размером в микроны под высоким давлением. Все компоненты струйного принтера были разработаны довольно быстро, принцип действия, который применяется в этих устройствах, применялся задолго до их появления для маркировки различного рода упаковок и грузов. Однако для длительной, скоростной и точной печати на бумаге нужны очень маленькие размеры капель, которые выбрасывают сопла, а это значит, что диаметр сопел тоже должен быть очень малым. Необходимость в уменьшении размера капель и диаметра сопел упиралось всего в одну вещь — чернила. Вся история струйных принтеров — это в первую очередь история поисков волшебной формулы чернил. Чернила для струйного принтера должны обладать широким спектром взаимоисключающих свойств. Например, они должны мгновенно высыхать на бумаге и совершенно не сохнуть внутри механизма принтера. Они должны быть достаточно вязкими, чтобы не распыляться при ударе о бумагу, и достаточно текучими, чтобы не застревать внутри сопла. Они должны быть стойкими и несмываемыми там, куда их наносят, и легко удаляться при очистке печатной головки. Недаром сегодня, купив недорогой струйный принтер,

пользователь с удивлением узнает, что картридж с чернилами составляет едва ли не половину стоимости покупки. Среди достоинств струйного принтера можно назвать высокое качество печати, в особенности цветной, при приемлемой цене. Недостатком (в сравнении с лазерными принтерами) являются повышенный шум во время печати (поскольку печатающий механизм, как и в матричном принтере, постоянно перемещается по всей ширине листа) и проблемы, которые могут возникнуть, если принтером долго не пользоваться (чернила со временем в печатных соплах подсыхают).



Рис. 10.7. Струйный принтер

Лазерные принтеры

В лазерных принтерах принцип печати такой же, как у большинства современных копировальных машин. Есть фотобарабан, поверхность которого в ходе подготовки к печати заряжается отрицательно. Затем на фотобарабан лазерным лучом наносится рисунок, то есть создаются точки, лишенные отрицательного заряда. Тонер (красящий порошок), обладающий магнитными свойствами, притягивается к фотобарабану только в тех местах, с которых лазерный луч снял отрицательный заряд. С фотобарабана тонер переносится на бумагу под действием расположенного с обратной стороны бумаги магнитного барабана. На последнем этапе бумага с тоном пропущается сквозь валики, сильно сжимающие и нагревающие ее, так происходит закрепление (запекание) изображения на листе.

У лазерных принтеров практически нет недостатков. Тонер для них достаточно дешев, скорость печати может быть гораздо выше, чем у струйных принтеров, качество изображения практически типографское. Цена на монохромные (черно-белые) лазерные принтеры сравнима с ценой на струйные принтеры. И к тому же лазерные принтеры почти бесшумны (все, конечно, зависит от модели, есть модели, которые при совершенно бесшумном процессе печати производят столько звуков в процессе захвата и выпуска листа, что все преимущество бесшумной печати сводится на нет). Однако преимущество в соотношении цена/качество для лазерных принтеров очевидно только при монохромной печати. Цветные лазерные принтеры весьма дороги, и это вполне объяснимо: для нанесения тонера четырех цветов (черный,

красный, зеленый, синий), из которых формируется цветное изображение, внутри одного цветного принтера практически работают четыре монохромных принтера, выстроенных в одну технологическую цепочку и требующих особо точного механизма позиционирования листа бумаги.

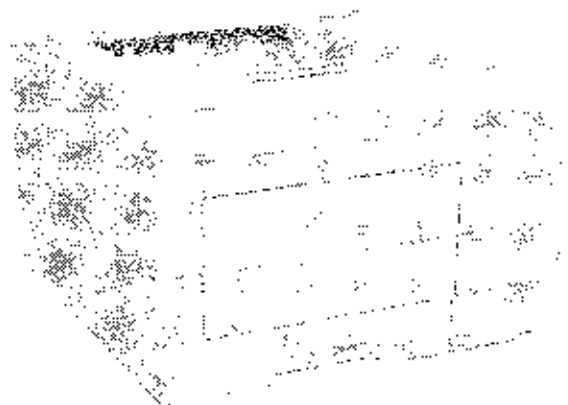


Рис. 10.8. Лазерный принтер

10.3.3. Устройства вывода звука

Звук выводится из компьютера через разъемы видеоадаптера на устройства воспроизведения звука — звуковые колонки или головные телефоны. При этом звук может иметь двойную природу — это может быть естественный звук, хранимый на компьютере в цифровом виде, или звук синтезированный. Для кодирования и хранения в цифровом виде естественный звук должен быть дискретизирован и преобразован в бинарный вид. Этим занимается расположенное на звуковой карте устройство аналого-цифрового преобразования. При воспроизведении оцифрованного звука нужно произвести обратное преобразование (цифроаналоговое), чем занимается цифроаналоговый преобразователь, также расположенный на звуковой карте. Он преобразует поток двоичных данных в аналоговый электрический сигнал, который в колонках или головных телефонах превращается в звук.

Немного иной процесс происходит при записи и воспроизведении синтезированного звука (Musical Instruments Digital Interface, MIDI). Синтезированный звук не нуждается в преобразовании из цифровой формы в аналоговую и обратно. При записи синтезированного звука в память компьютера записывается последовательность чисел, в которых закодировано всего три вещи: нота (высота тона), инструмент и длительность звучания. Такую запись можно сделать напрямую со специального музыкального инструмента (MIDI-клавиатура) или же при помощи программы — редактора MIDI-файлов. При воспроизведении такого звука специальный синтезатор, микросхема которого также расположена на звуковой карте, принимает записанные в память данные и генерирует звук того инструмента, той высоты и той длительности, которым соответствует полученное им число. Само звучание музыкальных инструментов, включая ударные и шумовые эффекты, хранится в MIDI-банках и в нужный момент просто извлекается и подставляется в последовательность воспроизводимых звуков.

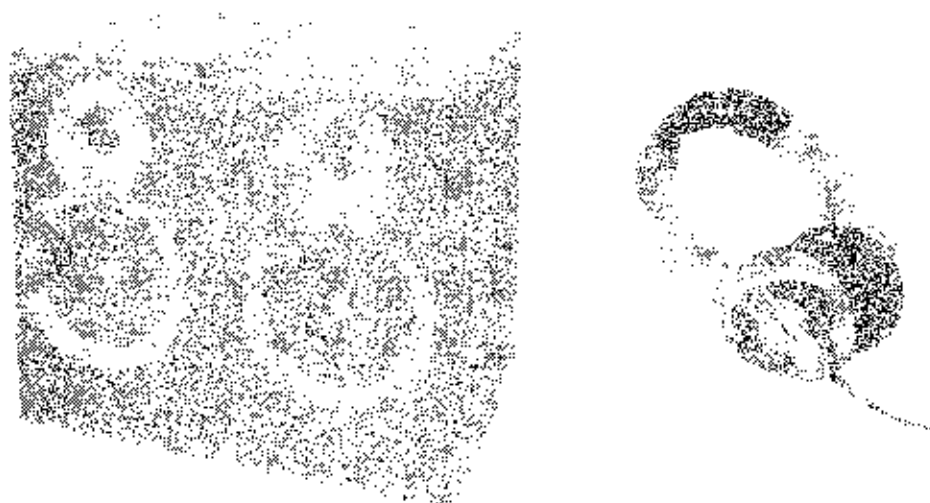


Рис. 10.9. Устройства вывода звука

10.3.4. Другие устройства вывода

Кроме перечисленных выше устройств, к компьютеру можно подключать дополнительные устройства вывода данных различной природы:

- ❑ *плоттер* для вычерчивания чертежей большого масштаба;
- ❑ *механизмы управления* роботизированными устройствами и технологическими процессами;
- ❑ *сигнальные устройства* различного рода;
- ❑ *факс-модем* — в этом случае при подключении к телефонной линии компьютер можно использовать для отправки факсов (в отличие от электромеханического факсимильного аппарата в качестве факса можно отправить изображение, отсканированное задолго до момента отправки).



Рис. 10.10. Плоттер и факс-модем

10.4. Материнская плата

Материнская плата (Mother Board, или MainBoard, MB) — это многослойная печатная плата, на которой устанавливаются основные компоненты персонального компьютера.

Основными компонентами, устанавливаемыми на материнской плате (рис. 10.11), являются: процессор, постоянная память (BIOS), контроллеры внешних устройств и чипсет.

Чипсет (chipset) — это набор микросхем, обеспечивающих подключение центрального процессора к оперативной памяти и контроллерам периферийных устройств.

Современный чипсет строится на базе двух СБИС: «северного» и «южного» мостов. Микросхемы чипсета, установленные на материнской плате, предназначены для работы с определенным типом процессора (или его подтипами). Разъем для процессора (*socket* или *slot*) также однозначно соответствует конфигурации корпуса микросхемы процессора и количеству его ножек. Таким образом понятия «процессор», «материнская плата» и «чипсет» оказываются тесно связанными: невозможно использовать процессор, не предназначенный для установки на данной материнской плате, и, наоборот, невозможно задействовать материнскую плату, не предназначенную для установки на нее определенного типа процессора.

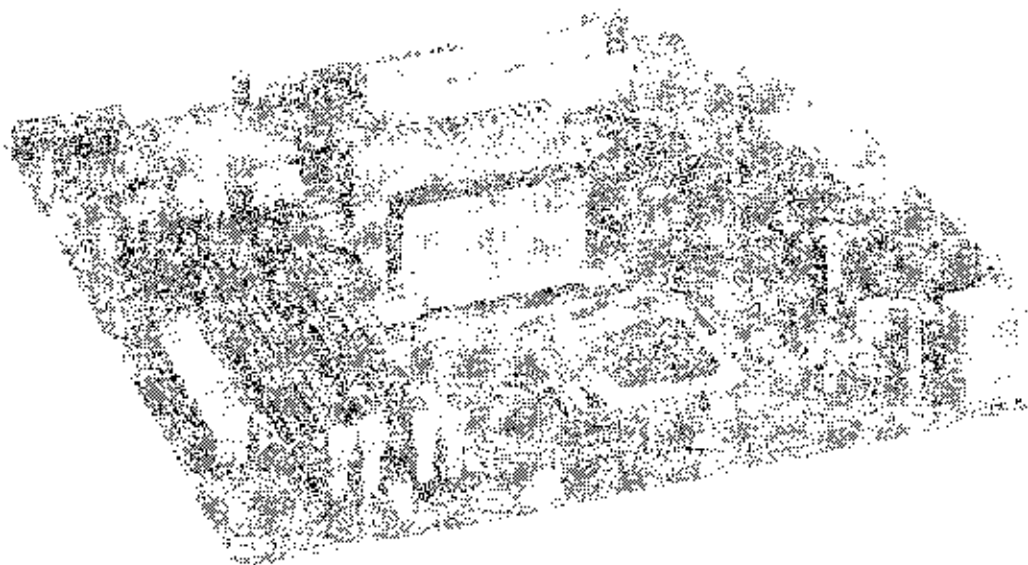


Рис. 10.11. Материнская плата

1 **Интегральная память (ИС)** — электронная схема, изготовленная на полупроводниковом кристалле и помещенная в неразборный корпус. Сверхбольшая ИС (СБИС) содержит до 1 миллиона элементов в кристалле.

10.5. Память компьютера

За функцию хранения данных в компьютере отвечает целый ряд самых разных компонентов, объединенных одним названием — память (запоминающее устройство), крайне различных как по назначению, так и по принципу действия.

Общим для всех компонентов памяти является то, что доступ к хранимым данным осуществляется по адресам, указывающим на *ячейки памяти*.

Ячейка памяти — это минимальный адресуемый элемент памяти компьютера.

В зависимости от принятой системы адресации ячейка памяти может иметь объем от байта до одной из степеней двойки (16, 32, 64 и более) битов.

В современном компьютере можно выделить четыре основных типа памяти:

- внутрипроцессорная память;
- оперативная память, или оперативное запоминающее устройство (ОЗУ);
- постоянная память, или постоянное запоминающее устройство (ПЗУ);
- дисковая память, или память на внешних носителях.

10.5.1. Внутрипроцессорная память

К внутрипроцессорной памяти относятся:

- регистровая память;
- кэш первого и второго уровней.

Регистровая память

Регистровая память имеет самый маленький объем и является самой быстрой. Если быть точным, регистровая память не просто расположена в микросхеме процессора, она является неотъемлемой частью арифметико-решающего и управляющего устройств. Именно в регистры загружается команда для ее декодирования и выполнения, а также данные для выполнения операций над ними. Каждый из регистров процессора представляет собой 32-, 64- или 128-битную последовательность.

Кэш-память

В современных процессорах встроенный кэш первого (L1) и второго (L2) уровней предназначен для временного хранения данных и команд, чтобы при повторном использовании не приходилось через системную шину вновь извлекать данные и команды из основной памяти компьютера. Поскольку производительность процессора всегда больше, чем производительность остальных компонентов, то повторное обращение через системную шину за уже извлекавшимися данными —

нежелательная процедура, которой лучше избежать. При этом кэш первого уровня весьма мал размером, от 8 до 16 Кбайт, но его скорость сравнима со скоростью работы регистров. Кэш второго уровня более объемистый, до 512 Кбайт, но немного более медленный, чем кэш первого уровня. Однако в любом случае кэши первого и второго уровней — гораздо более быстрый вид памяти, чем вся остальная память компьютера, которая находится вне процессорного чипа и к которой необходимо обращаться через системную шину.

Кэш-память играет важную роль в поддержании производительности процессора. Если отключить кэширование, то производительность компьютера может упасть в десятки раз.

По мере развития полупроводниковых технологий тактовая частота процессора увеличивалась гораздо быстрее, чем тактовая частота шины памяти. Для увеличения производительности кэш-память второго уровня старались всячески приблизить к центральному процессору, чтобы иметь возможность увеличить ее частоту по сравнению с частотой шины памяти. Если сначала кэш-память второго уровня выполнялась в виде микросхем, которые устанавливались на системной плате в непосредственной близости от центрального процессора (Pentium I), то затем их стали выполнять внутри корпуса микросхемы процессора (Pentium II), а позже разместили непосредственно на кристалле процессора (Pentium III и Pentium IV). Размещение кэш-памяти второго уровня непосредственно на кристалле практически полностью устранило разницу в быстродействии кэша и регистровой памяти.

Как показано в табл. 10.2, по мере увеличения тактовой частоты процессора кэш-память второго уровня все теснее интегрировалась с центральным процессором.

Таблица 10.2. Динамика развития кэшей первого и второго уровней в семействе процессоров Pentium

Тип центрального процессора	Pentium	Pentium II	Pentium III	Pentium IV
Тактовая частота центрального процессора, МГц	233	450	1400	2530
Длительность цикла (и тактовая частота) кэш-памяти первого уровня, нс (МГц)	4,3 (233)	2,2 (450)	0,71 (1400)	0,39 (2530)
Объем кэш-памяти первого уровня, Кбайт	16	32	32	20
Тип кэш-памяти второго уровня	Модуль на системной плате	На микросхеме	На кристалле	На кристалле
Соотношение тактовой частоты и частоты кэш-памяти второго уровня	—	1/2	1/1	1/1
Длительность цикла (и тактовая частота) кэш-памяти второго уровня, нс (МГц)	15 (66)	4,4 (255)	0,71 (1400)	0,39 (2530)
Объем кэш-памяти второго уровня	Различный	512	512	512
Тактовая частота шины центрального процессора, МГц	66	100	133	533
Длительность цикла (и тактовая частота) шины памяти, нс (МГц)	60 (16)	10 (100)	7,5 (133)	1,9 (533)

10.6.2. Оперативная память

Оперативная память компьютера (Random Access Memory, RAM) — это та область памяти, в которую загружаются программы и данные, откуда процессор берет инструкции для выполнения и данные для обработки. Физически основная память представляет собой платы небольшого размера, вставляющиеся в специальные слоты на материнской плате (рис. 10.12).

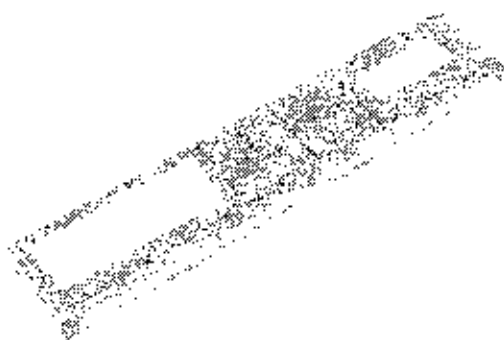


Рис. 10.12. Модуль («планка», «линейка») оперативной памяти

На плате расположены микросхемы, в которые записываются данные. Запись и считывание происходят следующим образом: у платы памяти есть два набора контактов, первый называется шиной адреса, второй — шиной данных. Чтобы прочитать что-то из оперативной памяти, процессор выставляет на шине адреса число, соответствующее адресу, из которого будет производиться чтение. После этого процессор посылает памяти сигнал «готов читать», память выставляет на шину данных значения битовых ячеек, записанные по заданному адресу, и процессор считывает эти значения. Примерно в таком же порядке происходит запись данных в память, только процессор вместо сигнала «готов читать» выставляет на шину данных данные, которые надо записать, и посылает памяти команду «запись».

В зависимости от разрядности шины адреса, *адресное пространство процессора*, или *диапазон адресуемой памяти*, может быть разным. Количество ячеек памяти, к которым может обратиться процессор, подсчитывается по формуле.

$$M = 2^n - 1$$

Здесь n — количество разрядов шины адреса. Для 16-разрядной шины адреса это значение 65 535 (64 Кбайт), для 32-разрядной шины — 4 Гбайт, для 36-разрядной — 64 Гбайт.

В современных компьютерах применяется два основных типа оперативной памяти: статическая (Static Random Access Memory, SRAM) и динамическая (Dynamic Random Access Memory, DRAM). Отличие между ними состоит в том, что динамическая память имеет малое время хранения информации и требует постоянной циклической перезаписи (обновления) информации в своих ячейках. В отличие от DRAM, SRAM может хранить память без перезаписи долгое время (долгое в сравнении с DRAM, статическая память не является энергонезависимой). За более высокую стабильность и скорость SRAM приходится платить более вы-

сокой ценой этого типа памяти, поэтому большинство персональных компьютеров в качестве основной памяти используют DRAM.

Динамическая память, в свою очередь, подразделяется на два типа: синхронную и асинхронную.

Асинхронная динамическая память имеет встроенный генератор импульсов, управляющий всеми сигналами и этапами цикла извлечения и записи данных, в то время как в *синхронной динамической памяти* все сигналы синхронизируются с импульсами системной шины.

Асинхронная память не получила широкого распространения, поскольку по параметрам быстродействия проигрывала синхронной. По этой причине практически вся основная память, которая сегодня применяется в компьютерах, построена на разновидностях синхронной динамической памяти (Synchronous Dynamic Random Access Memory, SDRAM).

Синхронная динамическая память, в свою очередь, имеет несколько разновидностей, которые сменяли друг друга по мере роста быстродействия процессоров: SDR (Single Data Rate – одинарная скорость передачи данных), DDR (Double Data Rate – двойная скорость передачи данных), DDR2 и DDR3.

10.5.3. Постоянная память

В широком смысле постоянная память означает память, которая записывается один раз, причем повторная запись в эту память невозможна. Другое, не менее широкое определение этого термина – память, данные в которой не теряются после выключения компьютера. Однако в более узком смысле этот термин применяют в основном для обозначения микросхемы, в которую записана часть программного обеспечения, называемая BIOS (Basic Input Output System – базовая система ввода-вывода). Микросхемы BIOS можно подразделить на три основных типа:

- ❑ память однократной записи (Read Only Memory, ROM) – микросхема однократно программируется на заводе и больше не подлежит перезаписи;
- ❑ память с возможностью перезаписи (Programmable ROM, PROM) – микросхему можно перепрограммировать («перепрожечь») при помощи специального программатора;
- ❑ память с возможностью перезаписи без извлечения из компьютера (Electrically Erasable Programmable ROM, EEPROM) – микросхему можно перепрограммировать при помощи специального программного обеспечения, подавая на нее определенные команды.

10.5.4. Дисковая память и флэш-память

Несмотря на огромную разницу в конструкции этих двух видов памяти, они рассматриваются в одном разделе. Дисковая память представляет собой несколько дисков из магнитного материала, запись на которые производится, как на магнитную ленту, путем намагничивания определенных зон на диске. Блок дисков вращается с очень большой скоростью.

Флэш-память — это полупроводниковая микросхема, в которой нет движущихся частей, и принцип записи совершенно иной. За последние несколько лет объем одно флэш-чипа возрос от десятков мегабайтов до единиц гигабайтов, а скорости записи и чтения также увеличились на порядок. Возможность объединения нескольких чипов в одном корпусе позволила создавать флэш-карты емкостью в 32, 64 и 128 Гбайт, что вполне сравнимо по емкости с жесткими дисками. Однако в отличие от жесткого диска во флэш-памяти нет подвижных механических частей, поэтому с механической точки зрения срок ее жизни приближается к бесконечности. С электрической же стороны флэш-память может выдерживать примерно миллион циклов перезаписи. Это очень важное ограничение, но прогресс не стоит на месте, и нет никаких сомнений, что данный порог также будет успешно преодолен. На сегодняшний день в мире уже выпускается множество устройств, в которых флэш-память заняла место жесткого диска. В то же время появились жесткие диски размером, сравнимым с размерами флэш-карты. В конце концов, оба этих устройства имеют одну и ту же назначение — они должны хранить данные и программы при отключении питания компьютера.

На рис. 10.13 рядом показаны миниатюрный жесткий диск и флэш-карта стандартного размера.



Рис. 10.13. Сравнение миниатюрного жесткого диска и флэш-карты

10.6. Центральный процессор

Центральный процессор (ЦП) или Central Processing Unit (CPU) — это часть аппаратного обеспечения компьютера, отвечающая за выполнение операций, заданных программами.

Центральные процессоры современных компьютеров обычно выполняются в одном корпусе в виде одной *сверхбольшой интегральной схемы (СВИС)*. Количество транзисторов внутри одного современного процессора — от десятков до сотен миллионов.

10.6.1. Классификация процессоров

С момента создания первого однокристалльного (то есть выполненного как одна микросхема) процессора (процессор 4004, выпущенный корпорацией Intel в 1971 году) темпы развития технологий в этой отрасли промышленности по-

стоянно увеличиваются. Поиски решений, нацеленных на достижение как можно большей продуктивности и эффективности, ведутся множеством фирм и исследовательских коллективов практически во всех возможных направлениях. Интенсивная работа на таком широком фронте привела к созданию тысяч разновидностей процессоров. На схеме, приведенной на рис. 10.14, показаны только наиболее важные критерии, по которым можно классифицировать современные процессоры.

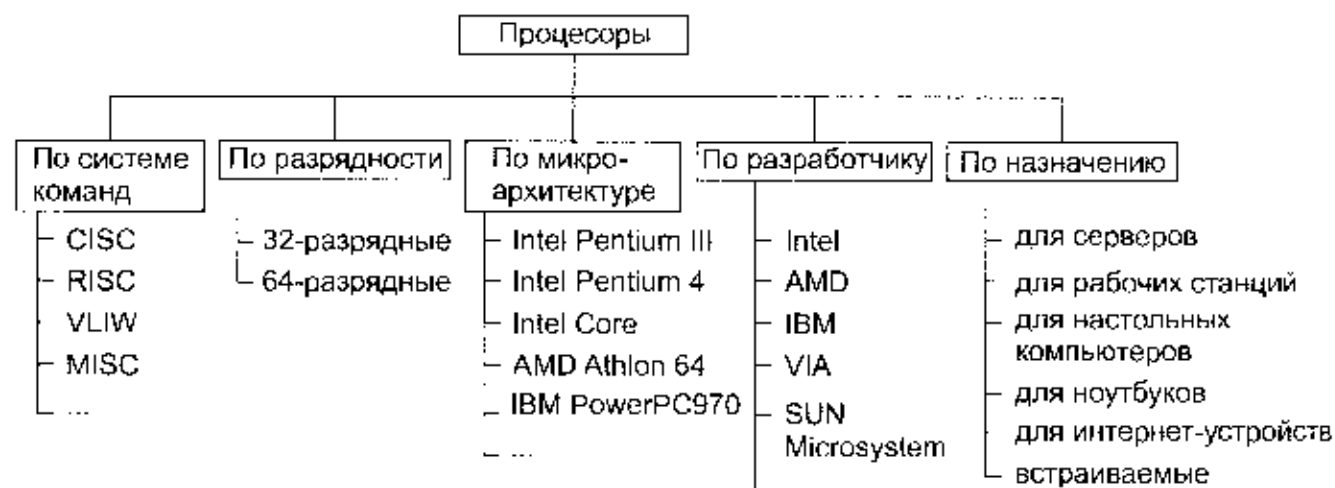


Рис. 10.14. Классификация процессоров

На рис. 10.14 используются следующие сокращения:

- ❑ *CISC* (Complex Instruction Set Computing) — процессоры с полным набором команд;
- ❑ *RISC* (Reduced Instruction Set Computing) — процессоры с сокращенным набором команд;
- ❑ *VLIW* (Very Long Instruction Word) — процессоры с набором длинных команд, у которых одна команда содержит описание сразу нескольких операций;
- ❑ *MISC* (Minimal Instruction Set Computer) — процессор с минимальным набором длинных команд.

10.6.2. Работа центрального процессора

Как центральный процессор осуществляет выполнение компьютерной программы и манипуляции данными?

Начнем с понятия машинного цикла. В центральном процессоре существует специальная область памяти (регистр), которая называется счетчиком команд. Когда включается компьютер, в счетчике команд находится число 0 (или другое фиксированное значение), которое соответствует адресу первой ячейки памяти. Центральный процессор «смотрит» в эту ячейку памяти, интерпретирует число, которое там находится, как команду, считывает данные, указанные командой, выполняет над ними операцию, записывает результат операции в память, увеличивает счетчик команд, считывает следующую команду по адресу, который указан

в счетчике команд, и все повторяется снова. Эта последовательность операций, изображенная на рис. 10.15, носит название машинного цикла.

Один из элементов процессора задает темп его работы — это генератор тактовой частоты. Каждый раз, когда генератор вырабатывает импульс, процессор выполняет одно действие. На рисунке 10.15 обработка одной инструкции процессора состоит как минимум из 8 действий (возможно и большее их число, если имеют место повторения, обозначенные пунктирными стрелками).

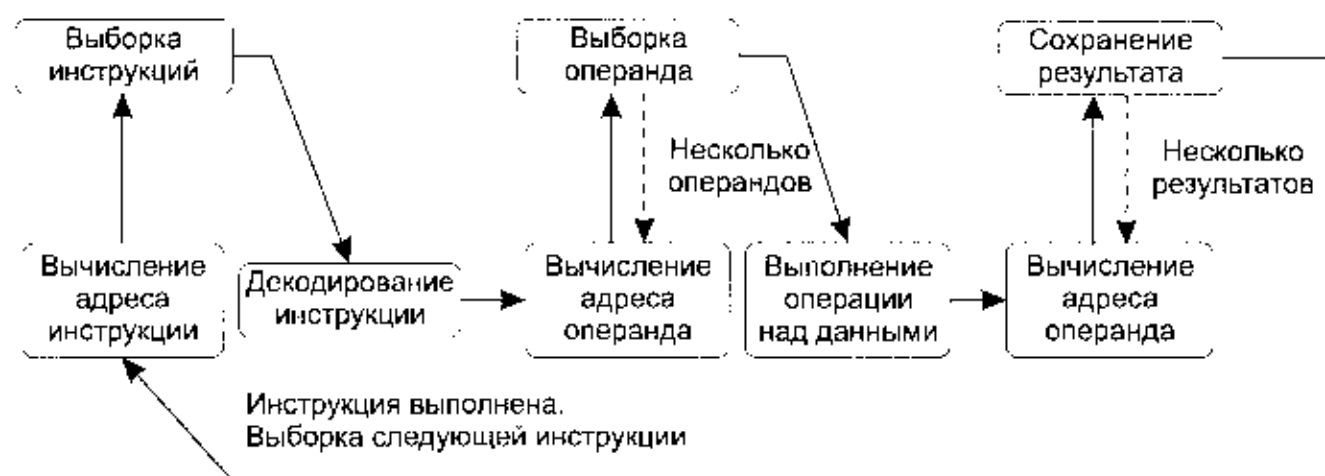


Рис. 10.15. Машинный цикл

10.6.3. Структура центрального процессора

В общем виде структура центрального процессора, соответствующая процессорам Intel 80486, представлена на рис. 10.16.

На рисунке видны три функциональных блока, из которых состоит процессор.

- *Операционный блок* предназначен для выполнения простейших действий над операндами. В нем располагаются:
 - *арифметическое логическое устройство (АЛУ);*
 - *регистры операндов* (буферный регистр и регистр-сдвигатель),
 - *регистр результата* (аккумулятор);
 - *блок регистров общего назначения (РОН);*

Два триггера, $Tz1$ и $Tz2$, нужны для временного хранения перенесенного из старшего разряда бита.

Триггер — это электронная схема, способная хранить один бит информации.

Арифметико-логическое устройство выполняет несколько простейших арифметических (сложение, вычитание) и логических (И, НЕ, ИЛИ) операций. Более сложные операции, для которых аппаратная реализация отсутствует, выполняются программно с использованием для хранения промежуточных результатов блока регистров общего назначения.

- **Блок управления** определяет порядок выборки инструкций, выполняя следующие функции:
 - считывает и запоминает текущую команду;
 - формирует адрес следующей команды;
 - реализует выполнение поступившей команды в соответствии с ее алгоритмом (декодируя выбранную из памяти инструкцию, блок управления разбивает ее на ряд последовательных *микрокоманд*, то есть элементарных операций, которые умеет выполнять АЛУ, формируя таким образом *микропрограмму* для выполнения данной инструкции);
 - управляет обменом информацией с внешними устройствами по системной шине.

В блоке управления под аббревиатурой *БФУС* скрывается блок формирования управляющих сигналов.

- **Интерфейсный блок** согласует работу процессора и системной шины (системная шина объединяет в себе шину данных, шину управления и шину адреса).

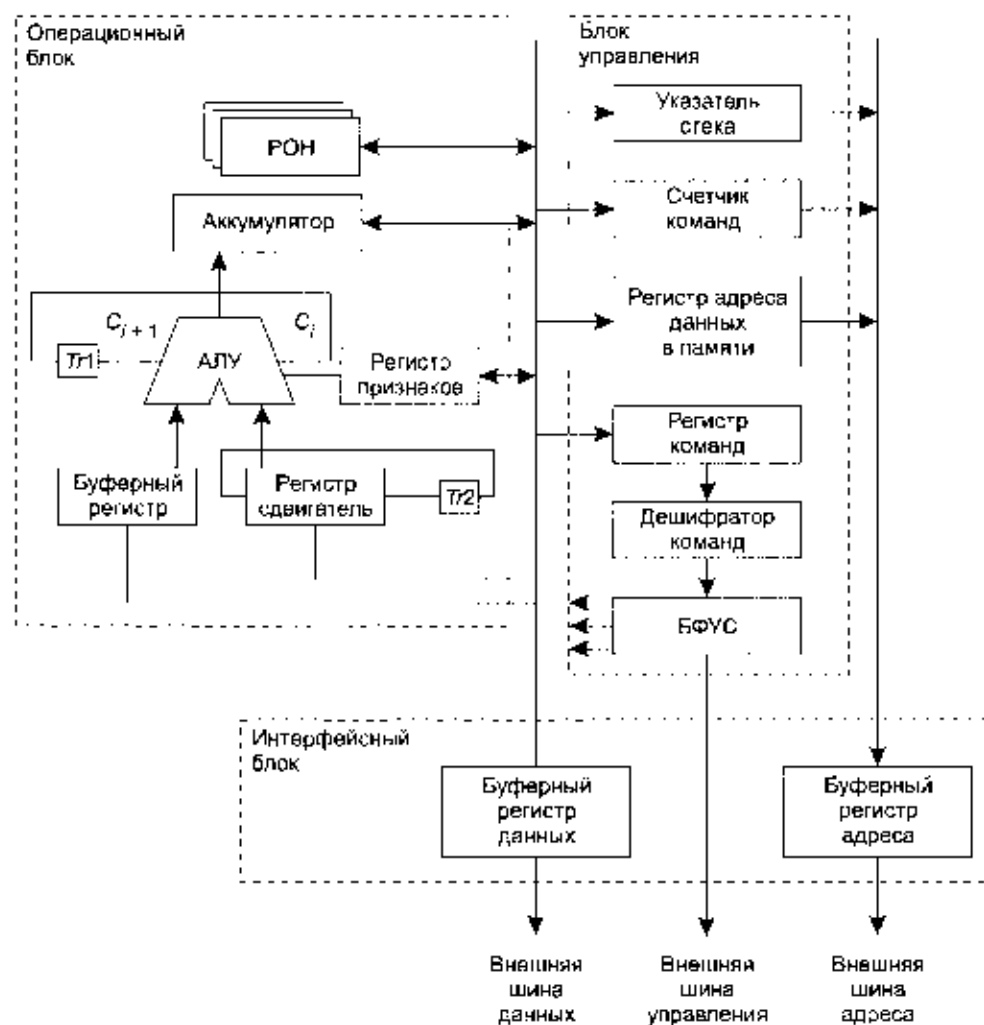


Рис. 10.16. Структура центрального процессора

10.6.4. Пути повышения производительности центрального процессора

Производительность центрального процессора выражается в миллионах элементарных операций, выполняемых за секунду, и измеряется в *mips* (Million Instructions Per Second).

Повышение тактовой частоты процессора. В приведенной упрощенной функциональной схеме процессора вся работа выполняется процессором последовательно, согласно командам блока формирования управляющих сигналов (БФУС), который в свою очередь выполняет микропрограммы выбранных из памяти и декодированных инструкций. При такой микроархитектуре единственным путем повышения производительности процессора является *увеличение количества выполняемых микрокоманд в единицу времени*, то есть путем повышения тактовой частоты процессора. Именно это вызывает постоянное уменьшение размеров и увеличение плотности расположения полупроводниковых элементов на кристалле СБИС (сверхбольшой интегральной схемы). Чем меньший размер имеют элементы, тем более высокую частоту работы они могут обеспечить.

Конвейерная обработка. В случае, когда микроинструкции выполняются последовательно, одна за другой, работу процессора можно сравнить с работой ремесленника-одиночки: он выполняет микрооперацию при помощи одних инструментов, а другие в это время простаивают. Принцип конвейерной обработки полностью соответствует конвейеру на промышленном производстве: инструкции выбираются каждый такт, их выполнение передается от одного инструмента к другому, таким образом все инструменты процессора оказываются равномерно загруженными, а результат выполнения инструкций (как и их выборка) формируется во время каждого такта работы процессора (рис. 10.17).



Рис. 10.17. Конвейерная обработка команд

Суперскалярная архитектура. При наличии в процессоре нескольких блоков декодирования и выполнения команд множество команд может выполняться параллельно.

Симметричная многопроцессорность (Symmetric MultiProcessor, SMP). При поддержке симметричной многопроцессорности в одном корпусе (на одной материнской плате) размещают несколько процессоров для параллельной обработки заданий.

Многоядерность. На одном кристалле (в одном корпусе процессора) размещают несколько практически полностью независимых процессоров, предназначенных для параллельного выполнения заданий.

10.7. Системный блок

В зависимости от того, о какой категории компьютеров идет речь, системный блок может быть выполнен в виде автономного блока (например, у настольных персональных компьютеров), составлять единое целое с клавиатурой, но иметь откидной, закрепленный на нем монитор (ноутбук), или быть монолитным, включающим в себя все составляющие компьютера (паладошные и некоторые типы планшетных компьютеров).

Системный блок служит одновременно и защитным корпусом, и тем механическим скелетом, на котором крепятся остальные функциональные узлы компьютера:

- ☐ источник питания;
- ☐ материнская плата;
- ☐ жесткие диски;
- ☐ дисководы;
- ☐ CD-приводы;
- ☐ вентиляторы (кулеры);
- ☐ карты адаптеров.

На рис. 10.18 показана типичная конструкция системного блока персонального компьютера.

Цифрами на рисунке обозначены следующие компоненты:

1. Процессор.
2. Лямейка оперативной памяти.
3. Блок питания.
4. Устройство для чтения-записи CD-DVD.

5. Кулер (вентилятор охлаждения процессора).
6. Жесткий диск.
7. Сетевой адаптер (специальное устройство, обеспечивающее соединение с компьютерной сетью), устанавливаемый на материнскую плату.
8. Материнская плата.

Сегодня к компьютеру можно подключить столько разных устройств, что пользователь невольно теряется перед огромным количеством разного рода гнезд, отверстий и штекеров. На самом деле, производители компьютерного оборудования сделали все, чтобы любой разъем было просто невозможно подключить в «неправильное» место. Все разъемы на компьютере отличаются размерами, количеством пожек и диаметром отверстий. В редких случаях, когда разъемы у разных устройств совпадают по конструкции (например, у мыши и клавиатуры или у микрофона и колонок), они обязательно маркируются цветом, который совпадает с цветом разъема подключаемого устройства. Рисунок 10.19 иллюстрирует основные типы разъемов компьютера.



Рис. 10.18. Конструкция системного блока персонального компьютера

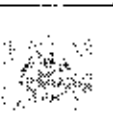
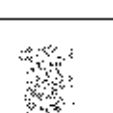
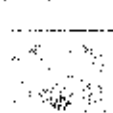
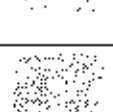
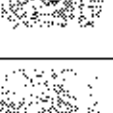
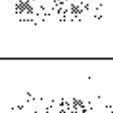
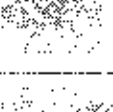
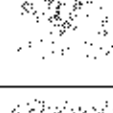
Аудиовыход (красный)		Мышь (зеленый)	
Кабельное телевидение		Сетевой кабель	
Композитный видеоканал		Принтер	
FireWire		Serial	
FM- приемник		Колонки (зеленый)	
Головные телефоны		S-Video (вход)	
Клавиатура (синий)		S-Video (выход)	
Микрофон (красный)		Телефонная линия (модем)	
Монитор		USB	

Рис. 10.19. Основные типы разъемов компьютера

Вопросы для самопроверки

1. Из каких элементов состоит процессор компьютера?
2. Что такое «прерывания» и зачем они нужны?
3. Какие типы мониторов вам известны?
4. Какие типы принтеров вам известны?
5. Какие шаги составляют машинный цикл?

6. Каковы способы увеличения производительности работы центрального процессора?
7. Что такое «микрокоманда»?
8. Каково назначение материнской платы?
9. Что такое «чипсет»?
10. Перечислите все известные вам устройства для ввода данных в компьютер.
11. Перечислите известные вам устройства для вывода данных.
12. Какие типы памяти вам известны?

Литература

1. Гук М. Ю. Аппаратные средства IBM PC. СПб.: Питер, 2006.
2. Кристи Дэрил и Виченцио Карлитто. Собери компьютер сам. М.: ИТ-Пресс, 2007.
3. Мураховский В. И. Железо ПК. Новые возможности. СПб.: Питер, 2005.
4. Таненбаум Э. Архитектура компьютера. СПб.: Питер, 2007.
5. Хамахер К. и др. Организация ЭВМ. СПб.: Питер, 2005.

Глава 11 Представление данных в компьютере

11.1. Форма представления данных в компьютере

11.2. Система счисления

При изучении любой темы, непосредственно связанной с компьютером, так или иначе речь все время будет идти о данных: компьютер обрабатывает и хранит данные, операционные системы и программы манипулируют данными, данные передают по сетям, данные хранят, данные охраняют, данными управляют, данные используют. Понимание того, каким образом данные представлены внутри компьютера, как они хранятся, извлекаются и в какой момент становятся информацией, — важно для освоения всего курса информатики.

11.1. Форма представления данных в компьютере

11.1.1. Общее представление

В каком виде хранится, как перерабатывается информация в мозгу человека, откуда появляется там еще одно нематериальное понятие, «смысл», в обозримом будущем наука вряд ли сможет достаточно аргументировано объяснить. А вот в каком виде информация хранится в компьютерах, что именно передается, кодируется, декодируется, извлекается, переводится в сигналы и посылается человеку, мы знаем точно — это данные.

Данные в компьютере — это сигналы, зафиксированные на физическом носителе в форме, пригодной для постоянного хранения, передачи и обработки.

Приведенное определение данных достаточно широко, чтобы отнести его не только к хранению и обработке при помощи компьютера. Когда в древние века люди вели учет казны на глиняных табличках, у них не было компьютеров, но уже были данные, ибо они фиксировали информацию на физическом носителе, в форме, пригодной для постоянного хранения и использования. В наше время этот процесс претерпел значительные усовершенствования, причем как на стадии преобразования информации в данные, так и на стадии хранения и передачи. В качестве физического носителя теперь используются различного рода электронные и магнитные средства, для обработки данных применяются компьютеры, а для передачи — различного рода компьютерные сети и другие радиоэлектронные устройства.

В то же время смысловое отношение между понятиями «информация» и «данные» сохраняется, и данные, как и много веков назад, по-прежнему являются физическим (материальным) носителем некоторых сигналов, которые становятся информацией только в момент осмысления их человеком.

Самым удобным представлением данных в компьютере оказалось представление с помощью двоичной элементной базы в виде дискретных сигналов с двумя состояниями. Это связано с простотой распознавания и хранения на физическом уровне сигналов, имеющих только два состояния: «включен — выключен», «да — нет», «есть ток (потенциал) — нет тока (потенциала)».

Для хранения данных в компьютере используется *ячейка памяти*. Каждая ячейка памяти представляет собой физическую систему, состоящую из однородных элементов — *битовых разрядов*, каждый из которых может принимать одно из двух возможных значений: 0 или 1. Представление данных с помощью битовых разрядов получило название *бинарного представления*.

Минимальное количество битовых разрядов в одной ячейке памяти, к которой можно обратиться по адресу, равно 8. Ячейка памяти, состоящая из 8 битовых разрядов, получила название *байта*.

Справка. Исторически, для обозначения двоичного разряда употребляли словосочетание *двоичная цифра* (binary digit).

В 1948 г. американский математик Джон Уайлдер Таки (John Wilder Tukey) предложил заменить это словосочетание более коротким. После недолгого выбора между сокращением «bigit» и «binit», он пришел к слову bit (бит), которое не только хорошо сокращало «binary digit», но и имело значение в английском языке (bit — маленький кусочек, самая малая часть чего-либо). На самом деле бит — это самая малая мера информации и самая малая величина, при помощи которой информацию можно перевести в данные.

В течение одного такта генератора тактовой частоты центральный процессор может обратиться к одному, двум, четырем или восьми байтам информации. Количество байтов информации, одновременно доступных центральному процессору, определяется количеством разрядов шины данных компьютера. Соответственно,

про компьютеры говорят — 8-, 16-, 32- или 64-разрядный. На сегодняшний день среди персональных компьютеров большинство 32-разрядных, и в настоящее время происходит переход от 32-разрядных к 64-разрядным моделям.

Представление сигналов в любом другом, не двоичном, виде потребовало бы реализации ячеек для хранения информации, в которых каждый разряд обладал бы тремя состояниями и более. В этом случае недостаточно было бы распознавать только факт наличия или отсутствия тока (потенциала). Необходимо было бы еще измерять его уровень, что намного удорожало бы технологию производства таких вычислительных машин.

Технологические особенности производства электронных компонентов, делающие двоичную элементную базу более дешевой, чем троичная, на долгое время отодвинули троичные машины на задний план. Возможно, в будущем компьютеры на троичной логике займут главенствующее положение в технологиях, однако сегодня об этом говорить не приходится. Однако можно с уверенностью утверждать, что представление данных в бинарном коде в недрах ЭВМ — не единственный вариант, и бинарность — всего лишь технологическая, а не информационная необходимость.

Таким образом, у всех современных компьютеров данные на физическом уровне представлены в двоичном (бинарном) виде.

Современные компьютеры способны при помощи специального оборудования воспринимать, преобразовывать в данные, хранить и воспроизводить следующие *виды информации*:

- числовую;
- символьную (текст);
- звуковую, или аудио (голос, музыку, шумы);
- графическую (рисунки, фотографии, чертежи, сканированные изображения);
- видео (фильмы, видеозаписи).

Если сравнивать этот перечень с типами сигналов, которые могут воспринимать органы чувств человека, то вкус, запах и все, что называется «кинестетическими ощущениями» (ощущения веса, давления, положения в пространстве, прикосновения, боли), пока не получило соответствующего места в ряду обрабатываемой компьютером информации. Однако сегодня в этом направлении ведутся активные научные и технические разработки, и тот день, когда компьютер сможет хранить, обрабатывать и воспроизводить данные, несущие в себе кинестетическую информацию, не так уж далек.

11.1.2. Числовые данные

Числовые данные в компьютере представляются двумя подмножествами: целыми и действительными числами (числами с плавающей запятой). В отличие от символов, ввод числовых данных не может происходить прямым преобразованием и записью кодов, передаваемых с клавиатуры. С клавиатуры мы передаем последовательность символов. Эта последовательность затем программным образом преобразуется в числа. Для того чтобы полученное число было выведено на экран, оно

снова программным путем преобразуется в последовательность символов. В памяти компьютера число хранится в виде двоичного представления.

Целые числа, в зависимости от типа данных, хранятся в памяти компьютера, занимая один, два, четыре и более байта. Каждый бит в байте — это один двоичный разряд. В случае, когда число должно иметь знак, один из разрядов используется для хранения знака. В табл. 11.1 показаны варианты целочисленных представлений с разным числом байтов.

Таблица 11.1. Представление целых чисел

Количество байтов	Знак	Диапазон
1	Без знака	0 ... 255
1	Со знаком	128 ... 127
2	Без знака	0 ... 65535
2	Со знаком	−32768 ... 32767
4	Без знака	0 ... 4294967295
4	Со знаком	2147483648 ... 2147483647
8	Без знака	0 ... 18446744065119617025
8	Со знаком	−9223372032559808513 ... 9223372032559808512

Как видите, даже 32-разрядного целочисленного представления, как со знаком, так и без, вполне достаточно для решения большинства задач.

Теоретически в памяти компьютера представление отрицательного целого числа может храниться в *прямом*, *обратном* или *дополнительном* коде. Предположим, нам надо представить в двоичном виде целое число 7. Без знака оно поместится в 8-разрядную ячейку следующим образом:

0	0	0	0	0	1	1	1

Если это десятичное число −7 (минус семь), то его прямое представление будет выглядеть так:

1	0	0	0	0	1	1	1

То есть старший бит призван показать, что это число со знаком «минус».

Однако на практике отрицательные числа не хранятся в виде прямого кода, а всегда приводятся к *дополнительному* коду. Этот код получается из прямого следующим образом: все разряды, кроме знакового, инвертируются (то есть вместо единиц подставляются 0, вместо нулей — единицы). Так число превращается в обратный код. Затем к результату прибавляется единица, и код становится дополнительным. Вот так будет выглядеть дополнительный код для числа −7:

1	1	1	1	1	0	0	1

Дополнительный код используется для уменьшения количества циклов, за которые процессор выполняет арифметические операции над этими числами, поскольку позволяет заменить операцию вычитания операцией сложения.

Действительные числа состоят из двух частей: мантиссы и порядка. Мантисса — это целое число, содержащее только значащие цифры исходного числа, а порядок — это позиция запятой, разделяющей целую и дробную части числа. Например, если в памяти надо сохранить число 00313,560, то записаны будут два целых числа: 31356 и 3.

При хранении действительных чисел возникает проблема: некоторые действия над числами приводят к потере точности.

Пример. При делении 1 на 3 результатом является бесконечная иррациональная дробь: 0,3333... Поскольку место в памяти, отведенное под запись этого числа, не бесконечно, то в определенном месте повторение троек придется остановить и записать в память округленное число. Это, в свою очередь, значит, что при умножении такой дроби на три уже не получится результат, равный единице, то есть происходит потеря точности.

11.1.3. Символьные данные

Символьные данные, которые несут в себе текстовую информацию, попадают в компьютер наиболее простым из всех известных способов: после нажатия клавиши на клавиатуре генерируется один байт данных, который записывается в память компьютера. Каждой клавише компьютера соответствует одно определенное сочетание битов в байте. Поскольку битов в байте 8, то общее количество сочетаний, которое может быть получено в результате различных комбинаций нулей и единиц внутри байта, равно 256 (2^8). Например, 10 нажатий клавиш создает в памяти компьютера последовательность из 10 заполненных байтов. Таков процесс поступления в компьютер и хранения в компьютере простого текста.

Любая из комбинаций битов в байте может рассматриваться как число, записанное в двоичном коде. Соответствие числа, которое при нажатии клавиши записывается в память компьютера, и символа, который появляется на экране при воспроизведении данного числа в символьном виде, называется *кодовой таблицей*. По мере развития компьютерной техники потребность во все более полном отображении символьной (текстовой) информации возрастала. Вначале казалось, что для кодирования символов при помощи чисел с избытком хватит 127 байт. Первое по порядку 31 десятичное число было отдано под служебные символы (перевод строки, перевод каретки, звонок, пустой символ) (табл. 11.2). Следующие по порядку числа (от 32 до 127) кодировали символы алфавита (табл. 11.3).

Этой таблицы символов вполне хватило для ввода символьной информации на английском языке, и она стала стандартом ASCII (American Standard Code for Information Interchange — американский стандартный код для обмена информацией).

Таблица 11.2. Коды управляющих символов (0–31)

Код	Обозначение	Клавиша	Значение	Отображаемый символ
0	nul	^@	Нуль	
1	Soh	^A	Начало заголовка	⓪
2	Stx	^B	Начало текста	●
3	Etх	^C	Конец текста	♥
4	End	^D	Конец передачи	♦
5	Enq	^E	Запрос	♣
6	Ack	^F	Подтверждение	♠
7	Bel	^G	Сигнал (звонок)	•
8	Bs	^H	Забой (шаг назад)	▣
9	It	^I	Горизонтальная табуляция	⓪
10	Lf	^J	Перевод строки	■
11	Vt	^K	Вертикальная табуляция	⋈
12	Ff	^L	Новая страница	♀
13	Cr	^M	Возврат каретки	↵
14	So	^N	Выключить сдвиг	♩
15	Si	^O	Включить сдвиг	⊖
16	Dle	^P	Ключ связи данных	▶
17	dc1	^Q	Управление устройством 1	◀
18	dc2	^R	Управление устройством 2	↑
19	dc3	^S	Управление устройством 3	!!
20	dc4	^T	Управление устройством 4	†
21	nak	^U	Отрицательное подтверждение	§
22	syn	^V	Синхронизация	—
23	etb	^W	Конец передаваемого блока	‡
24	can	^X	Отказ	†
25	em	^Y	Конец среды	↓
26	sub	^Z	Замена	→
27	esc	^[	Ключ	←
28	fs	^\ ^_	Разделитель файлов	└
29	gs	^}	Разделитель группы	••
30	rs	^^	Разделитель записей	▲
31	us	^_	Разделитель модулей	▼

Таблица 11.3. Символы с кодами (0–31)

Код	Символ	Код	Символ	Код	Символ	Код	Символ
32	пробел	56	8	80	P	104	П
33	!	57	9	81	Q	105	Q
34	"	58	:	82	R	106	R
35	#	59	;	83	S	107	S
36	\$	60	<	84	T	108	T
37	%	61	=	85	U	109	u
38	&	62	>	86	V	110	v
39	'	63	?	87	W	111	w
40	(	64	@	88	X	112	x
41	)	65	A	89	Y	113	y
42	*	66	B	90	Z	114	z
43	-	67	C	91	[	115	s
44	,	68	D	92	\	116	t
45	.	69	E	93	]	117	u
46		70	F	94	^	118	v
47	/	71	G	95		119	w
48	0	72	H	96	`	120	x
49	1	73	I	97	A	121	y
50	2	74	J	98	b	122	z
51	3	75	K	99	c	123	{
52	4	76	L	100	d	124	
53	5	77	M	101	e	125	}
54	6	78	N	102	f	126	~
55	7	79	O	103	g	127	del

Для кодирования символов, отличных от символов английского алфавита, была использована вторая часть из 256 комбинаций, числа от 128 до 256. Эта часть получила название *расширенной таблицы ASCII* и может содержать в себе символы того или иного национального алфавита. При этом если первая часть таблицы ASCII сформирована один раз и не меняется, то вторая ее часть изменяется в зависимости от того, в какой операционной системе вы работаете. Буквы алфавита остаются теми же самыми, а связанные с ними числа меняются. Так, только для русского языка появились кодировки CP-866 (MS-DOS), CP-1251 (Windows), KOI8-R (UNIX и Linux). Именно наличие нескольких кодовых таблиц для одного и того же языка является причиной появления на интернет-страницах потоков странных иероглифов вместо обычного текста: программа просмотра Интернета не всегда способна определить, какую кодовую таблицу выбрать.

Даже если отвлечься от проблем с путаницей между разными кодовыми таблицами одного и того же языка, кодирование многоязыковых текстов наталкивается на еще одно препятствие: что делать, если в одном тексте встречаются фрагменты, написанные на разных языках? Что делать, если текст необходимо совместить с математическими или химическими формулами? Для решения этой проблемы способ кодирования символов еще раз изменили — один символ стали кодировать не одним, а двумя байтами данных. Так появился способ кодирования UNICODE. Теперь в одном байте можно было хранить код символа, а во втором указывать, из какой кодовой таблицы этот символ извлекать.

Таким образом, текстовая информация в компьютере может быть представлена как однобайтной последовательностью, когда каждый символ кодируется одним байтом в памяти компьютера, так и двухбайтной, требующей вдвое больше памяти для хранения, но зато более универсальной и гибкой.

11.1.4. Мультимедийные данные

Графика

Графические изображения, как и любые другие данные, представлены в компьютере в виде двоичных чисел. Однако в зависимости от природы этих изображений (графического формата) числа играют разную роль. Существует два типа графических форматов, векторный и растровый.

Изображение в *растровом формате* в памяти компьютера представлено совокупностью точек, каждая из которых имеет атрибуты (такие как цвет или маска). Если каждая точка вместе с атрибутами кодируется 32-битным кодом (4 байта), то нетрудно посчитать, что рисунок размером 320 на 200 точек потребует для хранения $320 \times 200 \times 4 = 256\,000$ байт. Таким образом, растровый формат хранения изображений является весьма ресурсоемким. В то же время, с учетом того, что каждая точка изображения хранится отдельно, имеется возможность ее отредактировать и скорректировать. Таким образом, там, где важно достоверно воспроизвести или откорректировать сохраненную графическую информацию, например, фотоизображение, растровый формат незаменим.

Изображение в *векторном формате* представляет собой графический объект, составленный из примитивов, а именно: точек, отрезков, прямых, дуг, окружностей, прямоугольников, кривых. Эти примитивы могут быть описаны математическими формулами и выражены числами, обозначающими координаты точек начала и конца отрезка, опорных узлов кривых, радиусом и координатой центра окружности, и т. д. Даже если добавить к этим описаниям несколько чисел, кодирующих цвет, толщину и тип линий, все равно очевидно, что в сравнении с растровым форматом такого рода изображения можно хранить гораздо более компактно. Тот же самый рисунок размером 320 на 200 точек, содержащий на первый взгляд очень сложный чертеж, может в памяти компьютера занимать мало места. Например, изображение машины (рис. 11.2) в векторном формате занимает 41 Кбайт, а в растровом формате TIF — 395 Кбайт. Разница почти в 10 раз.

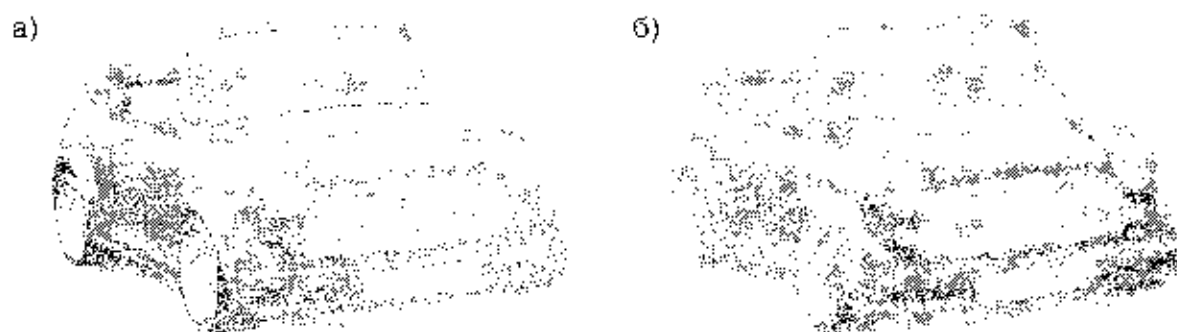


Рис. 11.2. Графическое изображение: а) векторный формат; б) растровый формат

Источником графических данных в векторном формате всегда являются векторные графические редакторы или системы автоматизированного проектирования. В любом случае это информация, создаваемая человеком в процессе рисования (черчения). Источниками же графических данных в растровом формате обычно являются цифровые фотоаппараты или сканеры, которые напрямую воспринимают сигналы внешнего мира и переводят их в форму цифровых данных (цифровая фотография или сканированное изображение).

Звуковые данные

Различные звуки и музыка, чтобы превратиться в данные компьютера, должны пройти ряд преобразований. Вначале некоторое устройство (обычно микрофон) должно воспринять звуковой сигнал и преобразовать его в сигнал электрический. Затем непрерывный (аналоговый) электрический сигнал при помощи другого устройства должен быть преобразован в двоичные числа (в цифровой вид), которые сохраняются в памяти компьютера. Устройство, преобразовывающее аналоговый электрический сигнал в цифровую форму называется АЦП (аналого-цифровой преобразователь). Этапы преобразования сигнала иллюстрирует рис. 11.3.

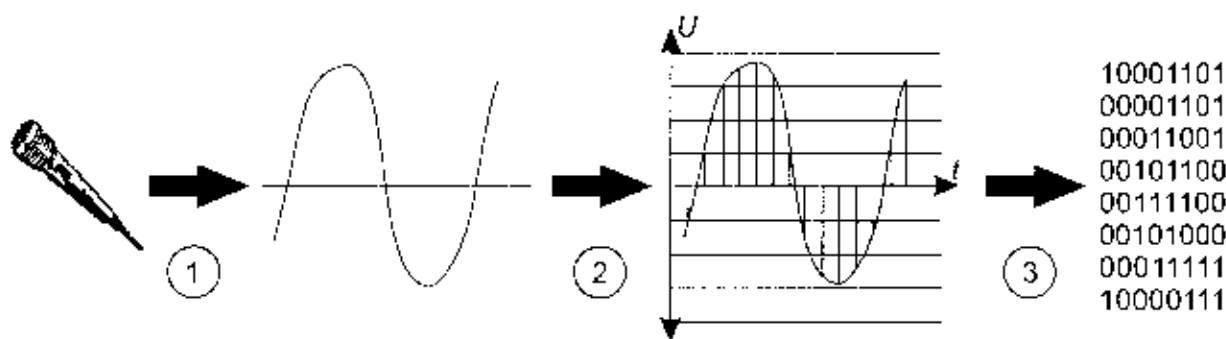


Рис. 11.3. Преобразование звукового сигнала в двоичные числа

На этапе, обозначенном цифрой 1, происходит преобразование звукового сигнала в аналоговый электрический сигнал, на этапе 2 электрический сигнал поступает на вход АЦП, где происходит его дискретизация. Это значит, что с заданной частотой измеряется уровень электрического сигнала. На этапе 3 полученные значения уровней записываются в память компьютера в двоичном виде. Чем на большее количество уровней разбивается весь диапазон аналогового сигнала

и с чем большей частотой эти уровни замеряются, тем точнее звуковая информация преобразуется в цифровой вид и с тем большей достоверностью она потом будет восстановлена.

С момента, когда звуковой сигнал приводится к цифровому виду, он уже необратимо искажается, однако звуковая информация не теряется. Если частота дискретизации выбрана высокой (в несколько раз выше, чем 22 000 Гц — частота, которую может различить человеческое ухо), а динамический диапазон (то есть количество уровней в каждой точке) достаточно велик, то при восстановлении невозможно будет различить аналоговый сигнал, записанный на аналоговый носитель (виниловый диск), и данные, сохраненные на цифровой носитель (компакт-диск).

Чем выше качество сохраняемого в компьютере звука, тем больше для него необходимо места в памяти. Для получения сигнала с гарантированно высоким качеством нужно задать достаточно большое количество уровней и высокую частоту дискретизации.

Пример. Если мы разобьем входной сигнал по амплитуде на 65 536 уровней (то есть для хранения каждого измерения будем использовать 2 байта) и измерим его с частотой 110 000 раз в секунду, то получим гарантированный запас по качеству. Несложный подсчет покажет, что для хранения музыкального фрагмента длительностью 3 минуты, оцифрованного таким образом, требуется $2 \times 110\,000 \times 60 \times 3 = 39\,600\,000$ байт, или 38,7 Мбайт компьютерной памяти. Примерно такой размер имеют музыкальные фрагменты, записанные на современные компакт-диски.

Для хранения в памяти персонального компьютера, а в особенности для передачи звуковых файлов через Интернет, объемы их слишком велики. Для уменьшения объема данных в несколько раз применяются специальные процедуры сжатия. Из исходного оцифрованного звука после сжатия получаются данные в 8–10 раз меньшего объема. Одним из популярных форматов хранения звуковой информации, получаемой при помощи таких специальных процедур, является MP3. Надо заметить, что при сжатии данных в формат MP3 часть звуковой информации необратимо теряется, поэтому он называется форматом сжатия звука с потерями.

В том случае, когда данные, несущие звуковую информацию, требуется преобразовать в звуковой сигнал, записанную в памяти последовательность байтов подают на вход цифроаналогового преобразователя (ЦАП). Это устройство преобразует числа сначала в уровни электрического сигнала, затем сглаживает эти уровни, преобразуя их в аналоговый электрический сигнал, а потом аналоговый сигнал подается на воспроизводящее устройство (аудиоколонки или головные телефоны).

Видеоданные

Видеоданные представляют собой практически те же данные в растровом графическом формате, и получаются они примерно таким же образом, как и цифровая фотография. Многие современные цифровые фотоаппараты позволяют производить видеосъемку, а цифровые видеокамеры — делать фотографии. Механизм один и тот же, но в случае, когда происходит видеосъемка, фотографии делаются с боль-

ной скоростью одна за другой и последовательно записываются в цифровом виде в память видеокамеры. Эта последовательность фотографий образует видеоряд, аналогичный кадрам на кинолентке.

11.1.5. Служебные данные

Когда мы производим сбор первичных данных и перевод их в цифровую форму, мы осуществляем кодирование сигнала выбранным способом и последовательно записываем полученные коды в память компьютера. Это верно как для символьной информации, так и для растровых графических данных при цифровой фото- или видеосъемке, при кодировании аудиосигнала процесс происходит примерно так же. Однако следующим шагом, который требуется, чтобы хранить и передавать данные, является их оформление. Например, введенный нами необработанный текст преобразуется текстовым процессором в документ. В документе, кроме самого текста, присутствует множество объектов, имеющих нетекстовую природу: рамки, разделительные линии, блоки, признаки разбиения на страницы и форматирования, другие служебные конструкции. Аналогично для хранения графических данных изображение определенным образом разбивается на слои, каналы, градации яркости, над данными производятся операции сжатия для уменьшения хранимого и передаваемого объема. Таким образом, к данным, хранящим собственно символьную, аудио- или видеoinформацию, добавляются специальные *служебные данные*, данные структурируются, и для хранения в постоянной памяти они приводятся к типу файлов определенного формата.

Любой *файл*, в общем случае, является последовательностью байтов, которая записывается на физический носитель и получает имя (или некоторый идентификатор). Как известно, файлы имеют разный формат. Слово «формат» в данном случае говорит о том, что разные группы байтов, записанные внутри файла, имеют разное назначение. Часть байтов несет в себе непосредственно данные, а часть — информацию о том, как эти данные правильно восстановить и отобразить на экране компьютера или воспроизвести другим необходимым образом. Таким образом, любой файл представляет собой неоднородную структуру данных, и тип файла непосредственно связан, с той последовательностью действий, которую нужно проделать, чтобы данные из файла превратились в сигналы, несущие информацию.

11.2. Системы счисления

11.2.1. Определение и классификация

Числа отображаются с помощью системы счисления.

Система счисления — это способ отображения чисел с помощью символов.

Совокупность символов, при помощи которых записывается система счисления, называется *алфавитом* системы счисления. Количество символов, составляющих алфавит, называется его *размерностью*.

Системы счисления делятся на позиционные и непозиционные. В *позиционной системе счисления* значение каждой цифры в записи числа зависит от ее позиции (разряда). В *непозиционной системе счисления* для обозначения чисел вводятся специальные знаки, количественное значение которых всегда одинаково и не зависит от их места в записи числа, например, числу 25 в непозиционной римской системе счисления соответствует запись XXV, а числу 53 соответствует запись LIII.

Последовательность чисел, каждое из которых задает «вес» соответствующего разряда, называется *базисом* позиционной системы счисления. Основным достоинством позиционной системы счисления является возможность записи произвольного числа при помощи ограниченного набора символов.

В табл. 11.4 сравниваются несколько двоичных и десятичных чисел, находящихся в одной и той же позиции. Обе эти системы счисления являются позиционными. Значение каждой цифры в числе легко определить, воспользовавшись формулой

$$V = B^N.$$

Здесь V — значение, B — основание системы счисления, N — порядковый номер позиции.

Таблица 11.4. Двоичные и десятичные числа

Двоичные			Десятичные	
Двоичное число (двоичное представление)	Двоичное число (десятичное представление)	Степень	Десятичное число (десятичное представление)	Степень
1	1	2^0	1	10^0
10	2	2^1	10	10^1
100	4	2^2	100	10^2
1000	8	2^3	1000	10^3
10000	16	2^4	10000	10^4
100000	32	2^5	100000	10^5
1000000	64	2^6	1000000	10^6

Очевидно, что десятичная запись намного удобнее двоичной, поскольку более компактна. Так, в 6-й позиции в десятичной системе закодирован миллион, а в двоичной — только шестьдесят четыре. В двоичной системе представление чисел получается весьма громоздким, но оно идеально для компьютера, так как не требует сложных инженерных и технологических решений.

Позиционную систему счисления называют *традиционной*, если ее базис образуют члены геометрической прогрессии, а значения цифр есть целые неотрицательные числа. Знаменатель P геометрической прогрессии, члены которой образуют базис традиционной системы счисления, называется *основанием* этой системы счисления. Традиционные системы счисления с основанием P иначе называют *P-ичными*.

Примерами P -ичной системы счисления могут служить двоичная, восьмеричная, десятичная, шестнадцатеричная системы счисления.

Для того чтобы показать, в какой системе счисления записано число, после числа указывают основание в качестве нижнего индекса. Например 10_2 — два в двоичной системе счисления, или 2_{10} — два в десятичной системе счисления.

Если в позиционной системе счисления базис не является геометрической прогрессией или значения цифр могут принимать отрицательные значения, такую систему счисления называют *нетрадиционной* (рис. 11.4).

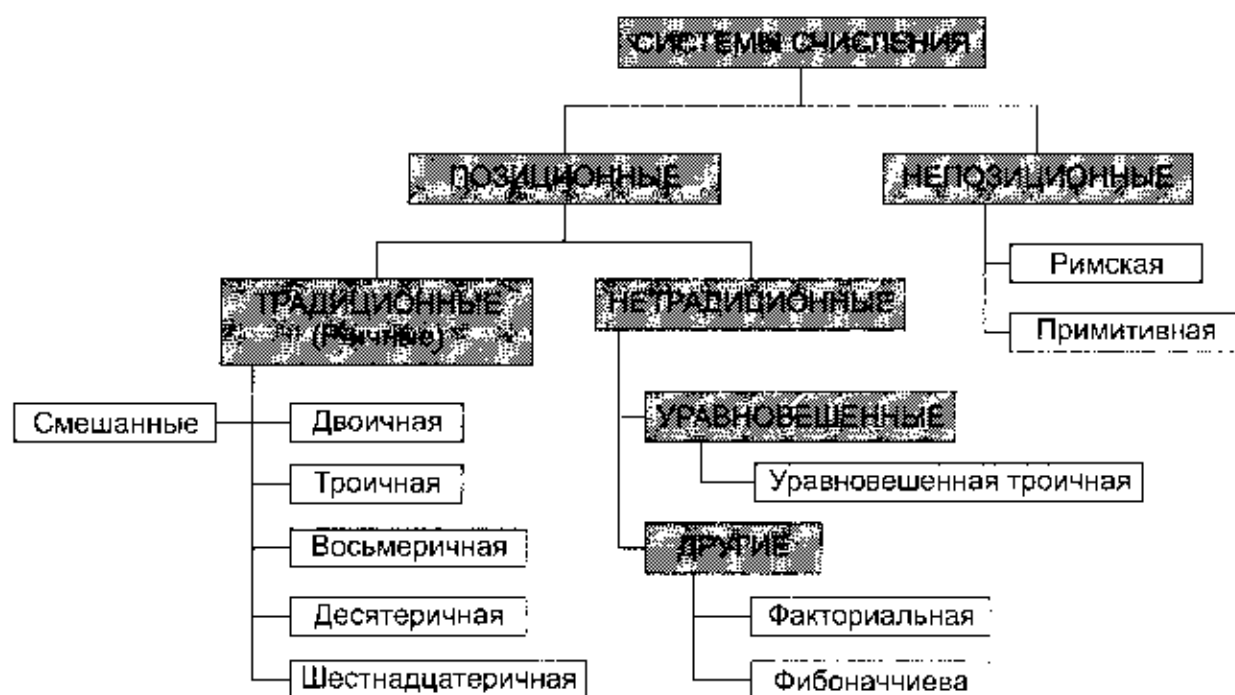


Рис. 11.4. Системы счисления

Любое натуральное число можно записать в P -ичной системе счисления единственным образом. В более строгом виде это утверждение можно сформулировать в следующем виде.

Пусть P — произвольное натуральное число, большее единицы. Существует единственное представление любого натурального числа X в виде степенного ряда:

$$X = a_n P^n + a_{n-1} P^{n-1} + \dots + a_1 P + a_0, \text{ где } 0 \leq a_i < P, \quad 0 \leq i \leq n.$$

В P -ичной системе счисления любое неотрицательное вещественное число, содержащее целую и дробную части, можно записать в виде

$$\begin{aligned} a &= a_n P^n + a_{n-1} P^{n-1} + \dots + a_1 P + a_0 + a_{-1} P^{-1} + a_{-2} P^{-2} + \dots = \\ &= \sum a_i P^i, \quad 0 \leq a_i < P, \quad n \geq 0. \end{aligned}$$

Здесь $P > 0$ — основание позиционной системы счисления, a_i — цифры числа a в P -ичной системе счисления.

Для записи чисел в P -ичных системах счисления используют десятичные цифры (0 ... 9), которые дополняются буквами латинского алфавита.

В табл. 11.5 приведен пример того, как будет выглядеть число 25, записанное в разных P -ичных системах счисления.

Таблица 11.5. Число 25, записанное в разных P -ичных системах счисления

Система счисления	Основание	Размерность алфавита	Алфавит	Пример
Десятичная	10	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	$25 = 2 \times 10^1 + 5 \times 10^0$
Двоичная	2	2	0, 1	$11001_2 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$
Троичная	3	3	0, 1, 2	$221_3 = 2 \times 3^2 + 2 \times 3^1 + 1 \times 3^0$
Восьмеричная	8	8	0, 1, 2, 3, 4, 5, 6, 7	$31_8 = 3 \times 8^1 + 1 \times 8^0$
Шестнадцатеричная	16	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F	$19_{16} = 1 \times 16^1 + 9 \times 16^0$

Для записи отрицательных чисел в P -ичных системах счисления используют знак «минус», помещенный перед числом. При записи вещественных чисел для отделения дробной части числа от целой ставится запятая. Таким образом на запись P -ичных чисел распространяются привычные нам по десятичной системе счисления правила, например: $-2,2_{10}$, $-F,0E_{16}$, $-101,001_2$.

11.2.2. Перевод из одной P -ичной системы счисления в другую

Перевод P -ичного целого в десятичное число

Алгоритм перевода:

1. Каждая цифра P -ичного числа переводится в десятичную систему.
2. Полученные числа нумеруются справа налево, нумерация начинается с нуля.
3. Десятичное число, соответствующее каждой P -ичной цифре, умножается на P^k , где основание P записывается в десятичной системе; k — номер этого числа, полученный на шаге 2. Результаты складываются, все арифметические действия производятся в десятичной системе.

Пример. Перевод числа 1110011_2 из двоичной в десятичную систему счисления.

1. Каждая цифра числа уже находится в десятичной системе.
2. Прономеруем каждую цифру справа налево, при этом номера цифр будем указывать в квадратных скобках справа от цифры:

$$1[6], 1[5], 1[4], 0[3], 0[2], 1[1], 1[0].$$

3. Результат будет выглядеть как сумма ($P = 2$):

$$\begin{aligned} a &= 2^6 \times 1 + 2^5 \times 1 + 2^4 \times 1 + 2^3 \times 0 + 2^2 \times 0 + 2^1 \times 1 + 2^0 \times 1 = \\ &= 64 + 32 + 16 + 0 + 0 + 2 + 1 = 114_{10}. \end{aligned}$$

Пример. Перевод числа $EF09_{16}$ из шестнадцатеричной в десятичную систему счисления.

1. Каждую цифру числа переведем в десятичную систему:

$$E_{16} = 14_{10}, F_{16} = 15_{10}, 0_{16} = 0_{10}, 9_{16} = 9_{10}.$$

2. Пронумеруем каждую цифру числа справа налево:

$$14[3], 15[2], 0[1], 9[0].$$

3. Результат ($P = 16$):

$$\begin{aligned} a &= 16^3 \times 14_{10} + 16^2 \times 15_{10} + 16^1 \times 0_{10} + 16^0 \times 9_{10} = \\ &= 4096 \times 14 + 256 \times 15 + 0 + 9 = 61193_{10}. \end{aligned}$$

Перевод P -ичного смешанного числа в десятичное число

Алгоритм перевода:

1. Отдельно переводится целая часть числа (как описано ранее).
2. Каждая цифра дробной части P -ичного числа переводится в десятичную систему.
3. Полученные числа нумеруются слева направо, нумерация начинается с единицы.
4. Десятичное число, соответствующее каждой P -ичной цифре, умножается на P^{-k} , где k — номер, присвоенный числу; основание P записывается в десятичной системе. Результаты складываются, все действия производятся в десятичной системе.

Пример. Перевод смешанного числа $11,1101_2$ из двоичной в десятичную систему счисления.

1. Переведем в десятичную систему целую часть: $11_2 = 3_{10}$.
2. В двоичной системе каждая цифра соответствует десятичной.
3. В дробной части пронумеруем цифры слева направо:

$$1[1], 1[2], 0[3], 1[4].$$

4. Дробная часть ($p = 2$):

$$2^{-1} \times 1 + 2^{-2} \times 1 + 2^{-3} \times 0 + 2^{-4} \times 1 = 0,8125.$$

Все число в десятичной системе счисления будет записано как $3,8125_{10}$.

Пример. Перевод смешанного числа $B,1CD_{16}$ из шестнадцатеричной в десятичную систему.

1. Переведем в десятичную систему целую часть:

$$B_{16} = 11_{10}.$$

2. Переведем каждую цифру дробной части в десятичную систему:

$$1_{15} = 1_{10}, C_{16} = 12_{10}, D_{16} = 13_{10}.$$

3. В дробной части пронумеруем цифры слева направо:

$$1[1], 12[2], 13[3].$$

4. Дробная часть ($P = 16$):

$$16^{-1} \times 1 + 16^{-2} \times 12 + 16^{-3} \times 13 = 0,0625 + 0,046875 + 0,003173828125 = \\ = 0,112548828125_{10}.$$

Все число в десятичной системе счисления будет записано как

$$11,112548828125.$$

Перевод десятичного целого числа в P -ичную систему

Алгоритм перевода:

1. Делим исходное число нацело на основание P в десятичной системе счисления и записываем целую часть результата деления в качестве нового значения десятичного числа.
2. Остаток от деления заменяем соответствующей цифрой в P -ичной системе счисления и приписываем ее слева к полученным ранее цифрам P -ичной записи числа (первая полученная цифра соответствует младшему разряду).
3. Выполняем шаги 1 и 2 до тех пор, пока не получим в результате 0.

Пример. Перевод числа 17_{10} в двоичную систему.

$$17 : 2 = 8 \text{ (остаток 1).}$$

$$8 : 2 = 4 \text{ (остаток 0).}$$

$$4 : 2 = 2 \text{ (остаток 0).}$$

$$2 : 2 = 1 \text{ (остаток 0).}$$

$$1 : 2 = 0 \text{ (остаток 1).}$$

Берем остатки от деления, указанные в скобках, начиная снизу, и, продвигаясь вверх, записываем их слева направо. Это и есть искомый результат, число 17 записанное в двоичной системе счисления:

$$17_{10} = 10001_2.$$

Пример. Перевод десятичного числа 3477_{10} в шестнадцатеричную систему счисления.

$$3477 : 16 = 217 \text{ (остаток } 5_{10} = 5_{16}).$$

$$217 : 16 = 13 \text{ (остаток } 9_{10} = 9_{16}).$$

$$13 : 16 = 0 \text{ (остаток } 13_{10} = D_{16}).$$

$$\text{Результат } 3477_{10} = D95_{16}.$$

Правильность результата можно проверить, выполнив обратное преобразование.

Перевод десятичной дроби в P -ичную систему

Перевод дробной части десятичного числа в P -ичную систему производится отдельно от целой части.

Алгоритм перевода:

1. Исходное число умножается на основание системы счисления P . В полученном произведении целая часть является первой цифрой после запятой в результате.
2. Если после этой операции дробная часть числа не стала равной нулю, умножаем ее на P и целую часть полученного числа (переведа ее в соответствующую P -ичную систему) приписываем справа к результату.
3. Выполняем шаг 2 до тех пор, пока дробная часть не станет равной нулю или пока не выделится период (повторяющаяся часть в результате).

Пример. Перевод числа $0,125_{10}$ в двоичную систему.

$$0,125 \times 2 = 0,250 \text{ (целая часть} = 0\text{)}.$$

$$0,25 \times 2 = 0,5 \text{ (целая часть} = 0\text{)}.$$

$$0,5 \times 2 = 1,0 \text{ (целая часть} = 1\text{)}.$$

Поскольку в этом месте дробная часть стала равной нулю, преобразование окончено. Результат $0,125_{10} = 0,001_2$.

Пример. Перевод числа $0,275_{10}$ в шестнадцатеричную систему счисления.

$$0,275 \times 16 = 4,4 \text{ (целая часть} = 4\text{)}.$$

$$0,4 \times 16 = 6,4 \text{ (целая часть} = 6\text{)}.$$

$$0,4 \times 16 = 6,4 \text{ (целая часть} = 6\text{)}.$$

Перевод закончился с выделением периода $0,275_{10} = 0,4(6)_{16}$.

11.2.3. Выполнение арифметических операций

Сложение

Если результат сложения двух цифр в P -ичной системе счисления больше $P - 1$ (то есть полученное число двузначное), то старшая цифра результата всегда равна 1. Таким образом, при сложении столбиком в следующий разряд может переходить только единица, а результат сложения в любом разряде будет меньше, чем P . Результат сложения двух положительных P -ичных чисел либо имеет столько же значащих цифр, что и максимальное из двух слагаемых, либо на одну цифру больше, но этой цифрой может быть только единица. Такая закономерность обуславливает возможность сложения в столбик не только десятичных, но и любых P -ичных чисел.

Поскольку сложение в десятичной системе для нас является привычным (как и другие арифметические действия), оно не вызывает затруднений. В других P -ичных системах для облегчения сложения (вычитания) применяются специальные таблицы, сходные с привычной нам таблицей умножения (табл. 11.6).

Пример. Сложение двух двоичных чисел: $110,112_2$ и $10,012_2$.

$$\begin{array}{r}
 + 110,11_2 \\
 10,01_2 \\
 \hline
 1001,00_2
 \end{array}$$

Таблица 11.6. Таблицы сложения в двоичной и троичной системах счисления

a \ b	0	1
0	0	1
1	1	10

a \ b	0	1	2
0	0	1	2
1	1	2	10
2	2	10	11

Вычитание

Алгоритм:

Если необходимо вычесть из цифры a цифру b и $a \geq b$, то в столбце b таблицы сложения ищем значение числа a . Самая левая цифра в строке, в которой найдено значение числа a , и будет результатом вычитания. Если же $a < b$, нам нужно заимствовать единицу из левого разряда, поэтому в столбце b мы ищем число $1a$, и левая цифра в соответствующей строке будет результатом вычитания.

Пример. Вычтем из двоичного числа $110,11_2$ двоичное число $10,01_2$.

$$\begin{array}{r}
 110,11_2 \\
 - 10,01_2 \\
 \hline
 100,10_2
 \end{array}$$

Пример. Вычитание чисел в троичной системе счисления: вычитаем из 2 $20,02_3$ число $100,10_3$

$$\begin{array}{r}
 220,02_3 \\
 - 101,10_3 \\
 \hline
 111,22_3
 \end{array}$$

В этом примере мы двигаемся позиция за позицией справа налево.

В случае вычитания 0 из 2 получаем 2

Ситуация, когда $a < b$ возникает во второй справа позиции, нам надо вычесть 1 из 0. В столбце b (в столбце с цифрой 1) мы ищем число $1a$ (поскольку $a=0$, это будет число 10), и в самой левой ячейке строки, в которой расположена это число (10) мы находим искомый ответ 2.

Поскольку в предыдущем шаге мы заимствовали единицу из старшего разряда (а он содержал в себе 0), прежде чем делать следующий шаг, мы должны еще раз выполнить операцию вычитания 1 из 0, получить в старшем разряде 2 (снова заимствовав единицу из старшего разряда), из 2 вычесть 1, и получить в третьей справа позиции 1.

Дальнейшие шаги представляются очевидными.

Умножение

Для того чтобы выполнять умножение чисел в P -ичной системе, нам необходимы как таблица сложения, так и таблица умножения для этой системы (табл. 11.7).

Таблица 11.7. Таблицы умножения в двоичной и троичной системах счисления

a \ b	0	1
0	0	0
1	0	1

a \ b	0	1	2
0	0	0	0
1	0	1	2
2	0	2	11

Умножение производится по правилам умножения столбиком с использованием соответствующих таблиц умножения и сложения.

Пример. Умножение двоичного числа $10,1_2$ на двоичное число $1,1_2$.

$$\begin{array}{r}
 \times \quad 10,1_2 \\
 \quad 1,1_2 \\
 \hline
 \quad 1,01_2 \\
 - \quad 10,1_2 \\
 \hline
 11,11_2
 \end{array}$$

Деление

Как и для умножения, для деления нужны обе таблицы, умножения и сложения в соответствующей P -ичной системе. Само деление выполняется по привычным правилам деления уголком с последовательным вычитанием сомножителей.

Пример. Деление двоичного числа 1111_2 на двоичное число 11_2 .

$$\begin{array}{r|l}
 \begin{array}{r}
 1111_2 \\
 - 11_2 \\
 \hline
 11_2 \\
 - 11_2 \\
 \hline
 0
 \end{array}
 & \begin{array}{l}
 11_2 \\
 \hline
 101_2
 \end{array}
 \end{array}$$

Вопросы для самопроверки

1. Являются ли понятия «информация» и «данные» синонимами? Дайте определение тому и другому понятию.
2. Какие типы информации вам известны?
3. Что такое « P -ичная система счисления»? Какие P -ичные системы счисления вы знаете?
4. Для чего нужен дополнительный код?

5. Верно ли высказывание: «информация в компьютере всегда представлена в бинарном виде»?
6. В каком виде хранятся в компьютере отрицательные целые числа?
7. В каком виде хранятся в компьютере действительные числа?
8. Какие проблемы могут возникнуть при сохранении результатов некоторых арифметических операций?
9. Как решается проблема наличия разных алфавитов в разных языках при кодировании и хранении символьной информации?
10. Что такое «ASCII»?
11. Что такое «необработанные данные»?
12. Зачем нужны служебные данные?

Литература

1. *Андреева Е. В., Босова Л. Л., Фалина Н. Н.* Математические основы информатики. Учебное пособие. М.: Бином, 2005.
2. *Бауэр Ф. Л., Голз Г.* Информатика. М.: Мир, 1990.
3. *Воройский Ф. С.* Информатика. Новый систематизированный толковый словарь-справочник (Введение в современные информационные и телекоммуникационные технологии в терминах и фактах). М.: ФИЗМАТЛИТ, 2003.
4. *Петзольд Ч.* КОД. Русская редакция: М., 2001.

Глава 12 История, состояние и тенденции развития компьютеров и вычислительных систем

12.1. История развития компьютеров

12.2. Классификация компьютеров и вычислительных систем

12.3. Настоящее и будущее компьютеров

Изучая информатику, невозможно обойти вниманием главный современный инструмент, при помощи которого производится обработка информации, — компьютер. В этой главе излагается краткая история развития компьютерной техники, приводятся разные способы классификации современных компьютеров, обсуждаются перспективы развития компьютерных технологий, а также преимущества и проблемы, которые несет активное проникновение компьютеров практически во все области человеческой жизни.

12.1. История развития компьютеров

Разные подходы к описанию истории развития компьютеров побуждают исследователей начинать отсчет с разных моментов времени. Один из подходов связывает развитие компьютеров с этапами развития методов вычислений и обработки

информации. Целесообразно было бы выделить следующие типы вычислительных устройств:

- механические устройства: вспомогательные механические устройства для вычислений; механические счетные машины, не имеющие программы; механические программируемые устройства;
- электронные программируемые устройства.

12.1.1. Механические устройства для вычислений

Вспомогательные механические устройства для вычислений

Абак. Глиняная или деревянная доска с углублениями, в которые помещались камушки или глиняные шарики. Прародитель современных счетов. Использовался для помощи в вычислениях в Вавилоне (приблизительно 2000 лет до н. э.), Египте (приблизительно 1500 лет до н. э.), Древней Греции (приблизительно V век до н. э.). Пройдя через три цивилизации, практически не изменил своей конструкции.

Римский абак. Приблизительно III век до н. э. Римская нотация была тяжела для вычислений, а вычислять приходилось много (империя собирала налоги и тонула в роскоши, развитие торговли вынуждало считать все больше и быстрее), поэтому в Древнем Риме абак применялся особенно широко. Конструкция абака развивалась, появились его наладонные версии (первые «паладонники»).

Китайский абак. Приблизительно IV век до н. э. Представлял собой прямоугольный деревянный каркас с натянутыми поперечными шнурами, на которые были нанизаны косточки. Это уже прямой прототип современных счетов.

Европейский абак. Приблизительно XV век. Постепенно римский абак в Европе приобрел вид, напоминающий сильно усложненные современные счета. В прямоугольном каркасе были протянуты нити, на которые нанизывалось разное количество косточек. Каркас был разделен на несколько частей, в каждой из которых хранились свои результаты вычислений. Устройство получило название линейной доски.

Русские счеты. Первое упоминание в 1658 г. Европейский абак, попав в Россию, был быстро упрощен до вида, в котором мы его сегодня знаем как счеты. Счеты, между прочим, широко использовались в нашей стране еще во второй половине прошлого века, вплоть до появления электронных калькуляторов.

Все перечисленные устройства даже при большом желании трудно назвать вычислительными машинами. Это скорее внешние механические запоминающие устройства для хранения промежуточных результатов вычислений.

Механические непрограммируемые счетные машины

Антикирский механизм. Древняя Греция, приблизительно I век до н. э. Устройство, состоящее из множества связанных зубчатых колес и диферבלата со стрелками. Позволяло рассчитывать движение небесных тел.

Счетная машина Леонардо Да Винчи. 1492 г. Машина была выполнена в виде эскиза в дневнике гениального итальянца. Она была реализована в XX в. и выполняла сложение двух 13-разрядных чисел.

С этого времени и до 1801 г. развитие счетных машин шло по двум основным направлениям:

- *механические счетные машины* на зубчатых колесах или ступенчатых валиках развивались вплоть до появления *арифмометров* (активно использовались в нашей стране еще во второй половине XX в.);
- *аналоговые механические вычислительные устройства*, основанные на таблицах логарифмов, являли собой *логарифмические линейки* разного типа (также активно применялись вплоть до появления электронного калькулятора).

Механические программируемые устройства

В 1801 г. произошло знаменательное событие: Жозеф Мари Жаккар изобрел *ткацкий станок с программным управлением*. Этот станок мог менять рисунок на ткани в соответствии с программой, вводимой с перфокарт. Впервые появилось программируемое, хотя пока еще механическое устройство. С этого момента научная мысль уже нацелилась на создание устройства, которое мы сегодня называем компьютером. Однако до появления первого «настоящего» компьютера было еще полтора века.

С 1822 по 1828 г. английский ученый Чарльз Бэббидж работал над созданием своей *разностной машины*. Машина была проработана до деталей в чертежах, но не построена из-за недостатка денег. Начиная с 1834 г., Бэббидж работал над идеей *аналитической машины*, которую можно смело назвать первым прообразом современного компьютера. Будучи механической, эта машина тем не менее содержала все основные блоки, из которых состоит современный компьютер: процессор («мельница»), запоминающее устройство («склад») на 1000 пятидесятиразрядных чисел, контроллер передачи данных и перфокарточное устройство ввода-вывода. Эту машину Бэббидж придумывал вместе с Адой Августой Лавлейс (дочерью Джорджа Байрона). Ада Лавлейс проделала огромную работу по популяризации машины Бэббиджа и составила первые программы для нее. В честь Ады Лавлейс назван язык программирования Ада. К сожалению, при жизни Бэббиджа не были созданы ни разностная, ни аналитическая машины. Но после того как великий изобретатель и ученый ушел из жизни, они были построены и доказали свою работоспособность.

В 1876 г. русский математик П. Л. Чебышев создал *суммирующий аппарат с непрерывной передачей десятков*, доработанный позднее для осуществления умножения и деления.

1887 г. был отмечен созданием американским изобретателем Холлеритом *электрического табулятора* для работы с перфокартами.

В 1912 г. появилась *машина для интегрирования простых дифференциальных уравнений* по проекту русского инженера А. Н. Крылова.

12.1.2. Электронные программируемые устройства

В 1936 г. английский математик А. Тьюринг выдвинул концепцию *абстрактной вычислительной машины*, почти одновременно с ним абстрактное представление

вычислительной машины предложил американский математик несколько квадратных, хотя и не существовавшие в реальности, предположений, которые ранее программное выполнение вычислений.

В период с 1936 по 1941 г. немецкий изобретатель Конрад предложил дискретных вычислительных машин: Z1, Z2 и Z3. Машину Z3, созданную на четрах площади, логики, принято считать *первым программируемым компьютером* в сантиметрах. с этого момента обычно начинают вести отсчет истории развития не только для как таковых.

Если темпы развития компьютеров вплоть до разработки и создания микроэлектронической машины Бэббиджа диктовались в первую очередь темпами трудившихся ретической научной мысли и представлениями о том, что может, и закон: каждый делать механическое устройство, то, начиная с компьютеров Пузе, то интегральной компьютеров стали определяться в основном технологическим до технологий. Определяющим в конструкции компьютера стала *элементная база*.

Классификация компьютеров по поколениям, приведенная далее, основана именно на названии признаке.

Первое поколение

ENIAC. Элементной базой для первого поколения компьютеров были разумные электронные лампы. Именно это определило их огромный размер и потребляемую мощность при сравнительно скромных вычислительных способностях. Первая ЭВМ, ENIAC (Electronic Numerical Integrator And Computer), создана в 1946 г. в Пенсильванском университете Джоном Моучли и Дж. Уильямом Эккертом. Эта машина весила 30 тонн, занимала 1500 квадратных метров площади, содержала более 18 000 вакуумных ламп и потребляла электрическую мощность 140 кВт. ENIAC была способна совершать 300 умножений или 5000 сложений чисел с плавающей запятой в секунду. Это намного превышало возможности любой другой вычислительной техники. Одним из самых больших недостатков ENIAC было ручное программирование, заключавшееся в переключении тумблеров и кабельных разъемов, что было весьма утомительно. Кроме того, ENIAC была десятичной машиной, выполнявшей все операции в десятичной системе счисления. Эта машина считается *первой электронной вычислительной машиной*, созданной человеком.

EDVAC и машина фон Неймана. Непомерно трудный процесс программирования ENIAC был бы проще, если бы программу можно было представить в том же виде, что и данные, и так же, как данные, хранить в памяти. Математик фон Нейман, консультировавший создателей ENIAC, выдвинул концепцию машины с хранимой программой в 1945 г. Фон Нейман назвал эту машину EDVAC (Electronic Discrete Variable Computer) и сразу же приступил к воплощению своей идеи в институте IAS (Institute of Advanced Studies) в Принстоне. Поэтому обобщенную схему машины фон Неймана еще называют IAS-компьютером. Эта обобщенная схема, изображенная на рис. 12.1, оказалась настолько удачной, что все современные компьютеры в том или ином виде ей соответствуют. Естественно, сама машина фон Неймана является не компьютером, а только концепцией, но зато реализуют эту концепцию все персональные компьютеры с одноядерными процессорами.

С этого времени массовой реализацией компьютеров первого поколения была направлена:

■ *механические* (до 1955 гг. на первый план в компьютерном бизнесе выходили развивались со своими машинами 700-й серии. С этого момента изготовление в нашей стране поставлено на поток.

■ *аналоговые* и логарифмов применялись

Механический

В 1801 г. ткацкий станок в соот-
программи-
учная мысль
компьютер
полтора г

С 18

своей р

но не России (тогда в СССР) образцами машин первого поколения могут быть идеи и МЭСМ (малая электронная счетная машина) — в 1950 г. самая быстро-
современная в Европе — и БЭСМ (большая электронно-счетная машина). Обе
жал разработаны под руководством академика С. А. Лебедева.

(«

Четвертое поколение

Компонентной основой компьютеров второго поколения были транзисторы. Потребляемая электрическая мощность и размеры транзисторов гораздо меньше их электронно-вакуумных предков. Кроме того, полупроводниковые устройства позволили значительно уменьшить физические размеры памяти компьютеров и повысить быстродействие. В этом поколении происходит серьезное усложнение арифметического логического устройства и устройства управления ходом выполнения программы. В это же время сформировалось понятие системного программного обеспечения и стало быстро развиваться программирование на языках высокого уровня.

Типичным представителем компьютеров этого поколения были компьютеры IBM серии 7000. В СССР реализовано сразу несколько серий ЭВМ второго поколения средней и малой мощности: БЭСМ, «Урал», «Минск», «Раздан», «Днепр».

Третье поколение

Третье поколение компьютеров во многих отношениях можно назвать особенной вехой в истории развития компьютерных технологий. Был реализован переход от дискретной (основанной на отдельных транзисторах, лампах и вспомогательных элементах) логики к БИС и СБИС, то есть к большим и сверхбольшим интегральным схемам. Наука и технология совершила шаг от электроники к микроэлектронике. Для этого потребовалось разработать очень точные технологии создания сверхслож-

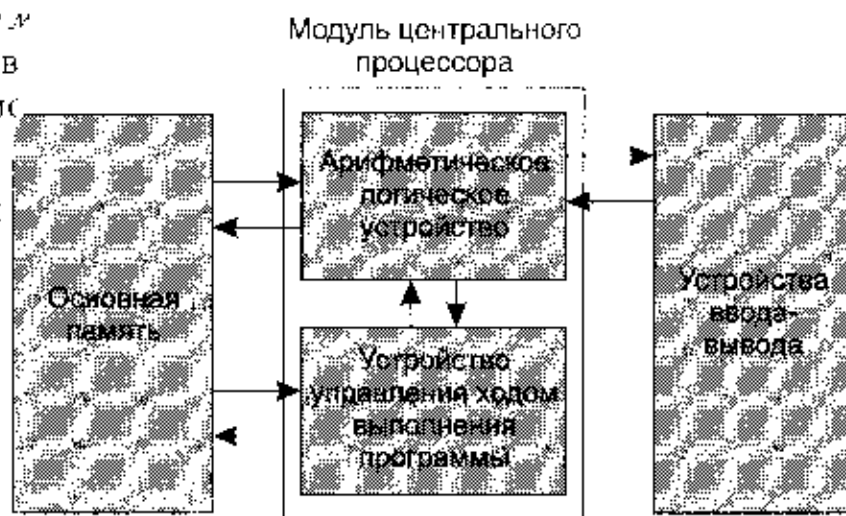


Рис. 12.1. Компьютер первого поколения

ного рисунка электронной схемы на площади размером всего в несколько квадратных миллиметров. Благодаря этим технологиям электронные схемы, которые ранее собирались на больших электронных платах (размером, к примеру, полметра на полметра) и содержали в себе сотни и тысячи транзисторов и других дискретных деталей, теперь создавались на нескольких квадратных миллиметрах площади, а будучи упакованными в корпус занимали два-три квадратных сантиметра.

Микроэлектронная технология оказалась удачным решением не только для компьютеров, но и практически для всех областей техники, где так или иначе применялись электронные устройства. Будучи широко востребованными, микроэлектронные устройства бурно развивались, и в 1965 г. один из ученых, сотрудничавших с корпорацией Intel, Гордон Мур, сформулировал свой знаменитый закон: каждый последующий год плотность транзисторов на единицу площади интегральной схемы будет удваиваться. Дальнейшее развитие микроэлектронных технологий подтвердило удивительную точность этого прогноза.

Типичным представителем компьютеров третьего поколения можно назвать IBM серии System/360. Это были все еще большие по размеру машины (на столе их разместить было невозможно, и стоили они сотни тысяч долларов), которые применялись в качестве мэйнфреймов и активно производились IBM начиная с 1964 г.

В СССР компьютеры третьего поколения были отмечены отечественными разработками серии ЕС (ЕС 1020, ЕС 1030, ЕС 1050) и БЭСМ-6.

Четвертое поколение

Практически параллельно запуску серии System/360 корпорация DEC выпустила на рынок компьютеры с совершенно новой организацией и новыми возможностями. Это были машины серии PDP-8. Отличительными особенностями этих машин были: линейная организация, обеспечивавшая исключительную гибкость в расширении и встраивании компьютера; микропроцессор, способный выполнять операции с плавающей точкой на аппаратном уровне; полупроводниковая память. И самая главная особенность — исключительная компактность и сравнительно низкая цена: компьютер уже можно было разместить на столе, а его стоимость составляла «всего» 16 000 долларов. Это еще не был массовый персональный компьютер, но небольшие лаборатории и производства, предприятия среднего бизнеса уже могли позволить себе приобрести такие машины.

То, что произошло дальше, смело можно назвать «путем Intel» в компьютерной технологии. Корпорация Intel, развивая технологии изготовления микропроцессорных устройств, последовательно предложила рынку целую линейку своих продуктов (табл. 12.1).

Таблица 12.1. Процессоры Intel

Десятилетие	Процессоры
1970–1980	4004, 8008, 8080, 8086, 8088
1980–1990	80286, 386 TM SX, 386 TM DX, 486 TM DX
1990–2000	486 TMSX, Pentium, Pentium II, Pentium III, Pentium IV
2000...	Itanium

На основе этих процессоров создаются и получают небывалое развитие персональные компьютеры. Это уже не столько техническое, сколько социально-культурное явление, важность которого трудно переоценить.

Параллельно компьютерам с архитектурой Intel, начиная с 1993 г., развиваются компьютеры с архитектурой PowerPC, известные нам как компьютеры Apple Macintosh. Эти машины изначально имели более производительную 32-битную организацию и еще в 1996 г. перешли на 64-разрядную шину. Используя суперскалярную архитектуру организации микропроцессора и встроенную (on-chip) кэш-память, компьютеры Apple Macintosh выигрывают по производительности у машин Intel, но дешевизна персональных компьютеров с процессорами Intel обеспечила последним заслуженную популярность и доминирование на столах пользователей.

В СССР четвертое поколение ЭВМ было отмечено целым букетом отечественных разработок: это модифицированные версии ЕС ЭВМ, целый ряд двухпроцессорных машин, многопроцессорные машины Эльбрус, мини-ЭВМ серии СМ, профессиональный персональный компьютер ЕС-1840 (он уже не занимал в кабинете все свободное место, но его еще нельзя было поставить на стол, поскольку размерами он сам был лишь немногим меньше письменного стола). Особенными событиями, породившими в СССР эпоху «наколенных» компьютеров, были выпуск компьютера «Электроника БК-0010-01» с процессором, совместимым с PDP-11, и компьютера «Москва», совместимого с ZX Spectrum. Выпуск этих компьютеров перенес компьютерное творчество в народные массы. Повсеместно «наколенным» способом собирались десятки, если не сотни тысяч, самодельных машин ZX Spectrum и модифицировались БК-0010. Трудно себе представить, сколько людей в нашей стране получило свое первое представление о программировании не на школьных уроках, а дома, сидя перед телевизором (поскольку у первых бытовых компьютеров не было дисплея, а в качестве такового использовался экран телевизора).

Российская наука внесла в компьютеры четвертого поколения свой вклад многопроцессорными машинами «Эльбрус-3» (1994 г.) и «Эльбрус-90 Микро» (1996 г.).

Пятое поколение

Поскольку за основу деления компьютеров на поколения мы приняли элементную базу, говорить о машинах пятого поколения пока еще рано. Несмотря на то что компьютерные вычисления совершенствуются, усложняется и миниатюризируется структура микропроцессоров, этот процесс пока продолжается в рамках все той же парадигмы сверхбольшой интегральной схемы. Появление многоядерных процессоров или преодоление суперкомпьютерами петафлопсного рубежа (от flops — float operation per second, то есть операций с плавающей точкой в секунду) не может с этой точки зрения быть границей перехода к следующему поколению. Возможно, успехи пано- или фемто-технологий позволят создавать процессоры на принципиально иной основе, и тогда уже можно будет говорить о компьютерах пятого поколения.

12.2. Классификация компьютеров и вычислительных систем

С момента своего появления компьютеры бурно развивались и активно входили во все сферы деятельности человека, иногда опережая прогнозы писателей-фантастов. Подобно любому явлению, широко и многопланово вошедшему в нашу жизнь, компьютеры могут быть классифицированы по самым разным признакам: по принципу действия, по мощности и габаритам, по назначению и области применения, по роли, которую они выполняют в сети, по внутренней архитектуре. Широта применения компьютеров, самых разных по мощности и размерам, проникновение компьютеров практически во все области жизни приводят к тому, что становится все труднее провести границы между условно выделенными группами. Так, благодаря технологиям кластеризации суперкомпьютеры легко собираются из обычных рабочих станций, в то же время микроЭВМ на уровне процессора повышают свою производительность за счет архитектур, свойственных суперкомпьютерам.

12.2.1. Классификация по принципу действия

Аналоговая ЭВМ, или аналоговый компьютер — это вычислительное устройство непрерывного действия, обрабатывающее аналоговые данные без преобразования их в цифровую форму.

ЭВМ такого типа в свое время позволяли производить моделирование сложных физических процессов, сразу же получая сведения о состоянии модели в любой ее точке и на любом этапе процесса. Несмотря на невысокую точность результатов (погрешность достигала 5 %), аналоговые ЭВМ отличались простотой эксплуатации, их программирование сводилось к выставлению определенных значений для параметров в тех или иных узлах и сборке схемы соединения узлов. Это определило популярность аналоговых машин и их довольно широкое применение при выполнении проектных и исследовательских работ. Однако по мере совершенствования цифровых ЭВМ, а также математических методов описания моделей процессов, аналоговые ЭВМ были вытеснены цифровыми и гибридными (рис. 12.2).



Рис. 12.2. Классификация компьютеров по принципу действия

Цифровая ЭВМ, или цифровой компьютер, — это вычислительное устройство дискретного действия, где вычисления производятся в цифровой форме.

Аналого-цифровая (гибридная) ЭВМ, или гибридный компьютер, — это компьютер, совмещающий в себе обработку сигналов как в аналоговой, так и в цифровой формах.

Современные гибридные ЭВМ чаще всего применяются в узкоспециальных областях, таких как медицина и навигация, а также выступают в роли аналого-цифровых и цифроаналоговых преобразователей в робототехнике.

12.2.2. Классификация по вычислительной мощности и габаритам

Классификация компьютеров по вычислительной мощности и габаритам представлена на рис. 12.3.



Рис. 12.3. Классификация компьютеров по вычислительной мощности и габаритам

Суперкомпьютер, или суперЭВМ, — это сверхпроизводительный компьютер, предназначенный для решения задач особой сложности.

Сверхпроизводительность достигается за счет параллельной архитектуры, сверхглубокого охлаждения процессора и полупроводниковой памяти, применения особых сверхбыстрых сверхбольших интегральных схем и каналов связи. Назначение суперкомпьютеров — решать задачи, в которых огромные объемы данных нужно обработать за короткое время (системы предупреждения ядерного нападения и стихийных бедствий, прогнозирования погоды, авиаперевозок).

Большая ЭВМ — это производительный компьютер, обычно с большим объемом оперативной и дисковой памяти.

Современные большие ЭВМ обычно построены с использованием многоядерной, многопроцессорной или многокомпьютерной технологии (все чаще встречаются случаи совмещения двух или трех названных технологий в одной машине). Назначение больших ЭВМ — серверы в мэйнфреймовых системах (собственно мэйнфреймы), крупные интернет-серверы, серверы данных в больших распределенных системах. По своему объему большие ЭВМ таковы, что требуют отдельного компьютерного зала. Нужно отметить, что большие ЭВМ — уходящая ветвь, развитие кластерных технологий сделало их практически ненужными: кластеры, собранные на основе персональных или серверных ЭВМ, могут обеспечить ту же вычислительную мощность и вдобавок являются масштабируемыми.

Средняя ЭВМ — это достаточно производительный компьютер, требующий для своего размещения отдельного помещения с поддерживаемыми климатическими условиями.

Назначение средних ЭВМ — серверы баз данных и информационных системах среднего масштаба, управляющие компьютеры в системах автоматизированного производства. Все замечания по вытеснению больших ЭВМ кластерными сборками верны и для средних ЭВМ.

Малая ЭВМ — это компьютер, соответствующий по размерам персональному рабочему месту, но значительно превышающий своего настольного собрата: размеры малых ЭВМ колеблются от платяного шкафа до небольшого холодильника.

На этих ЭВМ выполняются инженерные расчетные задачи, моделирование, они могут служить ведущим компьютером в системе управления станками ЧПУ.

МикроЭВМ — это вычислительная машина с габаритами, начиная от настольного компьютера и меньше. В эту категорию попадают персональные компьютеры, микроконтроллеры, встраиваемые микропроцессоры, игровые приставки.

12.2.3. Классификация по способу применения

Эта классификация, вероятно, будет наиболее развернутой и подробной, поскольку ЭВМ одной и той же мощности можно применять самыми разными способами и, наоборот, одним и тем же способом можно использовать ЭВМ разной мощности, в зависимости от потока обрабатываемой информации (рис. 12.4).

Научные вычислительные центры большой мощности — это поле деятельности суперЭВМ. Обычно в таких научных центрах решаются вычислительные задачи моделирования сверхсложных многофакторных процессов (например, моделирование процесса развития и существования различных космических объектов, моделирование биологических, социальных или психофизиологических процессов, прогнозирование погоды). С января по сентябрь 2008 г. в мире было установлено 53 суперкомпьютера, из которых 47 попали в университеты и научные учреждения, 4 — в национальные гидрометеоцентры и только 2 напрямую предназначались для использования в бизнесе и выполнения инженерных расчетов.

Мейнфрейм — это мощный компьютер в сетевой инфраструктуре, в которой на стороне пользователя находится терминал, т. е. устройства ввода (клавиатура, мышь или другой манипулятор) и вывода (дисплей).



Рис. 12.4. Классификация компьютеров по способу применения

Вся информация с клавиатуры поступает в базовый компьютер, а от него на рабочее место приходит информация, отображаемая на дисплее (рис. 12.5). В качестве мейнфрейма обычно выступают большие или средние ЭВМ, но все чаще возникают прецеденты замены в этой роли большой ЭВМ суперкомпьютером.

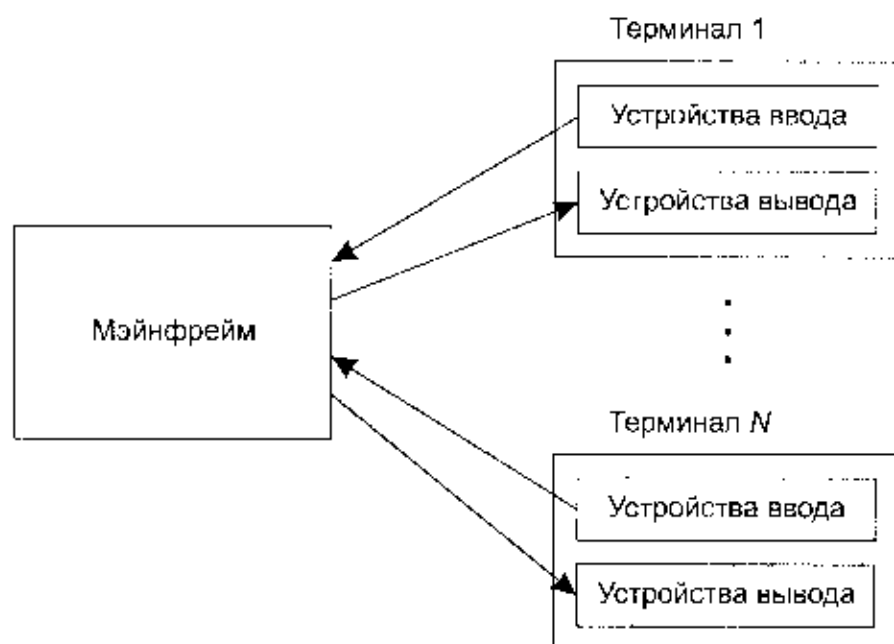


Рис. 12.5. Мейнфреймовая архитектура

Сервер в клиент-серверной архитектуре (сервер баз данных) — это компьютер, на котором расположена база данных и к которому подключаются персональные клиентские рабочие станции (обычно по локальной сети).

В зависимости от количества рабочих станций и объема хранимых и обрабатываемых данных в этой роли может выступать как ЭВМ средней мощности (специализированная серверная машина), так и обычный персональный компьютер, но с увеличенным объемом оперативной памяти и расширенным дисковым пространством (рис. 12.6).

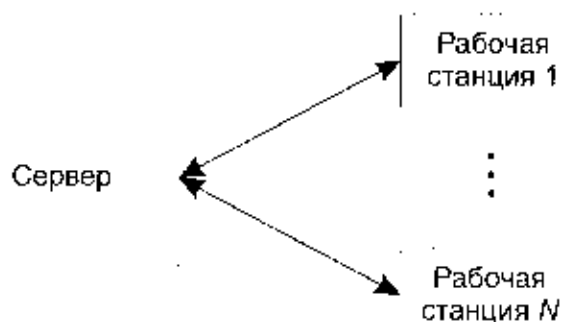


Рис. 12.6. Клиент-серверная архитектура

Клиентский компьютер, или рабочая станция, — это персональный компьютер с ограниченной функциональностью, минимальным объемом оперативной памяти и сокращенной номенклатурой устройств ввода-вывода.

Сервер приложений, представляет собой промежуточное звено между сервером баз данных и клиентским компьютером в многозвенной архитектуре построения информационной системы.

Сервер среднего звена обычно берет на себя часть нагрузки основного сервера системы, поэтому в качестве такого сервера может применяться как специализированная серверная ЭВМ, так и персональный компьютер с расширенным объемом оперативной памяти (рис. 12.7).

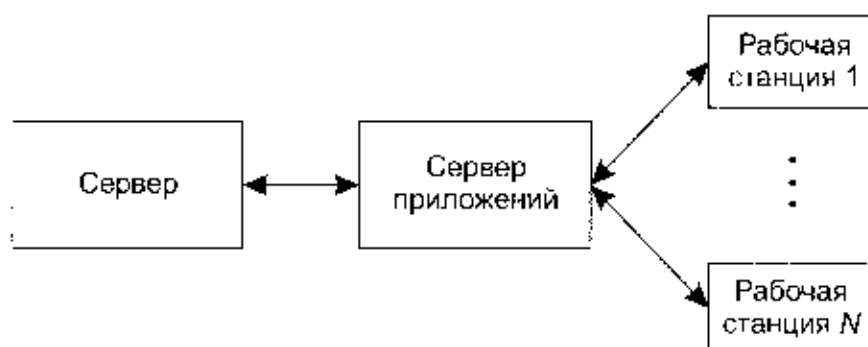


Рис. 12.7. Многозвенная (в данном случае — трехзвенная) архитектура

Сетевые серверы различного назначения (как в локальной сети, так и в Интернете) объединяют веб-серверы, ftp-серверы, почтовые серверы и серверы конференций.

Все эти серверы могут располагаться на одном мощном серверном компьютере или на нескольких рабочих станциях.

Помимо перечисленных, по способу применения различаются:

- *Бытовые ЭВМ* — домашние персональные компьютеры.
- *Персональные информационные менеджеры (ПИМ)* — малогабаритные «карманные» или «паладонные» ЭВМ для оперативного управления персональной информацией.
- *Игровые приставки* — персональные ЭВМ, специально предназначенные для компьютерных игр.
- *Борттовые ЭВМ* устанавливаются на борту средств передвижения: автомобилей, самолетов, кораблей, спутников с целью управления техническими параметрами и навигации.
- *Управляющие ЭВМ* предназначены для управления технологическими процессами на производстве.
- *Микроконтроллеры*, обычно встраиваемые в производственные роботы и автоматизированные линии, — это специальный вид микроЭВМ, оформленный в виде блоков с цифровыми каналами ввода (от датчиков) и вывода (для управления исполнительными устройствами).
- *Встроенные ЭВМ* — это микроЭВМ, управляющие сегодня широким спектром бытовой и специальной техники, хотя потребитель может даже не иметь представления об этом. Чипы микропроцессоров дешевы и малы по размеру, так что вы можете даже не предполагать, что ваш пылесос, холодильник, телевизор, музыкальный комбайн, стиральная или посудомоечная машина работают по программе, вшитой в запоминающее устройство однокристалльной ЭВМ. Встроенными микроЭВМ оснащаются сегодня игрушки, электронные кредитные карты и даже открытки.
- *Специализированные рабочие станции* — это персональные компьютеры, оснащенные специальным дополнительным оборудованием. Они могут работать изолированно или в сети, будучи подключенными к серверу. Одна из областей применения — медицина, в качестве диагностического рабочего места или в составе сети медицинского наблюдения.

12.3. Настоящее и будущее компьютеров

12.3.1. Микропроцессоры

Любой из современных компьютеров создан на основе сверхбольших интегральных схем и в качестве исполнителя программы и вычислителя имеет один или несколько микропроцессоров. Поэтому обсуждать настоящее состояние современных вычислительных машин мы начинаем именно с этого устройства.

Как уже отмечалось, развитие полупроводниковой технологии полностью подтвердило правоту прогноза Гордона Мура, согласно которому примерно каждый год число транзисторов в микросхеме будет удваиваться. Этот прогноз оказался настолько точным, что получил название закона Мура (табл. 12.2).

Таблица 12.2. Закон Мура в действии (на примере процессоров корпорации Intel)

Микропроцессор	Год выпуска	Количество транзисторов
4004	1971	2300
8008	1972	2500
8080	1974	5000
8086	1978	29000
286	1982	120000
Intel 386™ processor	1985	275000
Intel 486™ processor	1989	1180000
Intel® Pentium® processor	1993	3100000
Intel® Pentium® II processor	1997	7500000
Intel® Pentium® III processor	1999	24000000
Intel® Pentium® 4 processor	2000	42000000
Intel® Itanium® processor	2002	220000000
Intel® Itanium® 2 processor	2003	410000000

Этот закон одновременно демонстрирует беспрецедентную скорость развития микропроцессорной технологии и дает возможность вычислить срок, когда человечество подойдет к технологическим границам, делающим невозможной дальнейшую миниатюризацию. Дело в том, что увеличение количества транзисторов на одном микропроцессоре при сохранении размеров самого чипа должно вести к уменьшению размера транзистора. Но размер транзистора невозможно уменьшать до бесконечности: как дом не может быть размером с кирпич, из которого он построен, так и транзистор рано или поздно достигнет размеров, сравнимых с размерами составляющих его электронов и его дальнейшее уменьшение станет невозможным. Еще в 2000 г. сроком достижения такого барьера называли 2020 г., но сегодняшний прогноз приблизил эту границу, и «технологический тупик» ожидается уже в 2017 г. С приближением подобных сроков всегда происходит оживленный поиск в пограничных отраслях знания. Бизнес и правительства разных стран выделяют большие средства на поиск новых путей в микроэлектронике, вот почему так популярны сейчас темы «нетранзисторных» и «некремниевых» компьютеров — устройств, основанных на химических, биологических или световых процессах.

Тем не менее специалисты корпорации Intel уверяют, что у них уже есть научные и технические решения, позволяющие следовать закону Мура до 2020 г. В качестве таких решений называют многослойные структуры микропроцессора, применение нанотехнологий, и другие.

Кроме проблем, связанных с геометрическими размерами элементов, все сильнее проявляются проблемы сопутствующие. Увеличение плотности транзисторов на квадратном миллиметре площади (а в будущем — на кубическом миллиметре

объема) означает повышенную теплоотдачу. Каждый транзистор в момент переключения выделяет некоторое количество тепла. Чем больше транзисторов, тем этого тепла больше. Также повышается теплоотдача при увеличении частоты переключения транзисторов. Современные микропроцессоры излучают с каждого миллиметра столько тепла, сколько излучает металл, нагретый до температуры в тысячу с лишним градусов. Если это тепло не отводить, разрушение процессора неминуемо. Вот почему каждое новое поколение процессоров оснащалось все более массивными радиаторами и мощными кулерами (вентиляторами для отвода тепла). Современные микропроцессоры нередко оснащаются не только воздушными, но и водяными или гелиевыми средствами охлаждения, а в состав суперкомпьютера уже привычно включаются криогенные установки. Кроме того, микропроцессоры оснащаются специальным технологическим и программным обеспечением, позволяющим отслеживать температурный режим процессора, гибко регулировать его частоту, изменять интенсивность обдува воздухом, отключать процессор при приближении недопустимых температур.

ПРИМЕЧАНИЕ

Для того чтобы понять, как сильно нагревается микропроцессор даже в простом персональном компьютере, не надо быть специалистом: достаточно жарким летним днем попытаться поработать, положив ноутбук на колени. Через 15–20 минут вы в полной мере поймете, насколько близко к технологическим пределам подошла микропроцессорная техника.

Таблицы 12.3 и 12.4 достаточно точно характеризуют современное положение на рынке микропроцессорных устройств. Естественно, в эти таблицы попали только фирмы-лидеры и только лучшие образцы современных микропроцессоров. Несмотря на неполноту этих данных, по ним можно отметить как уровень, на котором находится микропроцессорная техника, так и направления, в которых она развивается.

Таблица 12.3. Микропроцессоры для персональных компьютеров и рабочих станций

Производитель	Название (серия)	Количество ядер	Частота процессора, ГГц	Дополнительные сведения
Intel	Intel® Core™2 Extreme Quad	4	3,2	FBD RAM, 12-мегабайтный кэш 2-го уровня
	Intel® Core™2 Quad	4	3,0	DDR2, DDR3, 12-мегабайтный кэш 2-го уровня
	Intel® Core™ 2 Duo	2	3,33	DDR2, DDR3, 6-мегабайтный кэш 2-го уровня
AMD	AMD Athlon™ X2 Dual-Core	2	3,200	32/64 bit operation mode
	AMD Athlon™ FX	1	3,0	
	AMD Athlon™	1	2,400	32/64 bit operation mode

Таблица 12.4. Микропроцессоры для серверов

Производитель	Название (серия)	Количество ядер	Частота процессора, ГГц	Дополнительные сведения
Intel	Intel® Xeon® серии 7400	6	3,5	64-bit, 16 мегабайтный кэш 3-го уровня, 256 Гбайт RAM, технология виртуализации, самое низкое энергопотребление из расчета на одно ядро
	Intel Xeon серии 5000	4	3,4	64-bit, 8-мегабайтный кэш 3-го уровня, 64 Гбайт RAM, технология виртуализации
	Intel® Xeon® 3000	2	3,0	64-bit, 12-мегабайтный кэш 3-го уровня, 8 Гбайт RAM, технология виртуализации
	Intel® Itanium®	2	1,6	Технология Hyper-Threading, автоматическая регулировка энергопотребления
AMD	Third-Generation AMD Opteron™	4	2,5	64-bit, мегабайтный кэш 2-го уровня, архитектура direct connect для повышения скорости межпроцессорного взаимодействия
	Second-Generation AMD Opteron™	2	3,6	64-bit, мегабайтный кэш 2-го уровня, архитектура direct connect для повышения скорости межпроцессорного взаимодействия
IBM	POWER6	2	4,7	128-мегабайтный кэш 1-го уровня, 4-мегабайтный кэш 2-го уровня, 32-мегабайтный кэш 3-го уровня, ядро с двумя потоками команд
	POWER7	8	5	
	CELL PowerPC® 8i	2 + 8 синергетических элементов	3,2	
SUN	UltraSPARC T2	8	1,4	На сегодняшний день единственный процессор этого класса, выполненный по технологии System On Chip, то есть имеющий встроенную поддержку сети, шифрования и расширение ввода-вывода PCI Express

Сведения, содержащиеся в этих двух таблицах, достаточно ясно намечают ближайшие перспективы развития микропроцессоров, на основе которых создаются как персональные компьютеры, так и мощные серверы, или суперкомпьютеры. Все эти перспективы разворачиваются в первую очередь в сторону попыток обойти физические и технологические ограничения, сдерживающие рост производительности процессоров.

- ❑ *Повышение количества ядер.* На одном кремниевом кристалле создается не один процессор, а несколько. При этом, в зависимости от выбранной разработчиком архитектуры процессора, эти несколько процессоров могут совместно, зависимо использовать общие, расположенные на одном кристалле ресурсы (таким образом повторяя в каких-то чертах структуру построения суперкомпьютеров) для повышения производительности, или могут быть настолько независимы друг от друга, чтобы на каждом из них одновременно можно было запустить свою операционную систему. Такая схема используется в процессорах Intel, что делает возможным эффективное применение технологии виртуализации.
- ❑ *Увеличение разрядности.* Постепенно происходит переход от 32-разрядных процессоров к процессорам с 64-разрядной шиной. Для серверных процессоров это уже де-факто стандарт, для процессоров персональных компьютеров это реализованная на уровне процессора возможность. Надо заметить, что повышение разрядности вдвое не означает повышения вдвое производительности компьютера в целом. Однако для решения некоторых задач, требующих массовой обработки данных, это именно так.
- ❑ *Автонастройка.* Все больше процессоров оснащаются встроенными механизмами регулирования производительности, контроля температурного режима, энергопотребления и производительности в зависимости от загрузки процессора.
- ❑ *Многопоточность ядра.* Ядра процессоров получают возможность образовывать внутри себя несколько независимых потоков команд и выбирать тот, который в данный момент будет выполняться.
- ❑ *System-On-Chip.* Целый комплекс задач, который раньше решался сочетанием аппаратных средств и средств операционной системы, теперь решается непосредственно в мультипроцессорном кристалле. Это на порядок повышает скорость выполнения многих функций за счет сокращения числа обращений к системной шине и облегчает программирование многих приложений (например, если шифрование данных является встроенным).
- ❑ *Встроенные механизмы интеграции и масштабирования.* Этим отличаются в основном серверные микропроцессоры; в их конструкцию изначально закладываются средства работы в «больших микропроцессорных коллективах», из которых собираются мощные серверы и суперкомпьютеры.

12.3.2. Суперкомпьютеры

Микропроцессорные устройства довольно близко подошли к технологическому пределу как своей миниатюризации, так и увеличения тактовой частоты. Самые производительные на сегодняшний день процессоры обеспечивают вычислительную мощность меньше одной четырехтысячной самого производительного суперкомпьютера. Каким же образом достигается эта невероятная скорость обработки информации? Механизм, при помощи которого удастся создавать суперкомпьютеры, этих титанов компьютерного мира, один: параллельные вычисления.

Если между двумя точками пути, *A* и *B*, расстояние 100 км и на это расстояние нужно переместить некий пакет с сообщением, то нет никакой разницы,

один человек будет его проходить, сотня или тысяча — результат будет одинаков, ускорения не произойдет. Однако если нужно почистить тысячу картофелин, то тысяча человек сработает в тысячу раз быстрее, чем один. То есть существуют такие задачи, которые можно решать параллельно (чистка картошки), и такие, которые решаются только последовательно (преодоление расстояния). К счастью, подавляющее большинство задач в компьютерном мире носят параллельный характер — будь то прием информации от метеорологических спутников, анализ состояния от сотен тысяч детекторов частиц адронного коллайдера или обработка запросов пользователей корпоративной информационной системы. Большинство научных расчетных задач и процессов моделирования также может быть разбито на параллельно выполняемые потоки. Каждая параллельная задача может обрабатываться одним микропроцессором. Таким образом, создание суперкомпьютера может быть сведено к решению следующей задачи: как соединить между собой множество микропроцессоров, чтобы каждый из них выполнял отдельное задание и в то же время они представляли собой единое целое, один микроэлектронный супермозг.

Существуют несколько способов заставить микропроцессоры выполнять параллельные вычисления и множество вариантов классификации этих конфигураций. Наиболее информативной и часто используемой (а также наиболее простой) является классификация Флинна (рис. 12.8).

	Одиночный поток команд (<i>Single Instruction</i>)	Множество потоков команд (<i>Multiple Instruction</i>)
Одиночный поток команд (<i>Single Data</i>)	<i>SISD</i>	<i>MISD</i>
Множество потоков данных (<i>Multiple Data</i>)	<i>SIMD</i>	<i>MIMD</i>

Рис. 12.8. Классификация Флинна

SISD (Single Instruction, Single Data) — один поток команд с одним потоком данных.

К этому классу можно отнести все настольные компьютеры с одним процессором. В этом случае мы имеем одну последовательность инструкций, которую выполняет процессор в одном потоке данных.

SIMD (Single Instruction, Multiple Data) — это один поток команд с множеством потоков данных.

В этом классе один поток инструкций выполняется сразу над множеством наборов данных. Такое поведение свойственно *векторным процессорам*, или *векторным машинам*. Одна инструкция обрабатывается сразу множеством процессоров,

каждым в своей памяти, или одним процессором, но сразу во множестве регистров. Это дает возможность за один такт обработать большой массив данных. Массив однотишных данных составляет вектор, отсюда и название подобного рода процессоров (архитектур).

MISD (Multiple Instruction, Single Data) — это множество потоков команд с одним потоком данных.

Этот класс является пустым, поскольку есть только теоретические предположения о том, как могла бы выглядеть подобная архитектура, но практической реализации ни одной нет.

MIMD (Multiple Instructions Multiple Data) — это множество потоков команд с множеством потоков данных.

В этом классе множество потоков команд выполняется над множеством потоков данных. К данному классу можно отнести практически все современные суперкомпьютеры, оснащенные большим числом микропроцессоров.

Кроме этой классификации есть еще множество других вариантов классификации компьютеров, выполняющих параллельные вычисления и уточняющих диаграмму Флинна. К примеру, очень важным показателем является то, как процессоры, входящие в состав суперкомпьютера, взаимодействуют с памятью.

SMP (Symmetric MultiProcessing) — это симметричное мультипроцессирование.

Все микропроцессоры, входящие в состав суперкомпьютера, подключены к одному адресному пространству, к одной памяти при помощи специальной высокоскоростной шины памяти. При этом все эти процессоры абсолютно равноправны с точки зрения доступа к любому адресу этой памяти. Эта архитектура дает наибольший выигрыш по производительности, но плохо масштабируется (распирается) и не может содержать большого количества процессоров, поскольку увеличение количества процессоров приводит к резкому возрастанию вероятности конфликтов при доступе к одним и тем же адресам памяти. Кроме того, сама высокоскоростная шина имеет физические ограничения, не позволяющие наращивать ее объем.

MPP (Massive Parallel Processing) — это обработка с массовым параллелизмом.

В этом случае общей памяти нет. Каждый процессор или модуль с несколькими процессорами является владельцем своего банка памяти, а между процессорами (модулями) устанавливаются соединения, образующие топологию вычислительной системы. Соединения могут быть выполнены как при помощи обыкновенных сетевых устройств, так и посредством специальных вспомогательных компьютеров, предназначенных для высокоскоростной передачи данных (*транзьютеров*).

NUMA (Nonuniform Memory Access) — это неоднородный доступ к памяти.

При такой архитектуре несколько процессоров объединяются между собой в SMP-узел, а SMP-узлы, в свою очередь, образуют MPP-архитектуру.

PVP (Parallel Vector Process) — это параллелизм с векторными процессорами.

Обычно это одна из перечисленных архитектур (SMP, MPP или NUMA), в которой задействованы не обычные скалярные процессоры, а специальные процессоры векторно-конвейерного типа. Это даст возможность эффективным образом организовывать параллельные вычисления любого типа. Недостатком такой архитектуры является большая стоимость как самих процессоров, так и программного обеспечения для организации вычислений.

Кластеры — это узлы, объединенные при помощи сети таким образом, что для конечного пользователя они предстают одним устройством.

В состав кластера могут входить как специализированные компьютеры, то есть изготовленные специально для организации кластера, так и обыкновенные рабочие станции. Физически кластеры могут быть организованы в одной локальной сети (гомогенная организация) или объединяться через разного рода сетевые соединения, включая Интернет (так называемая гетерогенная структура). Кластерная организация суперкомпьютеров на сегодняшний день является суперпопулярной. Достаточно сказать, что из 500 самых мощных суперЭВМ в мире 400 созданы на основе кластерной архитектуры.

И еще несколько важных понятий:

- **Суперскалярный процессор** — процессор, который способен выполнить несколько операций за один такт. Естественно, для того чтобы это стало возможным, у суперскалярного процессора должны быть «в подчинении» независимые устройства, которым можно разослать этот пакет команд для обработки.
- **Конвейерная обработка** — при такой обработке повторяющаяся последовательность операций делится на ряд подопераций и каждая подоперация выполняется отдельным процессором. В результате получается, что за один такт выполняется не одна, а множество команд.
- **RISC (Reduced Instruction Set Computer)** — компьютер с сокращенным набором команд. В результате сокращения в наборе команд процессора остаются только инструкции, которые можно выполнить за 1–2 такта (в то время как в полном наборе команд могут быть инструкции, требующие 4–6 тактов).

Одним из важнейших показателей, при помощи которых оценивают суперкомпьютеры, является производительность. Производительность измеряется в количестве операций с плавающей точкой в секунду (Float Point Operation Per Second, FLOPS). Рост этого показателя для суперкомпьютеров сравним с тенденцией роста плотности транзисторов для микропроцессоров. В 2000 г. еще только ставилась задача преодоления барьера в 1 TFLOPS (терафлопс, миллиард флопсов), а в 2008 г. уже был преодолен рубеж в 1 PFLOPS (петафлопс, триллион флопсов).

Современные достижения в суперкомпьютерной технике отслеживает специальный рейтинг пяти сотен самых производительных суперкомпьютеров в мире. В настоящее время 5 первых строчек этого рейтинга занимают американские компьютеры (табл. 12.5).

Таблица 12.5. Первые пять мест рейтинга самых мощных суперкомпьютеров

Место	Страна	Система и производитель	Кол-во процессоров	Макс. произв., TFLOPS	Пиковая произв., TFOLPS
1	DOE/NNSA/LANL, United States	BladeCenter QS22/LS21 Cluster, PowerXCell 8i 3.2 GHz / Opteron DC 1.8 GHz, Voltaire Infiniband IBM	122 400	1026	1375,78
2	DOE/NNSA/LLNL, United States	eServer Blue Gene Solution IBM	212 992	478,2	596,38
3	Argonne National Laboratory, United States	Blue Gene/P Solution IBM	163 840	450,3	557,06
4	Texas Advanced Computing Center/Univ. of Texas, United States	SunBlade x6420, Opteron Quad 2GHz, Infiniband Sun Microsystems	62 976	326	503,81
5	Oak Ridge National Laboratory, United States	Cray XT4 QuadCore 2.1 GHz, Cray Inc	30 976	205	260,2

Из этих пяти пунктов видно, что всемирно известный производитель суперкомпьютеров корпорация Cray находится только на пятом месте. Новейший микропроцессор IBM CELL PowerXCell 8i с его гибкой архитектурой, в которой кроме 2 базовых ядер есть еще 8 вспомогательных, работающих как в режиме векторных сопроцессоров, так и в режиме независимых ядер, позволил не только выйти на первое место в рейтинге, но и преодолеть петафлопсный барьер.

В рейтинге top500, благодаря выдвинутым в нашей стране приоритетам на оснащение суперкомпьютерами университетов и научных центров, присутствие России увеличивается каждый месяц. На сегодняшний день в рейтинге уже 9 российских суперкомпьютеров, причем лучший из них занимает 36-е место (СКИФ МГУ).

12.3.3. Серверные компьютеры

Сервер — это компьютер или система компьютеров, связанная через сетевые соединения с другими компьютерами и обрабатывающая их запросы.

Назначение современных серверов самое широкое: серверы Интернета, серверы телекоммуникационных систем, серверы информационных систем, серверы баз данных.

Несмотря на то что в случае малого количества клиентов и простоты выполняемых функций, сервер физически может располагаться даже на обыкновенном персональном компьютере, в случае предприятий или учреждений такое решение неприемлемо. Современный серверный компьютер должен отвечать требованиям, которым персональная ЭВМ соответствовать не может. Дело в том, что сегодняшний день плотно связал между собой несколько понятий: «информация», «деньги», «безопасность» и «экологичность». Надежность работы сервера и надежность хранения информации должны быть такими, чтобы полностью исключить, с одной стороны, возможность потери, искажения или несанкционированной выдачи даже небольших объемов хранящейся информации, с другой — полностью исключить возможность даже краткой остановки работы информационной системы, в которую входит сервер.

Если сервер обслуживает школьный интернет-портал, то сутки-другие простоя могут нанести некоторый ущерб имиджу школы, привести к неприятным, но не трагическим последствиям. Но что будет, если на сутки остановится сервер, обслуживающий информационную систему крупного банка или аэропорта? Это может привести к многомиллионным финансовым потерям или трагическим последствиям. Поэтому современные специализированные серверы изначально конструируются с учетом особенностей их работы.

- *Размещение.* Серверный компьютер должен быть размещен в специально выделенном помещении («серверная»). Это помещение должно иметь ограниченный доступ. Современные серверные компьютеры создаются с учетом размещения в специальных шкафах-стойках, оборудованных вентиляцией, резервной системой питания и разного рода сигнализацией.
- *Резервирование.* Современные серверы не должны останавливаться даже в случае, когда они выходят из строя на физическом уровне (что-то в компьютере «перегорело»). Поэтому обычно серверный компьютер создается так, чтобы продолжать свое функционирование даже после выхода из строя части комплектующих (например, если перестает функционировать часть линеек оперативной памяти). Кроме того, обычно в сети есть резервный компьютер, который может принять на себя выполнение функций сервера в случае его полного выхода из строя (или при остановке его для технического обслуживания). Для создания систем повышенной надежности (с многократным резервированием) активно используется кластерная архитектура. Надо сказать, что многие современные информационные системы используют не двукратное, а пятикратное (в случае атомных станций) или одиннадцатикратное (в случае информационных систем предупреждения ядерного нападения) резервирование.
- *Защита от вирусов.* Во все большем количестве современных серверов средства антивирусной защиты встраиваются непосредственно в сервер, иногда прямо в серверный процессор (System-On-Chip).
- *Резервное копирование и хранение данных.* Большинство серверов в информационных системах являются промежуточным звеном между пользователем и хранимыми на сервере данными. Данные, собственно говоря, важнее, чем сервер. Поврежденный и вышедший из строя сервер можно заменить, а утраченные

данные восстановить невозможно, если нет системы резервного копирования и хранения данных. Современные серверы оснащаются оборудованием для эффективного хранения данных и создания их резервных копий. Причем резервное копирование данных может производиться как в моменты отсутствия нагрузки на сервер (в небольших банковских системах это делается после завершения банковского дня), так и прямо во время работы (так происходит в информационных системах непрерывного цикла). Производители серверных систем обязательно предлагают вместе с ними системы хранения, резервного копирования и восстановления данных.

- *Удаленное управление.* Реализуется отдельный канал связи с сервером (часто вне сети, по которой обеспечивается обслуживание клиентских машин) при встроенной в сервер системе управления и тестирования.
- *Масштабирование* — сервер должен иметь возможность увеличивать свою мощность путем наращивания вычислительных блоков.

Современные серверы с точки зрения форм-фактора реализуются в трех исполнениях:

- *Напольное* — это исполнение предполагает размещение сервера на полу и предназначено для выполнения заданий сравнительно небольшого объема в малых учреждениях или предприятиях.
- *Стойное* — сервер размещается в специальной стойке. Такое размещение предполагает выделенное специализированное помещение с поддержанием температурного режима.
- *Блейд-серверное (лезвийное)* — серверы выполняются в виде сверхтонких блоков, предназначенных для установки в специальный блейд корпус. Такая конструкция дает возможность сверхкомпактного размещения. Кроме того, блейд-корпус берет на себя функции контроля за напряжением сети, охлаждением, аварийным энергопитанием установленных в него блоков.

Современные серверы выполняются на основе серверных процессоров (некоторые из них приведены в табл. 12.4, при этом в одном корпусе сервера размещается от 2 до 8 процессоров).

Ведущими фирмами-поставщиками серверного оборудования являются: Sun Microsystems (SUN), Hewlett-Packard (HP), корпорация Dell и корпорация IBM (International Business Machines).

12.3.4. Персональные компьютеры

Если с суперкомпьютерами и серверами сталкиваются напрямую в основном специалисты, то с персональными компьютерами и рабочими станциями сегодня так или иначе работают практически все.

Персональные компьютеры имеют множество исполнений и вариантов. Выпущенное в 1998 г. руководство «PC 99 System Design Guide» подразделяло персональные компьютеры на следующие категории:

- *Consumer PC* — массовый персональный компьютер домашнего применения. Должен обеспечивать возможность выхода в Интернет, но не возможность работы в локальной сети.
- *Office PC* — массовый персональный компьютер офисного применения. Должен обеспечивать работу в локальной сети, выполнения большинства офисных операций, а также возможность удаленного управления и обслуживания.
- *Mobile PC* — переносной персональный компьютер. Компьютер малых размеров, который можно брать с собой в поездки.
- *Workstation PC* — мощный компьютер с возможностью работы в локальной сети и Интернете. Предназначен для решения специальных задач, например, вычислительных или графических.
- *Entertainment PC* — развлекательный компьютер с повышенным качеством воспроизведения мультимедийного содержимого. Обычно признаком Entertainment PC является мощная система обработки графической информации и дополнительные устройства управления.

По габаритам и весу персональные компьютеры можно разделить на классы:

- *Стационарный компьютер* — компьютер, предназначенный для установки на фиксированном рабочем месте. Обычно состоит из двух отдельных блоков: монитора и системного блока. По типу корпуса стационарные компьютеры бывают трех базовых конфигураций:
 - Tower — вертикальное расположение корпуса для напольной установки;
 - miniTower — вертикальное расположение корпуса для настольной установки;
 - Desktop — горизонтальное расположение корпуса для настольной установки.

В последнее время появились новые конструкции стационарных рабочих станций: устанавливаемые в стойку и в блейд-корпус. В этом случае устанавливается единый блок рабочих станций (возможно, в отдельном помещении) и связывается проложенными кабелями с мониторами и устройствами ввода пользователей.

- *Переносной компьютер* — компьютер, выполненный в виде компактного системного блока или моноблока. В первом случае это системный блок массой 6–8 кг, оснащенный рукояткой для переноски, во втором — очень большой ноутбук, массой до 10 кг. Такого рода компьютеры предназначены для перемещения с одного рабочего места на другое, но не для постоянного ношения с собой.

ПРИМЕЧАНИЕ

Оба предыдущих класса компьютеров собираются на стандартных комплектующих и микропроцессорах, примеры которых приведены в табл. 12.3. Когда компьютер собран в виде моноблока, он оснащается плоским жидкокристаллическим дисплеем. В случае отдельного блока дисплей может быть как электронным, так и жидкокристаллическим. Однако в последнее время электронно-лучевые дисплеи весьма интенсивно вытесняются жидкокристаллическими плоскими экранами, и лет через 5, видимо, электронно-лучевых дисплеев не станет вовсе.

- *Ноутбук* — компьютер, выполненный в виде единого блока, раскрывающийся наподобие блокнота. Масса такого компьютера от 2 до 4 кг. Такие компьютеры призваны постоянно сопровождать владельца. Их легко переносить (обычно ноутбуки продают вместе со специальной наплечной сумкой), они быстро подключаются к электропитанию и позволяют работать в тех местах, где стационарные или даже переносные компьютеры были бы не совсем уместны (например, в кафе или на природе). Ноутбуки обычно оснащаются аккумуляторными батареями повышенной емкости, обеспечивающими работу в режиме чтения или ввода текста в течение 2–5 часов. Это позволяет работать с ними там, где подключение к электросети невозможно в принципе (за городом, в общественном транспорте). Если еще в 2000 г. ноутбук считался признаком бизнес-класса, то сегодня, благодаря резкому удешевлению ноутбуков (в особенности с одноядерными процессорами) и специальным программам по созданию ноутбуков для студентов и школьников, ноутбук поистине стал студенческим компьютером. Ноутбуки создаются на тех же типах процессоров и той же элементной базе, что и стационарные или переносные компьютеры, но в специальном исполнении, предусматривающем уменьшение габаритов и энергопотребления.
- *Субноутбук* — специально миниатюризированный и облегченный ноутбук. Если размер ноутбука (напрямую зависящий от размера диагонали экрана) может составлять до полуметра (475 мм при диагонали экрана 20,1 дюйма), то субноутбук размерами похож на малоформатную книжку и имеет массу от 600 г до 1,3 кг. Субноутбуки, выпускающиеся специально для работы в Интернете, получили название *NetBook*.
- *Палладонный субноутбук* — еще более миниатюризированная версия ноутбука, который в раскрытом виде уменьшается на ладони при массе 300–400 г. Надо признать, что такого рода модели вряд ли получат широкое распространение, поскольку они неудобны в применении: маленький размер клавиатуры делает на них набор текста крайне утомительной задачей.

Все компьютеры, относимые к ноутбукам, обычно работают под управлением тех же операционных систем, что и стационарные компьютеры (Windows, Linux, Macintosh или UNIX).

- *Карманные персональные компьютеры (КПК), или «палладонники»,* — миниатюрные устройства с экраном размером 3–4 дюйма. Современные КПК функционально все больше приближаются к своим старшим братьям, ноутбукам. Однако КПК имеют принципиально иное назначение: они призваны выступать в роли персонального информационного менеджера (записная книжка, небольшая база данных, ежедневник, устройство для чтения). Поскольку КПК обычно не оснащаются физической клавиатурой, то для ввода текста в них используется экранная клавиатура и специальное устройство — стилус. Экран в КПК сенсорный, то есть, касаясь стилусом той или иной точки экрана, вы можете выполнять те же действия, что и наводя на эту точку указатель мыши и щелкая по кнопке. Такая конструкция КПК делает практически невозможным набор на нем больших объемов текста, но идеально подходит для чтения, игр, прослушивания музыки или подготовки небольших заметок. В настоящее время

некоторые конструкции КПК оснащаются сверхминиатюрными жесткими дисками, но основным носителем информации для этого класса компьютеров все же надо признать карты флэш-памяти. Причем тенденции роста скорости передачи данных и объема флэш-памяти таковы, что к 2015 г. прогнозируется вытеснение жестких дисков даже из ноутбуков, а уж КПК точно будут работать только с этим видом памяти.

- *Коммуникаторы* — КПК, оснащенные дополнительными средствами связи. Можно рассматривать коммуникаторы как объединение КПК, сети и телефонной связи. Обычно коммуникаторы не отличаются от КПК размерами и формой, но оснащены дополнительными разъемами для подключения телефонной гарнитуры, а в их конструкцию встроен мобильный телефон и средства мобильной сетевой связи Wi-Fi. Такого рода устройства все чаще функционально рассматриваются как мобильные рабочие станции, позволяющие находиться на связи независимо от физического местонахождения сотрудника. Коммуникаторы обычно оснащены всем необходимым для чтения файлов большинства офисных приложений. Таким образом, будучи в пути или в другой ситуации, когда использование ноутбука затруднительно, вы можете принимать почту, получать документы, просматривать их и вносить коррективы, а также выходить в Интернет (коммуникаторы обычно оснащаются мобильными версиями интернет-браузеров).

КПК и коммуникаторы собираются на специализированных комплектующих: микропроцессоры и другие компоненты для них не имеют ничего общего с «большими» компьютерами. Эти устройства оснащаются собственными операционными системами (Windows Mobile, Linux или Symbian), созданными специально для них.

- *Планшетные компьютеры* представляют собой нечто среднее между КПК и ноутбуком. Оснащенные большим сенсорным экраном, планшетные компьютеры обычно либо не имеют клавиатуры, либо не предусматривают ее использования в рабочем режиме. Их назначение хорошо отражено в названии — записи и зарисовки в режиме планшета.

Обычно планшетные компьютеры работают со специальными версиями операционных систем.

- *Электронные книги* — специальные электронные устройства с достаточно большим жидкокристаллическим экраном, плоские и легкие, назначение которых — чтение электронных книг. Несмотря на то что электронные книги обычно оснащены какой-либо операционной системой, они не являются компьютерами в обычном смысле слова, поскольку адаптированы для выполнения только одной функции.

12.3.5. Тенденции развития компьютеров

О том, что ждет компьютерные технологии в недалеком будущем, в этой главе сказано уже немало. По многим направлениям микроэлектроники человечество уже достигает технологических пределов, но одновременно, благодаря вычисли-

тельной мощи и технологическому уровню, достигнутому на сегодняшний день, происходит интенсивный поиск новых путей. Направления развития определяются, как и ранее, не только техническими и технологическими параметрами, но и социальными. К примеру, вряд ли кто-то из составлявших в 1998 г. руководство «PC 99 System Design Guide» мог себе представить, что такая категория компьютеров, как Consumer PC, просто исчезнет, поскольку сегодня просто невозможно себе представить домашний компьютер, не оснащенный самыми современными средствами мультимедиа. Мультимедиа-компьютер, игровой компьютер и интернет-компьютер в одной упаковке — вот что такое современный домашний компьютер. Во многих случаях домашний компьютер — это еще и узел домашней локальной сети.

Беспроводные технологии, еще недавно вызывавшие оживленные дискуссии, сегодня стали повседневной реальностью, и ноутбук, если в нем нет устройства для беспроводной связи Wi-Fi, выглядит неполноценным. То же самое можно сказать о настольных компьютерах. Компьютеры все более интегрируются в сети, становясь узлами глобальной, городской и домашней сетей.

Хорошим примером того, как для достижения новых высот в одной области применяются достижения в другой, служит адронный коллайдер, запущенный недавно в Швейцарии. Для обработки результатов от миллионов каналов данных с коллайдера вычислительных мощностей CERN (от франц. Conseil Européen pour la Recherche Nucléaire — Европейский совет по ядерным исследованиям) не хватило, поэтому в обработке будут участвовать множество компьютеров, объединенных в единое вычислительное пространство по технологии GRID.

GRID (от англ. решетка, сеть) — специальная технология объединения вычислительных ресурсов множества компьютеров для решения специальных задач: получения вычислительных мощностей в широком диапазоне по требованию или увеличения производительности каналов ввода и вывода информации.

Результаты, полученные от коллайдера, в свою очередь, помогут микротехнологиям преодолеть нано-барьер и от нанотехнологий перейти к куда более «миниатюрным» петатехнологиям. А этот переход поможет создавать принципиально новые по принципу действия микропроцессоры.

На самом деле оценка перспектив развития компьютерной техники и информатики сегодня — дело весьма неблагодарное: слишком широким фронтом идет наука, слишком быстры темпы. Тем не менее можно выделить важные направления, в которых это развитие происходит, попытавшись представить, как разовьется компьютерная технология через 10 лет:

- **Интеграция в сеть.** Все больше компьютеров подключается к Интернету, и, вероятно, в ближайшие 10 лет автономных компьютеров просто не останется. В свою очередь, это приведет к тому, что Интернет станет не только основным источником информации (заменив собой книги, радио и телевидение), но и основным каналом связи. Мало того, Интернет станет основным поставщиком программного обеспечения. Уже сейчас корпорация Google реализовала концепцию виртуального офиса. Вам не надо устанавливать на компьютер тексто-

вый редактор или электронную таблицу, достаточно зайти на сайт в Интернете и вызвать на экран нужную программу.

- *Обучение и образование* также в основном станут компьютерными, чему поможет массовое подключение компьютеров к сети. Уже сейчас дистанционные системы обучения позволяют получить диплом государственного образца. Сегодня многие школы оснащены электронными дневниками и журналами. Дистанционное образование, вероятно, не станет полной заменой классного школьного образования, поскольку самостоятельно, не задавая учителю вопросов, может учиться не каждый. Однако ситуация, когда классы будут виртуальными, вполне вероятна. Двусторонняя связь через веб-камеры и трансляция урока в режиме телеконференции, опрос, тестирование — все это технически возможно уже сегодня и частично реализовано в некоторых учебных заведениях. Во многих университетах США такие формы обучения, как дистанционные семинары, давно уже стали повседневностью.
- *Конструкция.* Сами компьютеры будут претерпевать все более радикальные изменения. Ноутбуки и плоские жидкокристаллические экраны вошли в нашу жизнь почти незаметно. Следующий этап развития компьютеров, по-видимому, будет куда более революционным. Сейчас уже есть промышленные образцы проецируемых клавиатур. Не нужно обладать очень уж богатой фантазией, чтобы представить развитие этих принципов до проецируемых мыши и экрана. Таким образом, с учетом миниатюризации элементов памяти и других элементов компьютера, лет через десять персональный компьютер, скорее всего, будет напоминать футляр для очков. После включения из корпуса спроецируются экран (вначале на поверхности, а затем, вероятно, и в пространстве), клавиатура и мышь. Встроенный микрофон будет воспринимать голосовые команды, микродинамики — воспроизводить звук. После окончания работы компьютер можно просто уложить в нагрудный карман.
- *Интеллектуальность.* Сегодня исследования, посвященные машинному самообучению, составляют немалую долю в общем объеме компьютерной науки, а в промышленном производстве уже появились первые образцы нейропроцессоров, способные на микроэлектронном уровне реализовывать топологию нейронной сети. Машинное обучение и быстрое развитие теоретических основ нейронных сетей и машин с нечеткой логикой позволят в скором будущем создавать компьютеры, которые будут узнавать своего хозяина и даже поддерживать беседы на определенные темы. «Умный дом», «умный автомобиль» — эти словосочетания давно на слуху, но мы плохо себе представляем, что произойдет, когда от ступени автоматизации компьютеры шагнут в область активного самообучения. Ведь, как уже отмечалось, в распоряжении этих компьютеров будет не только информация, вводимая владельцем, но и Интернет. Кажется, самые смелые прогнозы фантастов 70-х гг. могут стать реальностью.
- *Выход в пограничные области знаний.* О том, что компьютер — это не обязательно микроэлектронное устройство, мы уже говорили. Однако реальные шаги к реализации типов компьютеров, отличных от электронных, до недавнего времени были очень малы и несмелы. Только в последние годы новое дыхание

обрели исследования, направленные на создание оптических, биологических, квантовых, химических и других неэлектронных типов компьютеров. Пока рано говорить о каких-то устойчивых прогнозах. Только в области квантовых компьютеров есть более или менее завершённые эксперименты и работающие системы. Хотя с точки зрения производительности и размера эти компьютеры даже близко не могут сравниться с микроэлектронными, никто не может предсказать, какое ускорение этим технологиям могут придать научные факты, полученные во время экспериментов, скажем, на том же адронном коллайдере.

Таким образом, подводя итог, можно сказать, что компьютеры развиваются именно в том направлении, которое лучше всего разработано писателями-фантастами: они становятся вседущими, все менее заметными, все более интеллектуальными, все более включёнными как в глобальное информационное пространство (Интернет), так и в наш быт.

Вопросы для самопроверки

1. Дайте определение компьютеру.
2. Каковы основные этапы в развитии вычислительной техники?
3. Как выглядит машина Тьюринга? Была ли она изготовлена?
4. Как устроена машина фон Неймана?
5. Перечислите основные поколения компьютеров.
6. Каковы основные признаки четвертого поколения компьютеров?
7. Какие виды классификации компьютеров вы знаете?
8. Как компьютеры можно классифицировать по назначению?
9. Что такое «мэйнфрейм»?
10. Чем клиент-серверная архитектура отличается от мэйнфреймовой?
11. Что такое «сервер приложений»?
12. Сформулируйте закон Мура.
13. Что понимается под достижением технологического барьера?
14. Каковы тенденции и пути развития микропроцессоров?
15. Каким образом достигается такая большая вычислительная мощность суперкомпьютеров?
16. В чем состоит классификация Флинна?
17. Какие дополнительные возможности, помимо обработки запросов клиентских компьютеров, должен реализовывать современный сервер?
18. Назовите основные отличия ноутбука от ПК.
19. Каковы перспективы развития компьютеров и вычислительных систем?

Литература

1. *Воеводин Вл. В., Жуматий С. А.* Вычислительное дело и кластерные системы. М.: Издательство московского университета, 2007.
2. *Поворознюк А. И.* Архитектура компьютеров: учебное пособие. Харьков: Торнадо, 2004.
3. *Пятибратов А. П., Гудыло Л. П., Кириченко А. А.* Вычислительные системы, сети и телекоммуникации. М.: Финансы и статистика, 2004.
4. *Частикова А.* Архитекторы компьютерного мира. СПб.: БХВ-Петербург, 2002.
5. *Таненбаум Э., Ван Стеен М.* Распределенные системы. СПб.: Питер, 2003.
6. *Таненбаум Э.* Архитектура компьютера. СПб.: Питер, 2007.

Глава 13 Основы построения компьютерных сетей

- 13.1. История появления и развития компьютерных сетей
- 13.2. Развитие сетевых межкомпьютерных коммуникаций в России
- 13.3. Классификация компьютерных сетей
- 13.4. Топология компьютерных сетей
- 13.5. Физическая реализация среды передачи данных
- 13.6. Модели и протоколы компьютерных сетей
- 13.7. Феномен века коммуникаций

Если прошлый век можно было назвать веком космоса или веком ядерной энергии, то век нынешний с той же уверенностью можно охарактеризовать как век коммуникаций. Сегодня компьютерная сеть есть практически везде, она соединяет между собой различные устройства внутри предприятий, учебных заведений, городов и домов. Нередким явлением стали внутриквартирные сети, а глобальная сеть Интернет за четверть века превратилась в явление, с которым считаются целые государства и международные корпорации. Вычислительные системы в этих условиях все больше и больше приобретают распределенный характер и включают в себя множество компьютеров и других устройств, рассредоточенных по всему земному шару.

В этой главе речь пойдет о компьютерных сетях, о том, для чего и каким образом они используются; о физических явлениях и технических устройствах, при помощи которых создаются сети; о тех правилах и законах, на основании которых через сети передаются и распределяются данные; а также о новых социальных явлениях, которые принесла с собой «сетевая революция» в нашу жизнь.

13.1. История появления и развития компьютерных сетей

На начальном этапе развития компьютерной индустрии данные обрабатывались на мэйнфреймах: один мощный компьютер в многозадачном режиме с разделением времени обрабатывал данные, а ввод данных осуществлялся с терминалов, представляющих собой клавиатуру и дисплей. При этом в пределах одного здания или даже одного предприятия можно было соединить терминальные рабочие места с мэйнфреймом при помощи кабеля. Создавалась структура, напоминающая современную одноранговую локальную сеть, соединенную звездой (рис. 13.1).

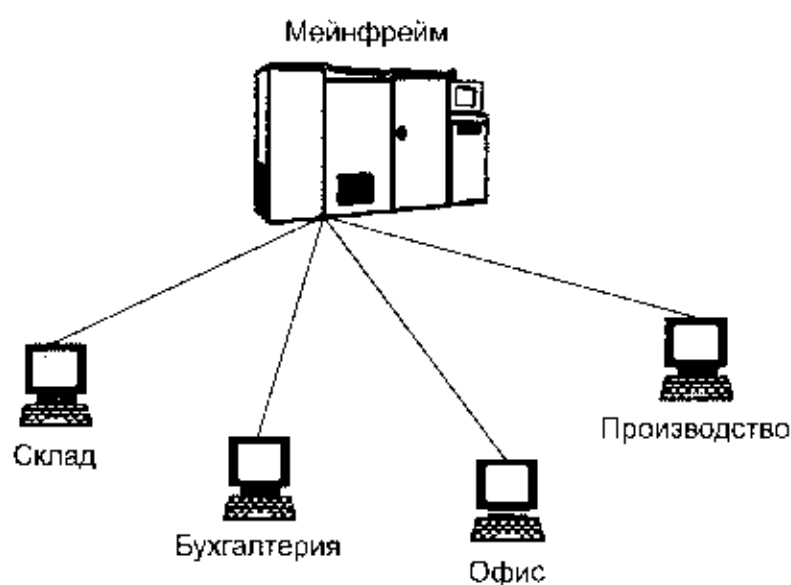


Рис. 13.1. Мейнфреймовая архитектура обработки данных

В этом случае по соединительному кабелю передавался весьма незначительный по сегодняшним меркам объем данных с очень невысокой скоростью. Несмотря на то что и в этом случае надо было вырабатывать какие-то правила кодирования цифровых данных для передачи их по медному кабелю, эти правила кодирования и сам физический процесс передачи не вызывали особых затруднений: можно было обходиться теми же правилами, по которым данные обрабатывались внутри компьютера, кодируя наличие или отсутствие высокого уровня сигнала на медной шине как ноль или единицу. В полной мере эти кабельные соединения между терминалами и мэйнфреймом еще не могли называться компьютерной сетью.

То, что годилось для промышленного предприятия, имеющего хотя и большую, но все-таки ограниченную площадь, оказалось непригодным для решения других задач. Компьютер стал активно использоваться в качестве хранилища и обработчика данных в таких областях, как продажа авиационных билетов, биржевая торговля, планирование и управление поставкой товара. Все эти очень разные сферы применения были сходны в одном: они требовали наличия терминальных рабочих мест по всей стране, в то время как мощный мэйнфрейм должен был оставаться единственным. Задача соединения мэйнфрейма с терминальными устройствами

была решена благодаря наличию сети телефонной связи, которая проложена в самые отдаленные уголки. Решать пришлось другую задачу: как передавать через аналоговые линии телефонной связи, специальным образом приспособленные для качественной передачи человеческого голоса, цифровые данные? Вполне логичным выглядело решение преобразовывать цифровые данные для передачи по телефонным линиям в колебания звуковой частоты, а затем вновь преобразовывать эти колебания в цифровые данные на принимающей стороне. Сам процесс преобразования цифровых данных в аналоговый сигнал носит название *модуляции*, а обратное преобразование — *демодуляции*. Устройство же, выполняющее эти операции, называется модулятор-демодулятор, или, сокращенно, *модем*.

Создание модемов и выработка технологии передачи данных по сетям телефонной связи заложило очень хорошую основу для будущего создания компьютерных сетей. Для последнего шага не хватало одного: компьютеров, между которыми должны были бы устанавливаться связи. Вычисления в терминальной сети нельзя было считать распределенными, поскольку все они выполнялись одним компьютером, да и сама сеть, как уже отмечалось, была не компьютерной, а терминальной.

Компьютерная сеть — это совокупность аппаратных средств, программного обеспечения и физической среды, обеспечивающая передачу данных между компьютерами.

Впервые задача создания именно компьютерных сетей в США была поставлена военными. Созданное в 1952 г. в СССР водородное оружие и запущенный в 1957 г. искусственный спутник земли поставили Соединенные Штаты перед неожиданным для них фактом: их территория стала доступна для удара советских ракет. В срочном порядке нужно было пересматривать всю концепцию информационной системы вооруженных сил, которая прежде строилась на мэйнфреймовой архитектуре. Дело в том, что в основе терминальной сети мэйнфреймовых информационных систем лежала сеть телефонной связи. В свою очередь, телефонная сеть имела иерархическую структуру, при которой все телефонные соединения сходились в крупных городах. Таким образом, вывод из строя одного лишь крупного телефонного узла мог оставить без оперативного управления половину территории страны. Требовалось решение, в котором хорошо защищенные компьютеры информационной системы министерства обороны США, во-первых, могли бы обмениваться данными напрямую, минуя терминальные сети, во-вторых, продолжали бы обмен данными даже при выходе из строя одного из центральных узлов связи или одного из этих компьютеров. Информационная система в целом должна была сохранять работоспособность и после отказа своих важных компонентов. Для этого нужен был принципиально иной подход как к работе самой вычислительной системы, так и к организации коммуникаций и передачи данных между компьютерами.

Еще в начале 50-х гг. прошлого века сотрудник корпорации RAND Пол Бэрл проводил по заказу министерства обороны США соответствующие изыскания и предложил структуру высоконадежной сети (рис. 13.2).

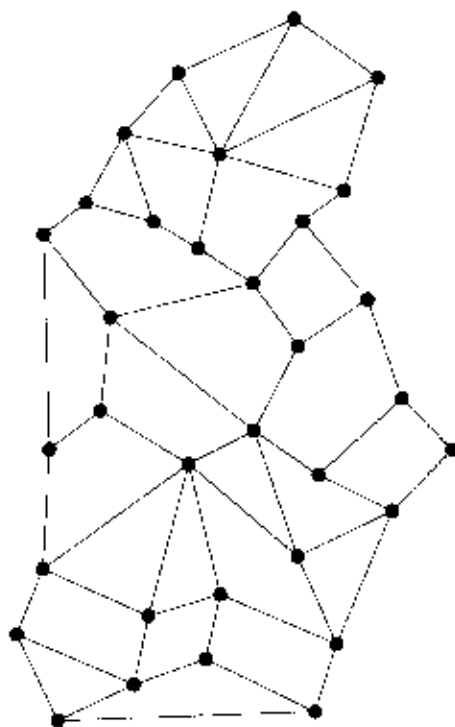


Рис. 13.2. Высоконадежная сеть Пола Бэрэна

В этой сети, благодаря продуманной, а вернее, просчитанной, топологии, при выходе из строя даже половины узлов оставшаяся сеть с высокой вероятностью продолжала работать как единое целое. При этом Бэрэн предложил использовать для передачи данных между узлами не непрерывный аналоговый сигнал, а пакеты, то есть делить данные на фрагменты небольшого объема, передача которых была более надежной и которые позволяли быстро повторить пересылку в случае неудачи. К сожалению, реализовывать идеи Бэрэна было поручено телефонной компании AT&T, которая отвергла эти принципы как нереализуемые.

Таким образом, к моменту, когда США остро нуждались в высоконадежной распределенной информационной системе, этой системы у них не было. Для быстрого решения проблемы были выделены деньги и создана единая научная организация ARPA (Advanced Research Projects Agency — управление по перспективным научным исследованиям). Результатом деятельности этой организации было несколько весьма важных решений.

- Во-первых, агентство ARPA вместо организации закрытых военных лабораторий предложило решать поставленные задачи университетам, выделив гранты на проведение исследований.
- Во-вторых, агентство реализовало идеи Бэрэна о пакетной передаче данных и распределенной сети, начав проектировать и строить национальную сеть, соединяющую исследовательские центры.
- В-третьих, благодаря щедрому финансированию ARPA, университеты получили возможность купить и установить в одном месте несколько компьютеров (в то время, когда персональных компьютеров еще не было, несколько компьютеров в одной организации были большой редкостью). Таким образом, каждый во-

влеченный университет, кроме работы в рамках проекта общенациональной сети, получил возможность проводить эксперименты в рамках университетской локальной вычислительной сети.

- В-четвертых, несмотря на пристальное внимание к развивающейся национальной компьютерной сети, агентство проводило исследования в области сетей, построенных на базе спутниковой и наземной радиосвязи. В результате этой работы уже в 1972 г. грузовик, переоборудованный по штату Калифорния и оснащенный терминалом, мог управлять компьютером, установленным в Лондоне.

Хорошее финансирование университетов и весомые денежные добавки от ARPA сделали свое дело. Сеть ARPANET, в которую поначалу были включены всего 4 университета (1969 г.), стала быстро расширяться, и к 1972 г. к ней подключились уже 34 университета, а сама сеть приобрела ярко выраженную смешанную топологию, в которой любой компьютер имел как минимум два пути передачи данных.

Однако во всем этом процессе развития сети не хватало того, что условно можно было бы назвать «резкостью» картинки. Если в сети ARPANET, объединяющей университеты, наблюдалось единообразие как программного обеспечения, так и технических решений (во всех узлах сети были установлены компьютеры одной марки, Honeywell DDP-316), то на уровне локальных сетей университетов царил настоящий «зоопарк» протоколов, которые каждый из университетов был вынужден изобретать для себя самостоятельно. Завершающим аккордом в процессе формирования сети ARPANET можно назвать создание в университете Беркли операционной системы UNIX 4.2 BSD. В состав этой операционной системы по заказу ARPA были включены сетевые разработки, успешно проявившие себя к этому времени: протоколы TCP/IP, сокеты и системные сетевые утилиты. Университеты получили унифицированное и готовое к применению программное обеспечение, при помощи которого можно было не только строить локальные сети, но и легко подключаться к сети ARPANET. С этого момента сеть ARPANET вступила в эпоху своего лавинообразного роста.

Бурный рост и безусловный положительный эффект, который оказала сеть ARPANET на научные исследования в университетах, вовлеченных в проект, привлекло внимание Национального научного фонда (National Science Foundation, NSF) США. Отмечая, что возможность практически мгновенно обмениваться результатами исследований многократно ускоряет ход научных исследований, фонд NSF выявил то, что сдерживало развитие сети ARPANET: для подключения к ней любой университет должен был заключать специальный договор с министерством обороны, причем министерство обороны не стремилось к дальнейшему расширению своего сотрудничества с университетами. Такое положение дел привело к решению о создании сети NSFNET, которая стала альтернативой ARPANET и к которой могли подключиться любые университеты. Первоначальное финансирование проекта по созданию NSFNET было внушительным. Была построена сетевая магистраль, связавшая 6 университетских центров, оснащенных суперкомпьютерами, и создано 20 региональных сетей, соединенных с этой магистралью. В результате практически все университеты США получили возможность доступа к суперкомпьютерам базовых научных центров. Нужно отметить также факт при-

ления в 1983 г. протоколов TCP/IP в качестве официальных сетевых стандартов. Массовый рост количества локальных сетей, подключенных к ARPANET и NSFNET, продолжался, сопровождаясь серьезным давлением бизнес-сообщества, которое давно уже поняло выгоду и прибыльность новой технологии.

В 1990 г. сеть NSFNET была переведена на коммерческие рельсы. Этот перевод облегчил участие бизнеса в дальнейшем развитии компьютерных коммуникаций. Появление персональных компьютеров ускорило процесс роста компьютерных сетей, к которым стали присоединяться не только научные центры, но и дома обычных граждан. Создание на Европейском континенте своих региональных сетей и объединение их с американскими сетями ознаменовало собой эпоху наступления Интернета, сети, объединяющей не столько компьютеры, сколько другие сети. Таким образом, Интернет — это не глобальная сеть, а сеть сетей.

13.2. Развитие сетевых межкомпьютерных коммуникаций в России

Развитие межкомпьютерных коммуникаций и сетей в России тесно связано с появлением в России зарубежных (главным образом американских) технологий и может обсуждаться в основном с исторической, но никак не с технологической точки зрения. Вся технология пришла к нам из США.

Если же говорить об истории, то первое в Советском Союзе подключение к Интернету было реализовано на деньги американской организации «Ассоциация за прогрессивные коммуникации» и осуществлялось по трем модемным линиям, к которым на основе коммутируемых соединений подключалось 50 пользователей. Так была создана общественная организация Гласнет. В 1993 г. эта общественная организация превратилась в коммерческого интернет-провайдера. Вся дальнейшая история межкомпьютерных коммуникаций СССР и России практически полностью основана на американских технологиях. Сеть Релком, создававшаяся как региональная научная сеть, технологически также полностью была основана на уже существующих решениях, стандартах и компьютерах американских моделей (PC-286, затем MicroVAX 3). Это вполне понятный выбор, поскольку в то время мы не могли быть законодателями компьютерной моды, и России оставалось только обеспечивать возможность подключения к уже сформированному в мире глобальным сетям.

На сегодняшний день Россия, по-прежнему основываясь в основном на зарубежных технологиях, обладает развитой сетевой инфраструктурой, позволяющей ей не только полноценно участвовать в жизни глобальной сети сетей (Интернета), но и создавать весьма сложные решения, основанные на распределенных вычислениях. Благодаря неоднократным усилиям и решениям руководства страны в области компьютеризации и интернетизации образования, компьютеры есть в большинстве учебных заведений страны, многие школы и практически все вузы подключены к Интернету. Локальные сети присутствуют повсеместно: в супермаркетах и небольших магазинах, в частных фирмах и государственных учреждениях, в вузах

и детских садах. Уже в прошлом то время, когда от сотрудников не требовалось владение компьютером даже на уровне пользователя, сегодня неумение работать с компьютером приравнено к неумению писать или читать.

13.3. Классификация компьютерных сетей

Компьютерные сети можно классифицировать множеством способов по самым различным признакам (рис. 13.3).

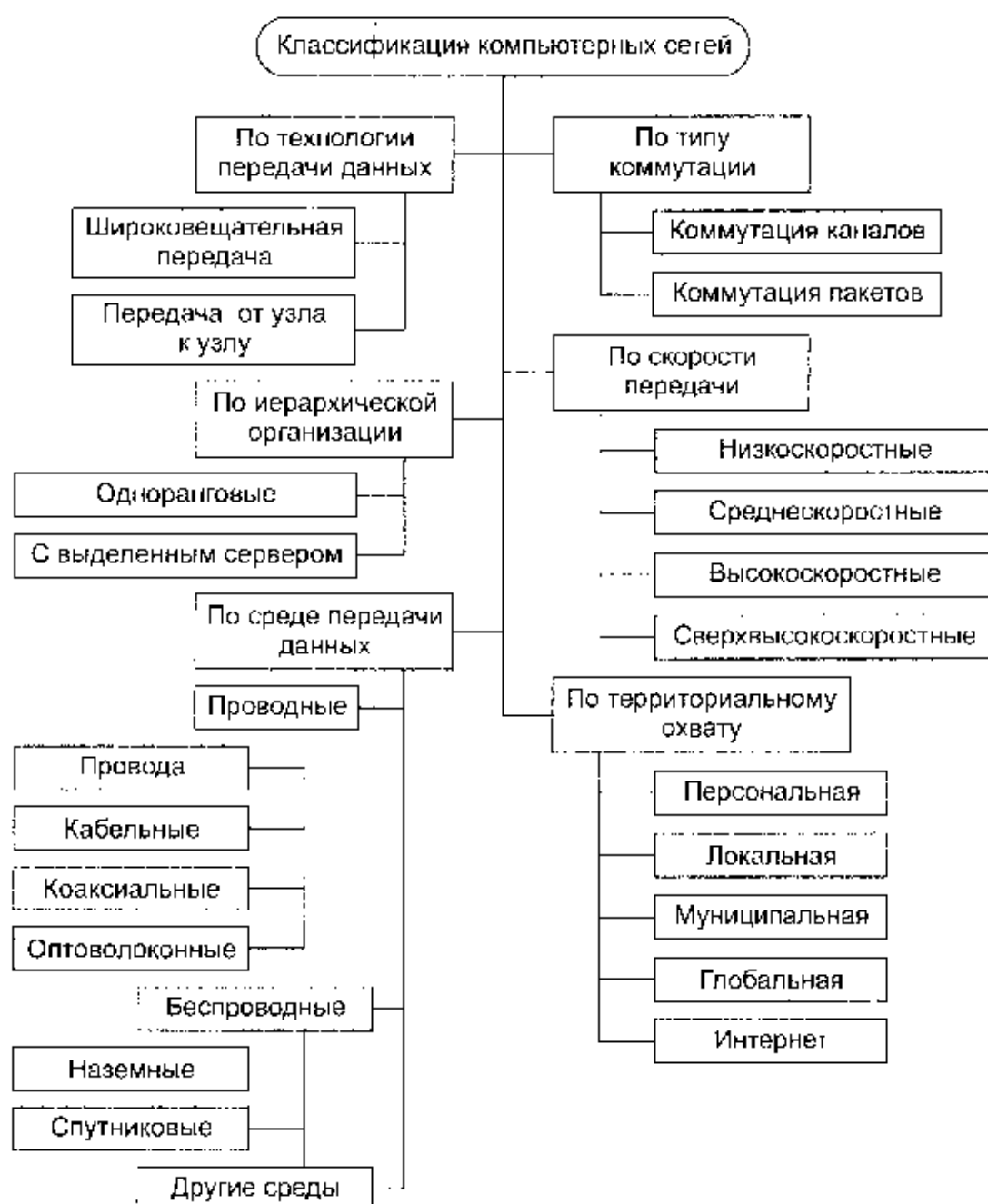


Рис. 13.3. Классификация компьютерных сетей

13.3.1. Классификация по технологии передачи данных

По технологии передачи данных различают широковещательную передачу (broadcasting) и передачу от узла к узлу (peer-to-peer).

Широковещательная передача

В технологии *широковещательной передачи* сообщение, отправленное одним компьютером в сеть, «слышат» все остальные компьютеры. При этом, поскольку в сообщении указано, кто является адресатом, принимает его только компьютер, для которого оно предназначено, остальные же компьютеры игнорируют это сообщение.

Передача от узла к узлу

Передача от узла к узлу, или *двухточечная передача*, подразумевает наличие изолированного канала между двумя компьютерами. В этом случае отпадает необходимость в каждом сообщении оповещать, для какого именно компьютера оно предназначено. Сам канал является идентификатором адресата. Вполне естественно, что аналогом двухточечной связи в нашей жизни являются телефонные переговоры между двумя людьми.

13.3.2. Классификация по типу коммутации между узлами

Когда речь идет о коммутации, имеется в виду физический и логический уровни, на которых происходит соединение двух узлов сети между собой, а также технология, используемая для передачи данных между этими двумя узлами.

Физический уровень определяет способ соединения узлов, относящийся к аппаратному обеспечению (модем и телефонная сеть, сетевая карта и кабель или радиоволны).

Логический уровень определяет взаимодействие физически соединенных узлов согласно определенному порядку обмена сигналами.

По типу коммутации между узлами различают сети с коммутацией каналов и пакетов.

Сеть с коммутацией каналов

При организации *сети с коммутацией каналов* на время передачи данных между двумя узлами образуется цепь взаимосвязанных последовательных отрезков пути передачи данных, которая образует канал. Этот канал представляет собой физическое соединение между взаимодействующей парой узлов, которое не может быть использовано в этот момент другими узлами для передачи своих данных. Естественным примером сети с коммутацией физических каналов является сеть, которая образуется при подключении к провайдеру Интернета по телефонной

линии. Нужно учитывать тот факт, что сетью с коммутацией каналов при этом будет только фрагмент канала от компьютера клиента до компьютера провайдера. Однако коммутация каналов может происходить и на логическом уровне, когда для данных, передаваемых между двумя компьютерами в сети с множеством вариантов маршрута, прокладывается один маршрут и на время установления соединения образуется логический (виртуальный) канал с зарезервированной и гарантированной скоростью прохождения данных.

Сеть с коммутацией пакетов

В случае *сети с коммутацией пакетов* логическая единица данных (например, файл), пересылаемая между двумя компьютерами в сети, разделяется на небольшие фрагменты, получившие название пакетов. Между пунктом отправки пакета и пунктом его приема может быть расположено множество узлов (примерно так, как показано на рис. 13.2). При этом пакеты могут передаваться от компьютера-отправителя к компьютеру-получателю разными путями. Пакеты, отправленные позже, могут быть приняты раньше. Пакеты, при пересылке которых произошла ошибка, могут быть посланы повторно. Передача заканчивается тем, что на компьютере-получателе накапливаются все пакеты и из них собирается передаваемая логическая единица данных. Благодаря своим бесспорным преимуществам с точки зрения оптимальной эксплуатации пропускной способности сети в целом пакетная коммутация в современных сетях является доминирующей.

13.3.3. Классификация по среде передачи данных

По среде передачи данных различают проводные и беспроводные сети.

Проводные сети

Проводные соединения компьютеров реализуются при помощи медного провода. Такого рода соединения появились на заре компьютерных сетей, когда скорость передачи данных была невысока, а каналы, связывающие два компьютера, не содержали промежуточных узлов и устройств. Сегодня происходит возврат к идее использования проводных сред передачи данных на новой технологической ступени. Любая развитая страна уже имеет проводные соединения, связывающие между собой практически все населенные пункты. Это — единая энергосистема страны. Провода уже проложены, остается только приспособить их к передаче компьютерных данных.

Кабельные соединения реализуются при помощи групп проводов, объединенных общей оболочкой. Обычно для коммутации в компьютерных сетях применяют кабель типа «витая пара», в котором каждая пара проводников перекручена между собой с небольшим шагом.

Кабельные коаксиальные соединения реализуются при помощи одножильного кабеля с экранирующей оплеткой, предназначенного для передачи переменного электрического тока высокой (радио) частоты.

Оптоволоконные соединения реализуются при помощи кабеля, проводящего не электрический ток, а световой луч. В центре такого кабеля лежит специальный материал, называемый оптоволоком.

Беспроводные сети

Радиочастотные наземные каналы представлены каналами наземной мобильной связи, технологиями *Wi-Fi* и *Bluetooth*. Данные в этом случае передаются электромагнитными волнами очень высокой частоты (в миллиметровом диапазоне), и расстояние передачи данных весьма невелико, от десятков метров до расстояния прямой видимости.

В случае *радиочастотных спутниковых каналов* данные передаются через искусственные спутники земли. При этом расстояние, на которое могут передаваться данные, может быть очень большим (с континента на континент), но прием и передача данных требует специального, пока еще довольно громоздкого оборудования (направленные спутниковые антенны, так называемые «тарелки»).

Возможности других сред

Помимо рассмотренных сред передачи данных в компьютерных сетях, теоретически существуют и другие. Так, можно представить себе передачу цифровых данных по сфокусированному световому лучу (например, лазерному) в вакууме. Тот факт, что данный способ сегодня не используется (или используется крайне редко), не значит, что он не будет востребован завтра. Никто не пытался задействовать для передачи информации гравитационное поле, для этого сегодня просто не хватает соответствующих знаний. Однако наука развивается, и, возможно, уже через несколько лет будут использоваться такие среды передачи данных, которые окажутся в десятки и сотни раз производительнее нынешних.

13.3.4. Классификация по территориальному охвату

Классификация компьютерных сетей по их размеру приведена в табл. 13.1.

Таблица 13.1. Классификация компьютерных сетей по размеру

Порядок расстояния	Расположение	Класс
1 м	Территория возле одного человека	Персональная
10 м	Комната (квартира)	Локальная
100 м	Здание	Локальная
1 км	Район города	Локальная
10 км	Город	Муниципальная
100 км	Страна	Глобальная
1000 км	Континент	Глобальная
10 000 км и более	Планета	Интернет

Персональная сеть создается в случае, когда несколько компьютерных устройств находятся в персональном пользовании одного человека, причем требуется передавать данные с одного устройства на другое (с ноутбука или планшета на настольный стационарный компьютер, с мобильного телефона на настольный компьютер, с коммуникатора на беспроводную гарнитуру).

Локальная сеть (Local Area Network, LAN) в зависимости от назначения и расположения может быть и предельно простой, и очень сложной. Тем не менее ее отличительным признаком можно считать то, что она обычно не требует сложного телекоммуникационного оборудования. Кабель прокладывается при необходимости подключения к сети того или иного абонента, и структура сети в основном определяется размещением абонентов. Локальная сеть учреждения, находящегося в одном здании, может быть по геометрическим размерам меньше, чем локальная сеть студенческого городка, но при этом иметь более сложную структуру по принципам, определяемым бизнес-логикой.

Муниципальная сеть (Metropolitan Area Network, MAN) прокладывается в масштабах города. В зависимости от размеров города и уровня технической оснащенной такая сеть может создаваться по мере необходимости (в малых городах) или планироваться заранее и даже закладываться в строительные планы зданий и муниципальных коммуникаций. Отличается от локальной сети высокой степенью вовлечения телекоммуникационного оборудования и оптоволоконной связи, а также высокой степенью интеграции с поставщиками телекоммуникационных услуг (кабельное телевидение, мобильная связь, телефонная связь).

Глобальные сети (Wide Area Networking, WAN) планируются и строятся в масштабах стран или регионов. Их признаком являются наличие высоко- и сверхвысокоскоростных *магистралей* (backbone) передачи данных и узлов связи, объединяющих локальные и муниципальные сети и создающих распределенные маршруты прохождения пакетов.

Интернет объединяет основные магистрали глобальных сетей, обеспечивая передачу данных из одной глобальной сети в другую и тем самым создавая единое информационное пространство для всего населения планеты.

Надо отметить, что данная классификация достаточно условна. В некоторых случаях принято делить сети всего на три: локальные, глобальные и Интернет.

13.3.5. Классификация по скорости передачи данных

По скорости передачи данных различают сети:

- ❑ *низкоскоростные* (до 10 Мбит/с);
- ❑ *среднескоростные* (до 100 Мбит/с);
- ❑ *высокоскоростные* (до 1 Гбит/с);
- ❑ *сверхвысокоскоростные* (до 10 Гбит/с).

13.3.6. Классификация по иерархической организации

По иерархической организации различают одноранговые сети и сети с выделенным сервером.

В *одноранговой сети* все компьютеры являются равноправными, то есть имеют одинаковый ранг.

В *сети с выделенным сервером* один или более компьютеров выполняют дополнительные функции по предоставлению услуг остальным компьютерам сети. Это могут быть услуги по хранению и выдаче файлов (файловый сервер), распространению электронной почты (почтовый сервер) и другие. Если сервер только хранит файлы и переправляет их клиентам, а клиенты полностью осуществляют обработку данных, то такую конфигурацию принято называть файл-серверной. Второй тип конфигурации — клиент-серверная. В этом случае обработка данных ведется и на клиентской, и на серверной частях (выборка данных и некоторые расчеты). В клиент-серверной архитектуре работает большинство современных серверов управления базами данных.

13.4. Топология компьютерных сетей

Топологией сети обычно называют схему сетевых соединений на физическом уровне, определяющую, как компьютеры соединяются один с другим и по каким маршрутам могут передаваться данные. *По топологии* компьютерные сети можно разделить следующим образом: общая шина, звезда, дерево, кольцо, смешанная топология. Есть еще один тип топологии, который применяется крайне редко, в случаях, когда от сети требуется исключительная надежность. Эта топология называется полносвязной.

13.4.1. Полносвязная топология

В случае *полносвязной топологии* каждый компьютер сети связан с каждым компьютером отдельным дуплексным (двусторонним) физическим каналом связи (рис. 13.4).

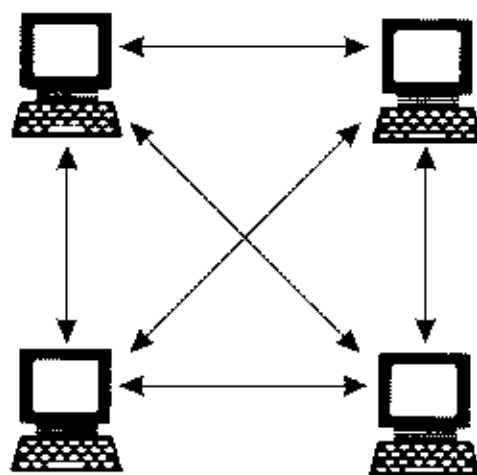


Рис. 13.4. Пример полносвязной топологии

Обратите внимание, что при всей простоте принципа построения такой сети даже на рисунке видно, как много связей приходится устанавливать между компьютерами, причем добавление каждого компьютера в полносвязную сеть увеличивает количество связей на число, равное числу компьютеров в сети. Если представить себе организацию с числом компьютеров 100, при условии, что компьютеры расположены в разных помещениях на разных этажах, то даже просто начертить полносвязную топологию на листе бумаги уже будет трудно, а уж реализовать ее просто невозможно.

По этой причине сеть с полносвязной топологией является скорее теоретической моделью, из которой можно получить все другие топологии сети путем отбрасывания связей.

13.4.2. Общая шина

В топологии с *общей шиной* между компьютерами прокладывается кабель, который является общей для всех компьютеров шиной передачи данных. Все компьютеры сети подключаются к этой шине (рис. 13.5).

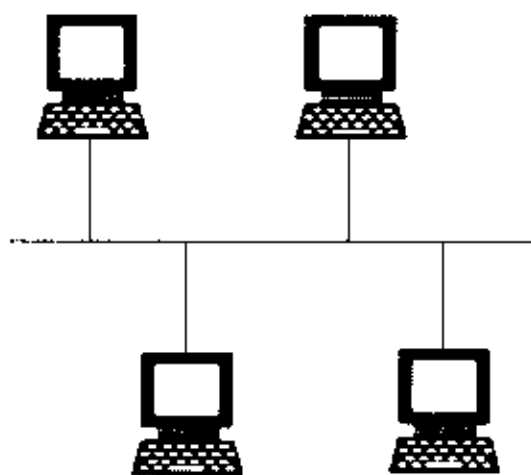


Рис. 13.5. Топология с общей шиной

В качестве шины выступает обычно коаксиальный кабель, отрезками которого через специальные разъемы соединяются компьютеры, или электромагнитный сигнал радиочастоты (например, в технологии Wi-Fi).

13.4.3. Звезда

В *звездообразной* топологии каждый компьютер подключается при помощи отдельного кабеля к общему устройству, называемому *концентратором*, или *хабом*. В качестве кабеля в этом случае может быть использована как витая пара, так и коаксиальный либо оптоволоконный кабель. В качестве концентратора может выступать как специальное устройство, так и еще один компьютер.

Преимуществом звездообразной топологии является то, что при выходе из строя одного из компьютеров или повреждении отдельного кабеля вся сеть про-

должна функционировать, поскольку компьютеры полностью автономны друг от друга.

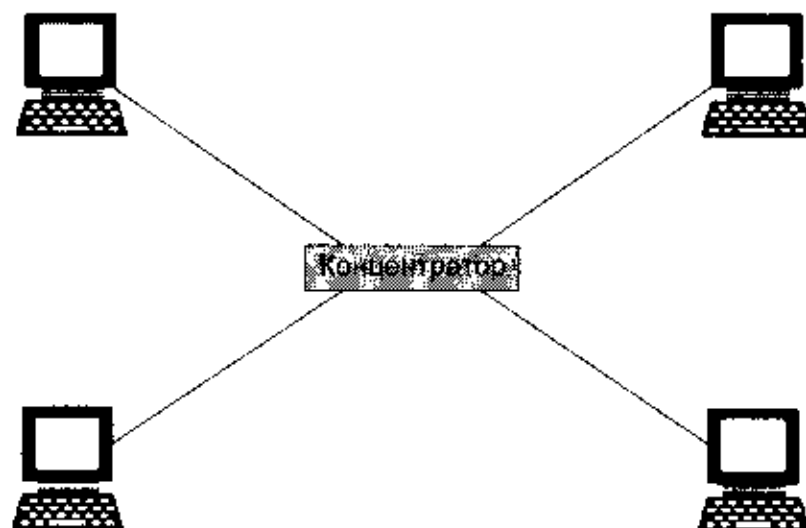


Рис. 13.6. Звездообразная топология

13.4.4. Кольцо

В *кольцевой* топологии компьютеры объединяются между собой круговой связью (рис. 13.7). При этом каждый компьютер связывается с последующим отдельным кабелем (нет общей шины). Это значит, что на каждом из компьютеров должно быть два сетевых устройства: для связи с предыдущим компьютером и с последующим. Кроме наличия дополнительного сетевого устройства в каждом компьютере, в кольцевой топологии есть еще один недостаток: выход из строя одного компьютера разрушает всю сеть. Этот недостаток кольцевой топологии компенсируется тем, что каждый компьютер в сети служит повторителем сигнала (данных) предыдущего компьютера, что повышает надежность передачи данных.

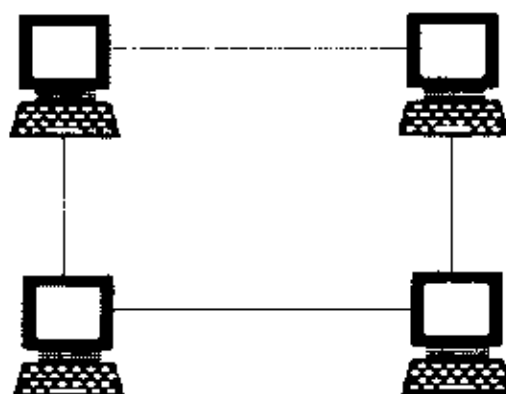


Рис. 13.7. Кольцевая топология

13.4.5. Дерево

Древовидная, или *иерархическая*, топология получается при объединении концентраторов нескольких звезд в иерархическом порядке. При этом возника-

ет древовидная структура с одним путем передачи для каждого из компьютеров (рис. 13.8).

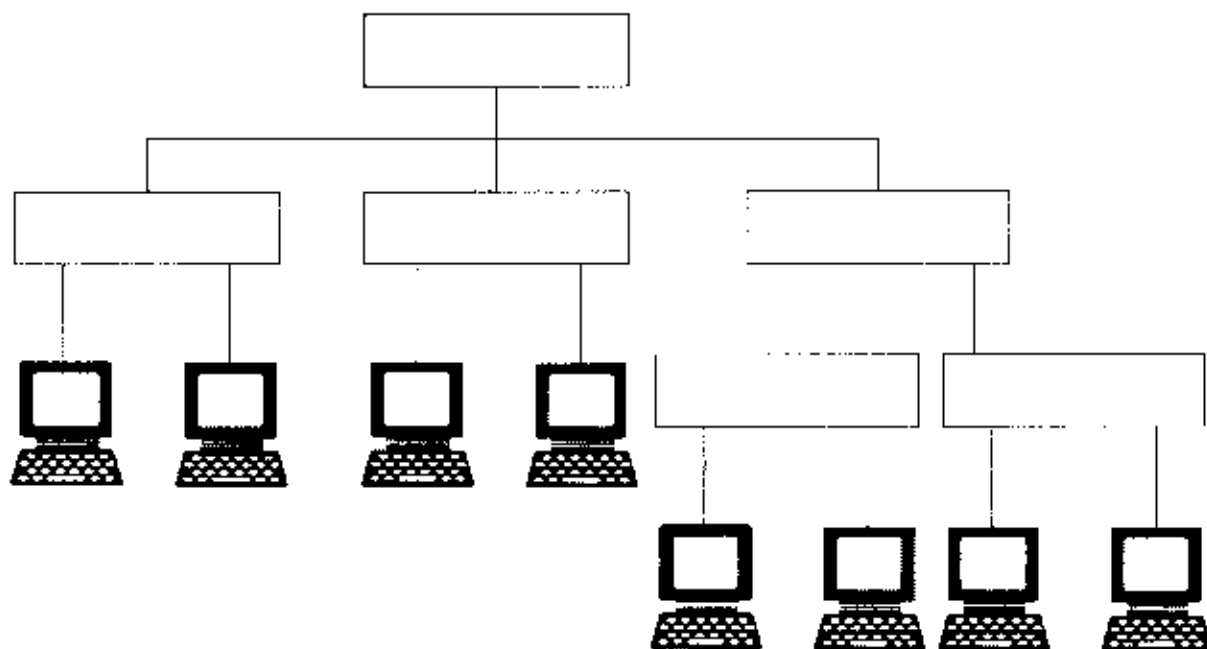


Рис. 13.8. Древовидная топология

13.4.6. Смешанная топология

Смешанная топология обычно возникает при объединении различных топологий, поэтому большие сети обычно строятся на основе смешанной топологии (рис. 13.9).

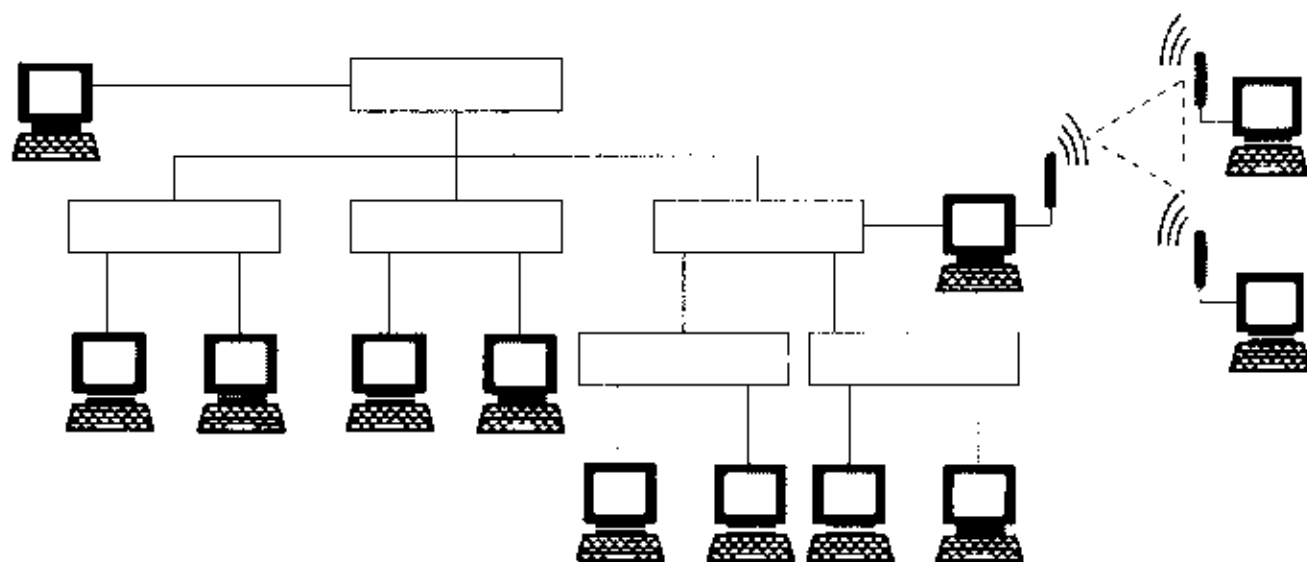


Рис. 13.9. Смешанная топология

Исторически компьютерные сети развивались достаточно хаотично, при этом приходилось решать возникающие тактические задачи. Не было генерального плана, в котором после длительных научных изысканий был бы выбран один и только

один правильный метод и одна правильная технология. Напротив, множество лабораторий и университетов искали решение одних и тех же задач и находили их. Естественно, решения оказались разными, в результате на сегодняшний день мы имеем большое количество вариантов (каждый со своими достоинствами и недостатками) организации компьютерных сетей.

Когда принимается решение о том, какова будет топология той или иной компьютерной сети, приходится учитывать множество факторов.

- На каком расстоянии друг от друга будут находиться компьютеры?
- Нужно ли разделять компьютеры на строго изолированные группы?
- Какова должна быть «ширина» (пропускная способность) каналов связи между различными участками (узлами) сети?
- На каких участках нужно обеспечить наибольшую безопасность?
- Насколько важен уровень помехозащищенности как самой сети, так и других устройств от помех, создаваемых сетью?

И, конечно же, есть еще такой немаловажный вопрос, как стоимость.

Решение всех этих вопросов взаимосвязано, и в процесс выбора топологии всегда вовлекается физический уровень:

- Через какую среду будут передаваться данные?
- Каким типом кабеля будут связаны компьютеры?
- Какие типы разъемов и сетевых адаптеров будут установлены на клиентские компьютеры?
- Сколько промежуточных устройств и какого типа будут находиться в узлах сети?
- Какова реальная конфигурация конструкций зданий или поверхности?

Выбор топологии неразрывно связан с выбором среды передачи и технологии физического уровня.

13.5. Физическая реализация среды передачи данных

13.5.1. Разделяемые среды передачи

К разделяемым средам передачи данных относятся физические среды, через которые осуществляется передача данных. Это провода, кабели, радиоволны, лазерные лучи или другие среды. Поскольку сегодня подавляющее большинство компьютерных сетей в качестве физической среды передачи использует кабели, остановимся более подробно на топологии кабельных компонентов.

Витая пара

Витая пара получила свое название оттого, что внутри такого кабеля каждая пара проводов скручена с большим шагом (причем с разным). Подобная конструк-

ция кабеля позволяет обеспечить хорошую помехозащищенность. Стандартный кабель содержит в себе 4 витые пары (то есть 8 проводников), что соответствует количеству контактов в разъеме 8P8C, используемом для подключения кабеля к сетевому адаптеру. Есть несколько категорий и типов кабеля типа «витая пара», отличающихся друг от друга максимальной пропускной способностью.

Кабели типа «витая пара» могут быть экранированными или неэкранированными. Лучшей помехозащищенностью обладает экранированный кабель. Чаще всего витая пара применяется при построении звездообразной топологии.

На рис. 13.10 изображены два типа кабеля, простая витая пара (слева) и экранированная витая пара (справа).



Рис. 13.10. Витая пара

Коаксиальный кабель

Коаксиальный кабель представляет собой кабель с медной жилой в центре в толстой изоляционной оболочке с экранирующей оплеткой. Изначально кабели такого рода были предназначены для передачи сигналов радиочастоты (таким, например, является кабель телевизионной антенны). Кабель, применяемый в компьютерных сетях, внешне напоминает телевизионный, но его технологические параметры (например, волновое сопротивление) другие, поэтому эти два вида кабеля не являются взаимозаменяемыми.

Для построения компьютерных сетей применяют два типа коаксиального кабеля: толстый и тонкий.

Толстый кабель (thick) имеет диаметр около 12 мм и является многослойным. Его составляют центральная жила, изоляция, оплетка, снова изоляция, вторая оплетка, оболочка. Недостатки толстого кабеля: он дорогостоящий, тяжелый, плохо изгибается, трудно монтируется. В качестве достоинств можно указать возможность организовать передачу данных на расстояние до 500 метров. Схематическое изображение толстого коаксиального кабеля представлено на рис. 13.11.

Тонкий кабель (thin) — имеет диаметр около 6 мм, центральную жилу, изолятор, оплетку и оболочку. К его недостаткам относится то, что расстояние, на которое

можно передавать данные, меньше, чем у толстого кабеля (до 185 м). Достоинства: меньшая стоимость, легкость, гибкость, удобство в прокладке, монтаже и подключении. Схематическое изображение тонкого коаксиального кабеля представлено на рис. 13.12.

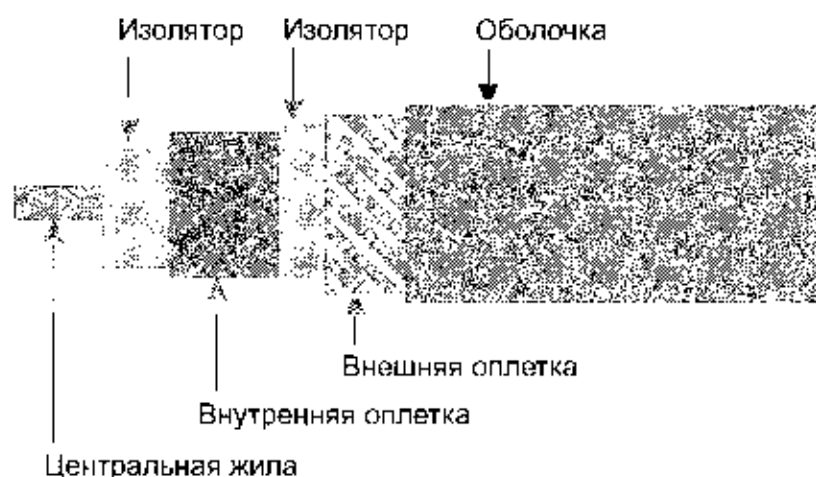


Рис. 13.11. Толстый коаксиальный кабель



Рис. 13.12. Тонкий коаксиальный кабель

Коаксиальный кабель служит в основном для организации сетей с топологией обшей шины. Для подключения компьютеров к шине, образуемой коаксиальным кабелем, используются специальные разъемы (BNC-коннекторы).

Оптоволоконный кабель

Передача данных через *оптоволоконный кабель* происходит посредством световых импульсов. Принцип действия оптоволоконного кабеля основан на способности света распространяться в некоторых средах, практически не теряя интенсивности, если в среде отсутствует преломление, а есть только отражение (то есть нет выхода сигнала за границы среды). В зависимости от того, каким образом распространяется свет, оптоволоконные кабели разделяют на три типа:

- *Многомодовый кабель с постоянной плотностью* — в таком кабеле свет проходит по разным путям отражения (модам) и за счет резкого излома при отражении подвергается сильному рассеянию (дисперсии) на выходе оптоволоконного канала.

- *Многомодовый кабель с переменной плотностью* — в этом кабеле свет также отражается от внутренней поверхности и проходит разными путями, но за счет переменной плотности сердечника, чем дальше от центра идет свет, тем быстрее он распространяется, поэтому дисперсия в нем значительно меньше.
- *Одномодовый кабель* — в этом кабеле толщина сердечника сравнима с длиной световой волны, в результате у света есть только один путь распространения и дисперсия практически отсутствует.

Отсутствие дисперсии сказывается на качестве передачи сигнала. Многомодовые кабели могут передавать данные со скоростью до 1 Гбит/с на расстояние до 2000 м. Одномодовый кабель способен передать данные со скоростью свыше 10 Гбит/с на расстояние до 20 км.

Пропускная способность и помехозащищенность могли бы сделать оптоволоконную технологию самой распространенной. Однако широкое использование оптоволоконной технологии передачи данных затрудняется следующими факторами:

- трудностью монтажа;
- высокой стоимостью кабеля и монтажных приспособлений;
- высокой стоимостью преобразователей цифрового сигнала в световые импульсы и обратно.

13.5.2. Сетевые адаптеры

Между компьютером и кабелем с разъемом должно быть еще одно устройство, которое преобразует сигналы среды передачи в компьютерные данные. Это устройство называют *сетевым адаптером*, или *сетевой картой*. Сетевая карта — это устройство, которое вставляется в слот материнской платы компьютера и имеет разъем для подключения витой пары или коаксиального кабеля. Некоторые сетевые карты оснащены разъемами обоих типов. Современные переносные персональные компьютеры обычно имеют встроенную сетевую карту с разъемом под витую пару. Кроме того, в последнее время широкое распространение получили внешние сетевые адаптеры, подключаемые к компьютеру через высокоскоростную шину USB2.

На рис. 13.13 изображены два сетевых адаптера, верхний с разъемом для подключения витой пары, нижний — коаксиального кабеля (разъем BNC).



Рис. 13.13. Сетевые адаптеры

13.5.3. Концентраторы

Как уже отмечалось, наиболее распространенная сегодня в локальных сетях звездообразная топология реализуется при помощи устройств, называемых *сетевыми концентраторами*, или *хабами* (hubs).

Изображенный на рис. 13.14 хаб имеет 16 портов, а это значит, что к нему можно подключить 16 компьютеров или 15 компьютеров и один *восходящий канал* (up-link), то есть канал связи с хабом верхнего уровня.

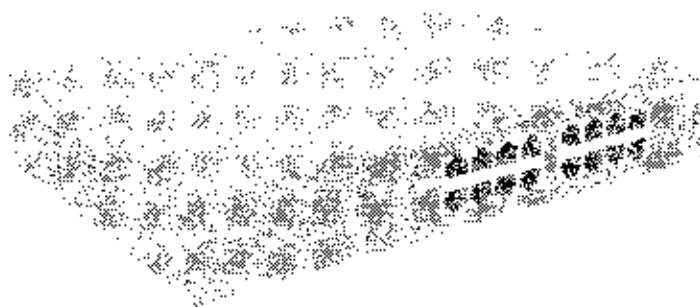


Рис. 13.14. Сетевой концентратор

Помимо хабов (концентраторов) для построения сетей применяются сходные по назначению устройства, называемые *коммутаторами*, или *свичами* (switch). Отличием коммутатора от концентратора является то, что коммутатор запоминает адреса компьютеров, подключенных к его портам, и данные отправляются не широковещательно, всем компьютерам в сети, а конкретному адресату. Такой режим значительно улучшает работу сети и снижает число коллизий, когда несколько компьютеров пытаются одновременно передать свои данные.

13.5.4. Маршрутизаторы

В отличие от хабов, *маршрутизаторы*, или *роутеры* (router), предназначены не для объединения компьютеров локальной сети, а для соединения сетей между собой. При этом соединяемые сети могут различаться не только назначением (локальная — глобальная), но и технологиями. В задачи маршрутизатора входит корректное преобразование данных из одного формата в другой, а также отправка их в нужные сегменты сети.

13.6. Модели и протоколы компьютерных сетей

13.6.1. Общее представление

Протокол в общем смысле представляет собой правила поведения, известные обеим взаимодействующим сторонам. То же самое представляют собой сетевые протоколы: правила поведения, известные обеим взаимодействующим сторонам. Что, в какой момент, в ответ на какое сообщение нужно сделать, какие данные

отослать, когда начать передачу, когда ее прекратить, когда перейти в режим ожидания — все это определяет протокол.

Протокол (сетевой) — это формализованные правила, определяющие порядок и формат сообщений, которыми обмениваются сетевые компоненты одного уровня, находящиеся в разных узлах сети.

Для планомерного развития и стандартизации сетевых соединений, служб, технологии и устройств, необходимо некоторое вкратце сформулированное соглашение, определяющее основные принципы, параметры и термины, на основе которых можно будет разрабатывать конкретные решения. Такое рамочное соглашение, в общих чертах определяющее порядок приема и передачи информации на всех уровнях сетевого взаимодействия, получило название сетевой модели.

Сетевая модель — это соглашения на концептуальном уровне о том, как принимать и передавать данные для всех этапов сетевого взаимодействия, начиная от передачи и приема битов до определения того, как интерпретировать принятые данные.

Логично было бы начать развитие сетевых технологий именно с создания сетевой модели и на основе этой модели строить различные протоколы, соответствующие различным уровням этой модели. Однако реальное развитие сети шло иначе: по мере необходимости создавались наборы сетевых протоколов, позволяющие эффективно реализовать сетевое взаимодействие в той или иной конфигурации сети, с тем или иным аппаратным и программным обеспечением. Такие наборы протоколов получили название стеков.

Стек протоколов — это иерархически организованная совокупность протоколов для обеспечения взаимодействия узлов компьютерной сети.

Известно несколько стеков протоколов, самыми широко распространенными из которых являются TCP/IP, IPX/SPX, NetBIOS/SMB. Мы ограничимся рассмотрением стека TCP/IP, поскольку на протоколах этого стека построен весь Интернет.

13.6.2. Стек протоколов TCP/IP

Уровень сетевых интерфейсов

Уровню *сетевых интерфейсов* не сопоставлен ни один протокол, но на нем реализована поддержка практически всех известных сегодня технологий и протоколов объединения компьютеров в сеть.

Уровень межсетевого взаимодействия

На уровне *межсетевого взаимодействия* решаются задачи маршрутизации данных. На этом уровне работают несколько протоколов.

- *IP* (Internet Protocol — протокол межсетевого взаимодействия). Решает задачу передачи данных между сетями.
- *RIP* (Routing Information Protocol — протокол маршрутной информации) и *OSPF* (Open Shortest Path First — выбор кратчайшего пути первым). Протоколы сбора и конфигурирования маршрутной информации, отвечающие за выбор оптимального маршрута передачи данных.
- *ICMP* (Internet Control Message Protocol — протокол межсетевых управляющих сообщений). При помощи этого протокола собирается информация об ошибках доставки и длительности жизни пакетов, а также передаются тестирующие сообщения, подтверждающие наличие запрошенного узла в сети.

Транспортный уровень

Транспортный уровень предоставляет механизмы доставки данных.

- *TCP* (Transmission Control Protocol — протокол управления передачей). Описывает правила создания логического соединения между удаленными процессами и механизм обработки ошибок доставки пакетов (механизм повторной передачи «сбойных» пакетов).
- *UDP* (User Datagramm Protocol — протокол пользовательских датаграмм). Упрощенный вариант протокола доставки данных без установления логического соединения и проверки ошибок доставки пакетов.

Прикладной уровень

К прикладному уровню относятся протоколы, носящие прикладной характер. Большинство этих протоколов связано с соответствующими прикладными программами, работающими на их основе.

- *FTP* (File Transfer Protocol — протокол передачи файлов). В качестве транспортного протокола этот протокол использует TCP, что повышает надежность передачи файлов через большое количество промежуточных узлов.
- *TFTP* (Trivial File Transfer Protocol — простейший протокол передачи файлов). Этот протокол базируется на UDP и используется в локальных сетях.
- *SNMP* (Simple Network Management Protocol — простой протокол управления сетью).
- *Telnet* — протокол, используемый для эмуляции терминала удаленной станции.
- *SMTP* (Simple Mail Transfer Protocol — простой протокол передачи сообщений). Передает сообщения электронной почты при помощи транспортного протокола TCP.
- *HTTP* (Hiper Text Transfer Protocol — протокол передачи гипертекста). Базовый протокол Всемирной паутины, без которой сегодня невозможно представить себе Интернет. Именно он обеспечивает передачу страниц сайтов на наши компьютеры.

Кроме перечисленных базовых протоколов, в состав стека TCP/IP на прикладном уровне входит еще множество протоколов.

13.6.3. Сетевая модель OSI

Когда стек протоколов TCP/IP уже в полную силу обеспечивал функционирование самых разнообразных сетей, международная организация по стандартизации (International Organization for Standardization, ISO) разработала концептуальную *модель взаимодействия открытых систем* (Open Systems Interconnection, OSI). Эта модель оказалась настолько удачной, что в настоящее время многие сетевые процессы и проблемы принято описывать именно в терминах модели OSI. В модели OSI три базовых понятия: уровень, интерфейс и протокол.

Уровни пронумерованы от 7-го (верхний уровень) до 1-го (нижний уровень). Чем выше уровень, тем более глобальны решаемые им задачи. Каждый вышестоящий уровень реализует свою функциональность, получая услуги от нижележащего уровня и управляя им. Управление и передача услуг осуществляются через стандартные интерфейсы, благодаря которым вышестоящий уровень изолируется от детализации того, как именно реализует услуги нижележащий уровень. Взаимодействие протоколов смежных уровней в одном узле осуществляется через интерфейсы.

Уровни модели OSI перечислены в табл. 13.2.

Таблица 13.2. Уровни модели OSI

Номер	Наименование	Описание
7	Прикладной	Взаимодействие с прикладными программами пользователя
6	Представления	Шифрование, дешифрование, кодирование, перекодирование
5	Сеансовый	Координация связи между двумя рабочими станциями в сети
4	Транспортный	Разбиение на пакеты и сборка пакетов в сообщения
3	Сетевой	Адресация и маршрутизация пересылаемых пакетов
2	Канальный	Формирование пакетов, требуемых для соответствующих сетей
1	Физический	Физические, механические и электрические характеристики линий связи

13.7. Феномены века коммуникаций

13.7.1. Военно-техническая область

Развитие сетей, было инициировано министерством обороны США и мотивировано в основном военными потребностями. Не удивительно, что именно военные в первую очередь используют современные преимущества, которые дает им сеть:

- ☐ получение информации всегда, везде и отовсюду;
- ☐ управление войсками всегда, везде и отовсюду;

- высоконадежная оценка угроз;
- военное прогнозирование и моделирование.

Что же касается возможных отрицательных явлений тотального внедрения компьютеров и компьютерных сетей в военную область, то пока их можно увидеть только в фантастических фильмах: суперкомпьютеры начинают мировую войну, суперкомпьютеры правят миром. Цена ошибки или сбоя в компьютерных сетях при передачи военных данных может быть весьма велика.

13.7.2. Финансовая сфера

Если поначалу в финансовой сфере компьютерные сети казались великим благом, то сегодня это благо надо рассматривать как минимум с большими оговорками. Виртуальные рынки, виртуальные биржи, виртуальные деньги, виртуальные товары и услуги — все это стало доступно в любой точке планеты благодаря опоясывающим планету компьютерным сетям. Но благодаря им же стали возможны и виртуальные аферы, и виртуально организованные финансовые потрясения.

13.7.3. Бизнес

В отличие от финансово-экономической области, которая занимается производством, регулированием и сбытом денег, сфера бизнеса, производящая реальные товары и услуги, от развития компьютерных сетей только выиграла.

Сетевой бизнес развивался так интенсивно, что для его обозначения даже было придумано специальное название — *электронная коммерция* (e-commerce). Специальные названия придуманы были даже для сетевых технологий, обеспечивающих электронную коммерцию на том или ином уровне взаимодействия. Английские аббревиатуры этих названий содержат символ двойки, который произносится на английском созвучно предлогу «to», в данном контексте обозначающему дателный падеж (табл. 13.3).

Таблица 13.3. Технологии электронной коммерции

Аббревиатура	Название	Примеры
B2C	Бизнес клиенту (Business-to-Client)	Покупка в виртуальных магазинах
B2B	Бизнес бизнесу (Business-to-Business)	Фирма, занятая сборкой компьютеров, заказывает комплектующие у фирмы-производителя
G2B	Государство бизнесу (Government-to-Business)	Распространение через сеть нормативных и законодательных актов, регулирующих работу бизнеса
C2C	Клиент клиенту (Client-to-Client)	Клиенты обмениваются мнениями о качестве и цене купленного товара. Форумы по продаже вещей, находившихся в употреблении
P2P	Одноранговые сети (Peer-to-Peer)	Совместное использование файлов

13.7.4. География и навигация

В области географии и навигации сетевые технологии привнесли в нашу жизнь следующие достижения:

- ❑ мгновенное получение спутниковых карт поверхности земли в любой точке, причем если для гражданских нужд степень детализации такая, что позволяет рассмотреть машины на стоянке, то военные карты дают возможность видеть номера этих машин и знаки различия на форме;
- ❑ современные навигационные системы, позволяющие прокладывать маршрут практически в любой местности.

13.7.5. Наука

Положительными факторами в развитии науки стали следующие возможности:

- ❑ мгновенный обмен информацией о результатах проведенных исследований;
- ❑ присутствие на виртуальных научных конференциях;
- ❑ мгновенная публикация результатов исследований;
- ❑ виртуальное участие в научных экспериментах.

Однако кроме огромных преимуществ сеть принесла в науку и отрицательные явления, к которым можно отнести появление большого количества квазинаучной информации, не просто доступной, но более доступной для широкой публики, чем истинно научная информация.

13.7.6. Знание

С появлением глобальных сетей человечество стало генерировать массивы данных, продуцировать самые разнообразные типы информации, которые нужно было каким-то образом критически осмысливать, упорядочивать, превращая в действенное и полезное знание. Наиболее ярким и впечатляющим примером такого технологического решения является Википедия (см. русский вариант по адресу <http://ru.wikipedia.org/>), через 5 лет после создания обошедшая по посещаемости общепризнанную, наиболее старую и универсальную энциклопедию, Британнику (<http://www.britannica.com/>). Таким образом, порождая хаос данных, сеть одновременно породила и инструмент для преодоления этого хаоса, инструмент эффективной и всесторонней добычи и упорядочивания знаний. Сама технология *wiki* сейчас широко используется для создания баз знаний самыми различными сетевыми сообществами.

13.7.7. Образование

Сетевые технологии в области образования также выступили в крайне неоднозначной роли. С одной стороны, они принесли несколько безусловно положительных моментов:

- ❑ образование стало более доступным благодаря различного рода дистанционным курсам и образовательным программам;

- работа преподавателя стала более эффективной, поскольку видеть на экране процесс выполнения задания любым учащимся куда продуктивнее, чем подходить к каждому компьютеру;
- учащиеся получили доступ к альтернативным источникам знания, и преподаватель перестал быть единственным поставщиком знания (теперь при наличии терпения и желания на любую тему в Интернете можно найти самую исчерпывающую и актуальную информацию).

С другой стороны, нельзя не отметить и отрицательные моменты, в частности, теперь учащиеся все реже задания выполняют самостоятельно, предпочитая искать готовые решения в Интернете.

13.7.8. Технологии

Само развитие сетевых технологий было основано на энтузиазме групп научных сотрудников и открытости. Первые сетевые операционные системы были открытыми, причем открытым было все: исходные коды, спецификации протоколов, результаты испытаний и направления исследований. Участники разработок свободно обменивались друг с другом любыми сведениями, и это привело к показательному результату: почти все, что было разработано на заре возникновения сетевых технологий, используется по сей день.

С определенного момента возникая на основе свободных принципов, развившаяся и полноценно функционирующая сеть привела к созданию новых технологий, которые мы сегодня знаем под общим названием СПО (свободное программное обеспечение). Без глобальных сетей даже представить себе невозможно, как группа энтузиастов из самых разных точек земного шара смогла бы организовать и создать полноценную сетевую операционную систему, конкурирующую сегодня на равных с продуктами самых мощных производителей программного обеспечения. Речь идет о GNU/Linux.

СПО ценно не только само по себе, оно будит творческую мысль у людей в самых разных отраслях науки и техники, подвигая их на создание открытых аналогов не только программного обеспечения, но и техники.

13.7.9. Общество

Влияние, которое оказывает сеть на социум, тоже двойственно. Благодаря сетевым технологиям все больше людей реализует свое право на информацию, в том числе на информацию о действиях власти. Практически все, что власть в любой стране хотела бы сделать тайно, благодаря Интернету сегодня становится явным. Таким образом, Интернет более полноценно, чем обычные средства массовой информации, реализует концепции свободы слова и свободы средств массовой информации. Кроме того, во многих странах давно уже разработаны и внедрены так называемые сетевые правительства, приемные, администрации и прочие сетевые представительства органов власти. Большое количество операций, на которые раньше тратились дни и часы, человек сегодня легко может выполнить, не выходя из дома: заказать справку, а если нужна личная встреча, то записаться на прием

к чиновнику в указанное время. Можно проводить через сеть коммунальные платежи или получать информацию о порядке оформления тех или иных документов.

Кроме этих «государственных» и «официальных» функций, Интернет оказался благоприятной средой для создания различных сообществ. Сегодня просто невозможно представить себе жизнь без этого слоя социальной активности: интернет-чаты, интернет-конференции, блоги, сетевые фан-клубы и, конечно же, драгоценный интернет-исейджер ICQ. Про популярность социальных сетей, таких как «Одноклассники» или «В контакте» знает любой пользователь Интернет.

Но есть и другая чаша весов. Именно сеть сделала простым и возможным то, что без нее было бы труднодоступно и трудновыполнимо. Призывы к насилию, разжигание расовой, национальной и религиозной розни, публикация фотографий и видеоматериалов, содержащих реальные сцены насилия и жестокости, порнография, пошлость, глупость и грубость — все это в избытке выплеснулось на страницы Интернета. Что делать с этими общественными проявлениями? Являются ли они отражением состояния общества, или просто чувство бесконтрольности и анонимности, которое дает сеть, обостряет в человеке не только хорошее, но и плохое? Так или иначе, обществу еще предстоит оценить данные феномены и выработать необходимый в этом случае иммунитет.

13.7.10. Здоровье и экология

Влияние сетевых технологий на экологию и здоровье человека пока мало изучено. Да, мы знаем, что сетевые технологии принесли безусловную пользу в сфере медицины: информация о новых лекарствах и способах лечения, виртуальные консультации у специалистов, оперативная информация о новых видах заболеваний и методах защиты от них составляют большую ценность и, безусловно, способствуют охране здоровья и развитию медицины. Сегодня уже не редкость проведение операций, курируемых консилиумом врачей, наблюдающих за ходом операции удаленно, на экранах компьютеров, и это тоже феномен, который без сетевых коммуникаций был бы попросту невозможен.

Однако все сетевые коммуникации — это мощнейший источник электромагнитных излучений, оказывающих определенное влияние на организм человека (особенно в миллиметровом спектре, на частотах работы технологий Wi-Fi, Bluetooth и мобильных телефонов). То, что такое влияние существует, никаких сомнений нет. Также нет и однозначных результатов научных исследований на этот счет.

Еще одним феноменом, напрямую связанным со здоровьем, является доступность в сети разного рода медицинских сведений и вызванная этой доступностью иллюзия медицинской осведомленности. Эти иллюзии наличия готовых рецептов от всего несут безусловную опасность для здоровья, поскольку человек, вместо того чтобы обратиться к врачу, начинает лечить себя сам.

Вопросы для самопроверки

1. Почему сеть на базе мэйнфрейма нельзя в полной мере считать компьютерной?
2. Какая идея Барэна до сих пор лежит в основе сетевого взаимодействия компьютеров?
3. По каким признакам можно классифицировать компьютерные сети?
4. Что такое «широковещание»?
5. Чем отличается сеть с коммутацией каналов от сети с коммутацией пакетов?
6. Перечислите известные вам виды проводных соединений.
7. Верно ли выражение «глобальная компьютерная сеть Интернет»?
8. Что такое «одноранговая сеть»?
9. Что такое «топология сети»?
10. Чем звездообразная топология отличается от кольцевой?
11. На каком основании выбирается топология компьютерной сети?
12. Что такое «витая пара»?
13. Сравните между собой кабели типа «витая пара» и «тонкий коаксиал». В чем преимущества и недостатки того и другого решения?
14. Почему оптоволоконный кабель не получил повсеместного распространения, несмотря на свои достоинства?
15. Каково назначение сетевых адаптеров?
16. Что такое «хаб» и для чего он используется?
17. Что такое «сетевой протокол» и зачем он нужен?
18. В чем разница между сетевой моделью и стеком протоколов?
19. Перечислите известные вам протоколы транспортного уровня стека TCP/IP.
20. За что отвечают протоколы прикладного уровня?
21. Перечислите уровни сетевой модели OSI.

Литература

1. Олифер В. Г., Олифер Н. А. Компьютерные сети. СПб.: Питер, 2006.
2. Паркер Т., Сиян К. TCP/IP для профессионалов. СПб.: Питер, 2004.
3. Таненбаум Э. Компьютерные сети. СПб.: Питер, 2003.

Часть третья

Алгоритмическое и программное обеспечение информатики

Глава 14. Основы теории алгоритмов	391
Глава 15. Классификация и тенденции развития программного обеспечения	432
Глава 16. Системное программное обеспечение компьютера	465
Глава 17. Офисное программное обеспечение	502
Глава 18. Средства мультимедиа	524
Глава 19. Технологии и инструменты программирования	552

Глава 14 Основы теории алгоритмов

- 14.1. Представление об алгоритмах
- 14.2. Способы записи алгоритмов
- 14.3. Базовые алгоритмические конструкции
- 14.4. Алгоритмы обработки данных разного типа
- 14.5. Алгоритмы сортировки и поиска

В основе работы любого механизма или устройства, будь то двигатель внутреннего сгорания, сложная компьютерная система или биологический механизм организма человека, лежит определенная и преследующая некоторую цель последовательность действий. Такая последовательность действий получила название «алгоритм». Настоящая глава посвящена описанию того, каким образом можно составить и представить на компьютере те или иные алгоритмы, какие задачи можно решать при помощи алгоритмов определенного типа.

14.1. Представление об алгоритмах

14.1.1. Понятие алгоритма

Во всех сферах своей деятельности, в частности в сфере обработки информации, человек сталкивается с различными способами или методиками решения разнообразных задач. Они определяют порядок выполнения действий для получения желаемого результата — мы можем трактовать это как первоначальное или интуитивное определение алгоритма.

Термин *алгоритм* происходит от имени средневекового узбекского математика Аль-Хорезми, который еще в 825 году описал правила выполнения четырех арифметических действий в десятичной системе счисления. Процесс их выполнения и был назван *алгоризмом*.

Впоследствии вместо слова алгоритм стали употреблять латинизированное слово *алгорисмус*, смысл которого состоял в комбинировании четырех операций арифметического исчисления: сложения, вычитания, умножения и деления. Затем алгоритмус преобразовался в *алгорифм*. Смысл этого понятия чаще всего связывался с алгорифмами Евклида — описаниями процессов нахождения наибольшего общего делителя двух натуральных чисел, наибольшей общей меры двух отрезков и т. п.

Под алгоритмом же понимали конечную последовательность точно сформулированных правил, которые позволяют решать различные классы задач. Такое определение алгоритма не является строгим математическим, так как в нем не содержится точной характеристики того, что следует понимать под классами задач и под правилами их решения. Таким образом, можно нестрого определить алгоритм как однозначно трактуемую процедуру решения задачи.

Алгоритм — это система однозначных инструкций (указаний), которая определяет последовательность действий над выбранными объектами с целью получения результата за конечное число шагов.

Дополнительные требования о выполнении алгоритма за конечное время для любых входных данных приводят к еще одному неформальному определению алгоритма.

Алгоритм — это заданное на некотором языке конечное предписание, задающее конечную последовательность выполнимых и точно определенных элементарных операций для решения задачи.

Пусть DZ — область (множество) исходных данных задачи Z , а R — множество возможных результатов, тогда мы можем говорить, что алгоритм осуществляет отображение $DZ \rightarrow R$.

Поскольку такое отображение может быть не полным, в теории алгоритмов вводятся понятия частичного и полного алгоритма. Алгоритм называется *частичным*, если мы получаем результат только для некоторых значений d , принадлежащих DZ , и *полным*, если алгоритм получает правильный результат для всех значений d , принадлежащих DZ .

Сегодня отсутствует одно исчерпывающее строгое определение понятия «алгоритм». Из разнообразных вариантов словесного определения наиболее удачные, видимо, принадлежат российским ученым А. Н. Колмогорову и А. А. Маркову.

Алгоритм (по Колмогорову) — это система вычислений, выполняемых по строго определенным правилам, которая после какого-либо числа шагов заведомо приводит к решению поставленной задачи.

Алгоритм (по Маркову) — это точное предписание, определяющее вычислительный процесс, идущий от варьируемых исходных данных к исковому результату.

Алгоритмы можно подразделить на два класса: численные и логические.

Численные алгоритмы — это алгоритмы, в соответствии с которыми решение задач сводится к арифметическим действиям.

Логические алгоритмы — это алгоритмы, в соответствии с которыми решение задач сводится к логическим действиям.

Несмотря на различие в определениях, к алгоритмам предъявляется ряд общих требований. Итак, алгоритм должен:

- содержать конечное количество элементарно выполнимых предписаний, то есть удовлетворять требованию конечности записи;
- выполнять конечное количество шагов при решении задачи, то есть удовлетворять требованию конечности действий;
- быть единым для всех допустимых исходных данных, то есть удовлетворять требованию универсальности;
- приводить к правильному по отношению к поставленной задаче решению, то есть удовлетворять требованию правильности.

Формально строгие определения понятия алгоритма связаны с введением специальных математических конструкций — формальных алгоритмических систем или моделей вычислений, каковыми являются рассматриваемые далее машины Поста и Тьюринга.

14.1.2. Формализация понятия алгоритма посредством машины Поста

Пост рассматривает общую проблему, состоящую из множества конкретных проблем, при этом решением общей проблемы является такое решение, которое определяет ответ для конкретной проблемы.

Например, нахождение решения уравнения $3 \times x + 9 = 0$ — это конкретная проблема, так как в уравнении используются константы, а нахождение решения уравнения $a \times x + b = 0$ — это общая проблема, так как коэффициенты уравнения являются переменными величинами. Алгоритм должен быть универсальным, то есть должен находить решения поставленной задачи в общем виде. Отметим, что сам термин «алгоритм» не используется Э. Постом, его заменяет понятие набора инструкций.

Машина Поста представляет собой абстрактный механизм для выполнения алгоритмов. Этот механизм состоит из двух узлов (рис. 14.1):

- *бесконечной ленты*, разделенной на ячейки, каждая из которых может быть помечена или не помечена меткой V ;
- *механизма чтения-записи* (в дальнейшем — *каретка*), который может перемещаться вдоль ленты вправо и влево, записывать метку в пустую ячейку, стирать метку в помеченной ячейке и проверять наличие метки в ячейке.

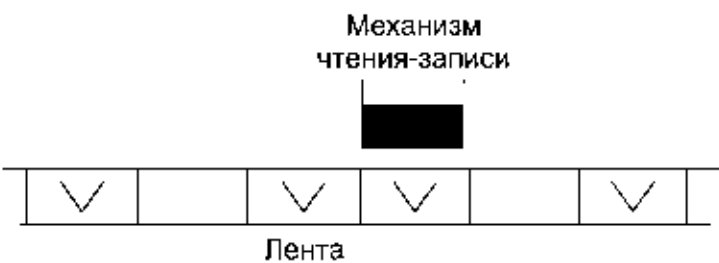


Рис. 14.1. Машина Поста

Конкретная проблема задается «внешней силой» (термин Поста) пометкой конечного количества ячеек, при этом очевидно, что любая конфигурация начинается и заканчивается помеченной ячейкой. После применения к конкретной проблеме некоторого набора инструкций, решение также представляется в виде набора помеченных и немеченных ячеек, распознаваемого той же внешней силой.

Пост в 1936 г. предложил набор инструкций, или элементарно выполнимых операций, которые выполняет «работник». В терминах современного состояния компьютерной области этот набор инструкций можно трактовать как минимальный набор битовых операций элементарного процессора. В табл. 14.1 приведены инструкции машины Поста. Латинскими буквами i, j, j_1, j_2 и т. д. обозначены номера инструкций; M — поместить ячейку, C — стереть метку.

Таблица 14.1. Инструкции машины Поста

№	Описание инструкции	Условное обозначение инструкции
1	Пометить ячейку, если она пуста, и перейти к инструкции j	i, Mj
2	Стереть метку, если она есть, и перейти к инструкции j	i, Cj
3	Переместиться влево на 1 ячейку и перейти к инструкции j	$i, \leftarrow j$
4	Переместиться вправо на 1 ячейку и перейти к инструкции j	$i, \rightarrow j$
5	Определить, помечена ячейка или нет. Если ячейка пуста, перейти на инструкцию j_1 , если помечена — на инструкцию j_2	$i, \rightarrow \begin{cases} \rightarrow j_1 \\ \rightarrow j_2 \end{cases}$
6	Остановиться	$i, \text{стоп}$

Программа представляет собой нумерованную последовательность инструкций, причем в инструкции 5 переходы производятся на указанные в ней номера других инструкций. Программа, или набор инструкций, в терминах Э. Поста, является одной и той же для всех конкретных задач, поэтому соотносена с общей постановкой задачи. Таким образом, Пост формулирует требование универсальности алгоритма.

Далее Э. Пост вводит следующие понятия:

- набор инструкций применим к общей проблеме, если для каждой конкретной проблемы не возникает коллизий в инструкциях 1 и 2, то есть программа никогда не стирает метку в пустой ячейке и не устанавливает метку в помеченной ячейке;
- набор инструкций заканчивается, если после конечного количества инструкций выполняется инструкция 6.

Начальное состояние машины Поста полностью определяется двумя факторами: положением каретки (на какой из ячеек она находится) и состоянием ленты (наличием на ленте помеченных ячеек).

В зависимости от начального состояния машины Поста могут быть три варианта выполнения одной и той же программы:

1. В ходе выполнения программы машина доходит до невыполнимой инструкции (печатаения метки в непустой ячейке или стирания метки в пустой ячейке). Происходит *безрезультатная остановка* машины.
2. В ходе выполнения программы машина доходит до инструкции *Стоп*. Происходит *результативная остановка*.
3. Работа машины продолжается бесконечно.

Машина Поста оперирует натуральными числами, которые представляются в непозиционной системе счисления (то есть 0 обозначается одной помеченной ячейкой, 1 — двумя идущими подряд помеченными ячейками, и т. д.). В общей форме натуральное число n представляется $n + 1$ смежными помеченными ячейками.

Пример. Рассмотрим следующую программу для машины Поста:

1. $\rightarrow \begin{cases} \rightarrow 4 \\ \rightarrow 1 \end{cases}$

2. С3

3. *стоп*

4. $\rightarrow 2$

Зададим такое начальное состояние машины, которое показано на рис. 14.2.

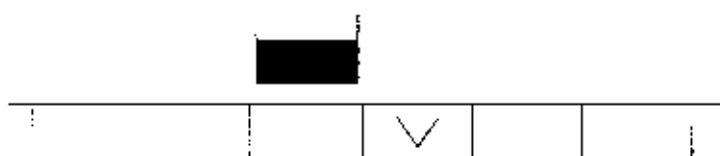


Рис. 14.2. Начальное состояние машины Поста

Поскольку каретка находится на пустой ячейке, первая команда программы вызывает переход на команду с номером 4. При этом каретка не меняет своего положения, и состояние ленты также не меняется (метка не ставится и не стирается).

Команда с номером 4 вызывает перемещение каретки на одну позицию вправо (теперь каретка находится на помеченной ячейке) и передачу управления команде 2.

Команда 2 стирает метку в текущей ячейке и передает управление команде 3.

Команда 3 останавливает работу машины.

14.1.3. Формализация понятия алгоритма посредством машины Тьюринга

Алан Тьюринг в 1936 г. опубликовал в трудах Лондонского математического общества статью «О вычислимых числах в приложении к проблеме разрешения»,

которая наравне с работами Поста и Черча лежит в основе современной теории алгоритмов. Предыстория появления этой работы связана с формулировкой Давидом Гильбертом на Международном математическом конгрессе в Париже в 1900 г. нерешенных математических проблем. Одной из них была задача доказательства непротиворечивости системы аксиом обычной арифметики, которую Гильберт в дальнейшем уточнил как «проблему разрешимости» — нахождение общего метода для определения выполнимости данного высказывания на языке формальной логики.

Статья Тьюринга как раз и давала ответ на эту задачу, но значение статьи Тьюринга выходило далеко за ее рамки. Вот как охарактеризовал эту работу Джон Холкрофт (известный американский ученый в области теории вычислительных систем): «Работая над проблемой Гильберта, Тьюрингу пришлось дать четкое определение самого понятия метода. Отталкиваясь от интуитивного представления о методе как о некоем алгоритме, то есть процедуре, которая может быть выполнена механически, без творческого вмешательства, он показал, как эту идею можно воплотить в виде подробной модели вычислительного процесса. Полученная модель вычислений, в которой каждый алгоритм разбивался на последовательность простых, элементарных шагов, и была логической конструкцией, названной впоследствии машиной Тьюринга».

Машина Тьюринга является расширением модели конечного автомата, включающим потенциально бесконечную память с возможностью перехода (движения) от обозреваемой в данный момент ячейки к ее левому или правому соседу. То есть «конструктивно» машина Тьюринга, так же как и машина Поста, состоит из двух основных узлов: ленты с ячейками и каретки.

Решаемая проблема задается путем записи на ленту конечного количества символов S_i , образующих слово в алфавите S , где e — это специальный пустой символ, показывающий отсутствие любого символа алфавита S в ячейке (рис. 14.3).

e	S_1	S_2	S_3	S_4	...	S_n	e

Рис. 14.3. Машина Тьюринга

В отличие от машины Поста, в машине Тьюринга каретка неподвижна, а лента проматывается лентопротяжным механизмом на одну ячейку вправо или влево. Состояния машины Тьюринга обозначаются как q_i и принадлежат множеству состояний Q . Управляется машина при помощи функции переходов, которая называется программой машины Тьюринга и может быть задана в виде таблицы или ориентированного графа. При этом при задании функции переходов используются как команды лентопротяжному механизму (смещение ленты на одну ячейку вправо, смещение ленты на одну ячейку влево, перемотка в начало, останов), так и команды каретки (запись нового символа, чтение символа, запись пустого символа). После задания проблемы машина переводится в начальное состояние (обозначаемое как q_0), и лента проматывается так, чтобы каретка находилась у самого левого непустого символа, после чего в соответствии с указанной функцией переходов машина начинает заменять обозреваемые символы, передвигать каретку вправо или влево

и переходить в другие состояния (путем стирания текущего символа и замены его новым). Останов машины происходит в том случае, если для пары «состояние машины — символ в текущей ячейке» ($q_i S_j$) функция перехода не определена или получена команда останова.

Тьюринг высказал предположение, что любой алгоритм в интуитивном смысле этого слова может быть представлен эквивалентной машиной в предложенной им модели вычислений. Это предположение известно как тезис Черча — Тьюринга. Каждый компьютер может моделировать машину Тьюринга, для этого достаточно моделировать операции перезаписи ячеек, сравнения и перехода к другой соседней ячейке с учетом изменения состояния машины. Значит, он может моделировать алгоритмы в машине Тьюринга, и из этого тезиса следует, что все компьютеры, независимо от мощности, архитектуры и других особенностей, с точки зрения принципиальной возможности решения алгоритмически разрешимых задач эквивалентны.

14.1.4. Современная теория алгоритмов

Начальной точкой отсчета современной теории алгоритмов можно считать теорему о неполноте символических логик, доказанную немецким математиком Куртом Геделем в 1931 г. В этой работе было показано, что некоторые математические проблемы не могут быть решены алгоритмами определенного класса.

Общность результата Геделя связана с вопросом о том, совпадает ли использованный им класс алгоритмов с классом всех алгоритмов в интуитивном понимании этого термина. Эта работа дала толчок к поиску и анализу различных формализаций понятия «алгоритм».

Первые фундаментальные работы по теории алгоритмов были опубликованы в середине 1930-х гг. Аланом Тьюрингом, Алоизом Черчем и Эмилем Постом. Предложенные ими машина Тьюринга, машина Поста и класс рекурсивно-вычислимых функций Черча были первыми формальными описаниями алгоритма, опирающимися на строго определенные модели вычислений. Сформулированные гипотезы Поста и Черча—Тьюринга постулировали эквивалентность предложенных ими моделей вычислений, как формальных систем, и интуитивного понятия алгоритма. Важным развитием этих работ стала формулировка и доказательство существования алгоритмически неразрешимых проблем.

В 1950-х гг. существенный вклад в развитие теории алгоритмов внесли работы Колмогорова и основанный на теории формальных грамматик алгоритмический формализм Маркова — так называемые нормальные алгоритмы Маркова. Формальные модели алгоритмов Поста, Тьюринга и Черча, равно как и модели Колмогорова и Маркова, оказались эквивалентными в том смысле, что любой класс проблем, разрешимых в одной модели, разрешим и в другой.

Появление доступных ЭВМ и существенное расширение круга решаемых на них задач привели в 1960—1970-х гг. к практически значимым исследованиям алгоритмов и вычислительных задач. На этой основе впоследствии выделилось несколько разделов теории алгоритмов (рис. 14.4).



Рис. 14.4. Разделы современной теории алгоритмов

Классическая теория алгоритмов: формулировка задач в терминах формальных языков; понятие задачи разрешения; описание сложных классов задач; формулировка в 1965 г. Эдмондсом проблемы $P = NP$; открытие класса TVP-полных задач и его исследование; введение новых моделей вычислений — алгебраического дерева вычислений (Бен-Ор), машины с произвольным доступом к памяти, схем алгоритмов Янова, стандартных схем программ Котова и др.

Теория асимптотического анализа алгоритмов: понятие сложности и трудоемкости алгоритма; критерии оценки алгоритмов; методы получения асимптотических оценок, в частности, для рекурсивных алгоритмов; асимптотический анализ трудоемкости или времени выполнения; получение теоретических нижних оценок сложности задач. В развитие этой теории внесли существенный вклад Кнут, Ахо, Хопкрофт, Ульман, Карп, Мошков, Кудрявцев и др.

Теория практического анализа вычислительных алгоритмов: получение явных функций трудоемкости; интервальный анализ функций; практически значимые критерии качества алгоритмов; методики выбора рациональных алгоритмов. Основополагающей работой в этом направлении, очевидно, следует считать фундаментальный труд Д. Кнута «Искусство программирования для ЭВМ».

Методы создания эффективных алгоритмов включают в себя множество алгоритмов, среди которых динамическое программирование, метод ветвей и границ, метод декомпозиции, «жадные» алгоритмы, специальные структуры данных и пр.

Обобщая исследования в различных разделах теории алгоритмов, можно выделить следующие основные задачи и направления развития, характерные для современной теории алгоритмов:

- формализация понятия «алгоритм» и исследование формальных алгоритмических систем (моделей вычислений);
- доказательство алгоритмической неразрешимости задач;
- формальное доказательство правильности и эквивалентности алгоритмов;
- классификации задач, определение и исследование сложных классов;
- доказательство теоретических нижних оценок сложности задач;
- получение методов разработки эффективных алгоритмов;
- асимптотический анализ сложности итерационных алгоритмов;
- исследование и анализ рекурсивных алгоритмов;

- получение явных функций трудоемкости алгоритмов;
- разработка классификаций алгоритмов;
- исследование емкостной (по ресурсу памяти) сложности задач и алгоритмов;
- разработка критериев сравнительной оценки ресурсной эффективности алгоритмов и методов их сравнительного анализа.

14.2. Способы записи алгоритмов

Для строгого задания различных структур данных и алгоритмов их обработки требуется иметь такую систему формальных обозначений и правил, чтобы смысл всякого используемого предписания трактовался точно и однозначно. Соответствующие системы правил называют языками описаний. К средствам описания алгоритмов относятся следующие основные способы их представления: словесный, графический (блок-схема), псевдокоды, программный, диаграмма Нэсси—Шнейдермана (рис. 14.5).



Рис. 14.5. Способы представления алгоритмов

14.2.1. Словесный способ представления алгоритма

Словесный способ записи алгоритмов представляет собой последовательное описание основных этапов обработки данных, заданное в произвольном изложении на естественном языке. Этот способ записи основан на общепринятых средствах общения между людьми. Его удобно использовать на начальном этапе алгоритмизации задачи.

Рассмотрим понятие словесного представления алгоритма на примере нахождения произведения n натуральных чисел — факториала числа n ($s = n!$), то есть вычисления по формуле $s = 1 \times 2 \times 3 \times \dots \times n$.

Этот процесс может быть записан в виде следующей последовательности шагов (инструкций, пунктов, указаний):

1. Присваиваем переменной s значение, равное единице, и переходим к следующему шагу.
2. Присваиваем переменной i значение, равное единице, и переходим к следующему шагу.

3. Вычисляем выражение $i \times c$, полученное значение присваиваем переменной c и переходим к следующему шагу.
4. Проверяем, равно ли значение переменной i значению переменной n . Если $i = n$, то вычисления прекращаем. Если $i < n$, то увеличиваем i на единицу и переходим к шагу 3.

Неудобства словесных определений связаны с проблемой однозначной трактовки терминов. В таких определениях должен быть, хотя бы неявно, указан исполнитель действий (предписаний). Например, алгоритм вычисления производной для полинома фиксированной степени вполне ясен тем, кто знаком с основами математического анализа, но для прочих он может оказаться совершенно непонятным. Это рассуждение заставляет указывать также вычислительные возможности исполнителя, а именно, уточнять какие операции для него являются «элементарными».

Другие трудности связаны с тем, что алгоритм заведомо существует, но его очень трудно описать в некоторой заранее заданной форме. Классический пример такой ситуации — алгоритм завязывания шнурков на ботинках или узла на галстуке. Практически невозможно дать только словесное описание этого алгоритма без использования иллюстраций.

К недостаткам словесного способа записи можно отнести следующее:

- полное подробное словесное описание алгоритма получается очень громоздким;
- естественный язык допускает неоднозначность толкования инструкций;
- при переходе к этапу программирования требуется дополнительная работа по формализации алгоритма, так как словесное описание может быть понятно человеку, но «непонятно» компьютеру.

По этим причинам словесный способ записи алгоритмов не получил широкого распространения.

14.2.2. Графический способ записи алгоритма

Запись алгоритма с помощью графических объектов в виде блок-схемы (ГОСТ 19.701-90, ИСО 5807-85) применяется довольно широко для представления простых алгоритмов небольшого размера. Однако по мере роста сложности отображаемого фрагмента алгоритма (программы) его логическая структура перегружается деталями и связями («спагетти») и схема становится нечитабельной. По этой причине в настоящее время блок-схемы алгоритмов используются в основном для иллюстрации программ¹.

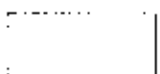
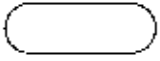
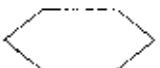
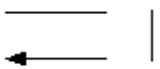
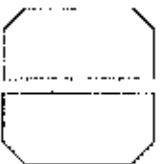
При графическом представлении алгоритма с помощью схемы каждый пункт алгоритма отображается на схеме некоторой геометрической фигурой (блоком) и дополняется элементом словесного описания. Блоки на схемах соединяются

¹ Кроме блок-схем алгоритмов, для тех же целей представления алгоритмов, программ или иных видов деятельности, в том числе и на этапе проектирования систем, могут использоваться и другие виды графической нотации. Среди них заметное место в современных методиках разработки занимает диаграмма деятельности (Activity diagram), являющаяся частью унифицированного языка моделирования UML.

линиями потоков информации. Основное направление потока информации идет сверху вниз и слева направо (стрелки могут не указываться), снизу вверх и справа налево (стрелки обязательны). Количество входящих линий для блока не ограничено. Выходящая линия должна быть одна (исключение составляет блок решения).

Основные элементы блок-схемы алгоритма приведены в табл. 14.2.

Таблица 14.2. Основные элементы схемы алгоритма (по ГОСТ 19.701-90)

№	Символическое обозначение	Наименование	Описание
1		Процесс	Блок функции обработки данных любого вида: выполнение определенной операции или группы операций, приводящее к изменению значения, формы, размещения информации или к определению направления дальнейшего движения
2		Решение	Блок решения или функции переключательного типа. Внутри блока записывается условие. Блок имеет один вход и два альтернативных выхода: «да» — условие выполнено, «нет» — условие не выполнено
3		Данные	Блок отображает данные, носитель данных не определен
4		Терминатор	Блок отображает выход во внешнюю среду и вход из внешней среды (начало или конец схемы)
5		Соединитель	Символ отображает выход в часть схемы и вход из другой части этой схемы и используется для обрыва линии и продолжения ее в другом месте. Соответствующие символы-соединители должны быть одинаковыми
6		Предопределенный процесс	Блок для отображения подпрограммы или модуля
7		Подготовка	Блок отражает модификацию команды или группы команд с целью воздействия на некоторую последующую функцию (установка переключателя, модификация индексного регистра или инициализация программы)
8		Комментарий	Символическое обозначение используется для добавления комментариев. Пунктирные линии в символе комментария связаны с соответствующим блоком. Текст комментария должен быть помещен около скобки
9		Линия	Символ отображает поток данных или управление
10		Граница цикла	Символ, состоящий из двух частей, отображает начало и конец цикла. Обе части символа имеют один и тот же идентификатор. Условия для инициализации, приращения, завершения и т. д. помещаются внутри символа в начале или в конце в зависимости от расположения операции, проверяющей условие

На рис. 14.6 в виде блок-схемы представлен алгоритм нахождения минимального числа M в последовательности из n чисел $a_1, a_2, \dots, a_n$ ($n \neq 0$).

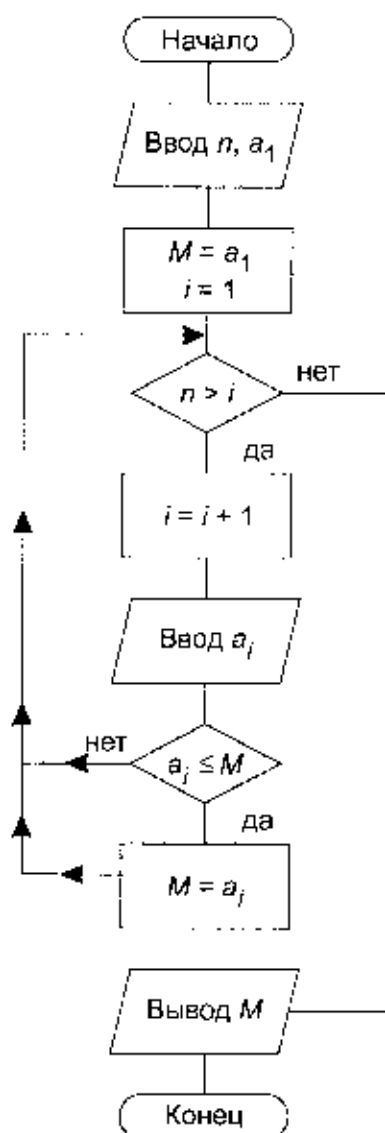


Рис. 14.6. Блок-схема алгоритма нахождения минимума в последовательности чисел

14.2.3. Представление алгоритма с помощью диаграммы Нэсси—Шнейдермана

Диаграмма Нэсси—Шнейдермана была предложена в сочетании со структурным программированием как средство борьбы с присущей схемам алгоритмов проблемой большого количества стрелок, отображающих передачу управления на другую часть программы (так называемого «спагетти-кода»). Эта диаграмма заменяет одномерное представление вложенных операторов двухмерным. Тем не менее и в этом случае по мере роста сложности программного кода проявляются проблемы отображения, поскольку элементы диаграммы быстро становятся все меньше и меньше.

На рис. 14.7 приведены составные элементы диаграммы Нэсси—Шнейдермана.

На рис. 14.8 представлен алгоритм нахождения действительных корней квадратного уравнения $ax^2 + bx + c = 0$, представленный при помощи диаграммы Насси-Шнейдермана.

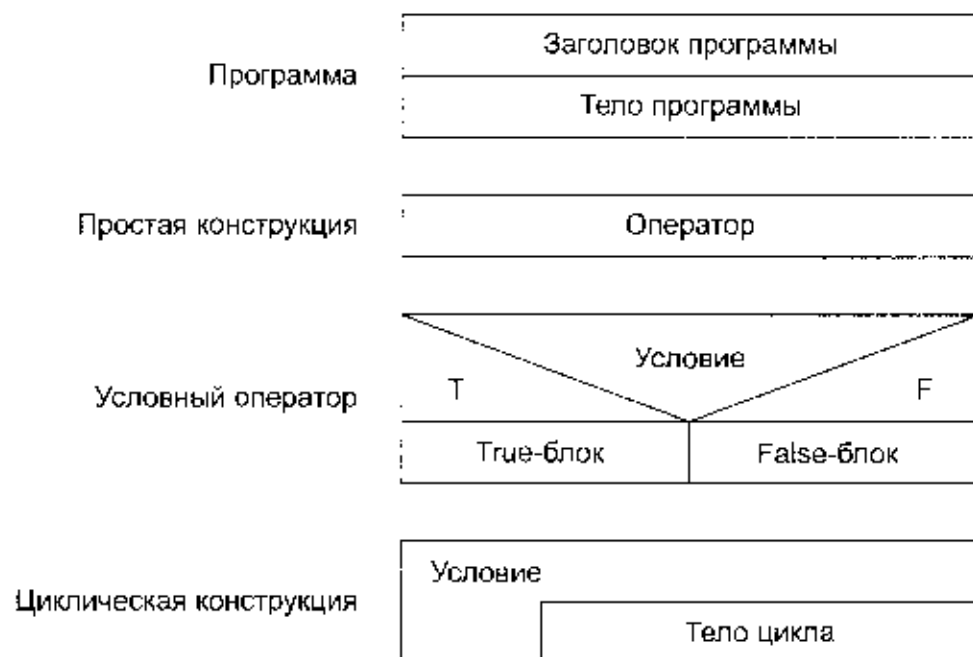


Рис. 14.7. Элементы диаграммы Нэсси—Шнейдермана

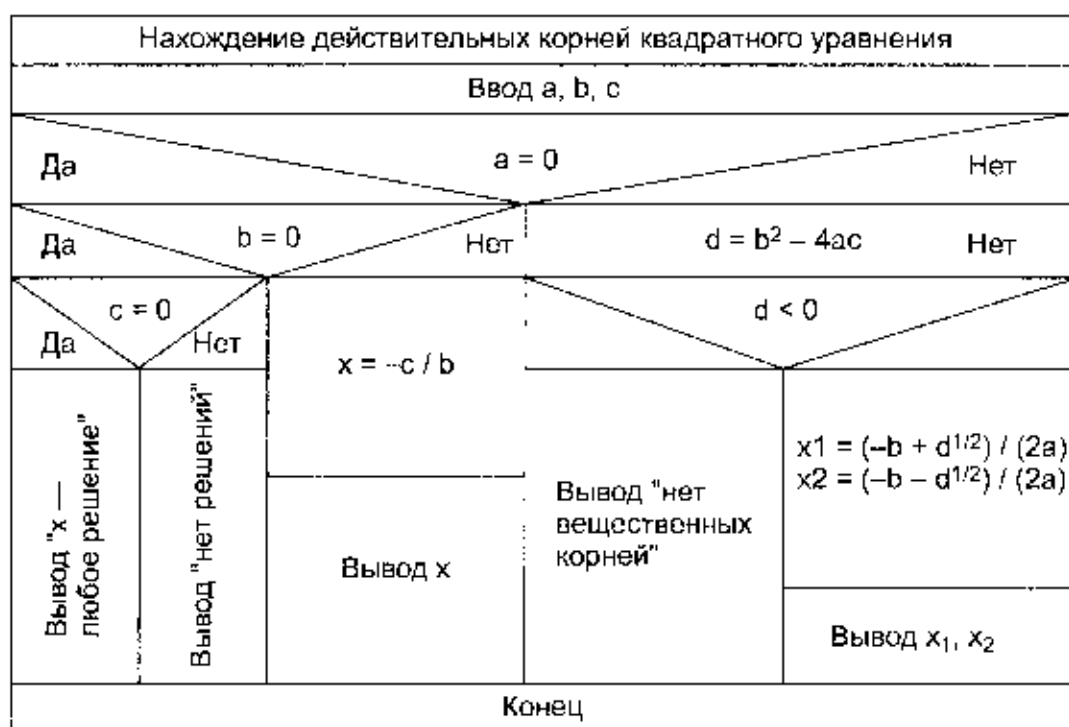


Рис. 14.8. Алгоритм, представленный при помощи диаграммы Нэсси—Шнейдермана

14.2.4. Представление алгоритма с помощью псевдокодов

Широкое распространение получил способ записи алгоритма при помощи псевдокода. Этот способ записи является промежуточным — к нему прибегают перед записью алгоритма в терминах выбранного языка программирования. Псевдокод представляет собой удобный для практического применения промежуточный язык. Это симбиоз естественного языка и языка программирования. Псевдокод похож на язык программирования тем, что в нем, с одной стороны, присутствуют некоторые инструкции и конструкции языка программирования, а с другой — допускается словесная или формульная запись там, где сложно использовать средства языка программирования.

Примером псевдокода является алгоритмический язык (АЯ), содержащий систему обозначений для единообразной и точной записи алгоритмов и задания правил их использования. Важной особенностью алгоритмических языков типа псевдокодов является их близость к языкам программирования.

Как и любой язык, АЯ строится на основе алфавита, включающего в себя набор символов, разрешенных к применению при написании алгоритмов. Алфавит АЯ включает в себя строчные и прописные буквы русского и латинского алфавитов; цифры десятичной системы счисления; специальные символы, имеющиеся на клавиатуре устройства ввода данных ПК и в наборах устройств печати; символы математических операций, используемые при написании выражений.

Для дополнения символов алфавита в АЯ вводятся так называемые ключевые (служебные) слова, которые позволяют сделать запись алгоритма более понятной и выразительной, формализованной. Они используются для формирования типовых синтаксических конструкций.

Перечислим основные ключевые слова алгоритмического языка: алг (алгоритм), рез (результат), нач (начало), кон (конец), арг (аргумент), знач (значение), тип, вещ (вещественный), цел (целый), лит (литерный), таб (табличный), сим (символьный), не, то, если, и, все, или, выбор, иначе, нц (начало цикла), кц (конец цикла), от, до, шаг, для, пока, := (оператор присваивания), при, да, нет.

Дополнительные ключевые слова: запись, истина, лог (логический), ложь, массив, множество, функция, дано, надо, ввод, вывод, утв (утверждение).

Пример программы, записанной на алгоритмическом языке:

```
алг Сумма квадратов (арг цел  $n$ , рез цел  $S$ )
дано |  $n > 0$ 
надо |  $S = 1 \cdot 1 + 2 \cdot 2 + 3 \cdot 3 + \dots + n \cdot n$ 
нач
цел  $i$ 
ввод  $n$ ;  $S := 0$ 
нц для  $i$  от 1 до  $n$ 
 $S := S + i \cdot i$ 
кц
вывод " $S =$ ",  $S$ 
кон
```

14.2.5. Программный способ представления алгоритмов

При записи алгоритма в словесной форме, в виде блок-схемы или на псевдокоде допускается определенный произвол при изображении команд. Вместе с тем такая запись точна настолько, что позволяет человеку понять суть дела и исполнить алгоритм. Однако на практике в качестве исполнителей алгоритмов используются компьютеры или иные вычислительные устройства (однокристальные микроЭВМ, промышленные компьютеры, технологические контроллеры и др.), поэтому алгоритм, предназначенный для исполнения на компьютере, должен быть записан на понятном ему языке. И здесь на первый план выдвигается необходимость точной записи команд, не оставляющей места для произвольного толкования их исполнителем. Следовательно, язык для записи алгоритмов должен быть формализован. Такой язык принято называть языком программирования, а запись алгоритма на этом языке — программой для компьютера.

В настоящее время в мире существует несколько сотен реально используемых языков программирования. Для каждого есть своя область применения. В зависимости от степени детализации предписаний обычно определяется уровень языка программирования — чем язык менее детален, тем выше его уровень. По этому критерию можно выделить следующие уровни языков программирования (рис. 14.9)¹:

- *машинно-ориентированные языки* — машинные языки и языки ассемблера;
- *машинно-независимые языки* — языки высокого уровня.



Рис. 14.9. Классификация языков программирования

Машинно-ориентированные языки — это языки низкого уровня, требующие указания мелких деталей процесса обработки данных. Языки же высокого уровня имитируют естественные языки, используя некоторые слова естественного языка и общепринятые математические символы. Эти языки более удобны для человека.

Языки высокого уровня делятся на процедурные, логические и объектно-ориентированные.

- *Процедурные, или алгоритмические, языки* (Basic, Pascal, C и др.) предназначены для однозначного описания алгоритмов в виде некоторой последовательности операторов языка.

¹ Приведенная классификация не является единственной.

- *Логические* языки (Prolog, Lisp и др.) ориентированы не на разработку алгоритма решения задачи, а на систематическое и формализованное описание задачи, из которого должно следовать решение.
- *Объектно-ориентированные* языки (Object Pascal, C++, Java, C# и др.), в основе которых лежит понятие объекта, сочетают в себе данные и действия над ними. Программа на объектно-ориентированном языке, решая некоторую задачу, по сути, описывает часть мира, относящуюся к этой задаче. Описание действительности в форме системы взаимодействующих объектов естественнее, чем в форме взаимодействующих процедур.

14.3. Базовые алгоритмические конструкции

Для выполнения типичных последовательностей действий в алгоритме разработаны базовые алгоритмические конструкции в виде определенного набора блоков и стандартных средств их соединения. К базовым алгоритмическим конструкциям относятся *линейные, разветвляющиеся и циклические*.

Линейный алгоритм — это алгоритм, в котором действия осуществляются последовательно друг за другом.

Пример линейного алгоритма приведен на рис. 14.10.

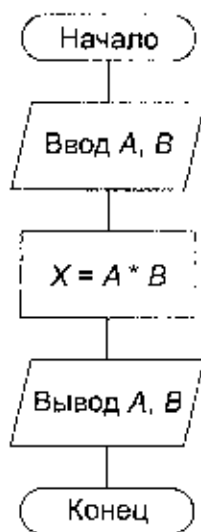


Рис. 14.10. Линейный алгоритм

В отличие от линейных алгоритмов, в которых команды выполняются последовательно одна за другой, в разветвляющиеся алгоритмы входит условие, в зависимости от выполнения или невыполнения которого выполняется та или иная последовательность команд (действий). Пример разветвляющегося алгоритма приведен на рис. 14.11.

Разветвляющийся алгоритм — это алгоритм, в котором действия выполняются по одной из возможных ветвей решения задачи в зависимости от выполнения условий.

В качестве условия в разветвляющемся алгоритме может быть использовано любое понятное исполнителю утверждение, которое может соблюдаться (быть истинным) или не соблюдаться (быть ложным). Такое утверждение может быть выражено как словами, так и формулой. Таким образом, алгоритм ветвления состоит из условия и двух последовательностей команд.

Многократные повторения одних и тех же действий обеспечиваются при помощи циклических алгоритмов (рис. 14.12). В цикл входят в качестве составляющих элементов блок проверки условия и блок, называемый телом цикла. Перед началом цикла осуществляются операции присваивания начальных значений тем объектам, которые используются в теле цикла.

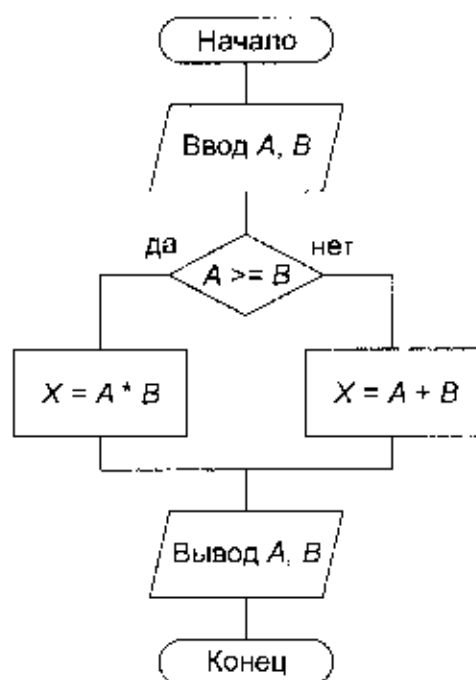


Рис. 14.11. Разветвляющийся алгоритм

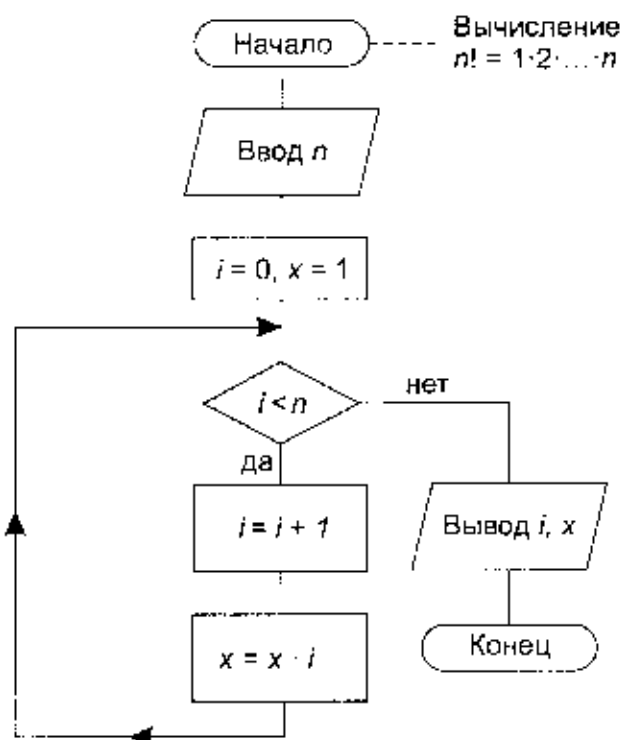


Рис. 14.12. Алгоритм с циклическим элементом

Циклический алгоритм — это алгоритм, в котором некоторая часть операций (тело цикла) выполняется многократно.

Слово «многократно» в определении циклического алгоритма не значит «до бесконечности». Организация циклов, не приводящих к остановке в выполнении алгоритма, является нарушением требования его результативности — получения результата за конечное число шагов.

Если тело цикла расположено после проверки условий (цикл с предусловием), то при определенных условиях тело цикла не выполнится ни разу. Такой вариант организации цикла, управляемый предусловием, называется *циклом с предусловием* (рис. 14.13).

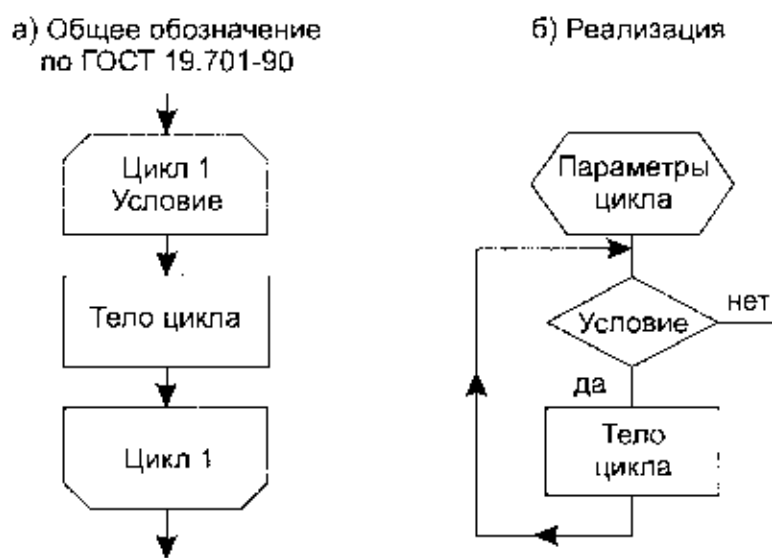


Рис. 14.13. Цикл с предусловием

Другой вариант построения цикла состоит в том, что тело цикла выполняется по крайней мере один раз, и будет повторяться до тех пор, пока условие не станет истинным. Такая организация цикла, когда его тело расположено перед проверкой условия, носит название *цикла с постусловием* (рис. 14.14). Истинность условия в этом случае является условием выхода из цикла.

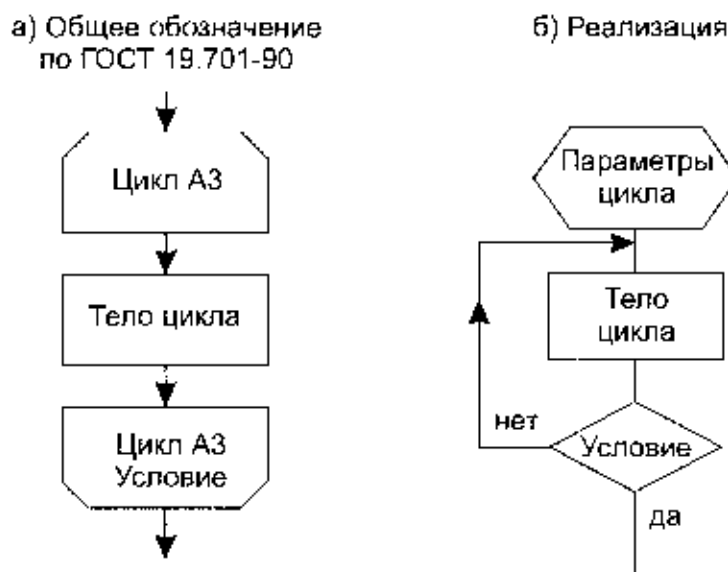


Рис. 14.14. Цикл с постусловием

Таким образом, цикл с предусловием завершается, когда условие ложно, а цикл с постусловием — когда условие становится истинным.

Третий тип циклов, в которых тело цикла *выполняется заданное число раз* (то есть еще до начала выполнения цикла точно известно, сколько раз он будет выполнен), реализуется с помощью счетчика. В цикле со счетчиком значения счетчика в теле цикла итеративно увеличиваются.

Процесс решения сложной задачи довольно часто сводится к решению нескольких более простых задач методом последовательной детализации. Соответственно, процесс разработки сложного алгоритма может разбиваться на этапы составления отдельных алгоритмов, которые называются вспомогательными. Каждый такой вспомогательный алгоритм описывает решение какой-либо подзадачи.

Процесс построения алгоритма методом последовательной детализации состоит в следующем. Сначала алгоритм формулируется в «обобщающих» блоках, которые могут быть непонятны исполнителю и записываются, как вызовы вспомогательных алгоритмов. Затем происходит детализация, и все вспомогательные алгоритмы детализируются до уровня команд, понятных исполнителю.

14.4. Представление и обработка данных разного типа

14.4.1. Общее представление о типах данных

Алгоритм, реализующий решение некоторой конкретной задачи, всегда работает с данными.

С точки зрения процесса обработки данные могут быть *входными (исходными)*, *выходными (результатирующими)* и *промежуточными*.

Данные подразделяют на переменные и константы.

Переменные — это данные, значения которых могут изменяться в процессе выполнения алгоритма.

Например, для алгоритма вычисления площади круга необходимы две переменные: переменная R , в которую будет заноситься значение радиуса окружности, и переменная S , в которую будет помещено значение площади круга, вычисленное по формуле $S = \pi R^2$. Число π — константа.

Константы — это данные, значения которых не меняются в процессе выполнения алгоритма.

Каждая переменная или константа в алгоритме должна иметь свое уникальное имя, которое задается идентификатором. *Идентификатор* представляет собой последовательность букв и цифр, начинающуюся с буквы.

С данными тесно связано понятие типа данных. Типы данных вводятся для того, чтобы использовать их в различных алгоритмах обработки. Следовательно, нужно иметь еще и множество операций, которые можно применять к данным того или иного типа.

Тип данных характеризует множество значений, к которым относится константа и которые может принимать переменная или выражение. Например, если переменная i в алгоритме может принимать только значение из множества целых чисел, то эта переменная относится к целому типу данных.

Тип данных — это характеристика данных, определяющая множество значений и операций, которые могут быть применены к этим данным, а также правила их выполнения. Приведенное определение допустимо при описании алгоритмов, однако при определении типа данных в конкретных языках программирования дополнительно считают, что тип данных также определяет внутреннее представление данных этого типа.

Типы данных принято делить на простые (базовые) и структурированные. К структурированным данным относят наборы данных, имеющие разное представление: массивы, символьные цепочки, деревья, графы.

Простой (базовый) тип данных — это тип используемой в алгоритме конкретной переменной или константы.

Структурированный тип данных — это набор однотипных или разнотипных данных, с которыми алгоритм может работать как с одной именованной переменной.

Наиболее общий метод получения структурных типов заключается в объединении элементов произвольных типов в одной переменной. Причем сами эти элементы могут быть в свою очередь структурными. Например, это могут быть комплексные числа, состоящие из вещественной и мнимой частей, или параметры точки, определяемые координатами и цветом.

14.4.2. Базовые типы данных

К базовым типам данных относятся следующие типы:

- **целые (INTEGER)** — подмножество допустимых значений из множества целых чисел;
- **вещественные (REAL)** — подмножество допустимых значений из множества вещественных чисел;
- **логические (BOOLEAN)** — множество допустимых значений состоит всего из двух значений, *истина* и *ложь*.
- **символьные (CHAR)** — буквы, цифры, знаки препинания и т. д.

Тип INTEGER задает подмножество целых чисел, диапазон которого зависит от такой характеристики компьютера, как размер машинного слова. Если для представления в компьютере целых чисел x используется n разрядов (причем один из них отводится под указание знака числа), то допустимые числа должны удовлетворять условию $-2^{n-1} < x < 2^{n-1}$ (Приведенный диапазон значений целых чисел в n -разрядном представлении соответствует одной из возможных форм кодирования. Чаще всего на практике используется другая форма кодирования, при которой модуль минимального отрицательного числа на единицу больше модуля

максимального положительного числа, т. е. $-(2^{n-1}) \leq x \leq (2^{n-1})-1$.) Считается, что все операции над данными этого типа выполняются точно и соответствуют обычным правилам арифметики. Если результат выходит за пределы допустимых значений, то вычисления прерываются. Такое событие называется переполнением.

Для целых чисел без знака может быть определен дополнительный стандартный тип — целое без знака (в некоторых языках программирования, кроме того, можно определить тип «натуральное число»), задающий подмножество целых неотрицательных чисел (соответственно, подмножество натуральных чисел). Если для представления целых без знака используется n разрядов, то переменным такого типа можно присваивать значения, удовлетворяющие условию $0 \leq x \leq 2^n - 1$ (соответственно, $0 < x \leq 2^n - 1$ или $0 < x \leq 2^n$, в зависимости от реализации).

Пример. Целые числа обычно применяют в тех случаях, когда необходимо записать некоторое значение, поддающееся подсчету: количество предметов, лет, дней, записей, слов. В алгоритмах тип INTEGER так же используют для подсчета итераций цикла или нумерации элементов массива.

Тип REAL обозначает подмножество вещественных чисел, границы изменения которых также определяются характеристиками конкретного компьютера. И если считается, что арифметика с данными типа INTEGER дает точный результат, то допускается, что аналогичные действия со значениями типа REAL могут быть неточными в пределах ошибок округления, вызванных вычислениями с конечным числом цифр. Это принципиальное различие между типами REAL и INTEGER.

Пример. Тип REAL обычно применяется для представления величин, которые можно измерить прямым измерением (веса, температуры), а также результатов деления, в общем, любых величин, имеющих дробную часть.

Два значения типа BOOLEAN обозначаются идентификаторами со значениями *истина* (TRUE) и *ложь* (FALSE). Операции над данными этого типа подчиняются правилам булевой алгебры.

Пример. Тип BOOLEAN играет важную роль при составлении алгоритмов: именно на основе значений, которые принимают переменные или выражения этого типа, осуществляется управление ходом выполнения программы (алгоритма).

В базовый тип CHAR входит множество печатаемых символов и символов-разделителей в соответствии с кодом ASCII.

Пример. Тип CHAR является основой всех простых и сложных типов, содержащих текст (строки, массивы и коллекции строк, записей).

К переменным типа REAL и INTEGER применимы арифметические операции, такие как сложение (+), вычитание (-), умножение (*) и деление (/), к переменным типа BOOLEAN — логические операции, к символьным переменным — операция конка-

тенации («склеивания») последовательностей символов. Операции отношения или сравнения применяются к данным любого типа, но правила их применения различны.

14.4.3. Представление и обработка данных в виде структур (массив, запись)

Наиболее известная структура данных — *массив*. Массив представляет собой упорядоченную структуру одноименных данных, которые называются *элементами массива*. Массив подходит для решения задач, в которых накапливаются и подлежат обработке данные одного и того же типа.

Доступ к каждому элементу массива осуществляется с помощью *индекса*. В общем случае индекс — это порядковый номер элемента в массиве.

Массивы могут быть как одномерными (адрес каждого элемента определяется значением одного индекса), так и многомерными (адрес каждого элемента определяется значением нескольких индексов).

Другим известным структурным типом данных является *запись*. Элементы записи называются *полями*. Примером данных такого типа может быть запись «Информация о студенте». Для представления такого типа в каждом языке программирования существуют встроенные средства языка, но, кроме того, такой тип (или переменная такого типа) может быть представлен графически (рис. 14.15).

Иванов	Петр	Николаевич	1	4	1990	2004	2239/22

Рис. 14.15. Графическое изображение записи данных в массиве

Из записей могут формироваться массивы. Такое сочетание двух этих структурных типов часто встречается при обработке данных, извлеченных из баз данных.

В качестве примера рассмотрим алгоритм обработки двумерного массива. Аналогом двумерного массива в математике является матрица. Общий вид матрицы A размерности $M \times N$ следующий:

$$A \equiv \{a_{ij}, i = \overline{1...M}, j = \overline{1...N}\} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M1} & a_{M2} & \dots & a_{MN} \end{bmatrix}.$$

Для матрицы размерностью $N \times N$ (квадратной матрицы) определены понятия главной и побочной диагоналей. Элементы главной диагонали — $\{a_{ii}, i = \overline{1...N}\}$, а элементы побочной — $\{a_{i(N-i+1)}, i = \overline{1...N}\}$.

Пример. Пусть необходимо построить матрицу, элемент которой a_{ij} вычисляется как $(i + j)^2$, а элементам главной диагонали присвоить значение 0.

Для решения поставленной задачи воспользуемся *двухмерным массивом целых чисел*. На первом этапе происходит построение вложенного цикла для расчета элементов и заполнения массива. На втором этапе элементам главной диагонали присваиваются нулевые значения.

Блок-схема такого алгоритма приведена на рис. 14.16 (N – размерность массива, I и J – индексы массива).

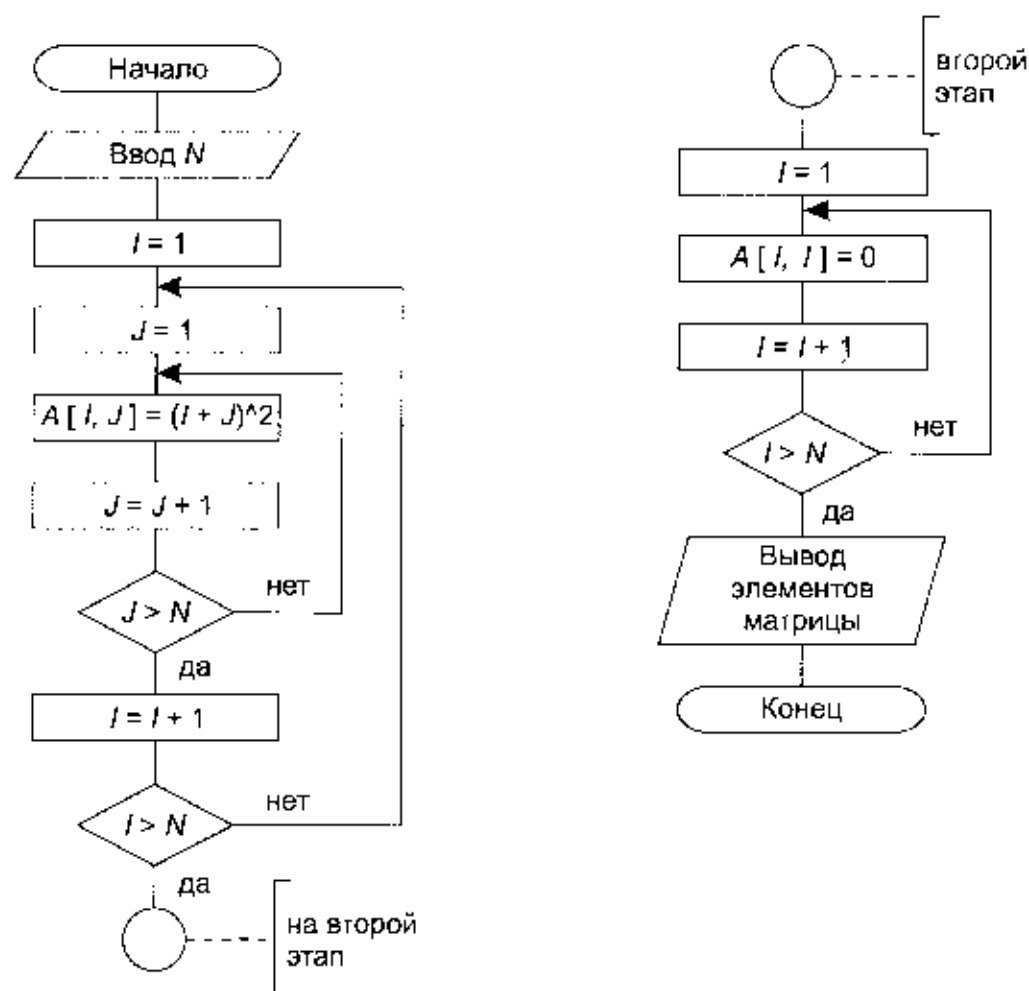


Рис. 14.16. Блок-схема алгоритма обработки двухмерного массива

14.4.4. Представление и обработка данных в виде символьных цепочек

Символьные цепочки с точки зрения физической организации представляют собой обыкновенные массивы типа CHAR. Особенность этого типа данных состоит в том, что символьный массив в подавляющем большинстве случаев является осмысленным текстом, а значит, имеет логическую структуру: разделен на слова, предложения, строки и абзацы. Поэтому алгоритмы обработки символьных цепочек почти всегда включают в себя поиск специальных символов-разделителей, разделение текста на логические единицы (слова, предложения, строки) и выполнение операций (подсчет, сортировка, поиск и замена) над этими единицами.

В качестве примера рассмотрим задание на определение количества слов в последовательности символов. Определим слово как часть последовательности (ненулевой длины), ограниченную с двух сторон (либо только справа или слева) символом пробела.

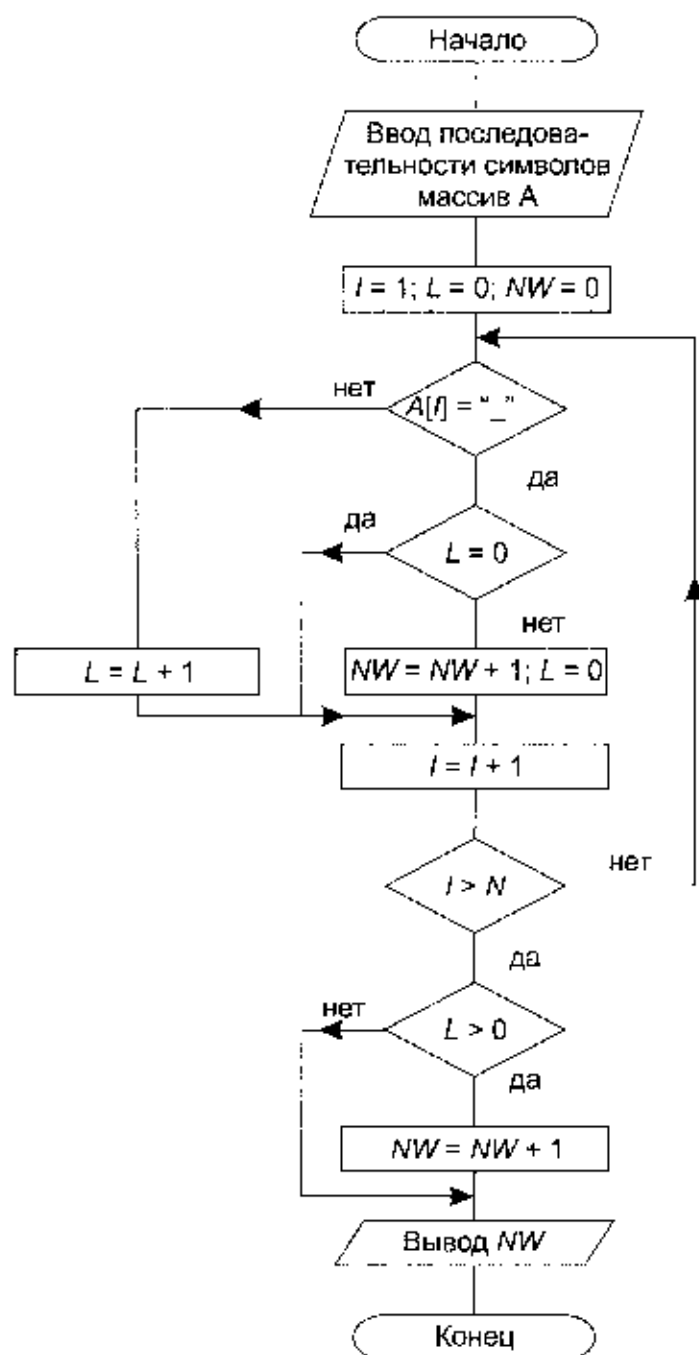


Рис. 14.17. Блок-схема алгоритма определения количества слов в последовательности символов

Для решения задачи разместим обрабатываемую последовательность в массиве размерностью N (где N – длина последовательности). При построении алгоритма необходимо учесть, что последовательность может начинаться или заканчиваться пробелами, причем в потоке символов может встретиться несколько смежных пробелов.

Блок-схема алгоритма приведена на рис. 14.17. Здесь L — длина очередного слова, NW — количество слов в последовательности, знак пробела обозначен символом подчеркивания ().

14.4.5. Представление и обработка данных в виде одно- и двусвязных списков

Структура данных, в которой объекты расположены в линейном порядке, называется связным списком. Однако в отличие от массива, в котором этот порядок определяется индексами, порядок в связном списке определяется указателями на объекты.

Элемент (узел) связного списка, помимо полей с данными, имеет поле *next*, в котором содержится указатель на следующий элемент списка. Если это последний элемент списка, поле *next* принимает нулевое значение. Помимо своих элементов, каждый список содержит указатель *head* на первый элемент списка. Если этот указатель равен 0, значит, список пуст. Поскольку этот указатель относится к списку в целом и не содержит в себе данных, на рис. 14.18, на котором схематически показан односвязный список, указатель не заключен в прямоугольник.

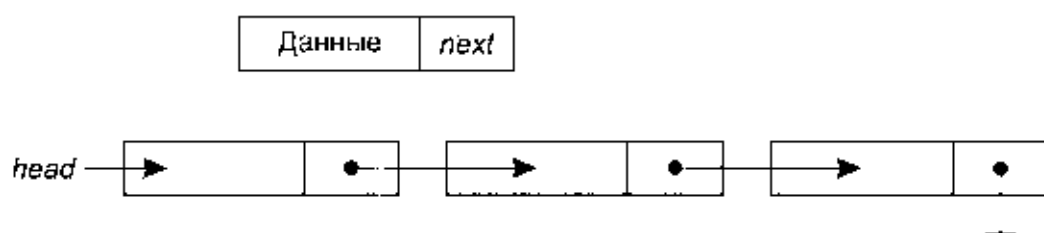


Рис. 14.18. Односвязный список

Односвязный список отличается тем, что пройти по нему можно только в одном направлении — от начала в конец списка. Это оказывается достаточно неудобно, поэтому гораздо большее распространение получили двусвязные списки (рис. 14.19), отличающиеся тем, что их узлы содержат по два указателя — на следующий (*next*) и на предыдущий (*prev*) элементы списка. Помимо указателя *head* на первый элемент списка, может существовать также указатель *tail* на последний элемент списка.

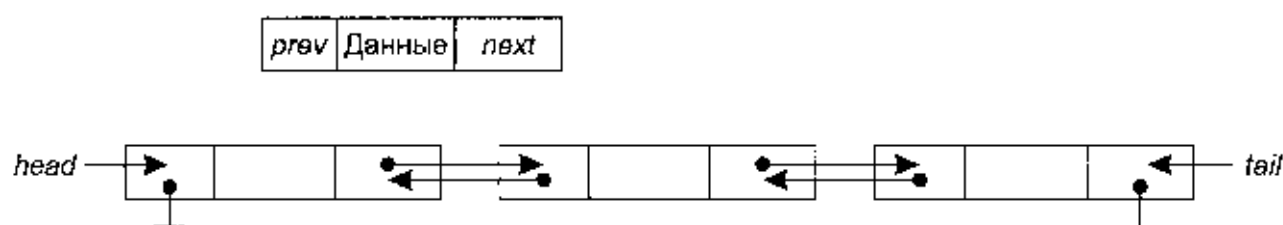


Рис. 14.19. Двусвязный список

Частный случай двусвязного списка — замкнутый (кольцевой) список, указатель *next* последнего элемента которого указывает на первый элемент, а указатель *prev* первого элемента — на последний элемент списка.

Главная особенность списка — быстрое выполнение операций вставки и удаления в произвольном месте списка. Эти операции требуют модификации указателей максимум у трех узлов: узла, над которым выполняется операция, и двух окружающих узлов. Схематично эти операции и изменения значений указателей показаны на рис. 14.20.

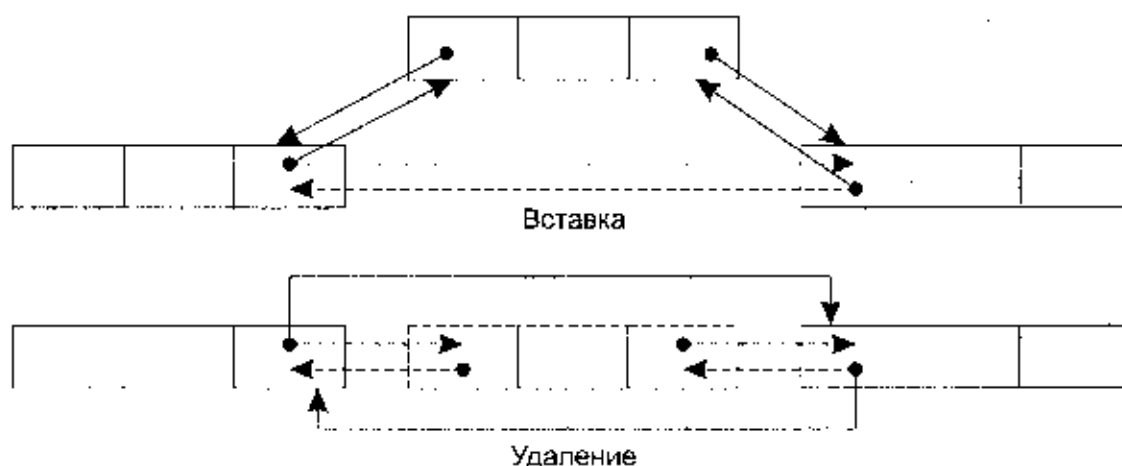


Рис. 14.20. Вставка и удаление в двусвязных списках

Алгоритм этих операций, приведенный на рис. 14.21 и 14.22, показывает, насколько простыми являются алгоритмические операции со списками.

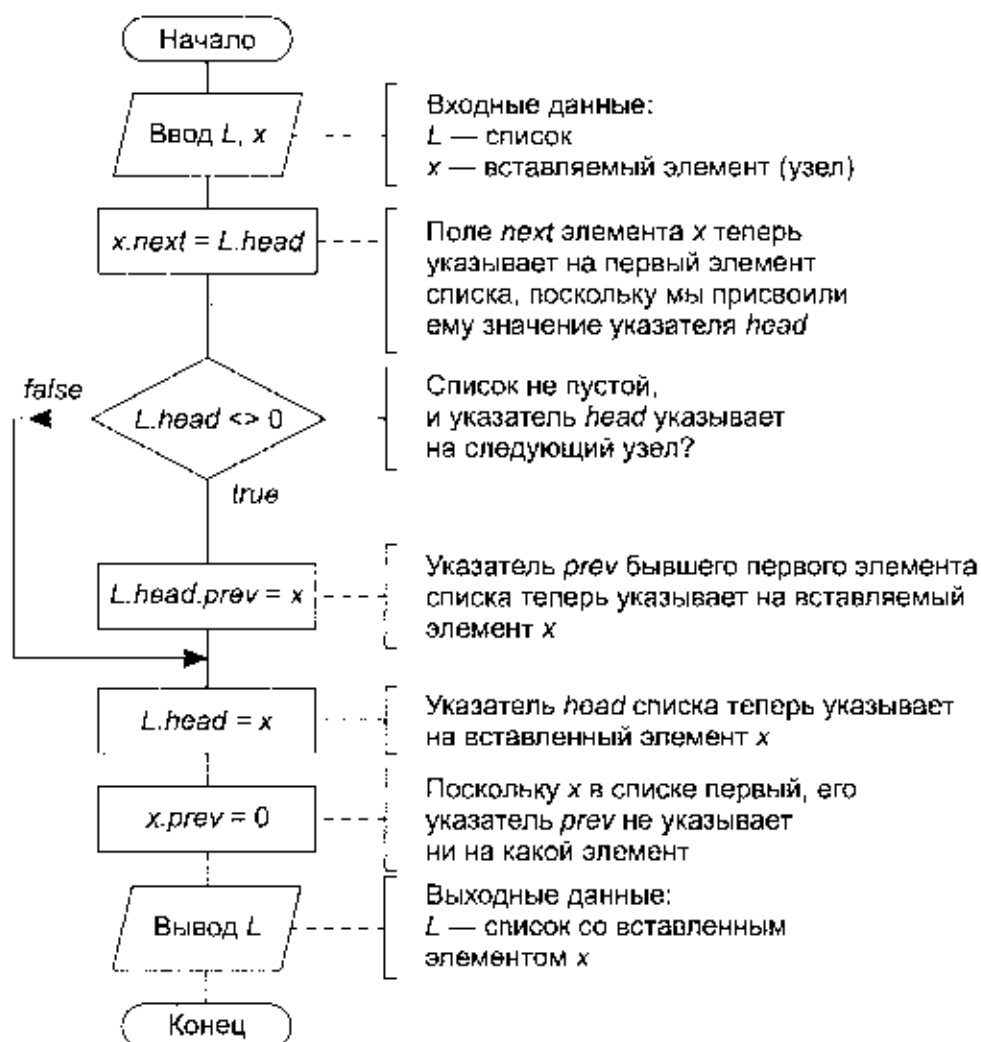


Рис. 14.21. Вставка элемента в начало списка

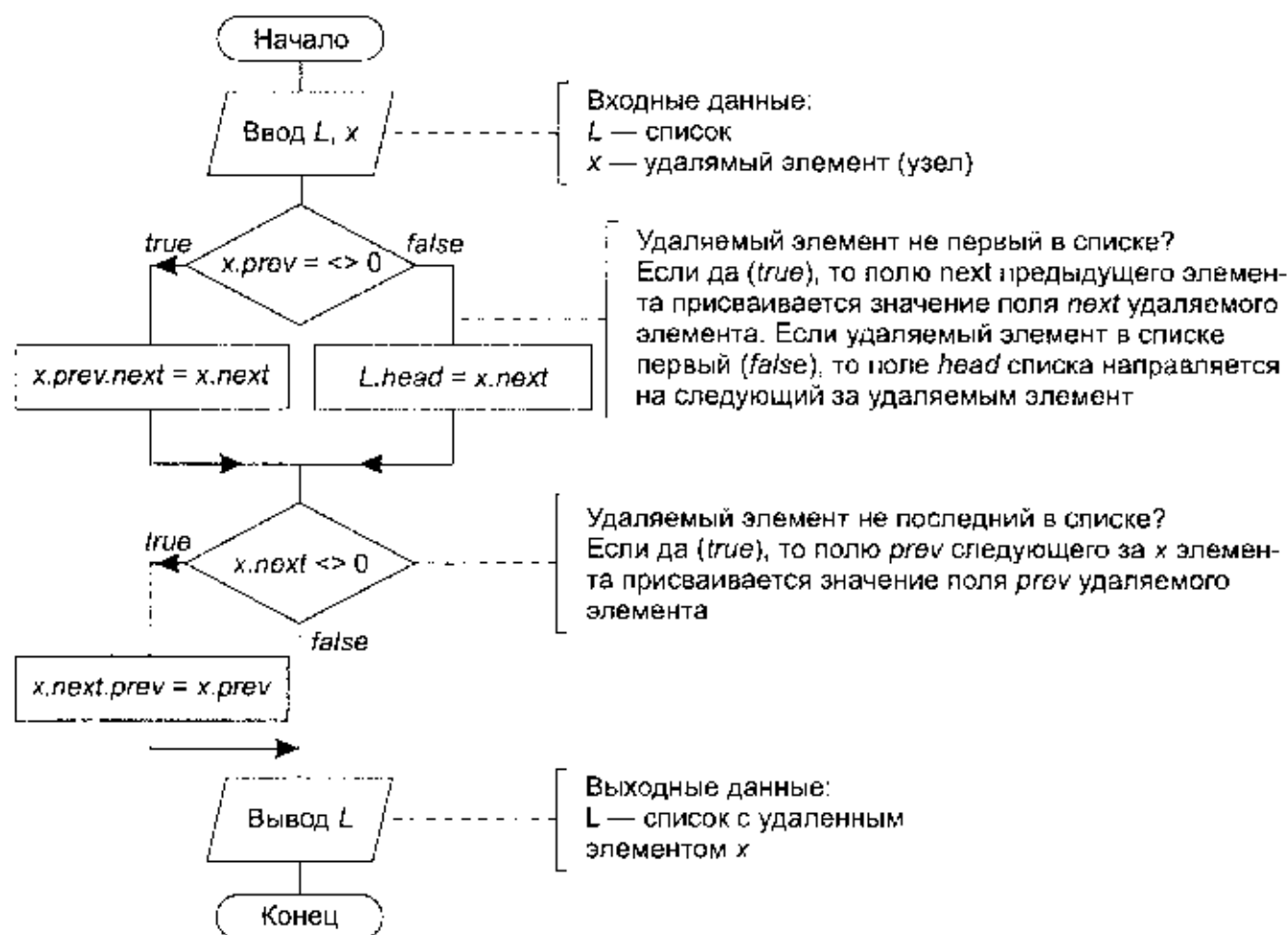


Рис. 14.22. Удаление элемента списка

14.4.6. Представление и обработка данных в виде деревьев

Элементы данных могут образовывать и более сложные структуры, чем линейный список. Часто данные, подлежащие обработке, образуют иерархическую структуру, которую необходимо отобразить в памяти компьютера и, соответственно, описать в структурах данных. Такая структура получила название *дерева*. Каждый элемент такой структуры, называемый *узлом*, может содержать ссылки на элементы более низкого уровня иерархии, а может быть, и на объект, находящийся на более высоком уровне иерархии. Узел, находящийся на самом верхнем уровне иерархии, называется *корневым*.

Дерево — это множество узлов, связанных таким отношением, в котором каждому из узлов, кроме корневого, можно сопоставить один узел, являющийся его предком (родительский узел).

Корень дерева — это единственный узел, не имеющий непосредственного предка.

Имеется множество типов деревьев. Важнейшим с точки зрения информатики подмножеством структуры типа дерева является подмножество *бинарных деревьев поиска*. У бинарного дерева каждый узел имеет не более двух дочерних узлов, причем левый и правый узлы различаются. Каждый узел содержит несколько полей: поля значения, хранящегося в узле, и полей, указывающих на левый и правый потомки данного узла, а также на родительский узел. Бинарным деревом поиска его

делает следующее свойство: значения в узлах дерева располагаются таким образом, что в любой момент для любого узла x значения всех узлов в его левом поддереве не превышают значения узла x , а значения всех узлов в его правом поддереве не меньше значения узла x . На рис. 14.23 показано несколько деревьев. Числа в кружках отражают значения данных, хранящихся в узлах дерева. Дерево на рис. 14.23, *а* не является бинарным, так как у узла 5 есть три дочерних узла. Дерево на рис. 14.23, *б* является бинарным, но не является деревом поиска, так как узел 2 является правым дочерним узлом по отношению к узлу 3, нарушая свойство дерева поиска. Дерево на рис. 14.23, *в* представляет собой корректное бинарное дерево поиска.

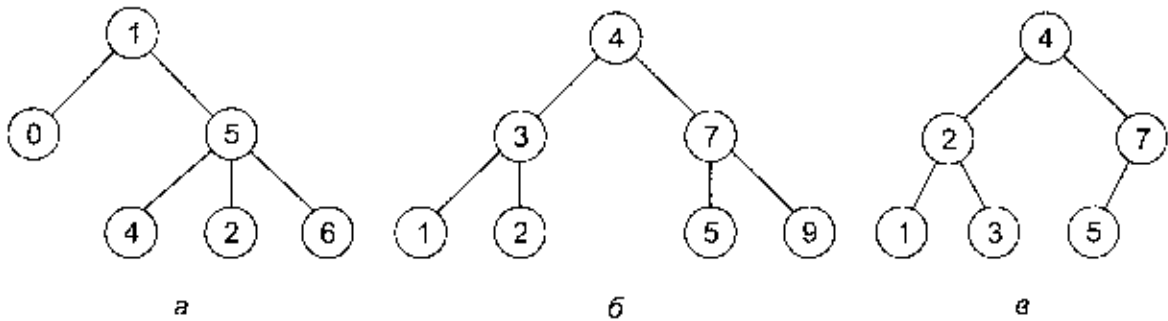


Рис. 14.23. Виды деревьев

На рис. 14.24 изображено возможное представление бинарного дерева поиска с использованием полей указателей. На этом рисунке, как и на предыдущем, цифры представляют собой значения, хранящиеся в каждом из элементов дерева, а стрелки показывают, как при помощи указателей каждый узел дерева связан со своими правым и левым поддеревьями, а также с родительским узлом. Так, для узла со значением 2 буквой *а* помечен указатель, связывающий его с родительским узлом, а буквами *б* и *с* — указатели, связывающие этот узел, соответственно, с его левым и правым поддеревьями.

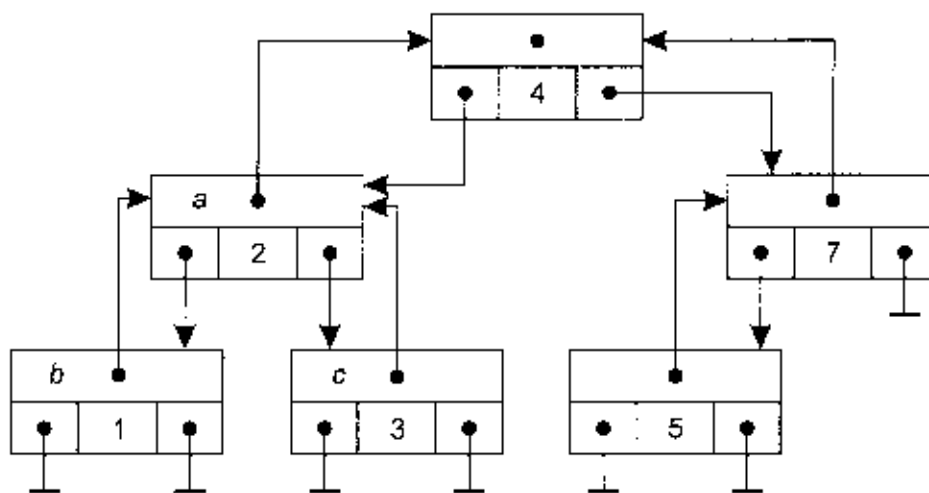


Рис. 14.24. Бинарное дерево поиска

Посетить все узлы дерева очень легко с помощью рекурсивной процедуры обхода. Основной вариант обхода бинарного дерева — симметричный обход дерева, когда для каждого узла сначала рекурсивно выполняется посещение его левого поддерева, затем — самого узла, а после этого — узлов его правого поддерева. Для дерева на рис. 14.24 симметричный обход дает следующую последовательность

узлов: 1, 2, 3, 4, 5, 7. Как видите, таким образом можно получить отсортированную последовательность значений узлов.

Два других распространенных метода обхода дерева — обход в прямом порядке, при котором сначала выводится корень, а потом значение левого и правого поддеревьев (для уже рассматривавшегося дерева это даст последовательность значений 4, 2, 1, 3, 7, 5), и обход в обратном порядке, при котором сначала выводятся значения узлов левого и правого поддеревьев, а затем — корня (для нашего дерева это даст последовательность 1, 3, 2, 5, 7, 4).

Псевдокоды описанных обходов дерева очень просты. Алгоритм процедуры симметричного обхода представлен на рис. 14.25.

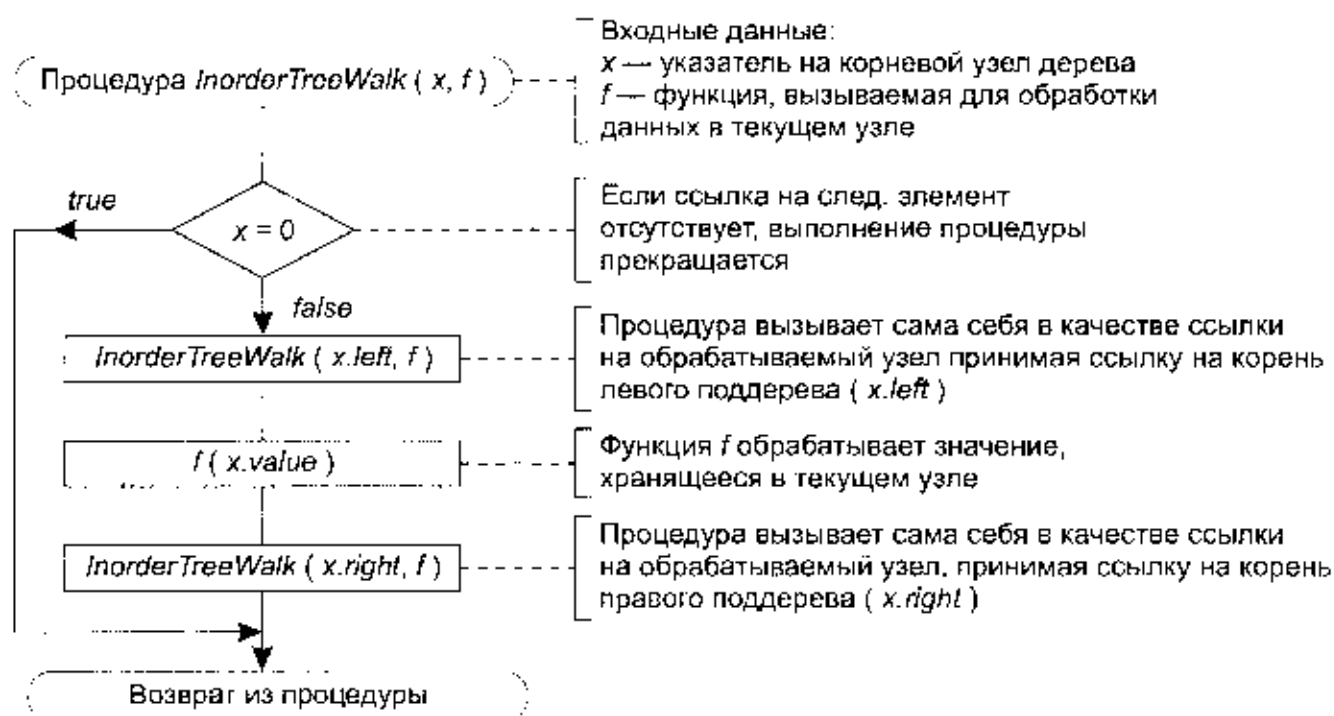


Рис. 14.25. Процедура симметричного обхода бинарного дерева

В этой процедуре для обхода всех узлов дерева и выполнения над ними определенных операций используется рекурсия.

Рекурсия часто применяется в алгоритмах обработки различного рода сложно структурированных объектов, в частности документов в формате XML, когда количество дочерних узлов и уровень их вложенности заранее не известны.

Рекурсия (в программировании) — вид процедуры или функции, во время выполнения которой процедура или функция вызывает саму себя.

Основные операции при работе с бинарным деревом поиска — поиск в нем определенного значения, а также поиск наименьшего и наибольшего элементов дерева и поиск предшествующего и последующего элементов для данного.

Выполнение операции поиска основано на том, что находясь на определенной вершине, всегда можно однозначно указать, в каком из поддеревьев находится искомое значение (если таковое имеется в данном дереве), так как согласно свойству бинарного дерева поиска все значения узлов в левом поддереве не больше, а в правом — не меньше значения в корне.

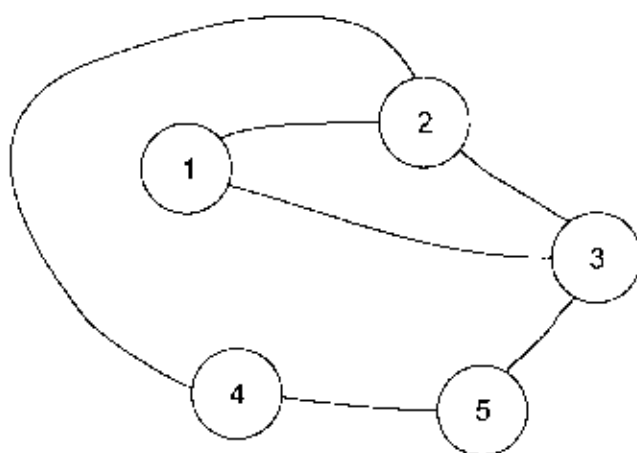
Что касается поиска наименьшего и наибольшего элементов в бинарном дереве поиска, то из свойства бинарного дерева поиска очевидно: чтобы достичь наименьшего (наибольшего) элемента, надо двигаться по левым (или, соответственно, правым) ветвям дерева до тех пор, пока это возможно.

14.4.7. Представление и обработка данных в виде графов

Графы формально описывают множество близких ситуаций. Самым привычным примером служит карта автодорог, на которой изображены перекрестки и связывающие их дороги. Перекрестки являются вершинами графа, а дороги — его ребрами. Графы могут быть ориентированы (подобно улицам с односторонним движением) или взвешены, когда каждой дороге приписана стоимость путешествия по ней (если, например, дороги платные).

С формальной точки зрения граф представляет собой упорядоченную пару множества $G = (V, E)$, первое из которых состоит из *вершин (узлов)* графа, а второе — из его *ребер*. Ребро связывает между собой две вершины. При работе с графами часто решается вопрос, как проложить путь из ребер от одной вершины графа к другой. В этом случае говорят о *движении по ребру*, что означает переход из вершины A графа в другую вершину B , связанную с ней ребром AB (ребро графа, связывающее две вершины, для краткости обозначается этой парой вершин). В этом случае говорят, что ребро A *примыкает* к ребру B или что эти две вершины *соседние*.

Граф может быть ориентированным или нет. Ребра неориентированного графа, чаще всего называемого просто графом, можно проходить в обоих направлениях (рис. 14.26). В этом случае ребро — это неупорядоченная пара вершин, его концов.



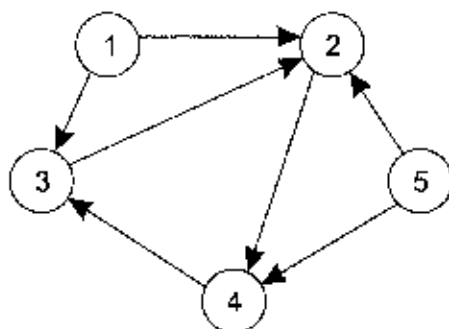
Граф $G = (\{1, 2, 3, 4, 5\}, \{\{1, 2\}, \{1, 3\}, \{2, 4\}, \{3, 5\}, \{2, 3\}, \{4, 5\}\})$

Рис. 14.26. Неориентированный граф

В ориентированном графе, или *орграфе*, ребра представляют собой упорядоченные пары вершин: первая вершина — это начало ребра, вторая — его конец (рис. 14.27).

Полный граф — это граф, в котором каждая вершина соединена со всеми остальными. Количество ребер в полном графе без петель с N вершинами равно $(N^2 - N)/2$. В полном ориентированном графе разрешается переход из любой вершины в любую другую. Поскольку в графе переход по ребру разрешается в обоих направ-

лениях, а переход по ребру в орграфе — только в одном, в полном орграфе в два раза больше ребер, то есть их количество равно $N^2 - N$.



Ориентированный
граф $G = (\{1, 2, 3, 4, 5\},$
 $\{(1, 2), (1, 3), (2, 4), (3, 2),$
 $(4, 3), (5, 2), (5, 4)\})$

Рис. 14.27. Ориентированный граф

Подграф (V_s, E_s) графа или орграфа (V, E) состоит из некоторого подмножества вершин, $V_s \subset V$, и некоторого подмножества ребер, их соединяющих, $E_s \subset E$.

Путь в графе или орграфе — это последовательность ребер, по которым можно поочередно проходить. Другими словами, путь из вершины A в вершину B начинается в A и проходит по набору ребер до тех пор, пока не будет достигнута вершина B . С формальной точки зрения путь из вершины V_x в вершину V_y — это последовательность ребер графа:

$$v_x v_{x+1}, v_{x+1} v_{x+2}, \dots, v_{y-1} v_y.$$

Необходимо, чтобы любая вершина встречалась на таком пути не более, чем однажды. У всякого пути есть *длина* — количество ребер в нем. Длина пути AB , BC , CD , DE равна 4.

Граф или орграф называется *связным*, если всякую пару узлов можно соединить по крайней мере одним путем. *Цикл* — это путь, который начинается и кончается в одной и той же вершине. В *ациклическом* графе или орграфе циклы отсутствуют. *Связный ациклический* граф называется *неукорененным деревом*. Структура неукорененного дерева такая же, как и у дерева, только в нем не выделен корень. Однако каждая вершина неукорененного дерева может служить его корнем.

Информацию о графах или орграфах в пригодном для обработки компьютерными алгоритмами виде можно хранить двумя способами: в виде *матрицы примыканий* и в виде *списка примыканий*¹.

Матрица примыканий обеспечивает *быстрый доступ* к информации о ребрах графа, однако, если в графе мало ребер, эта матрица будет содержать гораздо больше пустых клеток, чем заполненных. *Длина списка примыканий* пропорциональна числу ребер в графе, однако при пользовании списком время получения информации о ребре увеличивается.

¹ В определенных конкретных случаях может потребоваться представление и в других формах, также пригодных к обработке на компьютере: в виде матриц инцидентности или матриц смежности, а также других структур для представления «веса» и «цвета» связей, пометок вершин и т. д.

Матрица примыканий $ConMat$ графа $G = (V, E)$ с числом вершин $|V| = N$ записывается в виде двумерного массива размером $N \times N$. В каждой ячейке $[ij]$ этого массива записано значение 0 за исключением лишь тех случаев, когда из вершины v_i в вершину v_j ведет ребро, и тогда в ячейке записано значение 1:

$$ConMat[i, j] = \begin{cases} 1, & \text{если } \{v_i v_j\} \in E \\ 0, & \text{если } \{v_i v_j\} \notin E \end{cases} \quad \text{для всех } i \text{ и } j \text{ от } 1 \text{ до } N.$$

На рис. 14.28 изображены матрицы примыканий для графа и орграфа, изображенных на рис. 14.26 и 14.27.

	1	2	3	4	5		1	2	3	4	5
1	0	1	1	0	0	1	0	1	1	0	0
2	1	0	1	1	0	2	0	0	0	1	0
3	1	1	0	0	1	3	0	1	0	0	0
4	0	1	0	0	1	4	0	0	1	0	0
5	0	0	1	1	0	5	0	1	0	1	0

Рис. 14.28. Матрицы примыканий графа и орграфа

Список примыканий $ConList$ графа $G = (V, E)$ с числом вершин $|V| = N$ записывается в виде одномерного массива длины N , каждый элемент которого представляет собой ссылку на список. Такой список приписан каждой вершине графа и содержит по одному элементу на каждую вершину графа, соседнюю с данной.

На рис. 14.29 изображен список примыканий для орграфа, изображенного на рис. 14.27. На этом рисунке буквами L_{v_i} обозначены вершины графа, а буквами v_N — те вершины, к которым направлены выходящие из данной вершины ребра.

$L_{v_1}: v_2, v_3;$
 $L_{v_2}: v_4;$
 $L_{v_3}: v_2;$
 $L_{v_4}: v_3;$
 $L_{v_5}: v_2, v_4;$

Рис. 14.29. Список примыканий для орграфа

При работе с графами часто приходится выполнять некоторое действие по одному разу с каждой из вершин графа, например, когда некоторую порцию информации следует передать каждому из компьютеров в сети. При этом нерационально посещать какой-либо компьютер дважды. Аналогичная ситуация возникает, если нужно собирать, а не распространять информацию.

Подобный обход можно совершать двумя способами. При *обходе в глубину* проход по выбранному пути осуществляется настолько глубоко, насколько это возможно, а при *обходе по уровням* происходит равномерное движение вдоль всех возможных направлений.

Обход в глубину

При обходе в глубину происходит посещение первого узла, а затем передвижение вдоль ребер графа, пока не будет достигнут тупик. Узел неориентированного графа является *тупиком*, если уже посещены все примыкающие к нему узлы. В ориентированном графе тупиком также оказывается узел, из которого нет выходящих ребер.

После попадания в тупик следует вернуться назад вдоль пройденного пути, пока не будет обнаружена вершина, у которой есть еще не посещенный сосед, а затем двигаться в новом направлении. Процесс оказывается завершенным после того, как произошел возврат в отправную точку, а все примыкающие к ней вершины уже посещены (рис. 14.30).

При реализации алгоритма обхода в глубину используется рекурсивный вызов процедуры.

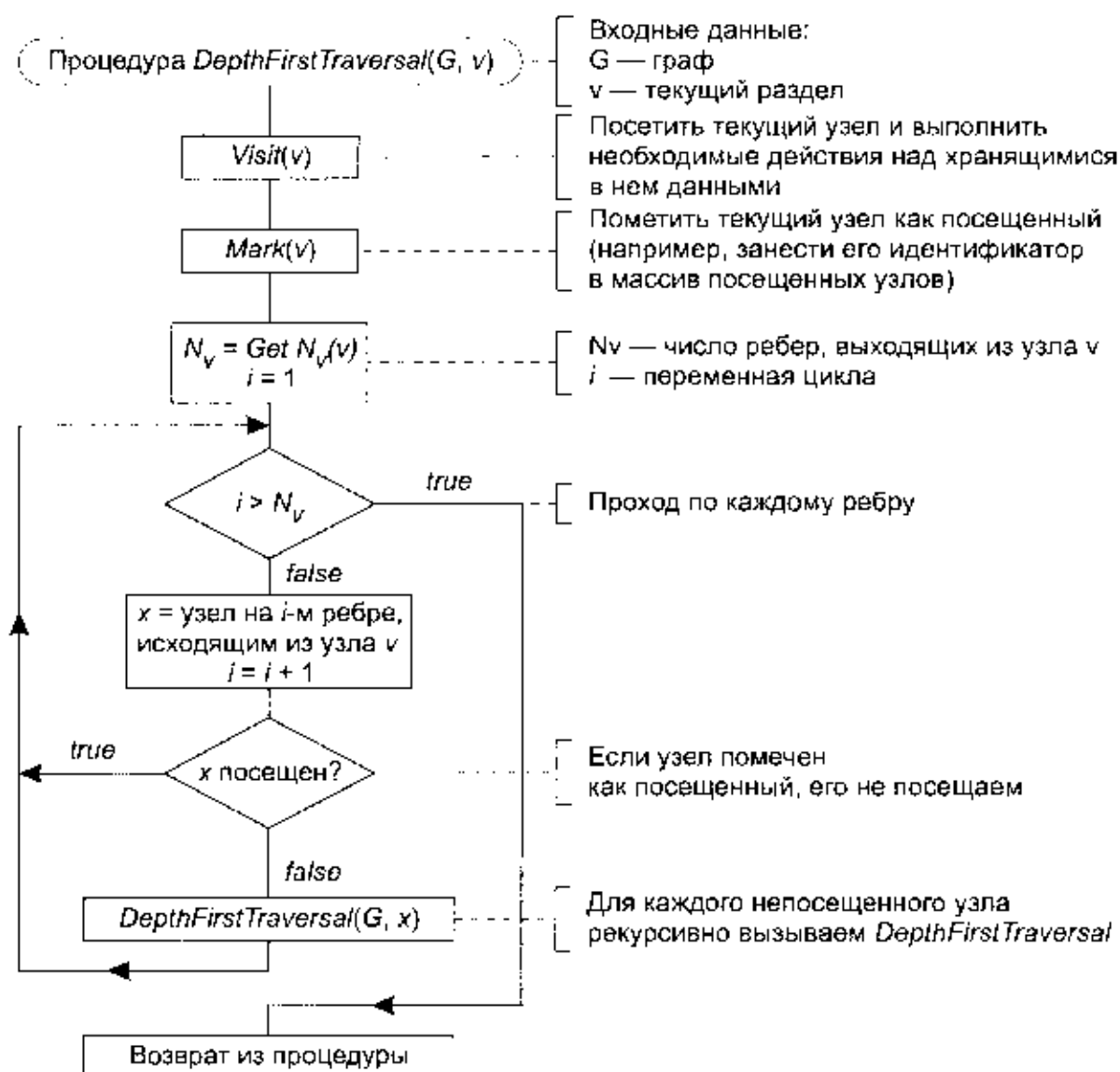


Рис. 14.30. Рекурсивная процедура обхода графа в глубину

Обход по уровням

При обходе графа по уровням после посещения первого узла происходит обход всех соседних с ним вершин. При втором проходе посещаются все вершины

на расстоянии «двух ребер» от начальной. При каждом новом проходе обходятся вершины, расстояние от которых до начальной на единицу больше предыдущего. В графе могут быть циклы, поэтому не исключено, что одну и ту же вершину можно соединить с начальной двумя различными путями. При обходе этой вершины впервые проходится самый короткий до нее путь, и посещать ее второй раз нет необходимости. Поэтому, чтобы предупредить повторное посещение, приходится либо вести список посещенных вершин, либо сопоставить каждой вершине флажок, указывающий, посещена она или нет.

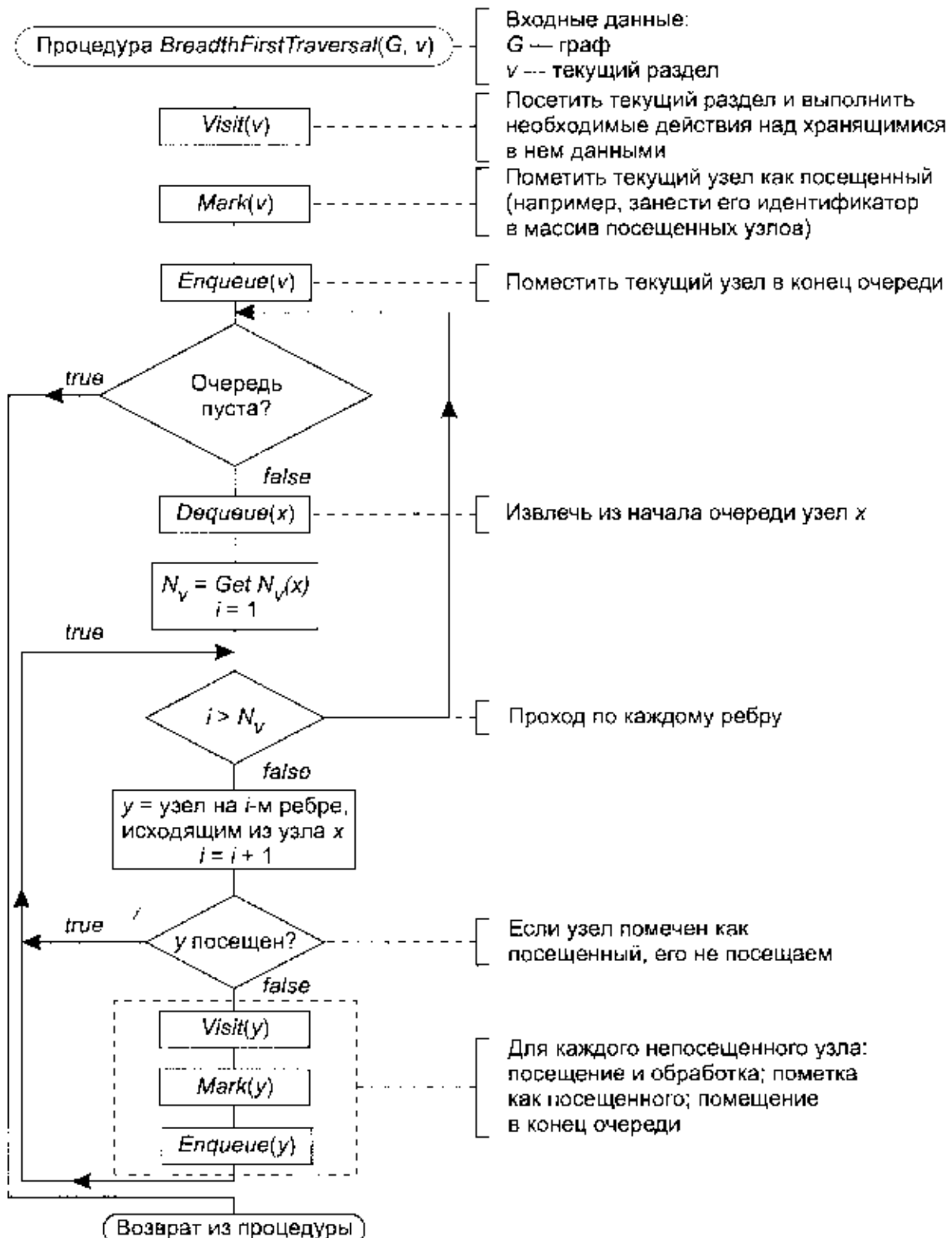


Рис. 14.31. Обход графа по уровням

Алгоритм обхода вершин графа по уровням приведен на рис. 14.31. В отличие от обхода в глубину, этот алгоритм реализован с использованием очереди. Очередь — это специальный программный механизм, предназначенный для упорядоченного хранения и извлечения данных. При помощи процедуры *Enqueue* заданный объект помещается в конец очереди, а процедура *Dequeue* извлекает из начала очереди размещенный там объект. Таким образом, пока идет обработка данных в узлах текущего уровня, узлы следующего уровня, связанные ребрами с текущими узлами, помещаются в очередь.

При анализе графов и применении их для решения различных информационных задач используют стандартные алгоритмы *поиска минимального пути, поиска пути с минимальным (максимальным) весом ребер, анализа компонентов двусвязности*.

14.5. Алгоритмы сортировки и поиска

14.5.1. Сортировка

Сортировка (то есть упорядочение по возрастанию или убыванию значений в списках, массивах, записях и других последовательностях элементов) является одной из наиболее важных алгоритмических задач, непосредственно связанных с проблемами эффективного управления, поиска и извлечения информации.

Известны следующие распространенные методы сортировки:

- сортировка вставками;
- пузырьковая сортировка;
- сортировка Шелла;
- корневая сортировка;
- пирамидальная сортировка;
- сортировка слиянием;
- быстрая сортировка.

Сортировка вставками

Основная идея сортировки вставками состоит в том, что при добавлении нового элемента в уже отсортированный список его сразу вставляют в нужное место вместо того, чтобы вставлять его в произвольное место, а затем заново сортировать весь список. При сортировке вставками первый элемент любого списка считается отсортированным списком длины 1. Двухэлементный отсортированный список создается добавлением второго элемента исходного списка в нужное место одноэлементного списка, содержащего первый элемент. После этого можно вставить третий элемент исходного списка в отсортированный двухэлементный список. Этот процесс повторяется до тех пор, пока все элементы исходного списка не окажутся в расширяющейся отсортированной части списка.

Пузырьковая сортировка

Основной принцип пузырьковой сортировки состоит в выталкивании меньших значений на вершину списка, в то время как большие значения опускаются вниз.

Алгоритм пузырьковой сортировки приведен на рис. 14.32. У пузырьковой сортировки есть много различных вариантов. В алгоритме пузырьковой сортировки совершается несколько проходов по списку. При каждом проходе происходит сравнение соседних элементов. Если порядок соседних элементов неправильный, то они меняются местами. Каждый проход начинается с начала списка. Сперва сравниваются первый и второй элементы, затем — второй и третий, потом — третий и четвертый, и т. д.; элементы с неправильным порядком в паре переставляются. При обнаружении на первом проходе наибольшего элемента списка он переставляется со всеми последующими элементами, пока не дойдет до конца списка. Поэтому при втором проходе нет необходимости производить сравнение с последним элементом.

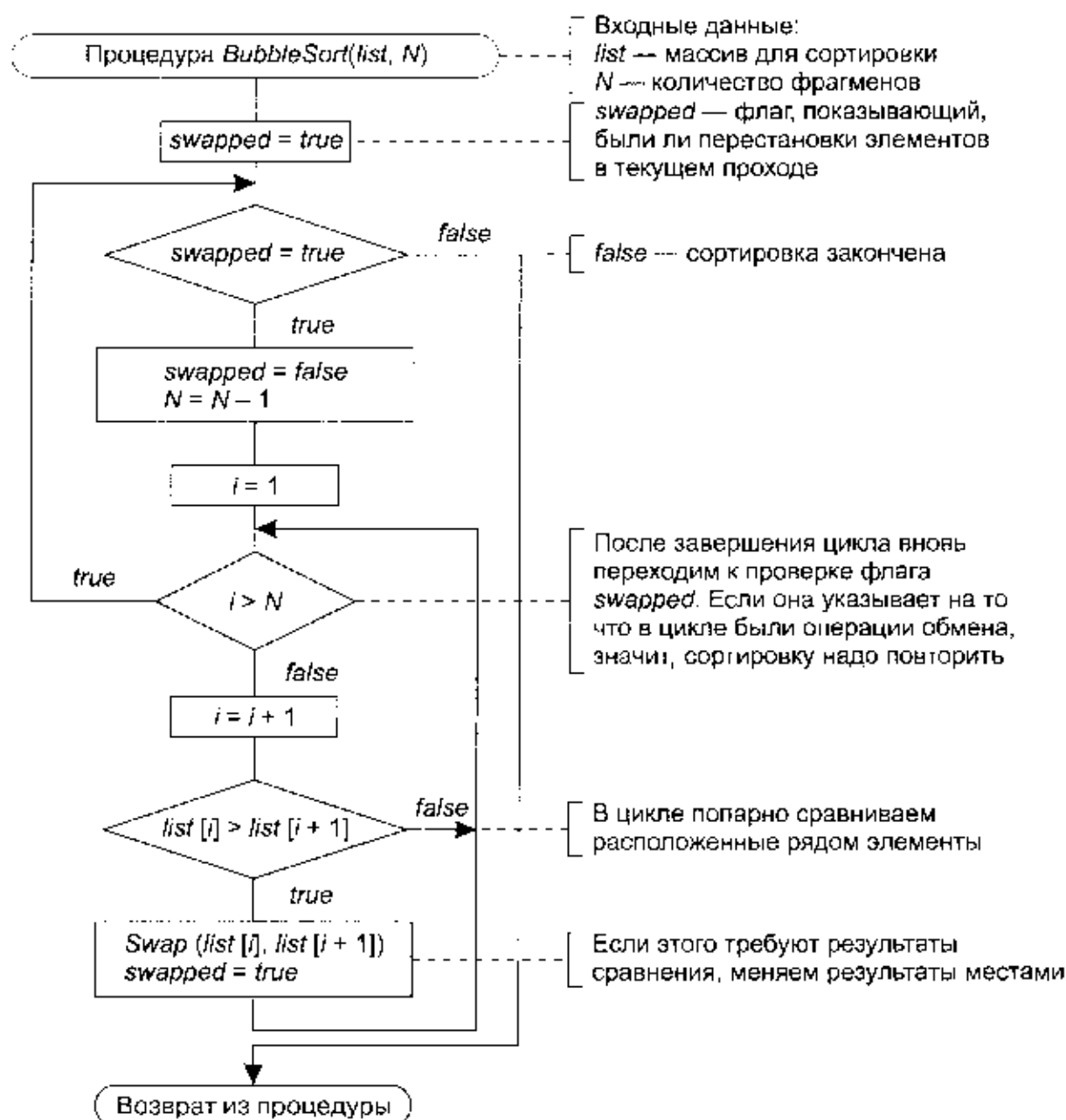


Рис. 14.32. Алгоритм пузырьковой сортировки

При втором проходе второй по величине элемент списка опускается во вторую позицию с конца. При продолжении процесса на каждом проходе по крайней мере

одно из следующих по величине значений встает на свое место. При этом меньшие значения тоже собираются наверх. Если при каком-то проходе не происходит ни одной перестановки элементов, это означает, что все они стоят в нужном порядке, и исполнение алгоритма можно прекратить. Стоит заметить, что при каждом проходе ближе к своему месту продвигается сразу несколько элементов, хотя гарантированно занимает окончательное положение лишь один.

Сортировка Шелла

Суть сортировки Шелла состоит в том, что весь список рассматривается как совокупность перемешанных подсписков. На первом шаге эти подписки представляют собой просто пары элементов. На втором шаге в каждой группе оказывается по четыре элемента. При повторении процесса количество элементов в каждом подписке увеличивается, а количество подписков, соответственно, падает.

Корневая сортировка

При корневой сортировке упорядочивание списка происходит без непосредственного сравнения ключевых значений между собой. При этом создается набор «стопок», а элементы распределяются по стопкам в зависимости от значений ключей. Собрав значения обратно и повторив всю процедуру для последовательных частей ключа, можно получить отсортированный список. Чтобы такая процедура работала, распределение по стопкам и последующую сборку следует выполнять очень аккуратно.

Пирамидальная сортировка

В основе пирамидальной сортировки лежит специальный тип бинарного дерева, называемый пирамидой; значение корня в любом поддереве такого дерева больше, чем значение каждого из его потомков. Непосредственные потомки каждого узла не упорядочены, поэтому иногда левый непосредственный потомок оказывается больше правого, а иногда наоборот. Пирамида представляет собой полное дерево, в котором заполнение нового уровня начинается только после того, как предыдущий уровень оказывается заполненным целиком, а все узлы на одном уровне заполняются слева направо.

Сортировка начинается с построения пирамиды. При этом максимальный элемент списка оказывается в вершине дерева; ведь потомки вершины обязательно должны быть меньше. Затем корень записывается последним элементом списка, а пирамида с удаленным максимальным элементом переформируется. В результате в корне оказывается второй по величине элемент, он копируется в список, и процедура повторяется, пока все элементы не окажутся возвращенными в список.

Сортировка слиянием

Сортировка слиянием — один из распространенных алгоритмов рекурсивной сортировки. В ее основе лежит замечание, согласно которому слияние двух от-

сортированных списков выполняется быстро. Список из одного элемента уже отсортирован, поэтому при сортировке слиянием список разбивается на одноэлементные куски, которые затем постепенно сливаются. Поэтому вся деятельность заключается в слиянии двух списков.

Быстрая сортировка

Быстрая сортировка — один из наиболее популярных алгоритмов рекурсивной сортировки. При быстрой сортировке после выбора в списке элемента список с его помощью делится на две части. В первую часть попадают все элементы, меньшие выбранного, во вторую — большие выбранного. В среднем такая сортировка эффективна, однако в наихудшем случае ее эффективность такая же, как у сортировки вставками и у пузырьковой сортировки.

При быстрой сортировке в списке выбирается элемент, называемый *осевым*, а затем список переупорядочивается таким образом, что все элементы, меньшие осевого, оказываются перед ним, а большие элементы — за ним. В каждой из частей списка элементы не упорядочиваются. Если n — окончательное положение осевого элемента, то нам известно лишь, что все значения в позициях с первой по $n - 1$ меньше осевого, а значения с номерами от $n + 1$ до N — больше осевого. Далее алгоритм вызывается рекурсивно на каждой из двух частей.

14.5.2. Поиск

Под алгоритмами поиска понимается процесс просмотра списка в поисках некоторого конкретного элемента, называемого *целевым*. При последовательном поиске всегда предполагается, что список не отсортирован, поскольку некоторые алгоритмы на отсортированных списках показывают лучшую производительность. Обычно поиск производится не просто для проверки того, что нужный элемент в списке имеется, а и для того, чтобы получить данные, относящиеся к этому значению ключа. Например, ключевое значение может быть номером сотрудника, порядковым номером или любым другим уникальным идентификатором. После того как нужный ключ найден, программа может, например, частично изменить связанные с ним данные или просто вывести всю запись. Во всяком случае, перед алгоритмом поиска стоит важная задача определения местонахождения ключа. Поэтому алгоритмы поиска возвращают индекс записи, содержащей нужный ключ. Если ключевое значение не найдено, то алгоритм поиска обычно возвращает значение индекса, превышающее верхнюю границу массива. Положим, что элементы списка имеют номера от 1 до N . Это позволит возвращать 0 в случае, если целевой элемент отсутствует в списке. Для простоты предполагаем, что ключевые значения не повторяются.

Последовательный поиск

В алгоритме последовательного поиска последовательно просматривается по одному элементу списка, начиная с первого, до тех пор, пока не будет найден целевой элемент. Очевидно, что чем дальше в списке находится конкретное значение

ключа, тем больше времени уйдет на его поиск. Это следует помнить при анализе алгоритма последовательного поиска.

Полный алгоритм последовательного поиска показан на рис. 14.33.

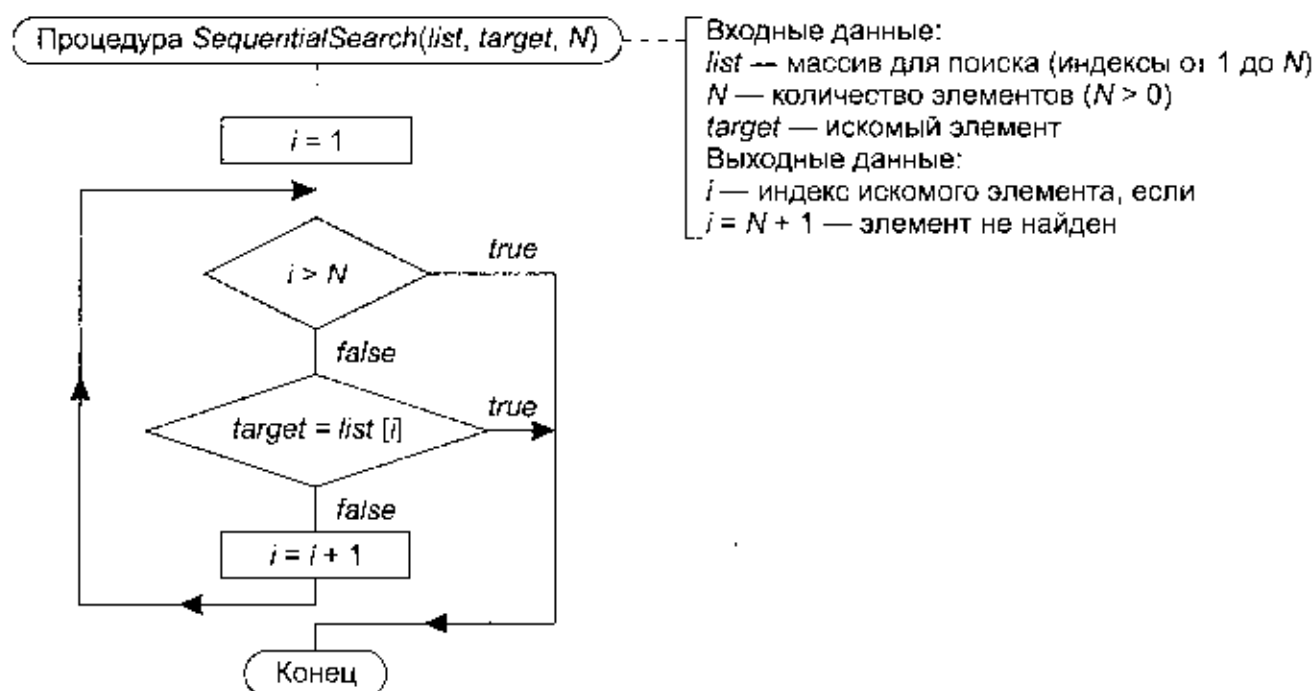


Рис. 14.33. Алгоритм последовательного поиска

Двоичный поиск

При сравнении целевого значения со средним элементом отсортированного списка возможен один из трех результатов: значения равны, целевое значение меньше элемента списка, целевое значение больше элемента списка. В первом, и наилучшем, случае поиск завершен. В остальных двух случаях можно отбросить половину списка. Когда целевое значение меньше среднего элемента, известно, что если оно имеется в списке, то находится перед этим средним элементом. Когда же оно больше среднего элемента, известно, что если оно имеется в списке, то находится после этого среднего элемента. Этой информации достаточно, чтобы стало возможно одним сравнением отбросить половину списка. При повторении этой процедуры можно отбросить половину оставшейся части списка. Очевидно, что такой алгоритм будет работать только на предварительно отсортированном списке.

Вопросы для самопроверки

1. Откуда произошел термин «алгоритм»?
2. Чем численный алгоритм отличается от логического?
3. Из каких разделов состоит современная теория алгоритмов?
4. Дайте определение алгоритма по Маркову.
5. Дайте определение алгоритма по Колмогорову.

6. Каким требованиям должен отвечать алгоритм?
7. Что собой представляет машина Поста?
8. Что собой представляет машина Тьюринга?
9. Какие способы записи алгоритмов вам известны?
10. Каковы недостатки словесного способа записи алгоритма?
11. Каковы достоинства и недостатки представления алгоритма в виде блок-схемы?
12. Приведите пример простейшего алгоритма, представленного при помощи диаграммы Нусси—Шнейдермана.
13. Что такое «псевдокоды»?
14. На какие группы можно подразделить языки программирования высокого уровня?
15. Какие базовые алгоритмические конструкции вам известны?
16. Что такое «линейный алгоритм»?
17. Что такое «разветвляющийся алгоритм»? Чем обуславливается выбор пути при ветвлении?
18. Что такое «циклический алгоритм»? Чем отличается цикл с предусловием от цикла с постусловием?
19. Дайте определение данным. Чем данные отличаются от переменных и констант?
20. Какие базовые типы данных вам известны?
21. Что такое «массив»?
22. Что такое «запись»?
23. Что такое «связный список»? «Односвязный список»? «Двусвязный список»?
24. Опишите в словесной форме алгоритм удаления элемента односвязного списка.
25. Опишите в форме псевдокода алгоритм удаления элемента двусвязного списка.
26. Что такое «деревья»?
27. Что такое «корневое дерево»? Что такое «корень дерева»?
28. Что такое «бинарное дерево»?
29. Что такое «граф»?
30. Чем различаются ориентированный и не ориентированный графы?
31. В решении каких задач применяются графы?
32. Какие алгоритмы сортировки вам известны? Опишите особенности каждого из них.
33. Какие алгоритмы поиска вам известны?

Литература

1. *Голицина О. Л., Попов И. И.* Основы алгоритмизации и программирования. М.: Форум, 2008.
2. *Давыдов В. Г.* Программирование и основы алгоритмизации. М.: Высшая школа, 2003.
3. *Кузюрин Н. Н., Фомин С. А.* Эффективные алгоритмы и сложность вычислений. Электронное издание. <http://discopal.ispras.ru/ru/book-advanced-algorithms.htm>
4. *Мозговой М. В.* Классика программирования: алгоритмы, языки, автоматы, компиляторы. Практический подход. СПб.: Наука и техника, 2006.

Глава 15 Классификация и тенденции развития программного обеспечения

- 15.1. Классификация программного обеспечения по степени взаимодействия с аппаратной частью компьютера
- 15.2. Классификация программного обеспечения по виду лицензирования
- 15.3. Прочие классификации
- 15.4. Промежуточное программное обеспечение
- 15.5. Программное обеспечение процесса разработки программного обеспечения
- 15.6. Области применения прикладного программного обеспечения
- 15.7. Перспективы развития программного обеспечения

По подсчетам специалистов, ежедневно в мире появляется до 200 новых программных продуктов, и темпы развития программного обеспечения только увеличиваются. Наверное, можно ожидать, что в не столь отдаленном будущем в компьютерной науке появится новый отдел, этимология и систематика компьютерных программ, в котором будут рассматриваться семейства, виды и классы программного обеспечения и обсуждаться причины исчезновения и появления тех или иных видов программного обеспечения (ПО), какова перспектива у видов ныне существующих. Такая наука более тщательно опишет общие признаки и различия программных продуктов. Многие из важных классов программного обеспечения, описание которых здесь сделано несколько поверхностно, будут раскрыты глубже в следующих главах книги.

15.1. Классификация программного обеспечения по степени взаимодействия с аппаратной частью компьютера

Бурное развитие компьютерной техники во всем мире привело к столь же бурному развитию программных продуктов. Область программного обеспечения пережила эпоху «первичной свободы и хаоса», когда программ было крайне мало и они вообще не рассматривались как продукт, который можно было продавать, а поставлялись вместе с вычислительной техникой. Мало было и программистов, а практически рядом с каждым компьютером работало несколько человек, которые «изобретали» собственные программы.

По мере распространения компьютеров и все большей зависимости работы этой техники от качества программного обеспечения, установленного на ней, программы начали приобретать собственные «вещ» и стоимость. Чем больше появлялось компьютеров, тем больше становилось потенциальных покупателей программ. Рынок мгновенно отреагировал на это созданием коммерческого программного обеспечения. Область торговли компьютерами разделилась на две большие части: по продаже «железа» (hardware — аппаратное обеспечение) и «софта» (software — программное обеспечение).

Несомненным достоинством коммерческого подхода на данном этапе развития программного обеспечения было резкое повышение качества и унификация программного обеспечения. Недавний хаос превратился в порядок, быстро приобретя хорошо узнаваемые имена производителей программного обеспечения: Microsoft, Oracle, Sun, Borland. Программные продукты этих компаний пользуются большим спросом, а в отдельных областях полностью заняли рынок.

По мере все более глубокого проникновения компьютеров во все сферы жизни, в процессе все возрастающей мощи суперкомпьютеров и все уменьшающихся размеров микро-, супермикро- и встраиваемых ЭВМ, количество и разнообразие программных продуктов с каждым годом растет в геометрической прогрессии. Если совсем недавно программное обеспечение для первых компьютеров можно было разделять только по их маркам, то достаточно быстро возникло разделение на системное и пользовательское (прикладное) программное обеспечение. Со временем между этими двумя большими классами возник третий, промежуточный, слой, который так и называли — промежуточное программное обеспечение. Необходимость создания все новых программных продуктов привело к появлению еще одного класса программного обеспечения — систем программирования.

Так было заложено первое основание для классификации *по степени взаимодействия с аппаратурой (физическим оборудованием) компьютера*.

Если представить, что программное обеспечение слой за слоем накладывается на аппаратную часть компьютера, давая возможность пользователю взаимодействовать с «железом», то схематично эту структуру можно представить на рис. 15.1.

Прикладное программное обеспечение не взаимодействует напрямую с аппаратным обеспечением компьютера. В качестве примера прикладного программного обеспечения можно привести текстовый процессор, графический редактор,

электронную таблицу, проигрыватель музыкальных файлов, программу для расчета надежности несущих конструкций, программу бухгалтерского учета и прочие программы, которые выполняют пользовательские задания.

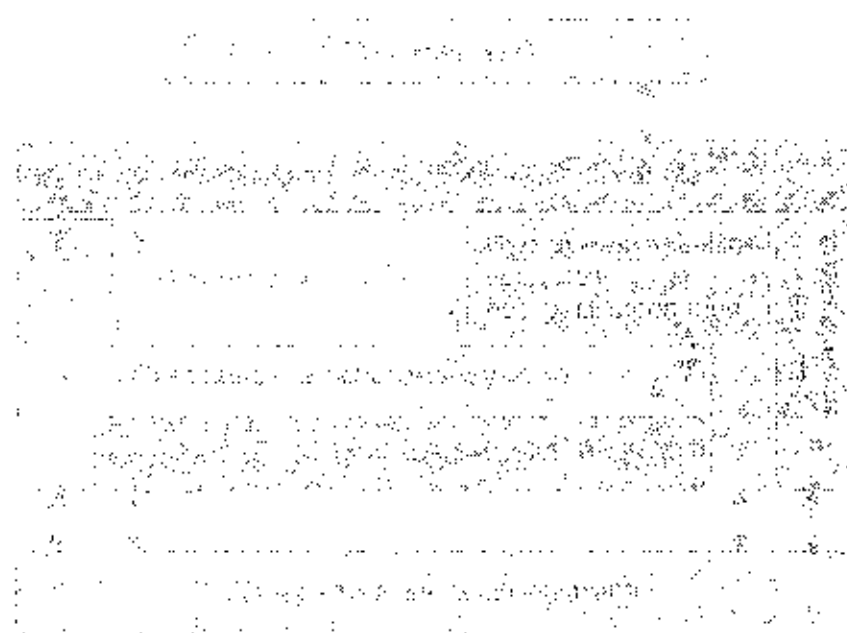


Рис. 15.1. Классификация программного обеспечения по степени взаимодействия с аппаратной частью компьютера

Прикладное программное обеспечение (application software) — это программное обеспечение, предназначенное для выполнения пользовательских заданий и имеющее дружелюбный пользовательский интерфейс.

Системное программное обеспечение составляют операционная система с набором инструментов (утилит) администрирования и настройки, а также базовая система ввода-вывода (Basic Input Output System, BIOS).

Системное программное обеспечение (system software) — это класс программ, обеспечивающих управление аппаратной частью компьютера и реализующих для пользовательских программ программный интерфейс с аппаратными средствами.

Промежуточное программное обеспечение — сравнительно недавно сформировавшийся класс программного обеспечения, располагающийся между системным и пользовательским программным обеспечением. Промежуточное программное обеспечение не взаимодействует с пользователем непосредственно, поэтому не может быть отнесено к разряду пользовательских программ. В промежуточное программное обеспечение встроены средства взаимодействия с клиентским программным обеспечением, которое, в свою очередь, является прикладным. К промежуточному программному обеспечению относят серверы баз данных, серверы приложений и другие серверные программы, обращение к которым производится через клиентскую часть.

Промежуточное программное обеспечение (middle ware) — это специальный класс программного обеспечения, взаимодействующий с аппаратной частью компьютера через программные интерфейсы системного программного обеспечения, но не имеющий интерфейса для взаимодействия с пользователями.

Программирование, так же как и производство средств разработки программного обеспечения, управления жизненным циклом программ и программной инженерии, сегодня составляет отдельную индустрию. Трудно провести точный водораздел между тем, где средства разработки программного обеспечения уже не принадлежат к системному программному обеспечению, а переходят на прикладной уровень. Поэтому следующий класс, класс инструментальных средств программирования, был выделен путем объединения программного обеспечения по сходному функциональному признаку.

Инструментальные средства программирования — это программы, предназначенные для создания программного обеспечения, в том числе системного, прикладного и промежуточного.

Теоретические вопросы виртуализации (то есть загрузки на одном компьютере нескольких операционных систем одновременно) рассматривались еще в 60–70-х гг., но серьезные и функционально законченные разработки в этом направлении активно ведутся только последние 10 лет. И даже широко известные виртуальные эмуляторы аппаратных платформ, такие как VMware, до какого-то момента можно было рассматривать только как прикладное программное обеспечение. Однако в последние несколько лет все основные производители компьютерного аппаратного обеспечения и центральных процессоров стали учитывать в конструкции своих продуктов возможность встроенной (built-in) виртуализации на «голом железе» (bare metal).

Средства виртуализации — это предоставляемое в составе аппаратной части компьютера или операционной системы программное обеспечение, позволяющее на одном компьютере организовать несколько виртуальных машин и на каждую установить собственную операционную систему.

15.2. Классификация программного обеспечения по виду лицензирования

На заре компьютерной эры все программы были открытыми и бесплатными, к компьютерам они прилагались, составляя крайне незначительную часть стоимости. По мере развития и массового внедрения в жизнь компьютерной техники наступила эпоха коммерциализации программного обеспечения. Исходные коды программ стали повсеместно закрываться патентами, лицензиями, соглашениями о неразглашении, и практически все программное обеспечение переводилось на коммерческие рельсы.

Символом успешности в области создания коммерческого программного обеспечения стал Билл Гейтс, владенец и в прошлом глава корпорации Microsoft. Наиболее известным и используемым программным продуктом этой всемирной корпорации является операционная система Microsoft Windows.

Монопольное положение на рынке сбыта программного обеспечения, удобное для его крупных производителей, во многих проявлениях не было воспринято обществом. Бывшее еще недавно бесплатным и свободным, программное обеспечение очень быстро было окружено частотолком патентов и лицензий, доступ к исходным кодам программ был закрыт, а изучение (декомпиляция) исполняемого кода стало незаконным. Первыми забии тревогу программисты: они не только потеряли возможность открыто обмениваться информацией, идеями и плодами своего труда, но само их существование вне рамок корпораций по производству программного обеспечения было поставлено под вопрос. Другим слоем общества, вкусившим горечь коммерциализации программного обеспечения, были учебные заведения и преподаватели, обнаружившие, что у них нет операционных систем и программных продуктов, на основе которых они могли бы преподавать на современном уровне основы операционных систем, инженерии программного обеспечения или программирование. Как результат, в обществе возникло противодействие тотальной коммерциализации программного обеспечения, что вновь оживило идею свободного и открытого программного обеспечения, но уже не на уровне первозданного хаоса, а на более высоком, упорядоченном уровне современных технологий программирования и сетевых коммуникаций.

Открытое программное обеспечение неразрывно связано с именем Ричарда Столлмана. Увидев в тотальной коммерциализации программного обеспечения не просто монополистическую тенденцию, но явную склонность к ущемлению базовых человеческих прав и свобод, Столлман предложил, используя имеющиеся правовые механизмы, производить программы, защищенные от коммерческой несвободы специальным видом лицензий (GNU GPL). В основу лицензии GPL, в отличие от термина «копирайт», которым защищаются права коммерческих производителей, он положил термин «копилефт», предложив этим термином и этими принципами защищать права программистов и пользователей программ. Ричард Столлман всегда подчеркивал, что его движение направлено на защиту прав программистов и пользователей, а не на производство дешевого или бесплатного программного обеспечения. Столлман показал, что у развития программного обеспечения есть альтернативный путь, который привел к созданию свободной и открытой операционной системы GNU/Linux — основного и сильного конкурента Microsoft Windows.

Свободное программное обеспечение (СПО) — это программное обеспечение, распространяемое под лицензией GPL или любой другой лицензией, реализующей четыре базовые свободы программного обеспечения и принцип «копилефт» (copyleft).

Под четырьмя базовыми свободами программного обеспечения понимаются:

- **свобода использования** — пользователь может запускать программу и применять результаты ее работы для любых своих целей;

- *свобода изучения и модификации* — пользователь может сколько ему угодно изучать, как устроена программа, а также модифицировать ее под свои нужды или для продажи;
- *свобода распространения* — программный продукт может быть свободно распространен (передан, подарен, продан или установлен), а любые ограничения на распространение программного продукта запрещены;
- *свобода распространения модификаций* — модифицированные версии продукта должны распространяться так же свободно, как и исходная версия.

Принцип «копилефта» выглядит следующим образом: при модификации и распространении исходного продукта его лицензия должна включать в себя перечисленные свободы и данное условие «копилефта».

Обращаем ваше внимание на тот факт, что нигде в лицензиях семейства GPL не указываются никакие ограничения, связанные с продажей программ. Свободное — не обязательно бесплатное. Более того, если программа соблюдает все условия лицензии GPL, но в текст лицензии добавлена одна-единственная строчка «свободна для некоммерческого использования» или, например, «свободна для использования в образовательных и академических целях», то такая программа считается несвободной или полусвободной.

Принципиальная позиция Столлмана и возглавляемого им фонда СНО состоит не в том, что программы должны быть бесплатными. Бесплатность или низкая стоимость свободных программ и возможность в связи с этим широко их применять в сферах и регионах, где есть затруднения с финансированием, — это уже вторичный эффект. Слово «свобода» является определяющим. На это направлена лицензия GPL и включенный в нее текст «копилефта», из-за которого эти лицензии называют «вирусными»: лицензия требует, чтобы все, во что включается свободный продукт, также было выпущено под свободной лицензией. Таким образом, лицензии GPL являются «разносчиками свободы» в мире программного обеспечения.

Несвободное программное обеспечение — это все программное обеспечение, которое реализуется под лицензиями, ограничивающими 4 базовые свободы, или не обеспечивающими принцип «копилефт».

К несвободному программному обеспечению относится полусвободное программное обеспечение (имеющее ограничение на свободу распространения в коммерческих целях) и другие классы несвободного программного обеспечения.

- *Проприетарное программное обеспечение* — программное обеспечение, выпущенное и распространяемое на условиях коммерческих лицензий и под защитой авторских прав «copyright».
- *Пробные версии проприетарного программного обеспечения (trial)* — проприетарное обеспечение, распространяемое бесплатно на ограниченный срок (или с ограниченной функциональностью) для того, чтобы потенциальный покупатель смог испытать программное обеспечение в деле. Обычно в полнофункциональные пробные версии встраивается механизм, который после истечения

пробного срока (или установленного количества запусков программы) запрещает ее дальнейшее использование.

- ❑ *Условно-бесплатное программное обеспечение* (shareware) очень схоже по способу распространения с пробными версиями проприетарного программного обеспечения. Распространяется бесплатно, но с ограничениями по функциональности, времени использования или количеству запусков. Познакомившись таким образом с программой, пользователь может выслать автору денежное вознаграждение и получить полнофункциональную версию.
- ❑ *Бесплатное программное обеспечение с рекламной добавкой* (ad-ware) — полностью бесплатная для пользователя полнофункциональная версия программы, в которую включены элементы рекламы (баннеры, надписи, всплывающие окна, надписи в строке состояния или заголовке окна). Оплату программисту в этом случае производит рекламодатель.
- ❑ *Бесплатное программное обеспечение* (freeware) — полностью бесплатная версия программного обеспечения, распространяемая без исходных кодов.
- ❑ *Программное обеспечение с открытым исходным кодом* не относится к свободному программному обеспечению, если выпущено не под лицензией GPL или родственной.

15.3. Прочие классификации

15.3.1. Классификация программного обеспечения по платформе назначения

Под словом «платформа» в данном случае понимается операционная система.

- ❑ *Одноплатформенное программное обеспечение* разрабатывается и выполняется только на одной операционной платформе (например, программы, написанные только для использования в семействе операционных систем Microsoft Windows или только для Linux).
- ❑ *Переносимое (portable) программное обеспечение* — программное обеспечение, разработанное в технологиях программирования, подразумевающих возможность создания вариантов программы под разные платформы. Создание переносимого программного обеспечения подразумевает наличие одних и тех же правил, например, одних и тех же библиотек, одинакового интерфейса системных вызовов на разных платформах. В качестве примера переносимых приложений можно назвать приложения на языках Perl и Python, переносимые между UNIX, Linux и Windows, или приложения на языке C++.
- ❑ *Межплатформенное программное обеспечение* — программное обеспечение, одинаково функционирующее в любой операционной среде. Полностью межплатформенных приложений всего два вида: Java- и веб-приложения. Те и другие требуют для своей работы установки на платформу, на которой они будут запускаться, специализированного программного обеспечения, написанного специально для данной платформы. В случае Java-приложений — это среда

выполнения Java (Java Runtime Environment, JRE), в случае веб-приложений — веб-браузер.

Еще одним типом межплатформенных программных приложений можно назвать технологию .NET от Microsoft. Потенциально технология .NET разрабатывалась на тех же принципах, что и Java, но на сегодняшний день она не получила широкого распространения нигде, кроме самой платформы Microsoft.

15.3.2. Классификация программного обеспечения по способу взаимодействия с пользователем

По типу взаимодействия с пользователем программное обеспечение можно разделить на следующие классы:

- ❑ *Программы с графическим оконным интерфейсом* — к этому типу относится подавляющее большинство прикладных программ, запускаемых на современных операционных системах. Поскольку практически все операционные системы оснащены графической оконной средой с рабочим столом пользователя, то и программы в них запускаются в оконном интерфейсе.
- ❑ *Консольные программы* — программы, запускаемые в окне командной строки. Программы такого рода могут также подразделяться на программы с графическим пользовательским интерфейсом (например, игры), с псевдографическим пользовательским интерфейсом (FAR в Windows или Midnight Commander в Linux) и программы командной строки.
- ❑ *Программы системного лотка* отображаются исключительно в области уведомлений панели задач, управляются из контекстного меню и не имеют оконного интерфейса (рис. 15.2).

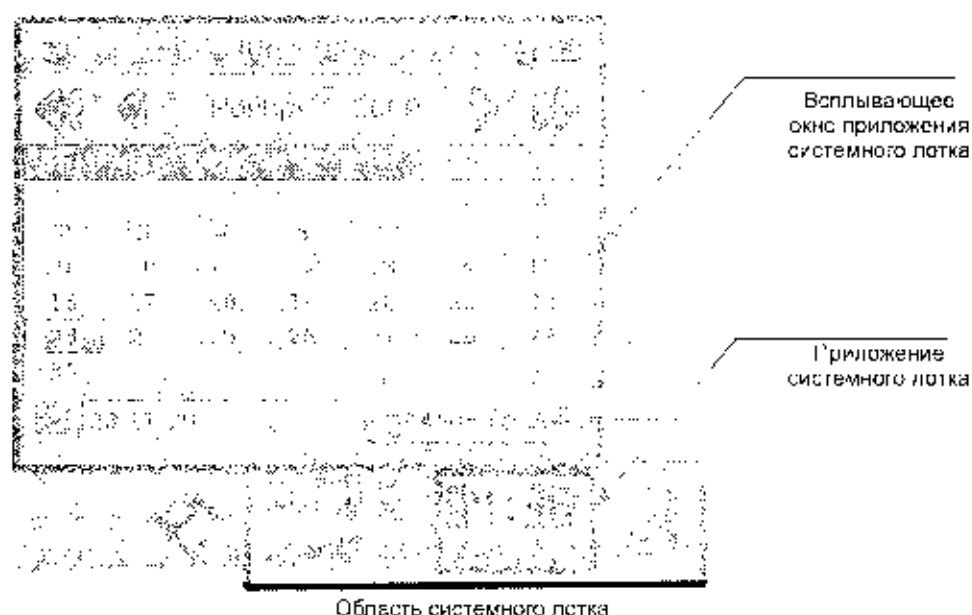


Рис. 15.2. Системный лоток

Системный лоток (system tray), или область уведомлений панели задач, — это область значков на панели задач, обычно располагающаяся в правом нижнем углу экрана.

- *Виджеты (гаджеты) рабочего стола* — программы, которые выглядят как элементы рабочего стола, но при этом являются полнофункциональными пользовательскими приложениями и не имеют отношения к операционной системе (например, виджет-интегратор информации с новостных лент).

15.3.3. Классификация программного обеспечения по отношению к сети

Одним из важных оснований для классификации является отношение программного обеспечения к сети. По этому признаку программы можно разделить следующим образом:

- предназначенные для создания и администрирования сетей — разного рода сетевые утилиты, мастера настройки и создания сетей, брандмауэры, прокси-серверы, программы контроля сетевого трафика;
- предназначенные для работы с сетью — электронная почта, веб-браузер, клиентское приложение службы мгновенных сообщений (ICQ);
- предполагающие наличие сети — офисные программы с возможностью отправки документов по электронной почте, программы со встроенным ftp-клиентом; ..
- обновляемые (устанавливаемые) по сети;
- не связанные с сетью.

15.3.4. Классификация программного обеспечения по способу установки

По способу установки программное обеспечение можно разделить на следующие классы:

- требующие установки;
- не требующие установки;
- переносимые (portable application), то есть запускаемые с любых сменных носителей.

ВНИМАНИЕ

В этой классификации имеется совпадение терминологии. Одним и тем же термином («переносимые») называют программы, запускаемые со сменных носителей информации и переносимые с платформы на платформу.

15.3.5. Классификация программного обеспечения по целям разработки

По целям разработки программное обеспечение можно разделить на следующие классы:

- *для массовых продаж* (распространения) — операционные системы, системы управления базами данных, банковские системы;

- ❑ *заказное* – программное обеспечение, которое разрабатывается под конкретного заказчика по его техническому заданию, например, информационная система или система управления технологическим процессом;
- ❑ *целевое* – программное обеспечение, создаваемое для выполнения конкретных задач самими пользователями этого программного обеспечения, например, программное обеспечение для уникального научно-исследовательского оборудования, созданное работниками лаборатории;
- ❑ *индивидуальное* – программное обеспечение, создаваемое для собственного использования

15.3.6. Классификация программного обеспечения по аппаратной платформе

По аппаратной платформе программное обеспечение можно разделить на классы для следующих компьютеров:

- ❑ суперЭВМ и систем параллельных вычислений;
- ❑ серверов и мэйнфреймов;
- ❑ десктопов и рабочих станций;
- ❑ ноутбучных компьютеров и коммуникаторов;
- ❑ встраиваемых микропроцессорных устройств.

15.3.7. Классификация программного обеспечения по степени опасности

Отдельным классом можно выделить *опасное программное обеспечение*. К опасному программному обеспечению можно отнести компьютерные вирусы, программы-шпионы, программы, осуществляющие насильственное внедрение рекламы, и прочее программное обеспечение, создаваемое с деструктивными целями, отрицательно влияющее на работоспособность операционной среды, доставляющее неудобство пользователю, угрожающее потерей или похищением конфиденциальной информации.

15.4. Промежуточное программное обеспечение

Как уже отмечалось, промежуточное программное обеспечение особенно активно развивалось в последнее десятилетие. Это связано с развитием информационных систем предприятий и сетевых технологий.

Как видно из рис. 15.3, такое программное обеспечение, как сервер баз данных в схеме, не является системным программным обеспечением (не входит в состав операционной системы, не является ее утилитой). С точки зрения операционной системы промежуточное программное обеспечение выглядит как прикладное при-

ложение. С точки зрения пользователя прикладным приложением является та программа, с которой он взаимодействует. Ни клиентскую часть сервера баз данных, ни самого сервера пользователь не видит, с ними напрямую не взаимодействует и иногда даже не подозревает об их существовании. Таким образом, промежуточное программное обеспечение по одному ряду признаков можно отнести к системному программному обеспечению, а по другому — к прикладным приложениям. Именно поэтому такое программное обеспечение выделили в отдельный класс.

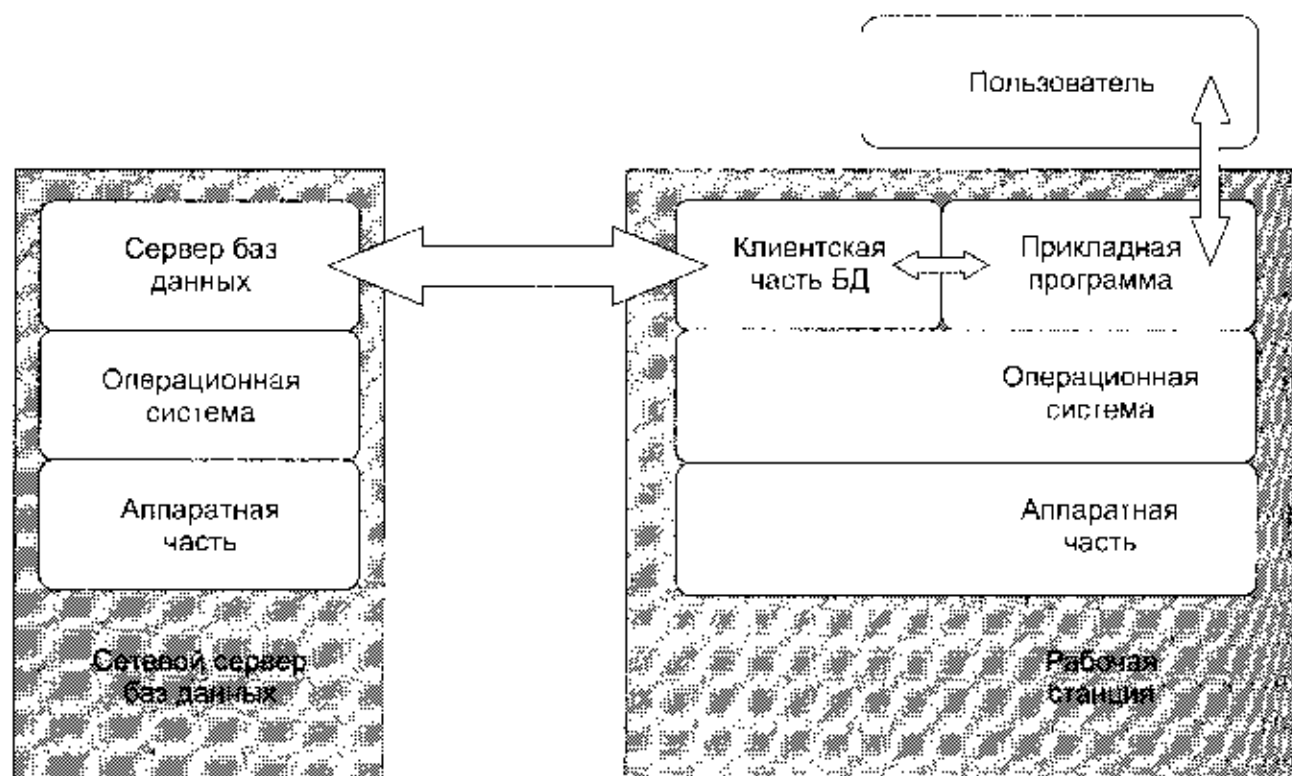


Рис. 15.3. Промежуточное программное обеспечение

К промежуточному программному обеспечению относятся:

- ☐ серверы баз данных и их клиентские части;
- ☐ серверы приложений;
- ☐ веб-серверы.

Все или некоторые из этих программ являются частью любой корпоративной информационной системы.

15.5. Программное обеспечение процесса разработки программного обеспечения

Разработка программного обеспечения сегодня стала важнейшей отраслью мирового промышленного производства. В разработку программного обеспечения вкладываются десятки миллионов долларов, прибыль исчисляется миллиардами. Не удивительно, что и разработка средств разработки тоже стала самостоятельной

отраслью. К программному обеспечению, связанному с разработкой программ, обычно принято относить компиляторы, интерпретаторы, отладочные и сервисные (вспомогательные) утилиты. Процесс разработки программного обеспечения можно кратко описать следующим образом:

- сбор, формулирование, уточнение и соблюдение требований к программному продукту;
- работа по созданию архитектуры программного продукта;
- написание и отладка исходного кода программ;
- тестирование программного продукта;
- поставка и внедрение программного продукта;
- сопровождение программного продукта.

Подход к разработке программного обеспечения как к промышленному процессу заставил по-новому взглянуть на создание инструментов разработки. Фирмы-производители средств разработки стали выпускать комплексные решения, включающие в себя программную поддержку каждого из перечисленных этапов.

15.6. Области применения прикладного программного обеспечения

Наиболее «чистым» по составу, функциональности, количеству названий и наиболее близким к конечному пользователю является, безусловно, класс прикладных программ. Самой очевидной для прикладных программ является их систематизация по функциональному назначению и области применения. С точки зрения функциональности прикладное программное обеспечение можно разделить на несколько больших групп:

- офисные приложения;
- приложения для управления проектами;
- приложения для работы с локальной сетью;
- приложения для Интернета;
- программы для научных исследований и расчетов;
- образовательные программы;
- программы для организации работы учебных заведений;
- программы для библиотек;
- программы для работы с мультимедиа;
- бухгалтерские программы;
- финансовые программы;
- программы для проектирования;
- программное обеспечение для бизнеса;
- программное обеспечение органов государственной власти;

- программы поддержания безопасности;
- программы для личного планирования;
- игры.

Здесь перечислены только основные направления, в которых функционально развивается пользовательское прикладное программное обеспечение. Охватить абсолютно все невозможно по той простой причине, что сегодня практически любая деятельность человека, любая область его жизни поддерживается тем или иным видом программного обеспечения. Рассмотрим более подробно основные категории.

15.6.1. Офисные приложения

К офисным приложениям можно отнести как готовые офисные пакеты (проприетарный Microsoft Office или открытый OpenOffice.org), так и отдельные программы, связанные с выполнением функций ввода, хранения, обработки и представления документов в электронном виде: различные текстовые редакторы и текстовые процессоры, электронные таблицы, программы для создания презентаций, графиков и диаграмм, программы индивидуального и группового планирования. Офисные приложения настолько глубоко пропикли в любую деятельность, что сегодня настольный компьютер немислим без офисного пакета, который воспринимается как неотъемлемая часть компьютера.

У каждого офисного приложения, из входящих в состав офисных пакетов, свое назначение и свой набор необходимых и дополнительных функций.

Текстовый процессор

Текстовый процессор — это приложение, основным назначением которого является создание и редактирование текстовых документов. Необходимыми для современного текстового процессора являются функции *ввода текста* и *выполнения над текстом операций редактирования* (копирование, вырезание, удаление и вставка в указанное место фрагментов текста), а также *сохранения текста* в файле на физическом носителе.

Дополнительные функции, поддерживаемые современными текстовыми процессорами, давно уже стали де-факто стандартом создания программного обеспечения такого класса:

- *форматирование текста* — изменение типа и параметров шрифта (цвета символов и фона, размера, зачеркивания, подчеркивания, расстояния между символами и прочих параметров);
- *форматирование абзацев* — изменение параметров выравнивания, нумерация, создание списков;
- *форматирование страниц* — разбиение на страницы, автоматическое и произвольное, изменение количества колонок, создание разделов;
- *поиск и замена* фрагментов в тексте документа;
- *печать* документа;

- *пересылка* документа адресату при помощи электронной почты;
- *средства совместной работы* над документами (рецензирование);
- *вставка* в документ изображений, графиков и диаграмм;
- *автоматизация обработки документа* — средства вставки оглавления, список, цитат, списка литературы, средства формирования структуры документа;
- *экспорт документа* в различные форматы — в особенности важен экспорт в межплатформенный формат HTML.
- *средства программирования функций* на встроенном языке программирования.

На рис. 15.4 показаны окна двух текстовых процессоров. Первый (Word) входит в стандартный программный продукт Microsoft Office, второй (Writer) — в свободный программный продукт OpenOffice.org. Видно, что две основные панели инструментов этих текстовых процессоров совпадают функционально почти полностью.

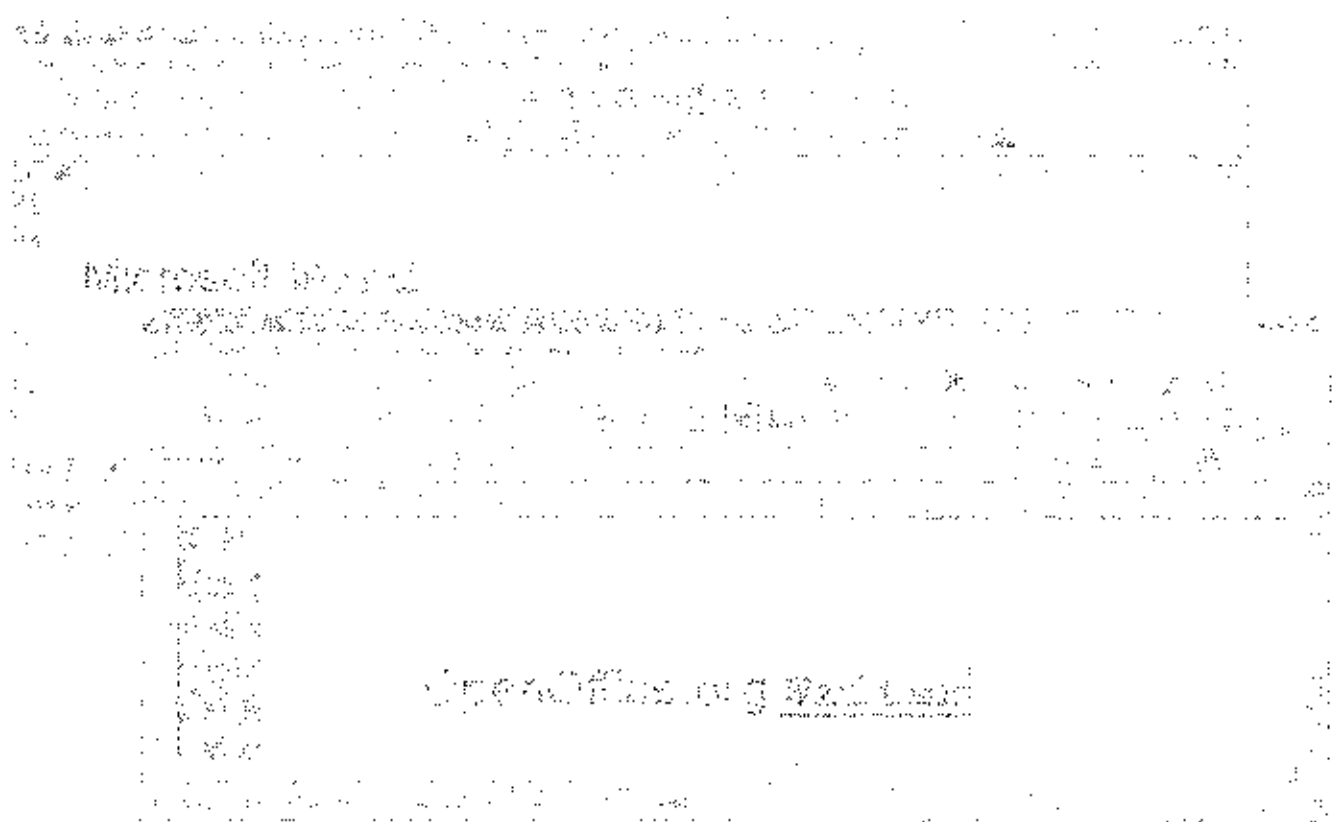


Рис. 15.4. Текстовые процессоры Word (Microsoft) и Writer (OpenOffice.org)

Электронные таблицы

Если текстовый процессор выполняет функции представления данных в виде текстового документа, то электронная таблица в первую очередь упорядочивает, сортирует, хранит и обрабатывает данные, представляя их в табличном формате.

Основная функция электронной таблицы — хранить данные в типизированных ячейках, позволяющих обратиться к единице данных путем адресации по имени (номеру) столбца (строки), а также обрабатывать данные, выполняя над ними арифметические операции или передавая их в виде аргументов встро-
енных функций.

Дополнительные функции электронной таблицы почти аналогичны функциям текстового процессора: форматирование текста, вставка изображений и различных объектов, стилевое и цветовое форматирование текста, фона и сетки таблицы, причем как безусловное (форматирование выделенного фрагмента), так и условное (в зависимости от значений в тех или иных ячейках). Кроме того, к дополнительным функциям электронных таблиц можно отнести расширение их функциональности за счет специализированных надстроек, предназначенных для выполнения статистических, финансовых, экономических и научных расчетов и экспериментов с данными. Так же как и для текстовых документов, для электронных таблиц важна способность экспорта в разные форматы, в особенности в формат HTML, и вывода таблиц на печать.

На рис. 15.5 показаны электронные таблицы Excel из пакета Microsoft Office и Calc из продукта OpenOffice.org.

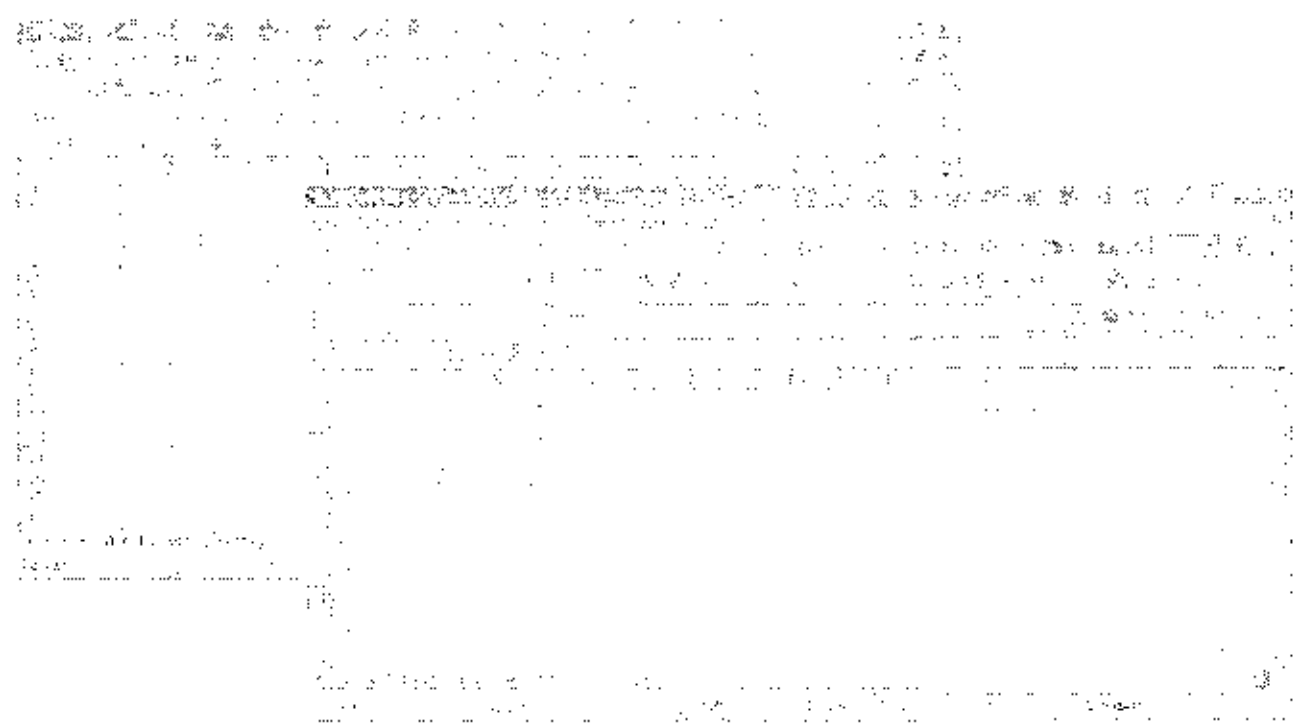


Рис. 15.5. Электронные таблицы

Свободная программа Calc и проприетарная Excel почти не различаются по функциональности. Все, что можно сделать в Microsoft Excel, можно сделать и в Calc. Документы, созданные в Calc, можно сохранить в формате Excel, а документы, созданные в Excel, могут быть открыты в Calc. Однако о полной идентичности, как и о полной совместимости говорить нельзя: некоторые операции (например, корректное восстановление ссылок на другие листы и книги), которые поддерживает Microsoft Excel, в Calc не поддерживаются. Есть несовместимость и на уровне прикладного программирования: встроенные языки в этих двух программах разные, поэтому макросы Microsoft Excel в Calc работать не будут.

Для того чтобы далее не повторяться, нужно отметить, что такого рода неполная совместимость как по функциональности, так и по встроенным механизмам автоматизации свойственна всем программам двух пакетов: Microsoft Office и OpenOffice.org.

Программы презентаций

Презентации стали неотъемлемой частью офисного пакета не сразу. Первые офисные пакеты включали в себя только текстовые процессоры и электронные таблицы, в некоторых случаях в офисные пакеты включался инструмент для создания подписок документов и некоторые другие. Однако по мере развития мультимедийных средств и проекторного оборудования все явственней становилась необходимость сжатого, визуального, красиво оформленного, иллюстрированного диаграммами и графиками представления информации. Так возник жанр компьютерной презентации, а вместе с ним и программы для создания презентаций.

Основной функциональностью программы создания презентаций следует считать возможность создания, оформления и воспроизведения в различных режимах компьютерных презентаций.

К дополнительной функциональности можно отнести следующие возможности:

- наличие большого количества и разнообразие визуальных и звуковых эффектов, воспроизводимых при переходе между слайдами и от одной части слайда к другой;
- создание автономных (standalone) презентаций, то есть презентаций, которые воспроизводятся независимо от базовой программы (это может быть исполняемый файл, а также формат flash- или pdf-файл);
- развитая система шаблонов и богатая коллекция изображений;
- взаимодействие с презентационным оборудованием;
- возможность внедрения сложных мультимедийных объектов и простое управление ими.

На рис. 5.6 показаны программы для создания презентаций Power Point из пакета Microsoft Office и Impress из продукта OpenOffice.org.

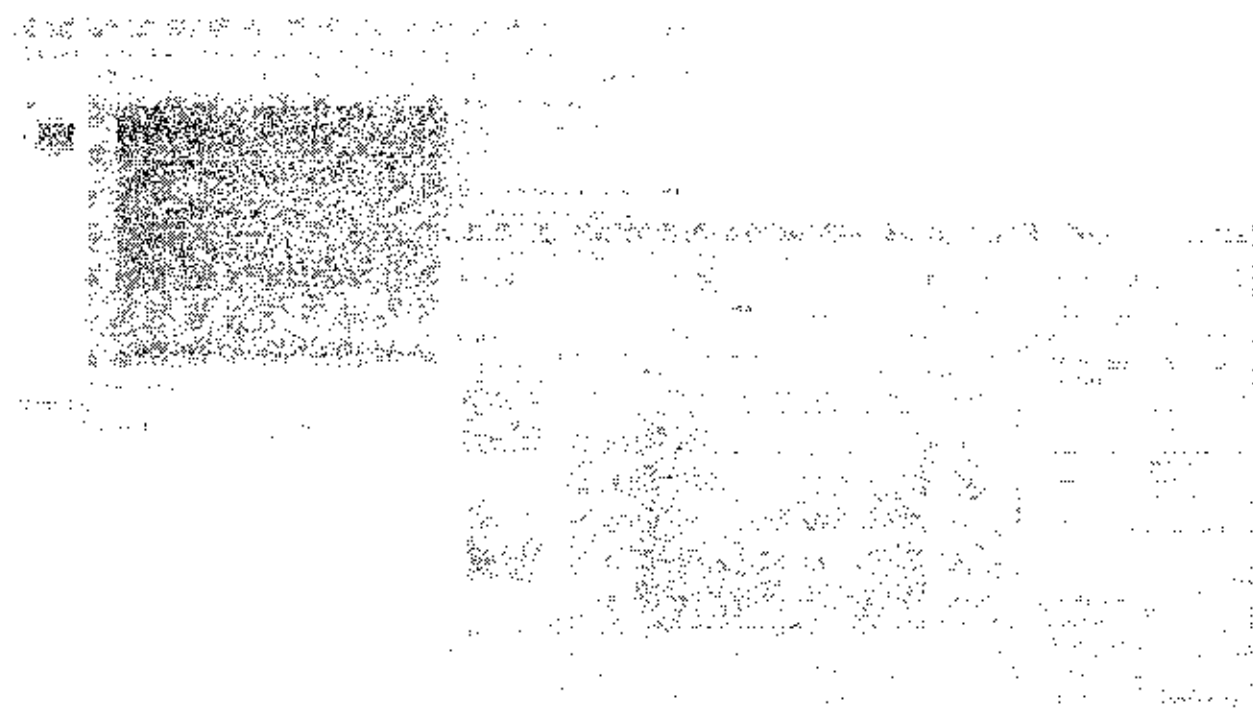


Рис. 15.6. Создание презентаций

15.6.2. Программы управления проектами

Одна из популярных сегодня сфер деятельности менеджеров на самых разных уровнях управления в бизнесе — управление проектами. Проектный способ управления, в котором комплекс взаимосвязанных задач бизнеса рассматривается как единый проект с точно определенными во времени началом и концом, бюджетом, штатом исполнителей, с полным распределением задач, оказался эффективным во многих отношениях: он хорошо алгоритмизируется, стандартизируется, оказывается легко переносимым из одной сферы в другую.

Не удивительно, что средства управления проектами для менеджеров как высшего, так и среднего звена являются довольно распространенным классом программного обеспечения. Самый известный программный продукт для управления проектом — безусловно, Microsoft Project в исполнении для настольного компьютера и в серверном исполнении. Этот продукт позволяет управлять как отдельными проектами малого и среднего масштаба, так и объединенными в портфель пакетами проектов корпоративного уровня.

Необходимыми для управления проектом являются следующие возможности:

- определять (задавать) ресурсы, включая материалыые, финансовые, человеческие, временные и пр.;
- определять работы (задачи), устанавливая их иерархию и взаимосвязанность;
- разрабатывать и отслеживать бюджеты проекта по разным срезам (времени, ресурсам, работам);
- эффективно распределять ресурсы и работы, отслеживать и отмечать выполнение задач и расходование ресурсов;
- получать отчеты о ходе выполнения проекта в разных формах (диаграммы Ганта, календарный план-график, бюджеты, еженедельные или ежедневные представления);
- гибко перестраивать созданные проектные конфигурации.

15.6.3. Клиентские программы для работы с сервисами Интернета

По протоколу HTTP работает самая известная служба Интернета, составляющая «Всемирную паутину» (World Wide Web, WWW). Эту службу используют программы, называемые интернет-браузерами, или программами просмотра Интернета. В задачу интернет-браузера входит загрузка интернет-страниц с заданного адреса, правильное их отображение, обеспечение взаимодействия пользователя с активными элементами интернет-страницы, поддержание необходимого уровня безопасности и защиты конфиденциальной информации пользователя. Наиболее популярными программами этого класса на сегодняшний день являются Microsoft Internet Explorer и свободный программный продукт Mozilla FireFox, быстро растет популярность еще одной программы-браузера — Google-Chrome. Окна этих трех браузеров представлены на рис. 15.7.

Из рисунка видно, что особенных новшеств ни один браузер, по крайней мере внешне, не внес. Надо отметить, что модель открытой разработки, в которой созда-

ется Mozilla FireFox, имеет свои преимущества: за время существования этой программы добровольцами для нее были разработаны десятки тысяч дополнительных модулей. Эти модули значительно расширяют функциональность браузера Mozilla FireFox. Некоторые модули позволяют полностью изменить сам способ представления информации внутри окна программы (рис. 15.8).

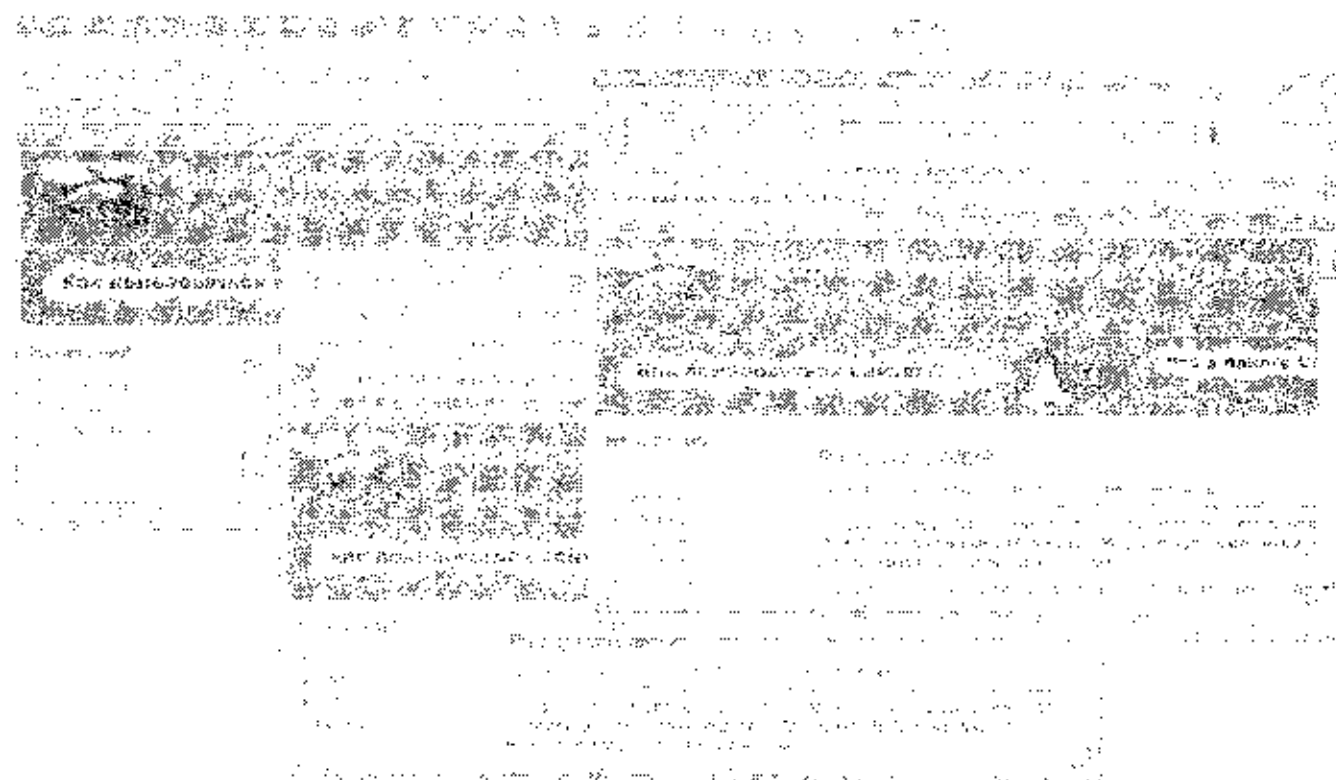


Рис. 15.7. Браузеры Интернета

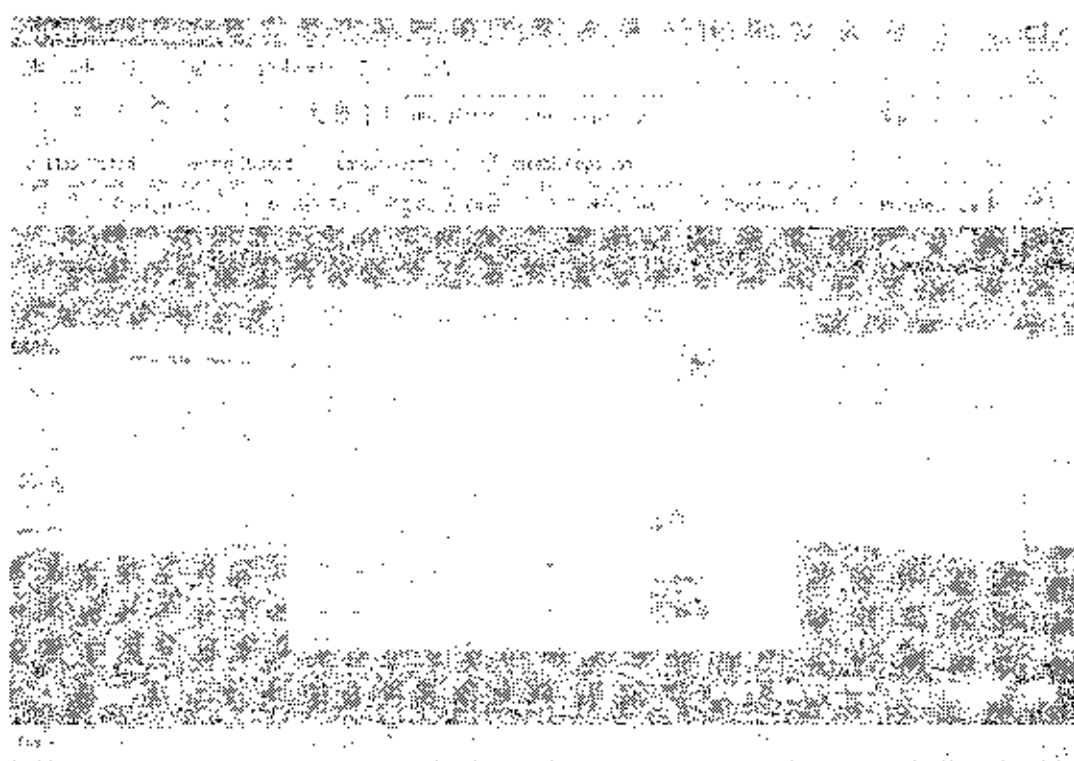


Рис. 15.8. Mozilla FireFox с подключенным модулем нестандартного представления вкладок

FTP

Сетевой протокол FTP предназначен для получения файлов с ftp-серверов, при этом ftp-серверы исполняют роль своеобразного склада файлов. Специальных клиентских приложений, которые работают с этим протоколом, на сегодняшний день практически нет, поскольку все браузеры Интернета способны читать ftp-каталоги и загружать из них файлы на компьютер пользователя. На рис. 15.9 можно увидеть, как выглядит один и тот же ftp-каталог в менеджере файлов Konqueror и в Internet Explorer.

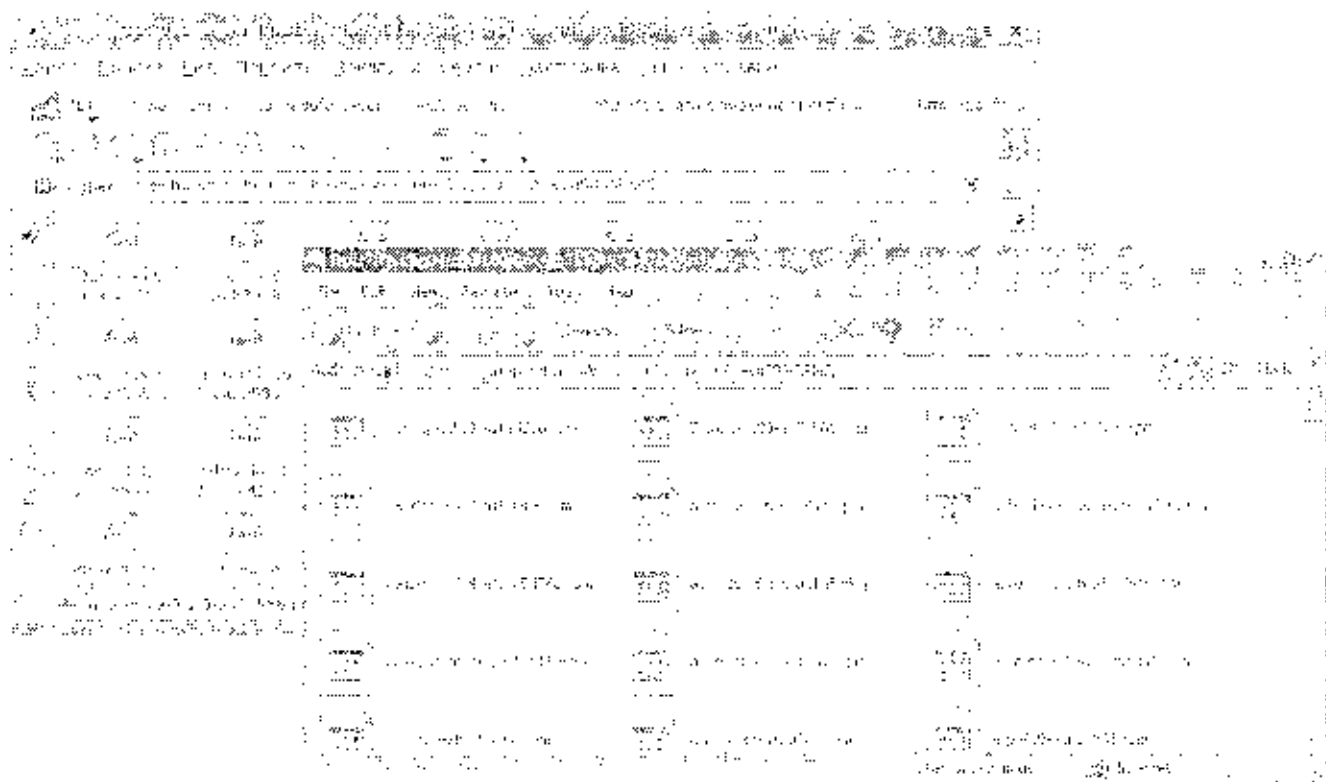


Рис. 15.9. Работа с ftp-каталогом

Из рисунка видно, что современные средства работы с ftp-серверами воспроизводят удаленные сетевые папки так же, как локальные каталоги на диске, и если у пользователя есть соответствующие права, то разница между сетевыми и локальными файлами практически стирается: можно открывать, редактировать, вырезать, копировать и перетаскивать файлы и папки как с жесткого диска на удаленный сервер, так и обратно.

Электронная почта

Электронная почта является одним из наиболее распространенных в Интернете средств обмена личной и служебной информацией. Программного обеспечения для работы с электронной почтой чрезвычайно много. Из наиболее известных почтовых клиентов с графическим пользовательским интерфейсом стоит отметить, видимо, коммерческие программы Microsoft Outlook и The Bat, а также свободную программу Mozilla Thunderbird. На рис. 15.10 можно видеть окна Microsoft Outlook и Mozilla Thunderbird.

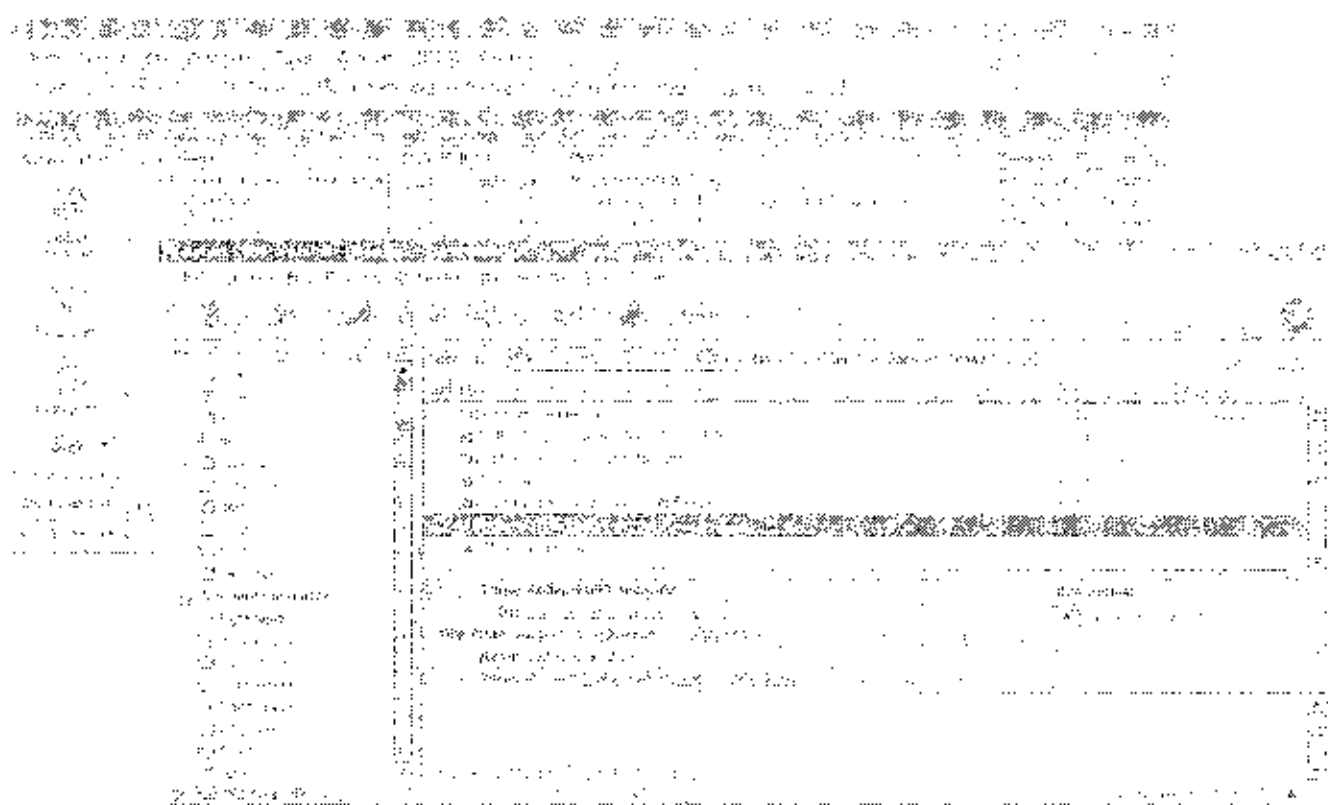


Рис. 15.10. Почтовые клиенты для приема и отправки электронных сообщений

Для современной почтовой программы одного умения принимать и отправлять сообщения недостаточно, чтобы выдерживать конкуренцию на рынке подобных программ. Дополнительно требуется поддерживать следующие возможности:

- прием и отправка сообщений не только в текстовом формате, но и в других форматах (например, HTML);
- отправка вложений;
- воспроизведение мультимедийного содержания;
- поиск внутри заголовков, тем и текста сообщений;
- ведение базы данных адресов;
- создание дополнительных папок;
- выполнение автоматических операций над приходящей почтой, в том числе раскладывание ее по различным папкам в зависимости от назначенных фильтров;
- защита от опасного содержимого, находящегося в сообщении или вложении.

Службы мгновенных сообщений

Службы мгновенных сообщений (интернет-пейджеры) с момента своего появления и по сегодняшний день беспрецедентно популярны среди пользователей Интернета. Первой и наиболее известной считается служба ICQ. Немного менее известен формат и приложения, поддерживающие протокол Jabber. Номер мобильного телефона и номер ICQ стали такими же неотъемлемыми идентификаторами личности, как номер паспорта. Службы мгновенных сообщений позволяют обмениваться сообщениями в одном окне, параллельно пересылать друг другу файлы

(например, фотографии). Из дополнительных функций, которые реализуют службы мгновенных сообщений, можно выделить такие как организация конференций и коллективного чата (одновременной беседы нескольких человек, отображаемой в одном окне).

Программ, поддерживающих мгновенные сообщения, на сегодняшний день довольно много, и все они либо свободны, либо бесплатны, но с рекламой. На рис. 15.11 показаны окна программ Корсет и QIP, поддерживающих одновременно форматы и Jabber, и ICQ.

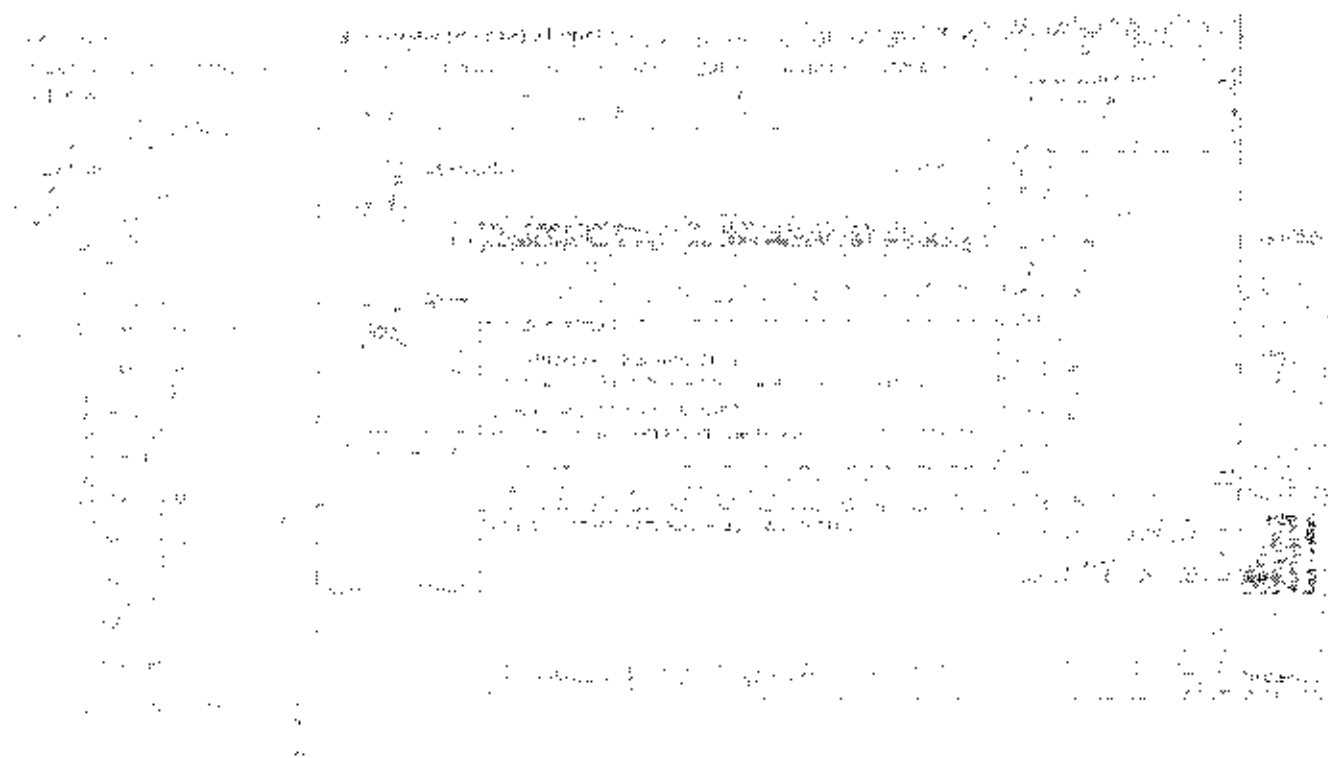


Рис. 15.11. Интернет-пейджеры

Одноранговые сети

Одноранговые, или пиринговые, сети (Peer to Peer, p2p) не нуждаются в сервере. Назначение этого сервиса — прямой обмен файлами между участниками сети. Работа пиринговых сетей основана на том, что каждый клиент одновременно является и сервером. Если у кого-то есть интересный файл, он указывает программе, что хочет выставить этот файл в сеть, а затем оповещает об этом заинтересованных людей. Программа разбивает файл на мелкие части, и другие люди, загружающие этот файл, одновременно предоставляют те «кусочки» которые они уже загрузили, следующим клиентам. Таким образом достигается эффект оптимизации нагрузки на сеть и отсутствие единого сервера в одноранговой сети. Среди наиболее известных программ организации p2p-сетей можно назвать *µTorrent* — клиентскую программу, окно которой можно увидеть на рис. 15.12.

Основной проблемой пиринговых сетей считается тот факт, что по ним часто с нарушением авторских прав распространяются электронные информационные продукты (программы, фильмы, книги и музыка).

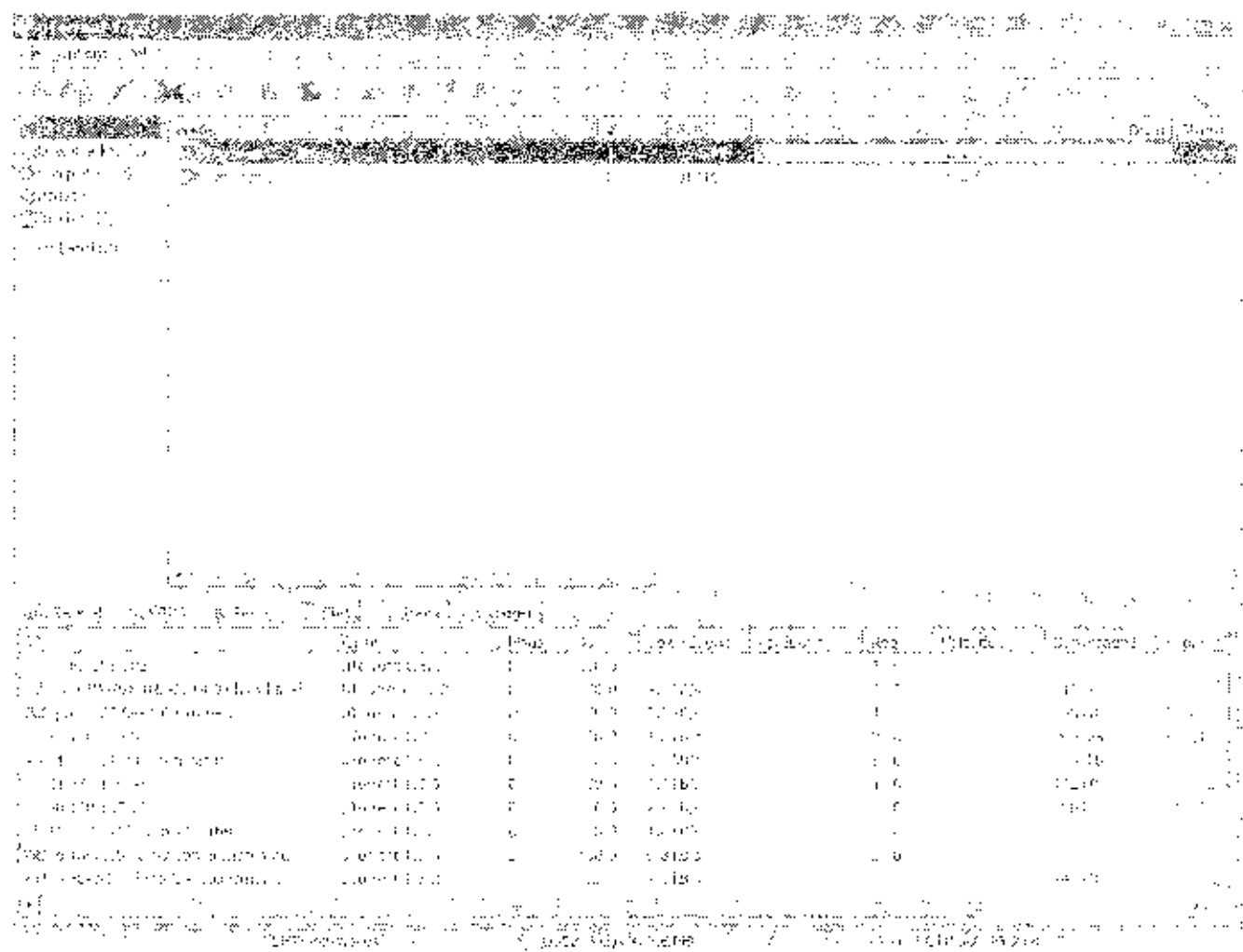


Рис. 15.12. Программа µTorrent

15.6.4. Программы для научных исследований и расчетов

Специализированное программное обеспечение для проведения научных исследований, сбора научной (экспериментальной) статистики и осуществления на основе собранных данных специальных научных расчетов не так широко известно, как, например, программное обеспечение для работы в Интернете, офисное или мультимедийное программное обеспечение. Одним из направлений научных исследований, на котором самым широким образом применяется специализированное научное программное обеспечение, является биоинформатика, тесно связанная с расшифровкой генома человека и последующими построениями генных моделей (геномной инженерией) для решения задач медицины, здоровья, сельского хозяйства. Программа Avogadro, например, позволяет создавать удивительные трехмерные модели молекул (рис. 15.13).

Программы для общих математических, статистических и физических расчетов встречаются наиболее часто (примеры таких программ: STATISTICA, MathCad, MathLab, MATHEMATICA). Третий по количеству программ тип — программы для астрономического моделирования и астрономических расчетов.

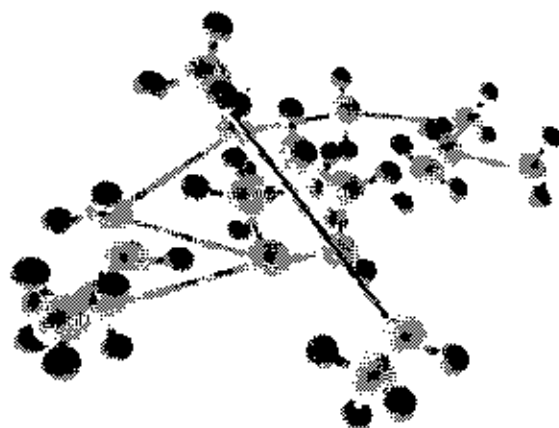


Рис. 15.13. Трехмерная модель молекулы, созданная в программе Avogadro

15.6.5. Образовательные программы

Тенденция интеграции компьютерных технологий в учебный процесс сегодня проявляется все сильнее и сильнее, одновременно активно идет разработка программного обеспечения, специально ориентированного на учебный процесс. Программное обеспечение для учебного процесса можно разделить на три основные группы:

- для взаимодействия;
- для передачи знаний по определенным предметам;
- для компьютерного тестирования и тренинга.

Программы для взаимодействия

При помощи современных программных и технических средств преподаватель может демонстрировать на мониторах учащихся свой рабочий стол и видеть рабочие столы учащихся на мониторе своего компьютера. Эти же средства позволяют учащимся получать доступ к рабочим столам друг друга. Обычно данный механизм эффективно работает внутри одного класса в рамках локальной компьютерной сети, но при хорошей пропускной способности сетевого канала может быть эффективным и в глобальных сетях или Интернете. Таким образом создается распределенная учебная среда, в которой все участники могут получить доступ к рабочим столам друг друга. В качестве примера программного обеспечения, реализующего эти принципы, можно привести программный продукт NetOp School, производимый компанией Axis Projects.

Программы для передачи знаний по определенным предметам

Программы этого типа в интерактивной форме позволяют получить знания по тому или иному предмету обучения или в той или иной области знаний. Таких программ на сегодняшний день множество, как коммерческих, так и свободных. В качестве примера приведем программу «Интерактивная таблица Менделеева», позволяющую получить исчерпывающие сведения о каждом элементе периодической таблицы (рис. 15.14).

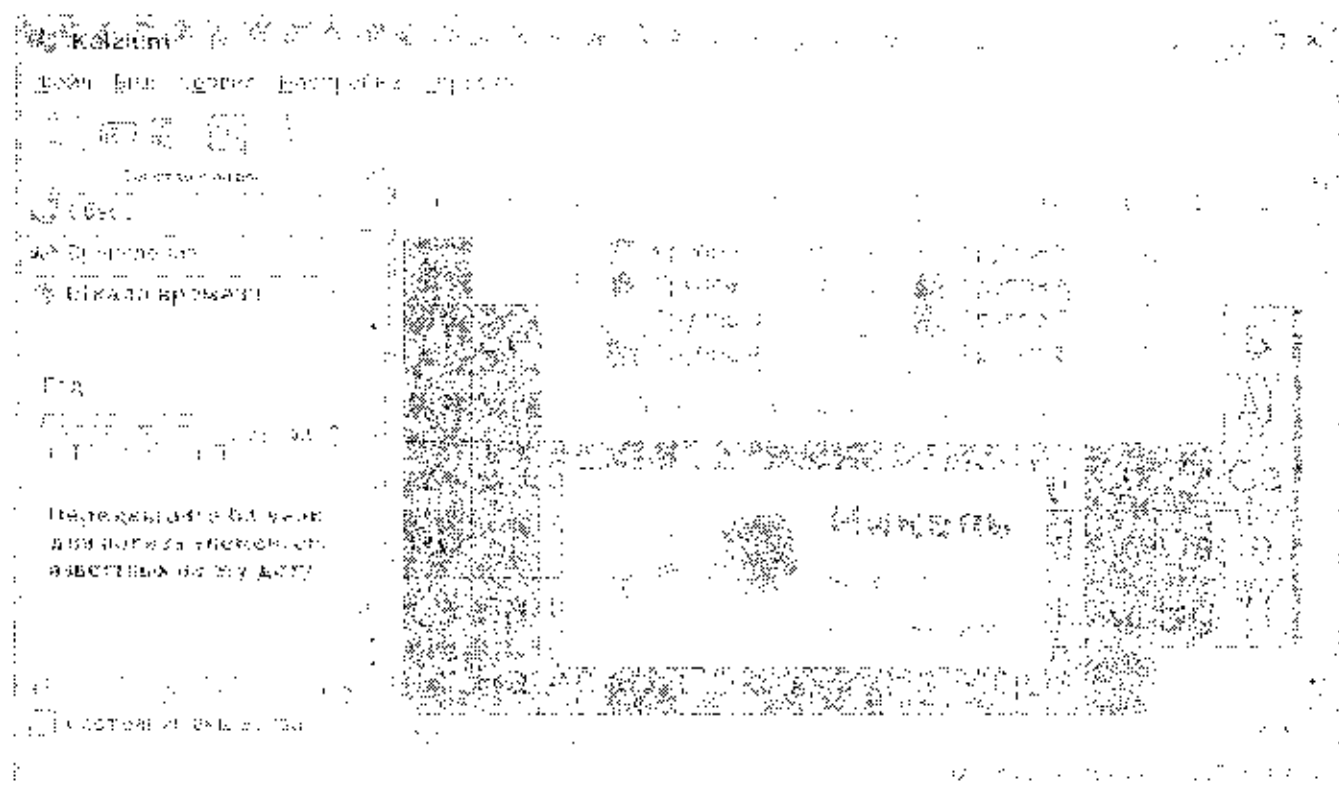


Рис. 15.14. Таблица Менделеева

Программы для компьютерного тестирования и тренинга

Программ для компьютерного тестирования и тренинга, как в свободном доступе, так и платных, сейчас великое множество, начиная от простых программ с ответом на десяток вопросов с одиночным выбором варианта и заканчивая мощными системами компьютерного тестирования и подтверждения квалификации с сетевой регистрацией, широким диапазоном методов задания вопроса и ответа на него, и базой вопросов, состоящих из десятков тысяч различных вариантов.

Профессиональные системы компьютерного тестирования обладают к тому же встроенным интеллектom, и если в ходе опроса вы не можете ответить правильно на какой-то вопрос, они зададут его снова, но перефразировав. Если ответ будет опять неверный, система начнет проверять знание темы в целом.

Образовательная операционная система России

В России в 2008 г. была закончена разработка и проведена апробация образовательного дистрибутива Linux, получившего общее название «Школьный Линукс». Этот образовательный дистрибутив, основанный на решениях Alt Linux Desktop и Alt Linux Server, имеет несколько исполнений:

- ❑ *Мастер* — наиболее полная версия, предназначенная для «хорошей» аппаратной платформы (с объемом ОЗУ 2 Гбайт и выше);
- ❑ *Юниор* — наиболее распространенное решение, предназначенное для большинства школьных компьютеров, отличается от исполнения Мастер только отсутствием наиболее ресурсоемких пакетов, таких как Eclipse;

- ❑ *Лёгкий* — специальное облегченное решение для старых компьютеров с объемом оперативной памяти от 512 до 128 байт;
- ❑ *Терминал-сервер* — решение для одного мощного компьютера и класса из старых компьютеров с объемом оперативной памяти от 32 до 64 Мбайт;
- ❑ *Сервер* — серверное решение с набором образовательного серверного программного обеспечения, предназначенное для интеграции школьных компьютеров в сеть с единым пульсом, контентной фильтрацией трафика, средствами коллективного взаимодействия (Media Wiki) и электронного обучения (Moodle).

Образовательный дистрибутив содержит исчерпывающий набор офисных, системных и сетевых программ на любой вкус. Кроме того, в образовательный дистрибутив включено множество специализированных научных, образовательных и игровых приложений. Мощная база средств разработки позволит учащимся хорошо освоить разнообразные приемы программирования и проектирования программ на разных языках программирования и в разных средах.

15.6.6. Программы для организации работы учебных заведений

Программы для компьютеризации управления школьным процессом и облегчения работы школьной администрации, взаимодействия с родителями, фиксации различных событий в жизни учащихся, мониторинга состояния их здоровья и сопровождения учебного процесса (компьютерный классный журнал, компьютерный дневник, сетевое родительское собрание) разработаны и применяются давно, но имеют в основном зарубежное происхождение. Однако, как известно, в некоторых сферах деятельности стандарты и формальные критерии расходятся. Так было с программами бухгалтерского учета, которые для нашей страны пришлось создавать практически «с нуля», так произошло и с программами управления работой школой или вузом: слишком разными оказались структуры учебных заведений, критерии оценивания, зачисления, разбиения на группы и дисциплины. Да и законодательная база долгое время не поощряла разработку таких программ.

Самыми первыми программными продуктами, облегчающими жизнь администрации учебного заведения, были программы составления расписания занятий с учетом нагрузки преподавателей, аудиторий, предметов и прочих параметров. Эти программы не требовали знания каких-то особенных стандартов и документов, решение задачи распределения ресурсов по времени — это чистая математика. Одна из успешных реализаций таких программ, Ректор, показана на рис. 15.15.

Однако составлением расписания жизнь и административные задачи учебного заведения не ограничиваются. Тематическое планирование уроков, учет посещаемости и оценки, различные школьные мероприятия, контакты с родителями — все это так же требует определенной программной поддержки. Такая поддержка реализована в программе Net-Школа (рис. 15.16).

В этой системе автоматизировано множество функций школьного администрирования. Но и она не может быть беспрерывно использована в школьном

процессе, и дело не в программировании, а в юридическом и финансовом оформлении многих операций: встает проблема школьного журнала, который придется дублировать дважды, в электронном виде, и в бумажном; не решена проблема финансирования рассылок родителям отчетов в виде SMS-сообщений.

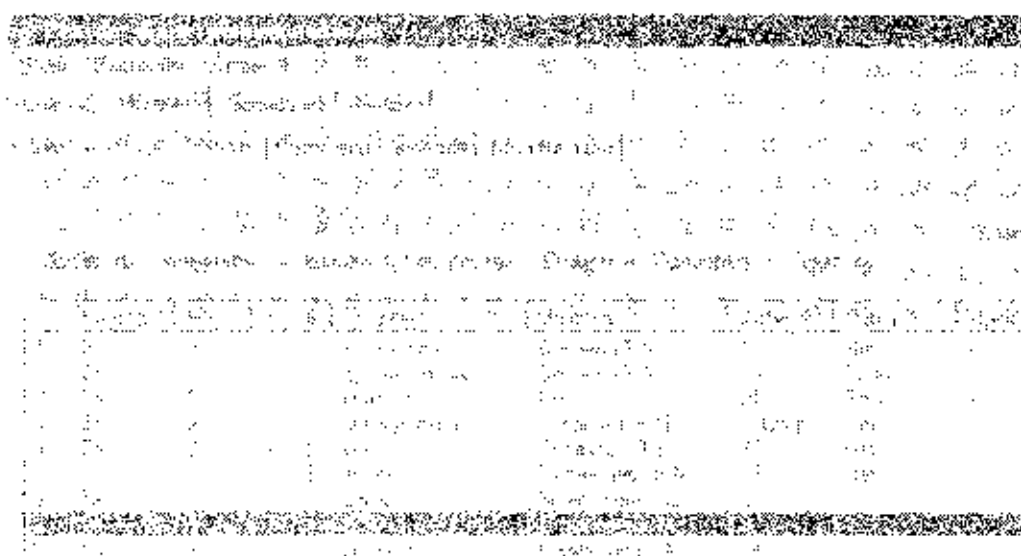


Рис. 15.15. Составление расписания в программе Ректор

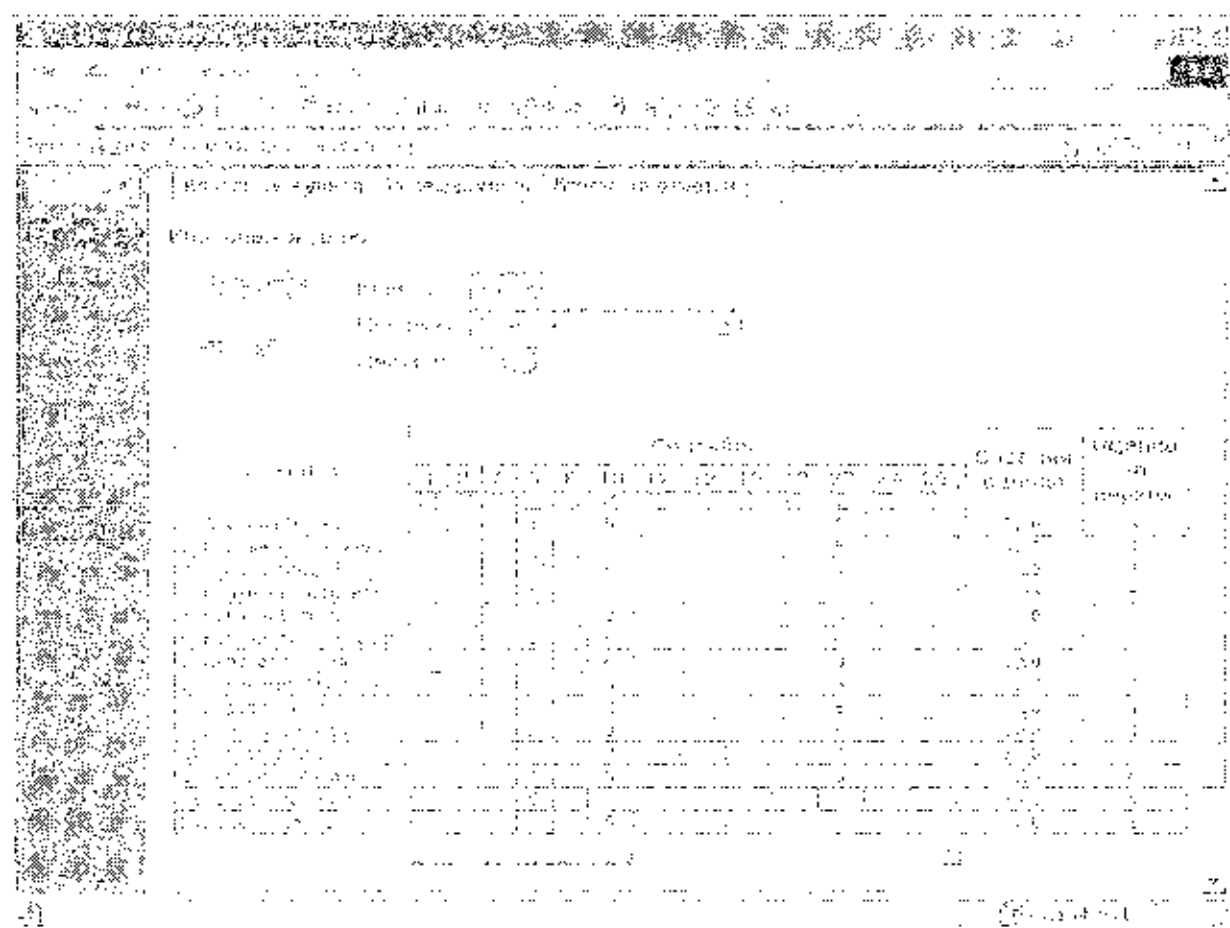


Рис. 15.16. Школьная административно-информационная система Net-Школа

15.6.7. Мультимедийные программы

К классу мультимедийного ПО относятся программы, при помощи которых можно создавать, редактировать, сохранять и воспроизводить мультимедийные данные, то есть данные, содержащие стационарные и движущиеся изображения, звук. К мультимедийному программному обеспечению относится целый ряд очень популярных программ: графические редакторы растровых форматов Adobe Photoshop и GIMP, векторные графические редакторы Corel Draw и Corel Xara, программы для создания и редактирования flash-анимации, программы для работы со звуком, и целый ряд проигрывателей мультимедийных форматов, от программ просмотра рисунков до проигрывателей DVD-дисков.

15.6.8. Бухгалтерские программы

Бухгалтерские программы представляют огромный класс приложений. Это могут быть и автономные программные продукты, и программные модули, входящие в информационную систему. Среди отечественных бухгалтерских программ наиболее известна программа 1С: Бухгалтерия. Начатая когда-то как автономная программная среда для бухгалтерских расчетов, в настоящее время она трансформировалась в информационную систему, включающую в себя модули для кадрового учета (1С-кадры), складского учета (1С-склад), планирования финансовой деятельности промышленных предприятий (1С-предприятие) и торговых фирм (1С-торговля). Этот программный продукт является коммерческим.

Среди свободного программного обеспечения тоже есть решение для автоматизации бухгалтерского и экономического учета предприятий (Ananas), которое при грамотном применении может оказаться во многих случаях более целесообразным, чем достаточно дорогая и требующая специального обучения система 1С.

15.6.9. Программы для финансовых расчетов и прогнозирования

Основное назначение таких программ — выполнение финансовых расчетов. Подобные программы могут выполнять следующие функции:

- ☐ разработка бизнес-плана предприятия;
- ☐ проектирование развития бизнеса;
- ☐ анализ финансового состояния предприятия на основе его финансовой отчетности;
- ☐ расчет финансовых показателей;
- ☐ расчет кредитоспособности заемщика;
- ☐ подготовка годового отчета предприятия;
- ☐ сравнение финансового состояния предприятия с компаниями-конкурентами;

- анализ рентабельности, платежеспособности, ликвидности и финансовой устойчивости;
- анализ планируемой инвестиционной деятельности.

В качестве примера программного обеспечения такого типа можно привести пакет программ фирмы Expert Systems: Project Expert, Audit Expert и Prime Expert. Эти программы позволяют производить все упомянутые виды финансового анализа и планирования, оценки рисков и возможностей предприятия.

15.6.10. Программы для технического проектирования

Современную промышленность и строительство невозможно представить без программных пакетов. Сроки разработки и выпуска продукции, так же как и сроки разработки проектной документации на строительство зданий, становятся определяющими в конкурентной борьбе. Современные системы автоматизированного проектирования позволяют создавать чертежи деталей, узлов и устройств на компьютере, причем сразу в трехмерном виде, и тут же производить расчеты прочности, износостойкости и других определяющих технических характеристик. Наиболее известными программами этого класса являются Autodesk Autokad во всех модификациях, позволяющих производить автоматизированное проектирование от механических деталей до химических соединений, и Graphisoft ArchiCAD, которая предназначена для архитектурного проектирования.

Кроме этих весьма недешевых программных продуктов есть целая линейка разного рода специализированных программ, как коммерческих, так и свободных.

15.6.11. Программы для бизнеса

Программное обеспечение для бизнеса включает в себя самые разнообразные типы программных пакетов:

- программное обеспечение управления работой промышленного предприятия;
- программное обеспечение управления технологическим процессом;
- специализированное программное обеспечение для отраслей промышленности;
- специализированное программное обеспечение по видам производств;
- специализированные информационные системы для видов бизнеса;
- программное обеспечение для малого бизнеса;
- программное обеспечение для сетевого бизнеса.

Для предприятий крупного и среднего масштаба стали уже стандартом готовые системы планирования ресурсов (Enterprise Resource Planning — ERP). Наиболее известными программными пакетами такого класса являются SAP R/3 от компании SAP AG и Oracle eBusiness Suite от компании Oracle. Из российских программных пакетов наибольшее распространение получил пакет Галактика ERP от корпорации Галактика, а также 1С: Предприятие.

ERP-системы получили широкое распространение за счет своей модульной структуры, которая позволяет осуществлять гибкое конфигурирование программного продукта под нужды любого предприятия. Например, Oracle eBusiness Suite включает в себя подсистемы управления:

- ☐ эффективностью бизнеса;
- ☐ материальными потоками;
- ☐ взаимоотношениями с клиентами;
- ☐ финансами;
- ☐ техобслуживанием и ремонтом;
- ☐ персоналом;
- ☐ производством;
- ☐ проектами;
- ☐ жизненным циклом;
- ☐ логистикой.

ERP-система очень гибкая в настройке, легко адаптируется по месту и может удовлетворить практически любые потребности бизнеса в управлении.

15.7. Перспективы развития программного обеспечения

15.7.1. Ориентация на взаимодействие

Одновременно с развитием сетевых технологий программное обеспечение становится все больше ориентированным не на индивидуальную работу или развлечение, а на взаимодействие -- рабочее или социальное. Все больше возникает интернет-порталов и социальных сетей, в которых люди находят себе группы по интересам, сами технологии взаимодействия непрерывно совершенствуются. Когда-то чудом было получить письмо через несколько минут после его отправки, сегодня уже не редкость сидеть за компьютером, общаясь с несколькими людьми голосом и визуально, видеть их на экране компьютера, зная, что они видят тебя. Компьютерные лаборатории ведут активные разработки технологий и программного обеспечения, позволяющие передавать через сетевые коммуникации вкус, запах, кинестетические ощущения.

15.7.2. Интеграция с сетью

Компьютерное программное обеспечение все больше и больше интегрируется с локальными и глобальными сетями. Все больше приложений, которые мы привыкли видеть в качестве настольных, становятся сетевыми. Показательным примером может служить служба документов Google, позволяющая выполнять

полноценные офисные операции с документами. Например, можно создавать документы, электронные таблицы и презентации в режиме подключения к сети, не устанавливая офисные пакеты на свой компьютер (рис. 15.17).

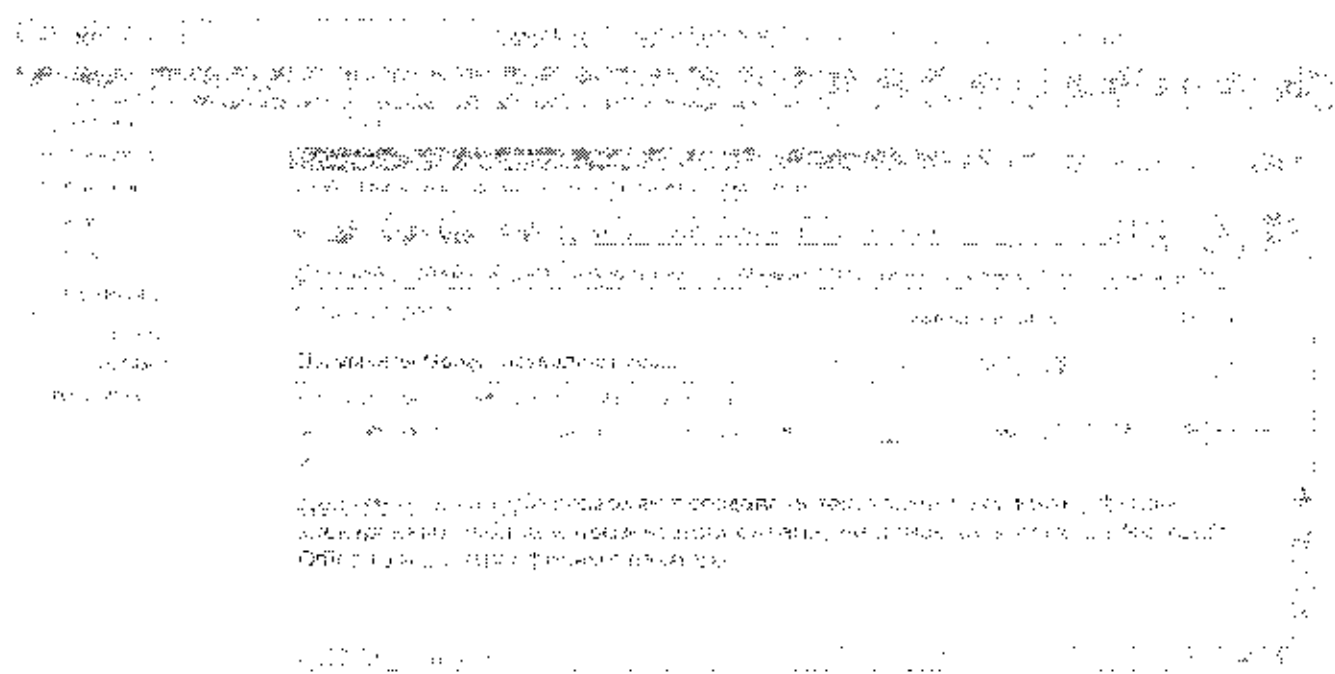


Рис. 15.17. Документ, создаваемый в сети

15.7.3. Компьютеры без операционной системы

Развитием идеи интеграции программного обеспечения с сетью будут компьютеры, состоящие из одного «железа» и небольшой программы подключения к сети и загрузки сетевых приложений. Необходимое программное обеспечение можно получить из Интернета после включения компьютера в сеть, например, рабочий стол, приложения, хранилища для файлов. Документы Google — это только первый шаг на пути тотальной сетевой интеграции.

15.7.4. Распределенность

Развитие и расширение каналов сетевых коммуникаций оказало влияние и на технологии создания приложений различных классов: все чаще на компьютере пользователя оказывается только каркас, оболочка, а все необходимые вычисления производятся где-то в сети. Причем тенденцией сегодняшнего дня является сокрытие от пользователя, средств выполнения вычислений и мест хранения «строительных блоков» запущенного приложения.

15.7.5. Автономизация и самообучение, повышение интеллектуальности программного обеспечения

Создаваемое сегодня программное обеспечение все чаще обретает черты интеллектуальности. Оно не просто решает наши задачи, оно пытается угадать, что

нам удобно, проанализировать, что мы делаем чаще, и предложить нам сформированные инструменты для этих действий или готовые решения. Ведутся активные разработки в области создания автономных компьютерных информационных систем, способных длительное время функционировать без вмешательства человека, самообучающихся, самообслуживающихся и самостоятельно определяющих свои цели.

15.7.6. Интеграция с бытовой техникой

Все большее программное обеспечение влияет не только на нашу работу и развлечения, но и на повседневную жизнь: программное обеспечение «умного дома» позволяет полностью переложить многие операции на плечи компьютеров и связанной с ними аппаратуры. Все больше электронных устройств имеют не просто автоматизированную, но интеллектуально-компьютеризированную начинку. По мере удешевления микропроцессоров все больше усилий вкладывается в создание программного обеспечения для них, способного немедленно объединить недавно купленную бытовую технику с тем семейством приспособлений, которое уже имеется в наличии.

15.7.7. Новые виды пользовательского интерфейса

Одна из тенденций в развитии программного обеспечения последних лет состоит в том, чтобы отказаться от сложившегося графического пользовательского интерфейса (рабочий стол, контекстные меню, главное меню в программах, стандартные кнопки окон) и найти какое-то новое решение. Сегодня еще нельзя сказать, в правильном ли направлении идут эти поиски, но ведь и компьютерная технология не стоит на месте: вполне возможно, лет через пять эти вопросы перестанут быть актуальными, и разрабатываться будут новые интерфейсы к трехмерному рабочему столу, проецируемому в пространство пакетом голографических лазеров.

15.7.8. Социальные вызовы развития программного обеспечения

Несвобода человека

Все возрастающая свобода компьютера от программного обеспечения, а иногда от дисковых накопителей и от устройств хранения пользовательской информации, во многом благо: все, что вам нужно, это компьютерная сеть и возможность подключиться к ней. В сети вы находите программы, возможно, даже операционную систему, в сети лежат ваши документы, отредактировав или переслав их, вы снова сохраняете их в сетевых хранилищах. Удобно? — Да, но при этом ваша свобода от программного обеспечения обменивается на несвободу в другом отношении. Какова степень конфиденциальности, защищенности, доступности документов?

Приятно заходить в «интеллектуальный виртуальный дом», который во всем соответствует вашим желаниям и старается их предугадать. Но кто может гаран-

тировать, что завтра это также будут ваши желания, а не желания какого-нибудь хакера, взломавшего информационную систему управления вашим «разумным домом»?

Поэтому можно предсказать период долгого и настороженного отношения к некоторым тенденциям развития программного обеспечения, по крайней мере до тех пор, пока они соответствующим образом и надежно не будут отрегулированы в правовой сфере. В противном случае свобода, даруемая технологиями, может обернуться неожиданным рабством.

Интеллектуальная деградация

Компьютеры становятся все «умнее» за счет все более сложного программного обеспечения. При этом интерфейс программ становится все проще и веселее. Не нужно думать, не нужно напрягаться, сложнейшие технологии, доступные в виде красочных значков и картинок, все сделают за вас: посчитают, оценят, спрогнозируют, подберут супруга, назначат диету. Ответов в Интернете стало гораздо больше, чем вопросов у людей, и это не может не вызывать тревогу. Если человек не тренирует мускулы, он деградирует физически, если человек перестает напрягать свой мыслительный аппарат, он деградирует интеллектуально. Это очень важный вызов, достойно ответить на который может только современная, построенная с расчетом на этот вызов система образования, в том числе система непрерывного образования на протяжении всей жизни.

Вопросы для самопроверки

1. Как можно классифицировать программное обеспечение по степени его взаимодействия с аппаратной частью компьютера?
2. Что такое «средства виртуализации»?
3. Какие классы программного обеспечения по виду лицензирования вы знаете?
4. В чем разница между свободным, открытым, коммерческим и проприетарным программным обеспечением?
5. В чем разница между переносимым и межплатформенным программным обеспечением?
6. Что такое «программа системного логика»?
7. Классифицируйте программное обеспечение по способу его взаимодействия с компьютерной сетью.
8. Что такое «опасное программное обеспечение»?
9. Перечислите известные вам классы прикладного программного обеспечения.
10. Каковы основные функции текстового процессора?
11. Каким требованиям должна отвечать современная электронная таблица?
12. Какова основная и дополнительная функциональность программы для создания и демонстрации презентаций?

13. Что такое «переносимое приложение» и какие два значения этого словосочетания вам известны?
14. Идентичны ли функционально программы, входящие в состав пакетов Microsoft Office и OpenOffice.org?
15. Для чего (с перечислением функций) предназначено программное обеспечение управления проектами?
16. Какие программы для работы с сервисами Интернета вам известны?
17. Какие функции выполняет образовательное программное обеспечение?
18. Как создавать документы и электронные таблицы в распространенных «офисных» форматах, не устанавливая на компьютер офисный пакет?
19. Каковы перспективы развития программного обеспечения?
20. Каковы социальные последствия интенсивного взаимодействия человека и современного программного обеспечения?

Литература

1. Виснадул Б. В., Гарашина Л. Г., Кокорева Е. В. Технология разработки программного обеспечения. М.: ФОРУМ, 2008.
2. Волков В. Б. Линукс Юниор. Книга для учителя. М.: ДМК, 2009.
3. Волков В. Б. Понятный самоучитель Excel 2007. СПб.: Питер, 2008.
4. Волков В. Б. Понятный самоучитель Word 2007. СПб.: Питер, 2008.

Глава 16 Системное программное обеспечение компьютера

- 16.1. Состав системного программного обеспечения
- 16.2. Классификация операционных систем
- 16.3. Сферы применения операционных систем
- 16.4. Представление о структуре операционной системы
- 16.5. Операционные системы семейства Windows
- 16.6. Операционная система Windows XP
- 16.7. Операционная система Alt Linux

Диапазон компьютерной техники на сегодняшний день широк, от суперкомпьютеров, занимающих целые машинные залы и имеющих терафлопсную производительность, до микрочипов в смарткартах, ориентированных на достаточно медленную работу и выполнение всего нескольких фиксированных операций. Несмотря на разительные различия в размерах, быстродействии, энергопотреблении и функциональности, все эти устройства объединяются между собой одним признаком: для их работы необходимо системное программное обеспечение.

16.1. Состав системного программного обеспечения

К системному программному обеспечению относятся: базовая система ввода-вывода (Base Input-Output System, BIOS), ядро операционной системы, утилиты системного администрирования, утилиты системного программирования, оболочки и прочее программное обеспечение, позволяющее пользователю запускать прикладные программы и работать с ними. Таким образом, к системному программному обеспечению можно отнести весь программный слой, обеспечивающий взаимодействие прикладных программ пользователя с аппаратными средствами компьютера (рис. 16.1).

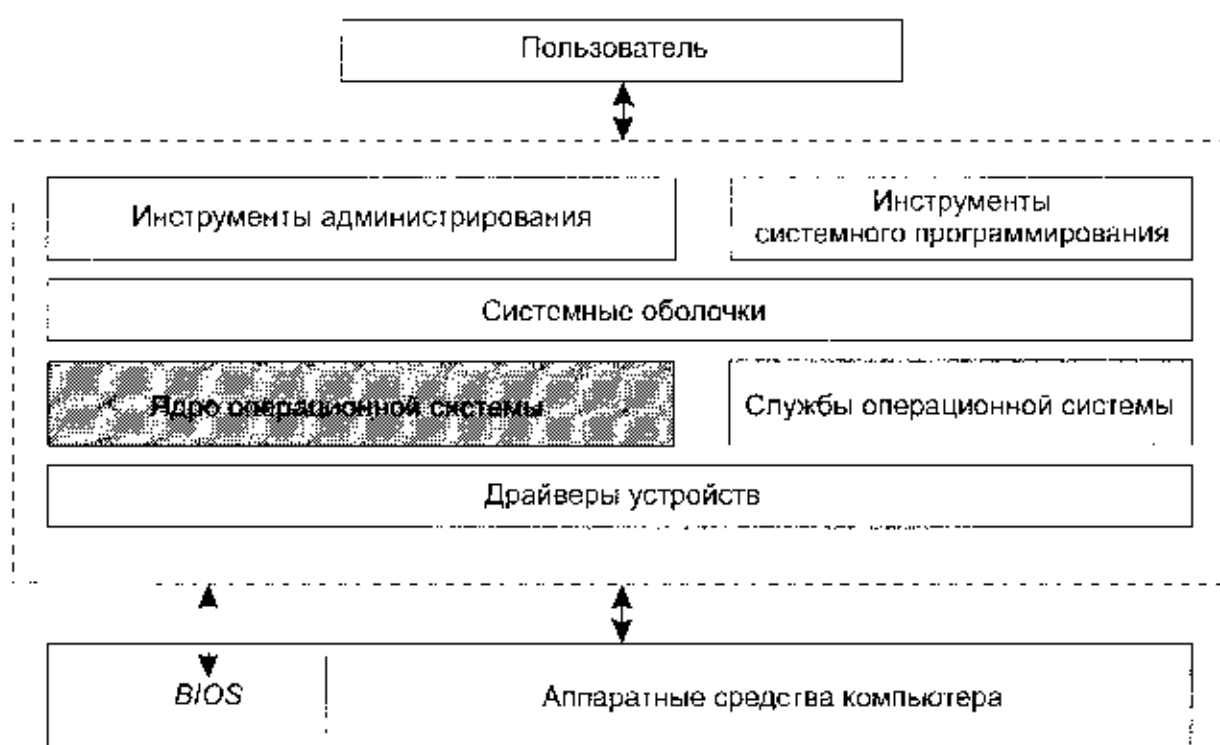


Рис. 16.1. Структура системного ПО

16.1.1. BIOS

Между моментом включения компьютера и моментом начала работы операционной системы есть промежуток времени, в который на компьютере работает подсистема BIOS, выполняющая несколько важных функций.

- ❑ *Тестирование аппаратного обеспечения.* Сразу после включения компьютера BIOS осуществляет тестирование аппаратного обеспечения. При этом проверяются все жизненно важные устройства компьютера. Если во время проверки обнаруживается, что то или иное устройство функционирует неправильно или отсутствует, из-за чего работа операционной системы становится невозможной (например, обнаружена ошибка в оперативной памяти), то процесс загрузки прерывается. Если же продолжение работы операционной системы возможно, на экран выводится предупреждение об отсутствии устройства (на-

пример, дисковод для гибких дисков), но процесс загрузки компьютера не прерывается.

- *Запуск загрузчика операционной системы.* Если тестирование прошло нормально, то осуществляется запуск загрузчика операционной системы, после чего основное управление компьютером передается операционной системе.
- *Проверка некоторых параметров аппаратного обеспечения компьютера.* Кроме этих двух операций, самопроверки при включении и запуска загрузчика, современные подсистемы BIOS выполняют контрольные и настроечные функции некоторых параметров аппаратного обеспечения компьютера. Например, используя параметры BIOS, можно в довольно широких пределах изменять частоту системной шины и тактовую частоту процессора, временные интервалы циклов чтения, записи и обновления оперативной памяти, параметры кэширования оперативной памяти. В компьютерах с многоядерными процессорами BIOS управляет режимами «многоядерности». Значительное расширение функциональности BIOS в современных компьютерах привело к тому, что операционные системы взаимодействуют с BIOS уже после того, как управление компьютером полностью передано им.

BIOS, в отличие от остального программного обеспечения, располагается не на дисковых устройствах компьютера, а внутри специальной микросхемы — постоянного запоминающего устройства. Микросхема поставляется вместе с материнской платой, и BIOS можно рассматривать как программную поддержку материнской платы.

16.1.2. Ядро операционной системы

После того как подсистема BIOS последовательно выполнит все действия по проверке и инициализации аппаратных устройств компьютера, загрузчик операционной системы загружает в память компьютера ядро операционной системы. С этого момента и до окончания сеанса работы с компьютером (до его выключения) ядро операционной системы постоянно находится в оперативной памяти компьютера, и функционирование всех остальных программных компонентов, как системных, так и прикладных, происходит только через взаимодействие с ядром.

Важным свойством ядра является то, что процесс ядра работает в привилегированном режиме, то есть процесс ядра может прервать любой другой процесс, но ни один процесс не может прервать процесс ядра; процесс ядра может выполнять инструкции, которые не могут выполнять программы. Ядро обеспечивает загрузку программ, как прикладных, так и системных, ядро выделяет программам ресурсы, ядро обеспечивает взаимодействие программ между собой и с аппаратным обеспечением компьютера. Ядро является самым необходимым компонентом системного программного обеспечения. На рис. 6.1 ядро специально выделено, а границы операционной системы (ОС) обозначены пунктирной линией. Прикладные программы внесены внутрь этих границ, так как они запускаются и работают внутри операционной системы и только под ее управлением.

16.1.3. Драйверы устройств

Современные компьютеры все больше и больше походят по своему дизайну на конструктор: они легко собираются из функциональных блоков, созданных разными производителями. На уровне механическом совместимость разных устройств, корпуса компьютера и его материнской платы обеспечивается стандартизацией размеров и разъемов устройств. На аппаратном уровне необходимую совместимость обеспечивает соблюдение стандартных значений напряжения, тока, параметров импульса и последовательности следования сигналов.

На уровне операционной системы взаимодействие устройств различных типов с операционной системой, а через нее -- и с прикладными программами, осуществляется через небольшие загружаемые блоки машинного кода, так называемые драйверы. *Драйвер* осуществляет стыковку стандартизированного программного интерфейса операционной системы и системы команд внешнего устройства.

Драйвер — программа, обеспечивающая операционной системе доступ к определенному устройству.

16.1.4. Службы операционной системы

Обычно программы, как прикладные, так и системные, запускаются по мере необходимости, выполняют свою работу, а затем завершаются. Но есть особый тип программ, которые должны постоянно находиться в ожидании запросов от других программ или отслеживать состояние некоторых параметров операционной системы. Такие программы запускаются и завершают свою работу одновременно с операционной системой. Если в такой программе происходит сбой и она аварийно завершается, то программа автоматически снова загружается в память. Такого рода программы называются *службами*, *сервисами*, или *демонами*, операционной системы. Примером службы операционной системы может быть служба печати: эта служба может принимать документы одновременно от нескольких программ (компьютеров), ставить их в очередь и затем поочередно выводить на печать.

16.1.5. Системные оболочки

Программы взаимодействуют с операционной системой через интерфейс, называемый API (Application Programming Interface -- прикладной программный интерфейс). Программа, которая должна воспринимать действия пользователя и переводить их на язык, понятный операционной системе, называется *системной оболочкой* (shell). Оболочки могут быть как графическими (оконные менеджеры), так и текстовыми, предназначенными для работы в терминальном режиме.

Одни операционные системы, например Windows и OS/2, изначально были ориентированы на графическую оконную среду. Другие изначально отдавали предпочтение общению с пользователем с помощью символов и текстов, поэтому графическая среда в них представляет собой надстройку, без которой операцион-

ная система может работать вполне эффективно. На рис. 6.2 показаны примеры графических оболочек Alt Linux и Solaris X, а также командного интерпретатора терминального режима Windows XP.

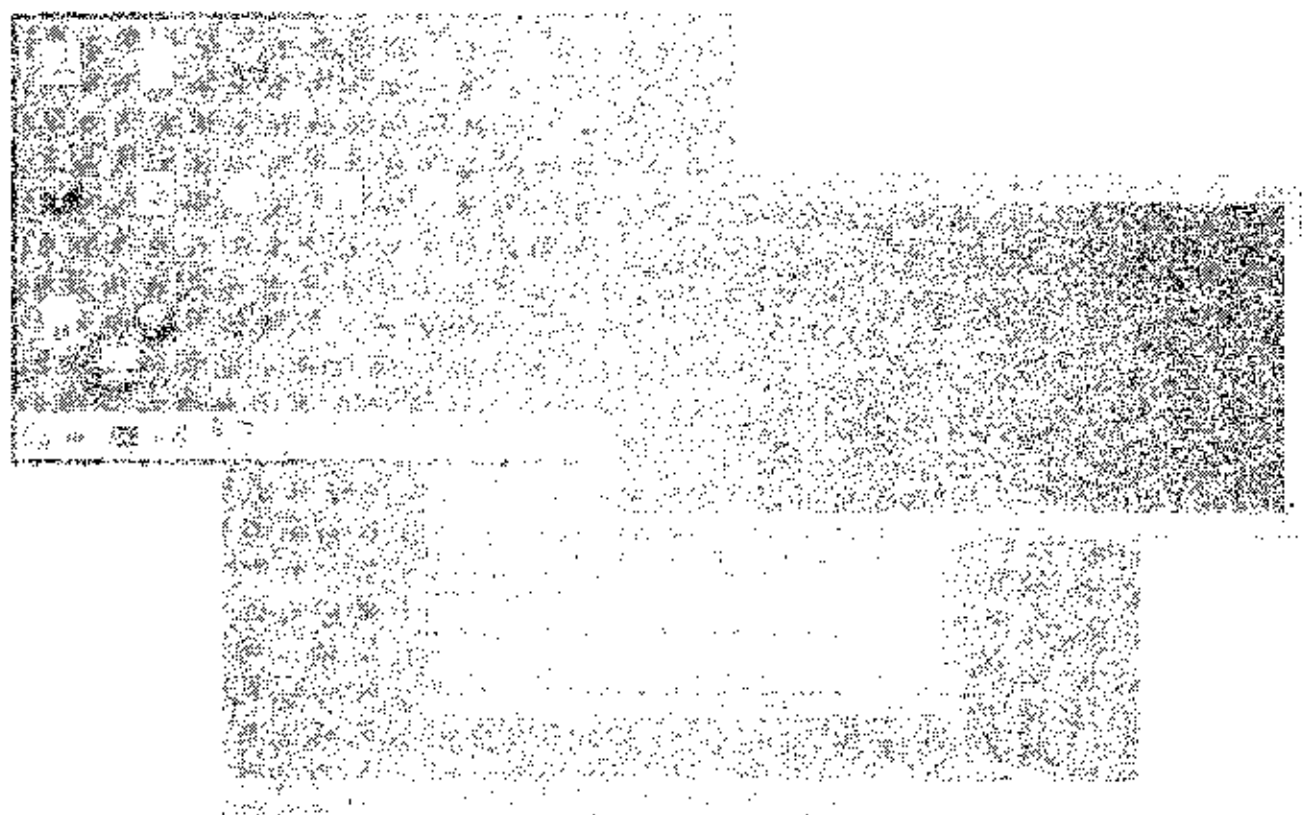


Рис. 6.2. Оболочки ОС

16.1.6. Инструменты администрирования

В операционной системе необходимо решать задачи, связанные с настройкой или перенастройкой определенных параметров, добавлением пользователей, оборудования, установкой новых прикладных программ и их конфигурированием. Кроме того, нужно поддерживать систему в рабочем состоянии и обеспечивать достаточный уровень защиты от несанкционированных действий. Для этих целей существует целый ряд программ, называемых *системными утилитами*. К функциям, выполняемым этими программами, можно отнести добавление, удаление и назначение прав пользователей; управление файлами и файловыми системами; управление дисковыми устройствами; управление сервисами операционной системы; добавление и конфигурирование новых устройств; установку и удаление программ; настройку графической оболочки; настройку сетевых подключений; настройку печати и множество других.

16.1.7. Инструменты системного программирования

Данный вид системного программного обеспечения, безусловно, существует во всех операционных системах, но далеко не во всех входит в комплект поставки. Чаще всего в коммерческих операционных системах инструменты системного

программирования продаются отдельно и стоят довольно дорого, как и различного рода закрытые спецификации или фрагменты исходных кодов. Для свободного и открытого ПО обычным является обратное — включение инструментов системного программирования в состав поставки (дистрибутив) операционной системы и полная доступность исходных кодов. К инструментам системного программирования можно отнести языки программирования, текстовые редакторы, трансляторы, редакторы связей, компоновщики программ, библиотеки подпрограмм. К современным средствам программирования можно также отнести большие интегрированные многоязыковые среды программирования, такие как KDeveloper, Borland Developer Studio или Microsoft Visual Studio.

16.2. Классификация операционных систем

Операционные системы можно классифицировать по разным критериям.

- ❑ По *количеству пользователей*, имеющих возможность одновременно работать с операционной системой:
 - однопользовательские (MS-DOS);
 - многопользовательские (UNIX, Windows XP, Linux).
- ❑ По *механизму обеспечения доступа*:
 - пакетные (OS 360);
 - интерактивные (UNIX, Windows, Linux);
 - реального времени (QNX).
- ❑ По *количеству одновременно решаемых задач*:
 - однозадачные (MS-DOS);
 - многозадачные (UNIX, Windows XP, Linux).
- ❑ По *функциональности*:
 - общего применения;
 - специализированные.
- ❑ По *возможности обслуживать разное количество процессоров*:
 - однопроцессорные;
 - с поддержкой многопроцессорности;
 - с поддержкой распределенных вычислений.

Однозадачные операционные системы непосредственной обработки

На ранней стадии развития компьютерной техники первые операционные системы работали в непосредственном режиме: из устройства ввода программа

и данные считывались с носителя информации в память компьютера. Затем процессор выполнял задание и выводил результаты на устройство печати. Далее вся последовательность действий повторялась (рис. 16.3).

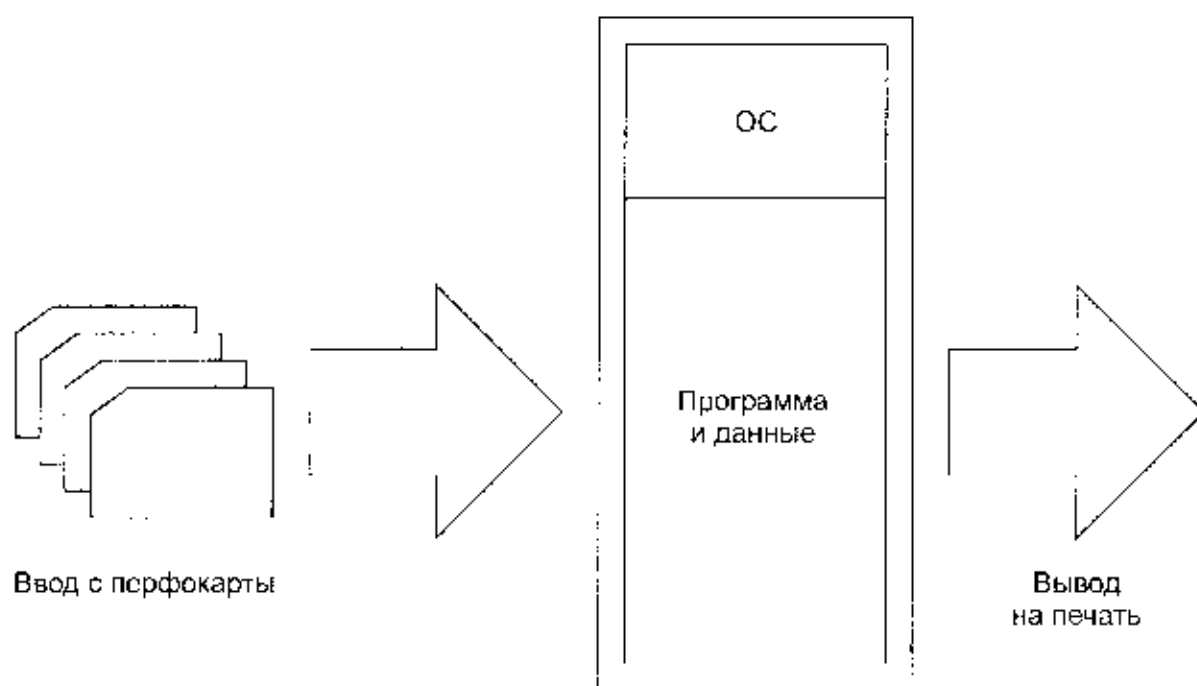


Рис. 16.3. Операционная система непосредственной обработки

Операционные системы такого рода были весьма несовершенны. Их главный недостаток состоял в том, что процессор довольно длительное время простаивал, ожидая ввода информации. Еще одним недостатком была невозможность управлять заданиями. Например, если в вычислительную машину загружалось длительное по времени выполнения задание, а в этот момент требовалось сделать срочный расчет, приходилось либо откладывать расчет, либо прерывать длительное задание и потом начинать его заново. То и другое вело к значительным потерям машинного времени.

Однозадачные операционные системы пакетной обработки

Появление устройств чтения и записи перфоленов и магнитных лент немного изменило ситуацию в лучшую сторону: задания (программа и данные для нее) записывались на магнитную ленту (МЛ) на отдельном устройстве. Когда лента заполнялась, получался пакет программ. Этот пакет переносился на вычислительную машину и запускался там. Результаты вычислений также записывались последовательно на вторую, выходную, ленту. Когда выходная лента заполнялась, ее заменяли новой, а заполненную распечатывали на отдельном от вычислительной машины устройстве. Такой режим работы позволял добиться почти непрерывной загрузки центрального процессора (рис. 16.4)

Группировка программ в пакеты позволила более эффективно использовать время центрального процессора при решении длительных по времени задач. Однако с задачами другого рода, например бизнес-задачами, этот режим работы

нельзя было назвать эффективным, так как время расчета было примерно равным, а иногда и много меньше, чем время операций считывания и записи на магнитную ленту. Задача прерывания одного задания ради выполнения другого решена не была.

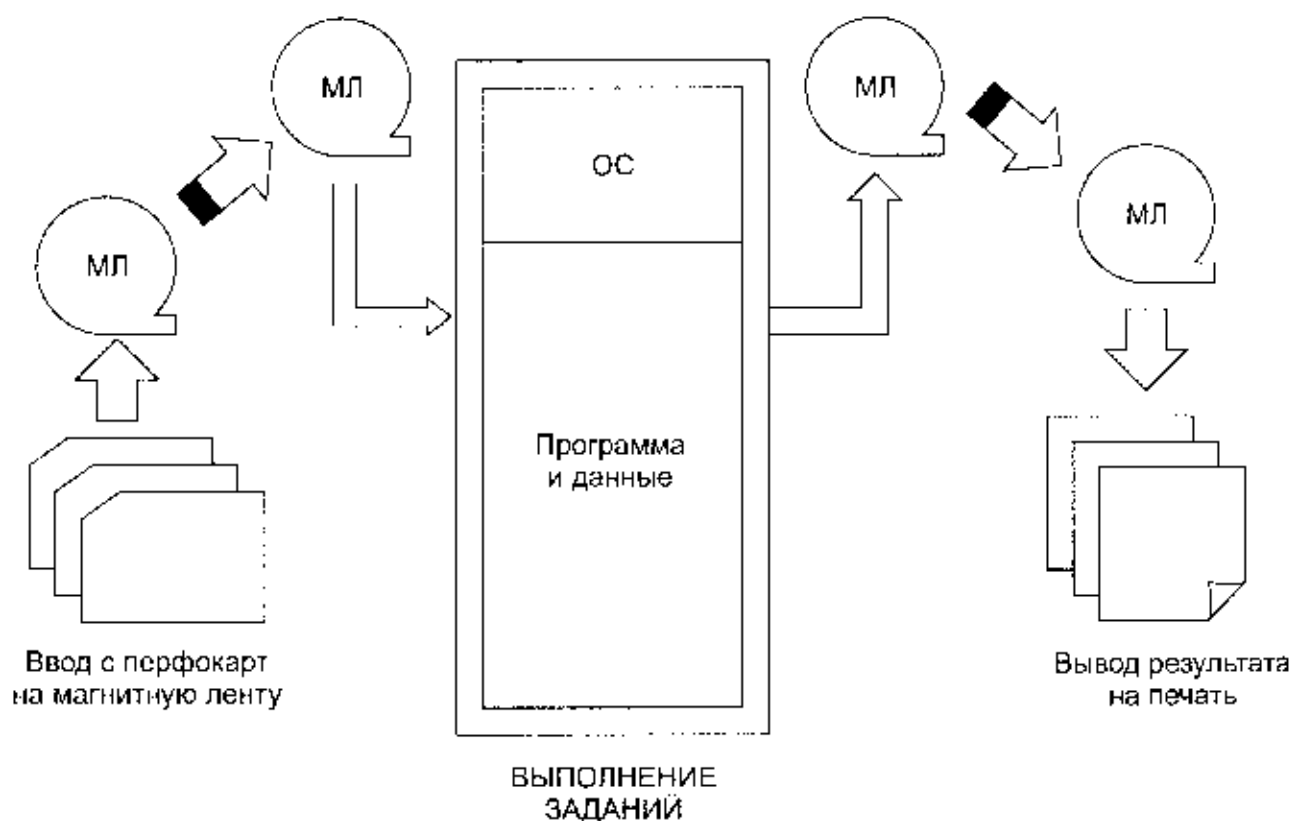


Рис. 16.4. Пакетная обработка

Многозадачные операционные системы без разделения времени

Очередные усовершенствования в архитектуре компьютеров вызвали к жизни появление нового поколения операционных систем. Увеличение объема оперативной памяти и появление дисковых устройств сделало возможным запись программ и данных на магнитный диск (МД) с последующим их считыванием оттуда в произвольном порядке, а не последовательно, как это было с магнитной лентой. Увеличение объемов оперативной памяти позволило загружать больше чем одну программу. Пока одна программа считывалась с диска в память, другая выполнялась. Пока результаты выполнения одной программы записывались на диск, процессор продолжал работу над другой программой (рис. 16.5).

Однако самым важным моментом на этом этапе развития операционных систем было появление возможности управлять заданиями. Поскольку задания можно было одновременно размещать в разных сегментах памяти, можно было в любой момент приостановить работу над одним заданием и переключиться на выполнение другого. Кроме того, порядок выполнения заданий стал не последовательным, а произвольным, причем этот порядок мог быть изменен оператором.

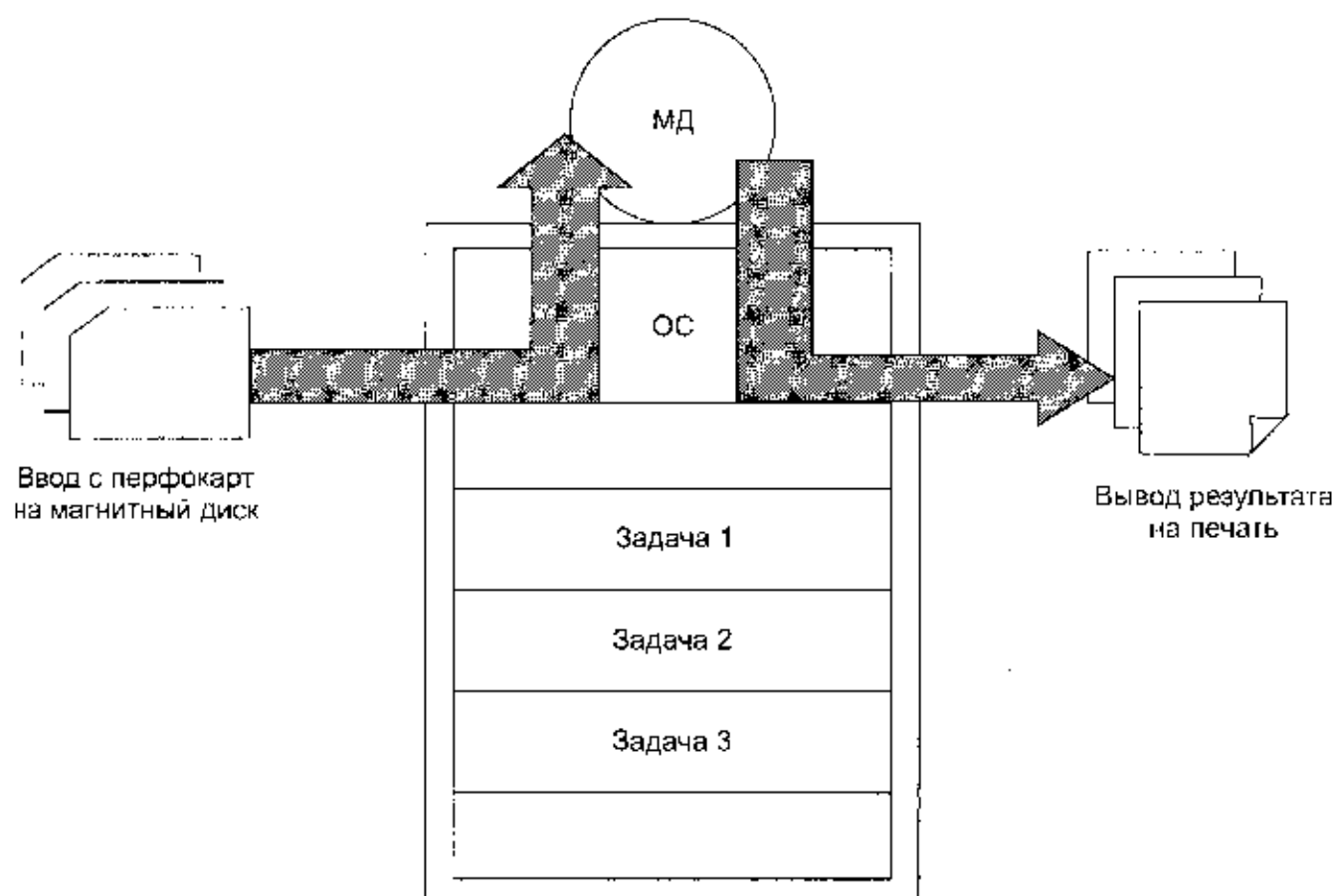


Рис. 16.5. Многозадачная операционная система без разделения времени

Многозадачные операционные системы с разделением времени

История современных операционных систем ведет отсчет от появления первой многозадачной ОС с разделением времени. В таких операционных системах между заданиями делится не только оперативная и дисковая память компьютера, но и время центрального процессора. Такое разделение позволило организовать как псевдопараллельное выполнение заданий, так и многопользовательский режим работы, когда обеспечивается подключение к центральному компьютеру множества терминалов (сочетание устройства ввода и вывода, дисплея и клавиатуры). Первой операционной системой с разделением времени, получившей сравнительно удачную реализацию, была ОС MULTICS. С момента разработки MULTICS компьютерный мир вошел в эпоху мейнфреймов и многопользовательских информационных систем.

ПРИМЕЧАНИЕ

Существует два типа многозадачности: кооперативная и вытесняющая. При кооперативной многозадачности решение о том, передать или нет управление от одного процесса другому, принимается на уровне текущего процесса. То есть несмотря на то, что отпущенное ему на выполнение количество квантов времени истекло, процесс может не отдать управление, если полагает, что выполняет критически важное задание. При вытесняющей многозадачности все решения о передаче управления от одного процесса к другому принимает ядро, которое принудительно останавливает выполнение процесса, сохраняет его текущее состояние и передает управление другому процессу.

Свое развитие ОС MULTICS получила в широко известном семействе операционных систем UNIX, которые и сегодня составляют значительную часть операционных систем, установленных на компьютерах во всем мире.

Операционные системы персональных компьютеров

С появлением персональных компьютеров отдельно обозначилось направление разработки однопользовательской однозадачной операционной системы.

Первые операционные системы MS-DOS, CP/M и APLLE DOS в качестве интерфейса имели командную строку и позволяли запускать в каждый момент времени одно задание. Пользователь мог либо играть, либо программировать, либо составлять текстовый документ, в любом случае в памяти компьютера в этот момент времени размещалась и выполнялась одна программа.

Большие изменения в концепции ОС для персональных компьютеров наступили в тот момент, когда, во-первых, появились аппаратные возможности реализовывать многозадачные приложения в компьютерах такого класса (к примеру, процессор Intel 80286 и 80386), во-вторых, были изобретены графические оконные интерфейсы.

Появление на экране нескольких окон изначально подразумевало, что в каждом окне может быть открыта отдельная программа, а этот факт сам по себе подразумевал наличие многозадачности. В одних случаях многозадачность сразу закладывалась в операционную систему компьютера вместе с графическим интерфейсом (в операционных системах Mac OS), или же, как в линейке Windows, к пей шли постепенно. Вначале Windows представляла собой просто графическую надстройку над MS-DOS, и только позже была разработана ОС Windows NT, принципы работы которой являются базовыми для всех современных версий Windows.

Так или иначе, все современные операционные системы персональных компьютеров являются многозадачными системами с разделением времени, а значит, они поддерживают возможность одновременного выполнения нескольких программ и многопользовательский режим. Реализуется многопользовательский режим или нет, зависит от комплекта поставки операционной системы.

16.3. Сферы применения операционных систем

Операционные системы различаются в зависимости от сферы применения.

- ❑ *Суперкомпьютеры.* Операционные системы для современных суперкомпьютеров представляют собой многозадачные многопроцессорные операционные системы с разделением времени, способные работать в сетевой распределенной среде. Обычно операционная система суперкомпьютера — это оснащенная специальными программными расширениями система UNIX или Linux. В последние два года наблюдаются попытки корпорации Microsoft выйти на этот рынок, становящийся все более популярным. Так, операционная система Windows HPC Server 2008 создана специально для высокопроизводительных кластерных

систем. В России высокопроизводительные кластерные системы работают под управлением отечественных разработок Linux (суперкомпьютер СКИФ-МГУ, операционная система ALT Linux 4.1 SKIF).

- *Мейнфреймы.* Операционные системы для мейнфреймов должны совмещать сразу три функциональности: высокопроизводительные автоматизированные пакетные вычисления (без вмешательства операторов), разделение времени для терминального доступа большого (сотни и тысячи) количества пользователей, обработка транзакций (групповые операции). Несмотря на то, что системы UNIX можно установить и сконфигурировать для успешной работы на мейнфрейме, оптимально там работают специализированные операционные системы, например OS/390.
- *Серверы.* Основная черта серверных систем — поддержка большого числа одновременно работающих служб, обслуживание множества клиентов и передача большого количества информации. Таким образом, серверные операционные системы должны быть многозадачными системами с разделением времени и развитыми сетевыми возможностями. Традиционно в качестве серверных систем использовались различные варианты UNIX. После появления Linux и Windows NT эти операционные системы также стали использоваться в качестве серверных¹.
- *Персональные компьютеры.* Операционные системы для персональных компьютеров в первую очередь должны обладать дружественным пользовательским интерфейсом. Хотя у систем семейства UNIX есть графическая оконная среда XWindows, долгое время она считалась менее «дружественной», чем Windows или Macintosh. Однако активная экспансия Linux в сферу персональных машин привела к тому, что эта операционная система сегодня может предложить пользователям не менее, а иногда и более удобные средства работы в графической оконной среде.
- *Налоботные компьютеры.* Малые размеры этих устройств изначально подразумевали ограничения в ресурсах, которые компьютер может выделить для выполнения заданий. В то же время название «компьютер» требует от этого устройства, чтобы номенклатура выполняемых пользовательских заданий была примерно такой же, какую обеспечивает пользователю настольный компьютер. Таким образом, операционная система наладонного компьютера должна эффективно функционировать с разделением времени в условиях ограниченных ресурсов. Эта задача решается либо специализированными версиями операционных систем (например, Windows CE или специальным образом собранный ОС Linux), либо разработанными специально для этого операционными системами (Palm OS).
- *Встраиваемые системы.* Встраиваемые операционные системы должны выполнять ограниченный набор операций, связанных со специфической функциональностью устройства, для которого они предназначены. Примером встраиваемой операционной системы может служить операционная система устройства для чтения электронных книг, операционная система аудио- или видеопроигрывателя или операционная система смарт-карты. Такого рода операционные

¹ Достаточно широко в качестве операционных систем серверов применяются различные ОС семейства BSD (например, FreeBSD).

системы крайне ограничены в ресурсах, но и их функциональность не слишком разнообразна. Очень часто они работают в однопользовательском однозадачном режиме или в режиме, когда пользователь вообще не имеет доступа к операционной системе (например, мини-роутер для работы в локальной домашней сети). Чаще всего в качестве встраиваемых операционных систем используются варианты Linux или виртуальная машина Java.

- ❑ **Системы реального времени.** Это очень важный класс операционных систем, предназначенных для работы с автоматизированными производственными, технологическими или военными устройствами. Особенность этих систем в том, что они должны реагировать на сигналы извне (например, от датчиков) в жестко фиксированные временные интервалы. Это значит, что обработке внешних сигналов в системе реального времени должен быть присвоен самый высокий приоритет, даже более высокий, чем процессам ядра. Одной из наиболее применяемых и известных сегодня системах реального времени является QNX.

16.4. Представление о структуре операционной системы

16.4.1. Операционная система в виде набора функций

Обобщенная структура операционной системы представлена на рис. 16.6. Любая операционная система, независимо от ее структуры, имеет определенный набор функциональных компонентов.

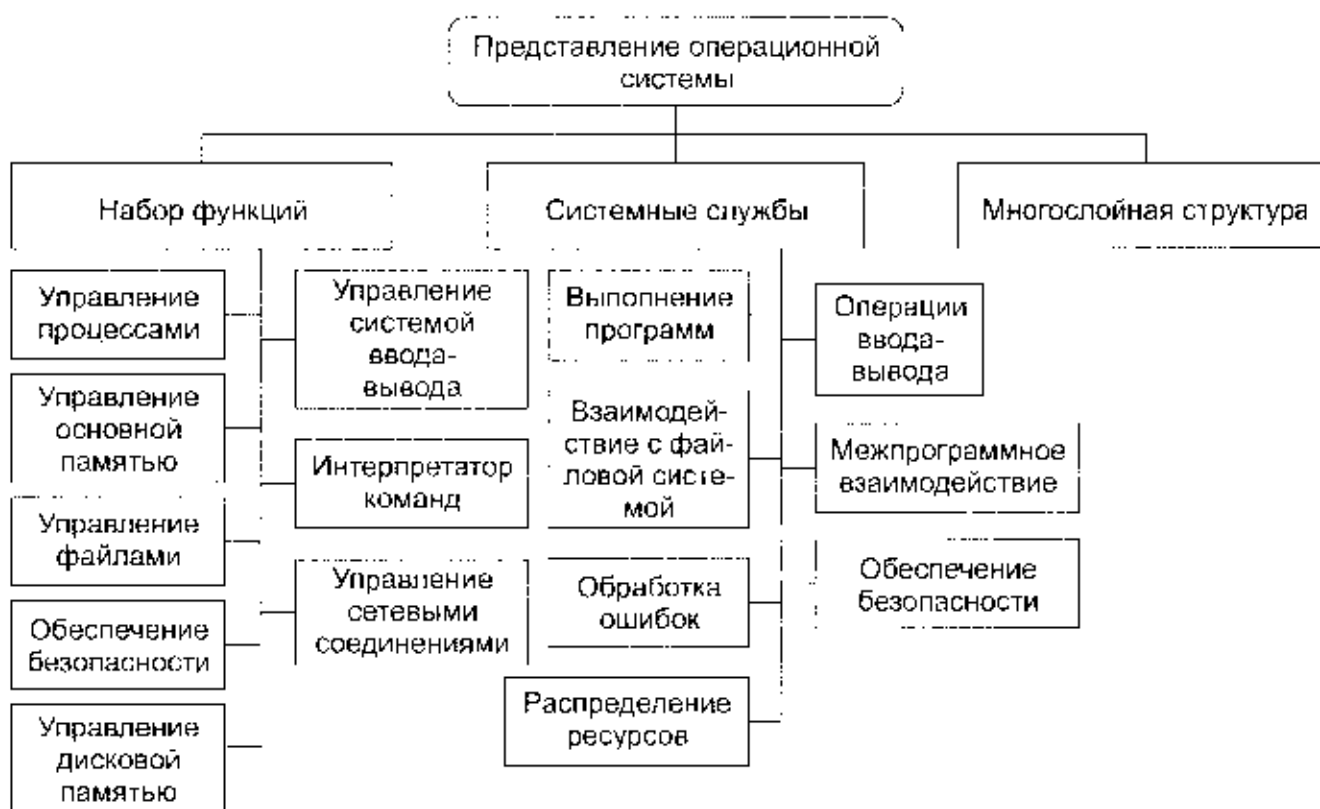


Рис. 16.6. Обобщенная структура операционной системы

Управление процессами

Понятие *процесса* является одним из наиболее важных в операционной системе. Процесс — выполнение в процессоре компьютера инструкций программы. До тех пор, пока набор команд процессора записан на диск в виде файла, он является программой. Когда же этот набор команд загружается в память компьютера и начинает последовательно выполняться, он становится процессом. Поскольку мы говорим о многозадачных системах, то в памяти компьютера одновременно находится и выполняется несколько таких последовательностей. Причем процессор, согласно заданным параметрам, периодически останавливает выполнение одной последовательности команд и переходит к другой, к третьей. Та последовательность команд, которая в данный момент не выполняется, должна хранить свое состояние, чтобы процессор мог продолжить ее выполнение с места ее остановки. Таким образом, любой процесс связан с памятью, которую он занимает, с точкой выполнения (счетчиком команд), пользователем и связанными ресурсами (открытыми файлами, соединениями или устройствами компьютера). Вся совокупность этих данных описывает состояние процесса и хранится в *таблице процесса*.

Операционная система должна быть способна:

- создавать или уничтожать системные и пользовательские процессы;
- приостанавливать и возобновлять выполнение процесса;
- поддерживать механизм синхронизации, взаимодействия и взаимного блокирования процессов.

Управление основной памятью

Основная, или оперативная, память — это быстродействующее хранилище данных, к которому у центрального процессора есть прямой доступ. В этом хранилище инструкции размещаются для их последующего выполнения. Процессор может разделять прямой доступ к основной памяти с устройствами ввода-вывода. В этом случае устройства ввода-вывода могут записывать в основную память массивы данных, которые процессор затем считывает для выполнения или обработки.

Операционная система отвечает за следующие аспекты управления основной памятью:

- следит за тем, какая часть памяти каким процессом или устройством используется;
- определяет, какие процессы должны быть загружены в память;
- выделяет и освобождает память.

Управление файлами

Файловая система является частью операционной системы и предназначена для организации, хранения и именования данных на носителях. Данные могут храниться на самых разных физических устройствах (жестких дисках, flash-картах, лентах или DVD-дисках). Операционная система обеспечивает логический уровень представления этих устройств, а также структуры хранящихся на них данных.

Логическая единица хранения данных — это *файл*. Файловая система определяет формат содержимого и физического хранения файлов. Для облегчения работы с файлами файлы объединяют в каталоги (папки). Для манипуляции с файлами операционная система должна предоставлять следующую функциональность:

- ❑ создание и удаление файлов;
- ❑ создание и удаление каталогов;
- ❑ поддержка базовых команд для манипулирования файлами и каталогами;
- ❑ осуществление резервного копирования.

Управление системой ввода-вывода

Операционная система должна скрывать от пользователя сложные детали взаимодействия с устройствами ввода-вывода. Поэтому в ОС всегда есть функциональная часть, отвечающая за взаимодействие пользовательских программ с этими устройствами. В этой области операционная система обычно обеспечивает:

- ❑ общий интерфейс с драйверами устройств ввода-вывода;
- ❑ драйверы для некоторых устройств;
- ❑ управление доступом к памяти, включая кэширование, буферизацию и спулинг, а также управление памятью компонентов ввода-вывода в целом.

Буферизация — выделение области памяти, в которую одно устройство (процесс) записывает данные и из которой другое устройство считывает эти данные.

Кэширование — заблаговременная загрузка данных из медленной памяти в быстродействующую память для быстрого к ним доступа.

Спулинг — особый вид буферизации, применяемый для согласования быстродействующих устройств (процессов) с медленными.

Смысл буферизации — избавление от взаимозависимости двух устройств (процессов) при вводе и выводе данных.

При спулинге быстродействующее устройство (процесс) записывает данные в буфер (спул) и продолжает работу, не ожидая, когда они будут считаны медленным устройством (процессом).

Управление дисковой памятью

Основным назначением центрального процессора является выполнение программы. Во время выполнения программа и связанные с ней данные располагаются в основной памяти компьютера. Программа должна храниться где-то в моменты, когда она не выполняется. Данные программы так же должны быть сохранены. Для постоянного хранения программ и данных используется память, которая в подавляющем большинстве современных компьютеров представлена дисковыми устройствами. Кроме того, дисковые устройства могут быть использованы операционной системой с целью имитации расширения основной памяти путем свопинга (выгрузки неактивных фрагментов основной памяти на диск и замены их программами, нуждающимися в срочном выполнении). Таким образом, блок управления дисковой памятью должен обеспечивать следующую функциональность:

- управление свободным пространством на диске;
- выделение дисковой памяти для хранения программ и данных;
- планирование использования дисковой памяти.

Управление сетевыми соединениями

Практически все современные операционные системы имеют в своем составе функциональный блок, отвечающий за работу с сетью. В случае, если это распределенные системы, этот блок является обязательным, в случае однопользовательских операционных систем он может присутствовать в качестве дополнительного сервиса системы.

Обеспечение безопасности

В современной многопользовательской операционной системе механизм защиты ресурсов системы от неавторизованного доступа является одним из основных функциональных блоков. Этот механизм обеспечивает запрет или разрешение на использование тех или иных ресурсов системы в зависимости от прав, которые выделены пользователю.

Интерпретатор команд

Операционная система нуждается в средствах общения с пользователем (оператором). Как уже отмечалось, общее название этого средства — *оболочка*. В качестве оболочки может выступать программа с интерфейсом командной строки (командный интерпретатор MS-DOS, UNIX или Linux) или оконная оболочка (так называемый дружественный пользователю интерфейс), позволяющая манипулировать графическими объектами (окнами) на экране при помощи мыши. Оконную оболочку на сегодняшний день имеют практически все известные операционные системы, предназначенные для использования в персональных компьютерах.

16.4.2. Системные службы операционной системы

В предыдущем разделе мы рассмотрели операционную систему как набор функциональных блоков. Однако если рассматривать ОС как среду, предоставляющую пользовательской программе определенные услуги и обеспечивающую комфортное пребывание в ней пользователей и их прикладных программ, то надо вести речь о том, какие сервисы должна реализовывать операционная система.

- *Выполнение программ.* Операционная система должна уметь загружать программу в основную память и запускать ее. Программе должна быть предоставлена возможность завершить работу нормально или же с ошибкой в случае внутреннего сбоя.
- *Операции ввода-вывода.* Когда прикладная программа запущена, у нее может возникнуть потребность в осуществлении операций ввода-вывода как со стандартных системных устройств (дисплей и клавиатура), так и с дополнительных устройств хранения, передачи или обработки данных. Операционная система

- должна предоставлять прикладной программе возможность осуществлять эти операции на основе некоторых стандартизированных соглашений (протоколов).
- *Взаимодействие с файловой системой.* Операционная система должна предоставлять возможность программе осуществлять операции с файлами и каталогами; программа должна получить механизм для создания, записи, чтения, поиска, удаления или изменения файлов.
 - *Межпрограммное взаимодействие.* Программы часто пуждаются во взаимодействии друг с другом в рамках одного компьютера или между компьютерами. Операционная система должна дать возможность программам передавать друг другу информацию через посылку сообщений, разделяемые области памяти или системные каналы.
 - *Обработка ошибок.* Работающим программам, как и создающим их людям, свойственно время от времени допускать ошибки. Операционная система должна уметь обнаруживать эти ошибки и правильно их обрабатывать (не допуская потери устойчивости, блокирования каналов ввода-вывода или других коммуникационных каналов, бесполезного расходования областей основной памяти ОС). Для каждого типа возможных ошибок в операционной системе должны быть предусмотрены действия, обеспечивающие изоляцию ошибки и продолжение дальнейшей работы операционной системы в нормальном устойчивом режиме.
 - *Распределение ресурсов.* Поскольку современные операционные системы являются многопользовательскими и многозадачными, операционная система должна уметь распределять ресурсы системы между несколькими пользователями и заданиями с наибольшей эффективностью.
 - *Обеспечение безопасности.* В многопользовательских системах прикладные программы и данные одного пользователя должны быть полностью и надежно изолированы от возможности доступа со стороны программ и данных другого пользователя. Механизм этой изоляции предоставляется операционной системой.

16.4.3. Представление операционной системы в виде слоев

С точки зрения проектирования, разработки и реализации операционных систем часто оказывается эффективным рассматривать операционную систему как ряд слоев программного обеспечения, располагающихся между компьютером и пользователем. Такой подход к структуре операционной системы имеет свои преимущества: поскольку слои разрабатываются последовательно, то функциональность каждого слоя полностью базируется на функциональности нижележащих слоев. Первый слой разрабатывается и воплощается в жизнь, то есть достигается полная его работоспособность. После этого начинается создание нового слоя, и любые ошибки или сбои, которые выявляются при его создании, легко локализуются и обнаруживаются внутри создаваемого слоя. Кроме того, каждый последующий слой создавать проще, чем предыдущий, поскольку каждый предыдущий слой скрывает в себе множество деталей, о которых совершенно не нужно знать вышележащему слою. В качестве примера такого послойного представления операционной систе-

мы можно привести первую операционную систему THE (Technische Hogeschool Eindhoven), построенную на принципе разделения на слои (рис. 16.7).

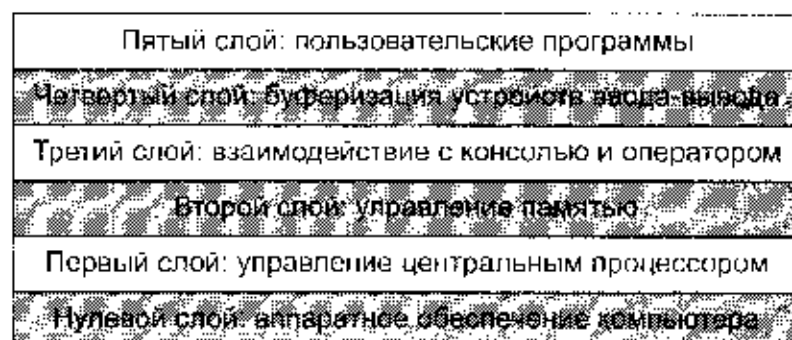


Рис. 16.7. Разделение на слои в операционной системе THE

16.5. Операционные системы семейства Windows

Операционные системы семейства Windows являются на сегодняшний день признанными лидерами среди операционных систем для персональных компьютеров. Несмотря на постоянную конкурентную борьбу и претензии, нельзя забывать, что начало эпохи массового персонального компьютера неразрывно связано с Windows, и операционные системы этого семейства на сегодняшний день установлены на большинстве персональных компьютеров.

16.5.1. Подсемейство Windows 3x

Операционные системы из ветки Windows 3x по сей день вызывают споры о том, были ли они на самом деле полноценными операционными системами или же это были просто графические надстройки над MS-DOS. Этому подсемейству были свойственны следующие недостатки:

- ☐ работа на основе MS-DOS;
- ☐ кооперативная многозадачность;
- ☐ неполное использование возможностей современных им процессоров.

16.5.2. Подсемейство Windows 9x

Операционная система Windows 95 стала известна пользователю задолго до того, как была выпущена. Корпорация Microsoft обеспечила хорошую рекламу своему флагманскому проекту, и эту ОС ждали с нетерпением. А самое главное, что ожидания не просто оправдались, а были даже превзойдены. Теперь, оглянувшись назад, можно сказать, что именно с версии Windows 95 корпорация Microsoft начала свое победоносное шествие в качестве производителя операционных систем для персональных компьютеров. Каждая последующая система несла с собой значительные и приятные для пользователя улучшения: Windows OSR2 — файловая

система FAT32 с поддержкой больших разделов; Windows 98 – новая система драйверов WDM, позволяющая использовать в операционных системах для персональных компьютеров драйверы, разработанные для серверных моделей Windows; Windows Me – улучшенная поддержка мультимедиа- и интернет-приложений. Однако у подсемейства Windows 9x существовало несколько серьезных недостатков:

- невозможность работы в многопользовательском режиме (поскольку отсутствует поддержка учетных записей);
- недостаточно эффективная модель управления памятью;
- отсутствие поддержки квотирования в файловой системе.

16.5.3. Подсемейство Windows NT/2000/XP/2003

Будучи несомненным лидером в среде операционных систем для персональных компьютеров, корпорация Microsoft не собиралась ограничиваться этой нишей. В 1988 году была начата длительная эпопея, закончившаяся в 1994 году выходом Windows NT 3.1. В отличие от ОС для персональных компьютеров ветки 9x, в Windows NT изначально были реализованы микроядерная архитектура, вытесняющая многозадачность и многопоточность, поддержка учетных записей, терминальный режим работы, файловая система с обеспечением квотирования и прав доступа, возможность работы с различными типами процессоров. В общем, это была операционная система, по функциональным возможностям готовая вступить в конкурентную борьбу с UNIX в сфере серверных решений. Разнообразная функциональность Windows NT 3.1, которая впоследствии была признана избыточной, была призвана обеспечить совместимость с другими программными платформами. В последующих версиях (NT 3.51 и NT 4.0) корпорация Microsoft последовательно избавлялась от всего лишнего, делая эту ОС все более специализированной. В версии Windows NT 4 в ядро операционной системы была встроена графическая подсистема от Windows 95, что обеспечило одинаковый графический пользовательский интерфейс для всех операционных систем Windows и облегчило задачу перехода от одной версии к другой: пользователь везде оказывался в знакомой ему оконной оболочке.

Крайне важным не только для завоевания серверной платформы, но и для сохранения позиций среди операционных систем для персональных компьютеров был перевод персональных и серверных версий ОС на одну основу. Так, системы Windows 2000 Advanced Server (ОС для серверов средней и большой мощности), Windows 2000 Server (ОС для серверов средней и малой мощности) и Windows 2000 Professional (ОС для рабочих станций и персонального использования) – это одинаковые операционные системы (с одинаковым ядром), отличающиеся в разных исполнениях только набором сервисных программ и параметрами.

Следующим шагом был пакет из систем Windows 2003 Server и Windows XP разных исполнений, которые также представляли собой одинаковые операционные системы (с одним и тем же ядром).

Операционная система Windows Vista стала одной из неудач корпорации Microsoft. По разным причинам пользователи плохо восприняли эту операционную

систему. Несколько популярнее оказалась последняя разработка – Windows 7. Тем не менее на сегодняшний день среди всех пользователей Windows число работающих на Windows XP – 63 %. Именно по этой причине в качестве типичного представителя операционных систем для персональных компьютеров от Microsoft мы выбрали Windows XP, а не более «свежую» версию.

16.6. Операционная система Windows XP

16.6.1. Графический пользовательский интерфейс

Пользовательский интерфейс в Windows XP построен на основе стандарта для графического оконного интерфейса Windows: рабочий стол с панелью задач и кнопкой Пуск, стандартизированные окна программ. Окна стандартно оснащены тремя кнопками с правой верхней стороны и одним системным меню слева вверху. Оформление окон и рабочего стола может в небольших пределах настраиваться. При выключении всех визуальных эффектов оформление окон полностью соответствует таковому в Windows 95, если же все визуальные эффекты и эффекты оформления включить, то окна соответствуют дизайну Windows XP, а различные элементы управления приобретают дополнительные функции: меню начинают отбрасывать тень, ниспадать или разворачиваться, тень появляется у указателя мыши, и т. д. Эти эффекты серьезно потребляют ресурсы компьютера, поэтому в инструментах настройки даже существует выбор: настроить для лучшей производительности (без «украшательства») или для большей красоты. На рис. 16.8 показано одно и то же окно Windows XP при этих двух вариантах настройки.

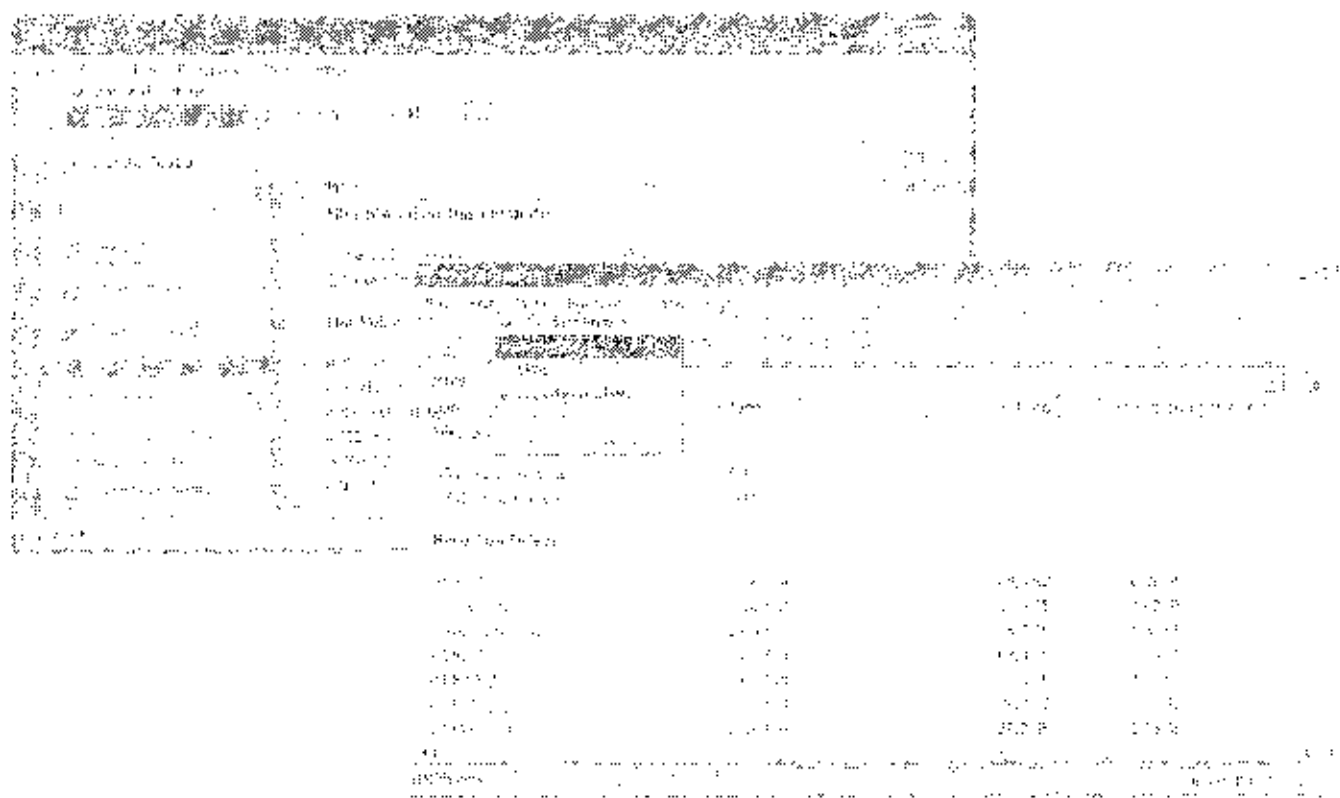


Рис. 16.8. Два варианта настройки окна

Благодаря тому, что, начиная с Windows 95, графический пользовательский интерфейс практически не менялся, пользователь каждый раз оказывался в знакомой ему среде и мог сосредоточить основные усилия по освоению новой операционной системы на новой функциональности.

16.6.2. Файловые системы

Windows XP поддерживает два типа файловых систем: FAT32 и NTFS. С точки зрения пользователя работа с этими файловыми системами различается мало: все операции с файлами и каталогами проводятся одним и тем же образом — при помощи обозревателя. При этом в обозревателе можно выполнять операции как при помощи команд контекстного меню, так путем перетаскивания. Можно запустить командный интерпретатор и выполнять операции с файловой системой из командной строки. Некоторые операции с файлами в Windows XP можно выполнять только из командной строки, например, операцию преобразования раздела NTFS в раздел FAT32. С точки зрения администрирования и защищенности многопользовательской системы, безусловно, предпочтительнее файловая система NTFS, в которой можно разграничивать права на доступ к файлам и папкам. Когда доступ к файлам и папкам разграничен, попытка открыть файл или папку, на которую у пользователя нет соответствующих прав, блокируется (рис. 16.9).

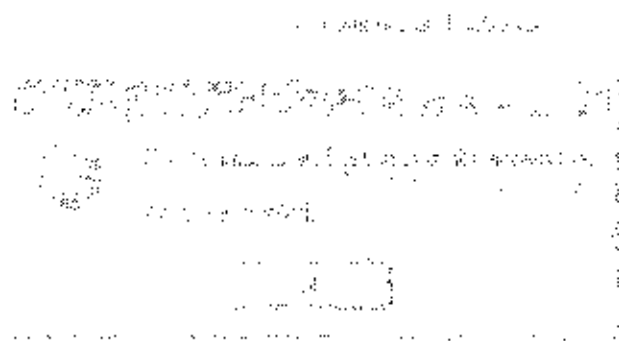


Рис. 16.9. Запрет на доступ к папке при использовании NTFS

В отличие от файловых систем семейства UNIX, в Windows принято представлять дисковые устройства как часть файловой системы.

16.6.3. Работа с локальной сетью

Все сетевые соединения в Windows XP производятся при помощи подключений. Подключение может быть как проводным (через сетевую карту в случае проводной сети), так и беспроводным. Доступ ко всем сетевым подключениям можно получить через соответствующий пункт меню. Там же находятся мастера, позволяющие настроить новое сетевое подключение или же локальную сеть той или иной конфигурации (рис. 16.10).

После того как сетевое подключение (одно или несколько) настроено, а локальная сеть должным образом сконфигурирована, доступ к локальной сети (вне

зависимости от того, по какому протоколу производится обмен данными) осуществляется единообразно — через папку My Network Places (рис. 16.11).

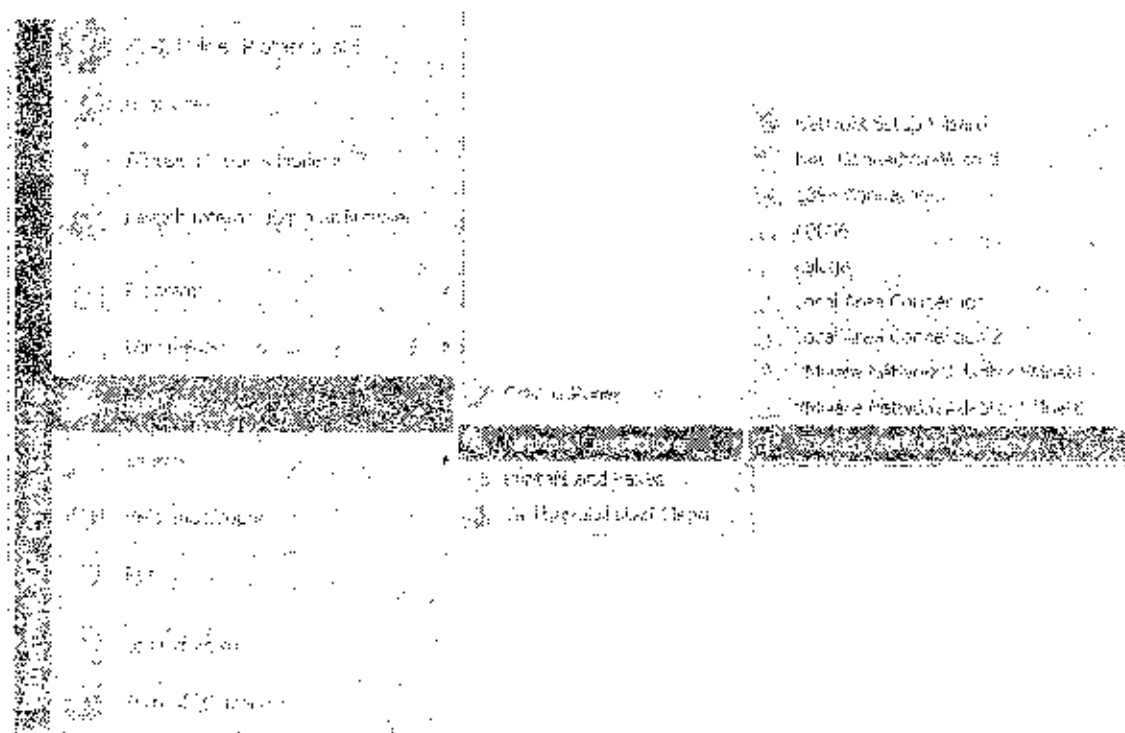


Рис. 16.10. Сетевые подключения

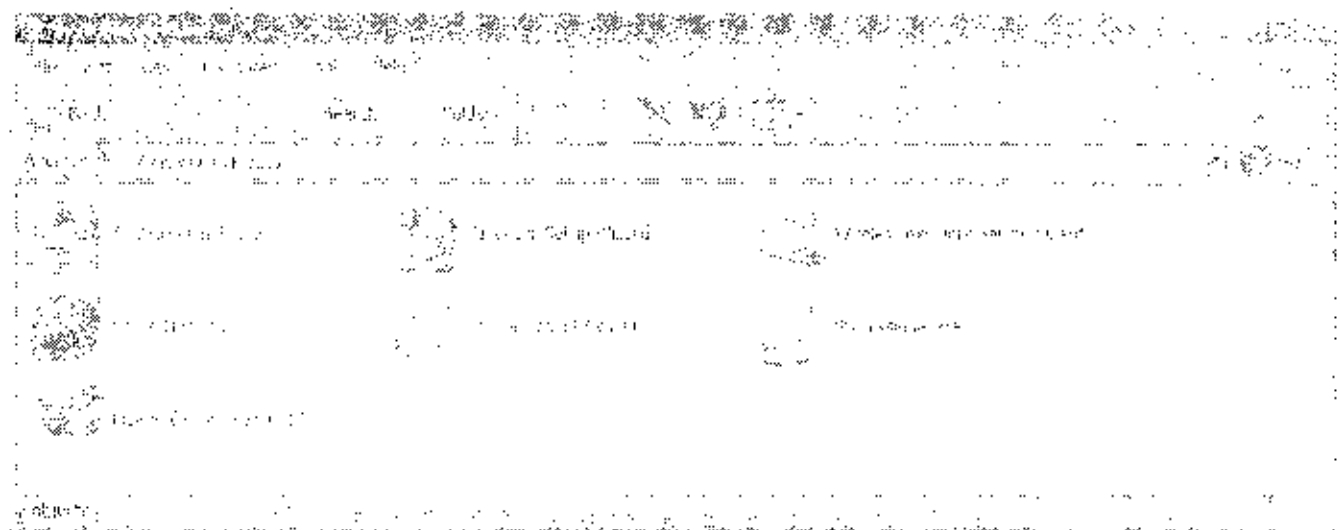


Рис. 16.11. Доступ к объектам локальной сети

Внутри этой папки работа с данными (файлами и каталогами) происходит так же, как и с локальными данными: в обозревателе при помощи команд контекстного меню или путем перетаскивания. Возможно и прямое открытие (и редактирование, если это разрешено правами доступа к папке) документов из сетевых папок.

К несомненному достоинству версий Windows, основанных на ядре NT, нужно отнести отсутствие необходимости перегружать операционную систему после внесения изменений в настройку сети (в Windows 9x иногда параметры сети приходилось менять довольно часто).

16.6.4. Работа с Интернетом

Доступ к сайтам Интернета осуществляется при помощи программы Internet Explorer (рис. 16.12). На самом деле Internet Explorer и обозреватель Windows это один и тот же многофункциональный инструмент. Чтобы убедиться в этом, достаточно скопировать адрес из адресной строки Internet Explorer в адресную строку любого окна Windows XP и нажать клавишу Enter.

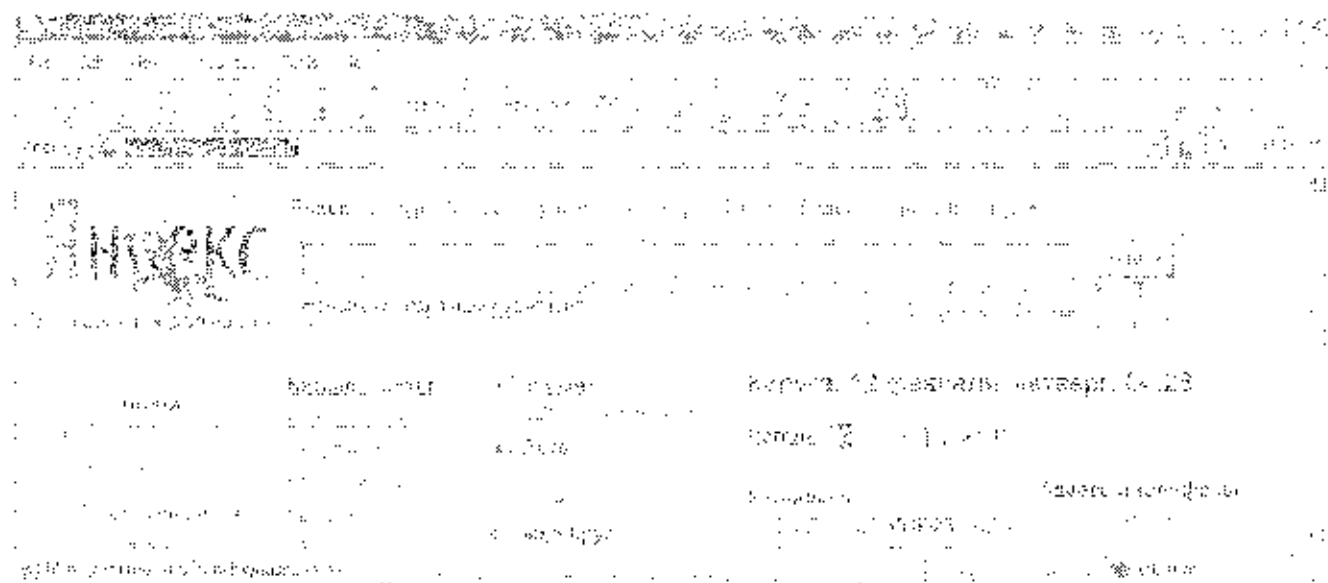


Рис. 16.12. Окно Internet Explorer

Internet Explorer представляет собой довольно сложный инструмент, позволяющий достаточно эффективно настроить защиту и уровни безопасности, блокировать опасное содержимое при посещении ненадежных сайтов и вообще оградить компьютер от вредоносных воздействий. Делается это путем настройки параметров Internet Explorer (рис. 16.13).

К сожалению, задание этих параметров требует не только вдумчивого подхода и чтения документации, но еще и постоянного внимания при навигации по Интернету: настроишь слишком высокий уровень безопасности — оказывается отключенной функциональность на нужных сайтах, перестраиваешь на более низкий — нужные сайты начинают работать, зато с других проскакивают вредоносные программы. То есть в индивидуальном подходе нуждается каждый посещаемый сайт, и трудно себе представить обычного пользователя, который при путешествиях по интернет-страницам будет постоянно заниматься настройкой уровня безопасности браузера. По этой причине, а также по причинам, которые мы обсудим позже, Internet Explorer является удобным, но крайне небезопасным средством навигации в Интернете.

Чтение электронной почты и групи новостей в Windows XP осуществляется при помощи программы Outlook Express (рис. 16.14).

Программа крайне проста в обращении, но ей очень не хватает функций персонального планирования, которые есть у ее «старшего брата» Outlook из пакета Microsoft Office. К сожалению, этот пакет не входит в состав операционной системы Windows XP.

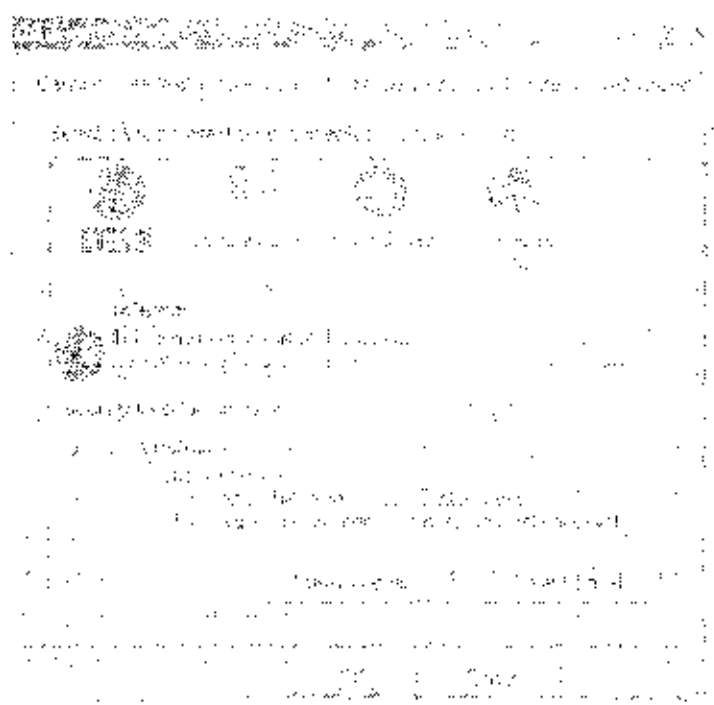


Рис. 16.13. Настройка уровней безопасности в Internet Explorer

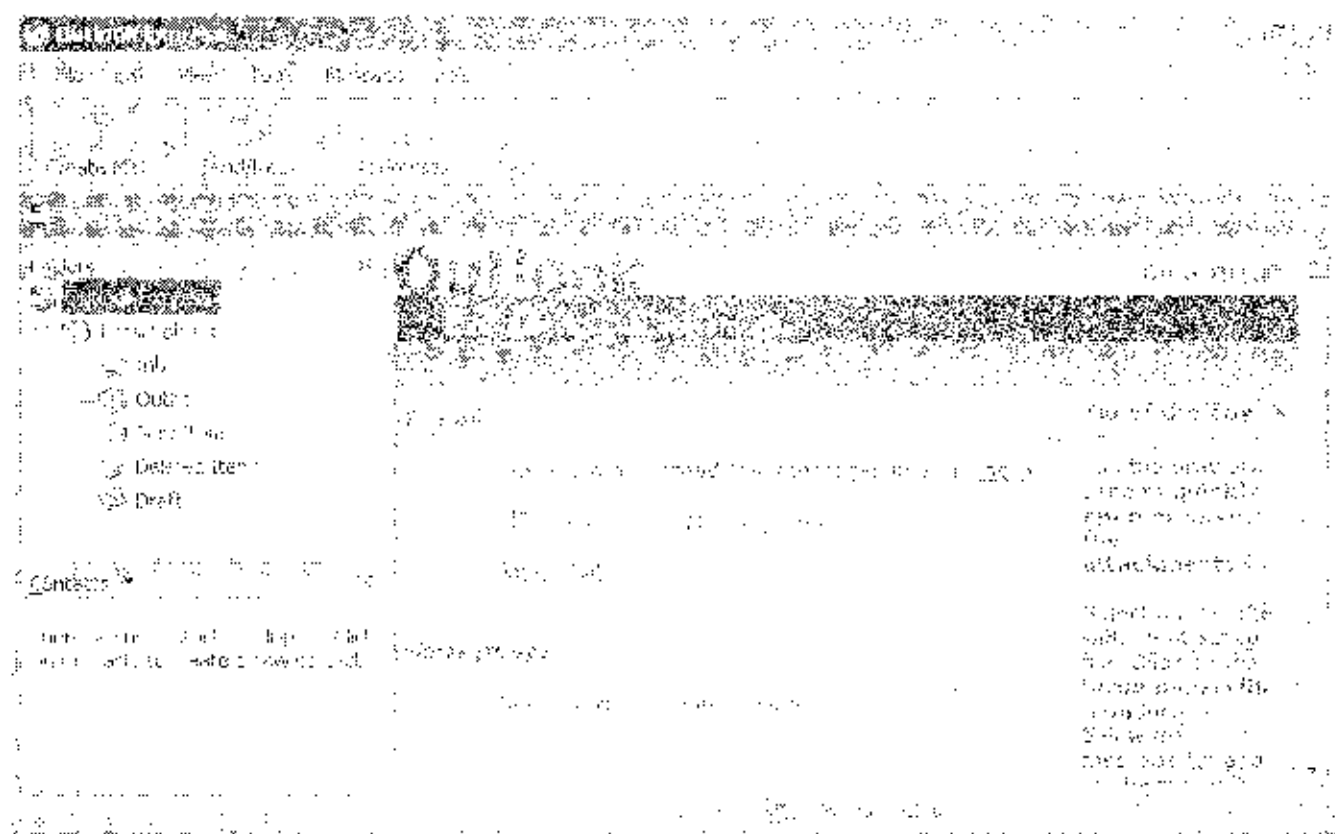


Рис. 16.14. Окно программы Outlook Express

16.6.5. Мультимедийные возможности

Воспроизведением различных видео- и аудиоформатов в Windows XP занимается программа Windows Media Player (рис. 16.15).

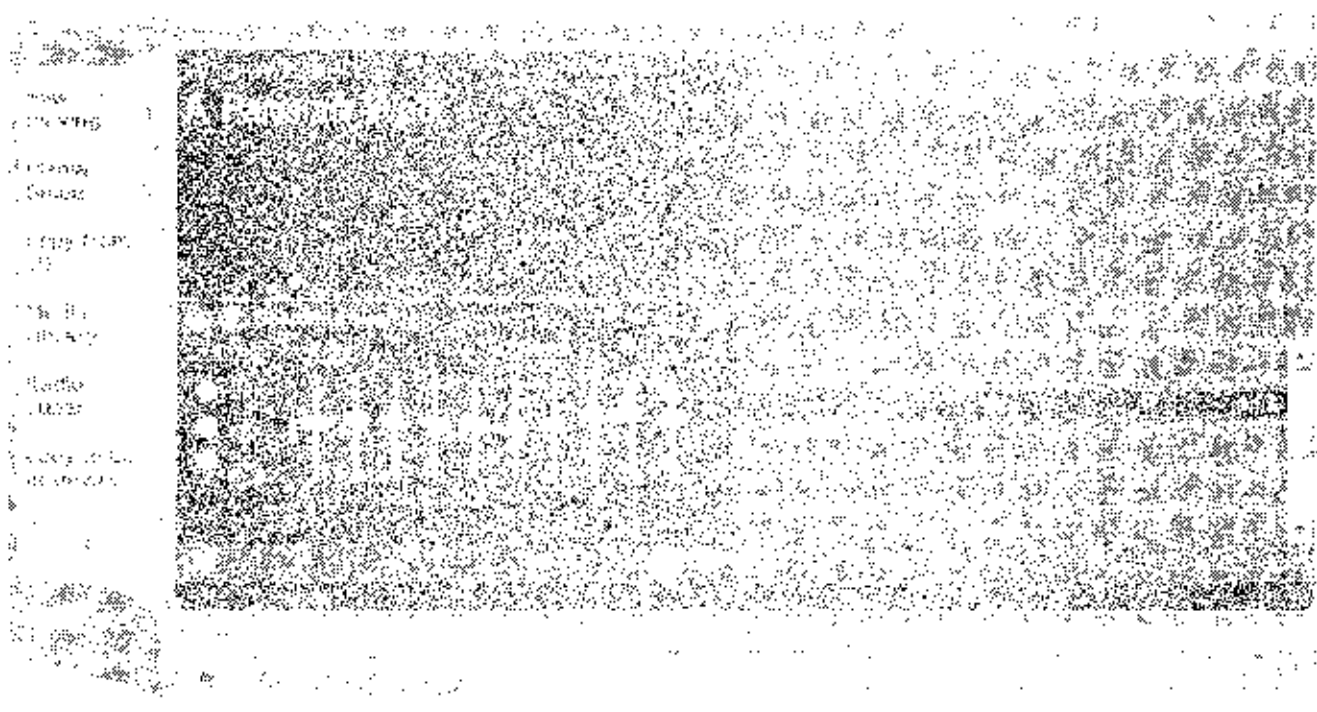


Рис. 16.15. Windows Media Player

Программа позволяет воспроизводить аудио- и видео-файлы в распространенных форматах (MP3, OGG, WAV, MIDI, AVI, WMA, MPEG), проигрывать CD-диски, копировать аудиотреки с CD-дисков в локальную библиотеку, записывать CD-диски, поддерживать потоковое вещание (как аудио, так и видео). Однако воспроизвести в этой программе видео-DVD не получится, а значит, в мультимедийной функциональности Windows XP наблюдается довольно большой недостаток. Кроме того, перечень поддерживаемых форматов аудио и видео не полон, поэтому довольно часто вы будете получать сообщение о том, что проигрыватель не может воспроизвести файл предложенного формата (например, MP4 или RA).

16.6.6. Обеспечение безопасности

Поскольку операционная система Windows XP построена на ядре серверной операционной системы, в ней присутствует развитая система настройки и обеспечения безопасности. В первую очередь, настройка параметров безопасности производится при помощи инструмента настройки локальных политик безопасности, который можно найти в группе Администрирование панели управления (рис. 16.16).

Помимо этого инструмента, существует еще инструмент управления групповыми политиками безопасности (`gpedit.msc`), но на этот инструмент нет ссылки в панели управления, его можно вызывать только из командной строки. Если же речь идет о сетевых подключениях и обеспечении безопасности приложений, работаю-

щих с сетью, то тут в Windows XP есть прекрасный инструмент — персональный брандмауэр (рис. 16.17).

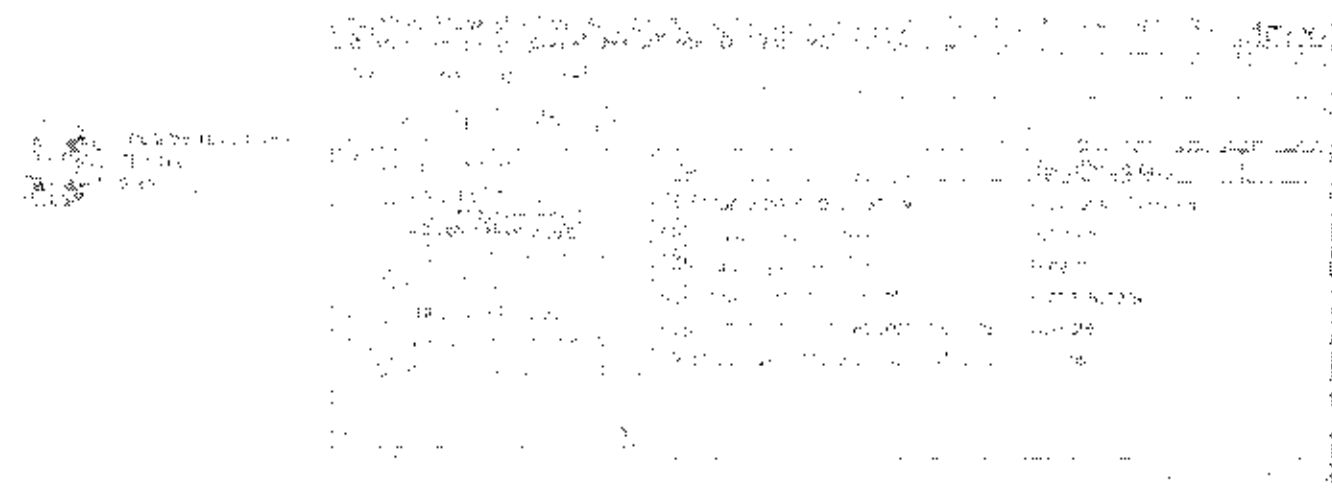


Рис. 16.16. Инструмент настройки локальной политики безопасности

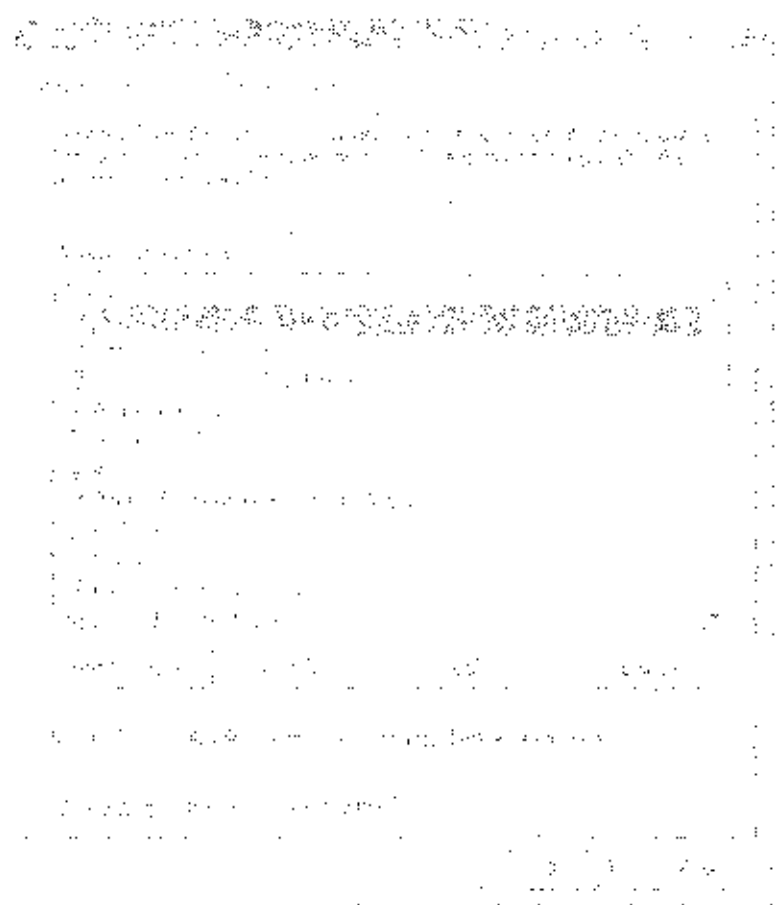


Рис. 16.17. Персональный брандмауэр Windows

При помощи брандмауэра любому приложению можно запретить или разрешить как сетевые соединения вообще, так и общение в сети по выбранному протоколу или с заданными параметрами. Конечно, как и другие инструменты безопасности, этот инструмент требует вдумчивости при настройке.

Кроме этих инструментов, в Windows XP существует еще три службы, прямо или косвенно связанные с обеспечением безопасности и целостности системы: *Центр управления безопасностью* поднимает тревогу, если вы отключили брандмауэр или не установили антивирусное ПО, *Служба обновлений* автоматически загружает с сайта Microsoft пакеты обновлений, закрывающие найденные в процессе эксплуатации уязвимые места в системе, *Служба восстановления* позволяет вернуть систему к состоянию, предшествующему установке вредоносного ПО (пользовательские документы с ее помощью восстановить невозможно).

Таким образом, Windows XP с точки зрения безопасности представляет собой хорошо защищенную операционную систему, правда, при условии, что пользователь хорошо обучен и соблюдает правила безопасности.

16.6.7. Средства коллективной работы

Для взаимодействия пользователей в локальной сети и Интернете в Windows XP включена программа NetMeeting. Это приложение позволяет осуществлять самые разные виды коллективного взаимодействия как между двумя пользователями, так и в групповом режиме: обмен текстовыми сообщениями, проведение аудио- и видеоконференций, совместное рисование на общей доске, трансляция рабочего стола на экраны других пользователей, передача доступа к своему рабочему столу другим пользователям и многое другое (рис. 16.18).



Рис. 16.18. Основное окно NetMeeting

Данное приложение может быть эффективно использовано как для совместной работы в распределенных рабочих группах, так и для организации учебного процесса внутри одного класса, физического или виртуального.

16.6.8. Средства настройки и администрирования

Все средства настройки Windows XP сосредоточены в одном месте — в панели управления. Система имеет достаточно простые инструменты с интуитивно понятным пользовательским интерфейсом, позволяющим настраивать систему даже неопытным пользователям. Хорошо организованная справочная система содержит в себе ответы практически на все возникающие в процессе настройки вопросы. Множество средств настройки представляют собой мастера, проводящие шаг за шагом через все этапы задания необходимых параметров. В операционной системе Windows XP средства настройки практически не отделены от средств администрирования. Несмотря на то, что внутри панели управления находится группа Администрирование, эта группа отличается от остальных инструментов настройки не столько функционально, сколько уровнем сложности: для того чтобы использовать инструменты этой группы, необходимо обладать некоторыми знаниями и навыками, более глубокими, чем навыки простого пользователя.

16.6.9. Приложения из состава ОС

Хотя вместе с Windows XP поставляется ряд полезных приложений (все они расположены в группе Стандартные), это скорее вспомогательные приложения, чем полноценные инструменты для работы. В Windows XP подразумевается, что вместе с операционной системой поставляются полнофункциональные инструменты для работы в Интернете (Internet Explorer и Outlook Express), для выполнения же других функций, включая работу с документами, черчение, сложную обработку изображений или выполнение математических расчетов, приложения должны приобретаться и устанавливаться отдельно.

16.7. Операционная система Alt Linux

16.7.1. История появления свободной операционной системы Linux

Корни современных операционных систем находятся в операционной системе MULTICS, которая в дальнейшем переродилась в UNIX. Некоторое время эта операционная система обладала открытыми исходными кодами. Было разработано множество коммерческих версий UNIX, и в конце копцев исходные коды современных систем UNIX в подавляющем большинстве были закрыты. Поскольку множество уникальных разработок UNIX привело к почти полной потере совместимости прикладных программ для этих систем не только в бинарном виде, но и на уровне исходных кодов, примерно в этот же период была разработан стандарт POSIX.

В стандарте POSIX описываются минимальные требования к системным вызовам систем UNIX, обеспечивающие совместимость на уровне исходного кода, то есть программа, разработанная на одном типе UNIX, должна без проблем компилироваться и запускаться на другом.

В связи с тем, что исходные коды UNIX стали закрытыми, в университетах США и Европы запретили использование исходных кодов UNIX в учебном процессе. А поскольку глубокое изучение операционных систем без исходных кодов практически невозможно, Эндрю Таненбаум, один из известнейших авторов и преподавателей в области операционных систем, разработал собственную операционную систему MINIX. Эта система была POSIX-совместимой, но весь исходный код в ней был написан «с нуля». Поскольку операционная система создавалась в учебных целях, исходный код ее был открытым и она приобрела большую популярность среди пользователей персональных компьютеров.

Одним из пользователей этой операционной системы был Линус Торвалдс. Он добавил одно усовершенствование, второе... написал собственную файловую систему, а на определенном этапе полностью переписал код ядра. Практически в этот момент родилась новая операционная система — Linux.

В настоящий момент Linux — это многозадачная многопользовательская и свободная операционная система, которая конкурирует с коммерческими ОС практически по всему фронту их применения. Достаточно сказать, что первая пятерка в списке ста мощнейших суперкомпьютеров мира работает под управлением Linux. В сфере серверных операционных систем Linux непрерывно расширяет свое присутствие (до 50 % серверов в Интернете работает под управлением Linux). Linux в качестве настольной операционной системы не только все более популярна, но и в ряде стран официально выбрана в качестве образовательной.

Как представителя семейства операционных систем Linux мы рассмотрим дистрибутив AltLinux, который стал официальным дистрибутивом российского школьного образования.

16.7.2. Графический пользовательский интерфейс

Человек, не знакомый с Linux, но слышавший об этой ОС, ожидает при первом знакомстве столкнуться с трудностями, вроде командной строки или необычайно скудного пользовательского интерфейса. Однако в действительности он попадет в очень знакомую обстановку оконного графического интерфейса с кнопкой главного меню, расположенной в том же месте, что и в Windows, с такими же окнами, которые ведут себя очень похожим образом, с теми же элементами управления, панелью задач и рабочим столом со значками на нем (рис. 16.19).

Элементы рабочего стола ведут себя тоже вполне предсказуемо: при выборе пункта меню или двойном щелчке на значке рабочего стола запускается приложение, значок приложения отображается на панели задач. Что же касается настройки внешнего вида во всех его деталях, то в этом AltLinux дает пользователю очень большие возможности. То, как может выглядеть на экране одно и то же окно, показано на рис. 16.20.

Оконный менеджер KDE, на базе которого построена графическая среда в AltLinux, дает возможности настройки внешнего вида, цветов, стилей и декораций окон, а также визуальных эффектов в гораздо более широком диапазоне, чем в Windows XP. KDE позволяет организовать несколько виртуальных рабочих

Работа с файлами и каталогами производится при помощи файлового менеджера Konqueror, который в определенной степени можно назвать аналогом обозревателя в Windows XP. Все операции с файлами и каталогами выполняются при помощи команд контекстного меню или перетаскиванием мышью. Существует также интересная возможность предварительного просмотра содержимого во всплывающем эскизе (рис. 16.21).

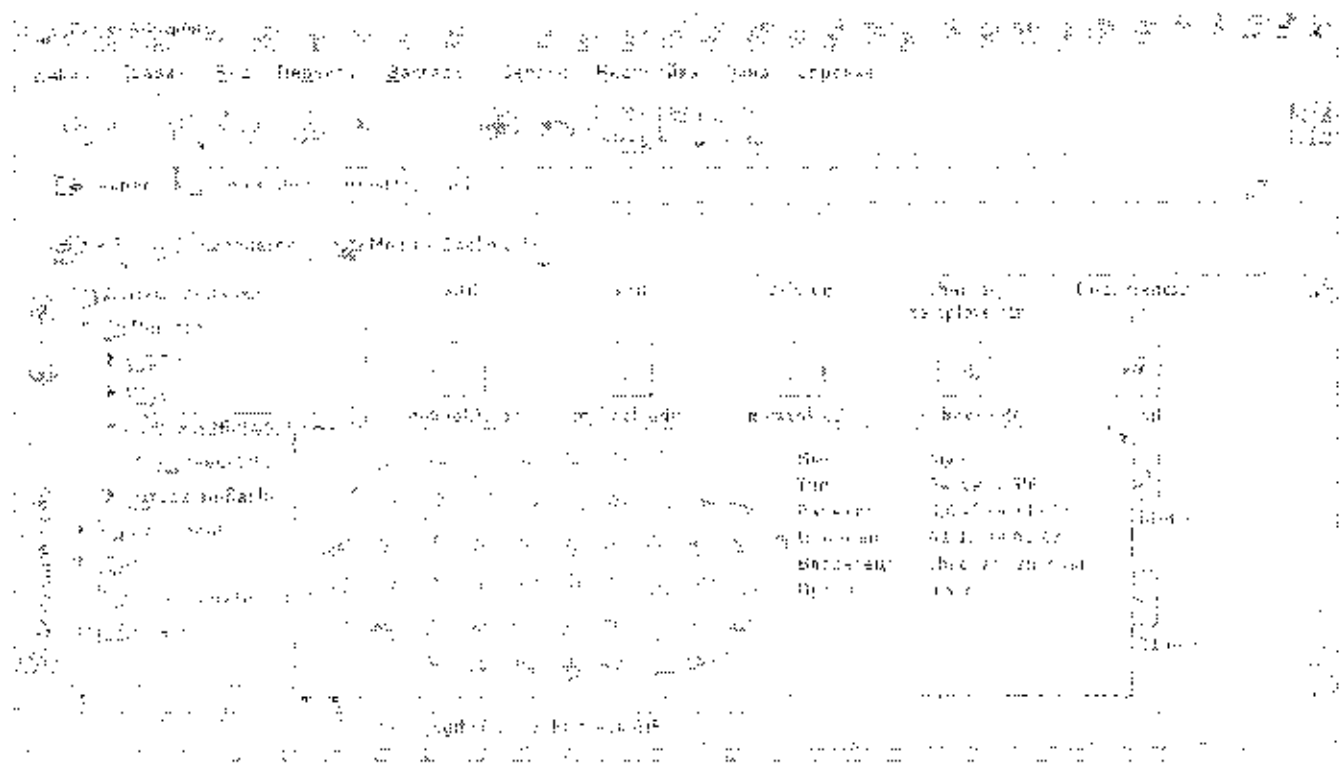


Рис. 16.21. Окно файлового менеджера

16.7.3. Файловые системы

AltLinux работает с большим количеством типов файловых систем. Часть из них являются «родными» для Linux, например, Ext2/3, ReiserFS, XFS, JFS. Все эти файловые системы являются протоколируемыми: то есть если при работе с файловой системой происходит, к примеру, отключение питания, то после включения файловая система автоматически вернется в исходное состояние. При этом AltLinux корректно работает на запись и чтение с файловыми системами Windows, включая FAT12, FAT16, FAT32 и NTFS.

Все файловые системы Linux поддерживают концепцию прав и разграничения доступа, при этом режим разграничения доступа включен по умолчанию. Таким образом, сразу после установки файловая система Linux защищает целостность системы от любых действий обычного пользователя: обычный пользователь (не администратор) может удалить все, что он сумеет удалить, при этом и система, и данные других пользователей останутся неприкосновенными.

Утилиты обслуживания файловых систем в Linux — это утилиты командной строки: проверка и восстановление системы производится в консольном режиме командой `/sbin/fsck`, но внесение изменений в файловую систему требуют административных привилегий.

16.7.4. Работа с сетью

Работа с сетью в AltLinux с точки зрения пользователя выглядит довольно просто: если в системе есть сетевые соединения, они настраиваются автоматически. Большинство современных сетевых устройств, будь то проводные, беспроводные или модемные, опознаются автоматически и настраиваются. Если автоматическая настройка не дала результата, настройку сетевых устройств можно выполнить при помощи специального инструмента настройки с графическим пользовательским интерфейсом (рис. 16.22).

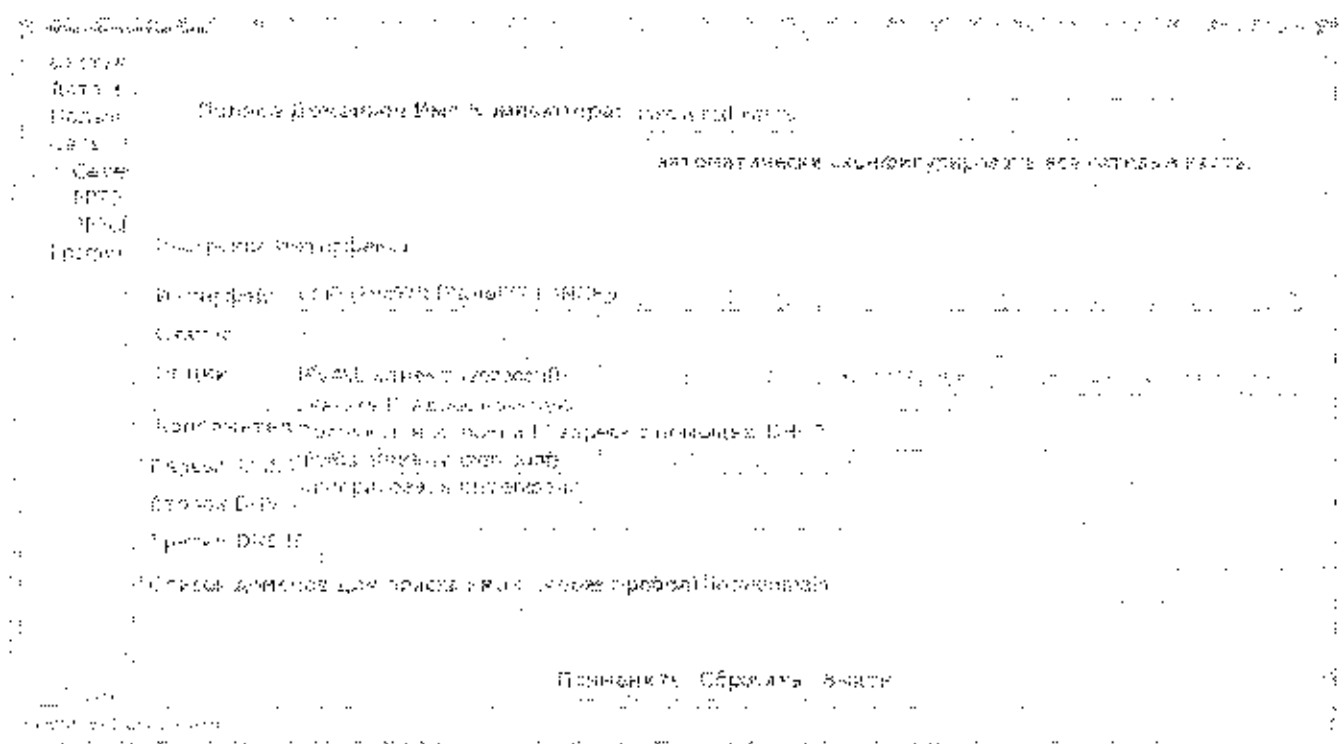


Рис. 16.22. Настройка сетевых соединений

Как и в Windows XP, в AltLinux автоматическая настройка чаще всего выполняется корректно, а настройка соединения вручную требует определенных знаний.

После того как соединения настроены, их функционирование можно отслеживать по значку на панели задач, в частности, можно в графическом виде наблюдать исходящий и входящий трафик (рис. 16.23).

AltLinux позволяет организовывать работу в локальных сетях как по протоколам webDav (организация и совместное использование сетевых папок по протоколу HTTP) и FTP, так и по протоколу SMB (протокол, на основе которого в Windows организуется доступ в локальную сеть). Поскольку все эти протоколы не принадлежат определенной операционной системе, являясь межплатформенными, в AltLinux-совместимой локальной сети свободно могут сосуществовать машины как с Linux, так и с Windows, и может быть организован прозрачный двусторонний доступ с одних машин на другие. С точки зрения пользователя, вся работа с файлами локальной сети (а в случае протоколов FTP и webDav это могут быть и папки, размещенные глобально) организуется при помощи файлового менеджера Konqueror. Рисунок 16.24 иллюстрирует доступ с рабочего стола Linux к разделяемым папкам машины с Windows.

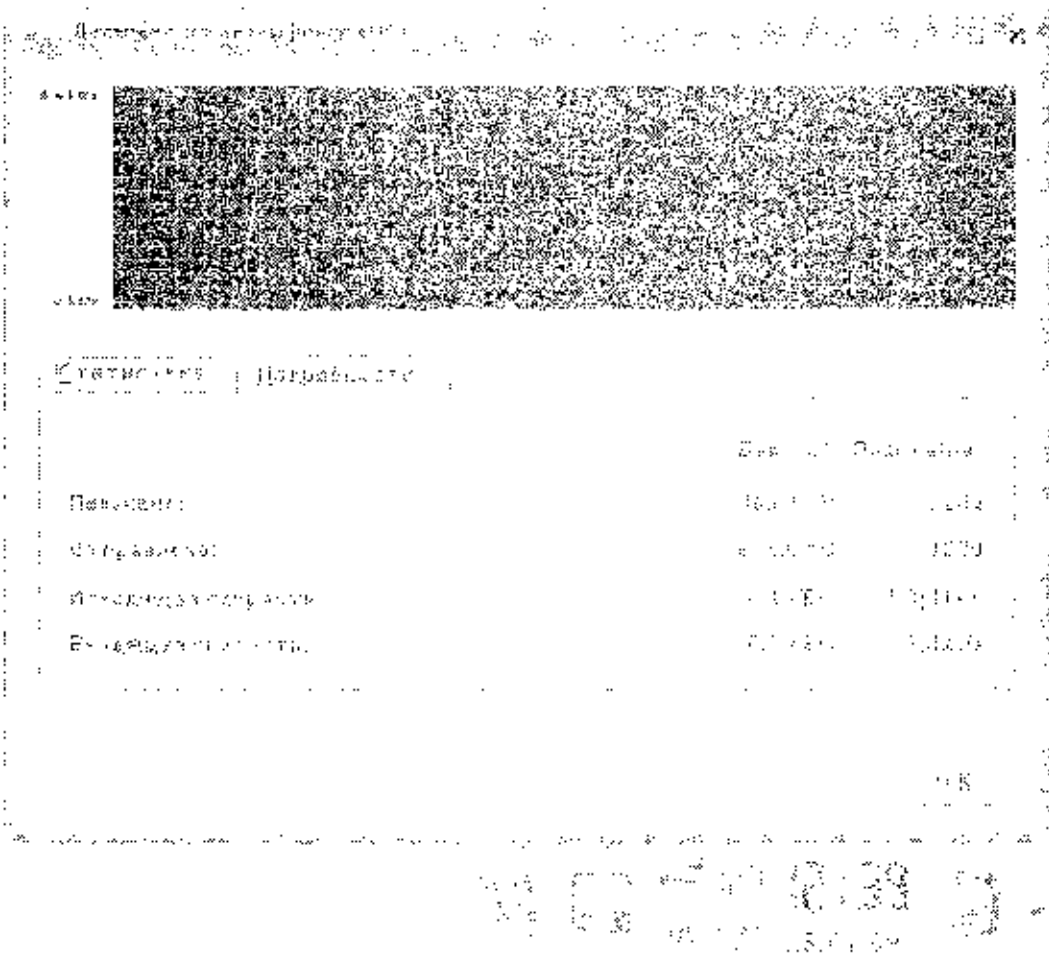


Рис. 16.23. Мониторинг трафика сетевого соединения

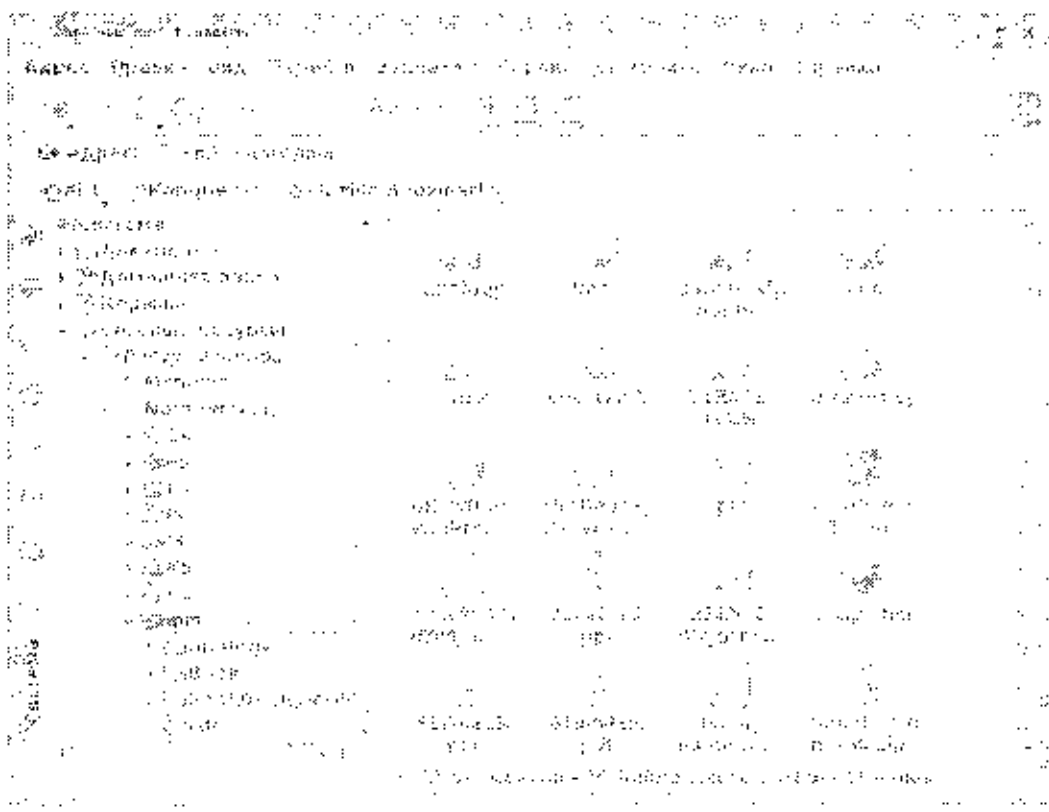


Рис. 16.24. Доступ к сетевым папкам Windows из AltLinux

16.7.5. Работа с Интернетом

AltLinux содержит все необходимые средства работы с Интернетом. Для навигации по веб-пространству можно использовать один из двух браузеров графического режима: Konqueror или Mozilla Firefox. На рис. 16.25 можно видеть оба этих браузера, воспроизводящих один и тот же видеоролик с сайта Youtube.

В обоих браузерах AltLinux поддерживаются все современные технологии Интернета, и оба этих браузера примерно равноценны по функциональности. Однако в Firefox встроена система фильтрации нежелательного контента, весьма удобная функция, позволяющая отключать загрузку рекламных баннеров на часто посещаемых сайтах.

Кроме интернет-браузеров, в AltLinux для работы в Интернете используются следующие приложения:

- ❑ *Mozilla Thunderbird* — клиентская программа для чтения электронной почты, групп новостей, новостных лент и блогов;
- ❑ *Kopete* — многопротокольный клиент для обмена мгновенными сообщениями (поддерживаются практически все известные на сегодняшний день протоколы, включая Jabber и ICQ).

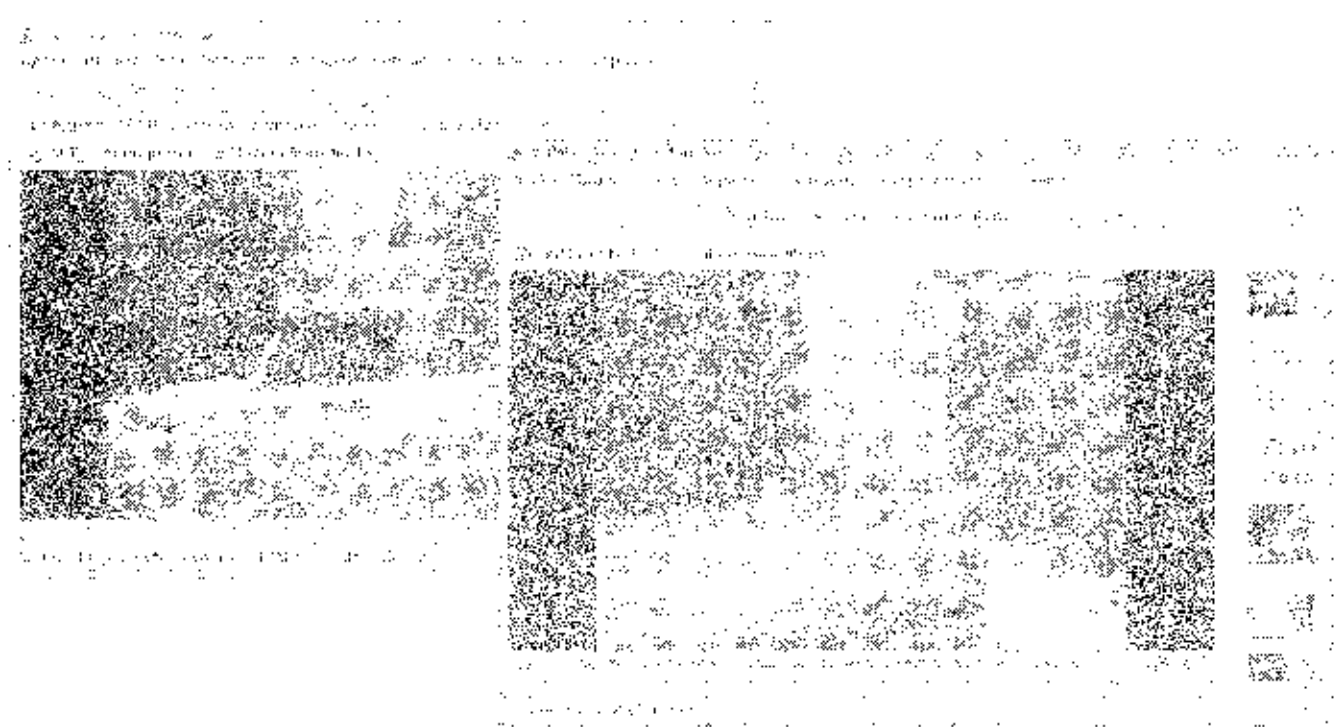


Рис. 16.25. Konqueror и Firefox в Интернете

16.7.6. Мультимедийные возможности

Для воспроизведения мультимедийного контента в AltLinux имеется сразу несколько программ, функциональность которых частично перекрывается. Наиболее широкопрофильным является медиаплеер Kaffeine, позволяющий воспроизводить практически весь диапазон мультимедиа, от звуковых файлов всех известных форматов и аудио-CD до видеофайлов и видео-DVD (рис. 16.26)

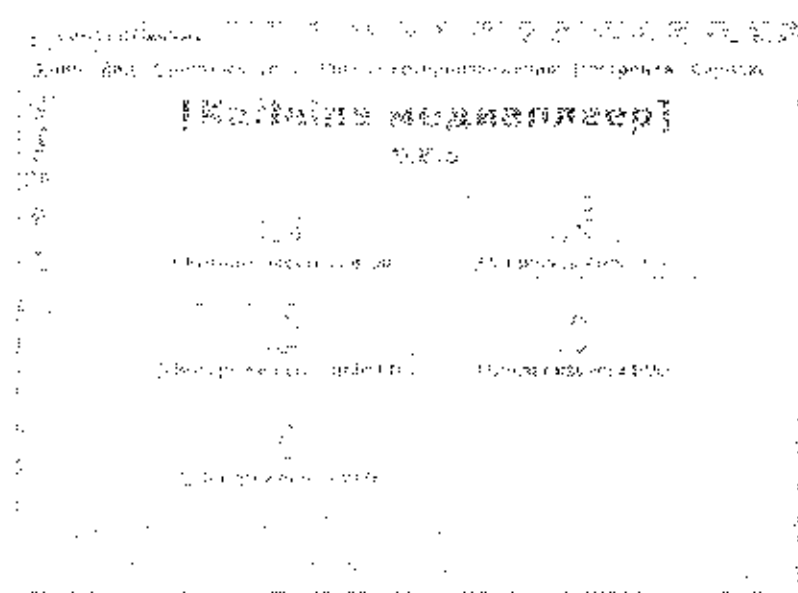


Рис. 16.26. Медиаплеер Kaffeine

Этот же медиаплеер поддерживает потоковое вещание, радио- и телетрансляции из Интернета.

Кроме этого базового средства в состав AltLinux включены программы, позволяющие более точно воспроизводить отдельные виды мультимедиа: проигрыватель аудио-CD (KsCD), проигрыватель аудиофайлов (AmaroK), еще один многофункциональный медиаплеер, с помощью которого можно организовать потоковое вещание (VLC-медиаплеер).

Помимо воспроизведения мультимедиа операционная система AltLinux включает в себя развитые средства создания мультимедиа: инструмент звукозаписи KRec, инструмент редактирования аудиофайлов (с профессиональными возможностями) Audacity, программа записи CD- и DVD-дисков K3B, программа создания и редактирования видео (Kino).

16.7.7. Обеспечение безопасности

Безопасность в AltLinux построена на стандартных средствах обеспечения безопасности UNIX — разделении прав доступа, ограничение прав простых пользователей, изоляция ресурсов учетных записей. При аккуратной работе с этими средствами специальных средств обеспечения безопасности в системах семейств Linux и UNIX не требуется. В системе AltLinux предприняты определенные меры, призванные затруднить злоумышленнику возможность нанесения вреда, например, сервисы, требующие для работы повышенных привилегий, запускаются в среде `chrooted environment` — своеобразной комнате с мягкими стенами, отделяющей их от остальной системы. А вот отдельного инструмента по управлению безопасностью нет. По этой причине, с точки зрения конечного пользователя, для обеспечения безопасности в AltLinux нужно только собственное разумное поведение: не отдавать свой личный пароль в чужие руки. Тем не менее в инструменте настройки оконной среды KDE есть раздел, касающийся безопасности и конфиденциальности. Он позволяет обеспечить спокойствие пользователя путем очистки в конце каждого сеанса всех следов пребывания пользователя в системе, включая историю

Менеджер пакетов Synaptic позволяет осуществлять как установку, так и удаление пакетов в AltLinux (программы в Linux устанавливаются в виде одного пакета или взаимозависимого набора пакетов). Кроме того, в Synaptic встроена функциональность, позволяющая обновлять пакеты системы по мере появления на серверах новых версий, а также выполнить обновление системы в целом. Если сравнивать с Windows XP, Synaptic выполняет функции сразу двух инструментов: программы установки и удаления программ, и центра обновлений.

16.7.10. Приложения из состава операционной системы

Далее перечислены некоторые приложения, входящие в состав операционной системы AltLinux:

- ❑ *Офисный пакет OpenOffice.Org* — полностью обеспечивает всю офисную функциональность, включая работу с документами и электронными таблицами, работу с базами данных, создание презентаций;
- ❑ *GIMP* — профессиональная обработка и создание растровых изображений;
- ❑ *Inkscape* — профессиональная обработка и создание векторных изображений;
- ❑ *scilab* и *tesMaxima* — программы для выполнения математических вычислений и инженерных расчетов;
- ❑ *Scribus* — программа для типографской верстки документов;
- ❑ *KDevelop* — универсальная многоязыковая интерактивная среда быстрой разработки приложений;
- ❑ *Quanta Plus* — среда разработки веб-приложений;
- ❑ *Lazarus* — среда визуального проектирования на языке Pascal.

Это далеко не полный перечень программного обеспечения, которое входит в комплект поставки AltLinux, но и этот перечень весьма внушителен.

Вопросы для самопроверки

1. Что такое «система BIOS», каковы ее функции?
2. Какую роль в операционной системе играет ядро?
3. Что такое «драйвер»?
4. Чем службы (сервисы) операционной системы отличаются от прикладных программ?
5. Для чего нужны системные оболочки?
6. Чем отличаются многозадачные операционные системы с разделением времени от многозадачных операционных систем без разделения времени?
7. Чем различаются кооперативная и вытесняющая многозадачность?
8. Что такое «операционная система реального времени»?
9. Что такое «топология сети»?

10. Каковы должны быть функции операционной системы, если рассматривать ее как набор необходимых функций?
11. Каковы должны быть предоставляемые сервисы операционной системы, если рассматривать ее как их совокупность?
12. Что такое «спулинг»?
13. Какими недостатками обладало подсемейство Windows 3.x?
14. Какие проблемы подсемейства Windows 9.x были решены в следующих версиях Windows и за счет чего?
15. При помощи какого инструмента производится настройка параметров безопасности в Windows XP?
16. Какие приложения поставляются в составе Windows XP?
17. С какими типами файловых систем работает операционная система AltLinux?
18. Каким инструментом обеспечивается настройка параметров безопасности в AltLinux?
19. Какие пользовательские приложения входят в состав операционной системы AltLinux?

Литература

1. Молчанов А. Ю. Системное программное обеспечение. СПб.: Питер, 2003.
2. Олифер В. Г., Олифер Н. А. Сетевые операционные системы. СПб.: Питер, 2007.
3. Таненбаум Э. Современные операционные системы. СПб.: Питер, 2007.
4. Таненбаум Э., Вудхалл А. Операционные системы: разработка и реализация. СПб.: Питер, 2007.
5. Таненбаум Э., М. Ван Стеен. Распределенные системы. Принципы и парадигмы. СПб.: Питер, 2003.

Глава 17 Офисное программное обеспечение

17.1. Текстовый процессор

17.2. Табличный процессор

17.3. Органайзер

17.4. Система электронного документооборота

Программное обеспечение для создания, редактирования, учета, хранения, обработки, поиска и представления документов на сегодняшний день играет большую роль в жизни нашей страны. Офисная работа, сопровождающая сегодня практически любой процесс, будь то промышленный или творческий, научный или спортивный, состоит не только из создания документов. Важную роль играет правильная и эффективная организация труда офисных работников. Для помощи в организации труда офисных работников, обеспечения взаимодействия, проведения собраний и планерок существует целый ряд программных продуктов: текстовые редакторы и процессоры, электронные таблицы, всевозможные органайзеры, планеры и прочие продукты. В этой главе освещены наиболее известные компьютерные информационные технологии, предназначенные для организации эффективной работы по созданию, редактированию, хранению, представлению, поиску, рассылке, перемещению и оформлению документов в современном офисе.

17.1. Текстовый процессор

17.1.1. Общее представление о функциональности

В эпоху текстовых редакторов главной задачей разработчиков было обеспечить при наборе, верстке и распечатке документа его максимальную схожесть с машинописным текстом. Для современного текстового процессора такого рода требования являются не верхним пределом, а нижней границей функциональности, минимальными требованиями. Какими же возможностями должен обладать полнофункциональный текстовый процессор?

Ввод и редактирование текста

Ввод и редактирование текста — это базовые задачи, для решения которых собственно и создается любой текстовый редактор (процессор). После ввода текста может возникнуть необходимость редактировать его: ввести дополнительный текст, удалить, переместить или скопировать часть уже введенного текста, исправить слова, в которых были обнаружены орфографические ошибки или пропущенные буквы. Для выполнения операций с фрагментами текста текстовый процессор должен обладать функциональностью выделения фрагментов текста.

Функциональность редактирования и выделения должна носить интеллектуальный характер. Текстовый процессор должен уметь отличать выделенный объект от фрагмента текста и уметь выполнять над нстекстовыми элементами документов различные дополнительные операции. например, выделить фрагмент текста и вставить ссылку на него, выделить векторный рисунок и вставить его растроную копию, выделить несколько элементов диаграммы и отредактировать их.

Поиск и замена

В современном текстовом процессоре должен быть механизм, обеспечивающий поиск заданного слова, словосочетания, сочетания символов или текста в заданном формате. Найденный текст может быть автоматически заменен предложенным фрагментом или выделен, для того чтобы пользователь сам мог принять решение, в каких случаях необходимо внести исправления.

Создание, открытие и сохранение документов

Механизмы создания, открытия и сохранения документов у разных текстовых процессоров могут различаться. Например, для этих операций может использоваться стандартное диалоговое окно открытия и сохранения файлов, либо текстовый процессор может иметь собственные диалоговые окна с расширенной функциональностью, позволяющие выполнять во время сохранения документов какие-то дополнительные сервисные операции (проверка, сжатие, структурирование).

Возможность сохранения документов в различных форматах — одна из важнейших функций современного текстового процессора. Поскольку мир современного

документооборота и распространения информации очень многообразен, один и тот же документ может быть востребован в разных своих представлениях:

- ❑ *в виде текста* — сохранение только текстового содержимого;
- ❑ *в виде документа* — сохранение в системе электронного документооборота (СЭД);
- ❑ *в виде документа другого формата* — для открытия в текстовом процессоре другого типа;
- ❑ *в формате PDF* — для последующей печати или чтения в другой операционной системе;
- ❑ *в формате HTML* — для публикации на веб-странице;
- ❑ *в виде записи блога* — для публикации в виртуальном дневнике.

Интеграция в документ дополнительных сведений, реквизитов и атрибутов

Поскольку сегодня практически весь крупный и средний бизнес, государственные учреждения и структуры, научные учреждения и учебные заведения оснащены тем или иным видом системы электронного документооборота, то наличие в документе дополнительных сведений, реквизитов и атрибутов в одном случае является необходимым, в другом облегчает процесс создания карточки документа и регистрации документа в информационной системе.

Форматирование

Для качественного отображения документа очень важно, чтобы текстовый процессор умел отображать фрагменты текста различными типами шрифта. Стандартным набором параметров шрифта, которые должен уметь изменять текстовый процессор, являются: *семейство* (рис. 17.1), *начертание*, в том числе обычное, полужирное, курсивное (рис. 17.2), и *размер* шрифта (рис. 17.3).

Arial Times Courier *Corsiva*

Рис. 17.1. Шрифты разных семейств

простой полужирный курсив *полужирный курсив*

Рис. 17.2. Текст разных начертаний

буквы в одном слове мОгУт быть разнОго разМера

Рис. 17.3. Символы разных размеров

Кроме того, текстовый процессор должен обеспечивать средства *подчеркивания* и *зачеркивания* шрифта (рис. 17.4), ввода *надстрочных* и *подстрочных* символов (рис. 17.5), задания *цвета* для шрифта и фона (рис. 17.6).

подчёркивание подчёркивание подчёркивание подчёркивание
 зачёркивание зачёркивание

Рис. 17.4. Варианты подчеркивания и зачеркивания

шрифт надстрочный и шрифт подстрочный

Рис. 17.5. Подстрочный и надстрочный текст

Фон темно-серый,
а шрифт – белый

Рис. 17.6. Разные цвета для шрифта и фона

К числу дополнительных параметров, имеющих не во всех текстовых процессорах, обычно относят средства для придания шрифту разнообразных эффектов, таких как *рельеф*, *тень* или *анимация* (рис. 17.7).

ОТТЕНЕНИЕ

Рис. 17.7. Дополнительные эффекты

Если стандартный набор параметров в основном используется в официальных документах и деловой переписке, то дополнительный может быть широко задействован при создании разного рода публикаций (буклетов, брошюр, афиш, веб-страниц).

Возможности некоторых текстовых процессоров, касающиеся изменения межбуквенного интервала и ширины букв, свойственны скорее настольным системам верстки.

Форматирование абзацев

Форматирование абзацев непосредственно связано с аккуратной версткой текста и заключается в задании величины выравнивания, отступов (слева, справа и первой строки) и междустрочного расстояния. При форматировании абзацев применяется четыре типа выравнивания: по правому краю, по левому краю, по центру и по ширине (рис. 17.8).

По левому краю

Форматирование абзацев непосредственно связано с аккуратной версткой текста, и заключается в задании выравнивания, отступов и междустрочного расстояния. При форматировании абзацев применяются четыре типа выравнивания: по правому краю, по левому краю, по центру и по ширине.

По правому краю

Форматирование абзацев непосредственно связано с аккуратной версткой текста, и заключается в задании выравнивания, отступов и междустрочного расстояния. При форматировании абзацев применяются четыре типа выравнивания: по правому краю, по левому краю, по центру и по ширине.

По центру

Форматирование абзацев непосредственно связано с аккуратной версткой текста, и заключается в задании выравнивания, отступов и междустрочного расстояния. При форматировании абзацев применяются четыре типа выравнивания: по правому краю, по левому краю, по центру и по ширине.

По ширине

Форматирование абзацев непосредственно связано с аккуратной версткой текста, и заключается в задании выравнивания, отступов и междустрочного расстояния. При форматировании абзацев применяются четыре типа выравнивания: по правому краю, по левому краю, по центру и по ширине.

Рис. 17.8. Варианты выравнивания

Создание списков

Создание списков, как нумерованных, так и маркированных, — одна из важнейших функций современного текстового процессора. При этом обычно поддерживается возможность создания вложенных списков (рис. 17.9).

Нумерованный

1. Первый
 - a. Первый-первый
 - b. Первый-второй
 - i. первый-второй-первый
 - ii. первый-второй-второй
 - iii. первый-второй-третий
2. второй
 - a. второй-первый
 - b. второй-второй
3. третий

Ненумерованный

- Первый
 - Первый-первый
 - Первый-второй
 - первый-второй-первый
 - первый-второй-второй
 - первый-второй-третий
- второй
 - второй-первый
 - второй-второй
- третий

Рис. 17.9. Нумерованный и маркированный списки с вложенными списками

Стилевое оформление документа

К числу инструментов текстового процессора, автоматизирующих подготовку документа, являются средства для автоматического создания оглавлений и указателей, списков иллюстраций и использованной литературы, а также для автоматического форматирования вводимого текста. Однако чтобы воспользоваться этими средствами, вводимый текст нужно оформлять предназначенным для этого стилем, например, заголовки — стилями заголовков с соблюдением уровней вложенности, текст — стилем обычного текста, и т. п.

Разбиение документа на разделы, страницы, нумерация страниц и оформление колонтитулов

При создании многостраничного документа весьма важно автоматизировать процедуры нумерации страниц, создания колонтитулов, разбиения документа на разделы, когда внутри каждого из разделов может существовать собственная нумерация страниц. Эти вопросы выходят на первый план, когда приходится оформлять в текстовом процессоре курсовую или дипломную работу, методические рекомендации или многостраничный документ, предназначенный для печати в малой типографии.

Использование текстовых фреймов для верстки документа

Текстовые фреймы обычно используются при создании шаблонов бланков официальных документов, а также разного рода брошюр, журналов, афиш и проспектов. С помощью фреймов любой объект, включая текст, иллюстрации или таблицы, можно разместить произвольным образом внутри печатного листа (рис. 17.10).

Использование текстовых фреймов для верстки документа

Этот прием редко используется для создания официальных документов, чаще используется для создания шаблонов бланков официальных документов (поскольку там нужно вставлять герб и атрибуты оформления слева сверху, справа сверху, то есть размещать фрагменты текста), и очень часто используется для создания разного рода брошюр, журналов, факсов и

Использование текстовых фреймов для верстки документа

Этот прием редко используется для создания официальных документов, чаще используется для создания шаблонов бланков официальных документов (поскольку там нужно вставлять герб и атрибуты оформления слева сверху, справа сверху, то есть

Использование текстовых фреймов для верстки документа

Этот прием редко используется для создания официальных документов, чаще используется для создания шаблонов бланков официальных документов (поскольку там нужно вставлять герб и атрибуты оформления слева сверху, справа сверху, то есть

Использование текстовых фреймов для верстки документа

Использование текстовых фреймов для верстки документа

Рис. 17.10. Верстка с помощью текстовых фреймов

Таблицы

Возможность табличного представления информации также является одним из важнейших свойств современного текстового процессора. В зависимости от возможностей текстового процессора таблица может допускать более или менее разнообразное стиливое оформление. Текстовый процессор может поддерживать преобразование текста в таблицу и таблицы в текст, заполнение таблицы из внешнего источника данных, некоторую автоматизацию подсчетов в таблице, например, суммирование по строкам и столбцам.

Вставка в документ нетекстовых объектов

Именно средства вставки в текстовый документ графических и мультимедийных объектов, гиперссылок и других сложных объектов отличают сложный текстовый процессор от обычного текстового редактора. Эта функциональность позволяет оформлять текстовые документы, которые затем могут быть легко преобразованы как в мультимедийную презентацию, так и в полноценную веб-страницу.

Рецензирование и отображение изменений

Эта функция обогащает текстовый процессор возможностью взаимодействия и коллективной работы нескольких человек над одним документом. Автор документа всегда может увидеть, что именно исправил рецензент, прочесть написанные примечания, принять или отвергнуть предложенные изменения.

Программирование и автоматизация рутинных операций

Часто некоторую последовательность операций приходится повторять. В офисных приложениях для того, чтобы рутинные операции такого сорта не выполнять многократно, есть возможность записи или программирования макрокоманд и связывания этих макрокоманд с определенным сочетанием клавиш или элементом управления. Многие офисные приложения имеют развитый макроязык (язык написания макрокоманд) и даже встроенные системы программирования.

Подсчет статистики документа

Хотя средства подсчета статистики документа обычно не относят к самым важным, они могут быть чрезвычайно полезны для систем электронного документооборота, позволяя выяснить, сколько документов, абзацев, предложений, слов и букв было обработано.

Печать конвертов для рассылки со слиянием данных

Автоматизированную печать конвертов (или автоматизированную рассылку документа по электронным адресам) в режиме, когда данные адресатов выбираются из источника данных, можно считать важной функцией, которая экономит время.

17.1.2. Сравнительная характеристика текстовых процессоров Microsoft Word, OpenOffice.org Writer и Abiword

В версии Microsoft Office 2007 была реализована новая концепция пользовательского интерфейса, получившая название *Fluent*. Место главного меню и панелей инструментов заняла так называемая *лента* (рис. 17.11).

Появление нового интерфейса объясняется большим количеством команд Microsoft Office (более полутора тысяч), в которых трудно ориентироваться с помощью стандартного для предыдущих версий интерфейса.

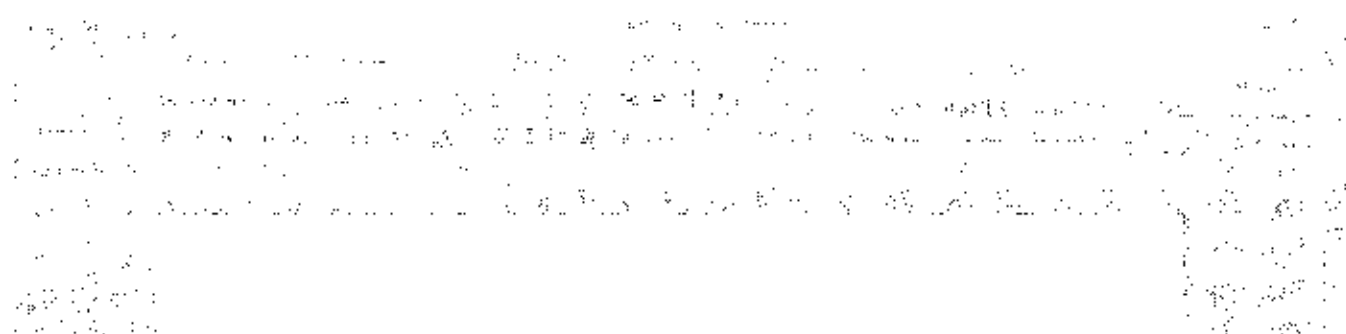


Рис. 17.11. Microsoft Word 2007

Текстовый процессор Writer, входящий в состав свободного пакета OpenOffice.org 3, является свободной альтернативой Microsoft Word. Можно сказать, что по основной функциональности эти программы совпадают и вполне могли бы конкурировать друг с другом (рис. 17.12).

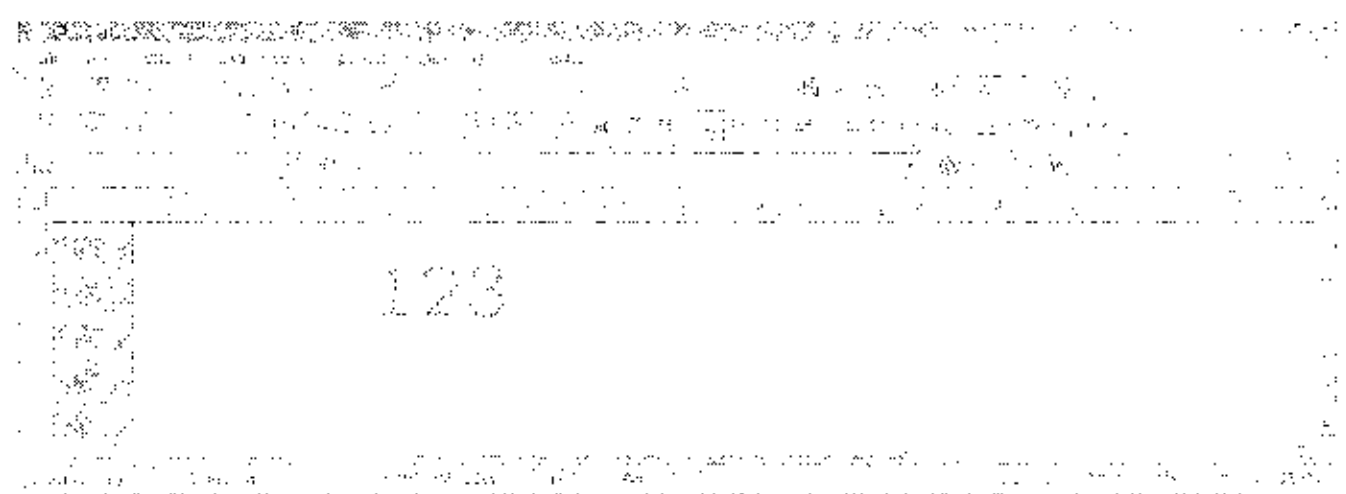


Рис. 17.12. OpenOffice.org Writer

Свободный текстовый процессор, Abiword, считается «легкой» программой, так как потребляет гораздо меньше ресурсов, чем Microsoft Word или OpenOffice.org Writer. Однако, несмотря на «легкость» и гораздо меньшее количество инструментов, по базовой функциональности этот свободный текстовый процессор практически им не уступает (рис. 17.13).

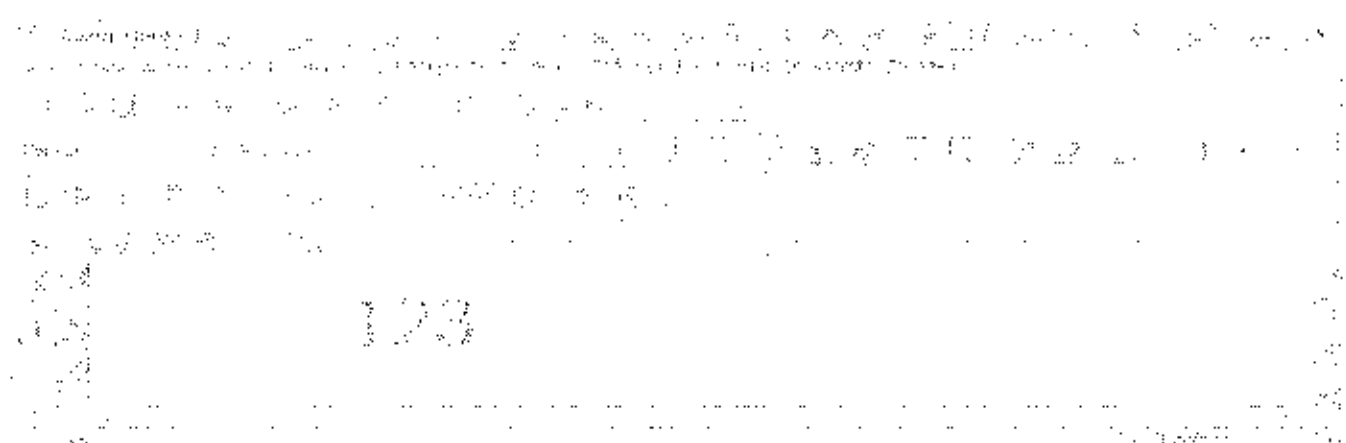


Рис. 17.13. Abiword

В табл. 17.1 приведены сравнительные характеристики рассмотренных текстовых процессоров.

Таблица 17.1. Сравнение функциональных характеристик текстовых процессоров

Характеристика	Microsoft Word	OpenOffice.org Writer	Abiword
Ввод и редактирование текста	Есть	Есть	Есть
Поиск и замена текста	Есть	Есть	Есть, отсутствует поиск по стилю
Создание, открытие и сохранение документов	Есть, функциональность расширенная	Есть, функциональность расширенная	Есть
Сохранение документов в различных форматах	Есть, в том числе PDF	Есть, в том числе PDF	Есть, в том числе PDF
Интеграция в документ дополнительных сведений, реквизитов и атрибутов	Есть	Есть	Есть
Форматирование шрифта	Есть, функциональность расширенная	Есть, функциональность расширенная	Есть
Форматирование абзацев	Есть	Есть	Есть
Создание списков	Есть, функциональность расширенная	Есть, функциональность расширенная	Есть
Применение стилей для создания структуры документа	Есть	Есть	Есть
Разбиение документа на разделы, страницы, нумерация страниц и оформление колонтитулов	Есть	Есть	Есть
Использование текстовых фреймов для верстки документа	Есть	Есть	Нет
Работа с таблицами	Есть	Есть	Есть
Вставка в документ нетекстовых объектов	Есть	Есть	Есть, только изображения
Средства программирования и автоматизации рутинных операций	Есть	Есть	Есть, поддерживаются сценарии
Автозамена	Есть	Есть	Нет
Подсчет статистики документа	Есть	Есть	Есть
Печать конвертов для рассылки со слиянием данных	Есть	Есть	Нет

Из таблицы видно, что функциональность текстовых процессоров практически совпадает. С точки зрения совместимости документов, созданных в различных текстовых процессорах, на уровне текста, стилей и стандартных таблиц совместимость практически полная.

17.2. Табличный процессор

17.2.1. Общее представление о функциональности

Какой бы темы мы ни коснулись, будь то список книг в библиотеке, перечень учебных предметов, список студентов или ассортимент товаров, все это желательно представить в табличной форме. Однако для простого сведения каких бы то ни было данных в таблицу достаточно тех средств, которые предлагают текстовые процессоры. Мощь же электронных таблиц проявляется в двух главных возможностях: адресации ячеек и совершении операций над данными в ячейках.

Адресация ячеек

В электронной таблице каждая ячейка имеет адрес, который состоит из имени столбца и номера строки и который однозначно ее идентифицирует. По этому адресу можно обратиться и прочитать записанные в конкретной ячейке данные (рис. 17.14).



Рис. 17.14. Адресация ячейки в электронной таблице

Как видно на рисунке, в таблице выделены столбец В и строка 4, на пересечении которых находится ячейка с именем В4. В ячейке находятся текстовые данные, а именно, слово **Данные**, которое и отражено в строке формул.

Совершение операций над данными в ячейках

Изменим данные в таблице, введя вместо текстовых данных в две ячейки числовые данные, а в третью ячейку формулу, в которой укажем адреса этих двух ячеек (рис. 17.15).



Рис. 17.15. Функция, адресующая две ячейки

Формула определяет сложение значения, которое находится в ячейке B4, со значением из ячейки C4. Результат этого действия можно увидеть в той ячейке, в которую вписана формула (рис. 17.16).

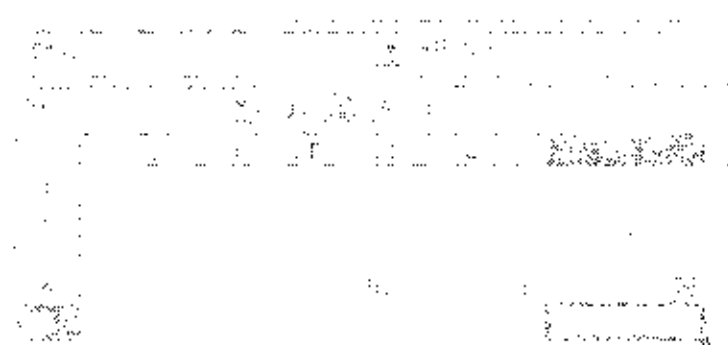


Рис. 17.16. Результат выполнения вычисления

При вводе формул есть возможность задать режим адресации. По умолчанию адресация является *относительной*. Это значит, что формула ищет значения своих аргументов в ячейках, имеющих фиксированное положение относительно самой формулы.

Пример. Рассмотрим формулу на рис. 17.15. Она обращается к двум ячейкам, одна из которых расположена на один, а вторая — на два столбца левее этой формулы. Зададим значения в ячейках B3 и C3, а затем скопируем формулу из ячейки D4 в ячейку D3. Как видите, формула теперь выполняет операцию суммирования над ячейками B3 и C3, расположенными относительно нее по-прежнему на один и на два столбца левее (рис. 17.17).



Рис. 17.17. Копирование формулы с относительной адресацией

Второй способ адресации называется *абсолютным*. В этом способе формула независимо от ее положения всегда указывает на те ячейки, адреса которых изначально были в нее введены. При абсолютной адресации имя столбца и номер строки в адресе ячейки предваряются знаком доллара \$.

Пример. На рис. 17.18 показан результат копирования формулы с абсолютной адресацией.

В случае абсолютной адресации новое положение формулы не изменило подставляемые в нее значения — формула продолжает ссылаться на ячейки B4 и C4.



Рис. 17.18. Копирование формулы с абсолютной адресацией

Главное назначение табличных процессоров состоит в *автоматизации расчетов*. Эта функциональность свойственна любой программе класса табличных процессоров. Различие между разными программными продуктами касается, как правило, набора встроенных функций, которые можно вызывать для выполнения операций, максимальном количестве столбцов и строк в таблице и наличии сценариев, призванных удовлетворить основные потребности пользователя в выполнении тех или иных расчетов и аналитических исследований.

17.2.2. Дополнительные возможности табличного процессора

Помимо вычислительных возможностей и вполне ожидаемой функциональности по созданию, сохранению и открытию документов, а также изменению параметров шрифта для отображения значений в ячейках, современный табличный процессор обычно обладает множеством дополнительных возможностей, которые превращают его одновременно в инструмент художника и в среду программирования. Перечислим наиболее часто встречающиеся дополнительные возможности табличного процессора.

- ❑ **Оформление результатов в виде диаграмм.** Во многих случаях результатами вычислений (или составления отчетов) бывают ряды данных. Колонки чисел менее наглядны, чем рисунки. По этой причине во все современные электронные таблицы встроен инструмент представления данных в виде диаграмм (рис. 17.19).

Обычно такой инструмент позволяет выбирать из широкого диапазона диаграмм разного типа, а после создания диаграммы — изменять ее внешний вид в широких пределах. Если на рис. 17.19 для заданного диапазона данных была создана диаграмма в виде двух линейных графиков, то на рис. 17.20 эта диаграмма преобразована в объемную гистограмму.

- ❑ **Сортировка и фильтрация данных.** Данные в каждом столбце можно отсортировать по возрастанию или убыванию, при этом можно сортировать данные только в выделенном столбце или во всех строках, связанных с данным столбцом. Также можно на выбранный столбец или на несколько столбцов накладывать фильтры по значению (рис. 17.21 и 17.22).
- ❑ **Автоформатирование.** В табличном процессоре обычно поддерживается широкая гамма вариантов встроенного стилевого оформления, которые можно применять по желанию пользователя. На рис. 17.23 представлен пример применения стилевой схемы Январь (OpenOffice.org Calc).

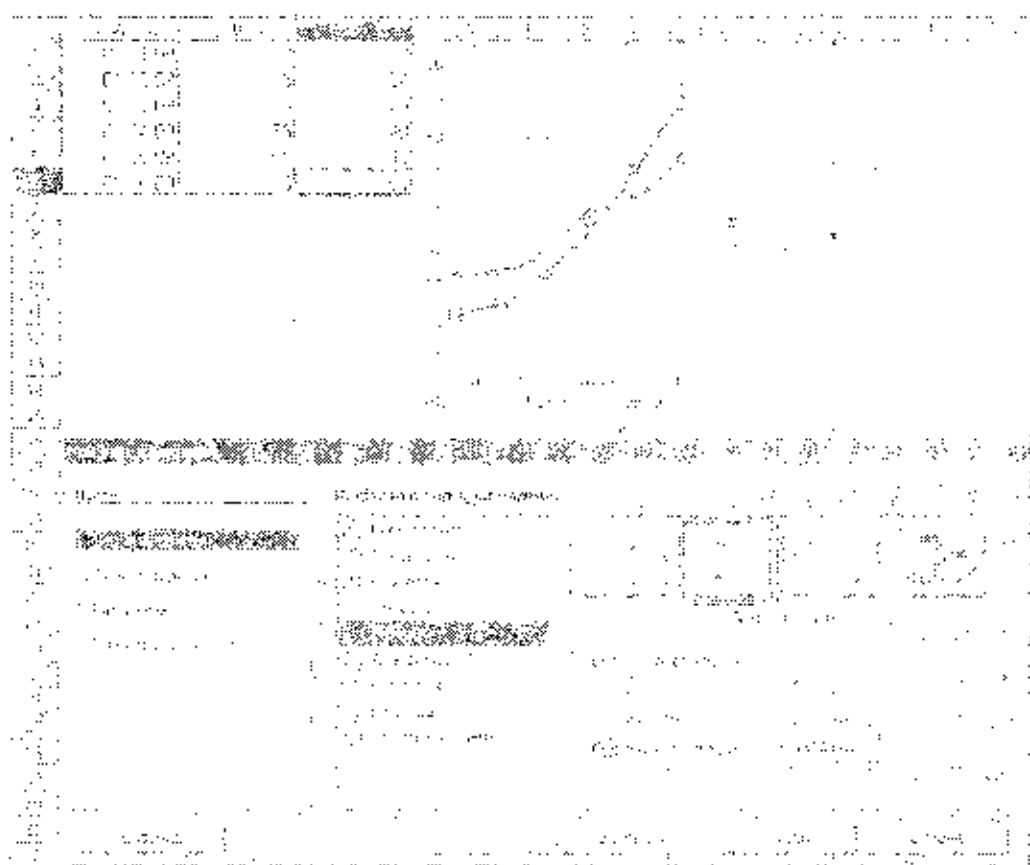


Рис. 17.19. Оформление результатов в виде диаграмм

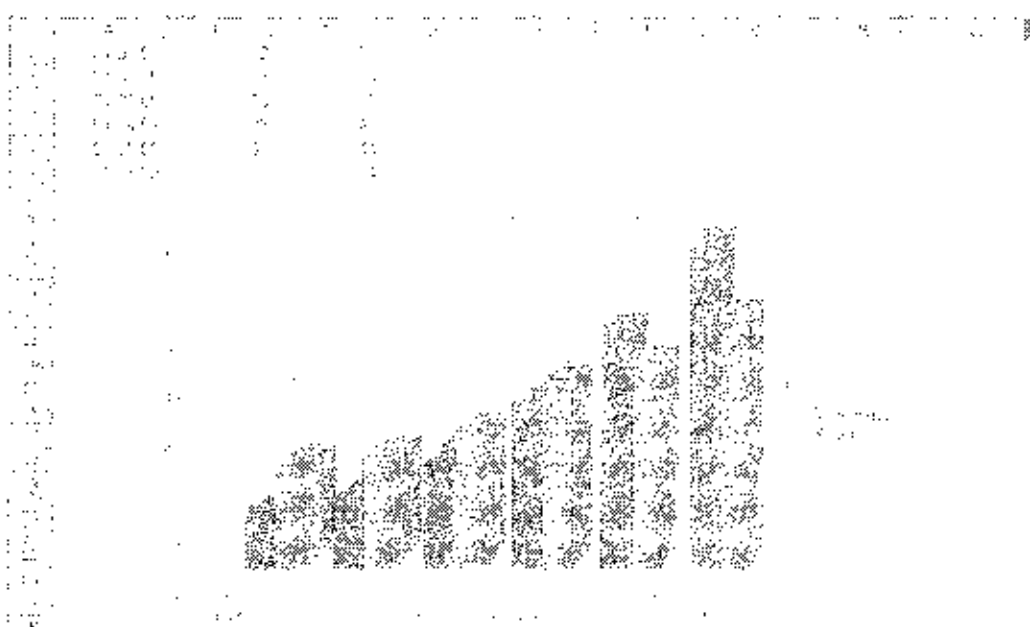


Рис. 17.20. Объемная гистограмма

- ❑ **Условное форматирование.** Эта функция предоставляет пользователю возможность определить формат представления ячеек, значения в которых отвечают заданному условию (рис. 17.24).
- ❑ **Извлечение данных из внешних источников и связь с внешними данными.** Эта функциональность расширяет работу табличного процессора, сразу переводя его из локальных приложений в распределенные. Возможность получать

в таблицы табличного процессора внешние данные и оперативно (если создана связь) обновлять их делает данные, обрабатываемые табличным процессором, динамичными и актуальными.

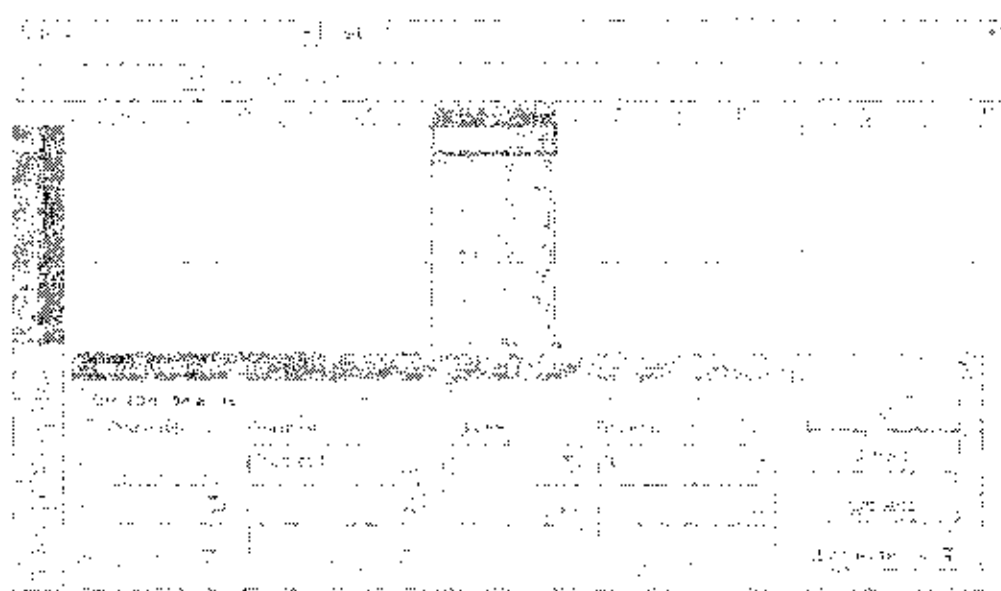


Рис. 17.21. Задание параметров фильтрации

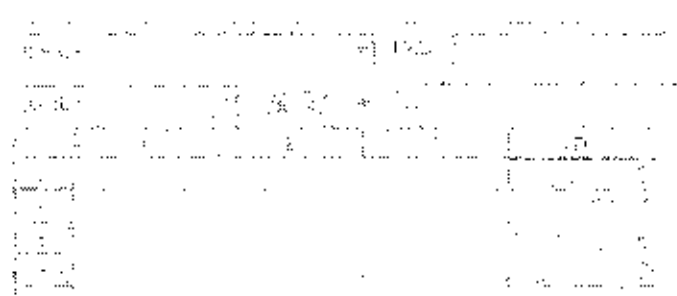


Рис. 17.22. Результат применения фильтра

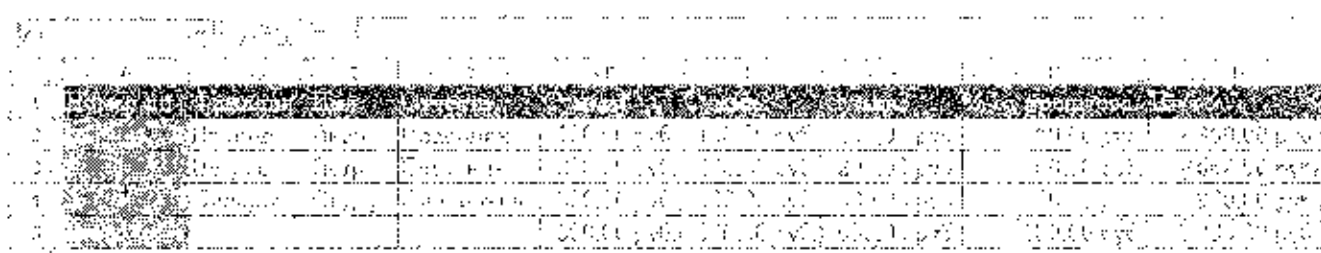


Рис. 17.23. Автоформатирование



Рис. 17.24. Условное форматирование

- *Построение сводных таблиц.* Сводные таблицы — это многомерные таблицы, которые позволяют на одном листе легко анализировать и просматривать зависимости типа «уровень продаж по годам», «уровень продаж по регионам», «уровень продаж по годам и по менеджерам».
- *Сценарии работы с данными.* Сценарии позволяют подводить итоги различного вида, выполнить проверку допустимого диапазона значений, провести поиск решения.
- *Вставка диаграмм.* Несмотря на то, что эта функция отнесена к вспомогательным, особенность человеческого восприятия делает ее одной из наиболее важных. Обычно, когда человек смотрит в таблицу, он видит очень мало, и ему нужно много времени, чтобы сделать какие-то выводы. Когда же человек смотрит на диаграмму, построенную на основании данных таблицы, он видит сразу все.
- *Вставка графических объектов.* Графические объекты расширяют диапазон восприятия числовых данных человеком. Например, при подготовке информации для представления потенциальному инвестору принято наполнять таблицу рисунками.

17.2.3. Сравнительная характеристика табличных процессоров Microsoft Excel, OpenOffice.org Calc и Gnumeric

Функционально эти три программы не очень сильно различаются, основные отличия в дизайне (рис. 17.25).

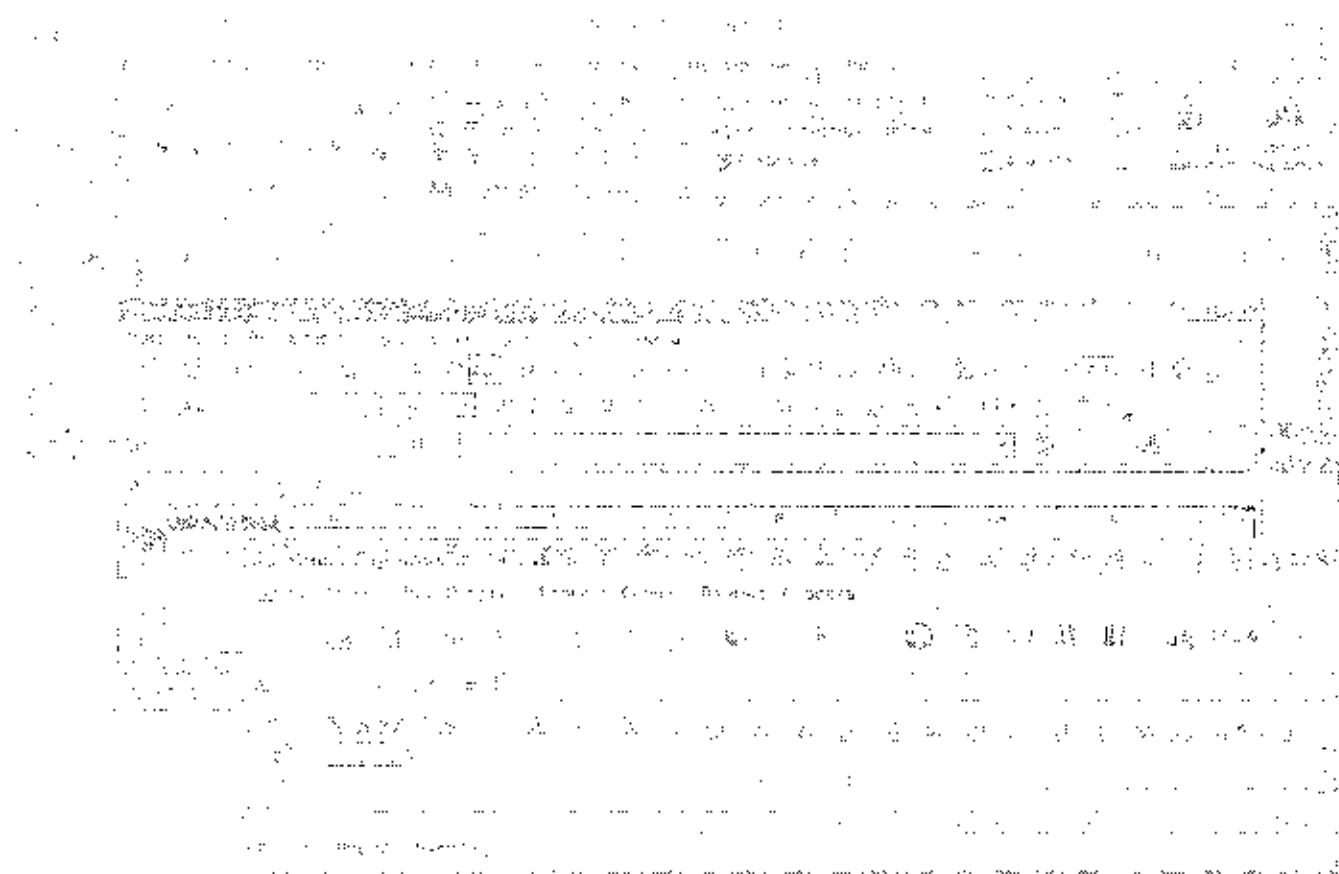


Рис. 17.25. Табличные процессоры

Сравнительные характеристики этих процессоров представлены в табл. 17.2.

Таблица 17.2. Сравнение характеристик табличных процессоров

Функциональность	Excel	Calc	Gnumeric
Лицензия	Проприетарная, нужно покупать лицензию	Свободная	Свободная
Оформление	Новый дизайн графического интерфейса пользователя	Стандартный дизайн	Стандартный дизайн
Работа в Windows	Хорошая	Удовлетворительная	Удовлетворительная
Работа в Linux	Не работает	Хорошая	Хорошая
Наличие встроенного языка	Полноценная среда программирования на встроенном языке	Полноценная среда программирования на встроенном языке	Язык сценариев
Ориентация на область применения	Финансовая и экономическая	Финансовая и экономическая	Статистические вычисления
Количество строк	1048576	65536	65536
Количество столбцов	65536	1024	256
Импорт внешних данных	Есть	Есть	Только из текстового файла
Связь с внешними данными	Есть	Есть	Нет
Импорт данных с веб-страниц	Есть	Есть	Нет
Интеграция с офисным пакетом	Высокая	Хорошая	Нет
Наличие в офисном пакете инструментов построения СЭД	Есть	Нет	Нет
Построение диаграмм	Есть	Есть	Есть
Вставка рисунков	Есть	Есть	Есть

17.3. Органайзер

17.3.1. Общие принципы использования программных инструментов организации деятельности

Организация деятельности подразумевает наличие нескольких поддающихся формализации и документированию описаний; необходимо вырабатывать решения, на их основе формулировать задания для персонала, устанавливать сроки выполнения заданий, устанавливать сроки контроля, обеспечивать взаимодействие между членами коллектива.

Возможности органайзера в отношении персональной самоорганизации:

- ☐ составление списка запланированных дел;
- ☐ планирование встреч;

- ☐ планирование событий;
- ☐ отображение запланированных встреч и событий на календаре в одном из режимов просмотра (один день, рабочая неделя, полная неделя, месяц, квартал, год);
- ☐ отображение запланированных дел на линии времени (в виде календарного графика);
- ☐ оформление повторяющихся дел, событий или встреч;
- ☐ гибкая настройка параметров рабочего времени, праздников, выходных дней;
- ☐ напоминание о важных регулярных событиях;
- ☐ фильтрация и поиск записей по заданным параметрам.

Возможности органайзера в отношении организации рабочей группы:

- ☐ подключение к выполнению задания членов группы в качестве ответственных за это задание;
- ☐ рассылка оповещений, напоминаний и приглашений на встречу группы;
- ☐ публикация и обновление календаря на сервере групповой работы, совместном сетевом журнале или в блоге.

17.3.2. Сравнительная характеристика органайзеров Microsoft Outlook, doOrganizer, KOrganizer и Mozilla Sunbird

Microsoft Outlook и doOrganizer — это коммерческие органайзеры, реализованные только для Microsoft Windows.

Microsoft Outlook — самый известный и широко применяемый в мире органайзер. Он совмещен с программой отправки почты и чтения новостей, полнофункциональной адресной книгой и журналом открытия документов. Его функциональность несколько перекрывает перечисленные ранее возможности.

doOrganizer — это даже не органайзер, а персональная информационная система, в которую, помимо функций персонального планирования и учета времени, входит еще ряд интересных функций: планирование персональных финансов, ведение журнала, создание записок, генератор отчетов, почтовая программа, накопитель ссылок и многое другое. Окна обеих этих программ представлены на рис. 17.26.

Программы KOrganizer и Mozilla Sunbird представляют собой свободные органайзеры, их можно загрузить из Интернета и установить на компьютер бесплатно. Одна из них, KOrganizer, устанавливается только на компьютеры с установленной оконной средой KDE (как правило, это компьютеры, работающие под управлением операционной системы Linux), вторая, Mozilla Sunbird, имеет исполнения для разных платформ.

С точки зрения функциональности KOrganizer полностью отвечает требованиям для органайзера индивидуальной и групповой работы. Mozilla Sunbird не имеет инструментов групповой организации и может применяться только как персональный органайзер. Окна обеих программ показаны на рис. 17.27.

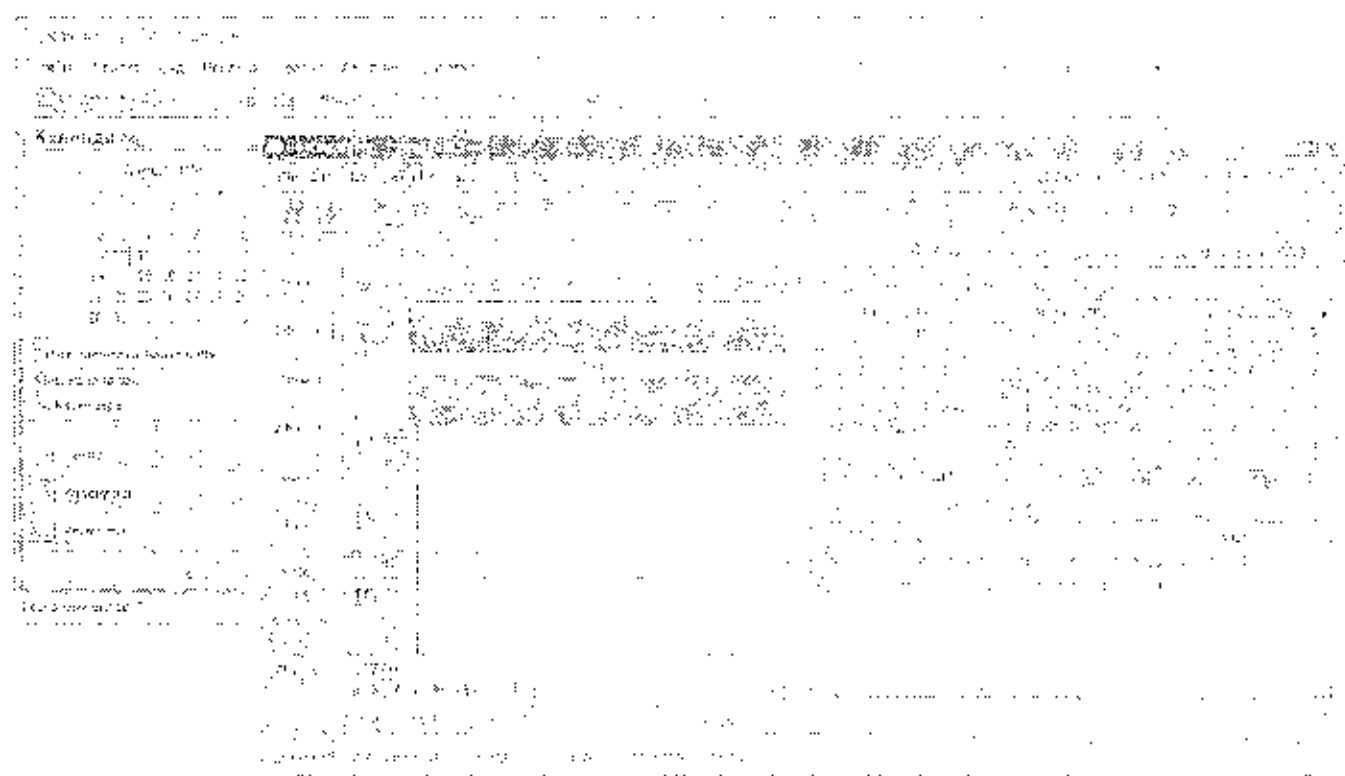


Рис. 17.26. Outlook и doOrganizer

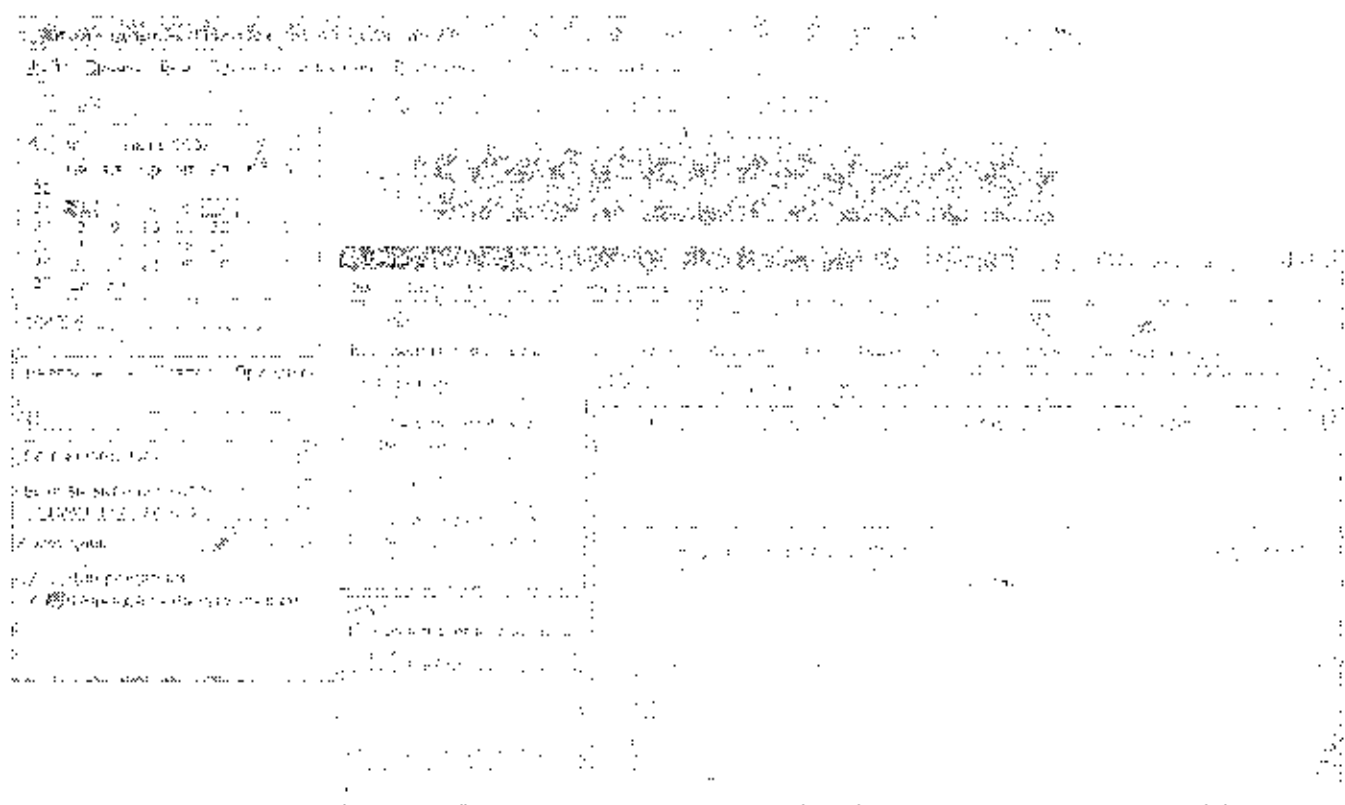


Рис. 17.27. KOrganizer и Mozilla Sunbird

17.4. Система электронного документооборота

17.4.1. Назначение и функциональность

В небольших компаниях документооборот вполне может контролироваться без использования специализированных программ. Поток документов невелик, прием, регистрация и учет документов не представляют трудности. В крупных компаниях управлять потоками документов становится все труднее. Если предприятие создает филиалы, то потоки документов становятся распределенными в пространстве, поиск документов превращается в самостоятельный вид деятельности, количество людей, занятых в сфере документооборота, растет.

Кроме того, есть организации, в которых документооборот является не сервисной функцией, а практическим содержанием работы. К таким организациям относятся архивные хранилища, юридические компании и многие другие. С точки зрения документооборота наша страна проходит довольно сложный этап: бумажный электронный документооборот тесно соседствует с бумажным, далеко не все виды документов могут быть выполнены в электронном виде, и это предъявляет дополнительные требования к системе документооборота, которая вдобавок к основным своим функциям должна еще поддерживать непрерывную связь между бумажными оригиналами и их электронными копиями.

Документы имеют свой жизненный цикл, от создания до уничтожения. Этот цикл состоит из четырех основных фаз.

1. *Создание документов.* Установление необходимого перечня используемых документов, их названий и функционального назначения, проектирование форм документов, наблюдение за их использованием с применением современных информационных технологий.
2. *Хранение и использование документированной информации.* Формирование дел, создание файлов и систем поиска документов и информации, развитие систем передачи информации, телекоммуникаций, копирования и тиражирования документов; создание центров хранения документов, или административных архивов (в российской терминологии – текущие архивы).
3. *Передача документов на постоянное (государственное) хранение.* Составление перечней документов со сроками хранения, оценка документов, идентификация и описание каждого документа и комплексов документов. Ценность и необходимость сохранения признается за документами, которые могут служить доказательством существовавших сделок или использоваться для проведения исторических исследований.
4. *Управление архивами.* Проектирование и создание архивохранилищ, совершенствование методов консервации и реставрации, систематизации и описания архивных фондов, доступа к документам, составления справочных средств и распространения в обществе информации об архивах.

В системе электронного документооборота с целью реализации этих фаз жизненного цикла документов используются перечисленные далее функции.

- ☐ Регистрация документов – оформление единой электронной карточки как на бумажный, так и на безбумажный документ.
- ☐ Ввод и преобразование документов в электронный формат:
 - ☐ потоковое сканирование;
 - ☐ получение документа по факсу;
 - ☐ получение документа по электронной почте;
 - ☐ получение документа из файловой системы;
 - ☐ извлечение штрих-кода.
- ☐ Организация хранения документов в защищенном хранилище повышенной надежности.
- ☐ Поиск документов по заданным реквизитам и содержанию.
- ☐ Работа с содержимым документа.
- ☐ Разработка и поддержание жизненного цикла документов разного вида.
- ☐ Работа с версиями документа.
- ☐ Защита документов от несанкционированного доступа.
- ☐ Удостоверение документа электронной цифровой подписью.
- ☐ Организация коллективной работы с документами.
- ☐ Вывод документов для просмотра и на печать.
- ☐ Создание комплексных отчетов по подборке документов.

17.4.2. Сравнительная характеристика систем электронного документооборота DIRECTUM, Digital Design, ЕВФРАТ-Документооборот

Одна из наиболее успешных реализаций в России систем электронного документооборота – DIRECTUM. Эта система состоит из нескольких функциональных модулей.

- ☐ *Управление электронными документами.* Создание и хранение различных неструктурированных документов (тексты Microsoft Word, таблицы Microsoft Excel, схемы Microsoft Visio, рисунки CorelDraw, видео и др.); расширенная поддержка версий документов и ЭЦП; структурирование документов по папкам; назначение прав доступа на документы; история работы с документами; полнотекстовый и атрибутивный поиск документов.
- ☐ *Управление деловыми процессами.* Поддержка процессов согласования и обработки документов на всех стадиях жизненного цикла; выдача электронных заданий и контроль их исполнения; взаимодействие между сотрудниками в ходе бизнес-процессов; поддержка свободных и жестких маршрутов; богатые расширяемые библиотеки блоков для формирования маршрутов.

- *Управление договорами.* Организация процесса согласования и регистрации договоров и сопутствующих документов, а также оперативной работы с ними (поиск, анализ, редактирование и т. д.).
- *Управление совещаниями.* Организация подготовки и проведения совещаний (согласование места и времени, состава участников, повестки); формирование и рассылка протокола; контроль исполнения решений совещания.
- *Управление взаимодействием с клиентами.* Ведение единой базы организаций и контактных лиц; ведение истории встреч, звонков и переписки с клиентами; сопровождение процесса продаж в соответствии с регламентированными стадиями; планирование маркетинговых мероприятий; анализ эффективности продаж и маркетинговых воздействий.
- *Канцелярия.* Регистрация бумажных документов в соответствии с требованиями Государственной системы документационного обеспечения управления (ГСДОУ); ведение номенклатуры дел с гибкими правилами нумерации; рассылка и контроль местонахождения бумажных документов; организация обмена электронными документами с ЭЦП с другими организациями.
- *Обращения граждан.* Организация работы с обращениями граждан в соответствии с ФЗ № 59-ФЗ «О порядке рассмотрения обращений граждан Российской Федерации» в государственных организациях и на крупных предприятиях.
- *Управление показателями эффективности.* Оперативный контроль и анализ бизнес-процессов предприятия по ключевым показателям эффективности с поддержкой сбалансированной системы показателей (BSC).

Автономный модуль «Делопроизводство и архив» от Digital Design, построенный на известном решении Microsoft под названием DocsVision, поддерживает следующую функциональность:

- обработка корреспонденции;
- создание иерархической структуры хранилища документов;
- хранение документов любого формата (текстовые документы, таблицы, чертежи и др.), а также их классификацию;
- автоматизированный перенос документов в электронное хранилище из файловой системы;
- совместная работа с документами;
- поиск документов в хранилище;
- построение отчетов по документам в архиве;
- автоматизация процесса регистрации входящих, исходящих и внутренних документов;
- поточное сканирование и штрих-кодирование бумажных документов;
- организация электронного архива;
- контроль исполнительской дисциплины, включая выдачу поручений (резолюций) и контроль их исполнения;
- согласование документов;
- отслеживание версий документов;

- ☐ поддержка механизмов ЭЦП (электронная цифровая подпись) и шифрования документов.

Система электронного документооборота «ЕВФРАТ-Документооборот» от Cognitive Technologies — один из старейших продуктов, обслуживающих документооборот на Российском рынке. Реализует следующую функциональность:

- ☐ автоматизированная регистрация документов и заданий;
- ☐ мгновенный поиск информации;
- ☐ эффективное взаимодействие сотрудников в рамках работ по документам;
- ☐ контроль выполнения работ, инициируемых документами и заданиями;
- ☐ мониторинг состояния выполняемых процессов и анализ загрузки персонала за счет формирования различных журналов и аналитических отчетов;
- ☐ разграничение прав доступа сотрудников к информации;
- ☐ долговременное хранение документов организации.

Вопросы для самопроверки

1. Назовите функциональность, важную для современного текстового процессора?
2. В чем различия интерфейса пользователя Microsoft Word 2007 и OpenOffice.org Writer?
3. Каковы функциональные различия Microsoft Word 2007 и OpenOffice.org Writer?
4. Перечислите главные функции табличного процессора?
5. В чем состоит дополнительная функциональность табличного процессора?
6. Чем табличный процессор отличается от базы данных?
7. Что такое «сводные таблицы», для чего они нужны?
8. Каковы основные функциональные различия Microsoft Excel 2007 и OpenOffice.org Calc?
9. Каково назначение и типичная функциональность органайзера?
10. Какие системы электронного документооборота вам известны?
11. Какова базовая функциональность системы электронного документооборота?

Литература

1. Волков В. Б. Линукс Юниор. Книга для учителя. М.: ДМК, 2009.
2. Волков В. Б. Понятный самоучитель Excel 2007. СПб.: Питер, 2008.
3. Волков В. Б. Понятный самоучитель Word 2007. СПб.: Питер, 2008.
4. Еременко А., Чернов Д. Русский перевод OpenOffice.org 3 Руководство по Calc. OOoAuthors — электронный вариант.

Глава 18 Средства мультимедиа

18.1. Представление о мультимедиа

18.2. Звук

18.3. Изображения

18.4. Видео

18.5. Мультимедиа-презентации

Несмотря на присутствие в слове «мультимедиа» корня «мульти», что значит «много», все многообразие мультимедиа сводится к двум понятиям: звук и изображение. Мультимедийный документ — это документ, в котором к тексту добавлены изображение и звук, только звук или только изображение, либо звук и изображение призваны заменить собой текст. Ощущения, воспринимаемые другими органами чувств (запах, вкус, боль, прикосновение, тяжесть), передавать через каналы связи и записывать на диск современная техника еще не умеет. Зато в передаче, хранении и воспроизведении звука и изображения за довольно короткий исторический период времени были достигнуты большие успехи.

18.1. Представление о мультимедиа

Превращению компьютера в мультимедиа-центр способствовали три основных фактора:

1. *Появление и развитие технологий отображения и воспроизведения информации одновременно с удешевлением технологий поставки.* С начала 90-х гг. XX столетия последовал взрывной рост количества устройств, подключаемых к компьютеру,

причем большинство из них было мультимедийными: мониторы со все улучшающимся качеством воспроизведения и увеличенным размером экрана; звуковые колонки, сначала монофонические, затем стереофонические; midi-клавиатуры; цифровые диктофоны; ручные и планшетные сканеры; графические планшеты и дигитайзеры; цифровые фотоаппараты и видеокамеры; цветные принтеры, в том числе предназначенные для печати на фотобумаге; тачпады и трекболы; микрофоны и гарнитуры; радио- и TV-тюнеры; устройства для воспроизведения, а чуть позже и для записи CD- и DVD-дисков; наконец, видеопроекторы и интерактивные доски со звуковым сопровождением и дистанционным управлением.

2. *Развитие индустрии компьютерных игр.* Это одна из наиболее важных движущих сил, заставляющих пользователей покупать все более производительные компьютеры.

3. *Развитие сетевых и мобильных технологий, в том числе интернет-технологий.* Передача звука и изображения при сравнительно малых скоростях на заре развития интернет-технологий привело к активному поиску способов сжатия и кодирования информации. После того как были разработаны соответствующие форматы и способы сжатия, это привело к появлению новых технологий: потокового радио, потокового видео, удаленных мультимедиа-презентаций и видеоконференций. Одновременно изобретение способов надежного высококачественного сжатия звука и видео, а также дисков большого формата привело к перелому в индустрии распространения звукозаписей и кинофильмов: стало возможным поместить полнометражный фильм хорошего качества на один DVD-диск.

Надо отметить, что бытовая и сетевая среда обитания в последние годы все больше и больше приобретают мультимедийный характер. Уже не в диковинку «говорящие» и «понимающие» человеческую речь бытовые приборы и автомобили, активно разрабатывается концепция «умного дома», в котором, возможно впервые, будут преодолены рамки «звук плюс видео», поскольку каналов задач управления информационной системы теперь будет еще и поддержание комфортной температуры, создание приятного аромата в атмосфере дома и пр. Интернет также все больше оснащается средствами мультимедиа. Появление технологии flash привело к взрывному росту количества сайтов.

Современная трактовка термина «мультимедиа» объединяет несколько понятий:

- *звук* — все то, что относится к записи, обработке, передаче, хранению и воспроизведению звука;
- *изображение* — все способы получения изображений в цифровом виде, их обработки, передачи, хранения, распознавания и отображения на разных носителях;
- *видео* — все то, что связано с последовательностью изображений (кадров), воспроизводимых как движущееся изображение;
- *мультимедиа-документы* — сочетание текста, изображения и звука;
- *мультимедиа-презентации* — компьютерные продукты, при работе с которыми объединяются текст, звук, разнообразная графика, анимация, видео;

- *мультимедиа диски* — сочетание мультимедиа-презентации с интерактивным мультимедиа-меню;
- *мультимедиа-хранилища и базы данных* — огромные массивы мультимедиа-данных, потребовавшие новых подходов к хранению, обеспечению целостности их разделения доступа, к технологиям поиска и извлечения информации.

18.2. Звук

18.2.1. Разновидности звуковых данных в компьютере

Для того чтобы получить звук в цифровой форме, то есть закодировать его в виде последовательности чисел, необходимо «поймать» каким-то образом звуковые колебания в воздухе, превратить их в колебания электрического тока, а затем преобразовать эти колебания в числа. В этом случае получается *оцифрованный звук* — первая из двух разновидностей звуковых данных, с которыми работают компьютерные устройства. Помимо оцифрованного звука, на компьютере для воспроизведения музыки применяется еще одна его разновидность — *синтезированный звук*, или *звук в формате MIDI*.

Оцифровка звука

Преобразование из аналогового электрического сигнала в цифровой код происходит так: амплитуда аналогового электрического сигнала измеряется через определенные промежутки времени и записывается в виде чисел. Частота, с которой производятся такие измерения, называется *частотой дискретизации*. Стандартом при преобразовании звукового сигнала к цифровому виду считают частоту 44 100 Гц.

Таким образом, при дискретизации амплитуда входного сигнала делится на одинаковые промежутки (*шаг квантования*), каждый промежуток обозначается целым числом, затем через установленные частотой дискретизации промежутки времени амплитуда сигнала измеряется, и в памяти сохраняется то целое число, к которому относится измеренное значение (рис. 18.1).

При восстановлении сохраненные в массиве числа преобразуются в аналоговый сигнал (рис. 18.2). На рис. 18.2 хорошо видно, что даже после сглаживания «ступенек» при помощи фильтров восстановленный сигнал все равно отличается от исходного. Для того чтобы вывести это отличие за границы различимого человеческим слухом, диапазон квантования разбивают на гораздо большее количество уровней. Стандартными значениями являются 8- (количество уровней квантования равно 2^8) и 16-битный звук (количество уровней равно 2^{16}). Для сохранения и записи цифрового звука используется 16-битное квантование и частота дискретизации 44,1 кГц. Именно в таком качестве диск записывается на аудио-CD. Несмотря на то, что частота дискретизации в 44,1 кГц вполне достаточна для восстановления звука без искажений, для любителей большей точности дискретизация может иметь еще большее значение (это зависит от характеристик звуковой

платы, сегодня на некоторых звуковых платах уже доступны значения 48, 96, 192 и даже 384 кГц).

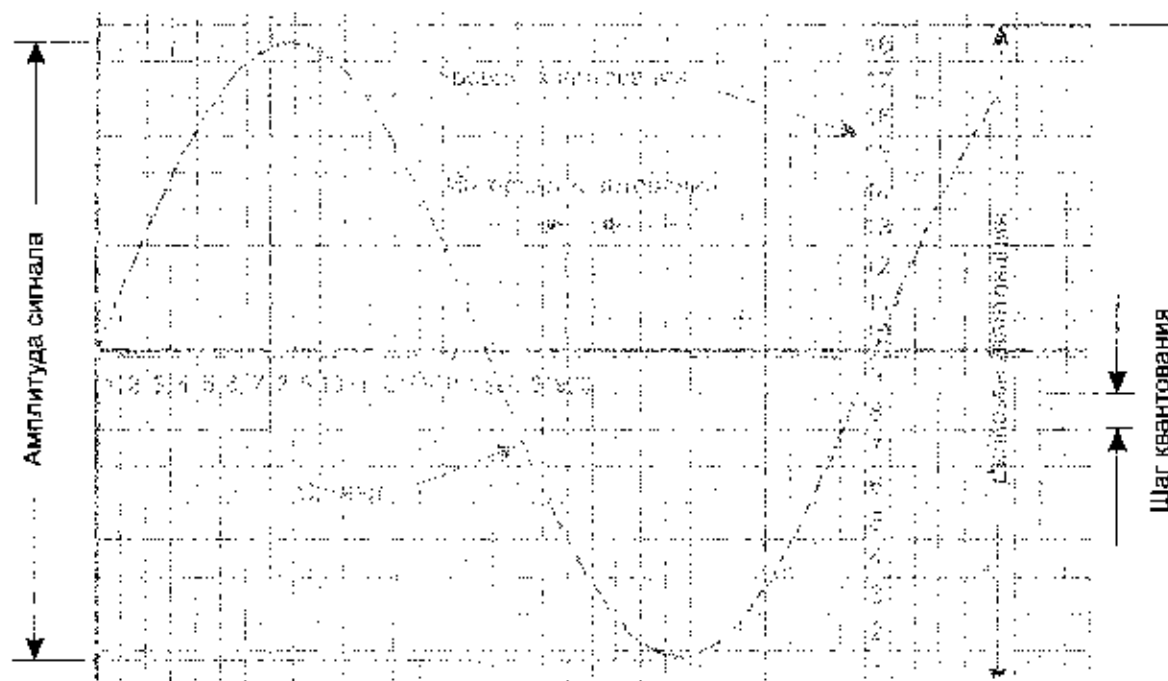


Рис. 18.1. Процесс дискретизации

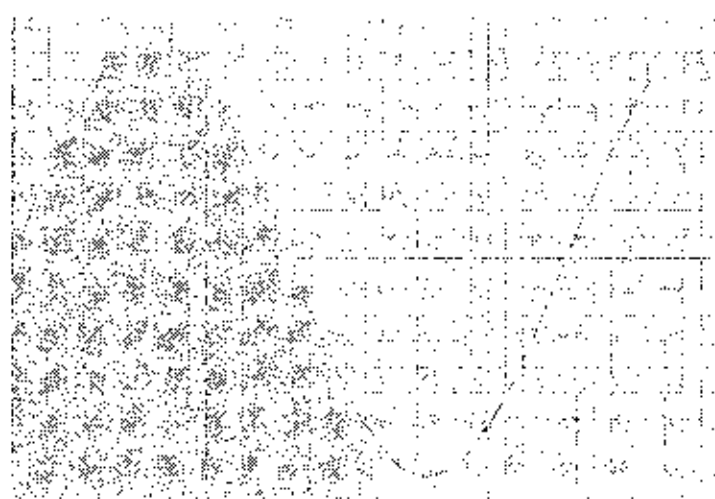


Рис. 18.2. Восстановление сигнала

После того как оцифровка звука состоялась, его записывают в один из имеющихся стандартных форматов сохранения цифрового необработанного звука (обычно это формат WAV).

Синтезированный звук

Синтезированный звук имеет совершенно другую природу по сравнению с оцифрованным. Он генерируется сразу из цифровых данных.

Для воспроизведения синтезированного звука применяют специальное устройство, *MIDI-синтезатор* (Musical Instrument Digital Interface). MIDI-синтезатор имеет встроенный банк данных, в котором записано, какой звук и какой длитель-

пости должен звучать при поступлении на вход синтезатора того или иного кода (MIDI-команды). В банк данных MIDI-синтезатора записана имитация звучания множества инструментов. Кроме того, там могут храниться реальные звуки, записанные в цифровой форме (свистки, хлопки в ладоши, удары барабанов или сирены). Таким образом, на вход MIDI-синтезатора на самом деле подается закодированная партитура музыкального произведения, а на выходе MIDI-синтезатор воспроизводит довольно близкое к реальному оркестру звучание, например, караоке-проигрыватель — это и есть MIDI-синтезатор. Формат MIDI-файлов предусматривает послоговое встраивание слов песни для их синхронного воспроизведения вместе с музыкой.

18.2.2. Сжатие звука

Файлы оцифрованного звука имеют довольно большой объем, так, на CD объемом 720 Мбайт удастся записать всего 22 композиции. Для того чтобы уменьшить объем файлов, которые хранятся на диске, применяют сжатие данных. Сжатие бывает двух типов:

- *Кодирование без потери качества (lossless coding)*. В этом случае при декодировании мы получаем абсолютно ту же последовательность чисел, которая подавалась на вход устройства кодирования. Наиболее распространенными форматами сжатия без потерь являются свободные форматы FLAC (Free Lossless Audio Codec) — это наиболее популярный и широко используемый формат, Musepack, True Audio и WavPack, а также закрытый формат Monkey's Audio (файлы с расширением .ape). При кодировании без потерь качества удастся сжать музыкальные файлы на 20–30 %.
- *Кодирование с потерей качества (lossy coding)*. Такое кодирование выполняется за счет специальных приемов освобождения музыкальной композиции от «непужных» фрагментов. Непужными эти фрагменты считаются по той причине, что на некоторых частотах внутри воспринимаемого частотного диапазона (20 Гц–20 кГц) человеческий слух имеет «провалы», в которых он практически не слышит наличия или отсутствия определенных частотных сочетаний. Программы кодирования с потерями отыскивают такие сочетания и удаляют их из композиции. Благодаря этой операции достигается не только меньший размер файла, но и большая однородность сжимаемого материала, который затем ужимается стандартной программой сжатия до размеров, в 10–15 раз меньших, чем размер исходной композиции. Наиболее распространенными форматами сжатия с потерями являются свободный формат Ogg Vorbis и коммерческий формат MP3, а также форматы WMA (Microsoft) и AAC (Apple).

18.2.3. Устройства для получения и воспроизведения звука

Для того чтобы получать звуковую информацию из внешнего мира и сохранять ее в виде цифровых данных, существуют специальные устройства, называемые

звуковыми картами, или звуковыми адаптерами. В современном звуковом адаптере обычно совмещены функции как оцифровки и воспроизведения аналогового звука, так и простой MIDI-синтезатор. Источниками аналогового сигнала, подаваемого на вход звуковой карты, служит микрофон, выходной разъем воспроизводящего устройства (магнитофона, цифрового проигрывателя, радиоприемника, CD- или DVD-проигрывателя) либо сигнал с CD- или DVD-привода, входящего в состав компьютера. Кроме аналогового сигнала на звуковую карту может подаваться цифровой сигнал от других устройств, например, от видеомэгнитофона или видекамеры. У многих звуковых адаптеров есть специальный разъем для подключения MIDI-клавиатуры (игрового джойстика). Выходной сигнал со звуковой платы через специальные разъемы выводится на звуковые колонки.

На рис. 18.3 приведен эскиз современной звуковой платы.

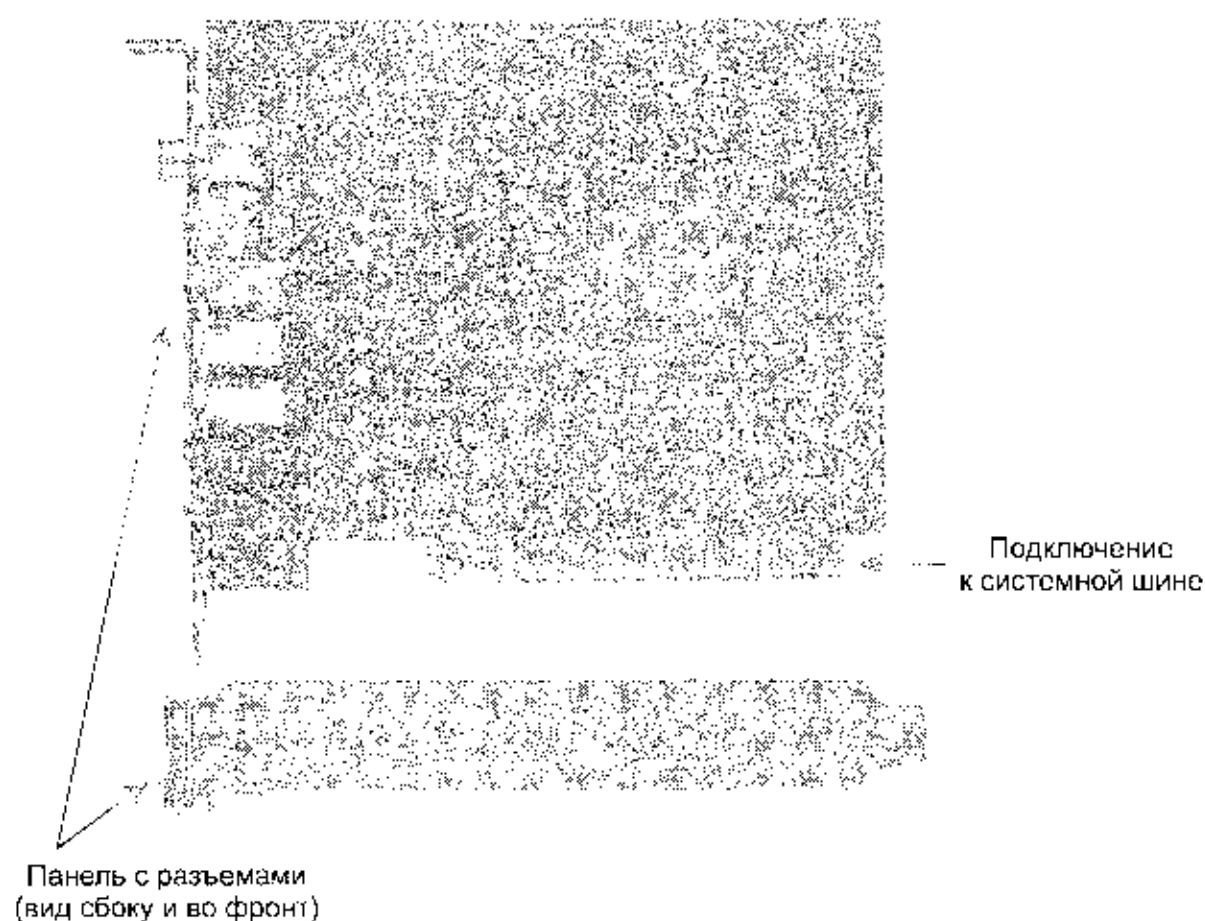


Рис. 18.3. Звуковая плата

Поскольку возможности интеграции различных цифровых устройств между собой, стандарты соединений и протоколов постоянно расширяются, сегодня можно встретить звуковой адаптер с гораздо большим количеством различных разъемов, так же как и звуковой адаптер, подключаемый к компьютеру в качестве внешнего устройства через шину USB.

18.2.4. Программные продукты для обработки и воспроизведения звука

Запись звука

Современные операционные системы обычно имеют в своем составе простые программные продукты для записи и воспроизведения цифрового звука. Так, для записи с микрофонного и линейного входов звуковой платы в Windows существует программа *Заукозапись* (Sound Recorder), а в Linux — программа *KRec* (рис. 18.4).

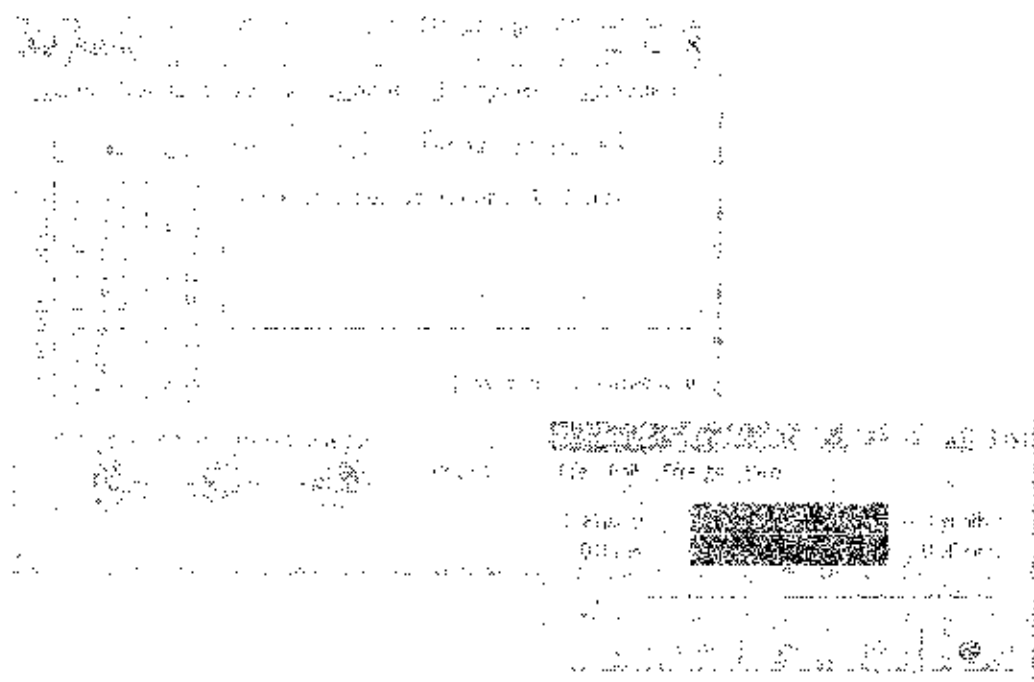


Рис. 18.4. Программы звукозаписи

Та и другая программы позволяют осуществлять запись звука, поданного на вход звуковой платы, и сохранения его в исходном формате (WAV) или в форматах сжатого звука, причем как с потерями, так и без. Кроме того, обе эти программы поддерживают простейший монтаж звуковых фрагментов (вырезание и вставку), а также добавление некоторых звуковых эффектов.

Управление звуковыми каналами (микширование)

Поскольку современные звуковые платы имеют множество входов и выходов, одновременно в работе платы могут участвовать несколько входных и выходных каналов, то есть существует возможность осуществлять запись или воспроизведение одновременно нескольких источников. Например, можно одновременно начитывать текст в микрофон и получать фоновую музыку с линейного выхода, записывая получающуюся комбинацию в звуковой файл на диск при помощи программы звукозаписи. Для того чтобы управлять включением-отключением и отдельно регулировать громкость воспроизведения или чувствительность записи в отдельных каналах, существуют специальные программы микширования звука. На рис. 18.5 показаны такие программы для Windows (вверху) и Linux (внизу).

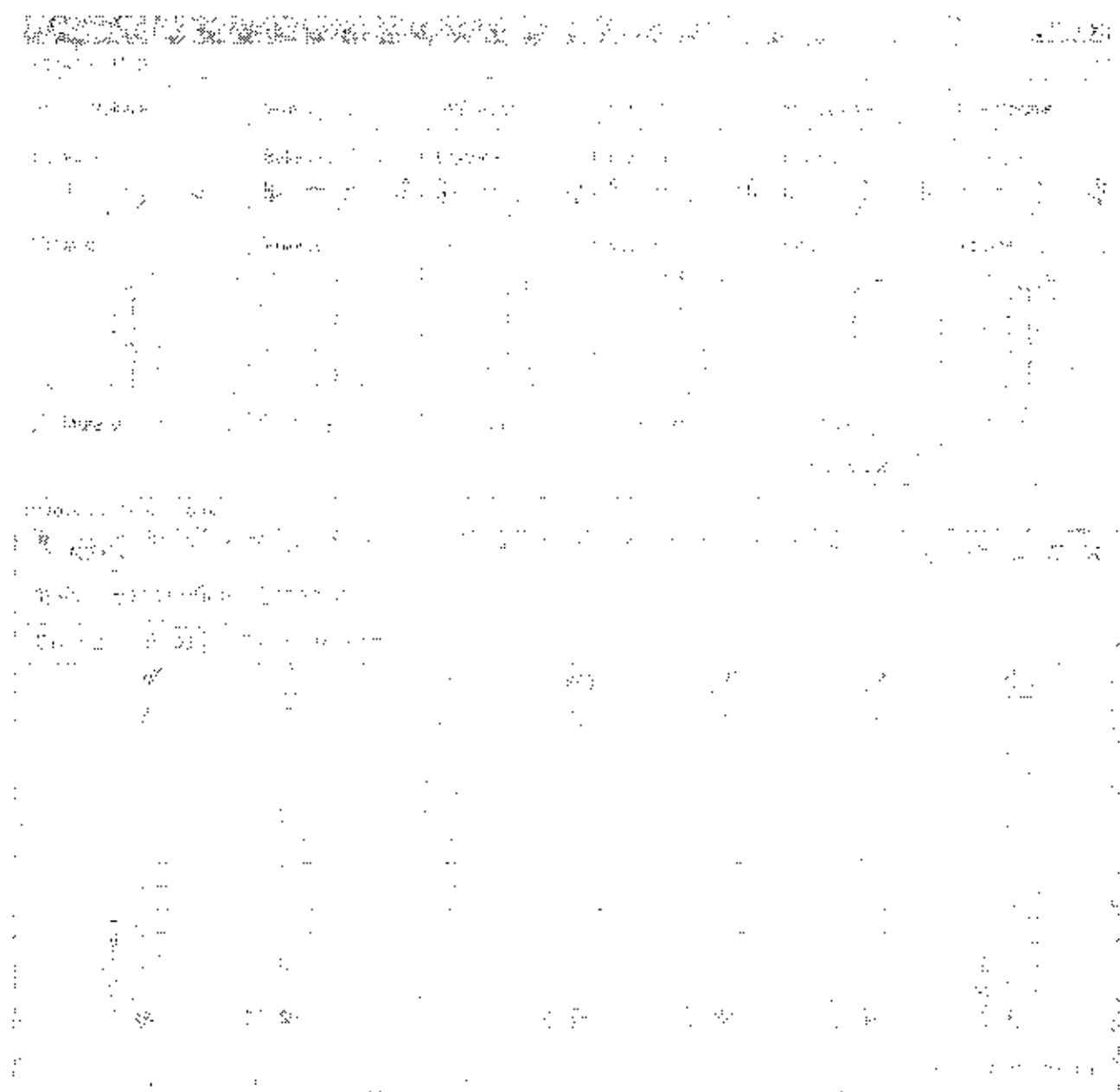


Рис. 18.5. Программы-миксеры

Воспроизведение оцифрованного звука

Для воспроизведения оцифрованного звука в различных форматах (WAV, OGG, MP3) существуют специальные программные продукты, называемые *медиапроигрывателями*. Если раньше воспроизведение звука и видео разделялось функционально в разных программах, то в современных условиях производители ПО стараются объединить функции воспроизведения звука и видео в универсальных программах. Существует целый ряд таких программ, позволяющих воспроизводить как звуковые, так и видеофайлы в различных форматах. В Windows это входящий в состав операционной системы медиапроигрыватель и различные медиапроигрыватели сторонних производителей ПО: RealPlayer, QuickTime Player, BSPlayer и другие. Окна перечисленных программ воспроизведения аудио- и видеозаписей показаны на рис. 18.6.

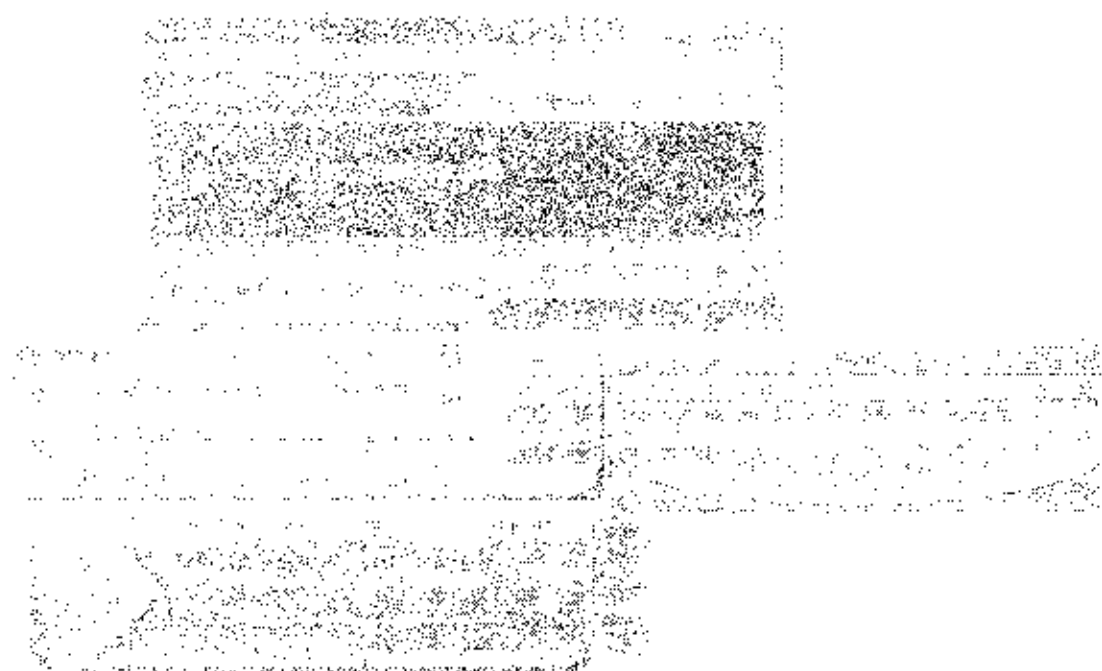


Рис. 18.6. Окна различных программ воспроизведения цифрового звука в Windows

Из программ проигрывания музыкальных файлов самым широким набором сервисных функций обладает Windows Media Player.

В операционной системе Linux количество универсальных программ для воспроизведения звука и видео еще больше, чем в Windows: достаточно назвать только наиболее распространенные: Kaffeine, Amarok, VLC, JuK, KsCD. Как эти, так и многие другие программы воспроизведения аудиофайлов, входят в состав дистрибутивов Alt Linux Desktop и Школьный Линукс (рис. 18.7).

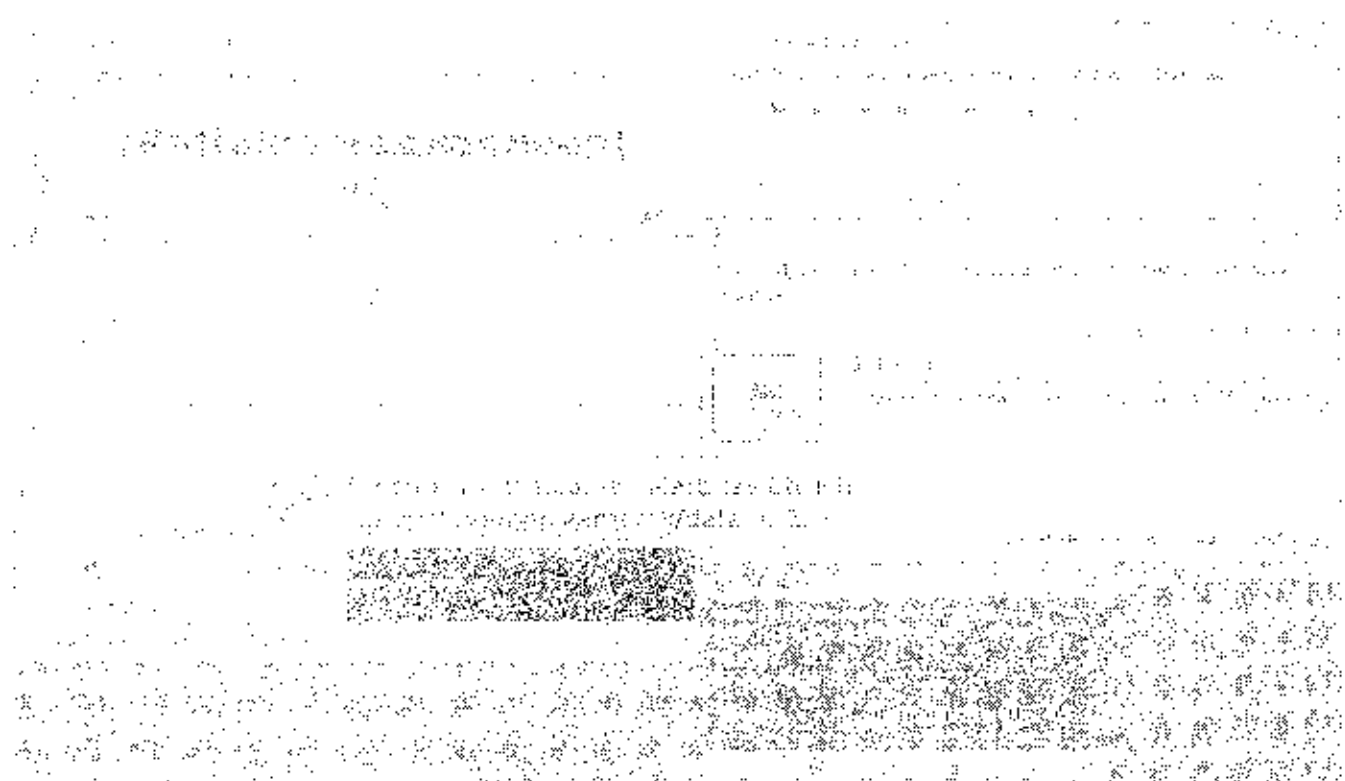


Рис. 18.7. Медиапроигрыватели в Linux

Редактирование оцифрованного звука

Оцифровка звуковой информации и представление ее в виде файлов на диске произвели настоящую революцию в среде любителей музыки: процессы раздельной записи каналов, сведения звука, добавления к готовой записи инструментов и голоса, различного рода звуковые эффекты и очистка записи от шумов — все то, что еще полтора десятилетия назад можно было сделать только в профессиональной студии на очень дорогом оборудовании, неожиданно стало доступно любому человеку, у которого есть компьютер. Еще 20 лет назад, когда речь шла о звукомонтаже, главным был вопрос оборудования, сегодня же главным стало личное мастерство звукооператора, поскольку с оцифрованным звуком можно сделать все, что угодно.

Для редактирования цифрового звука существует ряд коммерческих программ: Sound Forge, Cool Edit, WaveLab. Мы рассмотрим уникальную программу Audacity, окно которой показано на рис. 18.8. Уникальна она в первую очередь тем, что является свободной, распространяется бесплатно, а по своим возможностям во многом превосходит свои платные аналоги.

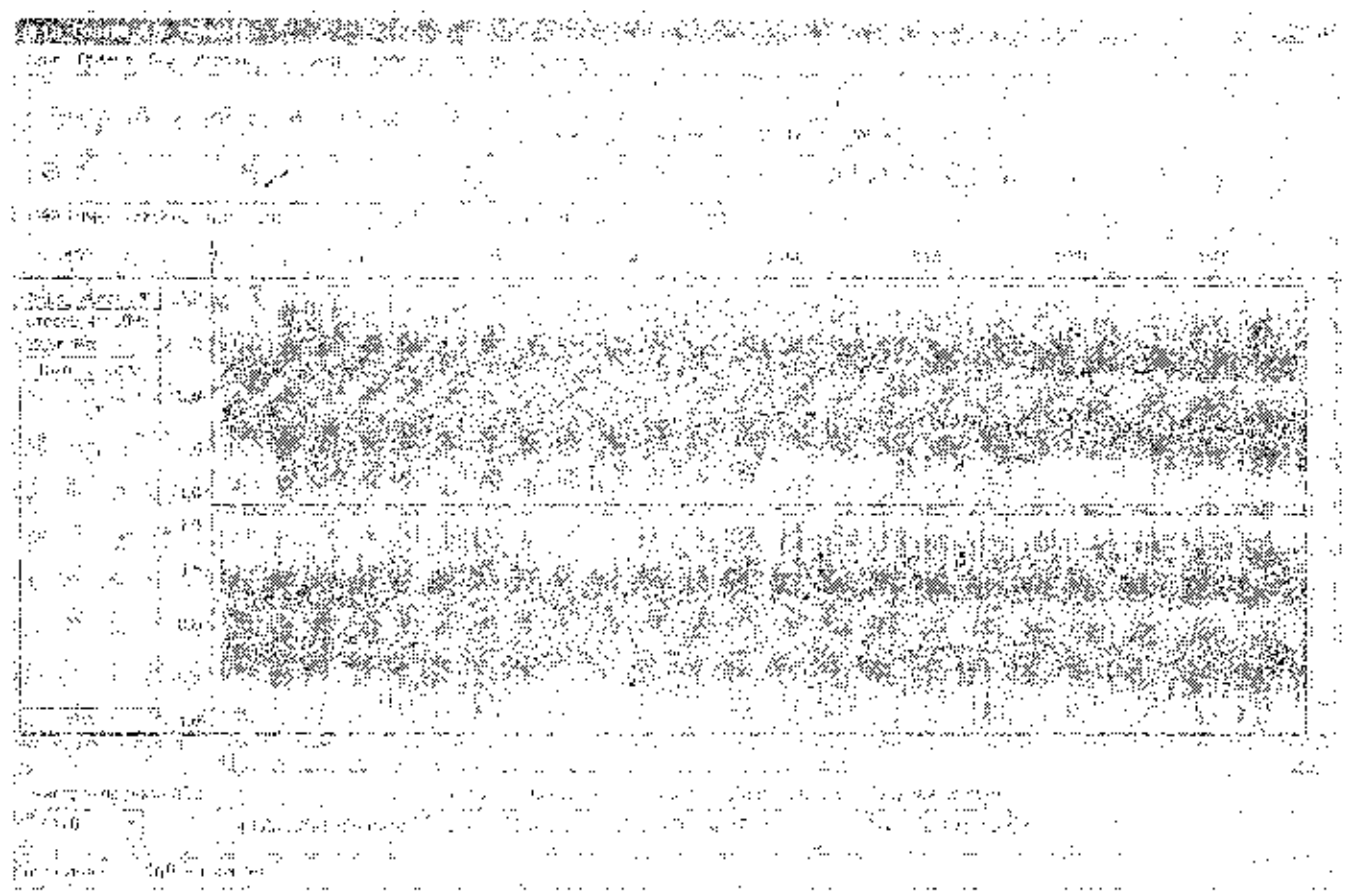


Рис. 18.8. Программа редактирования звуковых файлов Audacity

Audacity — программа профессионального уровня, позволяющая проделывать со звуком любые трюки, начиная от простейшего редактирования фрагментов и сведения дорожек (например, к оцифрованному звуку можно добавлять звучание инструментов MIDI-синтезатора) и заканчивая применением к звуку любых известных на сегодняшний день звуковых эффектов и точечным редактированием звуковой огибающей или удалением из стереозаписи вокальной составляющей.

При оформлении мультимедийных документов (например, создании мультимедийных презентаций или анимационных клипов) этот инструмент может оказать неоценимую помощь в генерации соответствующей звуковой составляющей документа.

Работа с синтезированным звуком

Файлы, содержащие данные с синтезированным звуком, получают в результате либо проигрывания этого звука на MIDI-клавиатуре с последующим преобразованием в цифровые данные, либо прямого ввода звуков. То и другое делается с помощью специальных MIDI-редакторов. MIDI-редакторы позволяют создавать, редактировать и сохранять последовательности MIDI-команд, как показано на рис. 18.9.

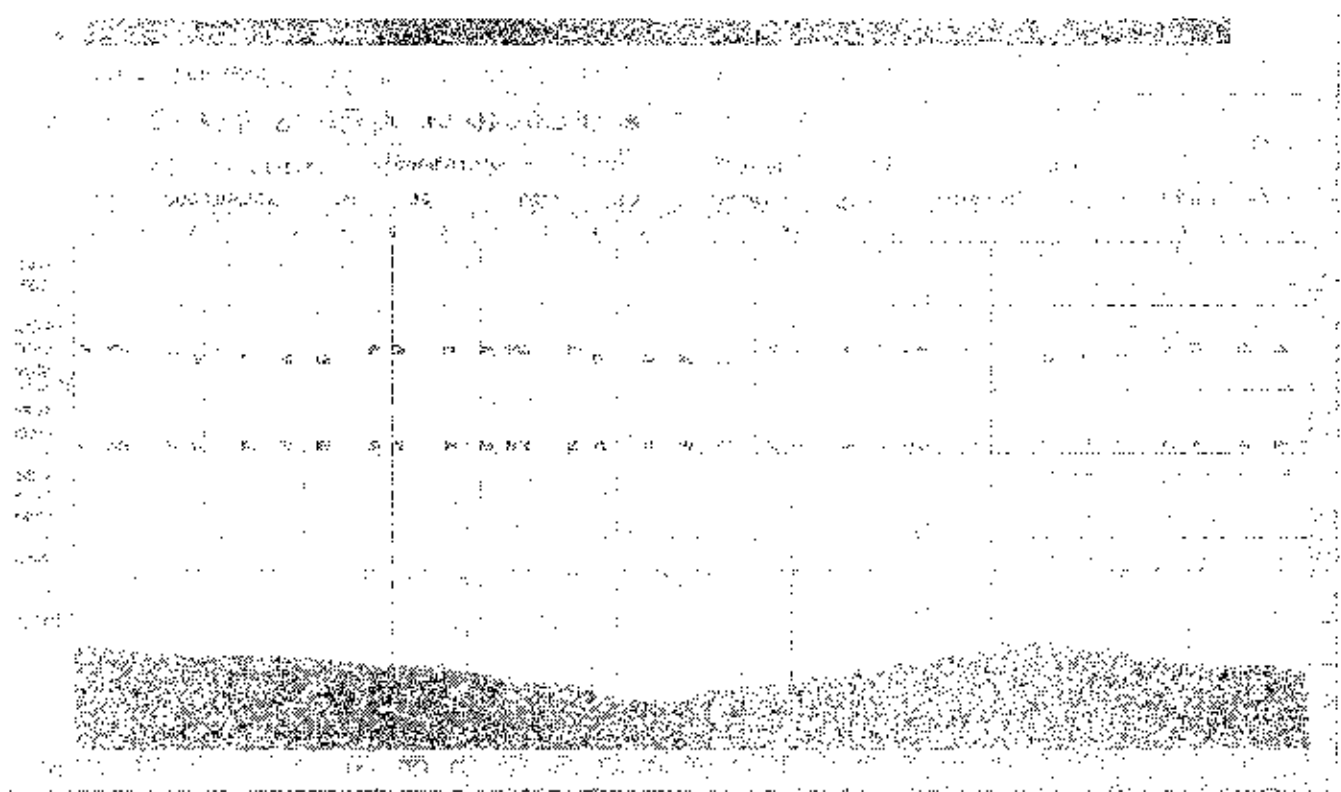


Рис. 18.9. MIDI-редактирование

У MIDI-редактора есть симулятор клавиатуры и встроенный метроном; можно «наиграть» нужные ноты в нужном темпе, а потом еще и подправить их, регулируя в редакторе длительность или высоту. Примерно таким же образом происходит запись MIDI-последовательности, принимаемой с MIDI-клавиатуры. При этом надо помнить, что MIDI — это многоголосый формат, поэтому используя MIDI-редактор, можно записать одну за другой партии всех инструментов, а затем воспроизводить их звучание одновременно, как при оркестровом исполнении.

Еще одна интересная особенность практически всех MIDI-редакторов — возможность опубликовать синтезированные композиции в виде нотной записи (рис. 18.10).

Среди множества MIDI-редакторов можно выделить коммерческие программы Cubase, Logic Audio, Cakewalk, а также свободные Rosegarden Studio, Seq246 Jazz++ MIDI Sequencer, SoftWerk.

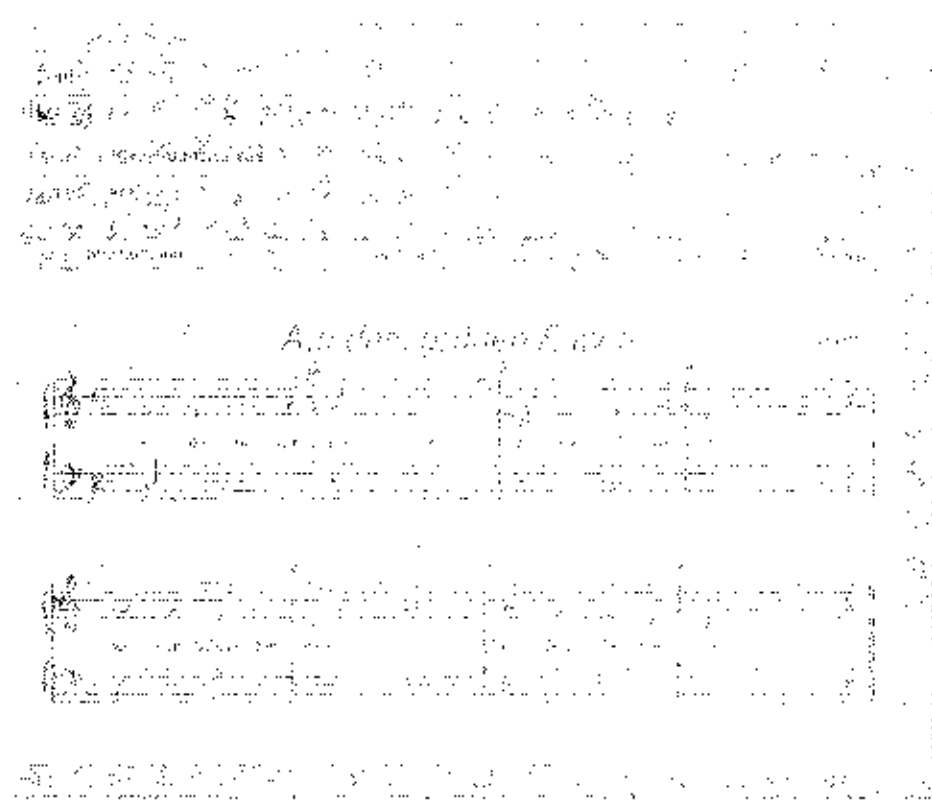


Рис. 18.10. Музыкальная композиция в виде нот

18.3. Изображения

18.3.1. Цифровые форматы изображений

Известно, что экран компьютера состоит из точек, поэтому любое изображение, которое появляется на экране компьютера, состоит из точек. Независимо от того, в каком формате хранится файл изображения, на экране компьютера он всегда представляется в виде матрицы точек разного цвета. То есть изображение на экране компьютера всегда имеет растровый характер. Деление изображений на векторные и растровые определяется форматом хранения изображения и алгоритмом его отображения на экране.

Векторные форматы

В векторном формате файл хранит не само изображение, а его описание, то есть этот файл не содержит массива точек.

Пример. Если мы нарисуем в векторном графическом редакторе эллипс и сохраним данный рисунок в файле, то в файл запишутся данные о том, какая фигура нарисована (эллипс), где находится ее центр, каковы ее размеры, толщина и цвет линии, толщина и цвет заливки и другая информация, позволяющая однозначно воспроизвести эту картинку в любое время.

Хранить рисунки в векторных форматах очень удобно: векторные форматы очень экономны с точки зрения сохранения в файле и передачи через сетевые со-

единения сравнительно простых рисунков и чертежей (векторных примитивов). В случае, когда в векторный формат переводится сложный рисунок (например, фотография пейзажа с максимальным приближением изображения к реалистичному), то размер векторного файла может быть даже больше, чем размер растрового.

У векторной графики есть одно серьезное преимущество, которое делает этот формат идеальным для хранения чертежей и схем: на качество отображения никак не влияет масштаб отображения рисунка. Если вы увеличите или уменьшите векторный рисунок в 10 раз и затем сохраните в файле, то после открытия файла вы получите изображение того же качества, что и исходное. Это выгодно отличает векторную графику от растровой, для которой аналогичная последовательность операций (уменьшение в 10 раз, сохранение, открытие, увеличение в 10 раз) приведет к необратимой потере качества изображения.

Растровые форматы

Растровая графика используется во многих областях повседневной деятельности при переводе сложных массивов данных в графическое представление. Растровые форматы изображений хранят в себе каждую точку изображения в том масштабе, в котором изображение было зафиксировано (сфотографировано, отсканировано, создано в растровом графическом редакторе). Хранение каждой точки изображения имеет определенные плюсы: если снимать фотографию при очень больших разрешениях и размере изображения, то можно получить высокое качество и широкие возможности в отношении правки такого изображения.

Растровые изображения крайне важны, но они требуют много места. Поскольку современные дисплеи передают множество цветовых оттенков, каждый пиксел принято интерпретировать в виде 24-битного числа, в котором компоненты красного, зеленого и голубого цветов занимают по 8 бит каждый. Такой 24-битный пиксел может отображать 224, или примерно 16,78 миллиона цветов. Так, изображение с разрешением 512 × 512 пикселей будет занимать 786 432 байт, а изображению размером 1024 × 1024 пикселей потребуется 3 145 728 байт.

Преимуществом же растровых файлов является точность отображения и возможность полипиксельного манипулирования для создания различных визуальных эффектов, от простейших (повышение резкости или нанесение дымки) до весьма замысловатых, например, создание псевдообъемного изображения из плоской фотографии.

18.3.2. Сжатие изображений

О сжатии векторных форматов мы уже говорили: практически все файлы векторных программ представляют собой набор текстовых примитивов, сжатый архиватором zip. Даже если применить к этим файлам самые современные и эффективные способы сжатия, можно добиться выигрыша процентов 10, но при этом значительно (особенно для больших файлов) увеличить время открытия и сохранения.

Текст можно сжимать за счет того, что способ кодирования текста изначально избыточен. Устраняя эту избыточность, можно закодировать ту же самую инфор-

мацию в меньшем количестве данных. Изображение для компьютера не обладает избыточностью: каждый пиксел находится на своем месте и имеет свой код.

Хотя компьютер не может распознать в изображении избыточность, зато можно создать алгоритмы, позволяющие распознавать и выделять в изображении информацию менее важную и плохо воспринимаемую при его рассмотрении человеком.

При сжатии с потерями происходит сжатие путем удаления несущественной информации. Изображение можно сжать с потерями, удалив несущественную информацию, даже если в нем нет избыточности.

Не все изображения сжимаются одинаково хорошо любыми методами. Рассмотрим несколько типов изображений.

- ❑ *Двухуровневое, или монохроматическое, изображение.* В этом случае все пикселы могут иметь только два значения, которые обычно называют черным (двоичная единица, или основной цвет) и белым (двоичный ноль, или цвет фона).
- ❑ *Полутонное изображение.* Каждый пиксел такого изображения может иметь $2n$ значений от 0 до $2n - 1$, обозначающих одну из $2n$ градаций серого (или иного) цвета.
- ❑ *Цветное изображение.* Существует несколько методов задания цвета, но в каждом из них участвуют три параметра. Следовательно, цветной пиксел состоит из трех частей (обычно из трех байтов). Типичными цветовыми моделями являются RGB, HLS и CMYK.
- ❑ *Изображение с непрерывным тоном.* Этот тип изображений может иметь много схожих цветов (или полутонов). Когда соседние пикселы отличаются всего на единицу, глазу практически невозможно различить их цвета. Обычно они получаются при съемке на цифровую фотокамеру или при сканировании фотографий или рисунков.
- ❑ *Дискретно-тоновое, или синтетическое, изображение.* Обычно это изображение получают искусственным путем. В нем может быть всего несколько или много цветов, но как правило, в нем нет шумов и пятен, характерных для естественного изображения. Примерами таких изображений могут служить фотографии искусственных объектов, машин, механизмов, страницы текста, карты, рисунки или изображения на дисплее компьютера.
- ❑ *Изображения, подобные моментальным снимкам мультфильма.* Это цветные изображения, в которых присутствуют большие области одного цвета. При этом соприкасающиеся области могут весьма различаться по своему цвету. Данное свойство можно использовать для лучшего сжатия.

Каждому типу изображений присуща определенная избыточность, но все они избыточны по-разному. Поэтому трудно создать метод, позволяющий одинаково хорошо сжимать любые типы изображений. Существуют разные методы для сжатия двухуровневых, непрерывно-тоновых и дискретно-тоновых изображений.

Изображения, сжатые различными способами, затем сохраняют в файлах различных форматов (отличающихся, как правило, расширениями).

Формат — способ организации информации в файле.

Поскольку способ организации мультимедийной информации непосредственно связан со способом ее сжатия с целью хранения или передачи, методы сжатия также часто называют форматами.

Формат JPEG

Формат сжатия изображений JPEG (Joint Photographic Experts Group) является сегодня наиболее распространенным, причем его популярность настолько высока, что его часто применяют для сжатия изображений, совершенно для этого не предназначенных (например, изображений, где большие площади занимает один цвет); результат обычно бывает неудовлетворительным. Изображения, сжатые таким методом, обычно хранятся в файлах с расширением `.jpg`. Свойства формата JPEG:

- высокий коэффициент сжатия;
- большое число параметров настройки сжатия;
- хорошие результаты для любых типов непрерывно-тоновых изображений;
- достаточно эффективен и не слишком сложен;
- несколько режимов сжатия, между которыми можно выбирать.

Форматы GIF, PNG и TIF

На рис. 18.11 представлено изображение в разных форматах сжатия. Из сравнения этих изображений становится ясно, почему сжатие в формате JPEG так хорошо подходит к фотографиям (причем чем больше там деталей, тем лучше оно работает), не годится при сжатии изображений с большими однотонными участками.

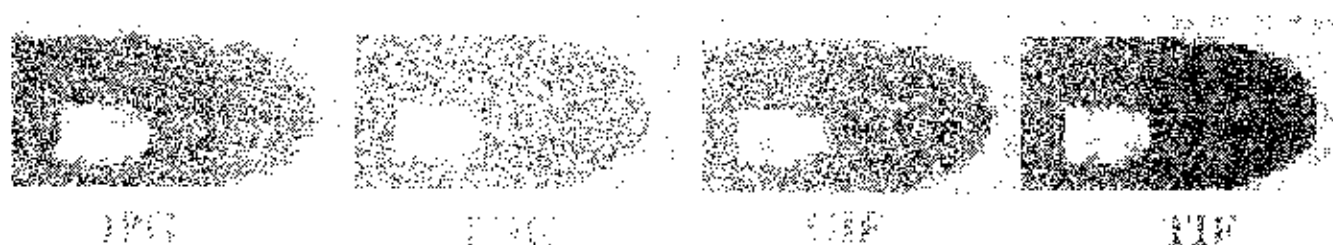


Рис. 18.11. Разные форматы сжатия

Из-за своего внутреннего алгоритма JPEG допускает ошибки при кодировании ровного цвета. Не лишен погрешности и формат TIF, который при сохранении изображения со сжатием искажает фон. А вот форматы GIF и PNG, будучи примерно в два раза менее эффективными в отношении сжатия, чем JPEG, оказываются идеальными для такого рода изображений. По этой причине при оформлении дизайна веб-сайтов в основном используют форматы GIF и PNG, для хранения и пересылки фотографий — формат JPEG, а для хранения рисунков высокой точности воспроизведения — формат TIF (но без сжатия).

18.3.3. Получение изображений

Для получения изображений есть несколько способов.

Первый способ — можно создать изображение вручную в специальной компьютерной программе, графическом редакторе. Если мы говорим о векторных форматах, то таким образом получают подавляющее большинство векторных изображений — они рисуются от руки, с использованием мыши или специальных приспособлений, позволяющих рисовать на экране компьютера привычным для нас способом, к примеру, это может быть графический планшет в комплекте со специальным пером. Сегодня уже не редкость графические планшеты, которые одновременно являются экранами — на них можно рисовать так же, как на листе бумаги. Векторные изображения обычно получают, растягивая графические примитивы и линии по экрану. Растровые изображения рисуют так же, как мы привыкли это делать — при помощи пера или карандаша.

Второй способ — сканирование. Сканер переводит изображение документа в цифровую форму с высоким разрешением. Сканированные документы всегда представляют собой растровые изображения большого размера.

Третий способ — получение изображений с цифровых фотоаппаратов или видеокамер.

Четвертый способ — получение векторных изображений из растровых с помощью специальных программ распознавания контуров и фоновой заливки.

18.3.4. Программы для работы с изображениями

Программы для работы с векторными изображениями

Одним из самых известных и одновременно наиболее серьезных графических пакетов для работы с векторной графикой является Corel Draw (рис. 18.12).

Истинная мощь этой программы раскрывается не сразу: кроме стандартных инструментов она позволяет реализовать множество эффектов, преобразовывать непосредственно во время редактирования векторные объекты в растровые, выполнять различные манипуляции с обеими типами объектов в одном документе. Внимательное и детальное освоение этого пакета откроет дополнительные возможности, касающиеся мощного инструмента автоматизации операций и расчетного аппарата, позволяющего просчитывать многие элементы сплайновой графики. Этот пакет применяется в основном для выполнения дизайнерских проектов, включая полиграфические проекты высокой степени сложности.

Вторым по популярности после Corel Draw можно считать пакет Adobe Illustrator. Это также профессиональная дизайнерская программа, обладающая большим набором возможностей и весьма дорогая.

Для индивидуального использования (например, для оформления иллюстраций к мультимедийным презентациям) вполне подойдет менее дорогой пакет, Corel Xara, который в своем варианте для Linux является свободной и бесплатной программой (рис. 18.13).

Эта программа предназначена для создания красивой и праздничной векторной графики. Хотя и лишая некоторых возможностей Corel Draw, эта облегченная

программа позволяет тем не менее получать рисунки очень высокого качества и имеет хорошие возможности создания псевдообъемных изображений (иллюстрацией чему служит рис. 18.13).

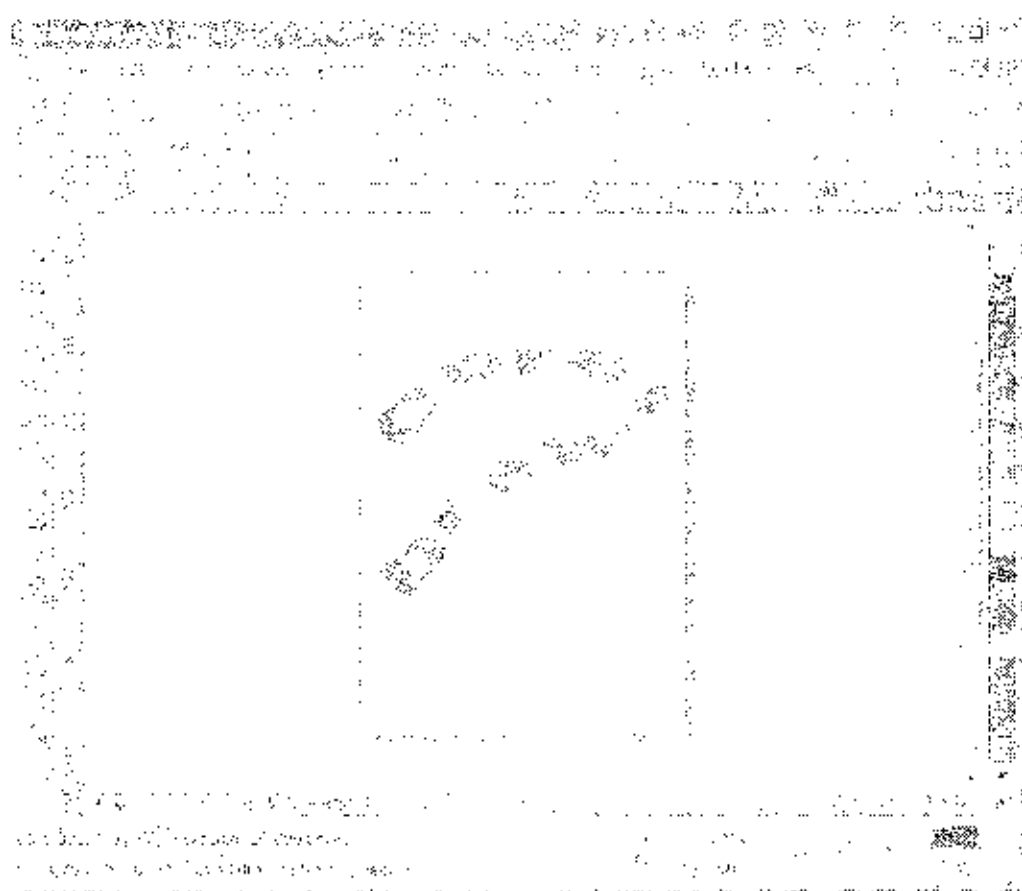


Рис. 18.12. Corel Draw

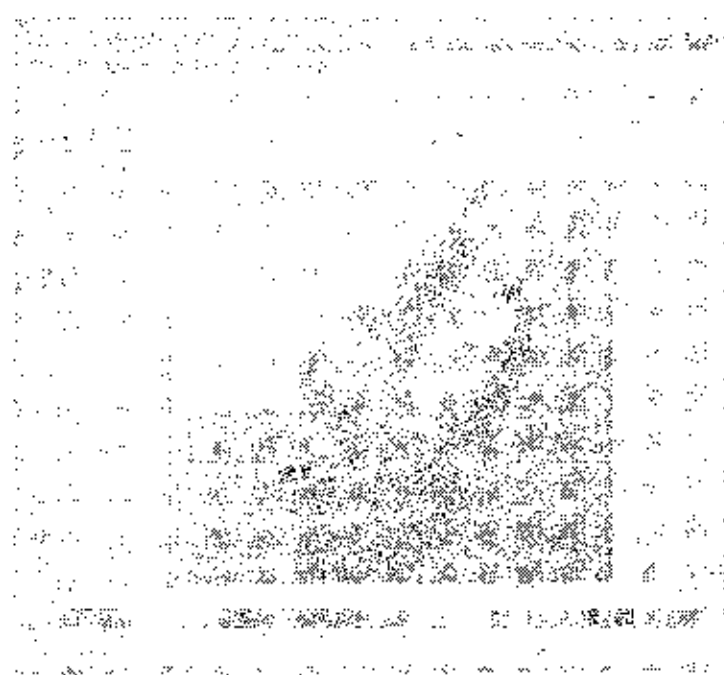


Рис. 18.13. Xara Xtreme для Linux

Для создания векторных блок-схем и диаграмм, которые тоже очень часто присутствуют в различного рода презентациях и мультимедийных документах, можно

использовать встроенные средства векторной графики, которые входят в состав как Microsoft Office, так и OpenOffice.org. Кроме того, существует специальный пакет, Microsoft Visio, предназначенный для создания схем и диаграмм различного назначения (рис. 18.14). Среди свободных программ ближе всего по функциональности к этому пакету программа Dia.

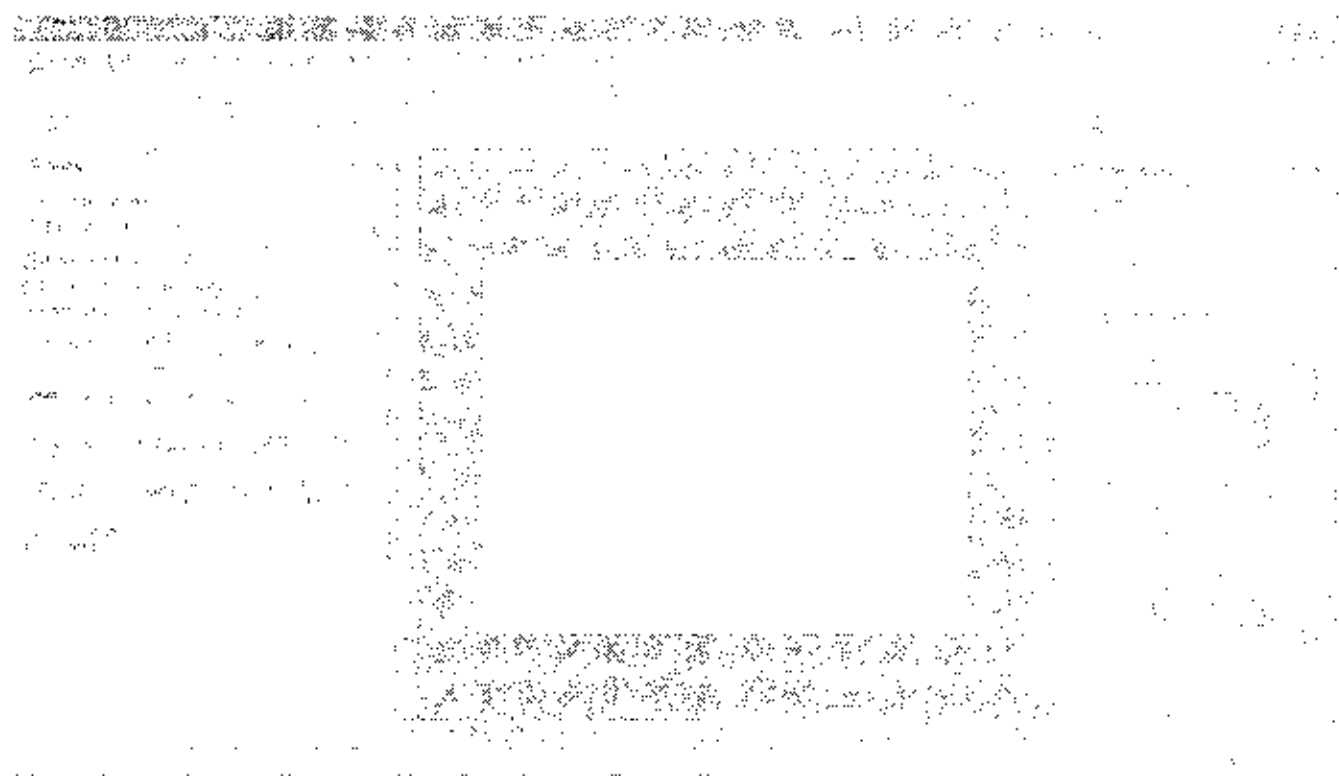


Рис. 18.14. Microsoft Visio

Программы для работы с растровыми изображениями

Количество программ для работы с растровыми изображениями очень велико, перечислить даже самые популярные не представляется возможности. По этой причине мы остановимся только на нескольких самых простых и на двух самых совершенных программах этого типа.

К простым программам обработки растровых изображений относятся программа Paint, входящая в состав операционной системы Microsoft Windows, и свободная программа ColourPaint, входящая в состав многих дистрибутивов Linux. Обе эти программы поддерживают простые операции по редактированию растровых изображений (вырезание, копирование, вставка фрагментов изображения, закрашивание цветом или узором, рисование пером и кистью, вставка примитивных геометрических фигур и текста) и сохранению их в наиболее распространенных растровых форматах. Обе эти программы скорее пригодны для развлечения, чем для создания сколько-нибудь серьезных графических композиций, и уж совсем непригодны для обработки сложных цифровых растровых изображений, таких как фотографии и сканированные образы документов (рис. 18.15).

Для серьезной работы существуют другие программные продукты — это коммерческие пакеты Adobe Photoshop и GIMP. Если говорить о сходствах и отличиях между программами GIMP и Adobe Photoshop, то функционально программы очень схожи, но различаются набором доступных встраиваемых модулей (для

Photoshop их, безусловно, на порядок больше, чем для GIMP) и степенью возможностей автоматизации операций и программирования (здесь GIMP, конечно же, превосходит своего проприетарного собрата). Обе программы способны выполнять самые разнообразные действия с изображениями в растровых форматах: изменять цветовые схемы, выполнять разнообразные коррекционные действия, накладывать фильтры. Мощным средством является представление изображения как многослойной структуры. На рис. 18.16 показано одно и то же растровое изображение, открытое в GIMP и в Adobe Photoshop.

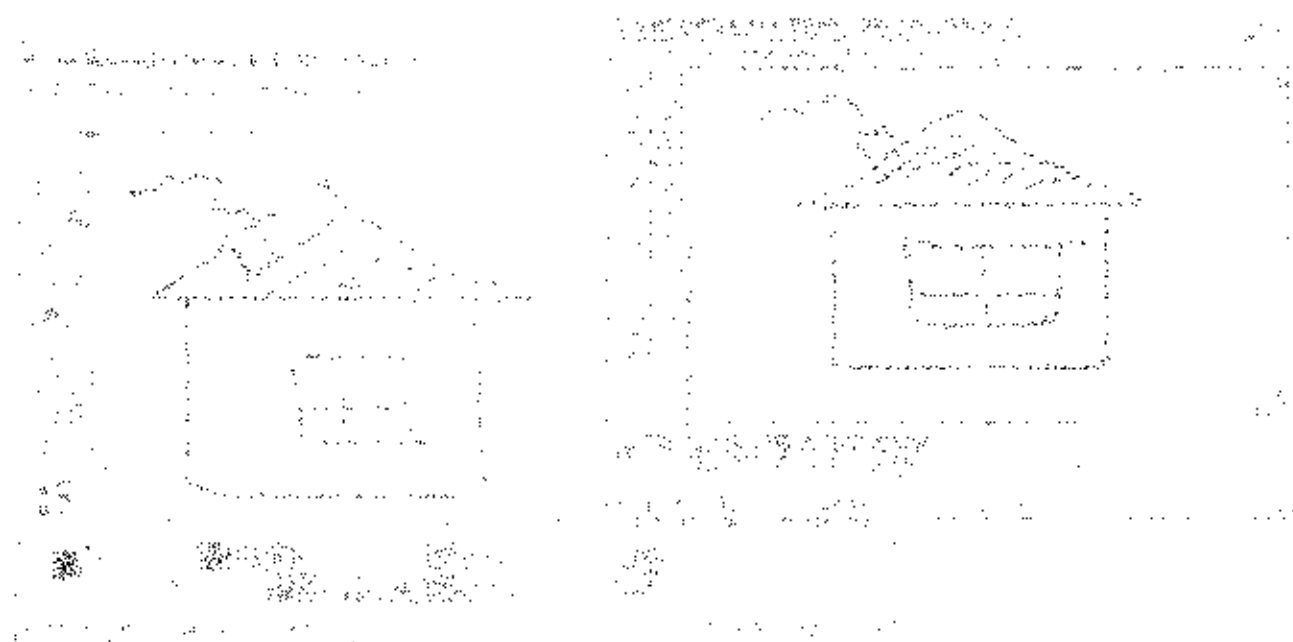


Рис. 18.15. Простые программы обработки растровых изображений

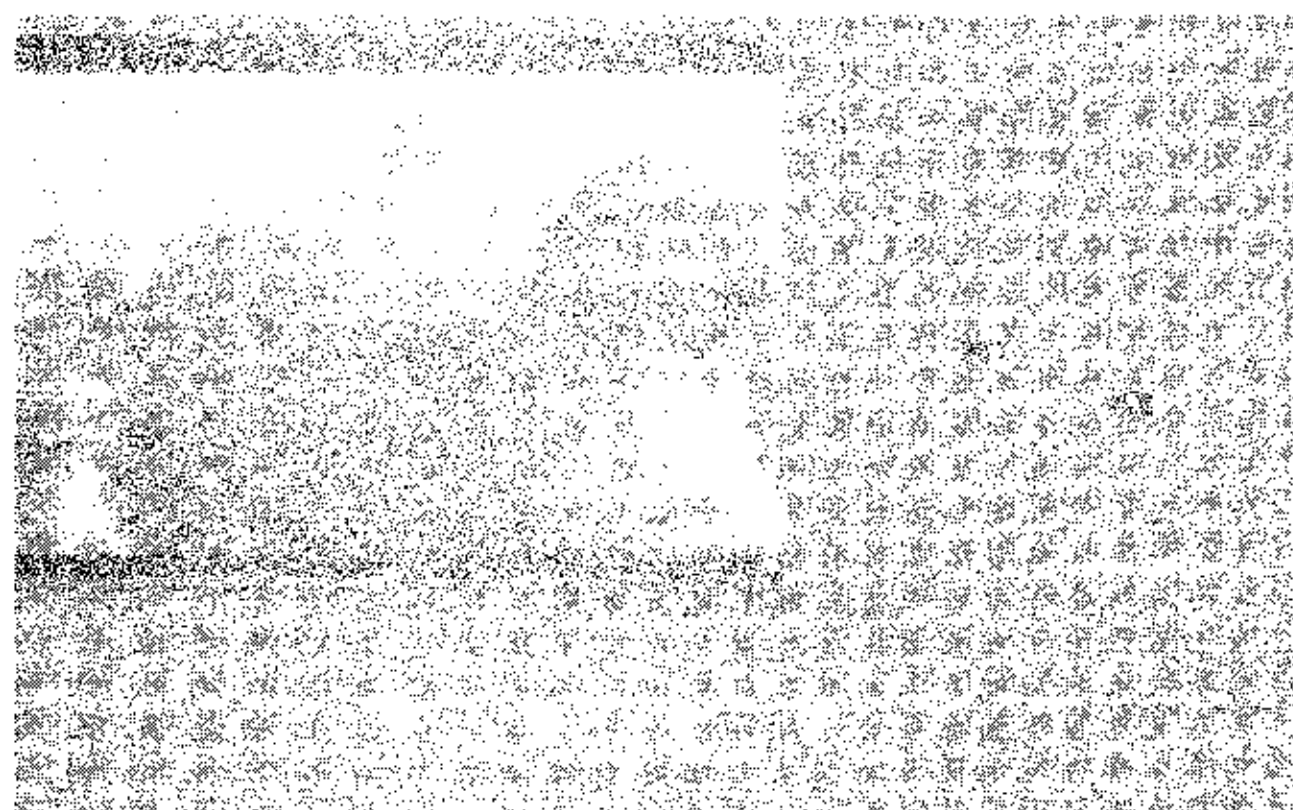


Рис. 18.16. Сложные программы обработки растровых изображений

18.4. Видео

18.4.1. Кодирование видеосигналов

При кодировании видеосигналов применяются операции, которые обеспечивают очень высокую степень сжатия.

Как кинофильм является практически потоком фотографий, демонстрируемых на экране с частотой 24 кадра в секунду, так и цифровое видео практически представляет собой последовательность цифровых кадров, то есть фотографий. Однако простой расчет показывает, что даже в случаях, когда видеофильм имеет сравнительно небольшой по сегодняшним меркам размер 512×512 пикселей, для хранения 1 секунды фильма требуется $786432 \times 24 = 18\,874\,368$ байт. Чтобы сохранить хотя бы часовой фильм, это число нужно умножить на 3600, и мы получим $67\,947\,724\,800$, то есть 68 Гбайт на один фильм. При таком объеме фильм невозможно уместить на носителе объемом 4 Гбайт (стандартный объем DVD-дисков). За счет чего же появляется возможность уменьшить размер хранимого видеофильма?

18.4.2. Сжатие видео

Для кодирования каждого кадра видеофильма применяется сжатие в формате JPEG, что даже при сравнительно небольшой потере информации может дать заметное (до 10 раз) уменьшение размера. Помимо сжатия каждого кадра, для уменьшения размер хранимого видеофильма используют некоторые дополнительные приемы.

- ❑ *Прореживание.* Кодер (программа, осуществляющая кодирование) выбирает кадры через один и записывает их в сжатый поток. Это приводит к двойному коэффициенту сжатия. Затем декодер принимает кадры и просто дублирует их два раза подряд.
- ❑ *Вычитание.* Кадр сравнивается со своим предшественником. Если разница между ними мала (всего в нескольких пикселях), то кодер кодирует только эти отличающиеся пиксели, то есть записывает в выходной поток три числа для каждого из отличающихся пикселей (его координаты и разность пикселей двух кадров).
- ❑ *Вычитание по блокам.* Этот метод является развитием предыдущего. Изображение делится на блоки пикселей, и операция вычитания производится не над парой кадров, а над каждой парой блоков в кадрах.
- ❑ *Компенсация движения.* Просмотр любого фильма наводит на мысль, что разница между последовательными кадрами мала из-за движения на сцене, перемещения камеры или в силу обеих причин. Это свойство тоже можно использовать с целью повышения коэффициента сжатия.

В результате совмещения всех этих операций при кодировании видеосигнала можно получить очень высокую степень сжатия. Все эти методы кодирования реализованы в современных форматах кодирования цифрового видео, таких как MPEG и AVI. Сам факт применения перечисленных приемов сжатия иногда ста-

повится заметен: если по той или иной причине происходит пауза в потоке передающихся кадров, видеовизображение начинает распадаться на прямоугольные зоны, при этом часть видеокадра остается необыкновенно четкой, другая же часть размывается.

Цифровое видео может быть получено тремя способами: оцифровкой имеющего кино, снятого на пленку; съемкой видео на профессиональную или любительскую видеокамеру; созданием видеофайлов в специализированных программах.

При съемке видеофильма на видеокамеру может быть два режима работы. В первом режиме видеокамера снимает, кодирует и записывает фильм во внутреннюю память (это может быть записываемый DVD-диск, flash-память или жесткий диск). Во втором режиме видеокамера подключается разъемом к цифровому входу видеокарты компьютера и напрямую передает поток видеоданных на компьютер, где специальная программа производит их кодирование и сохранение в том или ином распространенном видеоформате с заданным качеством.

18.4.3. Программы для обработки и воспроизведения видео

Программы для обработки цифрового видео можно разделить на четыре основные категории:

- ☐ захват видео;
- ☐ нелинейный видеомонтаж;
- ☐ перекодирование видео;
- ☐ создание DVD-образов.

Захват видео

В случаях, когда запись видео ведется на видеокассету, возникает проблема «перегона» записанных данных с видеокассеты на компьютер. Именно для этого существуют программы захвата видео. Несмотря на то, что практически любой видеоредактор имеет встроенную функцию захвата, использовать специализированные программы гораздо удобнее. Одной из наиболее известных программ для захвата видео является ScenealyzerLive. Ее возможности:

- ☐ захват видео по сценам;
- ☐ быстрый просмотр всех сцен на диске;
- ☐ быстрый поиск и сохранение на диске наиболее интересных стоп-кадров;
- ☐ захват фрагментов;
- ☐ считывание видеофайлов с диска и сохранение их на кассете в видеокамере с теми же именами файла и папки, в которой он находился;
- ☐ хорошая буферизация данных (это позволяет захватывать поток без пропущенных кадров на тех компьютерах, аппаратная конфигурация которых делает невозможным захват без потерь другими программами захвата);
- ☐ копирование на камеру набора файлов в виде одной записи (это позволяет обходить ограничения файловой системы FAT32).

Нелинейный видеомонтаж

Монтаж аналогового видео всегда носит линейный характер, поскольку для того, чтобы на киноленте или на видеокассете добраться до определенного фрагмента, необходимо промотать пленку до нужного места. Когда мы имеем дело с цифровым видео, все «нужные места» находятся перед нами в раскадровке видеозаписи, и мы совершенно свободны в своем творчестве: может любым образом выделять, вырезать, вставлять или удалять фрагменты, применять цифровые эффекты как к выделенным участкам, так и к отдельным кадрам. *Нелинейным видеомонтажом* называется процесс видеомонтажа, при котором можно в любой момент перейти к любому произвольному участку видеозаписи.

Нелинейный видеомонтаж возможен только для цифрового видео. Наиболее известными программами для осуществления нелинейного видеомонтажа являются:

- ☐ Adobe Premiere;
- ☐ iMovie;
- ☐ Virtual Dub;
- ☐ Windows Movie Maker.

Все эти программы предлагают примерно одинаковый набор функций (исключением является, пожалуй, программа Windows Movie Maker, которая изначально не делалась, как профессиональная, а была ориентирована скорее на простоту использования и позволяла работать даже неподготовленным пользователям):

- ☐ получение видео с цифровой видеокамеры;
- ☐ создание слайд-шоу из изображений;
- ☐ обрезание или склеивание видео;
- ☐ наложение звуковой дорожки;
- ☐ добавление заголовков и титров;
- ☐ создание переходов между фрагментами видео;
- ☐ сохранение проекта в заданном формате и с заданным качеством.

Разница в возможностях программ в основном связана с количеством и качеством цифровых фильтров и эффектов обработки изображения и переходов между фрагментами.

Все перечисленные программы являются коммерческими. Однако есть несколько свободных программ как для Windows, так и для Linux, обладающих схожими возможностями по обработке и монтажу цифрового видео: Cinelerra, Kdenlive, Kino, LIVES.

Перекодирование видео

Чем быстрее развиваются компьютеры и мультимедиа, тем больше различных форматов хранения и передачи потокового видео разрабатывается производителями. Часто возникает потребность преобразовать (конвертировать) файл из одного формата в другой. Наиболее популярной программой для перекодирования цифрового видео сегодня считается CANOPUS ProCoder, которая может работать как в качестве самостоятельного приложения, так и как дополнение к Adobe Premiere.

Создание DVD-образов

Созданные цифровые видеозаписи обычно требуется надлежащим образом оформить: не просто записать на DVD-диск как каталог с файлами, а снабдить интерактивным меню, обеспечить простой переход между фрагментами и быстрый поиск информации. Для этого служат специальные программы для создания и оформления DVD-образов цифровых фильмов: Ulead DVD MovieFactory, iDVD и DVD Architect.

18.5. Мультимедиа-презентации

Мультимедиа-презентации не просто получили широкое распространение, они стали одним из главных средств представления определенных видов информации. Сегодня ни совещания, ни важные переговоры не обходятся без мультимедиа-презентаций. При этом существующие средства мультимедиа позволяют подготовить презентацию за сравнительно небольшой промежуток времени.

Наиболее широко мультимедиа-презентации представлены на сегодняшний день в двух сферах: в рекламе и образовании, причем и там, и там они исполняют одну и ту же роль: быстро, эффективно, надежно и достоверно донести до аудитории новые знания.

18.5.1. Возможные типы мультимедиа-презентаций

Рекламные презентации. В этом случае презентация создается с целью рекламы, то есть продвижения торговой марки компании; распространения информации о компании; лучшей узнаваемости имиджа; рекламирования сопутствующих и дополняющих товаров, информационных ресурсов по данной теме. Ролик отличается от презентации линейным представлением информации, то есть в нем не предусмотрено взаимодействие с пользователем, который в этой ситуации является только зрителем. Электронные презентации и рекламные ролики — эффектный и комфортный способ привлечь внимание пользователей к товарам и услугам как самой компании, так и ее партнеров и представителей.

Электронные каталоги. Такие презентации могут быть использованы в качестве каталогов товаров и услуг. Торговые представители и менеджеры смогут задействовать весь материал целиком или отдельные его части. Во всем мире многие издательства и компании постепенно переходят с выпуска массивных рекламных каталогов и других печатных изданий к гораздо более эффективному и экономичному мультимедийному компакт-дискам. Это дает возможность распространять большие объемы информации быстро, качественно и эффективно.

Обучающие и тестовые программы. С помощью такой презентации можно обучать сотрудников, слушателей семинаров, курсов, и т. п. Такая презентация может быть приложением к технически сложной и наукоемкой продукции. Она не только обучит, но и проконтролирует уровень знаний.

Нормативно-техническая документация, методическая и сопутствующая литература. Презентация может быть использована в качестве оболочки для удобного

доступа к информации. Такая презентация может содержать чертежи, технические паспорта, руководства по эксплуатации и т. п. В этом случае презентация обеспечивает не только удобный доступ к данным, но и позволяет существенно экономить на тиражировании больших объемов информации.

Визитная карточка. Электронная визитная карточка с мультимедиа-презентацией — необходимая вещь для людей, ведущих активный, публичный образ жизни, включая бизнесменов, артистов, политиков, общественных деятелей. Любая «типографская» визитка будет «проигрывать» электронной. Размер CD-диска электронной визитной карточки сопоставим с размером обычной визитной карточки.

18.5.2. Программные технологии создания мультимедиа-презентаций

Для создания мультимедиа-презентаций в основном используют ставшие уже привычными программные средства, такие как Microsoft Power Point из пакета Microsoft Office, или свободные программы, такие как OpenOffice.org Impress. Та и другая программы поддерживают самый широкий спектр возможностей, позволяющих создавать презентации, насыщенные интерактивными и мультимедийными элементами.

Создание презентации начинается с запуска мастера создания презентации, выбора шаблона, если вы хотите создать презентацию по шаблону, и ввода необходимых сведений. Цель и содержание презентации продумываются до начала ее создания.

После того как мастер завершит свою работу, мы окажемся в основном окне программы Impress (рис. 18.17).

Область Слайды представляет собой перечень всех слайдов презентации, представленных в виде эскизов. Когда нужно выбрать один из слайдов, чтобы проделать над ним какое действие, нужно щелкнуть на соответствующем эскизе в этом списке. Если надо выбрать группу слайдов, нужно щелкать на нужных слайдах, удерживая нажатой клавишу Ctrl.

Выбор режима редактирования происходит с помощью соответствующих вкладок области редактирования.

- ☐ Режим рисования — основной режим работы с отдельным слайдом презентации. Позволяет добавлять на одиночный слайд графические объекты и текст, изменять форматирование и размещение текста. Именно в этом режиме происходит выбор макета для конкретного слайда.
- ☐ Режим структуры — на этой вкладке можно редактировать структуру презентации, то есть ее разделы, подразделы и заголовки отдельных слайдов. Основное назначение режима структуры — редактирование текста слайдов и изменение уровня структуры внутри каждого слайда.
- ☐ Режим примечаний — этот режим позволяет добавить к каждому слайду краткие примечания (типа «не забыть рассказать») или снабдить слайд полным текстом, который надо прочитать при появлении слайда на экране. Примечания, естественно, не будут показаны в момент демонстрации презентации. Для того

чтобы ими воспользоваться, нужно распечатать презентацию, не забыв установить в параметрах печати флажок Печатать примечания.

- **Режим тезисов** — этот режим позволяет распечатать презентацию в формате с 1, 2, 3, 4 или 6 слайдами на листе. Выбор количества слайдов осуществляется на вкладке Макет области Задачи.
- **Сортировщик слайдов** — на этой вкладке представлены все слайды презентации, где путем перетаскивания мышью можно менять порядок следования слайдов в презентации.

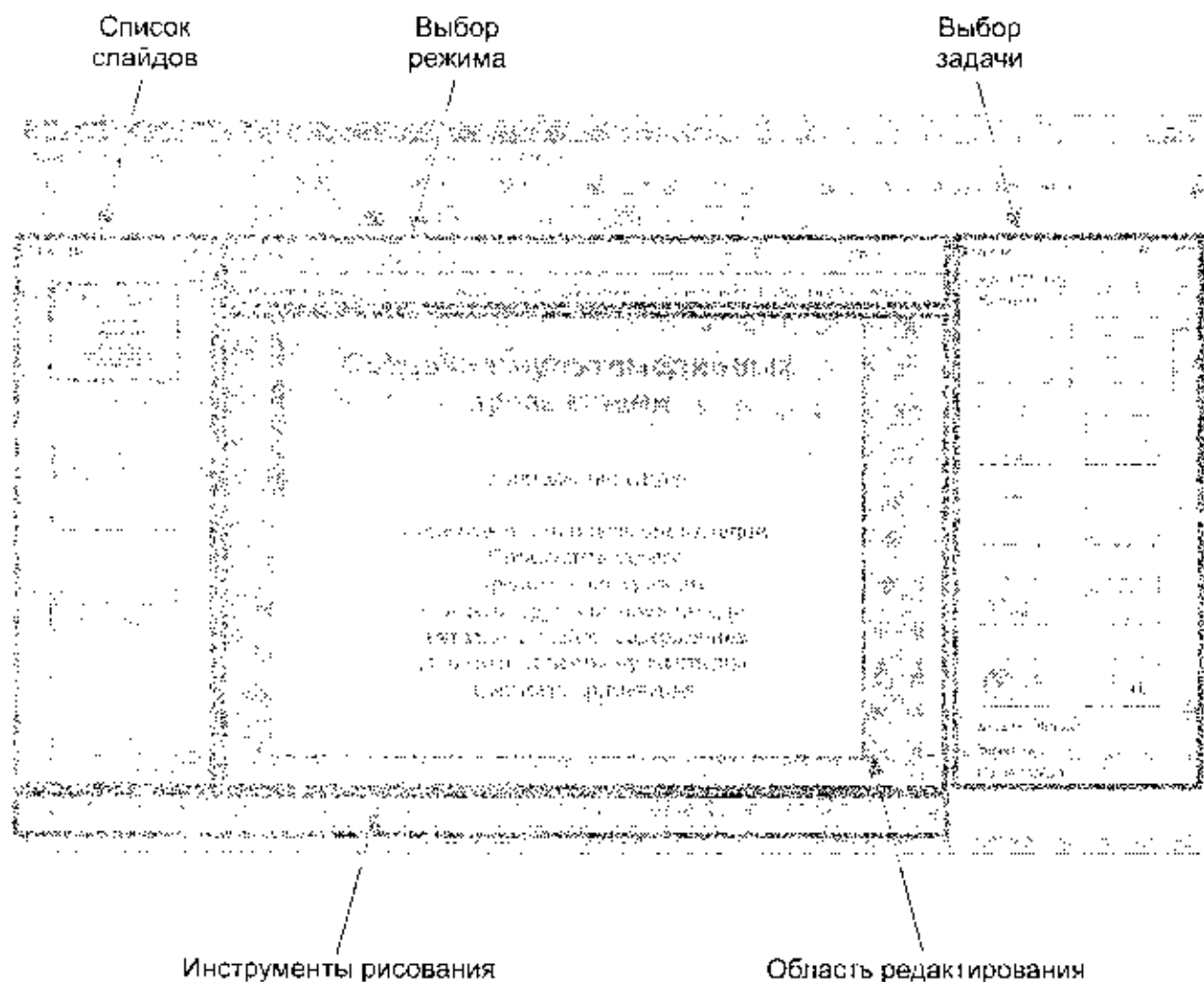


Рис. 18.17. Окно программы OpenOffice.org Impress

В области Задачи выбирается задача, которую требуется решить для выбранного объекта, одного или нескольких слайдов либо презентации в целом. Обратите внимание, что одни задачи (например, Фоны страниц) всегда выполняются над презентацией в целом, другие относятся к конкретным слайдам или группам слайдов (Макеты и Смена слайдов), третьи становятся доступными только при выборе внутри одного слайда конкретного объекта, текста или рисунка (Эффекты).

В области редактирования происходит основное действие по редактированию отдельных слайдов и всей презентации.

Инструменты рисования позволяют вставлять в презентацию графические объекты и специальным образом оформленный текст.

Рассмотрим основные этапы создания презентации в Impress. Подобные инструменты и команды, но с немного иным названием, есть и в программе Power Point.

1. Точно определяемся с типом, целями и задачами презентации.
2. На основе заранее подобранного материала выделяем структуру презентации, текст слайдов и текст примечаний.
3. Переносим структуру презентации на один из слайдов.
4. Выполнив команду *Расширить слайд для слайда со структурой*, получаем все слайды презентации, снабженные соответствующими структуре заголовками.
5. Оформляем шаблоны слайдов (с помощью мастера Слайды) для первого слайда презентации, заголовочных слайдов разделов и остальных слайдов презентации. Связываем с созданными шаблонами соответствующие слайды презентации.
6. Наполняем созданные слайды текстом и примечаниями.
7. Добавляем к слайдам иллюстрирующие изображения (диаграммы, графики, рисунки).
8. Помещаем в слайды анимацию, фрагменты видео, музыкальное или звуковое сопровождение. Анимация и видеосфрагменты автоматически начинают воспроизводиться при выводе на экран содержащего их слайда, то же самое касается музыкальных и звуковых файлов.
9. В случае необходимости в отдельных сложных слайдах задаем эффект постепенного появления текста и рисунков.
10. Сохраняем презентацию в нужном формате.

OpenOffice.org, и Microsoft Power Point способны сохранять презентацию в разных форматах. Помимо стандартного формата презентации, требующего наличия на компьютере программы, в которой презентация создавалась, можно сохранить презентацию в формате самостоятельного исполняемого файла, не нуждающегося в наличии на компьютере программ Power Point или Impress. Обе программы позволяют сохранять презентации в форматах SWF (то есть в виде flash-анимации) и PDF (а программа Adobe Acrobat Reader умеет воспроизводить формат PDF определенной структуры в режиме презентации). И формат PDF, и формат SWF являются межплатформенными, но требуют для своего выполнения соответствующих инструментов воспроизведения файлов этого формата. Наиболее универсальным решением является сохранение презентации в качестве веб-документа. В этом случае презентация сохраняется как локальная структура веб-сайта, содержащая меню навигации, и это решение является наиболее универсальным с точки зрения совместимости платформ.

18.5.3. Демонстрация мультимедиа-презентаций

Для воспроизведения мультимедиа-презентаций могут применяться несколько режимов.

- *Воспроизведение презентации на одном компьютере.* Этот режим применяется обычно при демонстрации рекламного ролика или учебного материала одному человеку.
- *Трансляция презентации с одного компьютера на несколько клиентских мест.* Такое решение обычно выбирается в компьютерных классах при наличии компьютеров, объединенных в локальную сеть. Наиболее частое применение — организация учебного процесса.
- *Трансляция презентации при помощи пассивного проектора.* В этом случае презентация, воспроизводимая на экране компьютера через проекционное оборудование транслируется на экран. Управление переключением слайдов может производиться как с компьютера, так и с устройства дистанционного управления проектором.
- *Применение для демонстрации интерактивных средств,* таких как большие плазменные сенсорно-чувствительные экраны и интерактивные электронные доски. В этом случае лектор может не только переключать слайды, но и управлять поведением объектов на экране, непосредственно взаимодействуя с поверхностью экрана (не только переключать или передвигать слайды, но и рисовать на экране непосредственно указкой или световым лучом, выделяя значимые элементы).

Вопросы для самопроверки

1. Что такое «мультимедийный документ»?
2. Что включает в себя современная трактовка термина «мультимедиа»?
3. Чем оцифрованный звук отличается от синтезированного?
4. Что такое «частота дискретизации», каково ее стандартное значение при промышленной оцифровке звука?
5. Что такое 16-битный звук, чем он отличается от 8-битного?
6. Чем табличный процессор отличается от базы данных?
7. Какие два типа сжатия звука нам известны?
8. При помощи какого устройства получают звук в формате MIDI?
9. Что такое «микширование»?
10. При помощи каких программ производится управления звуковыми каналами?
11. Какие программы для редактирования оцифрованного звука вам известны?
12. Какие программы для создания MIDI-композиций вы знаете?
13. В чем отличие растрового формата от векторного?
14. Есть ли избыточность в цифровом коде, представляющем изображение?

15. По какому принципу осуществляется сжатие изображения?
16. Какие программы для создания и редактирования векторных изображений вам известны?
17. Какие программы для создания и редактирования растровых изображений вам известны?
18. На основе каких принципов происходит кодирование цифрового видео?
19. Что такое «нелинейный видеомонтаж»? Можно ли применять нелинейный видеомонтаж при монтировании записей на видеокассетах?
20. Какие типы мультимедиа-презентаций вы знаете?
21. Назовите основные этапы создания мультимедиа-презентации.

Литература

1. *Белунцев В.* Звук на компьютере. СПб.: Питер, 2005.
2. *Ватопин Д., Ратушняк А., Смирнов М., Юкин В.* Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. М.: ДИАЛОГ-МИФИ, 2003.
3. *Миано Дж.* Форматы и алгоритмы сжатия изображений. М.: Издательство Триумф, 2003.

Глава 19 Технологии и инструменты программирования

- 19.1. Основные понятия и классификация языков программирования
- 19.2. Краткая история языков программирования
- 19.3. Концепция объектно-ориентированного программирования
- 19.4. Инструментальные средства и среды разработки программного обеспечения
- 19.5. Жизненный цикл программного обеспечения

В данной главе рассматриваются языки программирования, методы и средства создания программного обеспечения, а так же практические методики, применяемые при разработке ПО; дается определение различным понятиям, связанным с процессом создания программного обеспечения; описываются основные концепции объектно-ориентированного программирования. Особое внимание в главе уделено инструментальным средствам разработки программного обеспечения, а также средствам поддержки жизненного цикла программного обеспечения. Обсуждаются различные модели организации разработки программного обеспечения.

19.1. Основные понятия и классификация языков программирования

19.1.1. Основные понятия

Язык программирования предназначен для описания данных и алгоритмов их обработки на вычислительной машине. Языки программирования занимают промежуточное положение между естественными и формализованными языками. С естественными языками языки программирования роднит грамматический строй (употребление слов естественного языка, фразовая структура и т. п.), с формализованными языками — символы и понятия, а главное — строгие, точно описанные правила построения текстов.

Язык программирования — это формализованный язык, предназначенный для описания программ и алгоритмов решения задач на компьютере.

Существуют различные способы задания инструкций компьютеру, заставляющие его выполнять необходимые вычисления. В одних случаях достаточно полностью описать алгоритм на формальном языке, в других приходится управлять более сложными деталями реализации программы: выделять память, следить за состоянием регистров процессора. Сложность, гибкость и скорость выполнения программ, как и потребление ресурсов компьютера, зависят от языка программирования, выбранного программистом. Но в любом случае результатом работы программиста является компьютерная программа.

Компьютерная программа — это последовательность инструкций на машинно-ориентированном языке, предназначенных для выполнения центральным процессором вычислительной машины.

Часто компьютерной программой называют исходный код, написанный на одном из языков программирования. Компьютерную программу принято считать результатом программирования.

Программирование — это совокупность процессов, связанных с разработкой программ и их реализацией.

В широком смысле к указанным процессам относят все технические операции, необходимые для создания программ, включая анализ требований, а также все стадии разработки и реализации в виде готового программного продукта. Достаточно схематично процесс программирования представлен на рис. 19.1.

На рисунке показан приблизительный порядок действий при программировании. Программирование заключается в написании текста (исходного кода) программы на выбранном языке программирования.

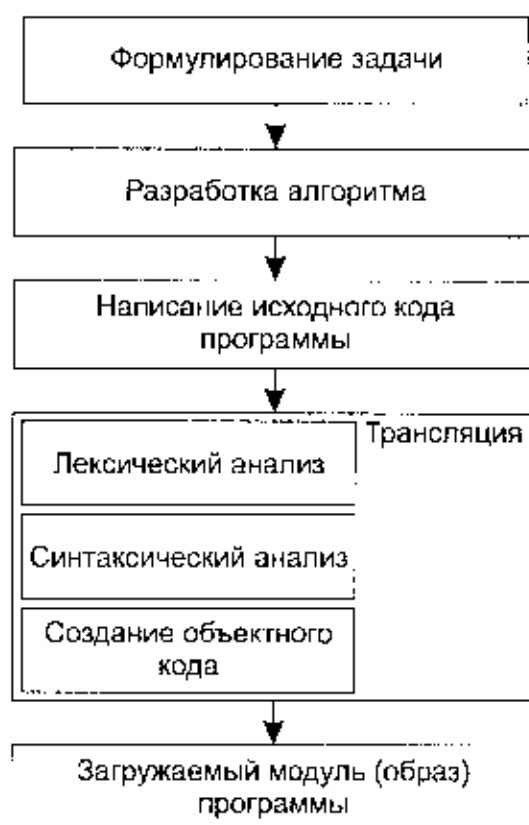


Рис. 19.1. Схематичное описание процесса программирования

Исходный код — текст компьютерной программы, написанный на выбранном языке программирования.

Исходный код преобразуется транслятором (специальной программой) в объектный модуль, который хранится на диске.

Транслятор — программа, осуществляющая трансляцию (перевод) компьютерной программы с языка программирования в программу на машинном коде с сохранением эквивалентности объектов программ.

Для выполнения программы должен быть определенным образом подготовлен объектный модуль — определены адрес его загрузки в оперативную память и адреса связи с другими объектными модулями. Затем загружается объектный модуль в оперативную память.

Объектный модуль — последовательность инструкций машинного кода, являющаяся результатом работы компилятора.

Машинный код — язык конкретной вычислительной машины (центрального процессора), инструкции которого выполняются непосредственно центральным процессором или его микропрограммами.

Исполняемый код — объединенные в загружаемую программу модули объектного кода.

Объединение модулей объектного кода в единую исполняемую программу является результатом работы компоновщика.

Процесс трансляции состоит из нескольких этапов.

1. Производится лексический анализ текста с разбиением его на элементы-лексемы (ключевые слова, имена-идентификаторы и т. д.).
2. Производится синтаксический разбор, то есть проверка правильности и допустимости созданных на основе лексем конструкций языка.
3. Исходный код преобразуется в исполняемый машинный код.

Это описание процесса трансляции подводит нас к понятиям, связанным с языками программирования — синтаксису и семантике языка.

Синтаксис искусственных и естественных языков определяет, какие конструкции из допустимых лексем являются правильными для данного языка.

Пример. Рассмотрим предложение: «Это кошка страус зеленая». В этом примере все используемые в предложении слова и символы допустимы для русского языка, но вместе они составляют недопустимую конструкцию.

Синтаксис языка программирования — это описание языка программирования, определяющее вид, форму и структуру конструкций языка.

Формализация описания языка программирования нужна для создания математической или логической модели работы транслятора, то есть те правила, на основании которых транслятор будет производить синтаксический разбор. Одним из известных способов формального описания синтаксиса языка является метаязык — формы Бокуса-Наура. Задание синтаксиса языка программирования сводится к перечислению всех его допустимых конструкций.

Таким образом синтаксис связан только с перечислением всех конструкций языка, которые могут быть преобразованы транслятором в машинный код. Смысловая же составляющая языка программирования описывается понятием семантики.

Пример. Рассмотрим предложение: «Зеленая кошка — это страус». Этот пример проходит синтаксическую проверку, поскольку предложение построено правильно, но с семантической точки зрения он лишен смысла.

Семантика языка программирования — это интерпретация абстрактного синтаксиса в терминах модели.

Семантические модели для языков программирования могут быть *операционными* (моделируется выполнение на абстрактной вычислительной машине), *деривационными* (модель описывается языком логических выражений) и *денотационными* (модель описывается понятиями, близкими к математическим, такими как множества, утверждения, суждения).

Синтаксические и семантические способы описания языков программирования служат основой для создания эффективных трансляторов.

Помимо программы-транслятора, разработка которой основана на формальной математической модели, можно выделить несколько менее формализованных требований к языку.

- ❑ *Полнота.* Язык программирования должен исчерпывающе описывать синтаксис и семантику, то есть все допустимые конструкции языка.
- ❑ *Ясность.* Язык программирования должен быть понятен человеку. Несколько мощных языков программирования, которым сулили прекрасное будущее, не были востребованы программистами из-за того, что программы на этих языках получались слишком сложными и не наглядными.
- ❑ *Естественность.* Условные обозначения, термины и конструкции языка должны быть близки программисту. Возможно, язык программирования, записываемый с помощью иероглифов, выглядел бы очень компактно и красиво, но если иероглифическое письмо не свойственно нашей культуре, то для нас такой язык будет неестественным.
- ❑ *Реализм.* Реализация языка программирования должна учитывать фактические условия его использования. Язык программирования, предполагающий наличие нереальных объемов памяти или нереальной производительности процессора компьютера, востребован не будет, так же как язык программирования, программы на котором выполняются слишком медленно.
- ❑ *Наличие среды разработки.* Многие программисты создают программы в простейшем текстовом редакторе и отлаживают из командной строки. Однако более быстрая и менее затратная технология разработки программ возможна только при наличии специальных сред и соответствующего инструментария.
- ❑ *Низкая стоимость использования.* Необходимо оценивать затраты на написание программ, их отладку и последующее выполнение. Это часто становится определяющим фактором при выборе языка программирования.

Перечислим основные понятия, свойственные большинству языков программирования.

- ❑ *Переменная* — область памяти, адрес (имя) которой можно использовать для получения или изменения хранящегося в ней значения. Переменные могут быть *простыми*, то есть не иметь внутренней структуры, и *сложными*, то есть иметь внутреннюю структуру, к элементам которой возможен доступ.
- ❑ *Константа* — область памяти, адрес (имя) которой можно использовать для получения хранящегося в ней значения.
- ❑ *Инструкция* — наименьшая автономная часть языка программирования. Программа представляет собой последовательность инструкций.
- ❑ *Подпрограмма* — помеченный фрагмент программы, который может быть неоднократно вызван и выполнен в процессе выполнения программы.
- ❑ *Функция* — подпрограмма, которая может быть использована в выражении.
- ❑ *Процедура* — любая подпрограмма, не являющаяся функцией.
- ❑ *Оператор* — специальный вид функции, записываемый в виде символа. Способ записи операторов делает их сходными с математическими операторами,

а возможность перегрузки (то есть замены действия, которое выполняется над операндами другим действием) делает операторы похожими на функции.

Пример. Оператор $+$ в математике и в языках программирования традиционно обозначает операцию сложения двух операндов. Однако в некоторых языках программирования (например, в Pascal) эту операцию можно выполнять над строками и в случае строковых операндов оператор $+$ реализует конкатенацию (соединение) двух строк в одну.

- *Выражение* — последовательность операций, результат выполнения которых может быть присвоен переменной.
- *Тип данных* — относительно устойчивая и независимая совокупность данных, которую можно выделить в рассматриваемом множестве. Есть два базовых типа данных: числовые (бинарные) и символьные. К первому типу данных относятся все виды чисел (целые, дробные, положительные, отрицательные), ко второму — символы и строки. Каждый язык определяет свое множество как числовых, так и символьных данных. Логический тип данных, представляющий собой тип с двумя значениями (`true` или `false`, 1 или 0), в одних языках выделяется в отдельный тип, в других реализуется через числовые типы. Кроме того, могут существовать сложные типы данных, состоящие из комбинации простых типов. Деление данных на типы удовлетворяет две потребности: во-первых, каждый тип данных имеет свой размер в памяти. Когда задается переменная определенного типа, то ясно, сколько места нужно выделить для хранения значения этой переменной и как она будет там представлена. Во-вторых, тип данных определяет, какие операции можно производить над этими данными. Например, можно перемножить между собой два числа, но нельзя проделать эту операцию над двумя строками.

19.1.2. Классификация языков программирования

По способу выполнения языки программирования делятся на компилируемые, интерпретируемые, компилируемые на основе псевдокода и совмещенные.

- *Компилируемые языки.* К этой группе относят языки программирования, исходный код которых преобразуется специальной программой-компилятором в объектные модули, которые затем собираются при помощи программы сборки (линковщика) в единый загружаемый модуль. Этот модуль представляет собой выполняемую программу. Компилируемыми являются языки C, C++, Object Pascal.
- *Интерпретируемые языки.* К этой группе относят языки программирования, исходный код которых считывается и выполняется специальной программой-интерпретатором инструкция за инструкцией. Интерпретируемыми являются большинство версий языков Basic и Forth.
- *Языки, компилируемые на основе псевдокода («шитого» кода).* В этих языках исходный код программы компилируется и выполняется как последовательность вызовов подпрограмм из существующих библиотек. Таковыми являются некоторые версии Basic.

- *Совмещенные языки.* В совмещенных языках исходный код проходит две стадии обработки. На первой стадии происходит компиляция исходного кода до уровня промежуточного языка (байт-код в Java или язык MSIL в технологии .NET). На второй стадии происходит интерпретация промежуточного кода (в Java этим занимается виртуальная машина Java) или докомпиляция промежуточного кода до выполняемого машинного кода (эту функцию осуществляет среда выполнения CLR в технологии .NET).

По факту созданию процесса языки программирования делятся на создающие процесс и сценарные.

- *Языки, создающие процесс.* После запуска программы создается отдельный процесс выполнения этой программы. Так происходит в языках C, C++, Object Pascal.
- *Сценарные языки.* Сценарий, или скрипт, — это программа, которую выполняет другая программа. В качестве примером можно привести скрипты оболочки в UNIX, программы на языках PHP, Python, Ruby.

По степени автономности языки программирования делятся на автономные и встроенные.

- *Автономные языки программирования* (C, C++, Java) являются автономным инструментом для создания программ.
- *Встроенные языки программирования* являются частью какой-то системы и позволяют создавать программы, предназначенные для работы только в этой системе. Пример наиболее известного встроенного языка — VBA (Visual Basic for Application), который используется только внутри приложений Microsoft Office для автоматизации и расширения их функциональности. Встроенными являются также язык программирования системы IC и язык JavaScript, который выполняется только внутри интернет-приложений.

По уровню отдаленности языка программирования от естественных языков их делят на низкоуровневые и высокоуровневые.

- *Языки низкого уровня* ближе к логике процессора вычислительной машины, например, машинный язык, ассемблер.
- *Языки высокого уровня* ближе лингвистически к человеческому языку. Это все остальные языки программирования.

По парадигмам языки программирования делятся на императивные (процедурные), функциональные, логические и объектно-ориентированные.

- *Императивные (процедурные) языки* описывают решение задачи как последовательность процедур. К императивным относится большинство современных языков программирования.
- *Функциональные языки* описывают требуемый результат в виде набора вложенных друг в друга функций (Haskell).
- *Логические языки* описывают требуемый результат в виде суммы логических операций (Prolog).

- *Объектно-ориентированные языки* способны определять абстрактные типы данных и реализовывать основные парадигмы объектно-ориентированного программирования (ООП): инкапсуляцию, наследование и полиморфизм (C++, Java, Object Pascal и др.).

19.2. Краткая история языков программирования

19.2.1. Первый этап — машинные коды

Появление первых языков программирования можно отнести к периоду 20–40-х гг. XX в. Самые первые опыты программирования, когда в память вычислительной машины вводились инструкции на машинном языке сначала с помощью переключек, а потом — набора тумблеров, сегодня могут вызывать только улыбку. Это была очень тяжелая и непроизводительная работа. Появление терминала и клавиатуры упростило процесс ввода, но не процесс программирования.

Пример. Фрагмент программы на машинном языке, представлен на рис. 19.2.

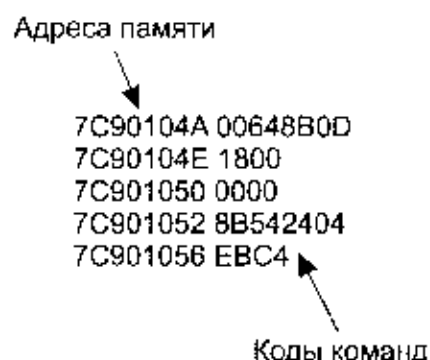


Рис. 19.2. Фрагмент программы в машинных кодах

Слева на рисунке находятся адреса, справа — машинные коды, которыми кодируются инструкции процессора (коды команд).

При программировании в машинных кодах программисту нужно постоянно под рукой держать справочник, описывающий каждое числовое значение: что это за команда, как именно она работает. Если учесть, что в современных процессорах число инструкций может превышать несколько сотен, то вполне понятно, что даже при долгой практике запомнить все инструкции — задача практически невыполнимая. А обращаться для ввода каждой инструкции к справочнику — значит, делать процесс написания даже маленькой программы длительным и утомительным. По этой причине программирование в машинных кодах длилось очень недолго, а для программирования непосредственно на аппаратном уровне довольно быстро был разработан язык ассемблера. В языке ассемблера команды легко группируются по функциональности.

Пример. Фрагмент программы на языке ассемблера представлен на рис. 19.3. По сути это те же машинные команды, но только в символьной записи (см. рис. 19.3).

```
add    [ebx+ecx*4+$0d],ah
sbb    [eax],al
add    [eax],al
mov    edx,[esp+$04]
jmp    -.$3c
```

Рис. 19.3. Фрагмент программы на языке ассемблера

Запомнить, что команда `MOV` означает «переместить», а указанные затем два операнда показывают, откуда нужно взять и куда положить значение, -- совсем не трудно.

19.2.2. Второй этап — языки высокого уровня

В языках программирования высокого уровня не требуется знания отдельных машинных команд. Каждая команда в языке программирования высокого уровня скрывает десятки и сотни команд на языке ассемблера. Программисты получают возможность сосредоточиться на алгоритме работы программы, не заботясь об аппаратной реализации компьютера, как было с языками первого поколения. В языках высокого уровня появились средства вызова подпрограмм и создания локальных переменных. В 60-е гг. были созданы такие языки, как ALGOL, Fortran, APL.

19.2.3. Третий этап — структурное программирование

Однако программные проекты на языках программирования высокого уровня оставляли желать лучшего: сроки выполнения проектов были очень велики, а материальные издержки выходили за самые пессимистические прогнозы. Наступило время смены уже не языков, а самого подхода к программированию. Программирование «снизу вверх», когда вначале разрабатывались подпрограммы, а затем эти подпрограммы объединялись в единое целое при помощи инструкций перехода, не оправдало себя. Код получался запутанным, неэкономичным и сложным в отладке. Требовалось изменить саму методологию программирования, вместо программирования «снизу вверх» надо было ввести программирование «сверху вниз», при котором сначала определялись стратегические параметры программного обеспечения, вырабатывался наиболее общий алгоритм работы программы, а затем производилась декомпозиция этого алгоритма до уровня подпрограмм размером в 50–70 строк кода. Далее подпрограммы объединялись в единую программу. Такой подход требовал усовершенствований в самих языках программирования на уровне команд управления ходом выполнения программы. Такие усовершенствования были внесены в языки ALGOL-68, C и Pascal, а сама новая методология получила название структурного программирования.

19.2.4. Четвертый этап — модульное программирование

Порядок, который был наведен с внедрением методологии структурного программирования, был достаточно эффективен, но эта методология не решала проблемы многократного использования уже разработанного кода. Следующим шагом в развитии языков и методологий программирования было создание модульного программирования. При модульном программировании программа делилась на модули, которые при компиляции образовывали отдельные объектные файлы. Объектные файлы затем собирались в единое целое и получалась исполняемая программа. Появилась возможность объединять фрагменты программы по функциональному признаку. В каждой программе есть код, который уникален только для этой программы, но всегда есть код, применимый при разработке другого программного продукта, например, код, отвечающий за обработку строковых данных. Фрагменты кода, объединенные в модули, можно было использовать многократно в разных программах как в виде исходного кода, так и в виде скомпилированных объектных файлов. Такой подход многократно увеличил скорость разработки программного обеспечения и позволил дополнительно упорядочить сам процесс разработки.

19.2.5. Пятый этап — объектно-ориентированный подход

Алгоритмические языки предыдущих этапов, в основном, обеспечивали реализацию функциональности. При объектно-ориентированном подходе программист должен думать в терминах объектов окружающего мира, нашедших свое абстрактное отражение в сложных типах данных, называемых классами. А языки программирования призваны были давать возможность отображать в коде объекты внешнего мира с их свойствами и поведением. Этот этап можно смело назвать революционным по значимости, поскольку все остальные новации в программировании продолжают основываться на принципах объектно-ориентированного программирования (ООП). Яркими представителями языков, поддерживающих ООП, являются C++, Java, Object Pascal и все семейство языков в технологии Microsoft .NET. Именно объектно-ориентированный подход лежит в основе создания средств визуального проектирования и быстрой разработки программ, таких как Borland Delphi, Borland JBuilder и Microsoft Visual Studio.

19.2.6. Шестой этап — компонентный подход

Компонентный подход предполагает построение программного обеспечения из независимых друг от друга «кирпичиков», которые объединяются между собой благодаря специальным стандартизированным интерфейсам. При компонентном подходе совместимость различных частей, или объектов, программы между собой обеспечивается тем, что все эти объекты соблюдают единые правила, декларируемые на уровне операционной системы. Компонентный подход позволяет в рамках одной программы работать компонентам, написанным на разных языках программирования и скомпилированным в разных средах разработки.

Наиболее известными технологиями, реализующими компонентный подход, являются COM, CORBA и .NET.

19.2.7. Седьмой этап — архитектура, управляемая моделью

Этот этап развития технологий программирования пока еще не достиг своего логического завершения, да и сама идея архитектуры, управляемой моделью (Model Driving Architecture, MDA), продолжает развиваться и совершенствоваться, в частности, в рамках технологии ECO для .NET. Смысл технологии MDA — исключить этап кодирования из процесса разработки программного обеспечения. Для того чтобы создать завершенное приложение, достаточно построить его модель на языке UML, а все остальное за вас сделает среда разработки: сгенерирует классы данных, процедуры их обработки, способы подключения к данным, пользовательский интерфейс. Хотя это направление развития является весьма многообещающим, на сегодняшний день основным способом создания программного обеспечения все-таки остается ручное кодирование.

19.3. Концепция объектно-ориентированного программирования

Поскольку все современные средства разработки программного обеспечения, языки и технологии создания программ продолжают основываться на объектно-ориентированном программировании, предлагаем ознакомиться с основными положениями этого подхода. Объектно-ориентированное программирование базируется на четырех основных принципах: абстракции, наследовании, инкапсуляции и полиморфизме.

19.3.1. Абстракция

Абстракция — это способность языка программирования отображать объекты внешнего мира в форме абстрактных структур в соответствии с решаемой задачей.

Абстрактные структуры, при помощи которых реализуется этот принцип, называются классами.

Класс — это структура, описывающая объекты внешнего мира с помощью двух типов элементов: состояние объекта описывается полями класса (переменными разного типа), а поведение объекта — его методами (процедурами и функциями).

Классы представляют собой абстрактные описания структур данных, но сами данные они не содержат. Данные появляются тогда, когда по описаниям классов в памяти программы выделяется необходимое пространство и в нем создаются экземпляры класса, или *объекты*. Тогда для каждого поля класса отводится необходимая область памяти и в эту область можно поместить нужное значение.

Пример. Предположим, у нас есть три измерительных прибора: термометр, барометр и весы. Чтобы описать эти приборы внутри программы, мы абстрагируем важные для программы свойства этих приборов и создаем три класса (рис. 19.4).

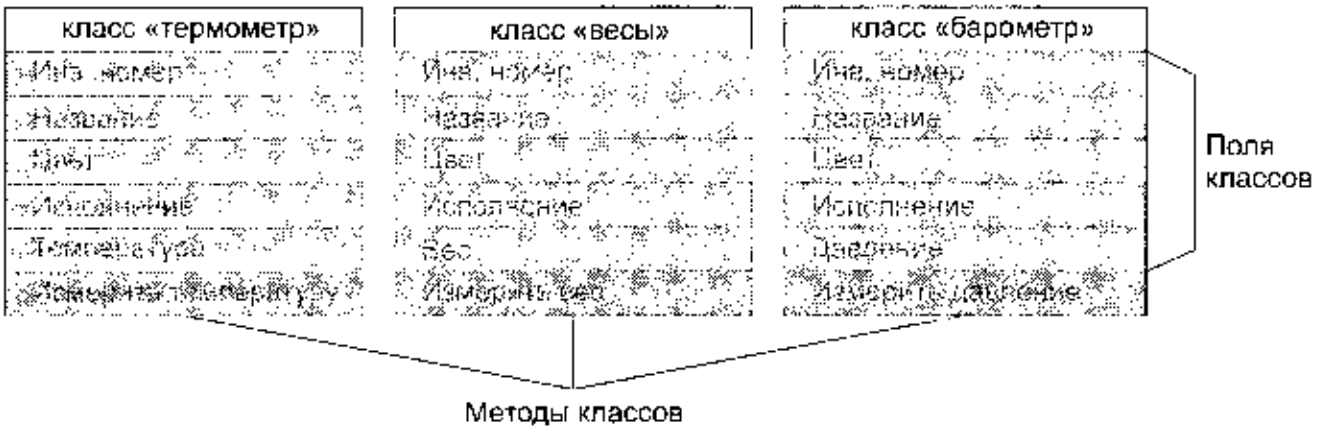


Рис. 19.4. Классы

По названиям в соответствующих ячейках видно, что поля класса содержат в себе данные, описывающие состояние класса. Методы класса представляют реализацию некоторых действий, то есть описывают поведение класса. Например, метод класса «термометр», который носит название «измерить температуру», призван помещать значение температуры в поле с именем «температура». На рис. 19.5 представлены объекты (экземпляры класса).

Экземпляр класса «термометр»		Экземпляр класса «весы»		Экземпляр класса «барометр»	
Инв. номер	1	Инв. номер	3	Инв. номер	3
Название	TRM	Название	VSV	Название	DRM
Цвет	белый	Цвет	серый	Цвет	хром
Исполнение	оконный	Исполнение	напольные	Исполнение	настенный
Температура	10	Вес	2	Давление	230
Измерить температуру		Измерить вес		Измерить давление	

Рис. 19.5. Экземпляры классов

19.3.2. Наследование

Наследование — это свойство класса порождать другие классы таким образом, что в порожденном классе (классе-потомке) содержится все поля и все методы класса-родителя (базового класса), а также дополнительно собственные поля и методы.

Пример. В рассмотренном примере у трех разных классов есть одинаковые поля: Инв. номер, Название, Цвет, Исполнение. Каждый раз при создании нового класса измерительного прибора приходится снова и снова создавать эти поля. Данная ситуация значительно упрощается за счет наследования: мы создаем класс Измеритель, в который помещаем эти повторяющиеся поля, а затем наследуем от него наши классы Термометр, Весы и Барометр (рис. 19.6).

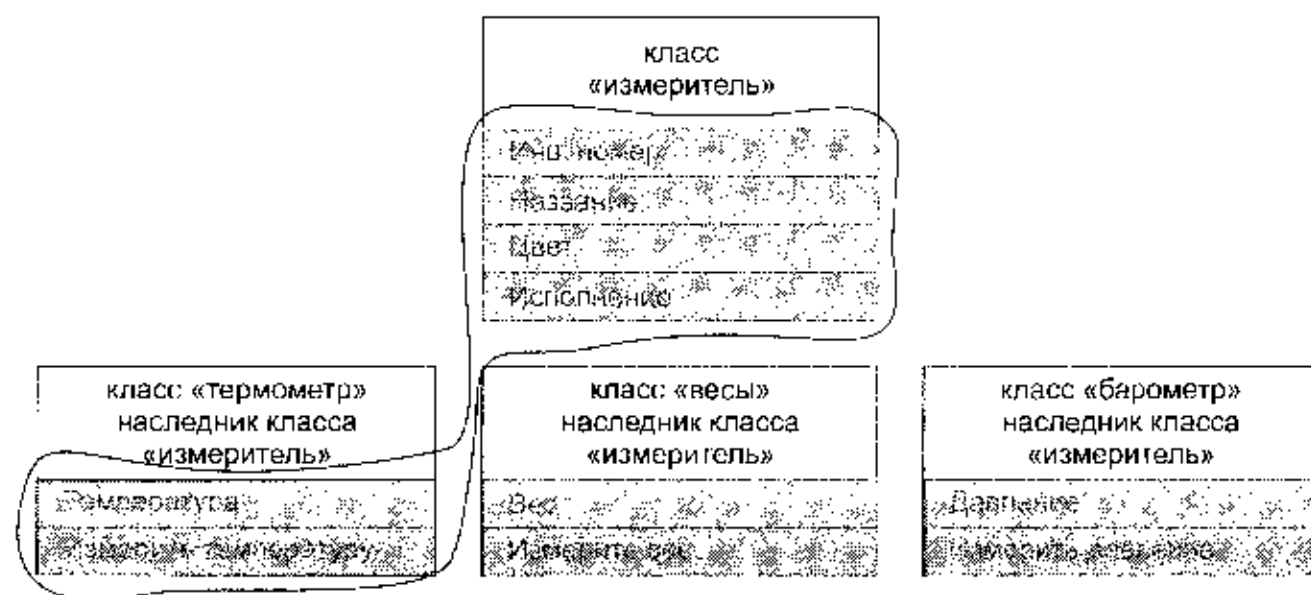


Рис. 19.6. Наследование

Широкая кривая на рисунке показывает, что при наследовании поля и методы базового класса включаются в состав класса потомка.

19.3.3. Полиморфизм

Полиморфизм — это способность объектов выполнять одну и ту же команду разными способами.

Полиморфизм основывается на двух свойствах, которые реализуют объектно-ориентированные языки программирования: способности трактовать объект класса-наследника как тип базового класса и способности класса иметь виртуальные методы. Особенность виртуального метода состоит в том, что он надежно связан с созданным экземпляром класса.

Пример. Если добавить в каждый из классов (в класс Измеритель и каждый из его наследников) метод Сигнализировать, а в реализации этого метода описать бы для каждого класса свой способ подавать сигнал (для термометра — мигать, для весов — издавать звук сирены, а для барометра — свистеть), то мы могли бы всем объектам, потомкам класса Измеритель, дать команду: измерители, сигнализировать! И каждый из них подал бы сигнал: термометры бы замигали, весы издали звук сирены, а барометры засвистели. Это и есть результат работы виртуальных методов.

19.3.4. Инкапсуляция

Инкапсуляция — это способность класса скрывать от внешнего мира детали внутренней реализации.

Инкапсуляция реализуется в объектно-ориентированном программировании при помощи областей видимости. Области видимости обозначаются в структуре класса специальными служебными словами. Например, слово `private` означает, что

все поля и методы, объявленные в этой области видимости, невидимы для внешнего мира, а `public`, наоборот, — что все, объявленное в этой зоне, для внешнего мира открыто. Таким образом, у класса появляется своего рода защитный корпус, предохраняющий от излишней перегрузки программы объявленными именами и методами. Наружу класс экспонирует только то, что составляет его интерфейс и предназначено для внешнего взаимодействия. Все остальные поля и методы он скрывает внутри, помещая их в область видимости `private`.

19.4. Инструментальные средства и среды разработки программного обеспечения

Система программирования включает в себя следующий набор инструментов:

- ☐ транслятор (компилятор или интерпретатор);
- ☐ отладчик;
- ☐ средства оптимизации кода (профилировщик);
- ☐ редактор связей (линковщик);
- ☐ набор библиотек;
- ☐ сервисные средства (утилиты).

Об *инструментальной среде пользователя* можно говорить в случае, когда средства программирования включаются в прикладной программный пакет (например, СУБД). В этом случае состав инструментов может быть таким:

- ☐ библиотека функций, процедур, объектов и методов обработки;
- ☐ макрокоманды;
- ☐ клавиатурные макросы;
- ☐ языковые макросы;
- ☐ конструкторы экранных форм;
- ☐ генераторы приложений;
- ☐ языки запросов высокого уровня;
- ☐ конструкторы запросов по шаблонам.

Интегрированные среды разработки (Integrated Development Environment, IDE) объединяют в одном приложении все или большинство инструментальных средств с целью ускорения разработки программ и удобства использования инструментов на разных этапах создания программного обеспечения. Обычно отличительной чертой современных интегрированных сред разработки является наличие большого количества различных сервисных возможностей, позволяющих программисту максимально сосредоточиться на реализации алгоритма работы программы. К таким возможностям относятся:

- ☐ инструменты визуального создания пользовательского интерфейса;
- ☐ средства разметки кода;

- средства облегчения редактирования (автоматическое завершение имен и служебных слов, автоматическая вставка конструкций языка и шаблонов кода, подсветка синтаксиса, синтаксическая проверка во время ввода и пр.);
- интегрированные средства отладки с детальной настройкой точек останова и разными режимами отслеживания значений переменных и объектов;
- встроенные средства документирования исходного кода и проекта в целом;
- средства переработки написанного кода с целью улучшения его внутренней структуры, расширения функциональности или приведения в соответствие с изменениями в модели, разработанной системным архитектором;
- средства интеграции с инструментами жизненного цикла ПО;
- средства коллективной разработки;

Наиболее известными интегрированными средами разработки в операционной системе Microsoft Windows являются Microsoft Visual Studio и CodeGear RAD Studio.

На сегодняшний день Microsoft Visual Studio является общепризнанным лидером и лучшей интегрированной средой разработки для семейства операционных систем Windows (рис. 19.7). Богатый набор визуальных компонентов и возможность одинаково эффективно создавать приложения на любом языке программирования делает Microsoft Visual Studio поистине уникальным продуктом.

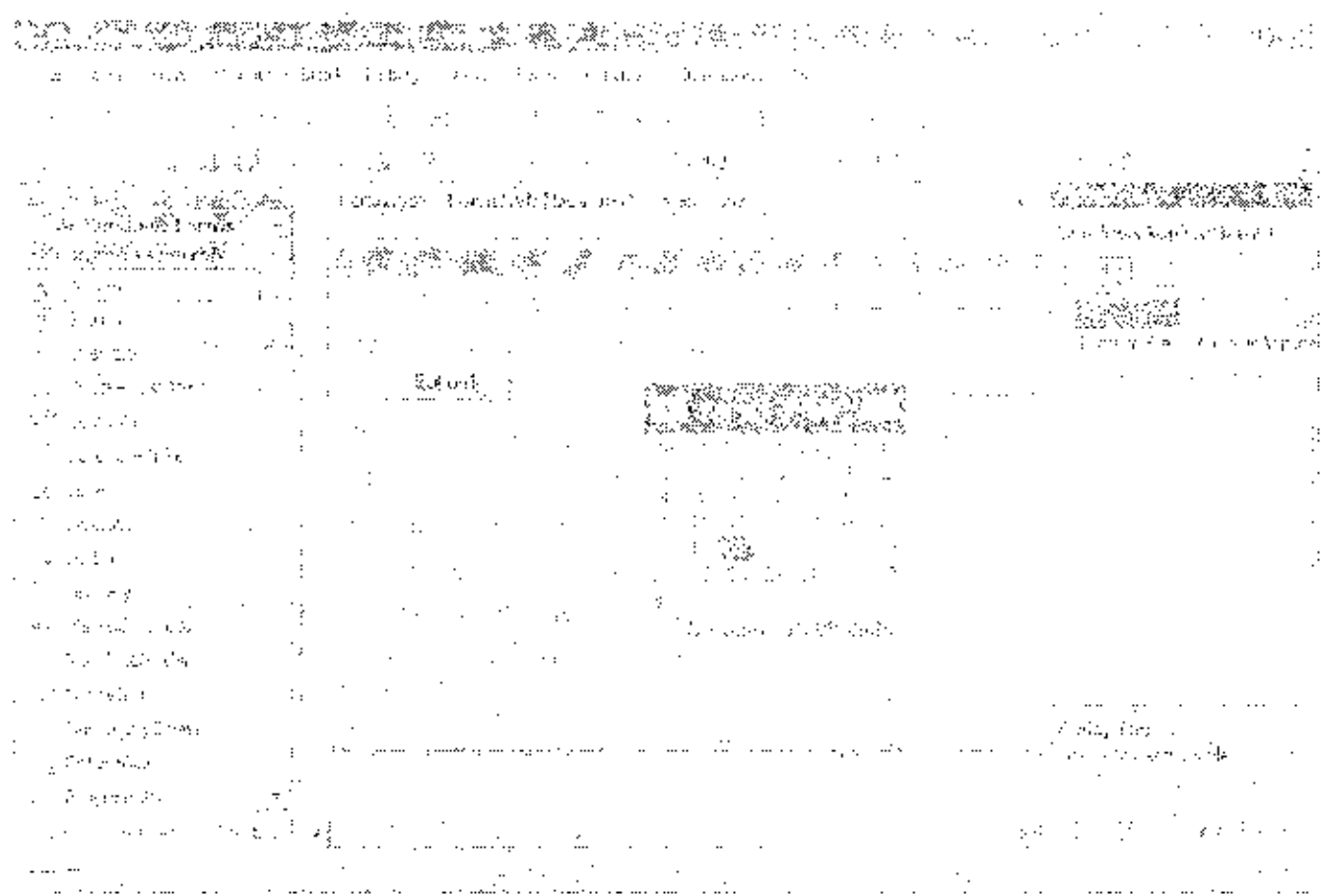
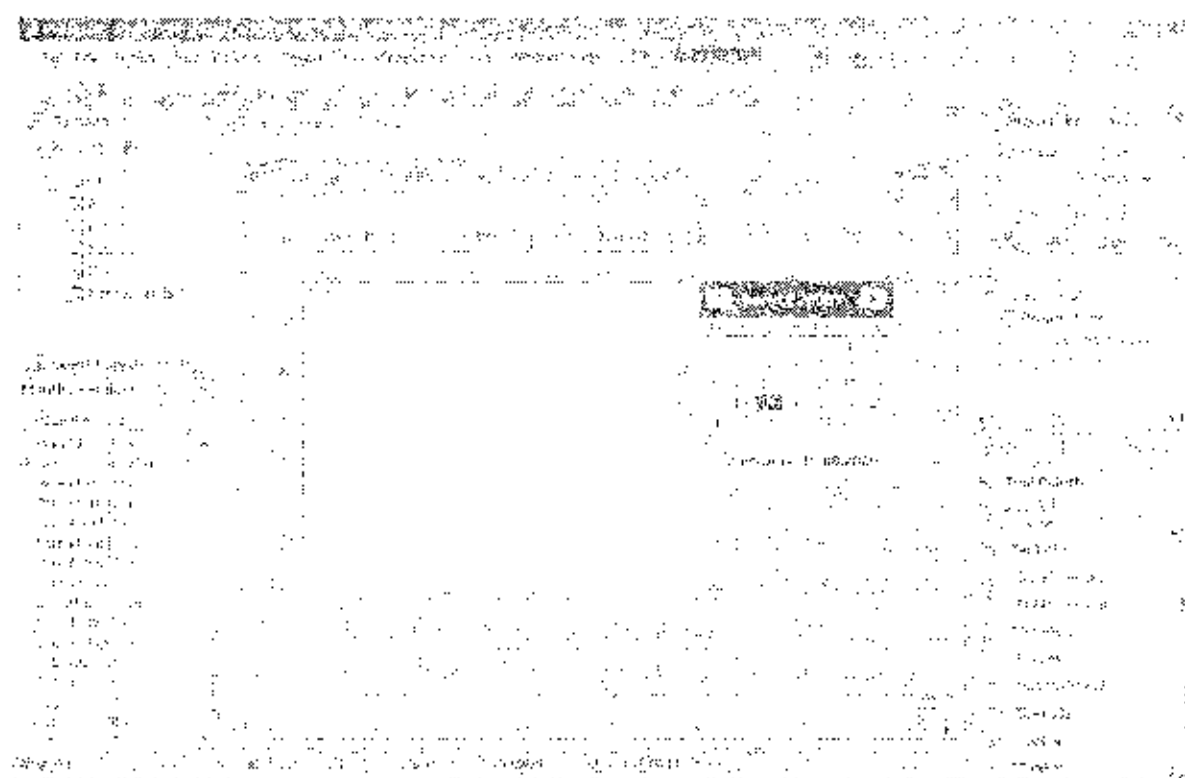
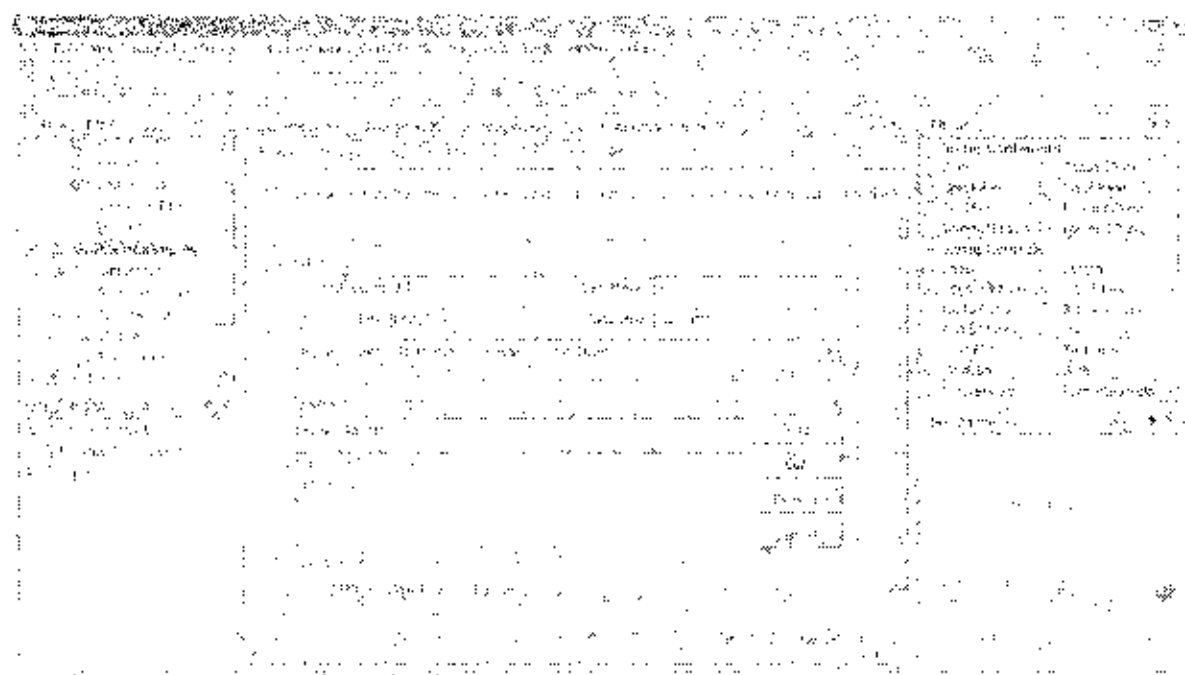


Рис. 19.7. Microsoft Visual Studio

Отличительной чертой CodeGear RAD Studio является полная интеграция этой среды со всеми продуктами поддержки жизненного цикла ПО производства корпорации Borland (рис. 19.8).

**Рис. 19.8.** CodeGear RAD Studio

Достаточно успешно конкурируют с коммерческими продуктами интегрированные среды быстрой разработки приложений из мира свободного ПО: NetBeans, KDevelop и Eclipse по своей функциональности уже могут соперничать с RAD и Visual Studio. Сравнивать их трудно, так как в разных плоскостях лежат их интересы. На рис. 19.9 показана интегрированная среда разработки NetBeans.

**Рис. 19.9.** NetBeans

19.5. Жизненный цикл программного обеспечения

Основными этапами жизненного цикла программного обеспечения являются:

- ❑ определение требований и разработка спецификаций;
- ❑ проектирование;
- ❑ кодирование;
- ❑ тестирование;
- ❑ внедрение и сопровождение.

В зависимости от того, в каком порядке и с какой частотой выполняются эти этапы, выделяют три основных модели разработки программного обеспечения: модель водопада, итерационную модель и спиральную модель. Существуют и другие модели.

В *модели водопада* к пройденному однажды этапу уже не возвращаются. Каждый предыдущий этап должен быть полностью выполнен, после чего управление проектом передается на последующий этап. Модель водопада представлена на рис. 19.10.

В *итерационной модели* возврат на предыдущую ступень разработки программного обеспечения возможен на любом этапе (рис. 19.11).

В *спиральной модели* передача управления происходит только вперед, но при достижении последнего этапа происходит переход вновь на первый, но уже на новом уровне качества, и далее опять вперед и т. д. Таким образом формируется восходящая спираль разработки (рис. 19.12). Такая модель разработки свойственна программным проектам промышленного масштаба, рассчитанным на массового пользователя (например, операционные системы).

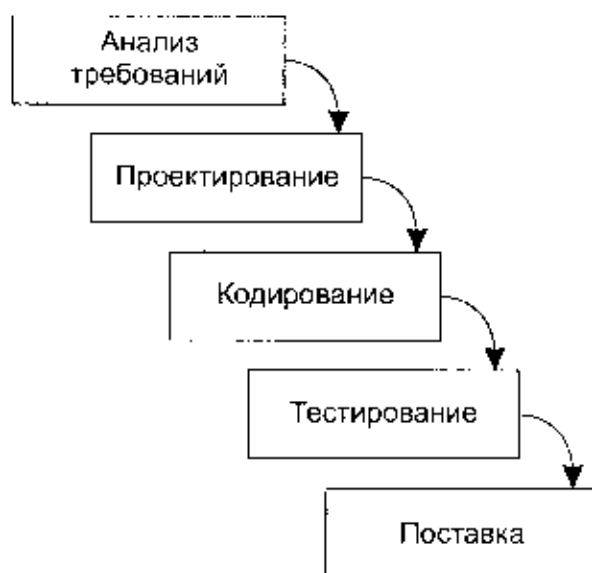


Рис. 19.10. Модель водопада

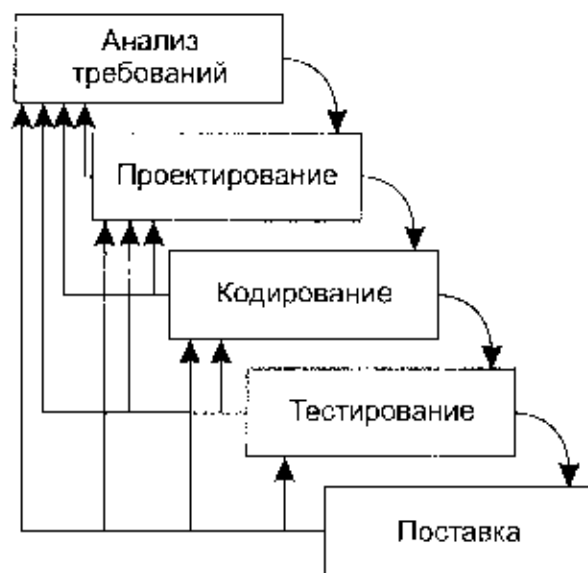


Рис. 19.11. Итерационная модель

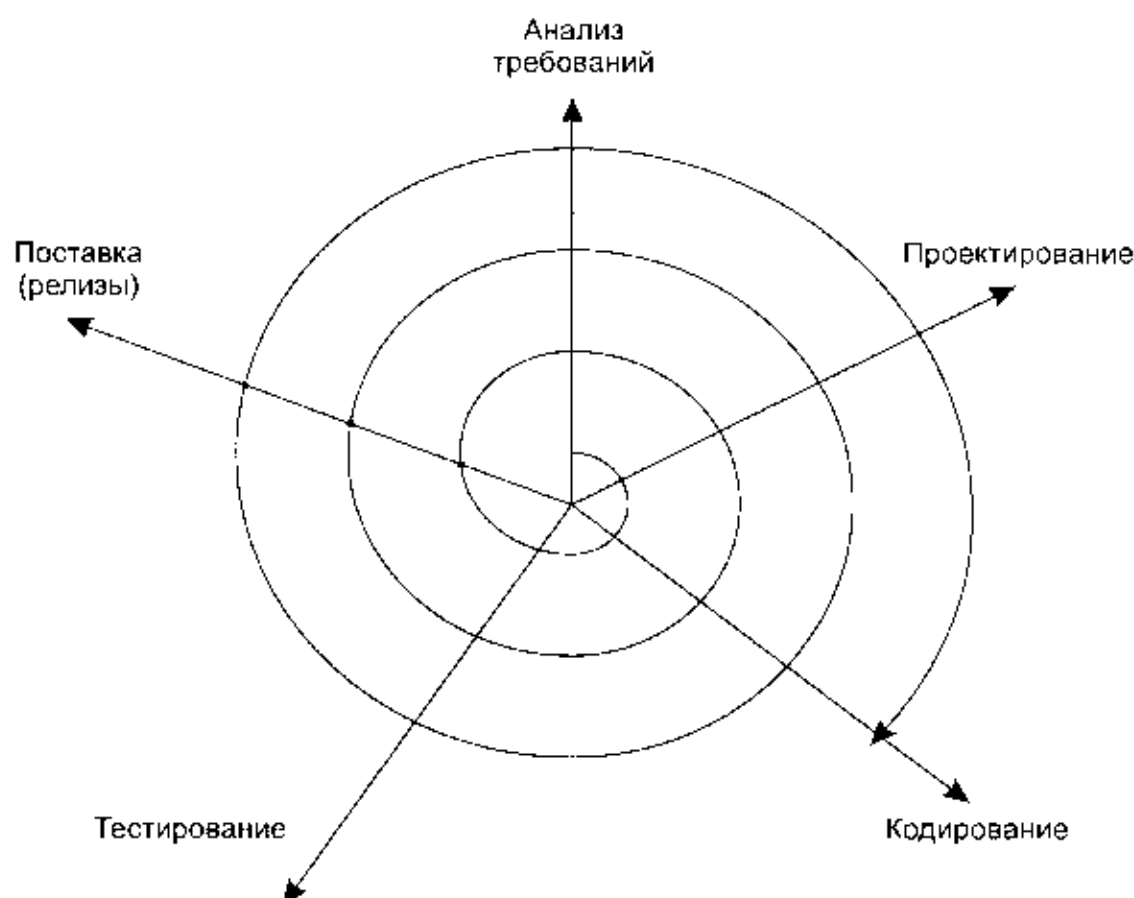


Рис. 19.12. Спиральная модель разработки

19.5.1. Анализ требований и разработка спецификаций

Прежде чем приступить к разработке программного продукта, необходимо сформировать требования, которым он должен соответствовать. Сбором, анализом и оформлением требований, как и разработкой спецификаций, занимается системный аналитик.

Лица у которых собираются требования:

- ☐ заказчик (инвестор);
- ☐ руководитель подразделения, в котором будет внедряться создаваемая программа;
- ☐ потенциальные пользователи программы;
- ☐ исполнители программного проекта;
- ☐ потенциальные пользователи результатов работы программы;
- ☐ консультанты.

В требованиях обязательно обнаруживаются противоречия и проблемы. Эти противоречия нужно выявить и согласовать до начала работ следующей стадии проектирования. Для этого нужно

- ☐ выработать соглашение о наличии проблемы;
- ☐ выделить основные причины возникновения проблемы;

- выявить заинтересованных лиц и пользователей;
- определить границы системы решений;
- выявить ограничения, которые необходимо наложить на решение.

Несмотря на то, что требования формируются до начала программного проекта, нужно помнить, что они будут сопровождать работу на каждом этапе разработки ПО. Поэтому требования должны строго документироваться. Лучше всего использовать для этого специальное программное обеспечение управления требованиями, например, Borland CaliberRM. Эта программа позволяет не только документировать требования, но и декомпозировать их до конкретных спецификаций и метрик, а также вывести граф метрик, показывающий сбалансированность собранных требований. Кроме того, Borland CaliberRM поддерживает режим коллективной работы над требованиями на протяжении всего жизненного цикла ПО (рис. 19.13).

На этапе анализа требований необходимо предъявить пользователям прототипы пользовательского интерфейса программного продукта. Расположение элементов управления, дизайн, эргономика и функционал пользовательского интерфейса — это то, что можно согласовать до начала проектирования.

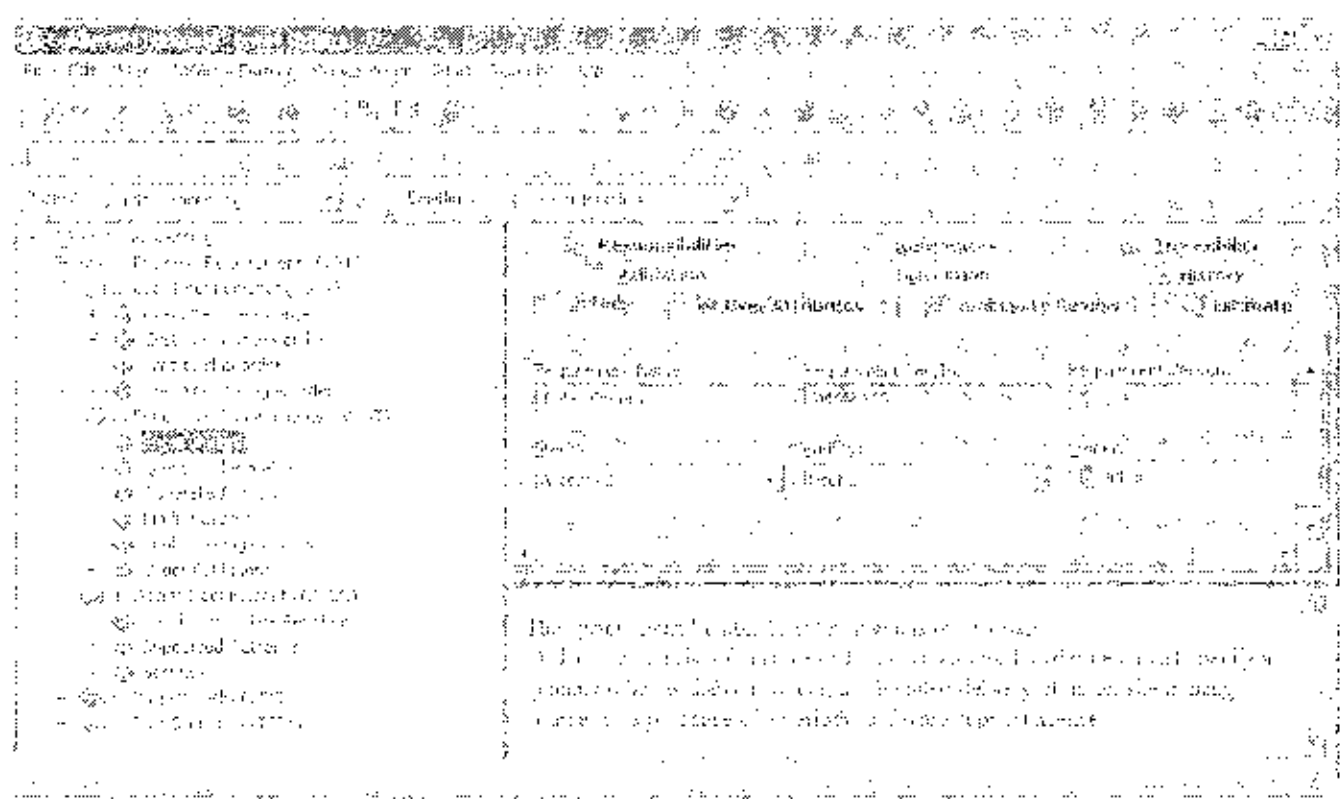


Рис. 19.13. Borland CaliberRM

19.5.2. Проектирование

Работу по проектированию при объектно-ориентированном подходе выполняет системный архитектор средствами языка UML. Задачи проектирования включают в себя две составляющие: разработку физической и логической структур. Логическое проектирование заключается в разработке классов для реализации

их экземпляров — объектов. Для этого требуется подробное описание полей и методов классов, а также связей между ними. Статические диаграммы классов и пакетов являются основными в UML-проектировании, поскольку именно они обеспечивают генерацию шаблонов исходного кода для следующего этапа, кодирования. Динамические диаграммы предназначены для более полного понимания программистом логики функционирования системы и взаимоотношения классов. Для проектирования физической структуры, предназначены UML диаграммы компонентов развертывания. Эти диаграммы описывают физическое размещение компонентов системы на аппаратном обеспечении заказчика.

Для повышения эффективности этапа проектирования нужно, чтобы программный пакет создания архитектуры был совместим со средством разработки кода. Если в качестве среды разработки выбрана Delphi, то системный архитектор может использовать пакет Together производства Borland, полностью совместимый с этой средой (рис. 19.14).

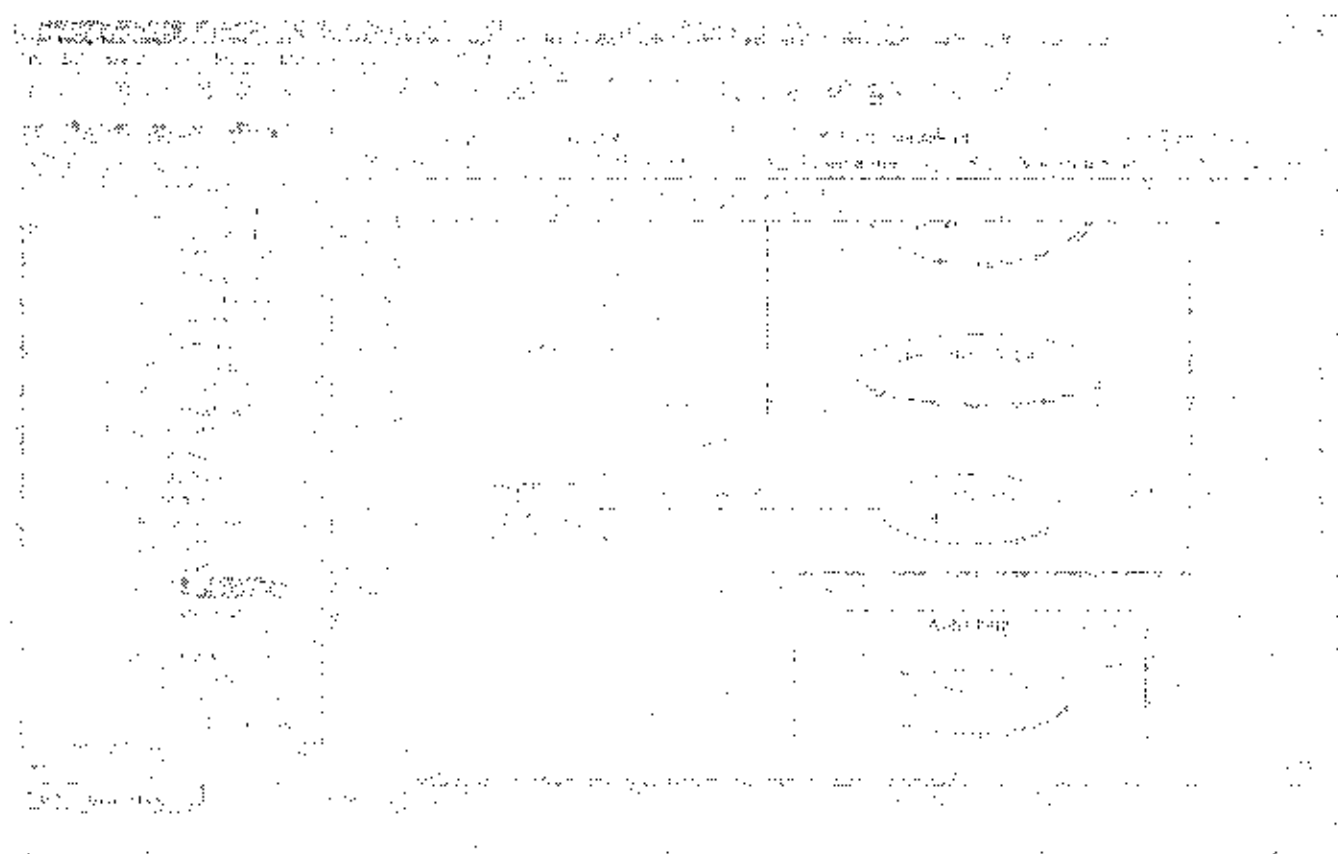


Рис. 19.14. Borland Together

19.5.3. Кодирование и тестирование

Разработка исходного кода ведется в интегрированной среде быстрой разработки, такой как Delphi или Microsoft Visual Studio, на основе полученных от системного архитектора UML-диаграмм и автоматически сгенерированного по этим диаграммам шаблонам исходного кода.

Тестирование программного обеспечения — это та ступень разработки ПО, на которую в последнее время обращается все более пристальное внимание. Этап те-

стирования, если он организован недостаточно хорошо, может вызывать затраты, равные половине всех затрат на создание ПО.

В процессе тестирования используются данные, характерные для системы в рабочем состоянии, то есть данные для тестирования нельзя выбирать случайным образом. План проведения испытаний должен быть составлен заранее, а большую часть тестовых данных нужно определить еще на этапе проектирования системы.

Тестирование подразделяется на три стадии: автономное, комплексное и системное.

При *автономном* тестировании каждый модуль подвергается испытаниям при помощи данных, подготовленных самим программистом. При этом программная среда модуля имитируется при помощи программы управления тестированием, содержащей фиктивные программы вместо реальных подпрограмм, к которым имеются обращения из данного модуля. Подобную процедуру иногда называют программным тестированием, а программу, подлежащую тестированию, — тестируемой программой. Модуль, прошедший автономное тестирование, подвергается комплексному тестированию.

В процессе *комплексного* тестирования производится совместная проверка групп программных компонентов. Как результат получают полностью проверенную систему. На данной стадии тестирования часто обнаруживаются ошибки, пропущенные на предыдущей стадии тестирования. Исправление этих ошибок может потребовать до четверти общих затрат.

Системное, или *оценочное*, тестирование завершает этап проверки системы, то есть завершает испытание системы в целом при помощи независимых тестов. Независимость тестов является существенным обстоятельством. Заказчик при приемке может настоять на проведении собственного системного тестирования. Для случая, когда сравниваются характеристики нескольких систем (например, когда требуемое ПО поставляется несколькими изготовителями), такая процедура известна как *сравнительное тестирование*.

Важные критерии тестирования:

- ☐ для программы необходимо располагать набором тестовых данных, позволяющих установить, что программа работает правильно по любой ветви;
- ☐ каждая инструкция программы должна быть выполнена по крайней мере один раз для заданного набора тестовых данных;
- ☐ каждая ветвь в программе должна быть испытана хоть один раз с использованием набора тестовых данных.

Вопросы для самопроверки

1. Дайте определение языку программирования.
2. Что такое «компьютерная программа»?
3. В чем заключается процесс программирования?
4. Что такое «трансляция», и из каких этапов она состоит?

5. Создается ли в результате работы компилятора программа, которую можно загрузить в постоянную память компьютера и выполнить?
6. Что такое «машинный код»?
7. Что определяет синтаксис языка программирования?
8. Что определяет семантика языка программирования?
9. Какие требования предъявляются к описанию современных языков программирования?
10. Дайте определение переменной и константы.
11. В чем сходство и различие функции, процедуры и оператора?
12. По каким признакам вы можете классифицировать языки программирования?
13. В чем состоит отличие структурного программирования от модульного?
14. Назовите семь этапов развития языков программирования.
15. Назовите четыре концепции объектно-ориентированного программирования.
16. Дайте определение абстракции.
17. Какими структурами в объектно-ориентированном программировании реализуются абстракция и наследование?
18. Какие инструменты включает в себя система программирования?
19. Какими возможностями обладают современные интегрированные среды разработки?
20. Что такое «переработка» (рефакторинг) кода?
21. Из каких этапов состоит жизненный цикл программного обеспечения?
22. Какие модели разработки вам известны?
23. В каком случае применяется спиральная модель разработки?

Литература

1. Браудэ Э. Технология разработки программного обеспечения. СПб.: Питер, 2004.
2. Гагарина Л. Г., Кокорева Е. В., Виснадул Б. Д. Технология разработки программного обеспечения. М.: ФОРУМ-Инфра-М, 2008.
3. Кериевски Дж. Рефакторинг с использованием шаблонов. М.: ООО «И. Д. Вильямс», 2006.
4. Константи́н Л., Локеуд Л. Разработка программного обеспечения. СПб.: Питер, 2004.

Наталья Владимировна Макарова, Владимир Борисович Волков
Информатика: Учебник для вузов

Заведующий редакцией
Руководитель проекта
Ведущий редактор
Литературный редактор
Художественный редактор
Корректор
Верстка

*А. Кривцов
А. Юрченко
Ю. Сергиенко
А. Жданов
Л. Адуевская
И. Тимофеева
Е. Егорова*

Подписано в печать 15.04.11. Формат 70х100/16. Усл. п. л. 56,76. Тираж 4000. Заказ 5690.
ООО «Питер Пресс», 198206, Санкт-Петербург, Петергофское шоссе, 73. лит. А29.
Налоговая льгота — общероссийский классификатор продукции ОК 005-93, том 2; 95 3005 — литература учебная
Отпечатано в соответствии с предоставленными материалами
в ЗАО «ИПК Парето-Принт», г. Тверь, www.pareto-print.ru