МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ЯДЕРНЫЙ УНИВЕРСИТЕТ «МИФИ»

Г.П. Аверьянов, В.В. Дмитриева

СОВРЕМЕННАЯ ИНФОРМАТИКА

Рекомендовано УМО «Ядерные физика и технологии»
в качестве учебного пособия
для студентов высших учебных заведений

Москва 2011

УДК 004(075)
ББК 32.97я7
А19

Аверьянов Г.П., Дмитриева В.В. **СОВРЕМЕННАЯ ИНФОРМАТИКА:** *Учебное пособие.* М.: НИЯУ МИФИ, 2011. — 436 с.

Учебное пособие посвящено основополагающим принципам, современному состоянию и перспективам развития компьютерных систем – основе современной информатики.

Достаточно подробно рассмотрены технические средства информатики, классификация, архитектура и структура компьютеров и компьютерных сетей. Даны современное состояние и тенденции развития кремниевых технологий, направления развития как настольных систем – персональных компьютеров, рабочих станций и серверов, так и мобильных систем. Изложены различные виды компьютерных сетей – локальных, глобальных и беспроводных.

Значительное внимание уделено интенсивно развивающимся в последнее время суперкомпьютерам, области их применения и динамике развития архитектуры.

Представлена классификация современных языков программирования и структура традиционных (императивных) языков. Затронуты вопросы состава, структуры и разновидностей программного обеспечения, актуальные вопросы защиты информации.

Пособие предназначено для студентов факультета автоматики и электроники и вечернего факультета.

Подготовлено в рамках Программы создания и развития НИЯУ МИФИ.

Рецензент канд. техн. наук, доц. С.Д. Чигирь (1-й МГМУ им. И.М. Сеченова)

ISBN 978-5-7262-1421-4

© *Национальный исследовательский ядерный университет «МИФИ», 2011*

Редактор М.В. Макарова

Подписано в печать 15.12.2010. Формат 60x84 1/16
Уч.-изд.л. 28,0. Печ.л. 27,25. Тираж 260 экз.
Изд.. № 1/1/113. Заказ № 6

Национальный исследовательский ядерный университет «МИФИ».
115409, Москва, Каширское ш., 31

ООО «Полиграфический комплекс «Курчатовский».
144000, Московская область, г. Электросталь, ул. Красная, д. 42

О Г Л А В Л Е Н И Е

ПРЕДИСЛОВИЕ 7
ОСНОВНЫЕ СОКРАЩЕНИЯ 14
ВВЕДЕНИЕ 16
ГЛАВА 1. АЛГОРИТМЫ И ОСНОВЫ АЛГОРИТМИЗАЦИИ 20
1.1. Элементы алгебры логики 20
1.2. Элементы исчисления предикатов 24
1.3. Алгоритмы 27
1.4. Проблемы алгоритмизации 32
1.5. Математическое моделирование – вычислительный эксперимент 34
Контрольные вопросы 36
ГЛАВА 2. ТЕХНИЧЕСКИЕ СРЕДСТВА ИНФОРМАТИКИ 36
2.1. Структура современных систем обработки данных 36
2.2. Общие сведения о структуре компьютера и краткая характеристика его составных частей 37
2.3. Типы и характеристики компьютерной графики 72
2.4. Архитектура компьютеров, понятие об интерфейсах и их разновидности 76
2.5. Классификация компьютеров, краткие характеристики суперкомпьютеров, мейнфреймов и мини-компьютеров 82
Контрольные вопросы 110
ГЛАВА 3. МИКРОПРОЦЕССОРЫ, МИКРОКОМПЬЮТЕРЫ. ПЕРСОНАЛЬНЫЕ КОМПЬЮТЕРЫ, РАБОЧИЕ СТАНЦИИ, СЕРВЕРЫ И СУПЕРСЕРВЕРЫ 111
3.1. Микропроцессоры – эволюция, классификация, перспективы 111
3.2. Персональные компьютеры. Краткая история, функциональные возможности, номенклатура 121
3.3. Архитектура ПК, системные и локальные шины, состав периферийных устройств 124
3.4. Рабочие станции, серверы и суперсерверы 133
3.5. О мобильных компьютерных устройствах 140
Контрольные вопросы 144

ГЛАВА 4. СЕТИ КОМПЬЮТЕРОВ И СРЕДСТВА ТЕЛЕКОММУНИКАЦИОННОГО ДОСТУПА.....145
4.1. Эволюция распределенных вычислительных систем.....145
4.2. Способы передачи данных по физическим линиям.....147
4.3. Понятие подхода открытых систем, проблемы стандартизации, многоуровневый подход, стек протоколов.....154
4.4. Глобальные вычислительные сети, методы доступа, современное состояние и ближайшие перспективы.....161
4.5. Internet: краткая история, принципы построения, адресация, основные приложения.....174
4.6. Локальные вычислительные сети, разновидности протоколов канального уровня, технические средства, используемые ЛВС.....182
Контрольные вопросы.....199
ГЛАВА 5. ЛИНГВИСТИЧЕСКИЕ СРЕДСТВА КОМПЬЮТЕРОВ.....201
5.1. Основные тенденции в развитии языков программирования.....201
5.2. Классификация и краткая характеристика современных языков программирования.....206
5.3. Структура и сравнительные характеристики процедурно-ориентированных (императивных) языков программирования.....214
Контрольные вопросы.....236
ГЛАВА 6. ПРОГРАММНЫЕ СРЕДСТВА СОД.....237
6.1. Общие сведения, классификация программного обеспечения и краткая характеристика отдельных частей.....237
6.2. Операционные системы, их эволюция, состав и функциональное назначение отдельных частей.....241
6.3. Разновидности построения ОС для различных СОД, сетевые операционные системы.....254
Контрольные вопросы.....260

ГЛАВА 7. ОРГАНИЗАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРАХ IBM-КЛОНА 261
7.1. Общая структура программного обеспечения 261
7.2. Краткая характеристика ОС, применяемых в ПК 264
7.3. Инструментальное программное обеспечение ПК 271
7.4. Пакеты прикладных программ 274
7.5. Программное обеспечение компьютерной графики 281
Контрольные вопросы 284
ГЛАВА 8. ОРГАНИЗАЦИЯ, ХРАНЕНИЕ И ПОИСК ДАННЫХ 286
8.1. Структуры данных 286
8.2. Типичные виды работ с данными 293
8.3. Организация хранения данных 300
8.4. Базы данных 302
8.5. Системы управления базами данных 308
8.6. Автоматизированные информационные системы 310
Контрольные вопросы 314
ГЛАВА 9. ТЕХНОЛОГИЯ ОБРАБОТКИ ТЕКСТОВОЙ И ГРАФИЧЕСКОЙ ИНФОРМАЦИИ В ПК IBM-КЛОНА 315
9.1. Функции текстовых редакторов 315
9.2. Сравнительные характеристики распространенных текстовых редакторов 316
9.3. Типы и характеристики компьютерной графики 319
9.4. Программное обеспечение компьютерной графики 323
Контрольные вопросы 327
ГЛАВА 10. ОФИСНЫЕ ТЕХНОЛОГИИ ИНФОРМАТИКИ 328
10.1. Виды офисных работ с использованием компьютеров 328
10.2. Электронные таблицы 329
10.3. Автоматическое распознавание текстов 332
10.4. Автоматизированный перевод документов 333
10.5. Подготовка презентаций 335
10.6. Электронные календари и скоросшиватели 337

10.7. Технология офисной работы в сети Internet.....339
10.8. Экспертные системы 343
10.9. OLE-технология подготовки составных документов.....347
Контрольные вопросы.....349
ГЛАВА 11. ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ.....350
11.1. Проблемы информационной безопасности.....350
11.2. Защита информации от непреднамеренных воздействий.....352
11.3. О некоторых понятиях компьютерной безопасности.....355
11.4. Основные методы защиты от удаленных и локальных атак.....362
11.5. Компьютерные вирусы.....373
11.6. Криптографические методы защиты информации.....378
11.7. Бизнес в Интернете.....385
11.8. Правовое обеспечение информационной безопасности.....390
Контрольные вопросы.....395
ГЛАВА 12. БЕСПРОВОДНЫЕ СИСТЕМЫ И СЕТИ ПЕРЕДАЧИ ДАННЫХ 396
12.1. Беспроводные среды.....396
12.2. Системы радиосвязи.....400
12.3. Спутниковая система связи 406
12.4. Система подвижной радиосвязи.....411
12.5. Беспроводное подключение узлов в локальных сетях 423
Контрольные вопросы.....428
СПИСОК ТЕРМИНОВ И ОПРЕДЕЛЕНИЙ 430
СПИСОК ЛИТЕРАТУРЫ.....436

ПРЕДИСЛОВИЕ

Существует большое количество учебников по информатике и их написание продолжается. В чем же причина этого изобилия?

Основы современной информатики были сформулированы еще в 40-х годах прошлого столетия и обычно связываются с именами таких ученых, как Джон (Янош) фон Нейман, Алан Тьюринг и др. К сожалению, очень редко упоминаются наши, отечественные, не менее значимые фигуры, как Лебедев С.А., Глушков В.М., Марков А.А., Ляпунов А.А. и многие др.

Следует отметить, что основы информатики пока не подвергаются серьезным изменениям. Тем не менее, развитие этой науки и расширение ее сферы влияния превзошло все мыслимые и самые оптимистические прогнозы и связано с колоссальными технологическими достижениями.

Все это приводит к расширению круга лиц вовлеченных в изучение этой дисциплины, а постоянное совершенствование технических, лингвистических и программных средств информатики (особенно за последние 60 лет) требует постоянных дополнений в изучении и описании информатики.

Чтобы понять революционные изменения, произошедшие за последние 40 – 50 лет, и повсеместное внедрение их в самые разнообразные области человеческой деятельности, необходимо провести небольшой экскурс в эволюцию традиционных компьютерных систем за этот период.

Так, в 60 – 80-х годах прошлого века сами компьютеры (технические средства) были отделены от пользователя и обслуживались специалистами-профессионалами по различным компонентам машин (по процессору, различным периферийным устройствам и т.п.). То же самое имело место и в разделе программного обеспечения. Работа трансляторов языков программирования, операционных систем, различных библиотечных программ обеспечивалась специальными группами людей, которые (по недоразумению) назывались математиками, а отделы назывались математическими. У пользователя не было необходимости разбираться в этих средствах. Он должен был знать сами языки программирования и команды ОС и трансляторов. Основное направление использования компьютеров – научно-технические расчеты.

В связи с этим для рядового пользователя изучение информатики ограничивалось изучением курса программирования. Этот курс считался традиционным, как правило, включая изучение языков программирования, ориентированных на различные предметные области, а также особенности их реализации на различных типах больших компьютеров (в современной терминологии – мейнфреймы).

С середины 70-х годов в связи с появлением мини-компьютеров, а в дальнейшем и персональных компьютеров и вычислительных сетей ситуация резко изменилась. Техника приблизилась к конечному пользователю (значительно расширилась сфера использования компьютеров) и потребовала от него определенных знаний, которые прежде были доступны только профессионалам, обслуживающим большие компьютеры. В большинстве случаев пользователь сам должен определять стратегию выбора и приобретения технических средств. К тому же значительно расширился спектр средств вычислительной техники. С появлением 16-разрядного микропроцессора, а затем 32- и 64-разрядных, кроме повсеместно распространенных персональных компьютеров, в рамках серьезных научно-технических и инженерных применений получили широкое распространение высокопроизводительные рабочие станции, а также серверы и так называемые суперсерверы – сетевые машины коллективного использования.

Появление новой элементной базы оказало значительное влияние на развитие мейнфреймов, которые успешно развиваются до последнего времени, несмотря на неоднократные предположения об их неминуемой гибели.

Высокую динамику демонстрирует рынок суперкомпьютеров. Это один из наиболее быстро развивающихся видов вычислительных систем. Новые архитектурные подходы при создании этих самых мощных компьютеров связаны с так называемыми кластерными системами и обязаны также интенсивному развитию твердотельной интегральной электронной техники. Широкое распространение этих машин связано не только с традиционными научно-техническими применениями, но также с использованием их в промышленности (автомобильная, авиационная, металлургическая и т.п.), для управления крупными базами данных, для принятия глобальных стратегических решений и т.п.

Производительность суперкомпьютеров постоянно растет, что связано, с одной стороны, с постоянно возрастающими потребностями приложений, а с другой – постоянный рост производительности компьютеров нижнего уровня, использующих основные новации разработчиков суперкомпьютеров. Наряду с традиционным использованием суперкомпьютеров в научных и инженерных расчетах они применяются при работе с базами данных, в криптографии, биологии и медицине исследования (расшифровке) генома живых организмов, моделировании человеческого мозга, медицинских экспертных систем и обработке изображений.

Безусловно, все эти сведения необходимо знать современному инженеру.

Вопросы языков программирования (лингвистического обеспечения) также имеют очень важное значение при освоении средств вычислительной техники рядовым пользователем. В этом разделе информатики также происходят значительные изменения в связи с расширением сферы использования компьютеров, разработкой современных программных интерфейсов, особенностями различных аспектов сетевого программирования и т.п.

Развитие языков программирования происходит по ряду направлений. С одной стороны, происходит значительное расширение возможностей таких традиционных (императивных) языков программирования, как FORTRAN, Pascal, C, которые иногда называют языками высокого уровня, так как они не связаны с кодом конкретной машины, но, тем не менее, они ориентированы на так называемую «архитектуру фон Неймана» и связаны с кодированием предварительно разработанного математического алгоритма решаемой задачи. При реализации программы на компьютере требуются программные средства для перевода ее в код конкретного компьютера – трансляторы. Для пользователя также требуются определенные усилия по освоению таких языков, так как они далеки от языков человеческого общения.

С другой стороны, развитие языков связано с переходом на более высокий уровень, не зависимый от архитектуры компьютера, и с исключением средств традиционного программирования.

К таким средствам можно отнести так называемые функциональные языки и языки логического программирования (LISP, Prolog, OCaml). Эти языки имеют пока очень ограниченное приме-

нение, хотя в определенных случаях оказываются достаточно эффективными.

Гораздо большее (массовое) распространение получили непроцедурные (так называемые дескриптивные) языки, которые также можно отнести к языкам сверхвысокого уровня. Это языки управления заданиями, языки пакетов прикладных программ, ориентированные на наиболее распространенные области применений, – языки электронных таблиц, систем управления базами данных, математические пакеты, экспертные системы и т.п.

Однако как в первом, так и во втором случае полностью исключить традиционное, процедурно-ориентированное программирование не удается, и подобные языки включают в свой состав его элементы. Так, известная фирма Microsoft включает в свои основные приложения язык Visual Basic, что позволяет пользователю самостоятельно и неограниченно расширять функциональные возможности приложений. Различные средства традиционного программирования включают в свои пакеты и другие фирмы. В связи с этим изучение средств программирования, развивающегося уже более 50-ти лет, по-прежнему актуально. В предлагаемом пособии описаны базовые элементы таких языков, их структура и направления развития, прошедшие за последние 50 лет (см. гл. 5).

Программное обеспечение современных вычислительных систем включает общесистемное, системное, инструментальное и прикладное обеспечение и по своему объему на несколько порядков превышает средства, устанавливаемые на первых персональных компьютерах. При этом программы приложения, которые непосредственно решают задачу пользователя, как правило, составляют незначительную долю программного обеспечения, установленного на компьютере. Однако пользователь должен иметь ясное представление обо всех программах, которые имеются на его компьютере для эффективного ее использования.

И в заключение следует отметить, что развитие современных компьютерных систем определяется (в значительной степени) развитием технологии кремниевых чипов. Эта технология развивается уже почти 60 лет. Один из отцов-основателей фирмы Intel Гордон Мур вывел в 1965 г. закономерность, позволяющую производителям приблизительно предвидеть перспективы: «количество транзисторов, которые могут быть размещены на чипе будут удваиваться каждые два года», а

в 1975 г. Мур уточнил свой прогноз: «Плотность схем полупроводникового микрочипа удваивается каждые 18 месяцев».

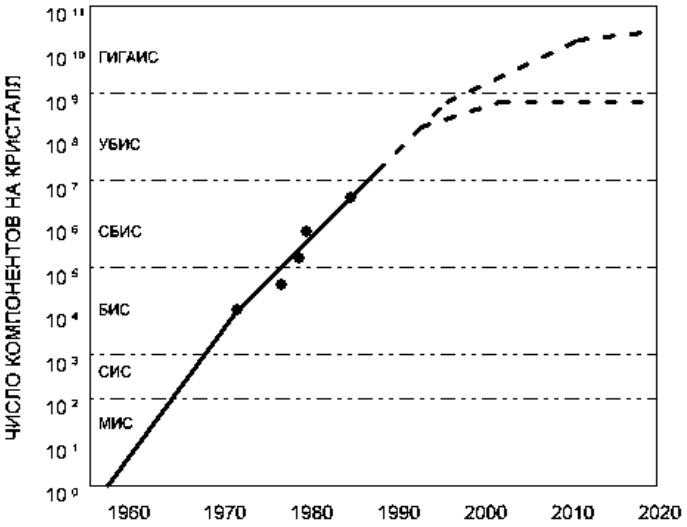


Рис. 1. Прогноз развития кремниевых технологий, сделанный в 80-е годы XX в. Обозначения на рисунке: МИС – малые ИС; СИС – средние ИС; БИС – большие ИС; СБИС – сверхбольшие ИС; УБИС – ультрабольшие ИС

Прогноз, представленный в 1980 г. (рис. 1), подтвердил справедливость «закона» Мура. В связи с этим представляется достаточно достоверным прогноз до 2018 г., представленный на рис. 2.

И хотя закон Мура давно превратился в ориентир для производителей чипов, в последнее время (учитывая предельные возможности современной технологии рис. 2) идут разговоры по поводу того, когда же будет исчерпан потенциал кремниевой технологии, не утихают и постоянно уточняются сроки ее заката. «Никакое экспоненциальное развитие не может быть вечно, но предел этой «вечности» мы еще можем отсрочить», загадочно предвещал 75-летний Мур в 2003 г.

Находящиеся в постоянном поиске альтернативы кремниевым чипам производители не слишком стремятся раскрывать свои кар-

ты, обычно они ссылаются на совместный план выпуска полупроводников до 2018 г. по усовершенствованной кремниевой технологии (см. рис. 2), хотя на самом деле ушли намного дальше. Так, на осеннем форуме Intel для разработчиков в Сан-Франциско (2006 г.) был представлен прототип процессора с 80-ю ядрами, который потенциально может обеспечить производительность уровня терафлоп-с. Ориентировочная дата выхода такого процессора – 2010 г.

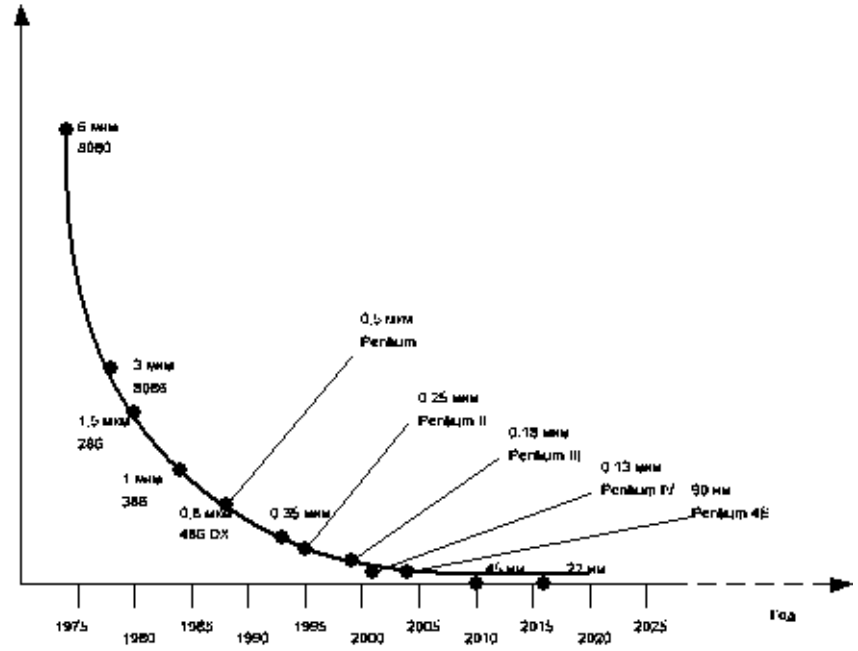


Рис. 2. Прогноз развития кремниевых технологий, сделанный фирмой Intel в 2008 г.

В то же самое время рассматривается довольно много различных экзотических альтернатив замены кремниевой технологии: квантовый компьютер, компьютер на основе ДНК-логики, молекулярный компьютер и т.п. Не одно десятилетие идут разговоры о нейрокомпьютинге. Однако пока не видно реальной альтернативы кремниевой технологии. Теоретические рассуждения, отнюдь, не всегда доходят до реальной технологии. Так, исследователи, начинавшие заниматься проблемой искусственного интеллекта, были оптимистами. Им казалось, что недалек тот час, когда будет скон-

струирована умная машина, которая не только научится играть в «крестики и нолики» или шахматы, но и сможет мыслить как человек. Однако с тех пор прошло уже более полувека, и как признаются те же разработчики: «мы, по-прежнему, далеки от этой цели».

Эта книга возникла в результате обработки конспекта лекций по вводным курсам «Информатика» и «Информатика и программирование», прочитанных авторами для студентов ряда групп факультета «А», а также вечернего факультета НИЯУ МИФИ; учебного пособия «Основы современной информатики», подготовленного для вышеперечисленных категорий студентов в 2007 г., которое было значительно переработано и дополнено.

ОСНОВНЫЕ СОКРАЩЕНИЯ

АКД – аппаратура канала данных
АЛУ – арифметико-логическое устройство
АСНИ – автоматизированные системы научных исследований
АСУ – автоматизированная система управления
АСУП – автоматизированная система управления производством
АСУТП – автоматизированная система управления технологическими процессами
АТС – автоматическая телефонная станция
АЦПУ – алфавитно-цифровое печатающее устройство
ВЗУ – внешние запоминающие устройства
ВОЛС – волоконно-оптические линии связи
ВЧ – высокие частоты
ГВС – глобальная вычислительная сеть
ГВЧ – гипервысокие частоты
ДНК – дезоксирибонуклеиновая кислота
ДОС – дисковые ОС
ЖК – жидкокристаллический
ЗУПВ – запоминающее устройство с произвольной выборкой
ИМС – интегральная микросхема
КВЧ – крайне высокие частоты
ЛВС – локальная вычислительная система
МКМД – много команд – много данных
МКОД – много команд – одно данное
МП – микропроцессор
НГМД – накопитель на гибких магнитных дисках
НЖМД – накопитель на жестких магнитных дисках
НМБ – накопитель на магнитных барабанах
НМД – накопитель на магнитных дисках
НМЛ – накопитель на магнитной ленте
НЧ – низкие частоты
ОВЧ – очень высокие частоты
ОЗУ – операционное запоминающее устройство
ОКМД – одна команда – много данных

ОКОД – одна команда – одно данное
ОНЧ – очень низкие частоты
ООД – оконечное оборудование данных
ОП – оперативная память
ОС – операционная система
ПЗУ – постоянное запоминающее устройство
ПК – персональный компьютер
ПЛОС – перфоленточные ОС
ППЗУ – перепрограммируемое постоянное запоминающее устройство
ППП – пакеты прикладных программ
ПУ – периферийные устройства
РНК – рибонуклеиновая кислота
САПР – системы автоматизированного проектирования
СВЧ – сверхвысокие частоты
СОД – средства обработки данных
СОЗУ – сверхоперативное запоминающее устройство
СПД – система передачи данных
СТС – сложная техническая система
СУБД – система управления базами данных
СЧ – средние частоты
УВВ – устройство ввода-вывода
УВЧ – ультравысокие частоты
УПДК – устройство подготовки данных на картах
УПДЛ – устройство подготовки данных на ленте
УУ – устройство управления
ЦП – центральный процессор
ЭВМ – электронно-вычислительная машина
ЭЛТ – электронно-лучевая трубка

ВВЕДЕНИЕ

Одним из основных понятий кибернетики является *информация,* под которой понимается совокупность данных (сведений), циркулирующих в реальном мире, в обществе, биологических и технических системах. Понятие информации принадлежит к исходным, неопределяемым понятиям науки. Как отражение явлений реального мира понятие информации раскрывается лишь указанием действий, в которых она участвует. Такими действиями могут быть измерение, передача, преобразование, обработка и хранение информации.

Важнейшие формы информации – дискретная и непрерывная. Дискретная форма информации является основной и представляется в виде конечных совокупностей качественно различных символов: это дискретные сигналы, печатные или рукописные документы, состояния цифровых автоматов и т.п. Непрерывная информация воплощается в образах (зрительных, звуковых и др.). Как в естественных, так и в искусственных процессах одна из этих форм информации может переходить в другую.

Информатикой называется наука о законах и методах измерения, передачи, обработки и хранения информации с использованием математических, технических и программных средств.

Термин «информатика», как название определенной выше области науки, был введен на рубеже 60 – 70-х годов практически одновременно во французской (Informatique) и немецкой (Informatik) научной литературе. Еще раньше в США стало использоваться в аналогичном смысле название Computer Science. В отечественной практике употребление термина «информатика» длительное время сдерживалось тем, что он использовался как название области знаний, связанной с архивным делом и документалистикой. Окончательное его утверждение произошло лишь после перевода в 1976 г. книги Ф. Бауэра и Г. Гооза «Информатика» [1] и образования в 1983 г. отделения информатики, вычислительной техники и автоматизации АН СССР.

Когда речь идет о таких науках, как физика, химия, биология и т.п., трудно говорить о единой науке. По существу, под этими названиями объединяются сильно различающиеся направления, имеющие общие объекты исследования. Если, например, сравнить

ядерную физику, механику, теорию электричества, то становится очевидным, что, несмотря на существование пограничных областей и взаимопроникновение методов и идей, каждое из этих направлений существует как самостоятельная наука. Аналогичная структура и у информатики. Под этим понятием объединяют ряд научных направлений, исследующих разные стороны одного и того же объекта информации.

Информатика родилась на основе достижений электротехники, электронике, точной механики, физики, автоматики, телемеханики, физиологии, математики, лингвистики и других, на первый взгляд даже не имеющих ничего общего между собой, наук.

Из областей знаний, являющихся разделами информатики или тесно связанными с ней, отметим следующие.

1. Теоретическая информатика – чисто математическая дисциплина, имеющая в своем составе такие разделы:

математическую логику (*математическая логика* – одна из ветвей общей логики, науки о формах мышления, которая развивается применительно к потребностям математики и вычислительной техники);

вычислительную математику и вычислительную геометрию (*вычислительная геометрия* рассматривает алгоритмы для решения геометрических задач, основное прикладное значение – машинная графика);

системный анализ;

исследование операций.

2. Кибернетика – наука об управлении. Возникла в конце 40-х годов прошлого столетия, когда Н. Винер впервые выдвинул идею о том, что системы управления в живых, неживых и искусственных системах обладают многими общими чертами. Наиболее активно развивается раздел технической кибернетики, в состав которой входит теория автоматического управления.

3. Программирование – научное направление, которое своим появлением полностью обязано вычислительным машинам (термин «программирование» не всегда понимается однозначно, так как имеется ряд таких разделов математики, как линейное программирование, нелинейное программирование, дискретное программирование и т.п., не имеющих ничего общего с программированием для

компьютеров; в связи с этим вместо термина «программирование» иногда употребляют термин «кодирование»).

4. Искусственный интеллект – самое молодое направление в информатике. Время его появления как научной дисциплины – начало 70-х годов прошлого столетия. Основная цель работы в области искусственного интеллекта – стремление проникнуть в тайны творческой деятельности людей, их способности к овладению навыками, знаниями и умениями. Это не чисто теоретическая наука, она занимается и прикладными вопросами. Примерами прикладных разделов этой науки являются робототехника или экспертные системы.

5. Информационные системы – направление, истоки которого лежат в области документалистики и анализа научно-технической информации. Информационные системы включают следующие разделы:

анализ и прогнозирование потоков разнообразной информации, перемещающихся в обществе;

исследование способов представления и хранения информации, создание специализированных языков для формального описания информации, разработку приемов сжатия, кодирования информации и т.п.;

создание информационно-поисковых систем и т.п.

6. Вычислительная техника – самостоятельное направление исследований, в рамках которого решается немало задач, не имеющих прямого отношения к информатике и ее проблемам (например, микроэлектроника, технология и т.п.). В то же самое время развитие вычислительной техники невозможно без использования результатов, полученных в теоретической информатике, программировании и других разделах, составляющих информатику.

7. Информатика в обществе – широкое внедрение компьютеров во все сферы человеческой деятельности, которое коренным образом изменяет среду обитания людей. Перспективы перехода к информационному обществу вызывают массу проблем социального, правового, технического характера. Все эти проблемы составляют объект исследования тех психологов, социологов, философов и юристов, работающих в области информатики.

8. Информатика в природе – изучение информационных процессов, протекающих в биологических системах, и использование

накопленных знаний при организации и управлении природными системами и создании технологических систем. Три самостоятельные науки входят в эту ветвь информатики:

биокибернетика, изучающая информационно-управляющие процессы, протекающие в живых организмах (диагностика заболеваний и поиск путей их лечения);

бионика (нейрокибернетика), исследующая насколько принципы работы живых систем могут быть применены в искусственных объектах;

биогеоценология, решающая проблемы, относящиеся к системно-информационным моделям поддержания и сохранения равновесия природных систем и поиска таких воздействий на них, которые стабилизируют разрушающее воздействие человеческой цивилизации на биомассу земли.

Разнообразие решаемых информатикой проблем, широкое применение ее в научной, хозяйственной, административной и культурной деятельности делают необходимым изучение элементов этой научной дисциплины в средних, средних специальных и, тем более, в высших учебных заведениях. При всем многообразии разделов информатики и направлений, непосредственно связанных с ней, их появление и развитие зависит от средств вычислительной техники. Эти средства имеют универсальный характер и на определенном уровне не зависят от приложений. Они имеют основополагающую, фундаментальную роль в науке, которая называется информатикой, именно с этим связано то, что в США вместо термина «информатика» употребляется Computer Science (компьютерные науки).

В то же самое время средства вычислительной техники включают в себя ряд взаимосвязанных частей, сильно различающихся по существу и требующих совершенно различных подходов при их изучении.

Это указанные выше разделы – вычислительная техника и программирование, включая языки программирования.

ГЛАВА 1. АЛГОРИТМЫ И ОСНОВЫ АЛГОРИТМИЗАЦИИ

1.1. Элементы алгебры логики

Математические основы информатики, наряду с другими разделами дискретной математики, включают в себя математическую логику. *Математическая логика* – раздел математики, изучающий математические доказательства, правильные способы рассуждений, логическую структуру и логические свойства различных объектов. Основу математической логики составляют исчисление высказываний и исчисление предикатов.

Исчисление (алгебра) высказываний, или *алгебра логики*, – раздел математической логики, изучающий высказывания со стороны их логических значений и логические операции над ними. При этом под *высказываниями* понимаются такие предложения, относительно которых имеет смысл утверждать, истинны они или ложны. Например, высказывание «а – буква» – истинно, а высказывание «1 – буква» – ложно. Отдельные высказывания в алгебре логики рассматриваются целиком, без учета их внутренней структуры. Для краткости в формулах, таблицах, схемах, описывающих различные логические конструкции, *логические значения* «истина» (TRUE) и «ложь» (FALSE) обычно указываются, соответственно, как «и» (T) и «л» (F) или, чаще всего, 1 и 0.

Логические операции называются также *логическими связками* [2]. С помощью *n*-местной или *n*-арной ($n \geq 0$) логической операции можно построить *n*-местную логическую функцию $f(x_1, x_2, ..., x_n)$, преобразующую тот или иной набор из *n* высказываний (аргументов) $x_1, x_2, ..., x_n$ в более сложное высказывание. Значения логических функций принято представлять в виде *таблиц истинности*, которые выражают истинностное значение (0 или 1) функции через истинностные значения входящих в функцию аргументов. *Нульместными* логическими функциями считаются логические константы 0 или 1. Основной *одноместной* (*сингулярной*) логической функцией является отрицание, а из формально существующих 16 *двуместных* (*бинарных*) логических функций лишь десять фактически зависят от двух аргументов. Наиболее распространенные функции представлены в табл. 1.1.

Особенно часто в информатике и, в частности, в алгоритмических языках программирования используются следующие логические операции: *отрицание* («не»), *конъюнкция* («и» – *логическое умножение*) и *дизъюнкция* («или» – *логическое сложение*). Эти три функции часто называют *булевыми*, а алгебру логики, в которой используются лишь эти функции, а также константы 0 и 1 – *булевой алгеброй*, поскольку они впервые появились в середине XIX в. в трудах английского математика Дж. Буля как аппарат символической логики. Однако иногда и другие логические функции, представленные в табл.1.1, также называют булевыми. Используемые в высказываниях основные логические связки типа «и», «или», «не», «если…, то», «тогда и только тогда» и другие можно рассматривать, соответственно, как «элементарные» булевы функции $x \wedge y$, $x \vee y$, $\neg x$, $x \supset y$, $x \equiv y$ и т.д.

Поскольку, как видно, имеется много логических функций, приобретает важность проблема их функциональной полноты. Некоторая система S логических функций называется *функционально полной*, если любую логическую функцию можно записать в виде формулы с помощью функций из S. Например, система булевых функций, состоящая из отрицания, конъюнкции и дизъюнкции, обладает функциональной полнотой. Однако функционально полными могут быть и системы, состоящие всего из двух логических функций. Так, полными системами являются отрицание и конъюнкция, отрицание и дизъюнкция, отрицание и импликация и др. Более того, функциональной полнотой могут обладать даже отдельные логические функции, например штрих Шеффера или стрелка Пирса. Следовательно, с точки зрения функциональной полноты, система булевых функций избыточна, однако избыточность позволяет короче и нагляднее записывать сложные высказывания. В алгоритмических языках программирования для нескольких наиболее распространенных логических связок («не», «и», «или», а иногда и др.) предусмотрены соответствующие символы или служебные (зарезервированные) слова.

Таблица 1.1

Обозначение связок	Другие обозначения и представления	Таблица истинности					Название связки	Как читать
		x	0	0	1	1		
		y	0	1	0	1		
$\neg x$	$\bar{x}$, x'	1		0			Отрицание, инверсия	Не x; отрицание x
$x \wedge y$	$x \,\&\, y$ $x \cdot y$ xy	0	0	0	1		Конъюнкция, логическое умножение, логическое «и», функция совпадения, *and* (*энд*)	x и y
$x \vee y$		0	1	1	1		Дизъюнкция, логическое сложение, логическое «или», функция разделения, *or* (*ор*)	x или y
$x \supset y$	$x \to y$	1	1	0	1		Импликация, материальная импликация	Если x, то y; x влечет y, x имплицирует y
$x \equiv y$	$x \leftrightarrow y$ $x \sim y$	1	0	0	1		Эквивалентность, функция равнозначности	x эквивалентно y; x тогда и только тогда, когда y
$x + y$	$x \dot{\vee} y$ $\neg(x \leftrightarrow y)$ $\neg(x \equiv y)$	0	1	1	0		Сумма по модулю 2, разделительная дизъюнкция, отрицание эквивалентности, функция неравнозначности, исключающее «или»	Либо x, либо y; x неэквивалентно y
$x \mid y$	$\neg(\bar{x} \vee \bar{y})$ $\overline{x \,\&\, y}$ $\bar{x} \vee \bar{y}$	1	1	1	0		Штрих Шеффера, отрицание конъюнкции, антиконъюнкция, *nand* (*нэнд*)	x и y несовместны; неверно, что x и y
$x \downarrow y$	$\neg(x \vee y)$ $\overline{(x \vee y)}$ $\bar{x} \wedge \bar{y}$	1	0	0	0		Стрелка Пирса, отрицание дизъюнкции, антидизъюнкция, функция Вебба, *nor* (*нор*)	Ни x, ни y

Окончание табл.1.1

Обозначение связок	Другие обозначения и представления	Таблица истинности					Название связки	Как читать
		x	0	0	1	1		
		y	0	1	0	1		
$x \not\supset y$	$x \not\to y$ $\neg(x \to y)$ $\neg(x \supset y)$		0	0	1	0	Отрицание импликации, антиимпликация, материальная антиимпликация	x, но не y; неверно, что x влечет y
$x \subset y$	$x \leftarrow y$ $y \to x$ $y \supset x$		1	0	1	1	Обратная импликация	x, если y; если y, то x; y влечет x; y имплицирует x
$x \not\subset y$	$x \not\leftarrow y$ $\neg(y \supset x)$ $\neg(y \to x)$ $\bar{x} \wedge y$		0	1	0	0	Отрицание обратной импликации, обратная антиимпликация	Не x, но y; неверно, что y влечет x

При записи формул алгебры логики можно избежать лишних скобок, пользуясь *старшинством логических операций*. В частности, отрицание считается старше, т.е. выполняется раньше конъюнкции, а конъюнкция – старше дизъюнкции. Как и в классической математике, логические умножение и сложение обладают свойствами коммутативности, ассоциативности и дистрибутивности.

При записи логических формул может быть также полезен *закон*, или *принцип двойственности*. Логическая функция f^* называется *двойственной* по отношению к логической функции f, если ее таблица истинности получается из таблицы истинности функции f заменой всех 0 на 1 и 1 на 0 (при этом замену нужно производить и для аргументов функции).

В силу принципа двойственности любую логическую функцию (формулу) f можно заменить на двойственную ей f^* по правилу

$$f(x_1, x_2, ..., x_n) = \neg f^*(\neg x_1, \neg x_2, ..., \neg x_n). \qquad (1.1)$$

Так, конъюнкция двойственна дизъюнкции, штрих Шеффера – стрелке Пирса, отрицание – самому себе и т.д. Пусть, например, A и B – произвольные логические формулы. Тогда в силу принципа двойственности

$$\neg(A \wedge B) = \neg A \vee \neg B, \quad \neg(A \vee B) = \neg A \wedge \neg B \tag{1.2}$$

(формулы (1.2) иногда называют *законами де Моргана*).

Для преобразования логических формул можно также использовать:

закон противоречия –

$$A \wedge \neg A = 0, \tag{1.3}$$

закон исключенного третьего –

$$A \vee \neg A = 1, \tag{1.4}$$

законы идемпотентности –

$$A \wedge A = A, \quad A \vee A = A, \tag{1.5}$$

законы поглощения –

$$A \wedge (A \vee B) = A, \quad A \vee (A \wedge B) = A, \tag{1.6}$$

а также формулы

$$\neg(\neg A) = A, \quad A \wedge 1 = A, \quad A \wedge 0 = 0, \quad A \vee 1 = 1, \quad A \vee 0 = A, \tag{1.7}$$

$$A \supset B = \neg A \vee B, \quad A \equiv B = (A \wedge B) \vee (\neg A \wedge \neg B).$$

(В выражениях (1.3) – (1.7) A и B – по-прежнему произвольные логические формулы.)

1.2. Элементы исчисления предикатов

Развитием исчисления высказываний является *исчисление предикатов*. Пусть x_i ($i = 1, 2, \ldots, m$) – заданные на некотором конечном или бесконечном множестве (поле) M переменные, называемые *предметными*, а их конкретные значения – индивидуальные *предметы* из этого множества. Предметными переменными могут быть, например, целые, действительные, комплексные числа; буквы различных алфавитов, химические символы, геометрические фигуры, фамилии (имена) членов некоторого коллектива и многое

др. Пусть также имеется набор функциональных символов (функций) с различным числом аргументных мест. Такими функциональными символами могут быть символы, описывающие математические функции; химические, экономические или финансовые формулы (правила), заготовки для слов, строк, таблиц, изображений, диаграмм, блок-схем с различным числом свободных (аргументных) мест и т.п. Из предметных переменных и функциональных символов можно конструировать так называемые *термы*, интерпретируемые как имена объектов исследования (математические выражения, слова и тексты, химические или ядерные реакции, изображения, блок-схемы и др.).

Тогда *n-местным* (*n-арным*) *предикатом* называется функция $P(t_1,t_2,...,t_n)$, которая представляет собой некоторое высказывание о термах $t_1,t_2,...,t_n$ $(n \geq 0)$ и может принимать лишь значения «истина» или «ложь» в зависимости от индивидуальных предметов $x_1,x_2,...$ из *M*, входящих в эти термы. При $n = 1$ предикат называют *свойством*, а при $n > 1$ – *отношением* или, конкретнее, *n-арным отношением*. Изучаемые в исчислении высказываний отдельные высказывания могут формально рассматриваться как *нульместные предикаты*.

Таким образом, аргументами функций исчисления высказываний являются лишь высказывания, тогда как аргументами предикатов могут быть любые предметные переменные. При этом число мест, т.е. число термов в предикате, вообще говоря, отличается от числа аргументов. Пусть, например, x_1,x_2,x_3 – буквы русского алфавита, а z_1,z_2,z_3,z_4 – действительные числа. Образуем термы $t_1 = x_1x_2x_1x_3$ (слово из четырех букв с одинаковыми первой и третьей буквами): $t_2 = z_1 + z_2$, $t_3 = z_3z_4$. Тогда высказывание «t_1 – имя собственное» – одноместный предикат (свойство) с тремя аргументами (x_1,x_2,x_3), а высказывание «$t_2 > t_3$» – двуместный предикат (отношение), зависящий от четырех аргументов (z_1,z_2,z_3,z_4). Для простейших математических отношений «меньше», «меньше или равно», «равно», «не равно», «больше или равно», «больше» в алгоритмических языках программирования имеются соответствующие символы или зарезервированные слова.

Отдельные предикаты – *элементарные формулы исчисления предикатов*. Связывая предикаты логическими связками, можно получать *сложные формулы исчисления предикатов*, например

$$P(t,u) \vee Q(v,z,w), \quad R(t) \wedge S(t,v), \quad T(u,v) \supset F(u),$$
$$(x > y) \wedge (y > z), \quad (u < v) \vee (u < 0), \qquad (1.8)$$

которые можно также рассматривать как новые предикаты. В этих формулах могут использоваться и высказывания алгебры логики (нульместные предикаты).

Помимо операций алгебры высказываний, в исчислении предикатов используются так называемые *кванторы* – логические операторы, переводящие одну высказывательную форму в другую и позволяющие указывать объем тех значений предметных переменных x, для которых данная высказывательная форма истинна. Обычно в логике предикатов ограничиваются двумя кванторами – *квантором общности* $\forall x$ и *квантором существования* $\exists x$. Пусть x – предметная переменная, принимающая значения из множества (поля) M. Тогда выражение

$$\forall x\, P(x), \qquad (1.9)$$

начинающееся квантором общности, представляет собой высказывание: «для всякого $x \in M$ истинно $P(x)$». Поскольку задание любых индивидуальных предметов (значений) $x \in M$ не нарушает истинность $P(x)$, высказывание (1.9) уже не зависит от x. Говорят, что предметная переменная x связана квантором общности, и поэтому такие переменные называют *связанными*.

Выражение

$$\exists x Q(x), \qquad (1.10)$$

начинающееся квантором существования, понимается как высказывание: «существует такой элемент $x \in M$, для которого $Q(x)$ истинно». Это высказывание также вполне определено и уже не зависит от x, т.е. переменная x связана квантором существования и является связанной переменной.

Кванторы общности $\forall x$ и существования $\exists x$ называются *двойственными* друг другу. В формулы (1.9), (1.10), кроме x, могут входить и другие предметные переменные (не показанные в формулах). Поскольку эти формулы не зависят от x, они полностью определяются заданием остальных переменных. Такие предметные пе-

ременные, не связанные никакими кванторами, называют *свободными.*

Существует закон, связывающий кванторы со знаком отрицания. Рассмотрим выражение

$$\neg(\forall x P(x)). \qquad (1.11)$$

Очевидно, выраженное (1.11) высказывание «$\forall x P(x)$ ложно» равносильно высказыванию «существует такой элемент $x \in M$, для которого $P(x)$ ложно» или, что то же самое, «существует элемент x, для которого $\neg P(x)$ истинно». Следовательно, выражение (1.11) равносильно выражению

$$\exists x \neg P(x). \qquad (1.12)$$

Аналогично, выражение

$$\neg(\exists x Q(x)), \qquad (1.13)$$

означающее высказывание «$\exists x Q(x)$ ложно», равносильно высказыванию «для всех x ложно $Q(x)$» или «для всех x истинно $\neg Q(x)$». Следовательно, выражение (1.13) равносильно выражению

$$\forall x \neg Q(x). \qquad (1.14)$$

Таким образом, стоящий перед квантором знак отрицания можно внести под знак квантора, заменив квантор на двойственный. Рассмотренные конструкции математической логики часто используются при записи алгоритмов.

1.3. Алгоритмы

Важнейшим понятием информатики является алгоритм. *Алгоритмом* называют понятное и точное предписание, которое задает порядок действий или процесс (называемый в этом случае *алгоритмическим*), начинающийся с произвольных исходных данных x (из некоторой совокупности значений X, допустимых для данного алгоритма) и направленный на получение полностью определяемого этими исходными данными результата y из некоторого допустимого множества Y [30, 31]. В современной математике понятие алгоритма относят к фундаментальным, т.е. к таким, которые не могут быть определены через более простые (таковы, например, фундаментальные понятия «точка», «линия» и др.). Поэтому данная

выше формулировка (кстати, далеко не единственная) не является математическим определением, а лишь пояснением смысла алгоритма. Заметим, что формулировку «понятное предписание» следует уточнить как «понятное исполнителю». Поэтому алгоритмы всегда составляются с учетом того, какие предписания могут быть поняты исполнителем, и, следовательно, для одного и того же алгоритма можно составить множество описаний, различающихся степенью детализации.

Важнейшие свойства алгоритма – дискретность, детерминированность, массовость и результативность. *Дискретность* означает, что алгоритмический процесс представляется в виде последовательности шагов, и переход к следующему шагу осуществляется лишь по завершении текущего.

Детерминированность предполагает, что алгоритм не допускает никаких произвольных, интуитивных толкований и действий и должен предусматривать строго определенную последовательность действий для всех случаев, которые могут встретиться в ходе выполнения алгоритма при любых возможных исходных данных и промежуточных результатах. Детерминированность алгоритма обеспечивает получение идентичных результатов любыми исполнителями, а отсутствие в алгоритме каких-либо сознательных действий делает возможным его выполнение автоматическими устройствами, например компьютерами. В то же время детерминированность не препятствует использованию в алгоритмах (например, в алгоритмах случайного поиска) случайных величин, но должны быть строго определены вероятностные характеристики используемых случайных величин и последовательность действий, предпринимаемых в зависимости от значения каждого очередного случайного числа.

Свойство *массовости* понимается как применимость алгоритма к решению бесконечного класса однотипных задач, различающихся исходными данными (из множества допустимых исходных данных). Например, массовость алгоритма численного интегрирования означает его пригодность для вычисления определенных интегралов от любых интегрируемых функций, а алгоритма синтаксического контроля программ – возможность проверки любых программ данного языка программирования. В математике проблему создания алгоритмов, обладающих массовостью, называют *массовой алгоритмической проблемой*.

Свойство *результативности* алгоритма означает, что он для каждого набора исходных данных $x \in X$ обеспечивает получение за конечное (хотя заранее, возможно, и неизвестное) число шагов некоторого результата $y \in Y$. (При этом неявно подразумевается, что результат должен быть получен за приемлемое время.) Если же для каких-либо $x \in X$ не удается получить результат за конечное число шагов или получаемый результат $y \notin Y$, то для достижения результативности алгоритма можно формально считать результатом сообщения о возникающей ситуации, соответственно пополнив область результатов Y этими сообщениями. Так, результативность алгоритмов трансляции (перевода) программ с алгоритмических языков достигается пополнением области результатов сообщениями об ошибках.

Помимо указанных выше четырех свойств, которыми должны обладать все алгоритмы, часто рассматриваются такие их желательные свойства, как *эффективность* (хорошая точность при малой загрузке компьютера) и *оптимальность* (наилучшие результаты при той же загрузке компьютера или, наоборот, наименьшая загрузка компьютера при том же качестве результатов).

На отдельных шагах основного, главного алгоритма возможны обращения к другим, вспомогательным алгоритмам с собственными описаниями. Некоторые вспомогательные алгоритмы, решающие типовые массовые задачи, можно стандартизовать. Поскольку вспомогательные алгоритмы могут, в свою очередь, обращаться к вспомогательным алгоритмам следующего нижнего уровня, алгоритм в целом иногда имеет сложную иерархическую структуру со многими уровнями вложенности.

Основными средствами описания алгоритмов служат словесная запись, блок-схемы, псевдокоды и алгоритмические языки.

Словесная запись обычно ориентирована на исполнителя человека, составляется на естественном языке с использованием при необходимости формул, почти не формализована и, как правило, служит предварительным описанием для дальнейшего представления в виде блок-схемы или на алгоритмическом языке. Основное требование к словесной записи алгоритма – понятность исполнителю. Типичные школьные примеры – словесные описания алгорит-

мов арифметических действий (сложение, вычитание, умножение, деление) «в столбик».

Блок-схема алгоритма представляется в виде конечного множества *узлов* (блоков, вершин) и соединяющих их направленных отрезков линий (*ребер*). В математике такая фигура называется ориентированным графом. Вершины графа описывают различные действия с информацией (данными), а ребра – передачи управления и потоки данных. В простейших блок-схемах достаточно использовать всего три типа узлов – функциональный, предикатный и слияния, которые считаются основными (рис. 1.1).

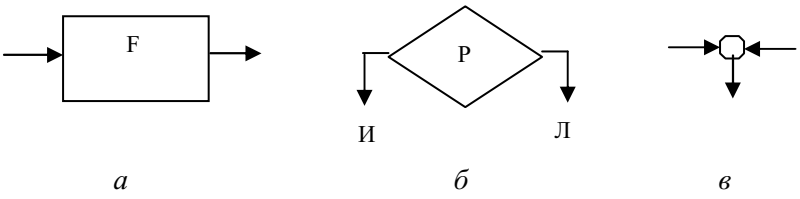


Рис. 1.1. Основные типы узлов в блок-схемах:
а – функциональный, *б* – предикатный, *в* – слияния

Функциональный узел изображается прямоугольником, имеет один вход и один выход и осуществляет указанную в нем обработку информации. *Предикатный узел* изображается ромбом, имеет один вход и два выхода и определяет истинность или ложность указанного в нем предиката. Если предикат приобретает значение «истина», осуществляется выход по стрелке, помеченной буквой «И». Если предикат приобретает значение «ложь», – происходит выход по стрелке, помеченной буквой «Л». Вместо «И» и «Л» часто используются также, соответственно, «да» и «нет». *Узел слияния* изображается кружком, не действует на данные и указывает, что два пути обработки данных сливаются в один. Помимо основных типов узлов в блок-схемах алгоритмов используют еще ряд дополнительных, которые здесь рассматривать не будем. В силу хорошей наглядности блок-схемы алгоритмов обычно используются в качестве пояснительных и промежуточных рабочих материалов. В качестве примера на рис. 1.2 изображена блок-схема поиска максимального числа $x_{\max}$ из множества $n > 1$ действительных чисел $\{x_i; i = 1, 2, \ldots, n\}$ и его номера $i_{\max}$.

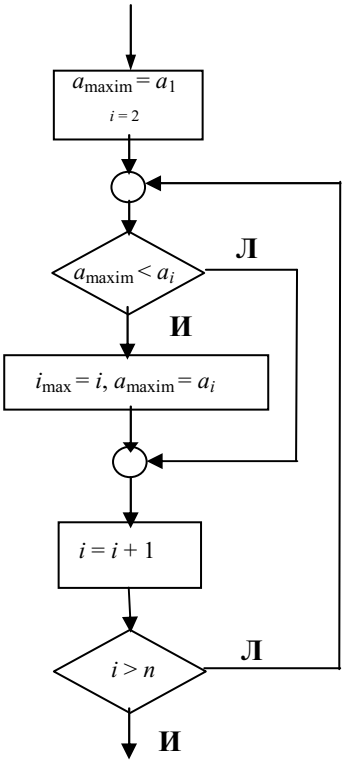


Рис. 1.2. Блок-схема алгоритма поиска максимального элемента последовательности n чисел

Алгоритмические языки предназначены, как следует из названия, для записи алгоритмов. Запись алгоритма на алгоритмическом языке настолько формализована, что его выполнение может осуществляться автоматизированными вычислительными устройствами – компьютерами.

Чем выше «интеллектуальный» уровень алгоритмического языка, тем менее детальным может быть «понятное компьютеру» описание алгоритма, тогда как на алгоритмических языках относительно невысокого уровня алгоритмы должны быть описаны вплоть до мельчайших подробностей. В частности, так называемые проблемно-ориентированные языки программирования высокого

уровня, используемые, например, в некоторых пакетах прикладных программ, зачастую вообще позволяют ограничиться лишь формулировкой задачи и указанием метода ее решения, а всю остальную работу, включая построение алгоритма, составление программы и ее выполнение, осуществляет компьютер.

1.4. Проблемы алгоритмизации

Алгоритмизацией называют процесс перехода от постановки задачи к алгоритму ее решения. При этом обычно предполагается, что методы решения уже выбраны или заданы и требуется лишь представить в виде алгоритма процесс решения задачи этими методами. Однако для многих задач, главным образом невычислительного характера, таких, как управление объектами, распознавание образов, составление расписаний и маршрутов, навигация, диагностика ситуаций, принятие решений и др., выбор подходящего метода решения также включается в алгоритмизацию. Наиболее распространенная схема алгоритмизации включает следующие этапы.

1. Предварительный анализ задачи алгоритмизации и объекта исследования. На этом этапе систематизируются основные сведения об объекте исследования, а также максимально допустимые расходы ресурсов компьютера (времени и памяти), используемых для решения.

2. Составление описания исходных данных. Для всех исходных данных алгоритмизируемой задачи должны быть конкретизированы форма представления (числовая, символьная, графическая), допустимое множество значений, структура (скаляры, массивы, таблицы, списки, изображения), каналы и последовательность ввода (до начала решения или порциями в ходе решения), место хранения (оперативная или внешняя память) и другие характеристики, которые могут понадобиться при алгоритмизации.

3. Составление описания результатов. Для результатов решения должны быть конкретизированы примерно те же характеристики, что и для исходных данных, с учетом их последующего использования. Особо отметим все шире применяемое в настоящее время аудиовизуальное представление результатов в виде движущихся, а иногда и озвученных многоцветных изображений (мульт-

фильмов), требующее специальных алгоритмических и технических средств компьютера.

4. Составление общего структурного описания исследуемого объекта (процесса, фирмы, магазина, банка, вуза, прибора, сети связи, склада и т.п.) и иерархии подчинения. Структурное описание составляется, по возможности, так, чтобы в дальнейшем отдельным структурным блокам отвечали соответствующие блоки алгоритма.

5. Разработка общего алгоритма решения. На этом, чрезвычайно важном, этапе обычно требуется длительная предварительная работа по изучению, систематизации, оценке точности, обобщению уже имеющихся алгоритмов решения аналогичных задач. Некоторые подходящие алгоритмы могли быть теоретически или эмпирически ранее найдены квалифицированными специалистами в соответствующих областях. При разработке алгоритмов решения невычислительных задач требуется еще личное (возможно, практическое) изучение объекта алгоритмизации и всех возможных в нем ситуаций, включая аварийные.

6. Составление укрупненной схемы алгоритма, выделение основного и вспомогательного алгоритмов. Схема должна давать наглядное представление о структуре алгоритма в целом, последовательности основных блоков и важнейших предикатных переходах без излишней на этом этапе детализации. В схемах алгоритмов, предназначенных для компьютеров, допускающих организацию параллельных вычислений, следует, по возможности, алгоритмически облегчить распараллеливание обработки информации в компьютере.

7. Пошаговая детализация алгоритма путем последовательной замены крупных алгоритмических блоков несколькими более мелкими, вплоть до операторов некоторого языка программирования.

Дальнейшие этапы включают контроль, отладку, тестирование, оценку эффективности, усовершенствование, оптимизацию алгоритма, причем обычно эти этапы объединяются с разработкой программы для компьютера.

Задача алгоритмизации в общем случае включает в себя также этап выбора языка программирования. Если же программа составляется для пополнения некоторого программного комплекса на определенном языке, то этап выбора языка исключается.

В связи с задачей алгоритмизации возникает естественный вопрос о том, все ли проблемы допускают представление решения в виде алгоритма. В общем случае ответ на этот вопрос отрицательный. Проблемы, для которых невозможно построить алгоритм решения, называются *алгоритмически неразрешимыми*. Однако теоретически удалось доказать алгоритмическую неразрешимость лишь для некоторых проблем, относящихся главным образом к основаниям математики. В практических задачах обычно удается обеспечить алгоритмическую разрешимость, подходящим образом сужая допустимую область исходных данных X.

1.5. Математическое моделирование – вычислительный эксперимент

С проблемами алгоритмизации тесно связано математическое моделирование различных реальных объектов как средств их изучения математическими методами. Исходным этапом здесь является разработка (или выбор) адекватной объекту математической модели. *Математическая модель* – приближенное описание какого-либо класса объектов или явлений внешнего мира, выраженное с помощью математической символики (уравнений, неравенств, логических отношений), таблиц, графиков, диаграмм и т.п. *Математическое моделирование* – процесс изучения этих объектов (явлений) на компьютере. Поскольку математическое моделирование во многом напоминает натурный эксперимент, его часто называют математическим, или численным, или вычислительным экспериментом. Во многих случаях математический эксперимент выполняется быстрее и дешевле натурного. Иногда на определенных стадиях исследования математический эксперимент – единственно возможный метод изучения многих объектов и явлений природы и общества.

Можно выделить четыре основных этапа математического моделирования:

1) формулирование законов, связывающих основные объекты математической модели (вывод уравнений);

2) исследование математической модели (решение уравнений на компьютере);

3) анализ адекватности модели сопоставлением результатов моделирования с практическими данными и аналитическими оценками;

4) усовершенствование модели по результатам этого анализа и в связи с накоплением новых практических и теоретических данных.

Различают *статические* модели, описывающие состояние объекта в текущий момент или период времени, и *динамические*, описывающие эволюцию объекта. Из статических моделей особо выделяют *равновесные*, с помощью которых изучаются состояния равновесия различных сложных систем или процессов. Различают также детерминированные и стохастические модели. В *детерминированных* моделях все параметры, характеристики, взаимосвязи являются строго определенными, а в *стохастических* учитывается влияние случайных, заранее неизвестных факторов, например метеоусловий или аварий.

Несмотря на указанные выше достоинства, математический эксперимент не исключает необходимость проведения практических исследований и теоретического анализа. Практические исследования необходимы, так как сама математическая модель является приближенной, а теоретические оценки – так как исследование математической модели на компьютере сопряжено с вычислительными погрешностями, иногда весьма значительными. Поэтому для успешного решения большинства сложных задач требуется разумное сочетание всех трех типов исследования – теоретического, вычислительного и практического.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Постройте логические константы 0 (ложь) и 1 (истина) с помощью логических функций «и», «или», «не».

2. Какой логической функции отвечает выражение «ни пуха, ни пера»?

3. Какой логической функции отвечает поговорка «назвался груздем – полезай в кузов»?

4. Каковы важнейшие свойства алгоритмов?

5. Каковы основные средства описания алгоритмов?

6. Раскройте скобки в выражении $\neg(x \wedge y)$.

ГЛАВА 2. ТЕХНИЧЕСКИЕ СРЕДСТВА ИНФОРМАТИКИ

2.1. Структура современных систем обработки данных

За сравнительно небольшой период развития электронных вычислительных систем появилось довольно много технических средств обработки данных (СОД).

На рис. 2.1 представлена примерная структура СОД в наиболее употребляемых терминах для их обозначения.

Сосредоточенные (централизованные) системы – СОД, отдельные компоненты которых располагаются на расстояниях, не превышающих возможности подключения стандартных периферийных устройств компьютеров. Базовым элементом таких систем является однопроцессорный компьютер с последовательной обработкой. К этой категории СОД относятся вычислительные комплексы: 1) многомашинные; 2) многопроцессорные. *Первые* представляют собой несколько близкорасположенных автономных компьютеров, имеющих высокоскоростные каналы связи, что обеспечивает повышение производительности и надежности комплекса. *Вторые* – единые компьютеры, имеющие большое количество процессоров, обеспечивающих распределенную между ними обработку данных, что может значительно повысить производительность СОД.

Под вычислительными системами, как правило, понимают СОД, решающие задачи конкретной области применения (на базе вычислительных комплексов). Это узкоспециализированные системы.

Под распределенными[*] (или децентрализованными) системами в данном случае понимаются системы, отдельные компоненты которых располагаются на значительном удалении друг от друга. Обмен данными между ними осуществляется с помощью традиционных каналов, используемых в телеграфии и телефонии (медный провод, оптоволокно, радиорелейные, спутниковые и т.п.). Нижний уровень распределенных систем занимают системы телеобработки. Это наиболее старые и давно развивающиеся системы, применяемые, как правило, для приема информации с удаленных объектов, ее обработки и управления этими объектами.

[*] Следует отметить, что иногда под *распределенными* системами понимают распределенную обработку данных в многомашинных и многопроцессорных комплексах.

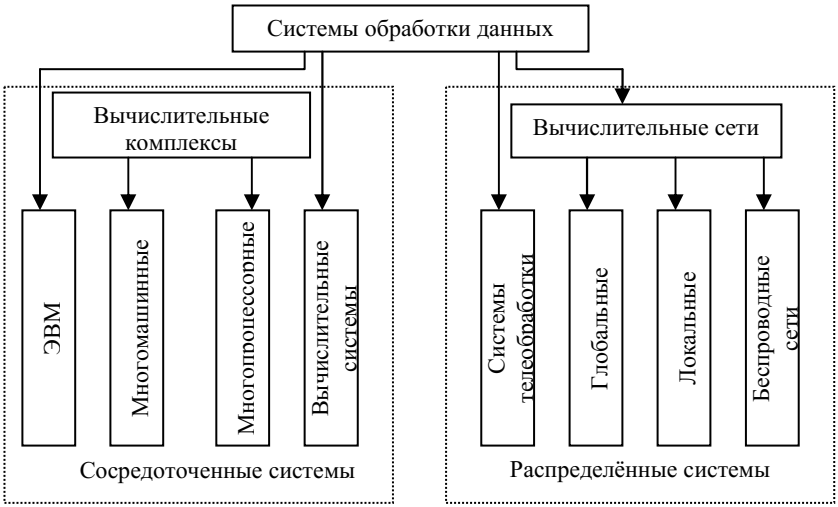


Рис. 2.1. Классификация СОД

Вычислительные сети, объединяющие самые разнообразные компоненты СОД, перечисленные выше, получили в последнее время очень широкое распространение. Эти сети на сегодняшний день разделяют на две категории: *глобальные*, охватывающие большие регионы и использующие стандартные средства телекоммуникаций, и *локальные,* работающие на ограниченной территории и использующие выделенные, узкоспециализированные линии связи. Эти две категории сетей имеют средства взаимного доступа и по своим техническим характеристикам постепенно сближаются.

Базовым элементом, на основе которого строятся все остальные компоненты СОД, является отдельный компьютер, поэтому изучение технических средств информатики (Hardware) начнем с изучения принципов его работы.

2.2. Общие сведения о структуре компьютера и краткая характеристика его составных частей

Более чем за 50 лет развития современной вычислительной техники прогресс в аппаратной реализации компьютеров и их технических характеристиках превзошел все мыслимые прогнозы, и пока

не замечено снижение темпов развития этой области техники. Но несмотря на большое разнообразие современных компьютеров, которые внешне резко отличаются от первых моделей 50-х годов, основополагающие идеи, связанные с понятием алгоритма, разработанным Аланом Тьюрингом, и с конструкцией (или архитектурной реализацией), предложенной фон Нейманом, пока не претерпели коренных изменений.

С самых общих позиций любой компьютер может быть рассмотрен в виде блоков (рис. 2.2).

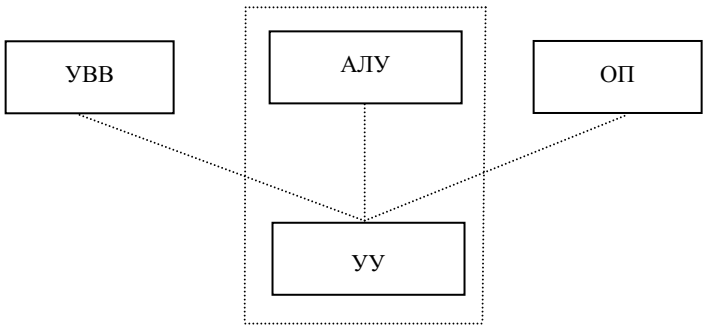


Рис. 2.2. Блок-схема оборудования компьютера:
УВВ – устройство ввода-вывода; ОП – оперативная память;
АЛУ – арифметико-логическое устройство; УУ – устройство управления

Оперативная (основная Main Memory) память (ОП), или оперативное запоминающее устройство (ОЗУ) – пассивное устройство, являющееся непосредственным местом хранения программ и обрабатываемых по этим программам данных. Говоря о запоминающих устройствах, необходимо отметить, что они не ограничиваются оперативной памятью, представляя собой систему, которая наряду с центральным процессором относится к тем критическим узлам, которые оказывают решающее влияние на его производительность. Эта подсистема построена по иерархическому принципу. Каждый уровень отличается от соседнего как быстродействием, так и объемом, причем быстродействие от уровня к уровню уменьшается, а объем увеличивается. На верхних уровнях иерархии (выше ОП) находятся сверхоперативные запоминающие устройства (СОЗУ), работающие в составе центрального процессо-

ра, а на нижнем уровне – внешние запоминающие устройства (или внешняя память) на уровне устройств ввода-вывода (УВВ).

Между ОП и центральным процессором, а также между внешней памятью и ОП может находиться так называемая кэш-память (Cache) небольшой емкости и со значительно меньшим временем обращения, чем память нижнего уровня. Явная необходимость в кэш-памяти при проектировании массовых центральных процессоров (ЦП) появилась в 1990 г., когда тактовая частота ЦП значительно превысила частоту системных шин и, в частности, шины памяти.

Справедливости ради следует отметить, что необходимость в промежуточной (Cach – кэш) памяти появилась намного раньше (в 60-х годах прошлого столетия). Причина появления этого типа полупроводниковой памяти заключалась в том, что технология производства вращающихся дисков не выдерживала темпов роста быстродействия процессоров. «Подкачка» информации с дисков была связана, естественно, с ограниченными возможностями ОП.

Также с начала 90-х годов принцип кэширования данных в оперативной памяти стал активно использоваться при обращении к дисковой подсистеме и в персональных компьютерах. При этом, как правило, не применяются отдельные микросхемы в кэш-памяти, а используются области обычной оперативной памяти компьютера в качестве дискового кэша, которая может изменяться непосредственно во время работы динамически в зависимости от ее общей загрузки.

Так, при частотах ЦП, достигших к 2006 г. 4 ГГц, частота выборки данных из ОП не превышала 200 МГц. В этой ситуации при прямом обращении к памяти функциональные устройства ЦП значительную часть времени простаивают, ожидая доставки данных. В какой-то степени проблемы быстродействия ОП могут быть решены увеличением разрядности шины памяти, однако этот путь имеет ограничения, связанные со значительным усложнением системной платы. В связи с этим эффективной альтернативы кэш-памяти в современных вычислительных системах не существует. Функционально кэш-память, так же как и основная, идентичны, являясь ОЗУ, с которым непосредственно взаимодействует ЦП. Когда необходимые данные ЦП не находит в кэш-памяти (кэш-промах), он обращается к ОП.

Следует отметить, что использование промежуточной памяти, которую также называют кэш, появилось значительно раньше (описанной выше) еще в первых суперкомпьютерах. Она размещалась между внешней памятью (дисковой) и ОП (иногда она называлась электронными дисками). Появление такой кэш-памяти было связано с тем, что скорость считывания данных с дисков отставала от потребности в «подкачке» данных в ОП из-за роста быстродействия ЦП.

Таким образом, в сбалансированной системе эффективность кэширования может быть достаточной для того, чтобы подсистема памяти в целом оказалась весьма близка к идеалу, когда быстродействия определялись самым быстрым участком (процессорными регистрами СОЗУ), а объем – самым емким (жесткими дисками).

Кэш-память имеет многоуровневую архитектуру (здесь речь идет о кэш-памяти, располагаемой между ЦП и ОП). Кэш первого уровня (L1) имеет небольшой объем, располагается в ядре ЦП и работает на той же частоте, что и узлы ЦП. В мощных процессорных архитектурах она разделена на кэш-инструкций (I-Cache-Instruction Cache) и кэш-данных (D-Cache-Data Cache). Это так называемая гарвардская архитектура кэша. Обеспечение очень высокой скорости считывания связано с ограничением объема этой памяти (от 16 до 32 Кбайт).

Кэш-память второго уровня (L2), как правило, унифицирована, т.е. может содержать как команды, так и данные. Если она встроена в ядро ЦП, то говорят об S-Cache (Secondary Cache – вторичный кэш), его объем составляет от одного до нескольких мегабайт; в противном случае вторичный кэш называется B-Cache (Backup Cache – резервный кэш), а его объем может достигать 64 Мбайт. В процессорах, имеющих интегрированные кэши первого и второго уровней (т.е. внутри корпуса процессора), часто устанавливается единый кэш третьего уровня (L3) вне процессора. Встроенный кэш третьего уровня именуется T-Cache (Ternary Cache – третичный кэш).

Как правило, каждый последующий уровень кэш-памяти медленнее, но больше по объему. Если в момент выполнения некоторой команды у ЦП не окажется данных для нее, то они могут быть затребованы из ближайшего уровня кэш-памяти, т.е. из D-Cache, в случае их отсутствия в D-Cache, запрашиваются в S-Cache и т.д.

Основным видом ОП по совокупности признаков в настоящее время является полупроводниковая память на интегральных схемах на базе биполярных или полевых транзисторов.

Микросхемы памяти выпускаются в виде ЗУПВ (RAM → → DRAM, SRAM) – запоминающего устройства с произвольной выборкой. RAM – random access memory, этот вид памяти реализуется в виде двух типов микросхем: DRAM* – динамическая память, обладающая высокой степенью интеграции, низкой стоимостью и потребляемой мощностью, но сравнительно малым быстродействием. Эта разновидность микросхем используется в качестве основной (Main Memory) или оперативной памяти компьютера. Их название связано с тем, что конденсаторы, входящие в состав микросхем нуждаются в постоянной подзарядке (примерно 15 раз в секунду), чем и объясняется их низкое быстродействие. Большое количество разновидностей динамической памяти связано со стремлением соблюсти баланс между увеличением производительности, надежности и стоимости оперативной памяти. При этом могут применяться дополнительные технологии, например технология ECC, позволяющая диагностировать и исправлять ошибки, возникающие при работе с оперативной памятью под действием солнечной радиации и других неблагоприятных факторов окружающей среды. Выбор типа микросхем зависит от многих факторов, включая производительность шин, к которым они подключаются. SRAM** – статическая память, наиболее быстрая, но более дорогая и с большим потреблением энергии, применяется в качестве внутренней памяти ЦП, а также в кэш-памяти. Основным недостатком оперативной памяти является то, что при пропадании напряжения данные в ней также пропадают. Поэтому для долговременного хранения применяются микросхемы постоянных запоминающих устройств (ПЗУ). Для того, чтобы в ПЗУ можно было повторно записать произвольные данные, предназначена процедура стирания памяти, в результате которой запрограммированные биты становятся снова незапрограммированными. Как правило, стирание памяти происходит электрически, либо при помощи ультрафиолетовой

* DRAM – Dynamic RAM.
** SRAM – Static RAM.

лампы. Память, которую можно стереть, называется *перепрограммируемым ППЗУ.*

В англоязычной литературе приняты следующие обозначения видов ПЗУ:

PROM – однократно программируемая память (programmable read only memory). Как правило, программирование битов производится воздействием повышенного напряжения, в результате которого пережигаются соответствующие «перемычки» внутри микросхемы памяти. Стирание памяти данного типа невозможно. Так же очень часто применяется аббревиатура OTP – one time programming, т.е. память, которую можно запрограммировать только один раз;

EPROM – стираемая программируемая память (erasable programmable read only memory). Память, которую можно стереть при помощи ультрафиолетовой лампы. Существенным недостатком такого вида памяти является то, что некоторые ячейки памяти могут самопроизвольно стереться под действием солнечной радиации;

EEPROM – электрически стираемая программируемая память (electrically erasable programmable read only memory). Память, стирание которой происходит электрическим путем. Одной из разновидностью EEPROM является флэш-память (Flash memory), повсеместно применяемая в данный момент в качестве так называемых USB-флэшек и карт памяти для цифровых фотоаппаратов и мобильных телефонов. Принципиальное отличие флэш-памяти от других микросхем EEPROM заключается в том, что программирование и стирание в них производится не побайтно, а большими блоками, называемыми страницами памяти (от 256 байт до нескольких десятков килобайт). Такой механизм несколько усложняет логику работы с микросхемой памяти, однако позволяет существенно увеличить ее объем и скорость записи.

Основные характеристики памяти – быстродействие и производительность.

Быстродействие памяти определяется временем выполнения операций записи или считывания данных. Основным параметром любых элементов памяти является минимальное время доступа и длительность цикла обращения. Время доступа (access time) определяется как задержка появления действительных данных на выхо-

де памяти относительно начала цикла чтения. Длительность цикла определяется как минимальный период следующих друг за другом обращений к памяти, причем циклы чтения и записи могут требовать различных затрат времени. В цикл обращения, кроме активной фазы самого доступа, входит фаза восстановления (возврат памяти к исходному состоянию), которая соизмерима по времени с активной. Временные характеристики самих запоминающих элементов определяются их принципом действия и используемой технологией изготовления.

Производительность памяти можно характеризовать как скорость потока записываемых или считываемых данных и измерять в мегабайтах в секунду. Производительность подсистемы памяти наравне с производительностью процессора существенным образом определяет производительность компьютера. Выполняя определенный фрагмент программы, процессору приходится, во-первых, загрузить из памяти соответствующий программный код, во-вторых, произвести требуемые обмены данными, и чем меньше времени потребуется подсистеме памяти на обеспечение этих операций, тем лучше.

Производительность подсистемы памяти зависит от типа и быстродействия применяемых запоминающих элементов, разрядности шины памяти, ее пропускной способности и некоторых «хитростей» архитектуры.

Размер (объем, емкость) памяти, минимальной единицей памяти является бит (bit → binary digit), – один двоичный разряд; в качестве единицы емкости памяти в настоящее время принят байт (byte – слог), который равен восьми двоичным разрядам (1 byte = 8 bit), 1024 байт = 1 Кбайт (килобайт), 1024 Кбайт = 1 Мбайт (мегабайт), 1024 Мбайт = 1 Гбайт (гигабайт), 1024 Гбайт = 1 Тбайт (терабайт), 1024 Тбайт = 1 Пбайт (петабайт).

Вся оперативная память машины разделена на слова. *Слово* – наименьшая адресуемая единица оперативной памяти, которая обрабатывается как единое целое. Длина слова определяет точность, с которой могут проводиться вычисления на данном компьютере. В различных типах современных компьютеров длина машинного слова изменяется от 8 до 64 бит.

Для поиска информации применяются два основных метода:

*1) адресный поиск***,** когда каждому слову памяти соответствует некоторый код (адрес), определяющий его местоположение в памяти (в различных компьютерах существует большое разнообразие способов адресации);

2) безадресный поиск, осуществляемый не по адресу, а по другим признакам.

Наиболее распространены запоминающие устройства двух типов:

стековая память (или магазинная) – одномерная память, представляющая набор ячеек, где могут быть записаны фрагменты программ, считывание которых осуществляется по указателю в порядке, обратном записи;

ассоциативная память, поиск информации в которой осуществляется одновременно во всех ячейках памяти по ее содержанию (содержимому), ассоциативному признаку, что позволяет в некоторых случаях существенно ускорить поиск и обработку данных.

Способ организации памяти, когда каждая программа может оперировать с адресным пространством, превышающим объем оперативной памяти, называется виртуальной (кажущейся) памятью. Эта организация памяти связана с использованием быстрой внешней (как правило, дисковой) памяти и страничной (или сегментной) структурой размещения на диске.

Центральный процессор (CPU – central processing unit) – основная часть («мозг») компьютера, непосредственно осуществляющая процесс обработки данных и управляющая ее работой. Центральный процессор (ЦП) предназначен для преобразования информации в соответствии с выполняемой программой, управления вычислительным процессом и устройствами, работающими совместно с ним. По мере того как функции вычислительной системы становятся все более распределенными, ряд функций ЦП возлагается на специализированные периферийные процессоры (в связи с этим основной процессор называется центральным).

К основным функциям ЦП могут быть отнесены следующие:

обращение к ОП, декодирование и выполнение команд программы в указанном порядке;

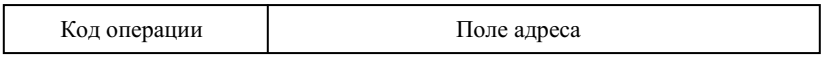
передача данных из оперативной памяти во внешние устройства и из внешних устройств в оперативную память;

ответы на запросы внешних устройств;

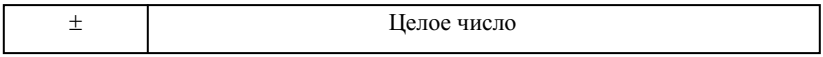
синхронизация работы всех элементов вычислительной машины.

Можно выделить два уровня ЦП, которые непосредственно отвечают за преобразование информации в соответствии с выполняемой программой.

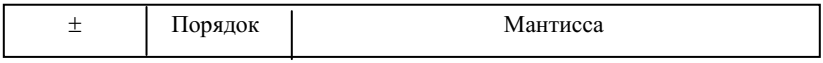
1. Традиционный машинный уровень. Он доступен пользователю и определяется набором из нескольких десятков или сотен сравнительно простых команд. По функциональному назначению можно выделить следующие категории команд: арифметические, логические, запись в память и загрузка регистров процессора из памяти, команды ввода-вывода и др. Любая пользовательская программа, подлежащая выполнению, находится в памяти и содержит вышеперечисленные команды. Структура машинного слова памяти, в которой размещается команда, состоит из двух частей (рис. 2.3): код операции (команда) и адресная часть, указывающая адреса ячеек (слов), необходимых для выполнения данной команды. В зависимости от количества адресов, располагающихся в адресной части, компьютеры могут быть трех-, двух-, одно- и безадресными.



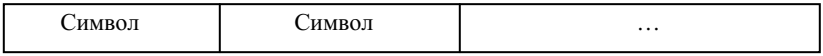
Формат команд



Формат данных (INTEGER)



Формат данных с плавающей точкой (REAL)



Формат символьных данных

Рис. 2.3. Структура машинного слова

При выполнении вычислений компьютер использует два типа чисел: целые и действительные. Последние в свою очередь имеют два способа представления: с фиксированной и плавающей запятой (точкой).

В первом случае место запятой, отделяющей целую часть от дробной, фиксируется, сразу же указывается количество разрядов, отводимых для изображения целой и дробной части, что упрощает выполнение машиной арифметических операций. Во втором случае положение запятой может меняться, что, как правило, позволяет повысить точность вычислений. При этом значащие числа (не нули) могут быть представлены сразу же после запятой с необходимым изменением знака порядка числа (нормализованное представление). Эта операция может выполняться как аппаратно, так и программно.

2. Микропрограммный уровень. Принцип микропрограммирования был предложен Моррисом Уилкесом в 1951 г. На этом уровне происходит разделение между программным и аппаратным обеспечением компьютера, и он, как правило, недоступен пользователю-программисту. В ранних моделях компьютеров команды выполнялись непосредственно на аппаратном уровне, т.е. для каждой команды существовали свои специальные схемы. Однако в настоящее время практически все типы компьютеров используют принцип микропрограммирования для реализации основных машинных команд.

Принцип микропрограммного управления основан на замене управляющих логических схем специальной программой, хранящейся в специальном запоминающем устройстве (ПЗУ), называемым памятью микрокоманд (микроприказов) или управляющем памятью. Эта память предназначена для хранения управляющих сигналов в форме микропрограмм, формирующих систему команд компьютера. Вычислительная машина, управление в которой осуществляется микропрограммой, называется машиной с микропрограммным управлением.

Практически происходит разложение сложной логической операции (необходимой для выполнения машинных команд) на элементарные, которые выполняются аппаратно с помощью простейших электронных схем. Эти схемы реализуют элементы математической логики (булевой алгебры).

Поскольку микропрограммный уровень реализуется конкретными схемными решениями, то программирование на этом уровне носит примитивный характер и затруднено. Часто необходимо принимать во внимание и временные соотношения.

Разложение сложной операции на простейшие элементы характерно для вычислительной техники. Так, из арифметических операций, по существу, ЦП выполняет только сложение, так как вычитание – сложение в дополнительном коде, умножение – сложение со сдвигом, деление – вычитание со сдвигом, возведение в степень – многократное умножение, извлечение корня – многократное деление.

В рамках выполнения функций управления вычислительным процессом, в том числе и для обеспечения режима мультипрограммирования, характерного для всех современных ЭВМ, важную роль играет ЦП. Под мультипрограммированием понимается такой режим работы, когда две или более задачи, одновременно находящиеся в оперативной памяти, используют ЦП попеременно.

Поскольку ЦП работает намного быстрее всех остальных устройств компьютера, участвующих в выполнении задачи, то при их совместной работе этот режим полезен. В однопрограммном режиме ЦП вынужден большую часть времени простаивать.

Задачи, выполняемые компьютером, разделяются по приоритетам на несколько уровней. Задачи с более высоким приоритетом могут прерывать реализуемые задачи более низкого приоритета, что обеспечивается системой прерываний. Режим мультипрограммирования оказывается полезным и для повышения эффективности работы ЦП при решении задач одного приоритетного уровня. Может случиться, что какая-то задача в ходе ее выполнения требует наличия каких-то ресурсов (внешних устройств, областей памяти и т.п.), и до тех пор, пока они не имеются в ее распоряжении, не может продолжаться выполнение этой задачи.

Это вынужденное время ожидания (а внешние устройства работают намного медленнее ЦП) может быть использовано для выполнения другой задачи (рис. 2.4). Задача *A* в момент t_1 обращается за вводом данных, и в это время подключается задача *B*, которая также через некоторое время освобождает ЦП, и он может использоваться задачей *C*. Так как ЦП работает с высокой скоростью, для пользователя создается впечатление, что все задачи работают од-

новременно. Под задачами здесь и далее понимаются не только задачи пользователя, но и задачи, управляющие ресурсами ЭВМ. На рис. 2.4 представлена так называемая невытесняющая многозадачность, когда следующая задача не может реализовываться до вынужденного прерывания предыдущей. Применяется и режим вытесняющей многозадачности, когда каждой задаче предоставляется определенный квант времени, по истечении которого процессор передается следующей задаче, а текущее состояние прерванной задачи запоминается и используется при следующем обращении (режим разделения времени). Возможен также режим Multithreading (или многонитевидная обработка), когда распараллеливание производится внутри одной задачи в режиме невытесняющей многозадачности. На рис. 2.5 представлены основные блоки ЦП.

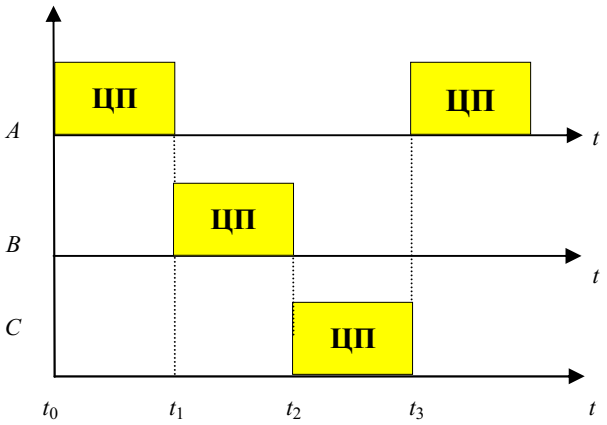


Рис. 2.4. Мультизадачный режим работы ЦП

1. Устройство управления. Оно предназначено для выборки команд из ОП, их интерпретации и инициализации. Обычно оно содержит следующие регистры:

программный счетчик PC – регистр, содержащий адрес той ячейки ОП, из которой будет выбираться следующая команда;

регистр команды IR – регистр, принимающий команду из основной памяти и хранящий ее во время дешифрования и выполнения;

слово состояния процессора PSW (оно может быть представлено отдельными триггерами, называемыми флажками или кодами условия, эти флажки фиксируют текущее состояние ЦП и важные особенности результата предыдущей команды);

указатель стека SP (во многих компьютерах применяется стек, который может быть образован набором внутренних регистров (разновидность внутренней памяти) или областью ОП; стек временно запоминает важную информацию на время выполнения подпрограмм и обработки прерываний).

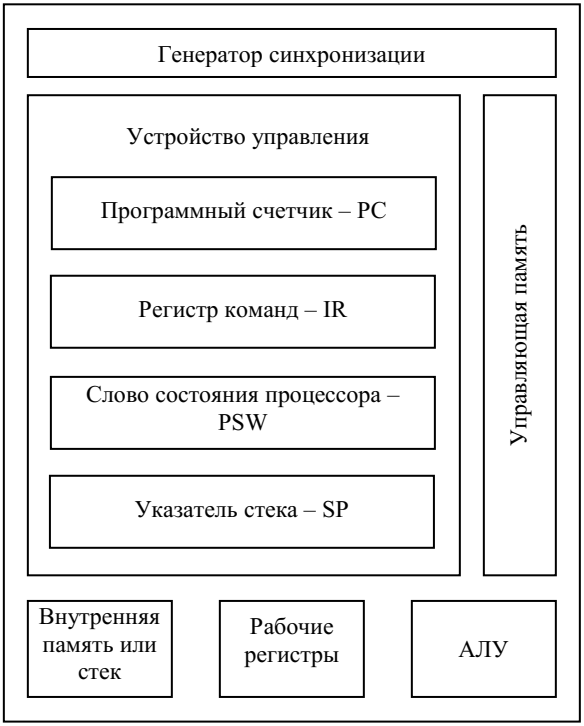


Рис. 2.5. Блок-схема ЦП

2. Управляющая память. Хранит набор микрокодов (микрокоманд), реализующих систему команд компьютера.

3. Рабочие регистры. Во всех компьютерах имеется несколько регистров двух видов:

аккумуляторы (или арифметические регистры), служащие «блокнотом» при арифметических и логических операциях;

регистры адреса, используемые для адресации данных и команд в основной памяти.

4. Арифметико-логическое устройство (АЛУ). Выполняет арифметические и логические операции над содержимым регистров и запоминает результаты по указанию команды.

5. Генератор синхронизации. Формирует один или несколько потоков равномерно распределенных импульсов, которые координируют все действия компьютера. Скорость работы компьютера в значительной степени определяется частотой генератора.

6. Внутренняя память. ЦП некоторых ЭВМ имеют небольшую внутреннюю память (иногда называемую кэш-памятью), которая обладает большим быстродействием, чем ОП, и служит буфером между ОП и регистрами ЦП. Это может значительно повысить быстродействие компьютера. Подробности кэш-памяти описаны ранее.

По мере развития технических средств компьютеров ряд функций ЦП передается на периферию. Создаются специализированные (периферийные) процессоры, которые аппаратно реализуют ряд часто используемых функций компьютеров, способствуя значительному повышению производительности компьютеров при выполнении основных операций по обработке информации.

Одними из первых специализированных процессоров следует считать процессоры ввода-вывода, которые выполняют одну из самых массовых процедур при работе компьютеров. Из-за низкой скорости (относительно скорости ЦП) работы периферийных устройств ввод-вывод значительно замедляет работу ЦП. Большое распространение получили специализированные процессоры, выполняющие основные арифметические операции, включая операции с плавающей точкой (в персональных компьютерах такие процессоры называют сопроцессорами). Дальнейшее развитие вычислительной специализации процессоров привело к появлению так называемых матричных процессоров, выполняющих операции линейной алгебры и быстрое преобразование Фурье.

ЦП – одно из важнейших устройств, определяющих производительность компьютера или его быстродействие. Можно выделить четыре типа оценки быстродействия компьютера:

1) пиковая оценка, или регистр-регистр (R-R), – предельная производительность ЦП, не учитывающая время обращения к ОП;

2) номинальная оценка (с учетом времени обращения к ОП (ЦП + ОП));

3) системная оценка (дополнительно учитывающая системное программное обеспечение, позволяющее работать ЭВМ);

4) эксплуатационная оценка – скорость выполнения прикладных программ (в зависимости от области применения (расчетные, экономические или другие задачи) используют разнообразные статистики (смеси), учитывающие среднее количество машинных команд, используемых в тех или иных приложениях; время же выполнения команд точно измеряется).

При оценке быстродействия микропроцессоров используются следующие характеристики:

MHz (мегагерц) – миллион тактов задающего генератора в секунду;

MIPS (Mega Instruction Per Second) – миллион инструкций (микрокоманд) в секунду;

MFLOPS[*] – миллион операций с плавающей точкой в секунду;

тесты Донгларра (решение СЛАУ[**]).

Устройства ввода-вывода (УВВ), или периферийные устройства (ПУ). Такие устройства компьютера используются для ввода-вывода, подготовки данных и запоминания больших объемов информации.

Отличительная особенность УВВ в том, что они в процессе работы преобразуют форму представления информации, не изменяя ее содержания. Для пользователя эти устройства представляют информацию в алфавитно-цифровой и графической формах. Для обмена с остальными устройствами компьютера используется множество самых разнообразных кодов, наиболее популярны следующие: ASCII (American Standard Code for Information Interchange), отече-

[*] Flops (flouting point operation per second) – количество операций с плавающей точкой в секунду – единица измерения производительности суперкомпьютеров. MFlops (мегафлоп) – миллион операций с плавающей точкой в секунду, GFlops (гигафлоп) – миллиард операций, TFlops (терафлоп) – триллион операций. Реальная производительность самого мощного суперкомпьютера 2005 г. превысила 136 TFlops, а в 2004 г. этот показатель составлял 35 TFlops.

[**] СЛАУ – системы линейных алгебраических уравнений.

ственный аналог которого – КОИ-8 – имеет восемь разрядов, семь из них значащие и один для проверки четности, применяется для работы устройств, подключаемых с помощью стандартных телеграфно-телефонных линий; EBCDIC (Extended Binary Coded Decimal Interchange Code), отечественный аналог ДКОИ – расширенный двоично-кодированный десятичный код для обмена информацией. Девятиразрядный (восемь значащих разрядов) код, применяемый фирмой IBM, как правило, предназначен для обмена информацией на магнитных носителях, в настоящий момент по большей части на мейнфреймах, работающих под управлением операционной системы z/OS. Увеличение международного сотрудничества вследствие бурного развития Интернета с конца 90-х годов привело к тому, что на данный момент повсеместно применяется кодировка Unicode, включающая в себя символы практически всех письменных языков мира.

Быстрое совершенствование ЦП и ОП компьютера, уменьшение их размеров и постоянное снижение стоимости привели к возрастанию роли УВВ. Так, уже сейчас стоимость УВВ составляет большую часть стоимости компьютера, а их габаритные размеры определяют размеры компьютера. Среди очень большого разнообразия УВВ можно выделить следующие категории.

Внешние запоминающие устройства (ВЗУ). Основная, или оперативная, память имеет высокую стоимость и ограниченную емкость (в значительной степени из-за высокой стоимости). В то же время использование компьютеров становится все в большей степени связано с накоплением и обработкой больших объемов информации. Потребности прикладных задач многократно превышают технические возможности электронной ОП. Это и явилось одной из причин появления внешних запоминающих устройств большой емкости, которые в то же самое время являются устройствами ввода-вывода информации. В ходе вычислительного процесса ВЗУ осуществляют двухсторонний обмен информацией с ОП, поэтому для них используются те же принципы обмена данными, что и для других внешних устройств. При значительно меньшей удельной стоимости (хранение бита информации) ВЗУ обладают очень большой емкостью (до гига- и терабайт). Относительно низкая стоимость, энергонезависимость и ряд других достоинств значительно расширили сферу применения этих устройств и способство-

вали их широкому распространению. Они используются и как устройства для хранения общесистемного программного обеспечения, определяющего оперативную работу компьютера (в дисковых операционных системах), и как архивная память компьютеров длительного хранения, эпизодического использования, как личные архивы пользователей и т.п. При этом сами носители информации могут храниться независимо от компьютера и считываться на различных компьютерах, имеющих соответствующие УВВ.

В настоящее время в компьютерах различных типов в качестве ВЗУ большой емкости применяются четыре типа накопителей: магнитные, магнитно-оптические, оптические и электронные (флэш-память). Магнитные ВЗУ появились первыми и представляют устройства с записью информации на подвижный магнитный носитель (рис. 2.6).

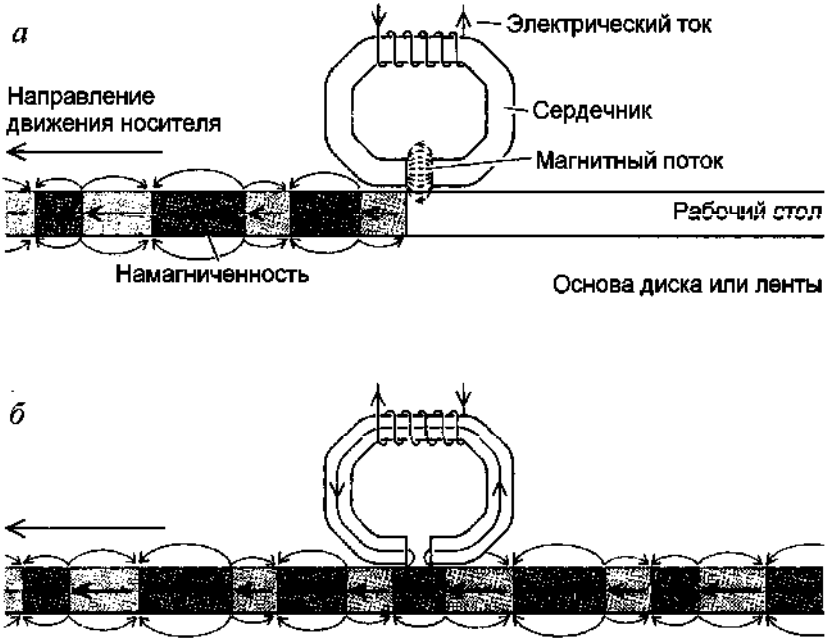


Рис. 2.6. Принцип записи на магнитный накопитель

Магнитное запоминание данных осуществляется намагничиванием определенных зон рабочего слоя на движущемся диске или ленте. Эти зоны намагничиваются в одном из двух противоположных направлений при помощи электромагнитной головки, состоящей из легко намагничиваемого сердечника с обмоткой из проводника (см. рис. 2.6, *а*). Когда по обмотке протекает ток, в сердечнике возникает магнитный поток. В зазоре сердечника силовые линии магнитного потока выходят за его границы и намагничивают проходящий под головкой магнитный материал. Если направление тока меняется на противоположное, ориентация намагниченности в рабочем слое также меняется на обратную. Данные, записанные таким образом, могут быть считаны, поскольку магнитное поле зон намагниченности простирается за пределы поверхности носителя. При движении носителя сердечник головки пронизывается изменяющимся магнитным потоком, который, в соответствии с законом электромагнитной индукции (Фарадея), индуцирует ток в обмотке (см. рис. 2.6, *б*). Если внимательнее приглядеться к зонам намагниченности и создаваемым ими полям, то видно, что поля в действительности начинаются и заканчиваются в промежутках между этими зонами, т.е. в переходных областях. Поскольку поля препятствуют намагничиванию среды, их называют размагничивающими.

К накопителям этого типа относят накопители на магнитной ленте (НМЛ), накопители на магнитных дисках (НМД) и накопители на магнитных барабанах (НМБ).

1. *НМЛ* – накопитель на магнитной ленте, построенный по тем же принципам, что и обычный бытовой магнитофон, вошел первым в практику использования как ВЗУ компьютера и относится к накопителям с последовательным доступом. Имеет самую низкую стоимость хранения бита информации, высокую надежность, удобен в эксплуатации. Однако НМЛ имеет очень большое время доступа к необходимой информации, так как для ее поиска необходим последовательный просмотр иногда всей ленты. Для больших магнитофонов время доступа может составлять 4 – 150 с. За время существования НМЛ применялись накопители, использующие два типа магнитной ленты:

1) для больших и малых компьютеров – обычно НМЛ с шириной ленты 12,7 мм;

2) в персональных компьютерах – накопители на компакт-кассетах (стримеры), в которых удалось добиться значительного увеличения плотности записи, как правило, путем отказа от старт-стопного режима работы и перехода к режиму «бегущей» (stream) ленты.

Такой режим хорошо согласуется с выгрузкой содержимого диска большей емкости с целью резервного копирования информации, расположенной на диске. По состоянию на 2010 г. фирмами IBM и FujiFilm анонсированы кассеты емкостью 35 терабайт.

Для первого типа магнитной ленты информация записывается на девяти дорожках, восемь из которых являлись информационными (девятая использовался для проверки четности). Длина ленты зависела от применяемых кассет и ее толщины и могла достигать более 1000 м, емкость более 100 Мбайт. В настоящее время такие НМЛ не используются.

Несмотря на обилие устройств хранения всевозможных типов, ленточные накопители в качестве устройств для резервного копирования сохраняют и даже упрочили свои позиции благодаря высокой надежности, скорости и низкой стоимости единицы информации.

2. *НМД* – накопители на магнитных дисках. В настоящее время наиболее популярны НМД, которые имеют большую емкость, малое время доступа (от 100 до 10 мс и менее) и невысокую стоимость хранения бита информации. НМД относятся к устройствам с прямым доступом, поскольку время поиска информации практически не зависит от его местонахождения на носителе.

Магнитный диск напоминает грампластинку с нанесенным на нее магнитным материалом. Запись или считывание информации производится по кольцевым или спиральным дорожкам с помощью перемещающихся головок (рис. 2.7).

Для увеличения емкости отдельные диски собираются в пакеты, в которых несколько металлических дисков (изготовленных из легких сплавов) устанавливаются на общую ось. Общий блок головок обеспечивает считывание информации с каждой из поверхностей. Существуют диски с постоянным (не перемещаемым) блоком головок над каждой из дорожек, они обладают очень малым временем доступа. Различают накопители на жестких магнитных дисках

(НЖМД – HDD*) и накопители на гибких магнитных дисках (НГМД – FDD**).

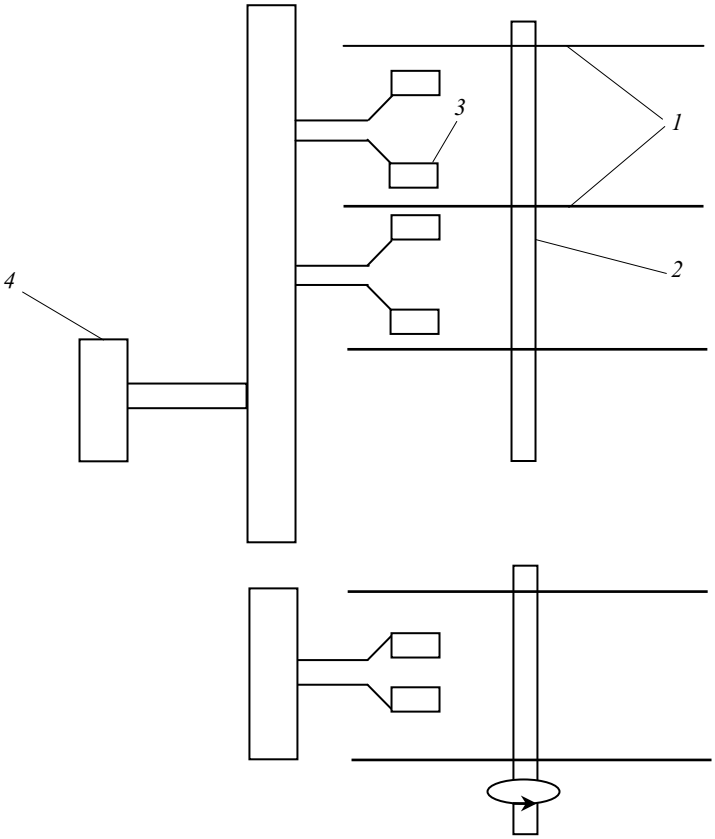


Рис. 2.7. Кинематическая схема накопителя на магнитном диске:
1 – диски, *2* – вал, *3* – магнитные головки, *4* – сервопривод

2а. *НЖМД* – накопители на жестких магнитных дисках имеют металлическую основу, вращаются с высокой скоростью, их считывающие головки находятся на расстоянии десятых долей микрометра от поверхности диска. Они обладают малым временем доступа, большой емкостью и высокой стоимостью. Поэтому такие диски

* HDD – Hard Disk Drive.
** FDD – Floppy (Flexible) Disk Drive.

обычно используются как системные, расширяющие возможности ОП для обеспечения работы операционной системы ее хранения и загрузки, а также для использования в различных прикладных программах. Наиболее популярными в последнее время стали так называемые винчестерские диски (название дисков связано с тем, что имелась некоторая аналогия в параметрах дисков с цифровыми параметрами оружия, выпускаемого известной американской фирмой). В этих устройствах за счет герметизации магнитного носителя, совершенствования считывающих головок магнитной ориентации частиц оксида вдоль дорожек и т.п. удалось, повысив плотность записи и уменьшив расстояние между дорожками, значительно увеличить емкость диска, а использование общего блока головок позволило сократить время доступа к информации.

Первый НЖМД был разработан на фирме IBM в 1956 г. Его размер (привод) составлял около двух бытовых холодильников, стоимость 150 тыс. дол., имел 50 дисковых дюралевых пластин с нанесенным слоем ферромагнетика, диаметром 64 см, скорость вращения 1200 об./мин, емкость 5 Мбайт, а время доступа – 1 с. К 2000 г. НЖМД стоимостью 150 дол. имел емкость более чем в $16x10^3$ раз, время доступа уменьшилось более чем в 100 раз, стоимость на единицу емкости (1 Мбайт) уменьшилось в 10 млн раз. Плотность записи растет, перекрывая даже известный закон Мура (для электронных схем), удваиваясь каждый год, несмотря на предсказываемый, так называемый сверхпарамагнитный предел, когда размеры одного магнитного домена на поверхности дисковых пластин, направление намагниченности которого регистрируется как один бит, станут настолько малыми, что тепловой энергии атомов (при комнатной температуре) будет достаточно, чтобы изменить его направление.

Любой современный НЖМД состоит из четырех основных частей:

носителя;

головки чтения-записи;

позиционера;

контроллера.

Носитель – пакет дисковых пластин, вращающихся на одной оси, на смену дюралюминиевым пластинам (дискам) пришли диски из керамики и стекла. На поверхность диска наносится слой ферромагнетика (в современных дисках – оксид хрома), покрытый сверху

тонким слоем алмазоподобного графита (для защиты от механических повреждений). Адресация дисков осуществляется следующим образом – номер диска, цилиндр (набор концентрических дорожек, расположенных на одинаковом удалении от центра вращения), дорожка, сектор (кластер). В современных дисках применяется зонно-битовая запись, имеющая различное количество секторов на дорожках с различным удалением от центра.

Благодаря аэродинамическому эффекту головка чтения-записи «летит» над поверхностью диска под воздействием потока предварительно очищенного воздуха. У первых винчестерских дисков расстояние между поверхностью диска и головкой равнялось 0,5 мкм, у современных дисков (по состоянию на 2010 г.) оно достигает 0,01 мкм (что естественно повышает плотность записи и емкость дисков). На смену ферритовым головкам пришли композитные, магниторезистивные, тонкопленочные.

Позиционер – устройство, наводящее головку на нужную дорожку. Существует два варианта привода – поворотный и линейный, каждый из которых имеет свои достоинства и недостатки. В обоих случаях привод осуществляется с помощью соленоидов.

Контроллер обеспечивает согласованное управление всеми элементами диска и передачу данных между компьютером и дисками (как правило, на базе микропроцессоров).

Накопители на жестких магнитных дисках применяются во всех используемых в настоящее время типах компьютеров и в зависимости от вида компьютера различаются:

по типу интерфейса – для персональных компьютеров, для серверов, для мейнфреймов и т.д.;

по типу размера накопителей – 3,5 дюйма для настольных систем, 2,5 дюйма для ноутбуков и других переносных систем;

по скорости вращения шпинделя: 15000, 10000, 7200, 5400 и 4200 об./мин.

Следует учитывать, что винчестер – устройство нежное и очень ненадежное (хотя определенные технологические успехи в этом направлении имеются). В то же время в ряде случаев информация, а не «железо» является наиболее важным ресурсом внешнего запоминающего устройства.

В связи с этим разработчики винчестеров предпринимают ряд решений, повышающих надежность хранения данных. Так, компа-

ния Western Digital представила технологию в рамках спецификации S.M.A.R.T. (Self Monitoring Analysis and Reporting Technology) – технология, принятая всеми ведущими производителями персональных компьютеров. Она предназначена для предотвращения потери информации путем предсказания возможности выхода из строя жестких дисков на основании результатов внутреннего тестирования состояния диска, а также выявления и восстановления информации в проблемных секторах. Происходит поиск поврежденных участков диска и перенесение информации в специально отведенные области.

Еще более кардинальным решением повышения надежности хранения информации является разработка так называемых RAID-массивов (Redundant Array of Inexpensive Device – избыточные массивы недорогих устройств). Данная технология позволяет распределить данные по нескольким носителям. Это совокупность из нескольких, обычно недорогих жестких дисков, управляемых программно или специальным контроллером таким образом, что результат выглядит как одно устройство, обладающее всеми необходимыми свойствами – высокой надежностью и скоростью, а также большой емкостью.

Все описываемые далее накопители относятся к устройствам хранения со сменными носителями (наряду с ленточными носителями – стримерами).

Этот вид накопителей используется для решения трех классов задач, предъявляющих к ним довольно сильно различающиеся требования. Во-первых, это публикация и распространение программного обеспечения, данных и мультимедиа – контента их изготовителями (объем которых постоянно растет). Здесь на первый план выходит совместимость носителей данных с наиболее распространенными на сегодняшний день накопителями. Во-вторых, это архивирование и резервное копирование информации. Главное требование в этом классе задач – максимальная емкость носителя, его надежность и долговечность, минимальная стоимость хранения единицы информации, максимальная производительность. В-третьих, это перенос информации между компьютерами, в частности обмен ею между пользователями. Основные требования для этого применения – дешевизна носителей и максимальная скорость записи и чтения.

2б. *НГМД (FDD)* (накопители на гибких магнитных дисках или флоппи-диски) – накопители на магнитной основе, появились в свя-

зи с распространением персональных компьютеров. Они обладают небольшой емкостью и довольно большим временем доступа. Однако их низкая стоимость и отсутствие жестких эксплуатационных требований при использовании и хранении дискет обеспечивали этому виду дисковой памяти высокую популярность у пользователей. Этот вид носителей применялся для хранения личных архивов пользователей, очень удобен для обмена как прикладными, так и системными программами небольшого объема. В настоящее время до сих пор используются дискеты диаметром 3,5 дюйма (89 мм), емкость диска 1,44 Мбайт, разработанные Sony в 1980 г. В конце 90-х годов получили широкое распространение так называемые ZIP-дискеты объемом 100 и 250 Мбайт. С появлением USB Flash-брелков дискеты были вытеснены с рынка, и в данный момент практически не применяются. У накопителях на магнитных барабанах была ограниченная область применения (в суперкомпьютерах), хотя и имелись сравнимые с жесткими дисками характеристики.

3. *Оптические и магнитооптические накопители.* Один из главных недостатков НЖМД, являющегося на сегодняшний день основным системным ВЗУ, – это низкая надежность к внешним механическим воздействиям (ввиду очень малого расстояния между считывающей головкой и поверхностью диска). В связи с этим в последнее десятилетие активно развиваются новые технологии, которые связаны с применением оптических элементов. Ближе всего к традиционным НЖМД приближаются магнитооптические устройства (рис. 2.8).

3а. *Магнитооптические (МО) накопители.* В магнитооптическом устройстве для чтения и записи данных применяется лазер. При записи лазер разогревает маленькую зону носителя, с ростом температуры коэрцитивная сила материала падает, для изменения направления намагниченности соответствующий участок диска должен быть нагрет до точки Кюри (около 200 °С). При чтении данных луч лазера переключается на меньшую мощность и поляризуется (эффект Керра). Поскольку плоскость поляризации поворачивается при отражении луча от намагниченной среды, второй поляризующий фильтр (называемый анализатором) может преобразовать изменение интенсивности светового луча, которое в свою очередь фиксируется фотодетектором.

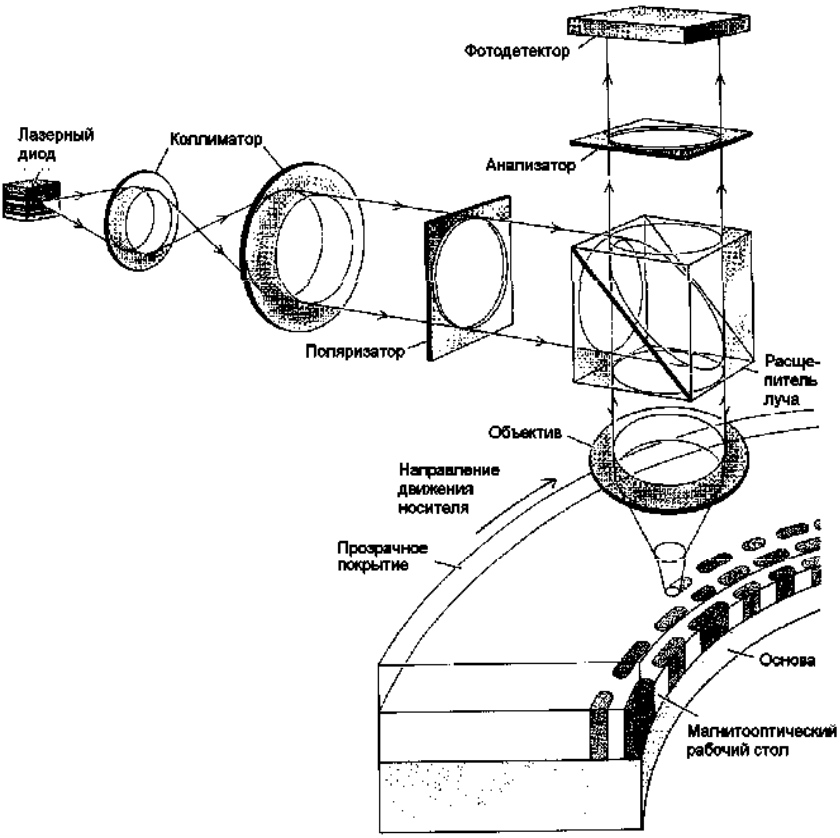


Рис. 2.8. Принцип работы магнитооптического устройства

В этих устройствах считывающая (записывающая) головка значительно удалена от поверхности диска. Они могут быть представлены и в виде магнитооптических библиотек с автоматической сменой большого количества дискет, общая емкость которых достигает очень больших величин.

К 2007 г. выпускалось два типа устройств магнитооптики с размером носителя 5,25 дюйма с максимальной емкостью 19 Гбайт и 3,5 дюйма с максимальной емкостью 2,3 Гбайт. Минимальное время доступа (поиска) в обоих типах – 19 мс. По емкости, быстродействию, надежности у магнитооптики нет равных (на указанный пе-

риод времени) среди устройств хранения со сменными носителями данных.

Магнитооптические носители выдерживают огромное количество циклов перезаписи, не чувствительны к внешним магнитным полям и радиации, гарантируют сохранность информации в течение полусотни лет. Именно поэтому эти устройства применяются тогда, когда предъявляются повышенные требования к объемам и надежности хранения данных (неслучайно, библиотека Конгресса США оборудована магнитооптическими устройствами).

3б. *Оптические накопители.* Оптические диски по принципу работы очень близки к магнитооптике (правда, поверхность, на которой хранится информация, не является магнитной).

Оптические диски относятся к устройствам хранения информации со сменными носителями (так же, как и магнитная лента, НГМД и т.д.).

Первый оптический диск CD-ROM появился в 1984 г. Первоначально выпускалось два формата CD-DA (Digital Audio) и CD-ROM. Запись осуществлялась так же, как и на всех дисках, – на концентрических дорожках в виде небольших углублений – вдоль дорожек – ямок (ямка – 1, гладкая поверхность – 0). Изготовление производилось штамповкой. С помощью таких дисков происходило распространение фирменного программного продукта или аудиоинформации. Считывание осуществлялось с помощью луча лазера, который, отражаясь от плоского участка диска, оказывался в противофазе с падающим лучом, а интенсивность результирующего (отраженного) луча близка к нулю (нулевые биты). При отражении от ямки луч проходит 1/2 длины волны, оказывается в фазе с падающим лучом и усиливает его (единичные биты), что и фиксируется фотодетектором. С начала своего использования и до последнего времени CD-ROM интенсивно развивались. Емкость их достигла 700 Мбайт, скорость считывания – 9 Мбайт/с (при скорости вращения до 12000 об./мин). При этом значительное удаление головки от поверхности диска существенно повышает его надежность. Правда, на сегодняшний день (после 2000 г.) он уже не в состоянии конкурировать с приближающимся к нему по цене DVD-ROM.

Первый записывающий диск с однократной записью (WORM – Write Once Read Many) CD-R (Recordable) был выпущен фирмой

Philips в 1993 г. В качестве «болванок» использовались обычные (как и для CD-ROM) поликарбонатовые диски, покрытые специальным красителем (цианиновым, фталоцианиновым или азокрасителем), на поверхность которого напыляется тончайших слой отражающего благородного металла (обычно чистого серебра или золота). При записи сфокусированный лазерный луч физически выжигает отражающую поверхность, и слой красителя образует непрозрачные участки, аналогичные ямкам в обычном штампованном CD. Развитие этих устройств в сторону многократной записи привело к появлению CD-WARM (Write And Read Many times). Развитмие этих устройств в сторону многократной записи привело к появлению CD-RWARM (Write And Read Many times).

В CD-RW активным слоем является специальный поликристаллический сплав (серебро-индий-сурьма-теллур), который переходит в жидкое состояние при сильном нагреве (500 – 700 °С) лазером. При последующем быстром остывании жидких участков они остаются в аморфном состоянии, поэтому их отражающая способность отличается от поликристаллических участков. Возврат аморфных участков в кристаллическое состояние осуществляется путем более слабого нагрева – ниже точки плавления, но выше точки кристаллизации (~ 200 °С). Выше и ниже активного слоя располагаются два слоя диэлектрика (обычно диоксид кремния), отводящие от активного слоя лишнее тепло в процессе записи. Сверху все это прикрыто отражающим слоем, а весь «сэндвич» нанесен на поликарбонатовую основу, в которой выпрессованы специальные углубления. В накопителе CD-RW используются три режима работы лазера, отличающиеся мощностью луча: режим записи (максимальная мощность, обеспечивающая переход активного слоя в неотражающее, аморфное состояние); режим стирания (возвращает активный слой в отражающее, кристаллическое состояние); режим чтения – самая низкая мощность (не влияющая на состояние активного слоя).

Максимально достижимая емкость дисков CD – 650 – 700 Мбайт. Невозможность дальнейшего увеличения ресурса (по емкости) этих дисков привели к появлению дисков DVD.

Первоначально этот диск должен был прийти на смену видеокассет и расшифровывался как Digital Video Disk, т.е. цифровой диск. В дальнейшем этот формат стал применяться в вычислительной тех-

нике, и его название поменялось на Digital Versatile Disk – цифровой многофункциональный диск. В 1995 г. появился единый стандарт, состоящий из пяти разновидностей: DVD-ROM; DVD-Video; DVD-Audio; DVD-R и DVD-RAM. Два последних стандарта в дальнейшем были преобразованы в DVD-RW; DVD+RW и DVD+R.

Каким же образом удалось значительно (в 7 – 25 раз) увеличить объем диска?

Во-первых, вместо инфракрасного (ИК) лазера с длиной волны 780 нм был применен лазер красного диапазона 650 нм. При этом произошло уменьшение размера (диаметра) углублений с 0,83 до 0,4 нм. Вследствие этого шаг дорожек уменьшился с 1,6 до 0,74 нм. Это привело к увеличению емкости диска в 4,5 раза.

Во-вторых, были разработаны двухслойные диски (материал первого отражающего слоя является полупрозрачным), это дает возможность увеличить емкость еще почти в 2 раза.

В-третьих, были разработаны двухсторонние диски, что позволило еще в два раза увеличить общую емкость дисков (правда, диск приходится переворачивать вручную). Возможны четыре варианта дисков DVD: односторонний, однослойный – 4,7 Гбайт; односторонний, двухслойный – 8,5 Гбайт; двухсторонний, однослойный – 9,4 Гбайт и двухсторонний, двухслойный – 17 Гбайт.

Основное направление развития индустрии оптических дисков связано с использованием лазера сине-фиолетового диапазона с длиной волны 405 нм. К настоящему времени преодолены основные трудности и ряд фирм (прежде всего, Sony) приступили к серийному выпуску Blu-Ray Disc. На носителе формата 5,25 дюйма выпускаются диски с емкостью 23,3 Гбайт/25 Гбайт/27 Гбайт/50 Гбайт/100 Гбайт.

При большой емкости и высокой надежности слабым местом этих устройств является большое время доступа (десятки и даже сотни миллисекунд).

И в заключение обзора устройств внешней памяти необходимо отметить, что сравнительно недавно и достаточно неожиданно появились электронные устройства внешней памяти USB Flash носителями и популярность их растет, а причин тому несколько. Прежде всего – это постоянно снижающаяся розничная цена при постоянно повышающейся емкости (2008 г. – 32 Гбайт). Вторая причина связана с широким распространением стандарта USB. Все современные материнские платы имеют встроенные USB-порты, а в компьютерных корпусах эти

порты чаще выносятся на лицевую панель, обеспечивая легкость подключения любой периферии. И третья причина – это удобство флэш-памяти. Они легкие, малогабаритные, не подвержены влиянию магнитных полей, терпимы к температурным перепадам и механическим воздействиям, их носители не подвержены износу и информация на них может храниться очень долго. Область их применения пока достаточно узка. Они служат для хранения и переноса информации между компьютерами. Для более серьезных функций они (пока) не используются из-за низкой скорости USB-порта (1,5 Мбайт/с).

Печатающие устройства. Они служат для вывода программ, данных и результатов обработки на бумажную ленту различной ширины, а также на отдельные листы бумаги. Было разработано большое количество разнообразных устройств печати, применяемых для различных классов компьютеров (рис. 2.9), большая часть из которых в настоящее время уже не используется.

Устройства *ударного действия* сравнительно дешевы, позволяют изготавливать несколько копий, качество печати вполне удовлетворительное для традиционных применений компьютеров. К недостаткам этих устройств относят повышенный уровень шума и сравнительно невысокую надежность. На больших компьютерах наибольшее распространение в свое время получили высокоскоростные алфавитно-цифровые печатающие устройства (АЦПУ). Скорость печати этих устройств от 600 до 1800 строк в минуту, они относятся к построчно-печатающим устройствам параллельного действия. Знаконоситель в этих устройствах – литерный, обычно вращающийся с высокой скоростью цилиндрический барабан (может быть шаровой или ленточный носитель литер). Устройство содержало ряд ударных механизмов, равных количеству знаков в строке (длине барабана), а также буферное запоминающее устройство, хранящее информацию об одной строке. На малых компьютерах применялись рычажно-литерные и знакосинтезирующие устройства последовательного действия, в которых печатающий узел последовательно знак за знаком пробегает строку. Эти устройства имели невысокую скорость печати, малые размеры и стоимость, так как, во-первых, использовалась кинематика телетайпа (пишущей машинки), а во-вторых, изображение знака создавалось игольчатой матрицей, содержащей 5x7 или 7x9 точек.

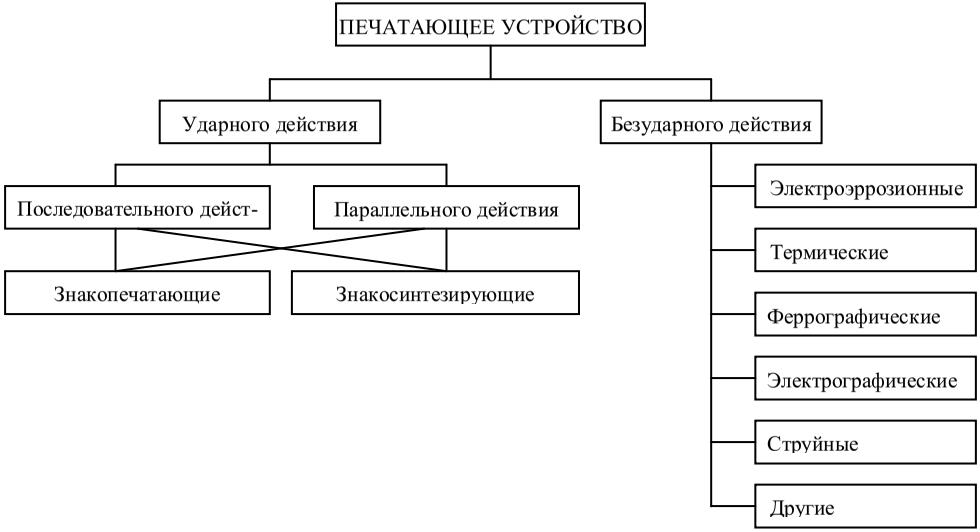


Рис. 2.9. Разновидности печатающих устройств

В настоящее время подобные принтеры используются при печати «под копирку»: на специализированных бланках «строгой отчетности», в некоторых банковских системах и т.п., когда необходимо четко идентифицировать, что данная копия документа была сделана именно с данного оригинала.

Принтеры *безударного действия* лишены многих недостатков ударных. Наибольшей популярностью в настоящее время пользуются лазерные (электрооптические) и струйные принтеры. Они обладают очень высокой скоростью печати, очень высоким качеством и надежностью. Лазерные принтеры обладают большей скоростью печати, надежностью и дешевле в обслуживании, хотя сами принтеры стоят дороже. Применяются для печати большого количества текстов, документов, для коллективного использования в различных организациях. Струйные принтеры существенно более компактные, обладают меньшим энергопотреблением, позволяют получить выше качество при печати фотографий и других цветных изображений, а также, в отличие от лазерных принтеров, при печати не нагревают носитель. Применяются, как правило, для домашнего использования, а также для специализированной печати цветных изображений единичными тиражами: фотографий, буклетов, для печати на компакт-дисках.

Печатающие устройства, получившие широкое распространение в персональных компьютерах, применяются и для изготовления графических изображений (в том числе и цветных), предварительно созданных на экране графического дисплея. Правда, эти устройства принципиально отличаются от традиционных графопостроителей (плоттеров), поскольку пишущий узел последовательно пробегает всю поверхность бумаги строчку за строчкой, оставляя след в местах копируемого изображения (бумага перемещается только в одном направлении). Скорость изготовления рисунка невысока.

Можно выделить два вида таких устройств.

1. Устройства графического вывода. К ним относятся координатографы и графопостроители планшетного и рулонного типа. В этих устройствах пишущий узел двигается непосредственно по линии изображения под управлением последовательности команд и программного обеспечения графического вывода. Графопостроители ориентированы на изготовление чертежей различного формата, координатографы обладают несколько более высокой точностью и

могут применяться для гравировки и изготовления печатных плат (при этом пишущий узел заменяется резцом).

2. Устройства ввода графической информации. Это планшетные устройства (дигитайзеры), обеспечивающие считывание, т.е. распознавание графических элементов (точка, линия, элементарный фрагмент) и их кодирование – преобразование в цифровой код по установленным правилам.

По степени участия человека в процессе считывания устройства ввода графической информации разделяют на автоматические и полуавтоматические. В связи с расширением сферы использования персональных компьютеров различные фирмы освоили производство автоматических и ручных устройств, предназначенных для ввода изображений (в том числе и цветных) или текстов без использования клавиатуры – сканеров. Стоимость их высока.

Дисплеи. Дисплеи относятся к периферийным интерактивным устройствам, являясь основным средством общения человека-пользователя с компьютером. Под дисплеем следует иметь ввиду три функционально различных устройства – монитор, предназначенный для вывода информации из компьютера, различные устройства для ввода информации и графические адаптеры, которые уже давно вышли за рамки своих прямых функций вывода на экран монитора подготовленного процессором изображения, став графическими ускорителями, выполняя операции по обработке графики, включая расчетные функции. Таким образом, адаптеры, забрав часть функций ЦП, превратились в высокоскоростные графические процессоры. Очень высокая скорость вывода информации (включая графическую), удобной для восприятия человеком, и возможность оперативного вмешательства в процесс решения задачи сделали дисплей незаменимым устройством для общения с компьютером. Мониторы являются важнейшей частью графической системы. Качество изображения является ключевым фактором (хотя и не единственным), которое определяется выбранным типом монитора.

В качестве мониторов в компьютерах до последнего времени применяются два типа устройств. Это традиционные мониторы на базе электронно-лучевых трубок (ЭЛТ), которые для получения изображения используют принцип телевизионного растра, вторыми являются устройства, использующие плоские панели. Плоские панели, или экраны матричного типа, могут основываться на разных

технологиях: жидкокристаллической, плазменной (газоразрядной), твердотельной, автоэлектронно-эмиссионной и др. Все они сейчас активно развиваются, но в массовых масштабах в компьютерной области применяется только одна их разновидность – жидкокристаллические (ЖК, LCD – Liguid Crystal Display)*.

ЖК-технологии не являют собой что-то однородное, скорее всего это конгломерат решений, объединенный общим принципом. Можно рассматривать разнообразие актуальных подходов как свидетельство интенсивного развития этой области.

В течение ряда лет, начиная с 2000 г., развитие компьютерных мониторов проходило в виде скрытого или явного соревнования этих двух основных технологий. Если до 2000 г. ЭЛТ были вне конкуренции, то затем ситуация стала резко меняться.

Среди основных недостатков ЭЛТ являются следующие: относительная громоздкость, особенно при экранах большого размера – 17 дюймов и выше, глубина монитора становится соизмеримой с размером экрана, и второй недостаток, который стал более заметен в связи с появлением плоских панелей, – не плоский экран. Правда, даже такие консервативные элементы, как ЭЛТ постоянно совершенствуются. Экраны становятся более плоскими, повышается яркость и контрастность изображения, уменьшаются габариты, снижается энергопотребление. Хотя следует отметить, что серьезных

---

* Жидкие кристаллы были открыты в 1888 г. австрийским ботаником (ученым, занимающимся флорой) Фридрихом Рейницером в процессе изучения значения холестерина в растениях. Он выделил некоторое вещество, странным образом ведшее себя при нагревании, – оно мутнело и начинало течь раньше, чем обращалось в темную жидкость. Субстанцию с секретом Рейницер вручил германскому физику Отто Леману, который обнаружил у нее еще одну необычность. Жидкость в своих оптических свойствах вела себя как кристалл. Так, в науку вошло грандиозное изобретение под названием «жидкий кристалл». ЖК – так называемая метаморфоза, состояние между твердым и изотропным состояниями вещества: оно и текущее, и сохраняет порядок расположения молекул; состояние в некоторых случаях устойчиво в большом диапазоне температур. По тем временам изобретение оказалось совершенно бесполезное на практике.

Текучее вещество впервые пристроили к делу через 80 лет после его открытия: компания RCA (Radio Corporation of America) произвела на свет первый в мире функционирующий дисплей.

Во второй половине 70-х годов (1970 г.) ЖК-технология начала активно внедряться в портативных устройствах – главным образом, калькуляторах и цифровых часах.

технологических прорывов в этом классе мониторов (которые имеют длительную историю) ожидать не приходится.

Плоские панели имеют целый ряд потребительских качеств, делающих их особенно привлекательными:

абсолютно плоский экран;

отсутствие геометрических искажений;

высокая яркость (у плазменных-газоразрядных);

малая глубина, компактность;

очень низкий уровень электромагнитных излучений.

Основным недостатком ЖК-мониторов на начальном этапе их развития была их стоимость, в связи с чем они применялись только в ноутбуках. Однако, начиная с 1999 г., в производство ЖК-панелей были вложены огромные инвестиции, что привело к улучшению их качества и значительно снизило цену. Так, если в 2000 г. доля мониторов на ЖК-панелях в общем объеме выпуска составляла 5 %, то к 2006 г. составила 30 %. По данным Stanford Resourses, по объему производства мониторы с ЭЛТ в 2001 г. превзошли ЖК-мониторы более, чем в 2 раза. К 2006 г. соотношение изменилось на обратное.

Справедливости ради, необходимо отметить, что пока мониторы с ЭЛТ все еще лучше справляются с динамическим изображением компьютерных игр и видеомониторов. Кроме того, в силу менее точной цветопередачи фиксированным и, как правило, не слишком высоким экранным разрешением* ЖК-дисплеев мониторы с ЭЛТ используются для профессиональной работы с графикой (правда, в данном случае речь об экономии не идет вовсе) и если ситуация с разрешением ЖК-мониторов меняется в лучшую сторону, то о тонкой настройке цветовой температуры у них говорить не приходится.

Однако в отличие от ЭЛТ, где улучшение характеристик происходит благодаря постепенному совершенствованию технологий, развитие плоских индикаторов происходит нередко скачкообразно, благодаря появлению совершенно новых, перспективных технологий.

---

* Качество изображения принято оценивать по числу пикселов на один дюйм. Пиксель (сокращенно от слов picture cell) – элемент изображения представляет собой единицу измерения разрешения экрана (монитора) или печатного изображения и соответствует отдельной светящейся точке, цветом и яркостью которой можно управлять. Для оценки качества изображения введена единица под названием dpi-dots per inch – количество пикселей на один дюйм.

Одним из наиболее многообещающих вариантов, с технологической точки зрения, является замена в ЖК-экранах аморфного кремния на поликристаллический. Это дает возможность снизить размер транзисторов (на матрице), что позволяет увеличить разрешение экрана, снизить энергопотребление, увеличить скорость изображения. Кроме того, увеличивается рабочий ток транзисторов, а следовательно, уменьшается время переключения транзисторов, т.е. снижается инерционность на играх и видео. И, наконец, благодаря уменьшению размеров транзисторов появляется возможность создавать логические и управляющие цепи прямо на подложке, устройство становится более надежным благодаря уменьшению количества компонентов и этапов сборки панелей. Происходят обширные исследования и в других направлениях в рамках развития технологии ЖК-панелей.

Конкуренция на рынке ЖК-дисплеев настолько усиливается, что некоторые компании переключаются на производство других перспективных панелей, в первую очередь на основе органических светодиодов (ОСД – OLED-Organic electroluminescent diods), неорганических электролюминесцентных дисплеев, светоизлучающих полимеров и т.п.

При использовании мониторов для ввода информации, кроме отмеченной выше клавиатуры, применяются еще два типа устройств, повышающих эффективность работы пользователя:

устройства прямого указания, позволяющие осуществлять ввод, непосредственно указывая положение точки на экране монитора; сюда относятся световое перо (внутри которого находится фотоэлемент, реагирующий на освещенность экрана), а также сенсорные экраны (тач-скрин, touch-screen), позволяющие пользователю давать указания компьютеру, прикасаясь пальцем к экрану;

устройства косвенного указания, позволяющие задавать координаты точек на экране с помощью графических планшетов, рычагов со сферой (джойстик) и т.п. (наибольшее распространение получили устройства типа «мышь», они предназначены для быстрого перемещения курсора (специального указателя, светящейся точки экрана) в любую позицию экрана и включения определенных функций в указанных точках, «мышь» перемещается по любой

гладкой горизонтальной поверхности; несколько кнопок, имеющихся на ее поверхности, аналогичны по своему назначению часто употребляемым клавишам клавиатуры).

2.3. Типы и характеристики компьютерной графики

Современные компьютеры все шире применяются для построения изображений (рисунков), используемых в научных исследованиях; для наглядного представления результатов; в конструкторских разработках, тренажерах, компьютерных играх; в инженерном, издательском, рекламном деле и других областях. Компьютерная графика служит основой *анимации*, под которой понимается изменение вида, формы, размеров, расположения объектов на экране, создающее эффект мультипликации. Различают три основных типа компьютерной графики: растровую, векторную и фрактальную графику. Обычно особо выделяют еще трехмерную (3D – three-dimensional) графику как средство построения объемных изображений. По цветности различают черно-белую и цветную компьютерную графику, а по областям применения – инженерную, научную, деловую, игровую (развлекательную) компьютерную графику, компьютерную полиграфию и другие типы.

В *растровой графике* изображение строится как множество точек, так называемых пикселов. *Пиксел* (сокращение от слов picture cell – элемент изображения) представляет собой единицу измерения разрешения экрана (монитора) или печатного изображения и соответствует отдельной светящейся точке, цветом и яркостью которой можно управлять. Растр экрана монитора с диагональю 20-21" может содержать от 0,3 млн до 3 млн пикселов. Поскольку изображение может быть цветным, для кодирования одного пиксела может потребоваться до трех байт информации. На весь экран, следовательно, может потребоваться от 1 до 10 Мбайт, т.е. весьма значительный объем, но изображение, тем не менее, может быть довольно грубым. Качество изображений принято оценивать по числу пикселов на 1" длины. Единицу такого измерения называют dpi – dots per inch. Для газетных иллюстраций достаточно около 70 dpi, для полноцветной полиграфической печати – 200-300 dpi, для фотоэкспонирующих устройств профессионального класса – 2500 dpi, тогда как экран монитора обычно обеспечивает лишь несколько

десятков (например, 70) dpi и расстояние между соседними точками около 0,25 мм, что недостаточно для получения изображений высокого качества.

При растрировании изображения на него как бы накладывается сетка линий, разбивающая его на квадратные ячейки. Число линий на дюйм Lpi (Lpi – lines per inch) называется *линиатурой.* Для лазерных принтеров рекомендуемая линиатура составляет 65-100, для газет – 65-85, для книг и журналов – 85-133, для художественных и рекламных работ – 133-300. Интенсивность тона, так называемая *светлота,* определяется числом точек (пикселов) в ячейке растра. Для человеческого глаза рекомендуется 256 уровней тона, т.е. в ячейке должно помещаться до 16*16 = 256 пикселов. Для изменения уровня тона можно также изменять размеры пикселов; максимальный размер пиксела равен, очевидно, размеру ячейки растра.

Растровая графика позволяет строить изображения очень высокого качества, но, как видно из приведенных оценок, для этого требуется очень большой объем компьютерной памяти (например, для журнальной иллюстрации – до 130 Мбайт и более). Помимо больших запросов на память, недостатком растровой графики являются трудности увеличения изображения для анализа его деталей. Поскольку при увеличении объем запасенной информации сохраняется, то без принятия специальных мер увеличение изображения приводит лишь к так называемой *пикселизации* – укрупнению отдельных пикселов с сохранением их численности. Изображение становится все более «зернистым» (фотографический термин), грубым, искаженным. Сглаживание пикселизации представляет собой самостоятельную проблему.

Если основным элементом растровой графики является точка, то в *векторной графике* основным элементом является линия (прямая или кривая). Объем памяти, требуемый для хранения линии, не зависит от ее длины, так как в памяти компьютера линия представляется формулой с несколькими параметрами, а не точками. Обычно ограничиваются линиями не выше третьего порядка, для построения которых достаточно иметь не более девяти коэффициентов и абсциссы двух концевых точек. При изменении размера линии меняются лишь параметры, а объем памяти сохраняется. Линии имеют свойства: форму, цвет, толщину, тип (сплошные, штриховые, пунктирные и т.п.). Для сохранения одной линии дос-

таточно 20-30 байтов оперативной памяти. Замкнутые линии имеют свойство заполнения. Заполнение описанного линией контура может быть выполнено цветом (несколькими цветами) или текстурой (узором). Концы линии – узлы, которые также обладают свойствами, например могут быть точками, кружками, квадратиками, различными фигурками. На экран линия по-прежнему выводится точками (таковы особенности монитора), но координаты этих точек вычисляются, а не хранятся в памяти компьютера.

Линия – элементарный, простейший объект векторной графики. Простейшие объекты могут объединяться в более сложные, например плоские и объемные фигуры. Типичные объекты сохраняются в памяти компьютера. Векторная графика позволяет легко увеличивать изображение или его фрагменты, например план дома или квартиры, чертеж механизма или схемы с сохранением их качеств: можно поворачивать изображения, совмещать их, изменять угол зрения, совершать другие манипуляции. При этом используются некоторые математические основы векторной графики. Даже достаточно сложные композиции, содержащие тысячи объектов, расходуют лишь десятки и сотни килобайт памяти. Векторная графика мало пригодна для создания художественных изображений и обычно применяется в оформительских, чертежных, проектно-конструкторских работах, системах автоматизированного проектирования (например, архитектурного проектирования) и аналогичных приложениях.

Фрактальная графика, как и векторная, также вычисляемая, но в ней в памяти компьютера не сохраняются никакие объекты, а хранятся лишь их формулы. Изображение строится согласно уравнению или системе уравнений. Меняя коэффициенты (параметры) уравнений, можно получить другое изображение. Характерная особенность фрактальной графики – наследование свойств. Например, фрактальный треугольник (точнее, его формулы) – простейший фрактальный объект. Можно построить треугольник другого размера с сохранением свойств исходного (например, равносторонний треугольник). Таким путем можно строить изображения необычного вида: декоративные узоры, орнаменты, имеющие очертания снежинок, кристаллов, листьев, сложных геометрических фигур.

Трехмерная графика широко применяется в таких областях, как научные расчеты, инженерное проектирование, моделирование фи-

зических процессов и технических объектов, а также в обучающих системах и «индустрии развлечений» (игр). Для создания модели трехмерного объекта используются *геометрические примитивы* (куб, параллелепипед, шар, эллипсоид, конус и др.) и гладкие поверхности, описываемые кусочно-гладкими бикубическими полиномами. Вид поверхности задается сеткой расположенных в пространстве опорных точек. Участки поверхности между опорными точками – границы объекта, которые обладают различными свойствами и могут быть гладкими, шероховатыми, прозрачными, непрозрачными, зеркальными и т.п. В соответствии с этими свойствами поверхности закрашиваются тем или иным способом. Движение объектов и анимация воспроизводятся движением геометрических примитивов и опорных точек по заданным законам. Для построения трехмерных изображений и анимации используется достаточно сложное алгоритмическое и программное обеспечение.

Цветность изображения характеризуется цветовой моделью и цветовым разрешением. Под *цветовой моделью* понимают способ разделения цвета на основные цвета. В наиболее простой цветовой модели, используемой в мониторах и цветных телевизорах, любой цвет считается состоящим из трех основных компонент: красного, зеленого и синего цветов, смешанных в определенной пропорции. Совмещение трех основных компонент в равной пропорции дает белый цвет. В такой модели цвет ячейки растра можно изобразить вектором, исходящим из начала координат в пространстве трех основных цветов. При этом проекции вектора дают относительный вклад основных цветов, а его модуль – интенсивность цвета. К трем основным цветам обычно добавляют для удобства еще черный цвет (цвет экрана). Имеются и другие цветовые модели.

Под *цветовым разрешением,* которое называется также *глубиной цвета*, понимается метод кодирования цветовой информации. И от него зависит, сколько цветов на экране может воспроизводиться одновременно. Таблица данных, в которой хранится информация о том, каким кодом закодирован тот или иной цвет, называется *цветовой палитрой.* Если на кодирование цвета отводится 1 бит информации, изображение будет двухцветным (черно-белым); один байт информации позволяет закодировать 256 цветов, два байта – 65536 цветов (так называемый режим High Color), три байта – около 16,5 млн цветов (режим, называемый True Color). В последнем

случае для кодирования каждого из трех основных цветов отводится один байт информации. При работе во всемирной сети Internet используется так называемая *«безопасная палитра»*, содержащая всего 216 цветов и жестко задающая их коды, а поэтому пригодная для любых компьютеров, подключенных к сети, в том числе не совместимых с IBM PC.

2.4. Архитектура компьютеров, понятие об интерфейсах и их разновидности

Большое значение имеет концепция взаимосвязи отдельных частей компьютера, структура или (как это принято называть в информатике) архитектура вычислительной системы. Архитектура компьютера значительно влияет на производительность и эффективность использования вычислительных машин.

Термин «*архитектура компьютера*» был введен в начале 60-х годов прошлого века одной из групп специалистов в области компьютеров фирмы IBM. Он предназначался для описания общей программной модели семейства IBM 360 на уровне языка Ассемблер, семейств компьютеров одинаковой архитектуры с точки зрения данного языка. Архитектуру компьютера или вычислительной системы следует понимать как систему основных функциональных средств, доступных пользователю, и принципов организации процесса обработки информации. В некоторых случаях, говоря о структуре (архитектуре) персональных компьютеров, употребляется термин «анатомия компьютера».

В современной литературе термин «архитектура» употребляется в различных контекстах, например, для теоретической классификации способа обработки данных (архитектура фон Неймана, параллельная архитектура, нейрокомпьютинг, нейтронный компьютинг, квантовые компьютеры, биокомпьютеры на основе ДНК-логики, молекулярные вычисления на основе молекулярной логики и т.п.), для определения принципов организации и функционирования вычислительных систем (архитектура IBM EISA, классическая архитектура, хабовая архитектура, архитектура DEC AXP и т.п.).

Прежде чем перейти к описанию различных структур построения компьютеров и динамики развития архитектуры, необходимо отметить, что в любом компьютере имеется еще один аппаратур-

ный блок, не указанный на схеме рис. 2.2 и имеющий непосредственное отношение к структуре компьютера.

Непосредственное присоединение различных функциональных элементов (УВВ и ОП) к центральному процессору осуществляется через определенные точки, именуемые точками сопряжения (стык). Поскольку все три основных блока компьютера (ОП, ЦП и УВВ) имеют различные характеристики своих выходных параметров, то для обеспечения их совместимости необходимо еще одно устройство, решающее эти задачи. Совокупность средств, обеспечивающих логические, электрические и конструктивные условия совместимости ЦП и функциональных устройств в точках сопряжения и их взаимодействия, получила название средств сопряжения, или *интерфейса*.

В качестве *логических условий* должны быть заданы виды сигналов (адресные, информационные и управляющие) и их количество, система кодирования и форма передачи данных, функции адресных и управляющих сигналов и т.п.

В качестве *электрических условий* обеспечения совместимости задаются значение напряжений (логических) двоичных сигналов, временные параметры этих сигналов, нагрузочная способность по входу и выходу сопрягаемых цепей и т.д.

К *конструктивным условиям* обеспечения совместимости относятся конструктивные соединения (тип разъема), распределение контактов в разъемном соединении, допустимые типы кабеля и др.

В отечественной практике для описания совокупности схемотехнических средств, обеспечивающих непосредственное взаимодействие составных элементов систем обработки данных (компьютерных сетей, систем передачи данных), подсистем периферийного оборудования, используются понятия «*интерфейс*», «*стык*» и «*протокол*».

Под *стандартным интерфейсом* понимается совокупность унифицированных, аппаратурных, программных и конструктивных средств, необходимых для реализации взаимодействия различных функциональных элементов в автоматизированных системах сбора и обработки информации при условиях, предписанных стандартом и направленных на обеспечение информационной, электрической и конструктивной совместимости.

Стык – место соединения устройств передачи сигналов данных, входящих в систему передачи данных. Это понятие используется вместо понятия интерфейса для описания функций и средств сопряжения элементов средств связи и систем передачи данных (СПД).

Под *протоколом* понимается строго заданная процедура или совокупность правил, регламентирующая способ выполнения определенного класса функций. Взаимосвязь понятий интерфейса и протокола не всегда однозначна, так как практически любой интерфейс содержит в большей или меньшей степени элементы протокола, определяемые процедурами и функциональными характеристиками интерфейса.

Основное назначение интерфейсов, стыков и протоколов – унификация внутримашинных связей.

Различают несколько видов интерфейсов:

системный (*внутрисистемный*), представляющий собой совокупность унифицированной магистрали для передачи информации электронных схем, служащих для согласования, преобразования и управления сигналами на магистрали, а также унифицированных алгоритмов (протоколов) обмена информацией между отдельными устройствами компьютера;

периферийного оборудования, включающий универсальные (к ним можно отнести различные типы параллельных и последовательных) и специализированные интерфейсы;

программируемых приборов, служащий для подключения нестандартной аппаратуры, измерительных и управляющих систем;

магистрально-модульных, микропроцессорных систем;

локальных вычислительных систем и т.п.

Понятия архитектуры, а также интерфейса связаны со следующими элементами, входящими в состав любого компьютера:

линии интерфейса – электрические цепи, являющиеся составными физическими связями интерфейса;

шина – совокупность линий, сгруппированных по функциональному назначению (шина адреса, шина команд, шина данных, шина состояния и т.п.);

магистраль – совокупность всех шин интерфейса.

Конструктивно (как правило) магистраль – печатная плата, обеспечивающая соединение контактов разъемов, с помощью ко-

торых к магистрали подключаются различные устройства ЭВМ (т.е. их интерфейсы).

С точки зрения шинной организации можно выделить два типа архитектур компьютера: машины с одношинной организацией (UNIBUS), имеющие общую (одну) магистраль для подключения всех устройств компьютера, и машины с многошинной организацией (MULTIBUS) и несколькими магистралями, например между ЦП и ОП одна магистраль, а между ПУ и ЦП – другая. Родоначальником промышленного внедрения общей шины является фирма DEC (Digital Equipment Corporation) и применяется она, как правило, в мини-, микрокомпьютерах и персональных компьютерах, т.е. недорогих машинах, имеющих невысокую производительность. Такая архитектура очень проста и удобна с точки зрения программирования, так как все устройства компьютера напрямую связаны между собой (каждый связан с каждым).

Многошинная организация применяется в больших компьютерах, а также в указанных выше мини- и микрокомпьютерах повышенной производительности. Разделение магистралей (например, по скоростям – низко-, высокоскоростные и т.п.), безусловно, усложняет конструкцию и удорожает ее, однако позволяет повысить производительность и эффективность использования компьютера и отдельных его частей.

Главным стимулом развития архитектуры компьютеров является повышение производительности. Поскольку классическая схема по мере повышения производительности ЦП не приводила к адекватному повышению производительности компьютера в целом, потребовалось дальнейшее ее совершенствование. На начальном этапе это было связано с некоторыми новациями, о которых частично упоминалось выше.

Таких новаций две – распараллеливание и специализация. Хотя мультипрограммная обработка не может считаться, в полном смысле, параллельной (так как в каждый момент выполняется только одна задача), тем не менее, ее относят к так называемому логическому распараллеливанию.

В рамках дальнейшего развития этой же архитектуры как большие компьютеры, так и самые первые микропроцессоры для персональных компьютеров (Intel 8086) использовали поточную (или многофункциональную) обработку, когда при выполнении очеред-

ной команды часть функциональных элементов ЦП одновременно проводит подготовку к выполнению следующей (загрузку из ОП и ее дешифровку). Дальнейшее расслоение ЦП связано с появлением процессоров ввода-вывода, арифметических процессоров (сопроцессоров), графических, криптографических процессоров и т.п. Один из способов повышения производительности вычислительной техники – специализация (как отдельных элементов компьютеров, так и создание специализированных вычислительных систем).

Специализация процессоров началась с 60-х годов прошлого века, когда центральный процессор больших компьютеров был освобожден от выполнения рутинной операции по вводу-выводу информации. Эта функция была передана процессору ввода-вывода, осуществляющему связь с периферийными устройствами.

Возможны *три способа специализации* в вычислительных машинах:

расширение системы команд универсальных компьютеров общего назначения, включение команд вычисления часто встречаемых функций с возможной аппаратной реализацией;

использование периферийных процессоров, подключаемых к универсальным компьютерам и реализующих некоторые вычислительные операции независимо от ЦП, например матричные процессоры, графические и т.п.;

создание специализированных компьютеров или процессоров, структура которых ориентирована на решение узкого класса задач большой сложности.

Следующие архитектурные новации связаны с уже упоминавшимся ранее расслоением ОП. Дальнейшее увеличение производительности предполагает многопроцессорную обработку.

Поскольку быстродействие однопроцессорных компьютеров ограничивается физическими возможностями (скоростью распространения электрических сигналов), дальнейшее совершенствование компьютеров связано и с развитием их параллелизма и переходом к так называемой параллельной архитектуре, включением в состав компьютеров большого количества основных (центральных) процессоров.

В соответствии с так называемой классификацией Флина (M. Flyne) возможны четыре типа архитектуры в зависимости от их распараллеливания:

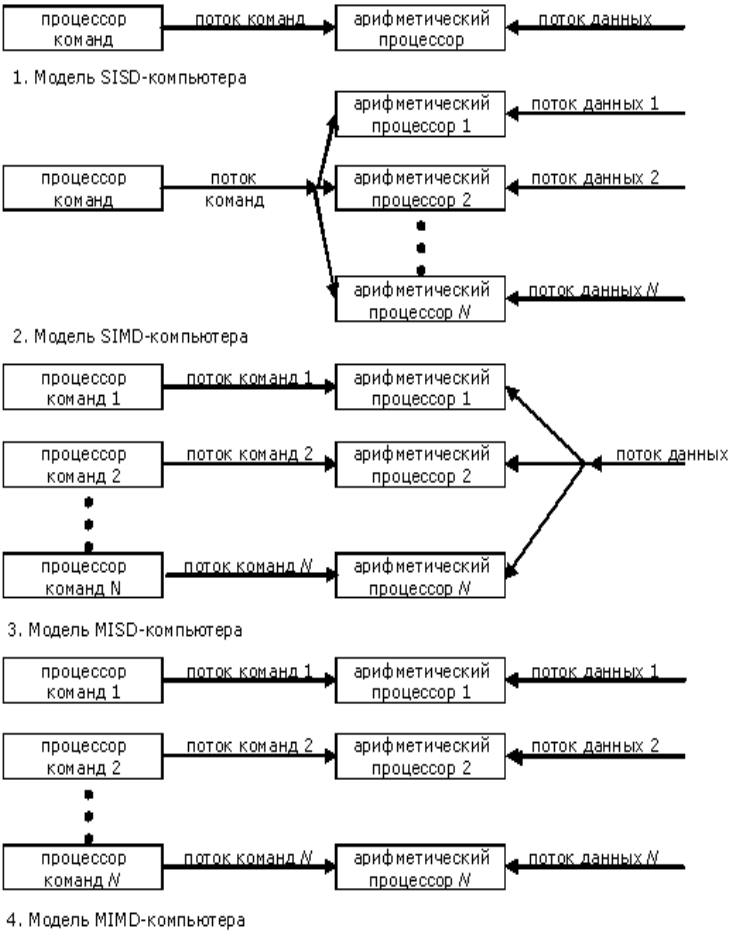


Рис. 2.10. Классификация компьютерных систем (классификация Флина)

1) без использования параллельных вычислений, когда один поток команд обрабатывает поступающий на вход один поток данных (ОКОД или SISD, single instruction, single data stream);

2) несколько процессоров по одному алгоритму (одной команде) обрабатывают одновременно несколько потоков данных (ОКМД или SIMD, single instruction, multiple data streams, параллельная обработка);

3) когда один поток данных обрабатывается большим количеством процессоров различного функционального назначения (МКОД или MISD, multiple instructions, single data stream, конвейерная архитектура);

4) самая высокая степень распараллеливания, когда множественный поток данных обрабатывает множественный поток команд (МКМД или MIMD, multiple instructions, single data stream, матричная архитектура).

Если выделить из процессора функциональные блоки, ответственные за доставку и подготовку команд, назвав эту часть процессором команд, а часть процессора, непосредственно осуществляющую обработку, назвать арифметическим процессором, то четыре разновидности распараллеливания можно представить в виде схем (рис. 2.10).

Развитие различных уровней запоминающих устройств, таких, как рассмотренные выше кэш-память, канал массовой памяти, электронные диски и т.п., также оказало заметное влияние на эволюцию архитектуры компьютеров.

Теперь рассмотрим, как же эти четыре принципа используются в конкретных типах СОД.

2.5. Классификация компьютеров, краткие характеристики суперкомпьютеров, мейнфреймов и мини-компьютеров

Хотя современные персональные компьютеры (ПК) обладают впечатляющими возможностями, которые существенно расширились в связи с появлением 64-разрядных микропроцессоров, не следует полагать, что они справятся с любой задачей. Неявным подтверждением этого является несколько уровней технических средств, традиционно развивающихся на протяжении многих лет в мире (суперкомпьютеры, мейнфреймы) и появившихся сравнительно недавно (мини-компьютеры, микрокомпьютеры, ПК, рабочие станции, серверы и суперсерверы).

Хотя сложившаяся классификация ЭВМ в последнее время подвергается значительным изменениям, тем не менее, до последнего времени эксплуатируются все перечисленные выше типы компьютеров.

Компьютерные технологии усложняются прямо на глазах, и пропорционально растет риск принятия ошибочных решений при внедрении новых аппаратных и программных продуктов. Если при покупке персонального компьютера цена ошибки не превышает сотни-другой долларов, то некомпетентное решение при компьютеризации научных, учебных или производственных подразделений может стоить потери десятков тысяч и более долларов.

При выборе вычислительных средств рекомендуется пользоваться следующим правилом: «из всех возможных вариантов построения системы наилучшим является тот, который обеспечивается наиболее простой архитектурой». Только когда ее возможностей не хватает, следует рассматривать более сложную организацию. Практикой доказано, что:

лучше использовать один быстрый компьютер, чем много медленных;

проблему создания необходимого количества рабочих мест (активных экранов) лучше решать с помощью многопользовательских систем, чем с помощью локальных систем ПК;

массовый параллелизм можно использовать только при полной уверенности в реально существующем параллелизме приложений.

Суперкомпьютеры – один из наиболее динамично развивающихся классов компьютеров, имеющих обширные и очень важные области применения.

По данным ведущих специалистов фирмы Intel рынок высокопроизводительных вычислений растет. К 2007 г. его объем уже достиг 10 млрд дол., а в некоторых секторах ежегодный прирост продаж превышает 30 %.

На начальном этапе потребность в компьютерах сверхвысокой вычислительной мощности была связана задачами в области аэродинамики, сейсмологии, метеорологии, атомной и ядерной физики, а также для физики плазмы, которые можно отнести к области моделирования сплошных сред.

Сплошную среду можно представить набором физических параметров, описывающих каждую точку некоторой трехмерной области. Численные значения параметров изменяются от точки к точке, и, как правило, под внешним воздействием они претерпевают изменения во времени. Параметрами могут быть, например, плот-

ность, скорость и температура движущегося газа, напряжение в твердом теле или составляющая электромагнитной силы.

Основные физические законы, действующие в сплошных средах, описываются системами дифференциальных уравнений в частных производных, которые на языке исчисления бесконечно малых связывают между собой значения и скорости изменения переменных в соседних точках среды. Строго математически можно доказать существование единственного решения такой системы уравнений при заданных начальных или граничных условиях – значениях переменных в начальный момент времени или на границе области. Точное решение полностью определяет поведение изучаемой материальной системы. Однако во всех реальных ситуациях, за исключением простейших, практически невозможно получить точное решение в явном виде. Поэтому приходится обращаться к численным методам, которые абсолютно точного результата не дают, но могут обеспечить нужную нам точность, правда, ценой увеличения объема арифметических вычислений, за счет замены непрерывного пространства его дискретным аналогом и аппроксимации исходного дифференциального уравнения в точках разбиения, приближенным линейным.

Реализация методов дискретизации на исследуемом пространстве приводит к необходимости решения больших систем линейных (нелинейных) уравнений для довольно большого количества параметров исследуемого пространства. По данным из различных зарубежных источников, требуемая производительность для решения задач NASA – 10^{18} FLOPS, физики плазмы $3*10^{20}$ FLOPS, предсказания погоды $10^{16} - 10^{22}$ FLOPS, молекулярной динамики $20*10^{15}$ FLOPS, вычислительной космологии 10^{-19} FLOPS (десять квинтиллионов).

В связи с этим разработчикам пришлось перейти на новый уровень размерностей для оценки производительности компьютеров и емкости оперативной и внешней памяти (табл. 2.1).

В настоящее время в связи с глобальными изменениями в технологии элементной базы суперкомпьютеров и естественным уменьшением стоимости сфера их применения значительно расширилась. Это и распределенные СУБД, криптография, нефтедобывающая промышленность и прочие применения.

Таблица 2.1

Giga	10^9
Tera	10^{12}
Peta	10^{15}
Exa	10^{18}
Zetta	10^{21}
Yotta	10^{23}

Кроме всего прочего, рядовому пользователю полезно иметь представление о суперкомпьютерах и потому, что все новейшие технологические достижения, обычно появляются в суперкомпьютерах, но рано или поздно далее становятся достоянием масс, работающих на ПК. Так было уже не раз: виртуальная память, кэш-память, конвейерное выполнение команд, предсказание ветвлений, суперскалярная архитектура и симметричное мультиплексирование, впервые нашли применение в мейнфреймах и суперкомпьютерах, уже стали привычным атрибутом настольных систем. Как показывают исследования, в среднем вычислительная мощь настольных ПК отстает от уровня производительности суперкомпьютеров на 13 лет. Иными словами, по уровню производительности сегодняшние профессиональные ПК практически полностью соответствуют суперкомпьютерам 13-летней давности.

Термин «суперкомпьютер» был использован в начале 60-х годов прошлого столетия, когда группа специалистов Иллинойского университета предложила идею реализации параллельного компьютера – проект, получивший название SOLOMON, базировался на принципе векторной обработки, которая была сформулирована Джоном фон Нейманом (J. von Neuman), и концепции матрично-параллельной архитектуры, предложенной С. Унгером (S.H. Unger) в начале 50-х годов.

В 60-х годах количество суперкомпьютеров исчислялось единицами, в 1988 г. (по данным США) их количество достигло 40 шт., в 1991 г. – 760 шт. (Cray Corp, Fujitsu, Hitachi, Nippon Electric (NEC)). После 2000 г. началось массовое производство машин этого класса.

В 80-е годы под суперкомпьютерами было принято считать вычислительные системы с производительностью не меньше 100 млн

операций с «плавающей точкой» в секунду (мегафлоп/с*), при работе с 64-разрядными словами в поле оперативной памяти не меньше одного мегабайта. «Плавающая точка» относится к двоичной версии хорошо известного в научной практике метода представления чисел, когда число записывается в виде произведения, в котором один множитель (мантисса) имеет величину между 0,1 и 1, а другой (порядок) является степенью 10, так 6600 можно записать как $0{,}66 \cdot 10^4$, а 66 – как $0{,}66 \cdot 10^2$. Такое представление удобно в научных расчетах, поскольку часто диапазон величин, входящих в задачу бывает очень большим. В компьютерах принято двоичное представление чисел с «плавающей точкой». Для суперкомпьютеров, имеющих длину слова 64 разряда, 49 разрядов отводится под мантиссу и 15 разрядов под показатель степени, что позволяет работать в диапазоне 10^{-2466} – 10^{2466} с точностью 15 десятичных разрядов мантиссы. К операциям с «плавающей точкой» относятся сложение, вычитание, умножение и деление двух операндов.

Выполнение таких операций требует несколько большего времени, чем соответствующие действия над числами с фиксированной точкой и целыми.

При определении быстродействия суперкомпьютера время, затрачиваемое на извлечение операндов из оперативной памяти и запись в оперативную память, входит в продолжительность операции.

В начале 90-х годов производительность машин этого класса достигает нескольких миллиардов операций в секунду, а позже уже десятков и сотен миллиардов. Суперкомпьютеры стали обязательным атрибутом парка вычислительной техники всех информационно-развитых стран. С их помощью решаются не только научно-технические, но и комплексные стратегические задачи.

---

* Различают пиковую и реальную производительность суперкомпьютеров. Пиковая производительность многопроцессорной системы – теоретическое значение, недостижимое на практике. Оно получается умножением пиковой производительности отдельного процессора на число процессоров в системе. Пиковая производительность отдельного процессора в общем случае получается путем умножения его тактовой частоты на максимальное число операций, выполняемое за один такт.

Реальной называют производительность, полученную при решении реальной задачи (академической или промышленной). Так, системы в рейтинге самых мощных суперкомпьютеров (TOP-500) ранжируются по результатам теста LINPACK – реальной академической задачи на решение системы линейных уравнений.

Рассмотрим, каким же образом достигается сверхвысокое быстродействие в суперкомпьютерах. Повышение быстродействия компьютера обеспечивается, в основном, двумя факторами: увеличение быстродействия микро- (нано) электронных компонентов и аппаратного обеспечения для одновременного выполнения максимально возможного числа операций.

Каким образом можно распараллелить алгоритмы пользователя и как этот параллелизм обеспечивается архитектурой компьютера?

На первых порах принцип фон Неймана предполагал единственную, как тогда казалось, архитектуру компьютера: процессор по очереди выбирает команды программы и также по очереди обрабатывает данные. Всё строго последовательно, просто и понятно, однако очень скоро выяснилось, что компьютерные вычисления обладают естественным параллелизмом, т.е. большая или меньшая часть команд программы может выполняться одновременно и независимо друг от друга. Вся дальнейшая история вычислительной техники развивалась в соответствии с логикой расширения параллелизма программ и компьютеров, и каждый новый шаг на этом пути предварялся теоретическим анализом.

Упоминавшийся ранее известный специалист по архитектуре М. Флин обратил внимание на то, что существует две причины, порождающие вычислительный параллелизм, – независимость потоков команд, одновременно существующих в системе, и не связанность данных, обрабатываемых в одном потоке команд. Если первая особенность параллелизма вычислительного процесса достаточно известна (это обычное мультипроцессирование) и не требует особых комментарий, то на параллелизме данных следует остановиться более подробно, поскольку в большинстве случаев он существует скрыто от программиста и используется ограниченным кругом программистов.

Простейшим примером параллелизма данных является последовательность из двух команд:

а) $A = B + C$; в противоположность б) $C = E * F$;
$D = E * F$; $A = B + C$.

Если строго следовать принципу фон Неймана, то вторая операция может быть запущена на исполнение только после завершения

первой операции. Однако очевидно, порядок выполнения этих команд не имеет никакого значения – операнды *A*, *B* и *C* не связаны с операндами *D*, *E* и *F* второй команды. Иными словами, обе операции являются параллельными. Практически любая программа содержит группы операций над параллельными данными.

Другой вид параллелизма данных, как правило, возникает в циклических программных конструкциях типа *DO*, в которых одни и те же операции выполняются десятки, сотни, а иногда и тысячи раз. И именно в этих операциях скрыт очень большой потенциал повышения производительности компьютеров. Известный специалист в области системного программирования Д. Кнутц (D. Knuth) показал, что циклы *DO*, занимая даже менее 4 % кода фортран-программ, требуют более половины времени выполнения задач. В связи с этим возникает естественное желание ускорить выполнение циклических вычислений, и для большинства прикладных задач решение этой проблемы найдено с помощью введения класса векторных вычислений. Для программиста вектор – лишь упорядоченный список данных или одномерная матрица $A[1{:}N]$ и $B[1{:}N]$, элементы которого должны быть занесены в память неким стандартным способом. Число элементов списка называется длиной вектора (в отличие от более известного определения вектора в математике и физике). Простейшим примером цикла *DO* может служить операция сложения двух одномерных матриц (векторов):

$A[1{:}N]$ и $B[1{:}N]$
Begin: DO $i = 1, N$
$C(i) = A(i) + B(i)$
END DO.

Понятно, что для обработки цикла, взятого в качестве примера, компьютер фон Неймана должен выполнить, как минимум, N сложений элементов векторов A и B, не считая $2N$ команд приращения индекса и условного перехода. Идея векторной обработки циклов такого рода заключается в том, что в систему команд компьютера вводится векторная операция сложения $<A + B>$, которая задает сложение всех элементов – операндов. При этом реализуются сразу две возможности ускорения вычислений: во-первых, сокращается число выполняемых процессором команд объектного кода, по-

скольку отпадает необходимость в пересчете индексов и организации условного перехода; и, во-вторых, все операции сложения элементов векторов-операндов могут быть выполнены одновременно в силу параллелизма этих операций.

В более общем случае, когда цикл *DO* содержит группу команд, можно говорить о том, что один поток команд (последовательность операций, записанных в теле цикла) обрабатывает множество потоков параллельных данных. Существует широкий класс задач (аэродинамики, ядерной физики, геологии, метеорологии, обработки изображений и т.д.), в которых процесс операций выполняемых в циклах *DO*, достаточно велик и достигает 80 – 90 %.

Теоретическое осмысление проблемы программного параллелизма привело к созданию достаточно близких по смыслу классификаций, из которых наиболее признанной считается классификация по шести уровням параллелизма, предположенная П. Треливеном (PC. Treleaven) из университета Ньюкастла (рис. 2.11).

Рис. 2.11. Классификация по шести уровням параллелизма

Три верхних уровня параллелизма занимают крупные программные объекты – независимые задания, программы и процедуры программы. Несвязанные операторы, циклы и операции образуют нижние уровни параллелизма. С учетом классификации Флина, которая была рассмотрена в разд. 2.4, становится почти очевидно, что параллелизм верхнего уровня, в основном, достигается за счет множества независимых командных потоков, а параллелизм нижнего уровня обязан своим существованием, главным образом, несвязанным потоком данных.

Каким же образом происходило развитие параллельных архитектур, и какими средствами располагают современные суперкомпьютеры для превращения программного параллелизма в характеристики производительности системы?

По мере развития вычислительной техники архитектура фон Неймана обогатилась сначала конвейером фаз операций, затем многофункциональной обработкой и получила обобщенное название SISD. Оба вида средств низкоуровнего параллелизма впервые были введены в компьютерах Control Data 6600 и 7600 в начале 70-х годов и с тех пор применяются во всех компьютерах повышенного быстродействия. Дело дошло до того, что конвейерная обработка используется даже в микропроцессорах, начиная с Intel 80486.

Из всех типов параллелизма операций архитектуры класса SISD не охватываем только один – параллелизм циклов и итераций, который тесно связан с понятием множественности потоков данных и реализуется векторной обработкой, что соответствует архитектуре SIMD (по М. Флину).

Первоначальный этап развития суперкомпьютеров связан с созданием так называемых матричных структур, реализующих SIMD-архитектуру. Это – группа процессоров, разделяющих общую память и выполняющих один поток инструкций, способных выполнять операции над матрицами, – примитивные инструкции. Количество процессоров, входящих в группу, может составлять 1000 и более.

Одним из важнейших вопросов для параллельных структур был вопрос обмена множества процессоров с модулями ОП (рис. 2.12). Разделение памяти позволяет всем процессорам параллельной системы иметь доступ к глобальным, или общим модулям памяти. В простой схеме разделения памяти (см. рис. 2.12, *а*) каждый процес-

сор непосредственно соединяется с каждым банком памяти. Недостаток здесь в том, что процессоры и банки памяти должны иметь очень большое количество соединительных линий. Другое решение (см. рис. 2.12, *б*) дает «шина» – общий канал связи, по которому каждый процессор посылает запросы к банкам памяти, а последние выдают данные. Такая «шина» может быть перегружена (и, следовательно, работать медленно), когда требуется передавать много сообщений. Еще одно решение представляет собой так называемая «сеть омега» (см. рис. 2.12, *в*), в ней процессоры связываются с модулями памяти коммутирующими устройствами, у каждого из которых два входных и два выходных канала. В такой сети каждый процессор может напрямую связываться с каждым модулем памяти, однако здесь нет нужды в таком большом количестве линий связи, которого требует система с прямыми связями.

Преимущества сети становятся все более очевидными по мере роста числа процессоров и модулей памяти. Недостаток ее в том, что иногда сообщения проходят через множество коммутирующих станций, прежде чем достигают абонента.

Рассматриваемая схема с общей, разделяемой ОП (которая на сегодняшний день используется в традиционных суперкомпьютерах), наряду с очевидными преимуществами простого программирования обменов данными между процессорами, по мере увеличения числа процессоров имеет серьезный недостаток – плохую масштабируемость. Система называется хорошо масштабируемой, если увеличение процессорных элементов (ПЭ) приводит к адекватному увеличению производительности). Это связано с увеличением потерь времени (при увеличении числа процессоров) на доступ к необходимым данным в любой из рассмотренных схем. Тем не менее, все эти схемы применялись и применяются на компьютерах различных типов, в современной классификации эта схема называется SMP (Symmetric Multi Processing).

Этого недостатка лишены системы с распределенной памятью. Здесь у каждого процессора, а в некоторых случаях группы, или кластера процессоров имеется своя собственная ОП, доступ к которой осуществляется без помех. Платой за это является невозможность использовать память соседнего вычислительного узла. Термин «Массивно параллельные системы» (MPS) применяется обычно для обозначения таких масштабируемых компьютеров с боль-

шим числом узлов (сотни, а иногда и тысячи). В соответствии с классификацией Флина такие системы идентифицируются как MSIMD.

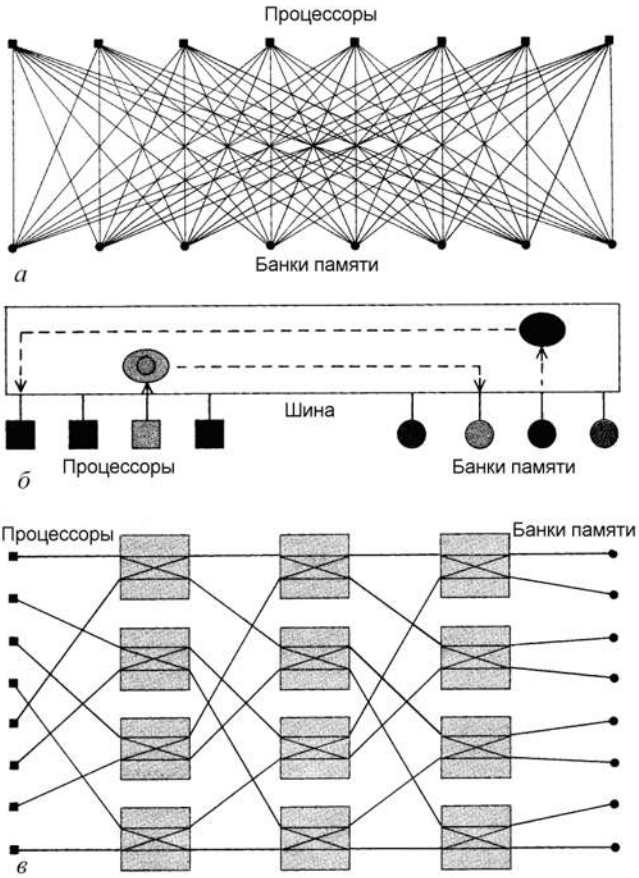


Рис. 2.12. Различные варианты обмена информацией для параллельных структур

В системах распределенной памяти устанавливаются соединения между процессорами, каждый из которых монопольно владеет некоторым количеством памяти. Простейшие схемы соединения процессоров – «кольцо» (рис. 2.13, *а*) и «решетка» (рис. 2.13, *б*). Более специализированная структура связей – «двоичное дерево»

(рис. 21.13, *г*). Особенно эффективна она в так называемых экспертных системах, в которых последовательности принятия решений могут быть представлены в виде «дерева». Можно также соединить каждый процессор с каждым (рис. 2.13, *в*), но для этого потребуется нереально большое количество соединений.

Оптимальным с точки зрения связей считается гиперкуб (рис. 2.13, *д*), реализованный в первом проекте американского суперкомпьютера. В этой схеме процессоры играют роль вершин многомерного куба и соединены его ребрами, они находятся в вершинах четырехмерного куба.

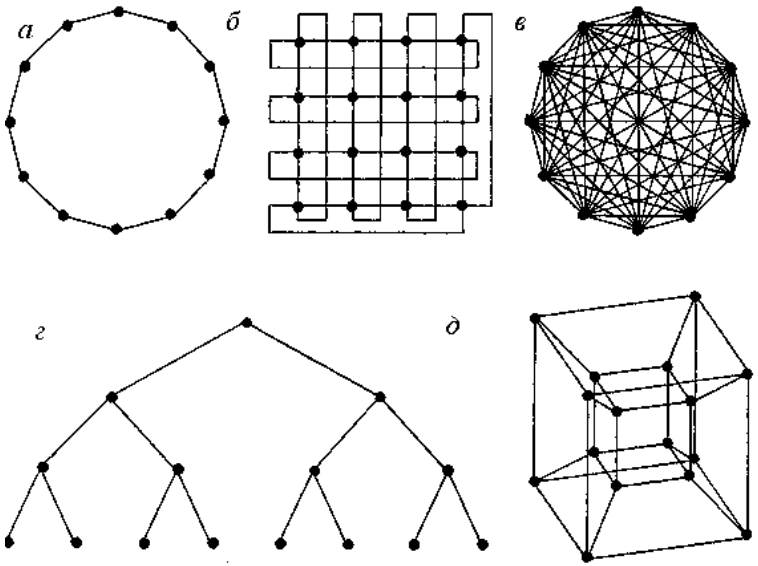


Рис. 2.13. Способы возможного взаимодействия процессоров:
а – «кольцо»; *б* – «решетка»; *в* – каждый с каждым; *г* – «дерево»; *д* – гиперкуб

В такой системе каждый процессор может посылать сообщения любому другому процессору по сравнительно короткому пути, при этом процессоры не перегружаются слишком большим количеством соединений.

Значительно расширились возможности суперкомпьютеров благодаря использованию векторно-конвейерной обработки, представляющей пространственно-временное распараллеливание процессов

обработки, внедрение которой связывают с именем Сеймура Крея, основателя фирмы Cray Research Inc., которая до 90-х годов лидировала в производстве суперкомпьютеров.

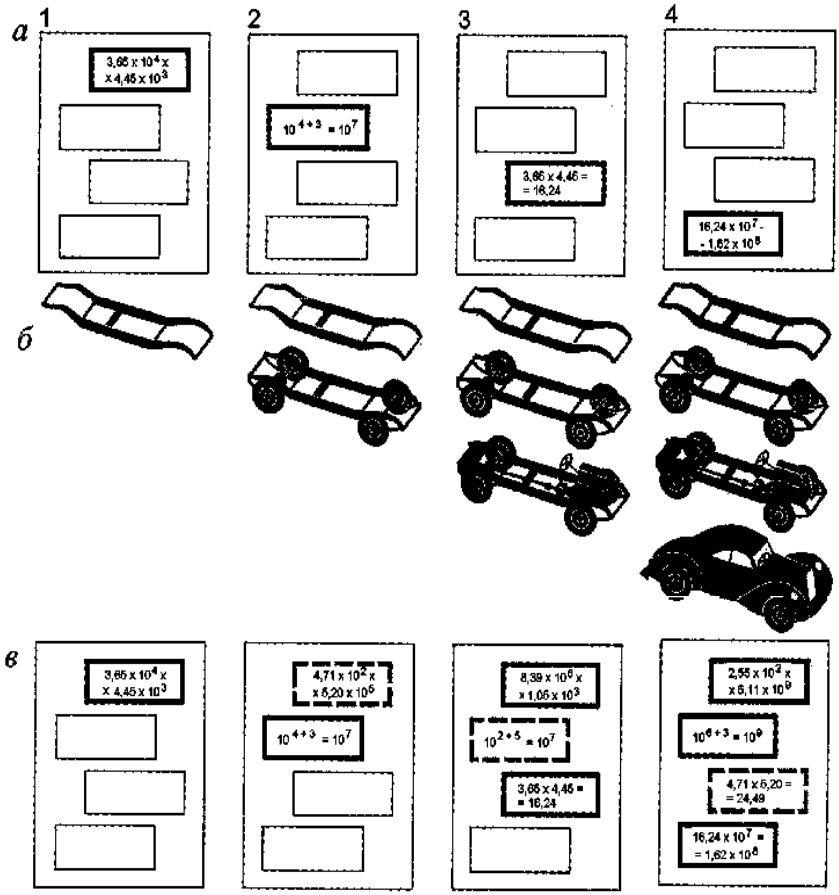


Рис. 2.14. Принцип конвейерной архитектуры

Конвейерная обработка – метод повышения быстродействия одиночного процессора – подобно линии сборки автомобиля на конвейере (рис. 2.14). Идея, впервые использованная Генри Фордом, (см. рис. 2.14, *б*) оказалась весьма перспективной и в области вычислительной техники.

Любая вычислительная операция распадается на ряд шагов, выполняемых специализированными компонентами процессора (реализующими его систему команд). В обычном процессоре (см. рис. 2.14, *а*) пока одна компонента работает, остальные бездействуют (простаивают). На рис. 2.14, *в* операция умножения распадается на следующие шаги: извлечение из памяти порядка и мантиссы обоих чисел и выделение этих частей, сложение порядков, умножение мантисс и представление результатов в требуемой форме.

Процессор с конвейерной организацией действует аналогично автомобильному конвейеру, при выполнении шага операций над одной парой чисел другая пара поступает для выполнения того же шага, не дожидаясь, пока первая пара пройдет все этапы операции. Естественно, что любая конвейерная обработка имеет смысл при массовом производстве и эффект от нее наступает после заполнения конвейера вновь поступающими данными («мертвое» время конвейера). При этом скорость конвейерной обработки зависит от длины конвейера. Так, если длина конвейера равна N компонентам, то после заполнения такой процессор будет работать в N раз быстрее обычного процессора последовательной обработки. Выбор длины конвейера – вопрос достаточно сложный, поскольку любая параллельная обработка (в том числе и векторная) предполагает соблюдение двух основных требований – независимость потока команд и независимость потока данных. Большие сложности при работе конвейеров представляют ветвящиеся алгоритмы. Хотя разработчики принимают немалые усилия для преодоления этих трудностей (сцепление конвейеров, «спекулятивное» выполнение инструкций, эвристическое предсказание переходов и т.п.). Начало конвейерной обработке положила поточная обработка в первых суперкомпьютерах с разделением процессоров на процессоры команд (осуществляющие доставку и дешифровку команд данных) и процессоры обработки данных, производящие непосредственную обработку. При этом для исключения потерь времени на доставку дешифровка следующей команды осуществляется процессором команд во время выполнения текущей команды процессорами обработки данных. Эта процедура в настоящее время реализована практически во всех современных микропроцессорах (начиная с Intel 8086).

Как известно, жизнь развивается по спирали. В 1972 г. после преодоления значительных проблем на аппаратном и программном уровне в исследовательском центре NASA в Эймсе был установлен первый в мире векторный суперкомпьютер с матричной структурой ILL IAC4. С ним связано введение элементов MSIMD архитектуры и начало параллельного программирования. Изначально система должна была состоять из четырех квадрантов, каждый из которых включает 64 процессорных элемента (ПЭ) и 64 модуля памяти, объединенных коммутатором на базе сети в гиперкуб. Все ПЭ квадранта обрабатывают векторную инструкцию, которую им направляет процессор команд, причем каждый выполняет одну элементарную операцию вектора, данные для которой сохраняются в связанном с этим ПЭ модулем памяти. Таким образом, квадрант ILLIAC4 способен одновременно обрабатывать 64 элемента вектора, а вся система из четырех квадрантов 256 элементов.

Результаты эксплуатации этой системы получили неоднозначную оценку. С одной стороны, использование суперкомпьютера позволило решить ряд сложнейших задач аэродинамики, с которыми не могли справиться ни одни из существующих в США машин. С другой стороны, ILLIAC4 так и не были доведены до полной конфигурации из 256 ПЭ, практически разработчики ограничились лишь одним квадрантом. Причинами явились не столько технические сложности в наращивании числа ПЭ системы, сколько проблемы, связанные с программированием обмена данными между процессорными элементами через коммутатор модулей памяти. Все попытки решить эту проблему с помощью системного программного обеспечения потерпели неудачу, в результате каждое приложение требовало ручного программирования передач коммутатора, что и породило неудовлетворительные отзывы пользователей.

Ни в 60-х годах, ни позднее удовлетворительное решение двух таких принципиальных проблем, как программирование параллельной работы нескольких сотен процессоров и при этом обеспечение минимума затрат счетного времени на обмен данными между ними, не было найдено. В связи с этим компьютеры данного типа были не в состоянии удовлетворить широкий круг пользователей и имели весьма ограниченную область применения.

Правда, было бы большой ошибкой считать, что развитие суперкомпьютеров матричной архитектуры не дало никаких результатов. Во-первых, ряд теоретических наработок и практическая реализация параллельных сверхскоростных вычислений были в дальнейшем использованы уже на новом витке развития этого типа компьютеров, и, во-вторых, в прикладной математике сформировалось самостоятельное направление по параллельным вычислениям, столь актуальное направление на современном этапе.

Решение большинства перечисляемых проблем было найдено в конце 60-х годов «патриархом» суперкомпьютерных технологий Сеймуром Крейем. Он представил машину, основанную на векторно-конвейерном принципе обработки данных.

Отличие векторно-конвейерной архитектуры от архитектуры матричных компьютеров заключается в том, что вместо множества ПЭ, использующих одну и ту же команду над разными элементами вектора, применяется единственный конвейер операций, имеющий всего один вход и один выход результата. Другими словами, в суперкомпьютерах с конвейерной обработкой данные всех параллельно исполняемых операций выбираются и записываются в единую память (не разделяемую) и отпадает необходимость в коммутаторе процессорных элементов, основной проблеме матричных суперЭВМ.

Дальнейшее развитие векторно-конвейерной обработки связано с многоконвейерной и построением многоконвейерных цепочек (или иначе векторно-командных цепочек). Суть его заключается в следующем: если в программе встречаются две связанные векторные операции (т.е. результат первой операции служит операндом второй), то в отличие от связанных скалярных операций, когда выполнение второй операции начинается только после завершения предыдущей, обе векторные команды могут обрабатываться практически параллельно, что удваивает производительность системы.

Допустим, программист задает цикл.

Do $i = 1, N$
$C[i] = A[i] + B[i]$
$E[i] = C[i] * D[i]$
Enddo

Тогда после векторизации компилятор организует две векторные команды

$$C = A + B$$
$$E = C * D$$

Формально вторая операция может стартовать только после окончательного вектора результата операции *C*, однако структура процессора позволяет запустить связанную операцию, как только сформируется первый элемент *C*.

В результате такого зацепления векторных операций суммарное время выполнения исходного цикла не на много превосходит этот показатель для одной векторной команды (рис. 2.15).

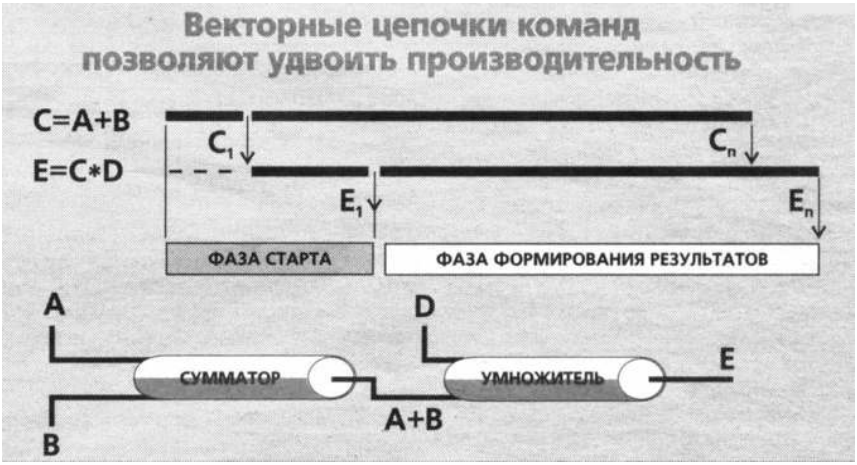


Рис. 2.15. Сцепление конвейеров

Важным достоинством векторно-конвейерной обработки является возможность использования традиционных (последовательных) языков программирования, а также разработка так называемых «интеллектуальных компиляторов», способных обнаруживать параллелизм в программах последовательного действия и преобразовывать их в векторизированный код. Следует отметить, что параллельные компиляторы хотя и называются интеллектуальными, в качестве обязательных средств содержат интерактивные оптимиза-

торы и анализаторы затрат, а знание особенности функционирования кэш-памяти позволяет программисту значительно увеличить производительность выполнения задач. В то же время в таких наиболее распространенных языках, как Фортран и Си, включены средства параллельного программирования.

До 90-х годов многопроцессорные, многоконвейерные суперкомпьютеры с общей для всех процессоров памятью доминировали, но затем ситуация стала возвращаться (на новом уровне) к системам с разделяемой памятью (вспомним ILLIAC4). Это объясняется тем, что примерно до середины 90-х годов XX в. основное направление развития суперкомпьютерных технологий было связано с построением специализированных многопроцессорных систем из массовых микросхем. Один из сформировавшихся подходов, упоминавшийся ранее, – SMP (Symmetric Multi Processing), подразумевал объединение многих процессоров с использованием общей памяти, что сильно облегчало программирование, но предъявляло высокие требования к самой памяти. Использовалась как поточная, так и векторная обработка. Сохранить быстродействие таких систем при увеличении количества узлов до десятков было практически невозможно. Кроме того, этот подход оказался самым дорогим в аппаратной реализации. На порядок более дешевым и практически бесконечно масштабируемым оказался упоминавшийся ранее способ MPP (Massively Parallel Processing), при котором независимые специализированные вычислительные модули объединяются специализированными каналами связи, причем те и другие создавались под конкретный компьютер и ни в каких других целях не применялись.

Идея создания кластерных рабочих станций явилась фактически развитием метода MPP, поскольку логически MPP не сильно отличается от обычной локальной сети. Локальная сеть стандартных персональных компьютеров при соответствующем программном обеспечении, использовавшаяся как многопроцессорный суперкомпьютер, и стала прародительницей современного кластера. Сейчас слова «кластер» и «суперкомпьютер» в значительной степени синонимы (хотя традиционные кластеры по-прежнему имеют широкое распространение). Эта идея получила воплощение, когда благодаря оснащению персональных компьютеров высокоскоростной шиной PCI и появлению дешевой, но быстрой сети Fast Ethernet кластеры стали

догонять специализированные MPP-системы по коммуникационным возможностям. Это означало, что современную MPP-систему можно создать из стандартных серийных компьютеров при помощи серийных коммуникационных технологий, причем такая система обходится дешевле, в среднем, на два порядка.

Некоторым промежуточным архитектурным решением представляется разработанный фирмой IBM суперкомпьютер SP (Scalable POWERparallel), получивший широкое распространение. Он представляет так называемую MSIMD-архитектуру, которую иногда называют массивно параллельной обработкой по схеме неразделяемых ресурсов, а иногда масс-процессорной макроконвейерной или динамической сетевой архитектурой. Компьютер построен на основе фактически автономных рабочих станций этой фирмы RS-6000, на базе 64-разрядного RISC-процессора POWER*. Модули объединяются с помощью специализированной коммуникационной матрицы. Система построена по модульному принципу, расширяется от 16 до 152 узлов, структура коммуникационной матрицы может настраиваться на конкретное приложение.

Мощный толчок развитию кластерных технологий дал быстрый рост производительности вновь выпускаемых массовых процессоров. Это сделало высокопроизводительные решения доступными даже для отечественных производителей и привело к появлению отечественных суперкомпьютеров на уровне западных и японских моделей. Самый мощный кластер в России на 2005 – 2006 гг. – МВС 15000 БН с реальной производительностью 5,3 TFlops, построен из вычислительных узлов компании IBM на базе процессоров POWER PC и системной сети Myrinet.

Доля кластеров в списке суперкомпьютеров за период с 2000 до 2004 г. увеличилась с 2,2 до 60,8 %. При этом более 71,5 % процессоров, используемых для создания суперкомпьютеров, – массово выпускаемые процессоры компаний Intel и AMD.

Кластерные технологии используются и в новейших суперкомпьютерных разработках ведущих изготовителей: например, в самом мощном суперкомпьютере IBM BlueGene/L с производитель-

* POWER – P(erformance) O(ptimization) W(ith) E(nhanced) R(ISC).

ностью более 136 TFlops использовались многие элементы кластерной архитектуры.

Сфера применения кластерных систем сейчас нисколько не меньше, чем суперкомпьютеров с другой архитектурой. Суперкомпьютерное моделирование может во много раз удешевить и ускорить выход на рынок новых продуктов, а также улучшить их качество. Так, вместо того, чтобы создавать дорогостоящие тестовые модели новых автомобилей и проводить натурные испытания для исследования их безопасности (разбивать их о стенку), можно быстрее и точнее все посчитать на компьютерных моделях. Благодаря этому многим западным автомобильным концернам удалось сократить срок разработки новых моделей автомобиля в пять раз, с десяти до двух лет.

Компьютерная обработка геофизических данных позволяет создавать высокодетализированные модели нефтяных и газовых месторождений, обеспечивая более эффективную, безопасную и дешевую разработку скважин. В табл. 2.2 и 2.3 представлены сравнительные данные по различным областям использования суперкомпьютеров и сравнительные данные их производительности в этих областях.

Несколько слов об архитектуре суперкомпьютеров на базе кластерных систем, как наиболее перспективном направлении развития подобного вида компьютеров. Как уже указывалось, кластер состоит из вычислительных узлов на базе стандартных процессоров. Большинство таких систем высшего уровня (Top 500) выполнены на процессорах Intel (Xeon, Xeon EM64, Itanium 2). Часто используются процессоры POWER 2 и POWER 2 PC компании IBM. Одно время популярностью пользовались процессоры AMD (Operton и его многоядерные версии).

Каждый узел работает под управлением своей копии операционной системы, в большинстве случаев – Linux. Состав и мощность узлов могут быть разными в рамках одного кластера, однако чаще они строятся из однородных компонентов.

Таблица 2.2

Сравнительные данные по использованию суперкомпьютеров

Области использования	В мире (Top 500), %	В СНГ (Top 50, 2-я редакция), %	В СНГ (Top 50, 1-я редакция), %
Промышленность: электронная; тяжелая (автомобильная, авиационная, металлургия и др.); добывающая (геологоразведка, нефте- и газодобыча)	44,3	16	2
Вычислительные центры, наука и образование: суперкомпьютерные центры, университеты и научные институты (физика, математика, химия, биология, генетика), поставщики суперкомпьютеров, системные интеграторы	21,2	66	96
Прогнозы погоды и климатические исследования	18,5	–	–
Стратегические исследования: программы Министерства обороны, космическая и ядерная программы	7,1	–	–
Финансы: банки, финансовые корпорации, страхование, финансовые прогнозы и консалтинг	3,5	18	2
Потребительский сектор: медицина и фармакология, транспорт, торговля, производство потребительских товаров, продуктов питания	3	–	–
Медиа: цифровые видеотехнологии, компьютерные игры и пр.	2,2	–	–
Государственное управление	0,2	–	–

Таблица 2.3

Сравнительные данные по производительности суперкомпьютеров

Области использования	Максимальный уровень, GFLops		Средний уровень, GFLops	
	в мире	в СНГ (Top 50-2)	в мире	в СНГ (Top 50-2)
Исследования	136 800	3052 (меньше в 44 раза)	1800	120 (меньше в 15 раз)
Финансы	4713	438,6 (меньше в 10,7 раза)	1500	139 (меньше в 10,7 раза)
Промышленность	3755	2032 (меньше в 1,8 раза)	1500	165 (меньше в 9 раз)

Кластер – сложный программно-аппаратный комплекс, и задача построения кластера не ограничивается объединением большого количества процессоров в один сегмент. Использование тех или иных компонентов и выбор конкретной коммуникационной среды зависит от многих факторов и, прежде всего, от типа задач, для которых строится кластер. Для некоторых хорошо распараллеливаемых задач (таких, как компоновка независимых сюжетов в видеофрагменте) главный фактор быстродействия – мощный процессор, а производительность коммуникационной среды не играет основной роли. В то же время для задач расчета сплошных сред (гидро-, газо-, аэро-, электродинамики и т.п.) важна производительность системной коммуникационной среды, иначе увеличение числа узлов в кластере будет мало влиять на скорость решения задачи.

Хотя конкретная архитектура кластера определяется архитектурой решаемой задачи, с самых общих позиций структура любого кластера включает в себя следующие взаимосвязанные элементы, показанные на рис. 2.16.

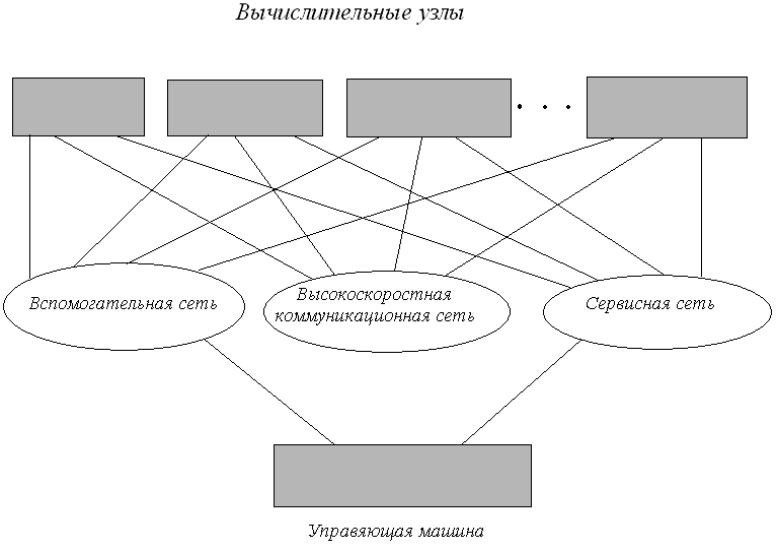


Рис. 2.16. Схема кластерной конфигурации

Системная сеть, или высокоскоростная коммуникационная среда, выполняет задачу обеспечения эффективности вычислений. Для задач, не требующих интенсивных обменов данными, наиболее доступный тип коммуникационной среды – это Gigabit Ethernet. Для задач, требующих высокие скорости обменов, применяются специализированные коммуникационные устройства, более чем в два раза превышающие пропускную способность Gigabit Ethernet. Правда, стоимость их значительно выше.

Задача эффективного доступа узлов к данным решается с помощью вспомогательной сети (как правило, Gigabit Ethernet). Сервисная сеть отвечает за распределение задач между узлами кластера, а также за управление узлами (без участия операционной системы).

Конкретная архитектура кластеров определяется областью их применения. Можно выделить четыре типа кластерных систем:

вычислительные кластеры;

кластеры баз данных;

отказоустойчивые кластеры;

кластеры для распределения загрузки.

Первые два типа кластерных систем не требуют пояснений, за исключением, может быть, того, что вычислительные кластеры – наиболее многочисленный вид кластеров и их разновидности определяются областями их применения, указанными в табл. 2.2 и 2.3.

Отказоустойчивые кластеры строят для того, чтобы наилучшим образом обеспечить надежность работы критически важных применений. Работа приложения дублируется на разных узлах, и в случае ошибки на одном из них приложение продолжает работать или автоматически загружается на другом.

Кластеры распределения загрузки используются для распределения большого потока запросов по многим серверам. Такие решения часто применяются для поддержки Web-узлов с динамическим содержимым, постоянно обращающихся к базам данных, например поисковых систем. В зависимости от размера сервиса кластеры распределения загрузки могут иметь достаточно большое количество узлов.

В заключении этого обсуждения компьютеров высшей производительности необходимо кратко остановиться на вопросе, который имеет к этим машинам непосредственное отношение. Очень пер-

спективным способом организации использования суперкомпьютеров является разработанный в рамках Европейского Союза так называемый Data Grid Project, нацеленный на создание сети европейских суперкомпьютеров Data Grid Intranet.

Грид (GRID) – перспективное направление развития информационных технологий (ИТ). Хотя оно не нашло воплощения в индустриальных стандартах (2005 г.), все страны – лидеры ИТ рынка имеют государственные программы разработки грид-технологий.

Термин «грид» применяется по аналогии с понятием «Power grid» – система, интегрирующая мощности электрических сетей в единое «хранилище» энергии, откуда она перераспределяется вне зависимости от ее источника. Внедрение таких технологий в сфере высокопроизводительных вычислений позволит кардинально упростить доступ к вычислительным ресурсам и сделать их использование на порядок более эффективным. Помимо интеграции вычислительных ресурсов грид-технологии позволят интегрировать разнородные емкости хранения информации и базы данных для создания глобального информационного пространства. Основные цели этого проекта следующие:

интеграция вычислительных мощностей – интеграция разнородных вычислительных систем в единое пространство с динамическим распределением ресурсов между приложениями;

интеграция источников данных, например интеграция в единую виртуальную базу разнородных баз данных, распределенных территориально и реализованных на разных аппаратных платформах.

Этот проект в Европе ориентирован на три основные приложения:

физика высоких энергий в рамках Международного центра ядерных исследований (CERN – Швейцария), где требуются колоссальные вычислительные мощности как для управления крупнейшими в мире ускорительными комплексами, так и особенно для обработки результатов физических экспериментов;

биология и медицина (в рамках организации CNRS – Франция);

мониторинг окружающей среды (в рамках Европейского агентства ESA – Италия).

Также следует отметить, что постоянно растущие потребности во всебольшей производительности компьютеров ставят вопрос о

предельных возможностях традиционной архитектуры фон Неймана на базе кремниевой твердотельной электрике.

Появляется ряд чисто теоретических разработок архитектур, определяющих дальнейшее развитие компьютерных технологий, – это квантовый компьютинг, молекулярный компьютинг, нейронный компьютинг, компьютер на основе ДНК-логики и т.п. Учитывая, как давно ведутся разговоры о нейрокомпьютинге, при отсутствии реальных результатов внедрения его в практическое использование трудно ожидать от всех перечисленных предложений прорывных результатов в ближайшем будущем.

Мейнфрейм – синоним понятия «большой универсальный компьютер». Мейнфреймы и до сегодняшнего дня остаются наиболее мощными (не считая суперкомпьютеров) вычислительными системами общего назначения, обеспечивающими непрерывный круглосуточный режим эксплуатации. Они могут содержать один или несколько процессоров, каждый из которых, в свою очередь, может оснащаться векторными сопроцессорами (ускорителями операций с суперкомпьютерной производительностью). Прогресс в области элементно-конструкторской базы позволил существенно сократить габариты основных устройств. Наряду со сверхмощными мейнфреймами, требующими организации двухконтурной водяной системы охлаждения, имеются менее мощные модели, для охлаждения которых достаточно принудительной воздушной вентиляции. Эти модели построены по блочно-модульному принципу и не требуют специальных помещений и кондиционеров.

Основными поставщиками мейнфреймов являлись известные компьютерные компании Amdahl, ICL, SIEMENS NIXDORF и некоторые другие, но ведущая роль всегда принадлежала, безусловно, компании IBM. Именно проект по созданию новой архитектуры компьютеров – мейнфреймов, реализованный IBM в первой половине 1960 г., определил одно из основных направлений развития компьютерной индустрии на долгие годы. Этот проект был нацелен на создание стандартизованной, хорошо масштабируемой, высоконадежной архитектуры компьютеров универсального назначения. 4 апреля 1964 г. появилось семейство вычислительных машин IBM-System/360, первоначально включающее в себя восемь моделей: нижнего (1), среднего (3) и высшего (4) уровня.

Большие универсальные компьютеры IBM 360 были первыми в мире системами, предназначенными как для коммерческих, так и для научных целей. Эта серия дополнялась в процессе развития еще шестью моделями как среднего, так и высшего (научного) уровня. Модели строились на базе стандартных аппаратурных и программных продуктов и были совместимы сверху вниз, имея единую систему команд.

В начале 1970-х годов IBM выпустила на рынок новое поколение машин, получивших название System 370, совместимых с системой IBM 360, но использовавших усовершенствованные процессоры, оперативную и внешнюю память. Это позволило расширить их возможности по поддержке одновременно работающих пользователей и более ресурсоемких и динамических приложений. Основные новации IBM 370 – возможность использования нескольких процессоров в рамках одной системы, полноценная поддержка виртуальной памяти и новый 128-разрядный блок вещественной арифметики. Всего было выпущено около 17 моделей разного уровня. Эти компьютеры использовали специализированные операционный системы: OS/360, OS/370, MVS (Multiple Virtual Storage) и ее вариации.

Новое поколение мейнфреймов System 390 появилось в начале 1990 г., но сохранило совместимость с предыдущими моделями. В процессе создания IBM 390 произошло обновление всей электронной базы – МП, ОП и ВЗУ на несколько поколений. В период с 1990 по 1999 г. было выпущено множество разнообразных систем под названием IBM S/390 Enterprise Server или ESA/390 (ESA – Enterprise System Architecture). Старшие две модели системы с названием «Summit» имели водяное охлаждение, младшие – воздушное.

В 2000 г. название «System 390» было заменено на «IBM e Server z Series». В октябре был представлен самый мощный на тот период компьютер массового коммерческого применения – z Series 900. Тогда же появилась новая 64-разрядная ОС – z/OS. В 2002 г. было представлено семейство z Series 800 для задач среднего уровня, 2003 г. был отмечен появлением новой модели z Series – e Server z Series 990, в 2004 г. появилась новая система среднего уровня z Series 890.

В середине 2005 г. семейство мейнфреймов пережило очередное переименование. С этого времени все системы этого класса обо-

значают System z9. Одновременно с этим объявлено о создании новой модели – 109. Эти модели отвечают современным жестким требованиям корпоративных систем по доступности и надежности. Возможность непрерывного профилактического обслуживания (без прерывания работы комплекса) вносит весомый вклад в то, что среднее время наработки на отказ системы z Series исчисляется десятилетиями, а предложенная IBM-система комплексирования – Parallel Sysplex (традиционные кластеры), характеризуется повышенной живучестью комплекса: его надежность сейчас 99,999 %, т.е. среднее время неработоспособности системы за год составляет не более 5 мин.

По мере того, как конфедициальность становится одним из главных требований бизнеса, поддержка SSL-транзакций (защищенные сетевые протоколы) превращается в определяющий фактор при выборе сервера. IBM z Series может работать с восемнадцатью криптографическими сопроцессорами, которые могут обрабатывать несколько тысяч защищенных транзакций в секунду без снижения скорости обработки.

В архитектурном плане мейнфреймы представляют собой многопроцессорные системы с одним или несколькими центральными и периферийными процессорами с общей памятью, связанными между собой высокоскоростными магистралями передачи данных, что оказалось неприемлемым для суперкомпьютеров из-за большого количества процессоров. При этом основная вычислительная нагрузка приходится на центральные процессоры, а периферийные процессоры (в терминологии IBM – селекторные, блок-мультиплексные, мультиплексные каналы и процессоры телеобработки) обеспечивают работу с широкой номенклатурой периферийных устройств.

Первоначально мейнфреймы были ориентированы на централизованную модель вычислений, работали под управлением патентованных операционных систем и имели ограниченные возможности для объединения в единую систему оборудования различных фирм-поставщиков. Однако повышенный интерес потребителей к открытым системам, построенным на базе международных стандартов и позволяющим достаточно эффективно использовать все преимущества такого подхода, заставил поставщиков мейнфреймов существенно расширить возможности операционных систем в направле-

нии совместимости. В настоящее время они демонстрируют свою «открытость», обеспечивая возможность использования протоколов межсоединений OSI и TCP/IP или предоставляя возможность работы на компьютерах под управлением операционной системы UNIX собственной разработки.

Стремительный рост производительности персональных компьютеров, рабочих станций и серверов создал тенденцию перехода с мейнфреймов на компьютеры менее дорогих классов – миникомпьютеры и многопроцессорные серверы. Эта тенденция получила название «разукрупнение» (downsizing). Однако этот процесс несколько замедлился. Основной причиной возвращения интереса к мейнфреймам эксперты считают сложность перехода к распределенной архитектуре клиент-сервер, оказавшейся выше, чем предполагалось. Кроме того, многие пользователи считают, что распределенная среда не обладает достаточной надежностью для наиболее ответственных приложений, которой обладают мейнфреймы.

Главным недостатком мейнфреймов в настоящее время остается относительно низкое соотношение производительность-стоимость. Однако фирмами-поставщиками мейнфреймов предпринимаются значительные усилия по улучшению этого показателя.

Следует также помнить, что в мире существует огромная инсталлированная база мейнфреймов, на которой работают десятки тысяч прикладных программных систем. Отказаться от годами наработанного программного обеспечения просто неразумно. Вновь создаваемые системы, с одной стороны, позволили модернизировать существующие системы, обеспечив сокращение эксплуатационных расходов, с другой – создали новую базу для наиболее ответственных приложений.

В связи с этим, несмотря на всевозможные (и постоянные) пророчества, предрекавшие неминуемую и скорую смерть этой платформы, в 2004 г. она успешно отметила свое 40-летие; при этом, по оценкам ведущих аналитических агентств мира, до 70 % критически важной корпоративной информации хранится и обрабатывается именно с помощью мейнфреймов.

Мини-компьютеры получили широкое распространение в 70 – 80-х годах. Популярность этих машин объяснялась их малыми размерами, стоимостью (относительно больших компьютеров) и универсальными возможностями. Производительность их была более

низкая, чем у больших систем, а длина слова равнялась 16 разрядам, однако параметры этих компьютеров вполне удовлетворяли большое количество пользователей. Наиболее популярные мини-компьютеры выпускали две фирмы США: Hewlett Packard (HP) и Digital Equipment Corporation (DEC), последняя выпустила серию компьютеров PDP 11 и VAX. Отечественные аналоги этих компьютеров были разработаны в рамках системы малых компьютеров (СМ, СМ 1, 2 (HP), СМ3, 4, 1420, 1300 и т.д.), однако к настоящему времени персональные компьютеры и рабочие станции по своим характеристикам превосходят большинство мини-компьютеров.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Опишите основные концепции архитектуры фон Неймана, ее составные части и их назначение. Назовите основные направления развития этой архитектуры. Существуют ли принципиально отличные виды архитектур в системах обработки данных?

2. Рассмотрите структуру запоминающих устройств современных компьютеров. В чем причины расслоения памяти компьютеров? Дайте определение понятия виртуальной памяти.

3. Опишите назначение, основные характеристики, разновидности схемотехнической реализации оперативной памяти компьютера. В чем достоинства и недостатки различных типов RAM, применяемых в современных компьютерах?

4. Каковы назначение и основные функции центрального процессора (ЦП)? Перечислите два уровня ЦП, форматы команд и данных. Дайте определение понятия микропрограммирования. Расскажите о многоуровневой организации ЦП, причинах ее расслоения и различных способах оценки быстродействия компьютеров.

5. Опишите различные способы распараллеливания процессов. Что такое векторная, конвейерная, векторно-конвейерная обработка данных? Назовите способы организации и ограничения в применимости этих методов.

6. Расскажите о назначениях, функциональных особенностях и разновидностях периферийных устройств компьютеров, различных формах представления информации, кодах, используемых в системах обработки данных.

ГЛАВА 3. МИКРОПРОЦЕССОРЫ, МИКРОКОМПЬЮТЕРЫ. ПЕРСОНАЛЬНЫЕ КОМПЬЮТЕРЫ, РАБОЧИЕ СТАНЦИИ, СЕРВЕРЫ И СУПЕРСЕРВЕРЫ

3.1. Микропроцессоры – эволюция, классификация, перспективы

Компьютерная революция, произошедшая во второй половине прошлого века, связана, прежде всего, с рядом знаковых достижений в электронной, а точнее в микроэлектронной, технике. Именно достижения электроники привели к фантастическому прогрессу в деле разработки технических средств информатики. В связи с этим необходимо хотя бы упомянуть об основных этапах развития электронной техники прошлого столетия – основах современных компьютеров.

Важнейшим событием в развитии электроники указанного периода является разработка транзистора. В 1951 г. Уильям Шокли, сотрудник Bell Lab., крупнейшего американского концерна по телеграфии и телефонии, продемонстрировал миру первый транзистор, представляющий трехслойный «сэндвич» толщиной около 1 см, заключенный в металлический корпус. Это совместная работа, в которой также участвовали сотрудники той же лаборатории Джон Бардин и Уолтер Браттейн, была вполне заслуженно оценена Нобелевской премией по физике в 1956 г. (кстати, Джон Бардин дважды лауреат Нобелевской премии – вторая за теплую сверхпроводимость).

Хотя изобретение транзистора было выдающимся достижением, оно не сразу завоевало достойное место из-за трудностей производства. На начальном этапе цена транзисторов была достаточно высокой (более чем в 10 раз превышала стоимость электронных ламп).

Вторым событием в сфере электроники, возможно менее глобальным, но также очень важным с технической точки зрения, было создание в 1958 г. первой интегральной схемы сотрудником Texas Instruments Джеком Килби и ее усовершенствование, позволившее начать их массовое производство Джином Херни и Робертом Нойсом. Инженеры окрестили эти устройства интегральными

микросхемами (ИМС), но чаще их называют чипами (Chip – щепка); это явилось началом эпохи миниатюризации в электронике.

Джек Килби также получил Нобелевскую премию, правда, уже в новом столетии за исследования в сфере нанотехнологии.

И, наконец, третьим важнейшим событием в развитии электроники, ориентированной на компьютерную технику, является разработка 34-летним сотрудником фирмы Intel (Integrated electronics) Эдвардом Хоффом в 1970 г. микропроцессора – интегральной схемы (Chip), реализующей функции центрального процессора компьютера. Этот процессор получил название Intel 4004, имел размер 3х4 мм и включал 2250 транзисторов (степень интеграции 2250). Эта работа велась в рамках разработки микрокалькулятора по заказу японцев.

Хотя изучают и разрабатывают микропроцессоры специалисты по электронной технике, ясное представление об этих устройствах необходимо иметь сегодня любому пользователю вычислительной техники по ряду причин. Именно микропроцессоры обеспечили появление микрокомпьютера, персональных компьютеров, рабочих станций, серверов и суперсерверов и привели к кардинальным изменениям в архитектуре суперкомпьютеров, о чем говорилось ранее. Для рядового пользователя возможности его ПК в значительной степени определяются типом используемого микропроцессора.

Микропроцессор (МП) – изготовленный при помощи технологии высокой степени интеграции прибор, который способен выполнять под действием программного управления функции центрального процессора компьютера. Появление микропроцессоров связано со стремлением конкурирующих фирм, изготовляющих электронную технику, создать более универсальную систему логических элементов на базе новых технологий. Области распространения МП значительно превзошли самые оптимистические предположения разработчиков. Уже в 1977 г. около 35 зарубежных фирм выпускали более 50-ти типов МП. За сравнительно небольшой промежуток времени в технологии производства МП был достигнут колоссальный прогресс. К концу 2000 г. степень интеграции МП превысила*

* Закон Мура. В 1965 г. Гордон Мур, один из основателей компании Intel, заметил закономерность: число транзисторов на единицу площади в компьютерных чипах удваивается каждые полтора года.

400 млн при количестве контактов около 500 и рассеиваемой мощности более 100 Вт, а в 2004 г. фирма IBM выпустила RISC-процессор POWER-5, имеющий 5370 контактов. Разрядность процессоров выросла с четырех до шестидесяти четырех. Считается, что разработчики электронной техники совершили экономическое чудо: если в начале 60-х годов транзистор стоил около 20 дол., то, если мысленно выделить такой элемент в 80-е годы из интегральной схемы, он уже стоит около 0,0000002 дол.

Можно выделить два направления применения МП:

1) МП используются в комплексе схемных элементов в виде микрокомпьютера, т.е. системы, собранной на одной или нескольких платах и содержащей МП, ОП и блоки (модули) ввода-вывода; имеется возможность применять эту систему самостоятельно совместно с обычными периферийными устройствами;

2) МП представляет семейство больших интегральных схем (БИС), встраиваемых как интегральное целое в создаваемую систему по усмотрению инженера-проектировщика.

До 1978 г. МП в основном применялись в информационно-измерительных и управляющих системах и, хотя интерес к ним был очень велик, тем не менее, они использовались относительно небольшим кругом высококвалифицированных специалистов по электронике и программированию. Однако появление в 1978 г. 16-разрядного МП фирмы Intel-8086 открыло новую эру в использовании МП – эру персональных компьютеров. Это была минимальная разрядность машинного слова для универсальных компьютеров, используемая в самых распространенных по тем временам (из наиболее доступных) мини-компьютерах.

МП можно классифицировать еще по ряду признаков.

По конструкции МП подразделяются:

на однокристальные, в которых, как правило, вся логика размещается в одном кристалле, они имеют постоянную разрядность и постоянный набор команд (по разрядности они могут быть 4-, 8-, 16-, 32- и 64-разрядными);

на процессорных элементах (секционированные или с наращиваемой разрядностью), в которых разрядность и система команд может изменяться и определяться в процессе разработки применительно к той прикладной области, где будет использоваться данный

МП (ввиду узкой специализации производительность этих систем может быть очень высокой).

Наибольшее распространение в настоящее время получили однокристальные МП.

Архитектура МП повторяет те наработки, которые применялись еще в суперкомпьютерах, и о которой по тем временам не было принято говорить в среде пользователей, далеких от этих проблем. Основные направления в развитии архитектуры следующие:

специализация процессоров, многофункциональная, поточная обработка (появление сопроцессоров, графических процессоров и т. п., встроенных в кристалл МП);

расслоение памяти и появление встроенной кэш-памяти;

параллельная обработка (как временное, так и пространственное распараллеливание);

конвейерная обработка (на уровне микрокода);

аппаратная реализация наиболее распространенных команд;

реализация возможности работы в многопроцессорных системах.

Можно выделить два основных типа архитектуры (постепенно сближающиеся), на базе которых разрабатываются два типа МП, имеющие различные направления использования: CISC и RISC.

CISC (Complex Instruction Set Computing) – архитектура с полным набором машинных команд; лидером в разработке таких МП является фирма Intel, хотя основоположником этой архитектуры можно считать фирму IBM с ее базовой архитектурой IBM 360, используемой с 1964 г.

RISC (Reduced Instruction Set Computing), или архитектура с упрощенным набором команд, используется для мощных рабочих станций, серверов и суперсерверов. Эта архитектура уходит корнями к компьютерам фирмы CDC при участии в разработке Сеймура Крэя. Применительно к микропроцессорам эти разработки связывают с научными центрами в Беркли и Стэнфорде, фирмой IBM и с именем Дэвида Паттерсона. Хотя следует отметить, что в разработанном в СССР суперкомпьютере «Эльбрус» использовались идеи RISC-архитектуры намного раньше.

В основе RISC-архитектуры лежит установленное разработчиками правило «80 – 20», которое гласит, что 80 % инструкций процессора (традиционной CISC-архитектуры) используется програм-

мами лишь 20 % времени, в то время как на 20 % относительно простых команд приходится львиная доля работы – 80 %. Таким образом, большая часть инструкций, занимая значительную часть кристалла, используется крайне неэффективно.

Основные принципы RISC-архитектуры следующие:

упрощенный и фиксированный набор команд с аппаратной реализацией выполнения вместо микропрограммной;

представление сложных команд в виде набора простейших, выполнение их (по возможности) за один цикл при использовании конвейерной обработки;

инструкции микропроцессора оперируют с данными, хранящимися только во внутренних регистрах, для загрузки данных из оперативной памяти и выгрузки результатов вычислений из регистров предусмотрены специальные команды, выполняемые отдельными функциональными блоками микропроцессора (т.е. нет потерь времени на поиск и загрузку данных и команд);

состав инструкций микропроцессора оптимизирован для языков высокого уровня (например, C).

Все это создает повышенную надежность, технологичность, экономичность и возможность повышения производительности (в 2 – 4 раза).

За последнее время подавляющее число RISC-процессоров стали *суперскалярными*, т.е. на нескольких конвейерах в каждом такте обрабатывается несколько инструкций и выдается несколько результатов. Задача распараллеливания решается на системном уровне, не затрагивая работу прикладных программистов. В одном из вариантов это решается на аппаратном уровне (Alpha-процессор фирмы DEC), и удается выполнить до четырех инструкций одновременно. Другой вариант связан с так называемой VLIW-архитектурой (Very Long Instruction Word), где работу по распараллеливанию осуществляет транслятор. При обработке такого типа три или более команды (на уровне трансляции) объединяются в одно слово. Затем выбранные команды обрабатываются одновременно. При этом порядок выполнения команд должен контролироваться самым тщательным образом (за этим следит транслятор). В процессорах такого типа может выполняться до десяти и более команд одновременно. Некоторым усовершенствованием этого подхода является EPIC-архитектура (Explicitly Parallel Instruction Computing – вычисления с

явным параллелизмом команд). Развитие этих двух типов архитектуры (CISC и RISC) привело к их сближению. Все современные CISC-процессоры имеют внутреннюю RISC-архитектуру. В то же время набор команд в некоторых RISC-процессорах может превышать количество команд в CISC.

МП очень часто классифицируют и по технологии, при этом развитие технологии увязывается с минимальными расстояниями между отдельными схемными элементами внутри кристалла, измеряемыми в долях микрометра (мкм). Уменьшение размеров формируемых на полупроводниковом кристалле элементов приводит к возрастанию производительности (за счет более высоких частот) и достижимой степени интеграции, а также снижению потребляемой мощности (из-за уменьшения напряжения питания) и стоимости (из-за уменьшения размеров кристалла на одной пластине формируется большее количество процессоров). В конце 90-х годов стандартной стала технология с 0,25 мкм, которая обеспечивает тактовые частоты 400 – 600 МГц, переход на технологию 0,18 мкм обеспечивает тактовые частоты от 600 – 800 МГц и выше. В 2001 – 2002 гг. переход на технологию 0,13 мкм обеспечит тактовые частоты выше 1 ГГц.

Лидером в разработке CISC-процессоров считается компания Intel со своей серией процессоров X 86, являющейся основой самых распространенных в мире ПК. К 2000 г. Intel выпустил шесть поколений процессоров этой серии. Приведем краткие характеристики этой серии без детальных характеристик моделей внутри серии.

С 2000 г. Intel перешел на новую архитектуру – процессоры седьмого поколения – Pentium 4 с начальной тактовой частотой 1,3 – 1,5 ГГц, которая к 2005 г. превысила 3 ГГц с техпроцессом менее 0,1 мкм, степень интеграции 42 млн.

Последние процессоры x86 архитектуры перестали быть CISC, но и RISC в полном понимании не стали (подобный переход не обеспечит сохранения обратной совместимости с существующим программным обеспечением), поэтому их можно классифицировать как псевдо RISC.

При разработке своих МП Intel большое значение придает мультимедийным приложениям. Последние версии процессоров пятого поколения (P5) обозначались как MMX (Multi Media Extention) и включали пятьдесят семь инструкций для выполнения операций

целочисленной арифметики в режиме SIMD с применением конвейерной обработки на уровне микрокода. Поскольку трехмерная графика требует быстрой арифметики с «плавающей точкой», в процессорах 6-го поколения (P6) были включены 70 инструкций в режиме SIMD для арифметики с «плавающей точкой», а процессоры 7-го поколения Pentium 4 (SSE2) уже имели 144 подобных инструкций. Названия этого направления в разработке МП также изменялись MMX → MMX2 → KNI → SSE1 → SSE2 и SSE3 и SSE4 (SSE-SIMD Stream Extention).

В состав последних процессоров включен генератор истинно случайных чисел, который очень важен в большом количестве приложений.

В связи с ростом производительности и внутренней частоты МП нарастает отставание частоты внешней шины процессора FSB (Front Side Bus), к которой подключается ОП и внешний кэш, в процессорах 7-го поколения предусмотрена размещенная в кристалле шина BSB (Back Side Bus), обеспечивающая работу встроенного в кристалле кэша первого уровня на внутренней частоте процессора. Перспектива развития технологии МП Intel по прогнозу фирмы на 2015 г. следующая:

достичь 0,01 мкм по техпроцессу;

достичь производительности процессора 1 трлн оп./с (производительность МП-P6 – 2 млрд оп./с);

обеспечить степень интеграции 2 млрд транзисторов на кристалл;

достичь тактовой частоты 30 ГГц.

На табл. 3.1 представлена динамика и перспективы изменения технологий наиболее распространенных МП фирмы Intel.

Конструктивно процессоры в зависимости от класса компьютера выпускаются в двух вариантах:

Slot – модульной конструкции с дискретными (внешними, не встроенными в кристалл) схемами (печатная плата с краевыми разъемами);

Socket, т.е. с интегрированной в кристалл кэш-памятью второго уровня (процессор, вставленный в соответствующий разъем на печатной плате).

Таблица 3.1

Эволюция развития МП x86 Intel

Год выпуска	1978	1982	1985	1989	1993	1996	1997	1997	1998	1999	2000	2004	2006	2006	2008	2013	2018
Тип МП	8086	80286	80386	80486	Pentium (P5)	Pentium MMX (P5)	Pentium Pro (P6)	Celeron (P6)	Pentium 2 (P6)P6)	Pentium 3 (P6)	Pentium 4 (P7)	Pentium 4 (P7)	Core 2 Duo	Core 2 Quad	Core i7		
Степень интеграции (млн транзисторов)	0,029	0,130	0,275	1,22	2,6	5,5	4,4	7,8	7,8	50	90	120	350	590	820		
Техпроцесс, нм	3000	1500	1000	500	500	350	350	250	250	180	130	90	65	65	45	35	22
Разрядность	16	16	32	32	32	32	32	32	32	32	32	32	64	64	64		
Эволюция частоты, МГц	4,7 – 8	10 – 25	16 – 40	33 – 133	60 – 233	60 – 233	150 – 200	400 – 600	400 – 600	600 – 800	1300	1500	> 3000	> 3000	> 3000		
Эволюция производительности относительно предыдущего поколения		1,6 – 2,5	1,5 – 2,5	2,4 – 4	2 – 3,5	1,3 – 4											

Все последние процессоры фирмы Intel имеют внутреннюю RISC-архитектуру. Что касается других фирм-производителей CISC-процессоров для ПК, то с определенного момента времени они отказались от собственных оригинальных разработок, не выдержав конкуренции с фирмой Intel. Такие фирмы, как AMD, Cyrix, UMC, Texas Instruments производят аналоги процессоров фирмы Intel, которые иногда превосходят оригиналы по определенным параметрам. Хотя в последнее время основная конкурентная борьба в этом классе процессоров развивается (с переменным успехом) между Intel и AMD. К 2009 г. МП фирма Intel имели значительно лучшие показатели. RISC-процессоры применяются, главным образом, для мощных рабочих станций, серверов и суперсерверов, а также для производства суперкомпьютеров. Производят эти процессоры фирмы – традиционные производители компьютеров: IBM, HP, Sun и др.

RISC-процессоры этих фирм применяются для серверов среднего и высшего уровней, а также для суперкомпьютеров. Среди особенностей этих МП следует отметить развитую иерархию кэш-памяти. В современных серверных системах количество уровней кэш-памяти может доходить до четырех, хотя наиболее часто используется трехуровневая схема. При этом все три уровня могут располагаться внутри кристалла (МП). Первый уровень кэш часто разделяется на кэш-команд (I-Cach) и кэш-данных (D-Cach). Вторая характерная особенность этих МП – способность работы в многопроцессорных системах различной топологии. Эти системы обеспечивают наиболее эффективное масштабирование по производительности, позволяют минимизировать дублирование вспомогательных систем машины (в отличие от *кластеров* – контроллеров ввода-вывода, дисковых массивов и т.п.), дают максимальную удельную производительность при пересчете на один ЦП. Все разрабатываемые в настоящее время специализированные процессоры имеют RISC-архитектуру. Это как процессоры, предназначенные для установки в мощные рабочие станции, сервера и суперсервера, так и миниатюрные сверхдешевые процессоры, предназначенные для работы в мобильных устройствах, различных электронных системах, окружающих нас повсеместно. Например, RISC-процессор A Ttiny 13 производства фирмы Atmel работает на частоте 20 МГц. Имеет всего восемь контактов и стоит меньше одного доллара.

Процессоры, устанавливаемые в суперкомпьютеры, производят традиционные производители серверов: IBM, HP, Sun и др.

Транспьютеры – необычный вид RISC-процессоров. Слово «транспьютер» – синтез двух слов: транзистор + компьютер. Транспьютер рассчитан на работу в мультипроцессорных системах с однотипными процессорами и аппаратной поддержкой вычислительных процессов. Особенностью транспьютеров является наличие коммуникационных, быстрых каналов связи (четырех на каждую схему), при этом каждый канал может одновременно передавать данные по одной магистрали в МП, а по другой – из него.

Однако в связи с разработкой МП RISC, способных работать в многопроцессорных системах, появлением многоядерных МП и кластерных систем актуальность в транспьютерах отпала.

Современное состояние и тенденции в развитии МП (в том числе и МП серии x86 см. табл. 3.1) определяются темпами развития технологии. Как указывают представители фирмы Intel (лидера производства МП), закон Мура будет действовать еще, как минимум, 10 лет. За это время (с начала 2008 г.) компания освоит технологические порты 32, 24, 15 и даже 10 нм, что позволит значительно повысить степень интеграции 1 млрд транзисторов на кристалл. Разговоры о том, что же делать с таким огромным количеством транзисторов в кристалле (т.е. об архитектуре МП), похоже, пришли к своему естественному логическому завершению. Будет продолжена тенденция, связанная с дальнейшим развитием функциональных возможностей МП, начавшаяся с появления первых МП фирмы Intel, устанавливаемых в ПК IBM.

Включение в состав МП аппаратно-реализованной арифметики (сопроцессоров), кэш-памяти (вначале первого уровня, а затем и последующих), контроллеров ОП шины BSB и т.п. завершилось созданием так называемых многоядерных процессоров. Практически это означает размещение в одном кристалле нескольких процессоров, имеющих ряд общих функциональных блоков с дальнейшим переходом на полноценные многопроцессорные системы (однокристальные). По-видимому, с этим связано изменение в названии новой серии процессоров Intel в 2006 г. (следующих за Pentium-4) Core (ядро). 2007 г. ознаменовался тем, что одноядерные МП серии x86 ушли в прошлое. Один из последних (представленных фирмой в 2008 г.) МП Core Extreme QX9770 (модели высшего

ценового звена 3,2 ГГц, технология 45 нм) имеет четыре ядра, а точнее пару двуядерных процессора.

Однако дальнейшее и значительное увеличение количества процессорных ядер, которое планируется рядом фирм (особенно производителями серверов), по-видимому, приведет к серьезным проблемам у разработчиков программного обеспечения. Если современные ОС преодолели параллельную обработку для небольшого количества процессорных ядер, однако их значительное увеличение создает серьезные трудности в этой области.

Лидером во внедрении подобных систем являются разработчики МП для серверов. Так, фирма Sun Microsystem для своего процессора последнего поколения Rock, включающего четыре блока по четыре процессорных ядра в каждом, рапортовала о запуске своей десятой версии ОС Solaris 10, адаптированной под Rock. Массовое внедрение подобных разработок для обычных ПК как всегда произойдет в будущем и потребует значительных усилий разработчиков программного обеспечения ПК (ПОПК).

3.2. Персональные компьютеры. Краткая история, функциональные возможности, номенклатура

Персональный компьютер (ПК) – небольшой компьютер, основой которого служит МП, т.е. микрокомпьютер. Однако не все микрокомпьютеры являются персональными компьютерами. Микрокомпьютер может быть ориентирован на решение одной задачи, например на управление станком или регулирование подачи топлива в автомобильный двигатель и т.п. Персональный компьютер, действуя как самостоятельная вычислительная машина, предоставляет в распоряжение индивидуального пользователя самые разнообразные возможности. ПК должен обладать низкой стоимостью (доступной для индивидуального пользователя), малыми размерами, малой потребляемой мощностью; иметь стандартную периферию – монитор, клавиатуру, НМД, печать и т.п., операционную систему, облегчающую взаимодействие пользователя с системой; обладать универсальностью, что дает возможность выполнять обширный набор программ для различных приложений и т.п.

Толчком к развитию персональных компьютеров послужило, прежде всего, создание микропроцессора. При этом динамика раз-

вития персональных компьютеров связана с развитием микропроцессоров.

Можно выделить три основных этапа в развитии персональных компьютеров.

Первый характеризуется как этап «самодеятельности», неформального любительского стиля работы, когда в 1974 – 1975 гг. Эд Робертс, основатель фирмы MITS, разработал первый микрокомпьютер, устройство со столь гибкой конструкцией, что его вполне можно было считать первым серийным ПК. Он получил название Altair 8800 и находил спрос, в основном, у любителей знакомых с электроникой и проявляющих интерес к устройству компьютера. Им самим представлялась возможность собирать сложные машины из отдельных конструктивных узлов. Altair использовал восьмиразрядный МП Intel 8080, а в качестве периферии применялись бытовые магнитофоны, телевизионные мониторы и т.п. Эти устройства имели определенный коммерческий успех и высокий потребительский спрос у энтузиастов вычислительной техники и программирования.

Одновременно с этим Пол Аллен, молодой программист из Бостона, в содружестве со студентом Гарвардского университета Биллом (Уильямом) Гейтсом написали программу, реализующую для Altair популярный язык Бейсик. Таким образом, владельцы компьютера получили очень удобный язык, значительно облегчающий составление программ.

Второй этап обычно связывают с основателями фирмы ЭПЛ (Apple) Стефеном Возняком и Стивеном Джобсом в 1977 г., которая в дальнейшем превратилась в корпорацию, вышедшую в мир большого бизнеса и явилась основателем всей отрасли производства персональных компьютеров. Начался выпуск систем блочно-модульной конструкции, которые отличались простотой эксплуатации и были рассчитаны на потребителей, не имеющих никакой подготовки в области вычислительной техники. Таким образом, была предпринята попытка внедрить компьютеры в деловую практику и домашний обиход.

И, наконец, третий этап начался в 1981 г., когда корпорация IBM, крупнейший в мире изготовитель оборудования для обработки данных (доминирующая на рынке больших компьютеров), занялась производством и сбытом персональных компьютеров и быст-

ро преуспела в своем начинании. Благодаря удачному конструктивному решению персональный компьютер IBM PC очень быстро занял доминирующее положение на рынке персональных компьютеров. В основе конструкции IBM PC заложена концепция «открытой архитектуры» и использования микропроцессоров фирмы Intel.

Достоинство этого решения связано со следующим:

возможностью расширения конструкции компьютера и его модернизации (UPGRADE) без привлечения квалифицированных специалистов и представителей фирмы;

аппаратной и программной совместимостью – все вновь появляющиеся модели и отдельные устройства совместимы сверху-вниз;

возможностью изготовления отдельных устройств и сборкой компьютеров различными фирмами-изготовителями и т.п.

Именно благодаря этим достоинствам началось лавинообразное нарастание производства ПК, совместимых с оригинальной моделью IBM, что способствовало снижению их стоимости. Это семейство компьютеров получило название клона IBM.

Впоследствии IBM довольно быстро утратила монополию на контроль рынка ПК. Именно этим объясняется разработка IBM нового семейства: машины IBM PS/1, PS/2, PS/VP и т.д. (PS-Personal System). В этом новом семействе IBM отказывается от принципа открытой архитектуры, компьютеры абсолютно не совместимы на аппаратном (конструктивном) уровне с предыдущими моделями IBM PC, хотя и сохраняют программную совместимость. Однако эта система не имела такой популярности, а девять ведущих американских фирм, объединившись, продолжили линию IBM PC. IBM предприняла очередную попытку обойти конкурентов на рынке персональных компьютеров, объединившись с фирмами Apple и Motorola. Совместными усилиями был разработан новый микропроцессор POWER, однокристальный вариант которого POWER PC используется для персональных компьютеров, разрабатываемых этими фирмами. И хотя по многим параметрам эти ПК значительно превосходят IBM PC, преодолеть монополию IBM PC пока не удается. Поскольку семейство ПК IBM PC – самые распространенные в мире настольные и переносные компьютеры общего пользования, полезно рассмотреть динамику их развития и современное состояние.

Первая модель ПК этого семейства IBM PC имела ОП 64 Кбайт, шину расширения – ISA, накопитель на гибких дисках – 360 Кбайт (1981 г.).

Следующий класс ПК – IBM PC/XT (Extended Technology) – имел ОП 1 Мбайт, винчестерские диски, процессор 8088 (1983 г.).

В 1984 г. появились ПК IBM PC/AT (Advanced Technology) с процессором 80286/287, шиной ISA, адресуемой памятью 16 Мбайт и набором стандартной периферии.

Современные тенденции развития этого класса ПК очень сильно связаны с политикой фирмы Intel, которая пытается исключить из рынка производителей МП конкурентов, разрабатывающих аналогичные процессоры. Развитие ПК IBM-клона вступило в этап сегментации, которая определятся спецификациями. Так, спецификация 1999 г., PC99, предусматривала разделение компьютеров по сферам применения на три категории: нижний уровень – базовый офисный или потребительский (Basic Office & Consumer PC), для досуга и развлечений (Entertaiment PC) и рабочие станции (Workstation PC). Многие новые продукты начинают выпускаться сразу в нескольких вариантах, хотя и имеющих единую основу, но ориентированную на разные сегменты рынка.

Для процессоров сегментация проявилась наиболее отчетливо. Фирма Intel выделила три класса процессоров – Celeron для компьютеров базового уровня, Pentium для компьютеров производительного уровня (досуг и развлечения) и Xeon для мощных рабочих станций и серверов. Все они созданы на единой основе[*], но отличаются типом разъема – Socket 370 Slot1, Slot2 и т.п., что ограничивает миграцию между сегментами.

При этом существующая тенденция, когда с появлением нового процессора ПК предшествующий тип МП опускается на более низкий уровень приложений, затрудняется.

3.3. Архитектура ПК, системные и локальные шины, состав периферийных устройств

Архитектура ПК определятся системой шин, с помощью которых ЦП связан с ОП и периферийными устройствами.

[*] За исключением Xeon.

Современная архитектура ПК является результатом довольно длительного развития, необходимость которого связана, прежде всего, с эволюцией МП и расширением спектра периферийных устройств, а также повышением их быстродействия.

Чтобы иметь ясное представление о современной архитектуре ПК, полезно посмотреть на динамику ее развития.

На начальном этапе в архитектуре ПК использовалась единая шина (эта конструкция была использована фирмой DEC еще в 1960 – 1970 гг. прошлого столетия для своих мини-компьютеров начального уровня).

ПК изначально также строились на основе одной системной шины ISA (Industry Standard Architecture), EISA (Extended ISA) или MCA (Micro Channel Architecture). Необходимость сохранения баланса производительности по мере роста быстродействия микропроцессоров привела к двухуровневой организации шин ПК, а затем и к трехуровневой (которая на начальном этапе применялась только в рабочих станциях и серверах). Динамика развития архитектуры отражена на рис. 3.1, 3.2 и 3.3, где представлена организация компьютеров различной производительности. В табл. 3.2 даны характеристики шин, применяемых ПК с начала их производства и до наших дней.

ISA – промышленный стандарт, для архитектуры первых моделей ПК (PC, PC/XT, PC/AT 286,386), характеризуется относительно невысокими скоростями обмена информации по шине и невысокой стоимостью, эта архитектура совместима с фирменными шинами IBM – XT-bus и AT-bus.

EISA – расширенный промышленный стандарт, предложенный для реализации большого объема адресуемой ОП и ускорения вычислительных и обменных операций ПК с микропроцессорами 80386, 80486, является значительно более дорогостоящим расширением шины ISA, поэтому при многошинной организации, как правило, не применяется.

MCA – стандарт фирмы IBM, разработанный для моделей семейства PS/2. Характеризуется значительным ускорением обмена данными между отдельными устройствами ПК (особенно с ОП), однако эта архитектура совершенно не совместима с ISA и EISA.

VL-bus – локальная шина, предложенная ассоциацией VESA (Video Electronics Standard Association). Предназначена для увели-

чения быстродействия видеоадаптеров и контроллеров дисковых накопителей (*локальной* называется шина, электрически выходящая непосредственно на контакты микропроцессора). Шина ориентирована на процессор 80486. Следующая спецификация этой шины для процессоров Pentium с ожидавшейся скоростью 400 Мбайт/с дальнейшего распространения не получила.

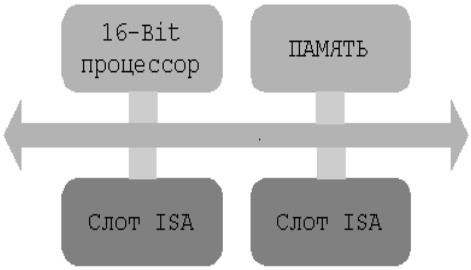


Рис. 3.1. ISA: все компоненты системы на одной шине

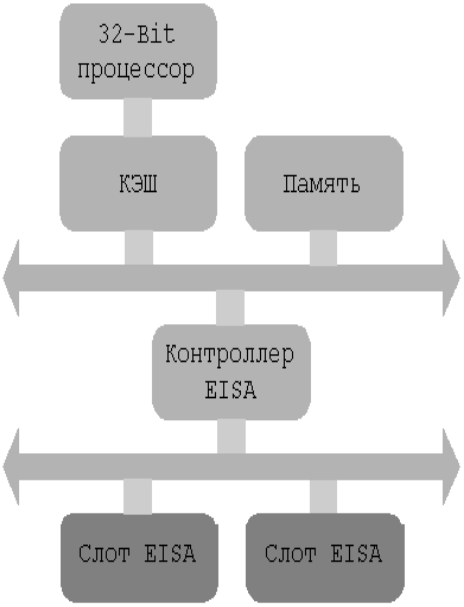


Рис. 3.2. Compaq Desk Pro 386/20: архитектура EISA

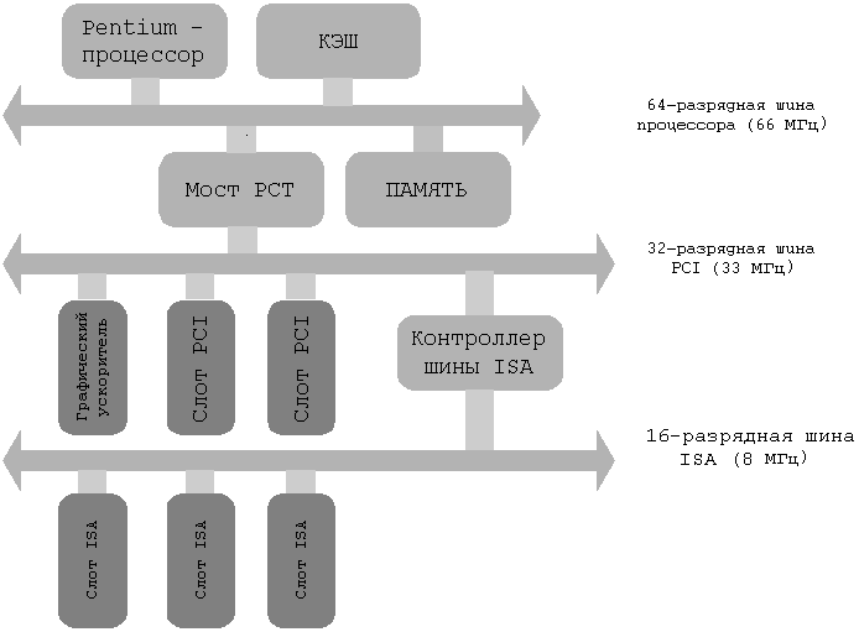


Рис. 3.3. HP Vectra XP: архитектура PCI

Таблица 3.2

Характеристики шин, применяемых в ПК (данные до 2000 г., быстродействие VSA и PCI указано пиковое, в скобках реальное)

Характеристика/Тип	XT-bus	AT-bus	MCA	ISA	EISA	VLB	PCI
Разработчик	IBM	IBM	IBM	Промышленный стандарт	Промышленный стандарт	VESA	INTEL
Разрядность шины адреса	20	24	24/32, 32	20	32	32	32 (64)
Разрядность шины данных	8	8,16	8, 16, 24, 32	8, 16, 32	8,16	32 (64)	32 (64)
Быстродействие, Мбайт/с	5	10	20 – 25	5 – 10	33	130 (80)	133 (60)

Шина PCI (Peripheral Component Interconnect) так же, как и VL-bus, поддерживает канал передачи между ЦП и периферийными устройст-

вами. При работе с процессором 80486 она имеет примерно те же показатели, что и VL-bus. В отличие от последней шина независима от процессора, подключается к нему через специальный адаптер (поэтому название локальной применительно к PCI не совсем точно). Эта шина получила очень широкое распространение и применяется не только в ПК, но и в рабочих станциях, и серверах.

Постоянное увеличение тактовой частоты МП, его способность обрабатывать потоки данных до гигабайт в секунду, привело к появлению самой быстрой внешней шины процессора FSB (Front Side Bus), к которой подключились самые быстродействующие устройства (внешний кэш, ОП, AGP – Accelerated Graphic Port).

Попыткой сбалансировать быстродействие ЦП и ОП является двойная разрядность этой шины (64 разряда) и организация двухканальной ОП. И, наконец, полная согласованность по частоте процессора и внутренней кэш первого уровня была достигнута благодаря размещению еще одной шины (BSB – Back Side Bus) внутри кристалла (об этом уже говорилось).

Конструктивно архитектура ПК (системный интерфейс) реализуется на системной плате и связана с понятием «чипсет» (Chip set) – набором интегральных схем, формирующих блок вычислительной системы, реализующий все необходимые связи основных компонентов – процессора, памяти и шин расширения.

Два варианта архитектуры ПК для МП фирмы Intel (в современной терминологии – дизайн) представлены на рис. 3.4.

Значительное развитие за истекший период получили также интерфейсы периферийного оборудования. Большую популярность приобрел интерфейс ввода-вывода SCSI (Small Computer System Interface – интерфейс малых вычислительных систем), который, как правило, применяется в высокопроизводительных рабочих станциях и серверах. Этот интерфейс располагается между системной шиной компьютера и периферийными устройствами. Предназначен, прежде всего, для подключения дисковых накопителей, но может использоваться и для подключения других устройств, например сканеров, цифровых осциллографов и т.д. Интерфейс постепенно развивался, увеличивая пропускную способность. Существует множество его разновидностей: SCSI-1, SCSI-2, Ultra 160, Ultra 320 и т.д., некоторые из которых несовместимы друг с другом. Другой широко распространенный интерфейс для подключе-

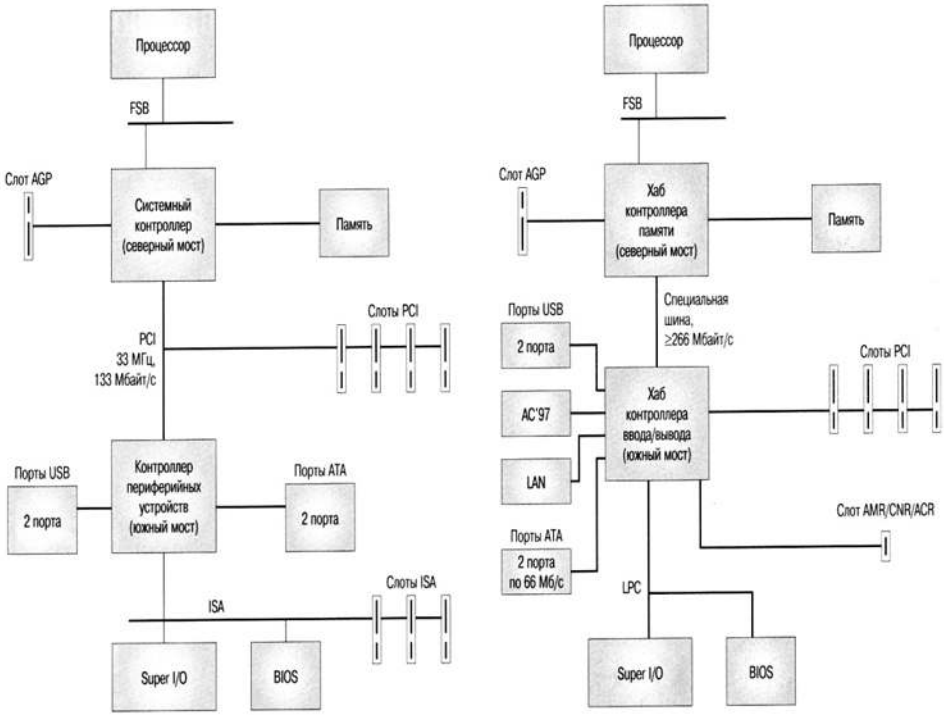
ния периферийных устройств, использующийся, как правило, в малых системах, – IDE (Integrated Drive Electronics – электроника, встроенная в диск). Такое название он получил из-за того, что, в отличие от ранних версий SCSI, контроллер управления головками и другими механическими частями был расположен не на системной плате компьютера, а на самом диске. В дальнейшем тот же интерфейс стали называть ATA (Advanced Technology Attachment – присоединение по передовой технологии).

И SCSI, и ATA – параллельные интерфейсы, т.е. их шина данных состоит из множества проводников (например, 8 или 16) и плюс к ним еще управляющие проводники. Разъемы и кабели данных для параллельных интерфейсов весьма громоздки, и, что самое главное, при выходе из строя хотя бы одного контакта вся система перестает работать. В данный момент идет общая тенденция к переходу на последовательные интерфейсы. В них данные передаются с гораздо большей частотой, но зато для передачи данных может использоваться всего два или четыре контакта. Современный последовательный потомок интерфейса SCSI – SAS (Serial Attached SCSI), потомок интерфейса ATA – SATA (Serial ATA). С появлением интерфейса SATA, обычный ATA, для того чтобы не путаться, стали называть PATA (Parallel ATA).

В связи с широким распространением мультимедийных приложений и значительным увеличением потоков данных разработан новый стандарт подключения графических адаптеров – AGP (Accelerated Graphics Port), получивший широкое распространение, так как значительно ускоряет обмен с видеоадаптером, не перегружая системную шину PCI.

Современные интерфейсы периферийного оборудования (контроллеры) представлены в табл. 3.3.

Для подключения внешних периферийных устройств (модемы, принтеры, мышь, внешние НМВ и др.) использовались интерфейсы COM (от слова Communication – связь) и LPT (Line Print Terminal – терминал линейной печати). В настоящий момент они заменены более современным последовательным интерфейсом USB (Universal Serial Bus – универсальная последовательная шина).



а *б*

Рис. 3.4. Классический (*а*) и хабовый (*б*) дизайн системных плат (2006 г.)

Таблица 3.3

Интегрированные контроллеры

Контроллер	Краткая характеристика	Функции	Примечание
PCI	32-разрядная 33-МГц шина расширения	Подключение плат расширения	В некоторых чипсетах встраивается в северный мост. Более производительные решения – PCI64, PCI-X – реализуются обычно на отдельных контроллерах, подключаемых к северному мосту
ISA	8- или 16-разрядная шина расширения	Подключение плат расширения	Относится к устаревшим. При необходимости реализуется на отдельном контроллере, подключаемом через LPC (Low Pin Cjnnector)
Super I/O	Контроллеры параллельного, последовательного портов, PS/2 мыши, PS/2 клавиатуры, инфракрасного порта, флоппи-дисковода и др.		Обычно реализуются в отдельном чипе, подключаемом через LPC. Относится к устаревшим
ATA (IDE)	Параллельный 16-разрядный порт: ATA66 – 66 Мбайт/с, ATA100 – 100 Мбайт/с, ATA133 – 133 Мбайт/с	Подключение жестких дисков и других устройств хранения данных	В большинстве моделей поддерживается ATA100. В некоторых ATA133
Serial ATA	Последовательный порт, 1500 Мбит/с (150 Мбайт/с)	Для подключения жестких дисков	Реализован в 2002 г.
USB 1.1	Последовательный порт, 12 Мбит/с	Подключение принтеров, сканеров, указательных устройств, устройств хранения данных и др. Должен заменить устаревшие параллельный и последовательные порты	

Окончание табл. 3.3

Контроллер	Краткая характеристика	Функции	Примечание
USB 2.0	Последовательный порт, 480 Мбит/с	Подключение скоростной периферии (сканеров, устройств хранения данных и др.)	Интегрированные варианты только начинают появляться
EEE 1394	Последовательный порт, 400 Мбит/с	Подключение скоростных периферийных устройств (сканеров, устройств хранения данных и др.) и аудио/видеоустройств	Интегрированные варианты только начинают появляться
AC'97	Цифровая часть звукового контроллера AC'97	Реализация недорогих программных звуковых плат или модемов	Требуются дополнительные компоненты (кодеки и др.)
LAN (Ethernet)	Последовательный порт, 10-1000 Мбит/с	Построение локальных сетей	Реализуется не во всех моделях. Иногда требует дополнительных компонентов. Нынешние решения – 10-100 Мбит/с
Мониторинг	Контролирует температуру, напряжения питания, скорость вращения вентиляторов и некоторые другие параметры		Реализуется не во всех моделях

Создание ПК привело к значительному прогрессу в разработке периферийных устройств компьютеров. Особенно эта тенденция просматривается на примере ВЗУ. При значительно меньших размерах они обладают значительно большей емкостью и быстродействием, чем устройства совсем недавнего прошлого. Спектр этих устройств также значительно расширился. В табл. 3.4 приведены основные типы ВЗУ, применяемые в ПК.

Устройства на магнитной ленте (компакт-кассетах) уже к 2000 г. достигли емкости в десятки гигабайт (тогда как первые такие накопители имели емкости немногим больше 100 Кбайт) и применяются они для резервного копирования и как архивы длительного хранения.

Таблица 3.4

Характеристики ВЗУ

Тип ВЗУ	Стримеры	Жесткие диски	Флоппи-диски		Оптические диски		Магнито-оптика
Обозначение	STD, SDX, STS, TS2	НЖМД (HDD), JAZ	НГМД (FDD, HIFO)	ZIP	CD-ROM, CD-R (worm), CD-RW (warm)	DVD-ROM, DVD-R (worm), DVD-RW (warm)	MO, MOD
Объем	4-25 Гбайт	6-36 Гбайт	1,44 Мбайт (100-200 Мбайт)	100-250 Мбайт	650 Мбайт	4,7-17 Гбайт	9,1 Гбайт
Время доступа	–	7 – 10 мс	100 мс	28 мс	70 – 300 мс	80 – 300 мс	17 – 19 мс

Жесткие диски по-прежнему составляют основу системных ВЗУ, хотя их емкость также значительно увеличилась (от 5 – 10 Мбайт в первых системах до нескольких десятков гигабайт в 2000 г.). В настоящее время объем РЖМД в ПК превышает несколько терабайт. Самыми «долгоживущими» среди ВЗУ являются флоппи-диски, представляющие личные архивы пользователей. До последнего времени используются дискеты на 1,44 Мбайт, разработанные фирмой Sony еще в 1980 г. Попытки заменить их аналогичными дисками емкостью свыше 100 Мбайт имели незначительный успех.

Широкое распространение получили компактные устройства магнитооптики, имеющие довольно малое время доступа. Повсеместно в практику рядовых пользователей вошли оптические ВЗУ, развивающиеся в сторону WARM (Write And Read Many times). С переходом на новый формат DVD их емкость значительно увеличилась, о чем говорилось ранее.

3.4. Рабочие станции, серверы и суперсерверы

Создание RISC-процессоров и микросхем памяти с большой емкостью привело к окончательному оформлению настольных систем

высокой производительности, которые сегодня известны как рабочие станции. Первоначальная ориентация рабочих станций на профессиональных пользователей (в отличие от ПК, которые вначале ориентировались на самого широкого потребителя-непрофессионала) привела к тому, что сейчас рабочие станции – хорошо сбалансированные системы, в которых высокое быстродействие сочетается с большим объемом оперативной и внешней памяти, высокопроизводительными внутренними магистралями, высококачественной и быстродействующей графической подсистемой и разнообразными устройствами ввода-вывода. Это свойство, хорошая сбалансированность, выгодно отличает рабочие станции среднего и высокого классов от ПК.

В литературе существует неоднозначность в определении термина «рабочая станция». Совсем недавно под рабочей станцией понимался любой терминал (в том числе и персональный компьютер), работающий в вычислительной сети и с помощью которого пользователь может использовать ресурсы вычислительной техники, объединенные в сеть.

В современном представлении принадлежность к сети не является единственным признаком рабочей станции. Сама станция – особый вид компьютера, имеющий очень много общего с ПК, в то же время существенно отличающийся от них. Основная сфера применения рабочих станций – области, требующие большого количества вычислений, и управление высококачественной графикой. Это, прежде всего, системы автоматизированного проектирования (САПР), автоматизированные системы научных исследований (АСНИ), а также системы автоматизации разработки программного обеспечения.

Если ПК очень эффективны для использования в различных экономических приложениях (бухгалтерии, торгово-промышленных фирм и других аналогичных организаций), а также как бытовые компьютеры, то применение их для указанных выше приложений (САПР, АСНИ) малоэффективно, поскольку рабочие станции имеют значительно большую мощность, а также периферию с повышенными техническими характеристиками.

Тем не менее, быстрый рост производительности ПК на базе новейших МП Intel в сочетании с резким снижением цен на эти изделия и развитием технологии локальных шин, позволяющей устра-

нить многие «узкие места» в структуре ПК, делают современные ПК весьма привлекательной альтернативой рабочим станциям. В свою очередь производители рабочих станций создали изделия так называемого «начального уровня», по стоимостным характеристикам близкие к производительным ПК, но все еще сохраняющие лидерство по производительности и возможностям наращивания. Все это привело к понятию «персональная рабочая станция», которое объединяет эти два направления.

Пять фирм США производят около 90 % рабочих станций. До середины 90-х годов прошлого столетия (и тысячелетия) рынок рабочих станций (которые активно используются и в настоящее время) выглядел следующим образом:

1) Sun Microsystem, на базе разработанного этой фирмой 64-разрядного RISC МП SPARC (Scalable Processor Architecture) выпускала рабочие станции с аналогичным наименованием. Разработанный этой фирмой МП Ultra SPARC III (1997 – 1998 гг.) имел тактовую частоту 600 МГц и быстродействие 600 – 700 MIPS (32-разрядный МП Pentium 66 выпуска этого года имел быстродействие 25 – 100 MIPS). Sun занимала около 30 % мирового рынка рабочих станций;

2) Hewlett – Packard (HP), на базе 64-разрядного МП RISC собственной разработки (начальная серия PA 8000, с тактовой частотой 500 ÷ 560 МГц) выпускала рабочие станции серии HP 8000 (9000). Следует отметить, что эта фирма и фирма Intel разрабатывают МП PA-RISC (IA-64), совместимый с процессорами Intel. HP занимает 22,6 % мирового рынка рабочих станций;

3) IBM – на базе 64-разрядного RISC МП POWER (Performance Optimization With Enhanced Risc) выпускает рабочие станции RS/6000 (на основе которых эта же фирма производит суперкомпьютеры PS). Начальные варианты этого процессора POWER II оформлялись в виде восьми микросхем, объединяющих порядка 23 млн транзисторов на площадке 1217 мм^2. Эволюция этих процессоров в направлении малых систем привела к однокристальному процессору POWER PC. Этот же процессор использует фирма Apple для своих персональных рабочих станций. IBM занимает 16,1 % рынка рабочих станций;

4) разработанные фирмой DEC Alpha-станции, рабочие станции на основе 64-разрядного процессора RISC МП Alpha. МП Alpha –

первый на рынке однокристальных 64-разрядных процессоров и до 90-х годов самый мощный процессор. Производительность процессора 21164 (1994 г.) с тактовой частотой 300 МГц оценивалась как 1200 MIPS. Эти процессоры фирма Cray Research Inc использовала в своих суперкомпьютерах. Слабое место этого МП – соотношение цена-производительность. Эти рабочие станции занимают 10,9 % мирового рынка;

5) Silicon Graphics Computer System (SGCS) – ведущий производитель рабочих станций для геоинформационных систем, использует 64-разрядные RISC МП фирмы MIPS, один из процессоров которой имеет производительность около 800 MIPS. До 90-х годов занимала 10,2 % мирового рынка рабочих станций.

Таким образом, практически до 2000 г. рынок был поделен между компьютерной техникой на основе процессоров Intelx86 (и их аналогов) и производительными, но дорогими рабочими станциями на различных RISC-архитектурах.

С ростом популярности высокопроизводительных процессоров на основе архитектуры Intelx86 параллельно со снижением их цены сложилась ситуация, в которой доля рынка на основе «настоящих» RISC-архитектур сократилась до нерентабельного уровня, что привело к окончательному смещению приоритетов разработчиков этих архитектур в сторону серверов и суперкомпьютеров. Многие из перечисленных архитектур остановились в своем развитии и их исчезновение – дело времени. Так, прекратилась разработка DEC Alpha, MIPS и фактически HP PARISC. IBM POWER позиционирует на рынок дорогих высокопроизводительных серверов. Наибольшие перспективы широкого использования имеет МП Itanium, совместной разработки Intel и HP.

Сервер – компьютер, обеспечивающий обслуживание пользователей сети: разделяемый доступ к процессорам, дискам, файлам, принтеру, системе электронной почты и т.д. Этот класс компьютеров имеет ряд специфических особенностей. Прежде всего, это связано с необходимостью обеспечения высокоэффективного многопользовательского и многозадачного режима: высокая производительность, надежность, дублирование, резервирование дисков, их повышенная емкость и т.д. В то же время требования к мониторам невысоки, так как эти компьютеры, как правило, не используются в качестве персональных станций, а являются компьютерами коллективного поль-

зования, регламентирующими работу вычислительных сетей и предоставляющие свой ресурс конечным пользователям. Серверы и рабочие станции производились одними и теми же фирмами. Однако и в этом разделе вычислительной техники, в связи с указанными ранее причинами, произошли серьезные изменения.

Стремительное увеличение ресурсов массовых ПК, появление 64-разрядных процессоров для них (AMD-Athlon, Operton и Intel-Xeon) привели не только к постепенному вытеснению с рынка рабочих станций традиционных МП RISC-архитектур, перечисленных ранее, но и к появлению серверов так называемого младшего уровня, ориентированных на эти МП, 2- – 4-процессорные модели. Серверы младшего уровня присутствуют в модельных рядах всех крупнейших компаний, производителей серверных решений Fujitsu-Siemens, Dell, HP, IBM, Sun. Поставляемая этими фирмами продукция содержит узлы, изготавливаемые, как правило, самими фирмами самостоятельно. Огромный опыт проектирования серверных решений и фирменные технологии позволяют этим компаниям создать гораздо более совершенные устройства в сравнении с моделями на базе готовых платформ, однако они заметно дороже.

Так, фирма HP под торговой маркой Proliant выпустила к 2005 г. 10 млн 2- – 4-процессорных серверов на базе МП как Intel, так и AMD.

Фирма IBM поставляет серверы нижнего уровня под маркой Server x Series. Однопроцессорные серверы используют МП Pentium 4, двух- и четырехпроцессорные модели используют МП Xeon. Sun поставляет подобные серверы под маркой Sun Fire на базе МП AMD Operton – двух- и четырехпроцессорные модели, иногда применяются для таких серверов и фирменные МП Sun.

Весьма перспективным считается направление разработки серверов на базе готовых платформ. *Платформой* принято называть сочетание корпуса и системной платы (иногда к ним добавляется блок питания, шасси для дисков и другие элементы). Использование таких платформ имеет огромные преимущества как с технической, так и с маркетинговой точки зрения. Использование стандартных платформ делает создание сервера столь же простым, как и сборка обычного ПК, что позволяет даже небольшим компаниям пополнить модельный ряд серверов младшего уровня. Изготовители платформ имеют большую свободу при проектировании, могут

принимать нестандартные компоновочные решения. Очевидны и маркетинговые преимущества – цена платформы будет ниже, чем стоимость купленных по отдельности системной платы и корпуса. Именно по этому пути идут многие российские компании – их модельные ряды фактически представляют собой аналог продуктов модельного ряда того или иного изготовителя платформ, выпущенных на рынок под собственной торговой маркой. Среди изготовителей платформ фигурируют такие известные фирмы, как ASUSTeK Computer Inc., Intel, MSU.

Успешное развитие серверов младшего уровня свидетельствует о том, что, несмотря на то, что уровень технического прогресса определяют самые современные и высокопроизводительные устройства, к которым приковано всеобщее внимание, успехи рынка зависят вовсе не от них – им принадлежит лишь незначительная доля рынка, основную же прибыль компании-изготовители получают от массовых устройств. Именно с этим связано то, что крупнейшие изготовители высокопроизводительных, «тяжелых», UNIX-серверов включились в производство серверов младшего уровня. Это вовсе не означает, что эти компании прекратили дальнейшее совершенствование своих традиционных высокоуровневых серверов. К этим компаниям относятся, прежде всего, HP, IBM, и Sun, у каждой из которых имеются свои подходы при реализации высокопроизводительных устройств. Общим является использование «классических» RISC-процессоров собственной разработки, изначально предназначенных для однопроцессорных рабочих станций. Постоянное совершенствование этих процессоров – переход от 32- на 64-разрядные, иерархия кэш-памяти и интеграция ее в процессоре, появление двухъядерных и многоядерных процессоров.

Особенности разработки высокопроизводительных серверов перечисленными выше фирмами, заключаются в следующем.

HP. В 2005 – 2006 гг. HP вступила в завершающую стадию перевода «тяжелых» серверов со своей традиционной архитектуры PA-RISC на архитектуру IA 64 (Itanium 2 – по заказу фирмы Intel). Серверы на процессорах Alpha заканчивают свое существование. Это свидетельствует о том, что даже такой мощной фирме, как Hewlett-Packard, сложно поддерживать одновременно два процессора. Хотя пока компания одновременно с новыми Itanium-серверами Integrity продолжает выпускать свои традиционные модели HP9000. Тради-

ционные серверы разделяются на серверы начального уровня, включающие от четырех до восьми процессоров PA8900, оперативную память от 32 до 128 Гбайт, имеющие симметричную многопроцессорную архитектуру (SMP), серверы среднего уровня – 16- – 32-процессорные серверы на процессорах PA8900; суперкомпьютеры HP9000 Superdome с количеством процессоров до 128 и оперативной памятью 1 Тбайт. Серверы HP на базе процессора Itanium, которые обозначаются как HP Integrity, имеют такие же градации и примерно одинаковые с описанными выше традиционными HP-серверами характеристики.

Платформа IBM POWER. Платформа развивается с 1990 г., когда было разработано семейство высокопроизводительных рабочих станций RS6000, ныне известных как e Server p Series. За прошедшее время прошло шесть поколений MP POWER (от POWER1 до POWER6). В последних процессорах произошла интеграция функций, для исполнения которых ранее применялись отдельные микросхемы, кроме размещения двух уровней кэш в кристалл был встроен контроллер оперативной памяти. Благодаря новым технологическим решениям появилась возможность строить решения с многоядерной архитектурой. Уже в 2001 г. IBM интегрировала на одном кристалле два 64-разрядных процессорных ядра, что позволило на старших серверах IBM получить вдвое повышенную производительность. Процессор POWER в 2006 г. был рекордсменом по числу выходных контактов – 3181. Фирма IBM выпускает широкий спектр серверов, имеющих модульную структуру.

Серверы RISC/UNIX платформы Sun SPARC связаны с появлением в 1989 г. первой массовой рабочей станции SPARCStation1. В 1991 г. появился первый сервер Sun-SPARC server 600 MP. Развитие SPARC-платформы связано с тремя компаниями: Sun Microsystem, Fujitsu и Texas Instruments.

Первые две занимаются разработкой и совершенствованием архитектуры SPARC, последняя изготавливает процессоры.

Будущее SPARC с двумя процессорами нового поколения Niagara и Rock. Niagara, выпущенная в 2006 г. по технологии 90 НМ, включает в себя восемь самостоятельных ядер на кристалл. Кроме этого высокая степень интеграции позволит разместить в кристалле контроллер памяти и четыре контроллера Gigabit Ethernet. Проект

Niagara принципиально однопроцессорное решение для построения SMP-систем.

Микропроцессор Rock (планируется к выпуску во второй половине 2008 г.) представляет собой дальнейшее развитие идей, реализованных в проекте Niagara.

Для очень больших сетей (сотни и более компьютеров рабочих станций) используются мощные многопроцессорные серверы с возможностями наращивания ОП, дискового пространства, быстрым интерфейсом дискового обмена (FAST SCSI и т.п.), несколькими сетевыми интерфейсами. Серверы подобного рода получили название *суперсерверы*. По уровню общесистемной производительности, функциональным возможностям отдельных компонентов, отказоустойчивости, поддержке многопроцессорной обработки и т.д. суперсерверы вышли на уровень мейнфреймов и мощных миникомпьютеров.

3.5. О мобильных компьютерных устройствах

Революционные изменения в компьютерных технологиях, безусловно, связаны с появлением 16-разрядных МП и наступлением эры персональных компьютеров «ПК». Продвижение этой технологии в области компьютеров высокой производительности, как было рассмотрено ранее, привело к созданию новых типов компьютеров – рабочих станций, серверов и суперсерверов, а также кардинально повлияло на традиционные существовавшие до появления ПК компьютеры – мейнфреймы и суперкомпьютеры.

Продвижение же ПК в сторону компактности и мобильности связано с разработкой популярных ноутбуков.

Ноутбук – полностью автономная вычислительная система, имеющая полный набор устройств ввода-вывода и питающаяся от встроенной аккумуляторной батареи.

Одна из главных задач разработчиков – обеспечение максимальной мобильности такой системы, т.е. минимизация массогабаритных характеристик при сохранении, по возможности, уровня производительности.

Нельзя не отметить, что с уменьшением размеров портативных ПК сложно сохранить приемлемый уровень эргономики: пользователи не хотят мириться с маленьким экраном и клавиатурой, огра-

ниченным набором портов ввода-вывода и упрощения дисковой системы.

Комплектующие для ноутбуков изначально проектируются фирмами с учетом именно мобильного применения. Так, ведущие производители процессоров Intel и AMD имеют в своих модельных рядах специализированные процессоры для ноутбуков. Выпускаются специализированные форматы практически всех видов дисковой памяти. Ноутбуки сыграли важнейшую роль в распространении ЖК-мониторов, поскольку они явились пионерами, их применения для ПК способствовали к их широкому внедрению и совершенствованию технологии их изготовления, что позволило значительно снизить их стоимость. Стоимость таких ПК на начальном этапе была достаточно высока, именно из-за стоимости ЖК-мониторов.

Фирмы выпускают широкий спектр ноутбуков, различающихся по техническим характеристикам, по массогабаритным параметрам и стоимости, однако однозначно классификацию их провести затруднительно.

Ноутбуки, предназначенные для замены настольного ПК, имеют большие ЖК-экраны (16 и более дюймов), полноразмерные клавиатуры, полный набор портов ввода-вывода и т.п.

Но обратная сторона этих преимуществ – большие габариты, а также малое время автономной работы. Такие ноутбуки полезны тем, кто редко перевозит их с одного рабочего места на другое и практически никогда не работает в поездах. В последнее время такие ноутбуки часто используют в качестве замены стационарных ПК с питанием от электрической сети.

Большинство современных ноутбуков относится к «тонким и легким», которые значительно более мобильны, их вес не превышает 3 кг, а толщина 30 мм, экраны не более 15 дюймов. Они имеют полноразмерную клавиатуру и достаточно большой набор устройств ввода-вывода. Однако их производительность значительно ниже производительности ПК и у них довольно простая мультимедийная подсистема.

Тем не менее, говоря о производительности и других технических характеристиках ноутбуков, необходимо учитывать стремительное развитие компьютерной технологии. Поэтому временной период отставания технических характеристик ноутбуков от на-

стольных ПК значительно меньший, чем отставание ПК от суперкомпьютеров.

Дальнейшее развитие мобильности ноутбуков связано с так называемыми ультрапортативными бюджетными ноутбуками к которым этот термин уже прижился, но называть их так не корректно. Они образуют новый класс мобильных компьютеров – нетбуки, которые произвели настоящий потребительский бум, значительно превосходящий спрос даже на легкие и тонкие ноутбуки.

Первопроходцем стала компания ASUS, которая осенью 2007 г. анонсировала Eee PS 700 машину с 7-дюймовым экраном, экономичным процессором Intel 512 Мбайт ОП и твердотельным накопительным в 2 Гбайт. После выхода процессора Intel Atom с техпроцессом 47 нм, частотой 1,6 ГГц, 47 млн транзисторов и очень низким энергопотреблением, в среднем всего 0,6 Вт, практически все виды компактных и суперкомпактных компьютеров стали использовать его разновидности.

Появились нетбуки с ОП 1 Гбайт, объемом флэш-дисков до 8 Гбайт, а также HDD-диски 80 Гбайт, web-камеры, сетевой интерфейс, модуль беспроводной связи wi-fi.

Нетбуки по размеру и весу сравнимы с небольшой книжкой и предназначены для тех, кто много времени проводит в поездках и часто работает в пути, однако платой за миниатюрность является малый экран и клавиатура с клавишами меньшего размера.

Так в чем же отличие нетбука от ноутбука? Дело здесь не столько в его размерах, не во времени автономной работы, не в аппаратной конфигурации и даже не в производительности, а в различной модели использования ноутбуков и нетбуков.

Нетбуки – простые в эксплуатации и не дорогие «сверхбюджетные» устройства, предназначенные, в основном, для доступа в Интернет. Они ориентированы на решение таких задач, как web-серфинг, просмотр видео, общение в социальных сетях и т.п.

В прочем утверждать, что нетбук необходим только как средство доступа в Интернет, было бы не верно. Нетбуки можно эффективно использовать в качестве печатной машинки «причем не простой, но и с выходом в сеть», однако нужно учесть, что для повседневной работы с офисными приложениями нетбуки не слишком удобны из-за размеров клавиатуры.

Нужно четко представлять, что с помощью любого портативного ноутбука можно с успехом решать любые задачи, доступные нетбуку, но обратные утверждения не верны.

И в заключение о сверхпортативных мобильных компьютерах нижнего уровня иерархии. Их появление также связано с разработкой МП, хотя они развивались независимо от ПК и ноутбуков. Родоначальником этого направления принято считать так называемые карманные ПК КПК, прадедушками которых можно считать калькуляторы (первый МП был разработан для калькулятора). Включение в состав КПК сотовых модулей привело к появлению двух типов изделий коммуникаторов и смартфонов. Смартфоны ориентированы, прежде всего, на голосовые коммуникации, работа с информацией (от просмотра web до редактирования документов) для них вторична. Коммуникаторы наоборот. Иными словами, смартфон – телефон с дополнительными функциями: коммуникатор – ручной ПК (КПК) со встроенным сотовым модемом. Разные сценарии использования предъявляют разные требования к конструкции аппарата. Смартфон – умный телефон, несколько ограниченный по возможностям с коммуникатором.

Разговоры об исчезновении традиционных КПК (без сотовых модулей) пока преждевременны, поскольку объем их продаж до последнего времени не уменьшается.

И в заключение на одном из форумов, посвященном компактным и сверхкомпактным ПК, в 2007 г. было объявлено о появлении MID-устройств (Mobile internet divece) – мобильных устройств доступа в Интернет. По формату MID-устройства являются аналогами КПК, однако их функциональные возможности гораздо шире, чем у КПК. Именно поэтому предполагается, что в будущем MID-устройства полностью заменят КПК. Они могут иметь разные экраны (4,5 – 6 дюймов), наличие клавиатуры в них необязательно, но предусматривается операционная система Linux или Windows XP.

MID-устройства позволяют пользователям общаться через Интернет, получать доступ к информации, а кроме того, прослушивать музыку и видеоролики. Предполагается, что MID-устройства станут основой нового класса потребительской техники следующего поколения – портативных Интернет-видеоплееров, навигаторов и пр.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. В чем причины повышенного интереса к микропроцессорам? Дайте их определение. Какова их классификация, динамика и основные тенденции в развитии микропроцессоров фирмы Intel?

2. Перечислите различные типы архитектур микропроцессоров и области их применения. Назовите основные тенденции в развитии архитектуры микропроцессоров. Перечислите номенклатуру и ведущие фирмы-производители.

3. Каковы три этапа в развитии ПК, их основные достоинства концепции открытой архитектуры ПК IBM-клона? Назовите причины высокой популярности этих компьютеров. Перечислите основные этапы их развития. Каково современное состояние и направления их использования?

4. Какова динамика развития архитектуры ПК? В чем причины ее усложнения? Каков современный дизайн, типы использования шин. Назовите состав периферийных устройств и их интерфейсы.

5. Сформулируйте определение рабочих станций и серверов, общность и различия. Каковы основные направления их использования, классификация и современные тенденции в их развитии? В чем причины популярности серверов младшего уровня?

ГЛАВА 4. СЕТИ КОМПЬЮТЕРОВ И СРЕДСТВА ТЕЛЕКОММУНИКАЦИОННОГО ДОСТУПА

4.1. Эволюция распределенных вычислительных систем

В начале XIX в. передача электроэнергии и коллективное ее использование привели к технической революции во всех областях деятельности человеческого общества.

В наше время решается, быть может, более важная задача – задача передачи информации, накапливаемой в банках данных. Коллективный опыт и знания, сконцентрированные в этих хранилищах информации, становятся доступны очень широкому кругу абонентов.

В 1993 г. группа экспертов подготовила по заказу Европейского союза материал «Европейское глобальное информационное сообщество (рекомендации для Европейского союза)». В этом материале утверждалось, что информационные и коммуникационные технологии вызвали новую революцию – информационную. Эта революция базируется на информации как таковой, а технологические возможности беспрецедентно увеличивают возможности интеллекта человека и изменяют способы совместной работы, общения и жизни человеческого общества. В качестве примера такой технологии в рекомендациях было названо глобальное сообщество компьютерных сетей Internet.

Средства связи и компьютеры находились в тесном взаимодействии с момента возникновения вычислительных машин, в которых использовались телетайпы, перфораторы и другое телеграфно-телефонное оборудование. Объединение средств телекоммуникации и компьютеров связано с созданием компьютерных сетей. Начало компьютерным сетям положили разработки первых машин с разделением времени* в начале 60-х годов, когда компьютер был дорогостоящим и дефицитным оборудованием. Штатное удаление периферии от большого компьютера не превышало 60 м (системы IBM 390 позволяют увеличить это расстояние до 11 км, применив

* Режим разделения времени (Time Sharing) реализуется на уровне операционных систем и обеспечивает одновременную работу большого количества пользователей с использованием одного и того же ЦП; при этом, безусловно, предполагается аппаратная поддержка многозадачности ЦП. Более подробно этот вопрос будет рассмотрен в гл. 6.

для связи оптоволокно). Использование стандартных телеграфно-телефонных каналов позволяло большому количеству пользователей на значительном удалении иметь возможность доступа к ресурсам вычислительных комплексов большой (по тем временам) мощности. Этот вид сетей интенсивно развивается и в настоящее время представляет так называемые глобальные вычислительные сети, хотя их роль и задачи значительно изменились по сравнению с первоначальными замыслами. Расширился также круг используемых средств телекоммуникаций (радиорелейная, спутниковая связь и т.п.). Появление малых компьютеров, а затем и персональных значительно уменьшило потребности в вычислениях и другой обработке информации на больших компьютерах. И хотя стоимость передачи данных снижается, темпы снижения стоимости каналов отстают от темпов снижения стоимости компьютеров, что на определенном этапе препятствовало развитию систем передачи данных, так как использование местных компьютеров оказывалось дешевле.

Однако в связи с расширением сферы использования компьютеров развитие вычислительных сетей становится актуальнее, чем это было раньше. Среди причин, требующих объединения компьютеров и создания вычислительных сетей, можно отметить следующие:

компьютер все в меньшей степени используется как вычислитель, более актуальными становятся информационные системы для хранения, модификации и поиска данных, количество и сложность организации которых становятся все более трудоемкими и требующими компьютеров очень большой мощности;

информация в банках (базах) данных очень часто обновляется с абонентских пунктов или с периферийных компьютеров, чтение и запись осуществляется в реальном масштабе времени (как, например, система резервирования билетов, банковские операции и др.), что требует хранения информации в одном месте и ее централизованной обработки;

повышение надежности и резервирование, использование общей (дорогостоящей) периферии и многое другое;

значительно повышается эффективность использования систем передачи данных (и их скорости) и уменьшается стоимость их использования.

4.2. Способы передачи данных по физическим линиям

Взаимное проникновение вычислительной техники и технических средств связи оказало серьезное влияние как на структуру компьютеров, так и на структуру каналов связи.

Средства связи, предназначенные для передачи информации между людьми, имеют длительную историю, развитую структуру (в мировом масштабе), мощную научную и технологическую базу и, начиная с 60-х годов, стали использоваться для передачи данных, т.е. для передачи информации между техническими средствами вычислительной техники, что потребовало включения в каналы связи дополнительных технических устройств.

В настоящее время для распределенных вычислительных систем наиболее широко используются телефонные каналы.

На рис. 4.1 представлена упрощенная схема линии аналоговой междугородней телефонной связи.

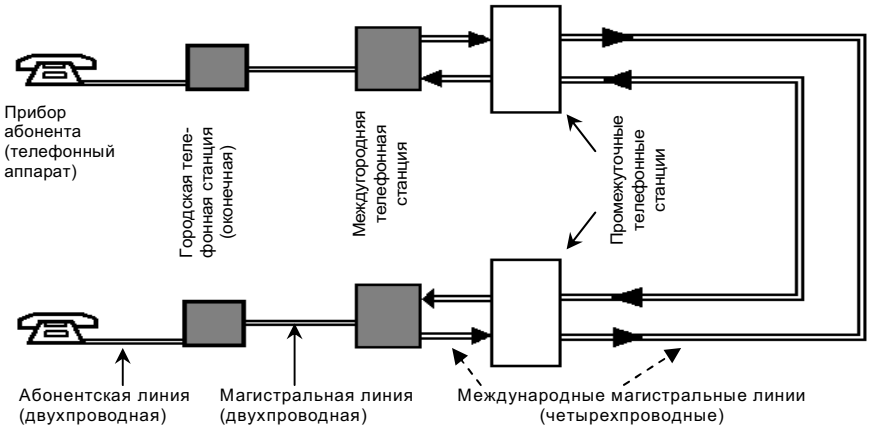


Рис. 4.1. Схема междугородней телефонной связи

На участке от телефонного аппарата до местной АТС происходит передача в первичной полосе частот $\approx$ 200 – 3100 Гц (полоса частот человеческого голоса). При этом от каждого аппарата до АТС проводится двухпроводная электрическая линия для передачи этого сигнала, в дальнейшем происходит преобразование его в иную форму с целью уплотнения передачи. В каждом из последующих кана-

лов идет очень большое количество передач. Существует два типа уплотнения: *частотное* и *временное*. В традиционных линиях связи, как правило, используется частотное уплотнение.

Сущность частотного уплотнения для трех абонентов представлена на рис. 4.2.

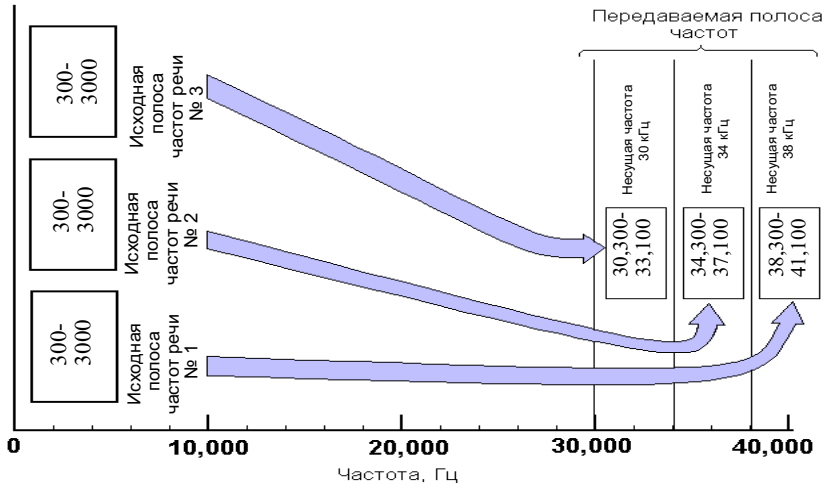


Рис. 4.2. Частотное уплотнение телефонных каналов

Передаваемый сигнал на входе в магистральную линию «смешивается» с так называемой несущей (более высокой) частотой и одновременно с другими сигналами (которые передаются на других несущих частотах) распространяется по магистралям. На выходе передаваемые сигналы выделяются из несущих частот и по индивидуальным линиям доставляются абоненту.

Процедура на входе в магистральные линии связана с различного вида модуляцией несущих частот, а обратное преобразование на выходе из магистральных линий, соответственно, с демодуляцией.

Модуляция несущей – изменение ее амплитуды, частоты, фазы или комбинации этих характеристик в соответствии с передаваемым сигналом.

Использование описанных средств связи для передачи данных, т.е. замена абонента-человека на техническое устройство (ЭВМ),

требует включения в эту систему дополнительных устройств, адаптирующих информационные СОД к передаче по каналу связи.

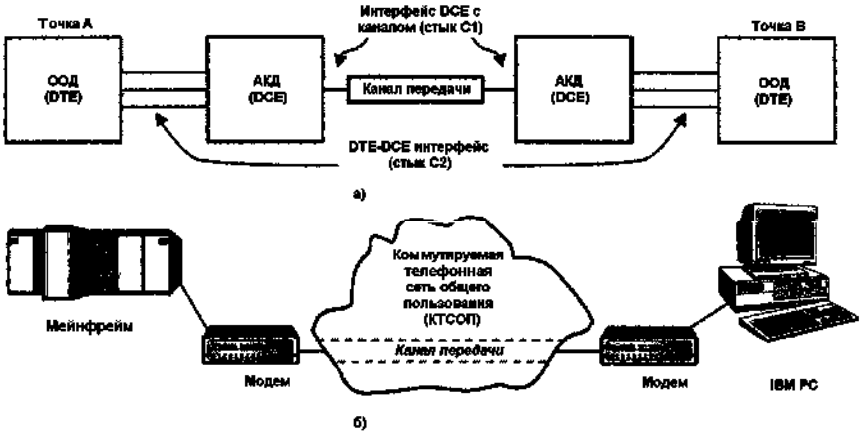


Рис. 4.3. Типовая схема передачи данных: *а* – блок-схема системы передачи данных; *б* – реальная система передачи данных

На рис. 4.3, *а* представлена структурная схема передачи данных, а на рис. 4.3, *б* – реальные физические устройства, которые могут быть включены в канал.

На рис. 4.3, *а* приведены традиционно используемые сокращения для обозначения устройств систем передачи данных (СПД):

ООД – оконечное оборудование данных, в качестве которого может выступать персональный компьютер, большой компьютер, терминал и т.п. В литературе часто употребляется международный термин DTE (Data Terminal Equipment);

АКД – аппаратура канала данных, которую иногда называют аппаратурой передачи данных (АПД), функции которой состоят в обеспечении возможности передачи информации по каналу определенного типа, эти устройства, как правило, называются *модемами* (модулятор-демодулятор); DCE (Data Communications Equipment) – международный термин, используемый для обозначения этого устройства.

Канал передачи включает описанную выше структуру, если используется коммутируемая телефонная сеть общего назначения.

Интерфейс между каналами передачи и АКД в отечественной практике называется «стык 1» (С1), а интерфейс между ООД и АКД «стык 2» (С2).

В зависимости от типа передачи различают аналоговые (традиционно используемые, имеющие длительную историю развития) и цифровые каналы (систем ИКМ, ISDN* и др.), являющиеся битовым трактом с цифровым импульсным сигналом на выходе и входе канала. Цифровые каналы отличаются рядом преимуществ перед аналоговыми, поэтому вновь создаваемые системы передачи данных стараются строить на основе цифровых каналов. Следует отметить, что цифровые каналы весьма успешно применяются не только для передачи данных, но и в средствах бытовой связи (звук, изображение и т.д.), при этом аналоговые сигналы кодируются в цифровые перед передачей в канал.

Термины «аналоговый» и «цифровой» соответствуют непрерывным и дискретным процессам и используются при обсуждении коммуникационных систем в различных контекстах – данных, сигналов и передачи.

Аналоговые данные представляются физической величиной, которая может изменяться в непрерывном диапазоне значений. Величина прямо пропорциональна данным или является их функцией.

Цифровые данные принимают дискретные значения – текст, целые числа, двоичные данные.

Аналоговый сигнал – непрерывно изменяющаяся электромагнитная волна, распространяющаяся в различных средах.

Цифровой сигнал – дискретный (разрывной) сигнал, такой, как последовательность импульсов напряжения.

Возможны четыре вида передачи данных:

цифровые данные – цифровой сигнал, используется наиболее простое оборудование;

аналоговые данные – цифровой сигнал, необходимо преобразование аналоговых данных в цифровую форму, что позволяет использовать современное (высокоэффективное) оборудование передачи данных;

цифровые данные – аналоговый сигнал, необходимость преобразования связана с тем, что через некоторые среды (оптоволокно,

* ISDN – Integrated Services Digital Network.

беспроводные среды) может распространяться только аналоговый сигнал;

аналоговые данные – аналоговый сигнал, традиционная передача, аналоговые данные легко преобразуются в аналоговый сигнал.

Среди преимуществ цифровой передачи необходимо отметить следующие.

Быстрое развитие цифровых систем и уменьшение цены и размеров оборудования, цены и размеры аналогового оборудования остаются на прежнем уровне. Обслуживание цифровых систем намного дешевле аналоговых.

Использование повторителей (в цифровых системах) вместо аналоговых усилителей позволяет передавать данные на большие расстояния по менее качественным линиям (нет накопления шумов) – сохранение целостности данных.

Большая пропускная способность дает возможность более полно использовать пропускную способность оптоволокна и спутниковых средств связи. Временное разделение оказывается более эффективным, чем частотное.

Используется интеграция, когда при обработке аналоговой и цифровой информации по цифровым технологиям все сигналы имеют одинаковую форму (вид). Это позволяет сэкономить на оборудовании и трудозатратах при интеграции: голос, видео, цифровые данные.

Термин «модем» (DCE) применяется в настоящие время (в связи с распространением цифровых каналов) достаточно широко, при этом необязательно подразумевается какая-либо модуляция, а просто называются определенные операции преобразования сигналов, поступающих от DTE для их дальнейшей передачи по используемому каналу.

Существует очень много разновидностей модемов, отличающихся:

по области применения – для коммутируемых и выделенных каналов, для цифровых систем передачи (CSV/DSU), для сотовых систем связи, радиорелейных и т.п.;

по конструкции – внутренние (вставляемые в разъемы компьютера) и внешние, портативные, групповые и т.п.;

по методу передачи – асинхронные, синхронные, синхронно-асинхронные.

Асинхронный метод передачи (или стартстопный) – посимвольный режим передачи с контролем начала и конца символа, имеет низкую скорость и малую эффективность. *Синхронный метод передачи* осуществляет объединение большого количества символов или байт в отдельные блоки – кадры, которые передаются без задержек между восьмибитными элементами.

Очень важной характеристикой канала передачи являются режимы его работы в зависимости от направления возможной передачи данных:

симплексный – передача осуществляется по линии связи только в одном направлении;

полудуплексный – передача ведется в обоих направлениях, но попеременно во времени (технология Ethernet);

дуплексный – передача ведется одновременно в двух направлениях.

Дуплексный режим – наиболее универсальный и производительный. Самым простым вариантом организации дуплексного режима является использование двух независимых функциональных каналов (двух пар проводников или двух световодов) в кабеле, каждый из которых работает в симплексном режиме, т.е. передает данные в одном направлении. Такая организация дуплексного режима применяется во многих сетевых технологиях (Fast Ethernet, ATM и т.п.).

Теперь несколько слов о повышении эффективности использования среды передачи данных.

Существует два основных способа повышения эффективности.

Первый способ, о котором уже упоминалось, – уплотнение информации (или в современной терминологии – мультиплексирование), т.е. функция, позволяющая двум или более источникам данных совместно использовать общую среду передачи данных таким образом, что каждый получает собственный канал передачи данных.

Традиционным считается частотное мультиплексирование, о котором достаточно подробно говорилось ранее, – разделение системы передачи на два или более канала путем разделения всей доступной полосы частот на более узкие полосы, каждая из которых образует отдельный канал.

Второй тип – временное мультиплексирование, интенсивное внедрение которого связано с современным развитием систем пе-

редачи на два или более канала путем поочередного подключения общей линии к разным информационным каналам. Распространению временного мультиплексирования способствовало значительное увеличение пропускной способности каналов, поскольку чем выше пропускная способность канала, тем больше эффективность временного мультиплексирования.

Различают синхронное временное мультиплексирование – методика временного мультиплексирования, когда порядок выделения временных интервалов жестко задан, в отличие от статистического (асинхронного) временного мультиплексирования, при котором временные интервалы, на которые общая линия выделяется устройством, а также порядок их выделения не определены заранее (определяется динамически).

Частотное мультиплексирование – разделение системы передачи на два или более канала путем разделения всей доступной полосы частот на более узкие полосы, каждая из которых образует отдельный канал.

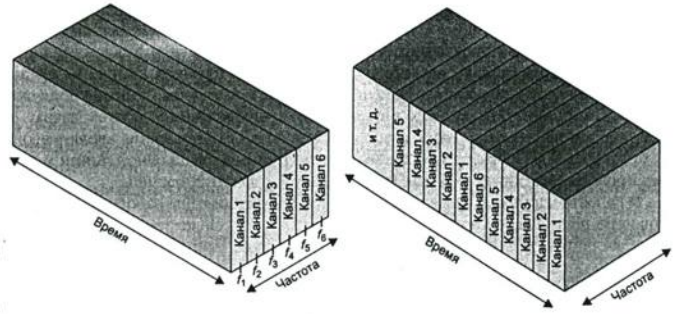
Общий случай частного мультиплексирования иллюстрирует рис. 4.4, *а*. Шесть источников сигналов подключены к мультиплексору, который модулирует каждый сигнал определенной частотой (f_1, …, f_6). Для передачи каждого сигнала требуется полоса частот, центр которой находится вблизи несущей частоты. Эта полоса частот обычно называется *каналом*. С целью предотвращения перекрестного шума каналы разделяются защитными полосами, которые не используются для передачи информации.

Сигнал, передаваемый по среде передачи, является аналоговым. Однако входные сигналы могут быть как цифровыми, так и аналоговыми. В случае цифровых сигналов их необходимо пропустить через модемы для преобразования в аналоговую форму. В любом случае должен модулироваться для сдвига в необходимую полосу частот.

Временное мультиплексирование – разделение системы передачи на два или более канала путем поочередного подключения общей линии к разным информационным каналам.

Временное мультиплексирование становится возможным, когда скорость распространения сигналов в среде превышает скорость их передачи. В таком случае ряд цифровых или аналоговых сигналов может передаваться одновременно путем поочередной передачи «порции» каждого сигнала. Общий случай временного мультип-

лексирования показан на рис. 4.4, *б*. Шесть источников сигналов подключены к мультиплексору, который чередует биты сигналов, поочередно передавая информацию от каждого из источников. Этот мультиплексор обладает шестью входами. Если каждый из них может поддерживать скорость передачи данных, скажем, 9,6 Кбит/с, тогда единая линия пропускной способностью 57,6 Кбит/с будет передавать сигналы сразу от всех шести источников.



а *б*

Рис. 4.4. Частотное (*а*) и временное (*б*) мультиплексирование

Второй способ повышения эффективности использования среды передачи данных – компрессия или сжатие данных, заключающееся в уменьшении количества битов, требуемых для представления данного объема информации. Это позволяет увеличить объем информации, передаваемой по линии, сократить сеанс связи, кроме этого кодирование информации, связанное с компрессией, повышает информационную безопасность.

4.3. Понятие подхода открытых систем, проблемы стандартизации, многоуровневый подход, стек протоколов

Применение метода открытых систем в настоящее время является основной тенденцией в области информационных технологий и средств вычислительной техники, поддерживающих эти технологии. Идеологию открытых систем реализуют в своих разработках все ведущие фирмы-поставщики средств вычислительной техники,

передачи информации, программного обеспечения и разработчики прикладных программных систем. Их результативность на рынке информационных технологий и систем определяется в значительной степени реализацией стандартов открытых систем.

Потребность в развитии открытых систем возникла давно. Это связано, в частности, с растущим использованием разнообразных аппаратных платформ, использующих приложения, предназначенные для решения все более широкого диапазона задач.

Внедрение гетерогенных систем (от греческого – разнородный) и желание разделять между такими системами информацию привело к необходимости их совместной работы. Кроме того, совместимые компьютеры сулили переносимость программ с платформы* на платформу.

Потребность в применении возможностей открытых систем исходит не только от пользователей. В таких системах заинтересованы также разработчики, озабоченные необходимостью сокращать расходы и время переноса своих приложений на различные платформы. В них заинтересованы как производители, старающиеся разрабатывать новые платформы, на которых можно выполнять широкий диапазон приложений, так и международные стандартизирующие организации, которым необходимо своевременно разрабатывать стандарты, отвечающие требованиям промышленности.

Ранняя концепция открытых систем вызвана стремлением пользователей к переносимости приложений. Этот этап связан с созданием компьютеров IBM 360 («360» расшифровывается как «третье поколение, 60-е гг.), обладающих единым набором команд и способных использовать одну и ту же операционную систему. Однако системы IBM 360 не были массовыми, и на большей части рынка их переносимость никак не сказывалась.

Следующая фаза в развитии концепции открытости связана с областью интерактивной компьютерной обработки и расширяющимся диапазоном, требующим переноса продуктов. В этот период DEC разработал свои VAX-системы (VAX – Virtual Address Extension), работающие под управлением операционной системы VMS (Virtual

* *Платформа* – программно-технические условия, в рамках которых работает (или не работает) данная прикладная программа.

Machines System). Каждый компьютер этой линии, вне зависимости от размера мог исполнять один и тот же набор приложений.

И, наконец, окончательное оформление принципов открытых систем связано с взрывным развитием сетевых технологий, таких, как DEC net или Internet (TCP/IP), объединявших системы и локальные сети самых разных организаций. Когда сетевая обработка стала реальностью, пользователи начали обращать внимание на совместимость и интеграцию как на необходимый атрибут открытых систем.

Одно из простейших определений открытой системы приводится в руководстве французской ассоциации пользователей UNIX (1992 г.).

Открытой называется система, которая состоит из компонентов, взаимодействующих друг с другом через стандартные интерфейсы. Общие свойства открытых систем:

1) расширяемость-масштабируемость – Extensibility/ scalability;

2) мобильность (переносимость на различные платформы) – portability;

3) интероперабельность (способность к взаимодействию с другими системами) – interoperability;

4) дружественность к пользователю, в том числе легкая управляемость – driveability.

Эти свойства, взятые по отдельности, использовались предыдущими поколениями информационных систем и средств вычислительной техники. Новый взгляд на открытые системы связан с распространением сетевых технологий и необходимостью рассматривать эти свойства в комплексе. Этот подход при построении распределенных СОД дает следующие преимущества:

возможность построения сети из аппаратных и программных средств различных производителей, придерживающихся одного и того же стандарта;

возможность замены отдельных компонентов сети другими, более совершенными, что позволяет сетям развиваться с минимальными затратами;

возможность легкого взаимодействия сетей между собой;

простота освоения и обслуживания.

Полезность стандартизации, справедливая для всех отраслей, в компьютерных сетях приобретает особое значение. *Сеть* – соеди-

нение большого количества очень разнообразного оборудования и организация взаимодействия между устройствами, проблема совместимости которых является одной из наиболее сложных.

Идеологическая основа стандартизации в распределенных СОД – многоуровневый подход к разработке средств сетевого взаимодействия. Вся система передачи сообщений (данных) разбивается на иерархические уровни (принцип декомпозиции) и на каждом уровне применяются свои правила взаимодействия – протоколы. *Протоколом* в мире коммуникаций называют распределенные алгоритмы, определяющие, каким образом осуществляется обмен данными между физическими устройствами и логическими объектами. Аналогом многоуровневого подхода может служить организация почтовой связи (рис. 4.5).

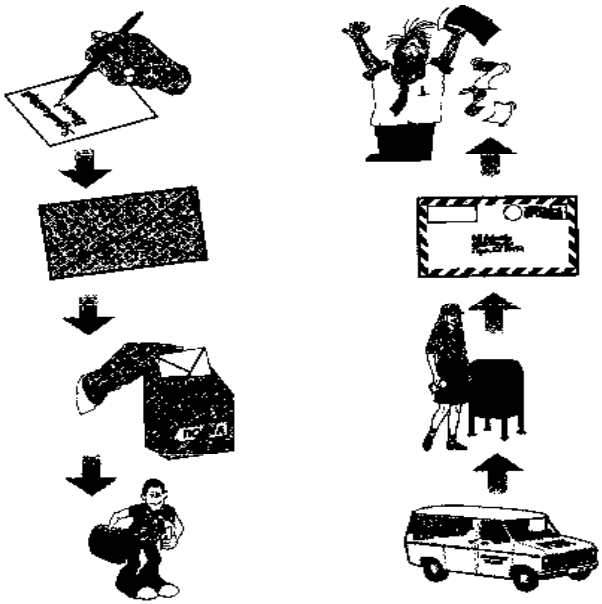


Рис. 4.5. Функциональные уровни системы почтовой связи

Теоретическую основу современных информационных систем определяет базовая эталонная модель открытых систем (OSI – Open

System Interconnection), разработанная международной организацией стандартов (ISO – International Standards Organization).

Рис. 4.6. Стек протоколов OSI

Иерархический набор протоколов, достаточный для организации взаимодействия узлов в сети, называется *стеком коммуникационных протоколов*. Стек OSI (рис. 4.6) включает в себя следующее:

первый уровень – уровень приложений (Application Layer), отвечает за поддержку прикладного программного обеспечения пользователя (протокол FTP, Telnet и т.п.);

второй уровень – уровень представления (Presentation Layer), служит для преобразования данных из внутреннего формата компьютера в формат, приемлемый для компьютера абонента, с которым устанавливается связь (ситуация связи с неоднотипными компьютерами – IBM PC, Macintosh, DEC, NEXT и т.п.);

третий уровень – сеансовый (Session Layer), обеспечивает координацию связи между двумя прикладными программами (создание сеанса связи, управление передачей и приемом сообщений и завершение сеанса);

четвертый уровень – транспортный (Transport Layer), в функции которого входит разделение сообщения на пакеты (при необходимости) при передаче и сбор пакетов при приеме, контроль за правильностью их прохождения, согласование сетевых уровней различных, не совместимых между собой, сетей через специальные шлюзы;

пятый уровень – сетевой (Network Layer), выполняет функции переключения и маршрутизации пакетов, отвечает за адресацию и доставку пакетов;

шестой уровень – канальный (или уровень управления линией передачи данных – Data Link), определяет правила использования физического канала связи (обеспечивает контроль доступа к среде в соответствии с используемыми протоколами и осуществляет логический контроль связи – разбиение на кадры, контроль за ошибками, передачи по каналу и т.п.);

седьмой уровень – физический (Physical Layer), определяет физические, механические и электрические характеристики связи, отвечает за аппаратное обеспечение (предыдущий уровень носит программно-аппаратный характер).

В стандарте ISO для обозначения данных, с которыми имеют дело протоколы разных уровней, используется общее название «протокольный блок данных» (Protocol Data Unit, PDU). Однако размеры этих блоков могут быть различны на разных уровнях протоколов. В связи с этим для обозначения единиц данных могут применяться различные названия. Наряду с термином «сообщение» (message), применяемом на самом верхнем уровне, используется термин «пакет» (packet), применяемый на сетевом уровне, «кадр» (frame) – на нижнем уровне.

Типичная структура пакета сообщений, характерная для любого уровня, представлена на рис. 4.7.

Рис. 4.7. Структура пакета сообщений

На рис. 4.7 CRC – контрольная сумма для обнаружения ошибок при передаче. На каждом уровне передачи применяются свои протоколы, имеющие общие интерфейсы с соседними уровнями. При этом используется подход, при котором протоколы как бы наслаиваются один на другой (рис. 4.8) в виде известной матрешки. На каждом уровне выполняются только собственные функции, функции других уровней для них недоступны.

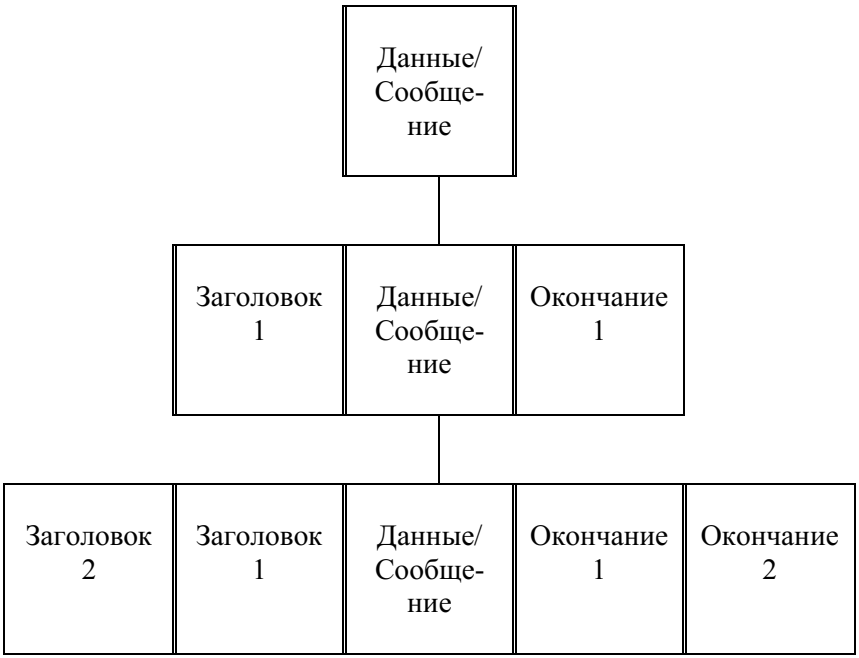


Рис. 4.8. Вложенность пакетов сообщений

Когда сообщение по сети поступает в машину (адресат), оно принимается ее физическим уровнем. Каждый уровень анализирует и обрабатывает заголовок своего уровня, выполняя соответствующие данному уровню функции, а затем удаляет заголовок и передает сообщение вышестоящему уровню.

4.4. Глобальные вычислительные сети, методы доступа, современное состояние и ближайшие перспективы

Глобальные или всемирные сети (GAN – Global Area Network) предназначены для связи абонентов, расположенных в различных географических регионах, и включают в себя множество сетей различной протяженности – материковые (WAN – Wide Area Network), региональные или городские ГВС (MAN – Metropolitan Area Network), а также локальные (LAN – Local Area Network), о которых будет сказано отдельно, так как они имеют ряд принципиальных отличий от перечисленных выше, кампусная сеть, объединяющая локальные сети, близко расположенных зданий (CAN – Campus* Area Network).

Особый вид сетей – так называемые корпоративные (EWN – Enterprise Wide Network). Термин «корпоративная» отражает, с одной стороны, величину сети, так как корпорация – крупное большое предприятие. С другой стороны, этот термин несет в себе смысл объединения, т.е. корпоративная сеть – сеть, получившаяся в результате объединения нескольких, как правило, разнообразных сетей. Кроме этого дух корпоративности – дух некоторого единства, общности, в этом смысле корпоративные сети – сети, в которых неоднородные компоненты живут в единстве и согласии. Эти сети могут включать в себя все перечисленные выше типы сетей, сложно связаны и могут покрывать город, регион или даже континент.

В глобальных вычислительных сетях (ГВС) на первый план выходят администрация сети и средства связи, так как именно они составляют основную эксплуатационную стоимость сети. В сеть объединяются большие и суперкомпьютеры, а также центры с многомашинной структурой. В качестве абонентских пунктов, как правило, используются персональные компьютеры, в качестве каналов

* Campus – студенческий городок.

связи – традиционные средства телекоммуникаций (телеграфно-телефонные линии, оптоволоконные каналы, спутниковые каналы, радиорелейная связь и т.п.). Топология сетей имеет сложную разветвленную структуру.

В качестве методов доступа (протоколов нижнего уровня) обычно используют коммутацию цепей, коммутацию пакетов, селекцию пакетов. Рассмотрим кратко достоинства и недостатки этих методов.

1. *Принцип коммутации цепей* – традиционная коммутируемая телефонная связь. На рис. 4.9 представлена схема установления связи между двумя абонентами при таком методе доступа (эту схему следует рассматривать с учетом рис. 4.1, 4.2 и 4.3).

Как видно из рисунка, каждый разговор или передача данных резервирует на время некоторую часть сети, отводимую переключающим устройством (коммутатором). При этом два абонента соединяются напрямую одной цепью. Главное достоинство этого способа – повсеместное распространение персональных телефонных линий. Однако он имеет целый ряд недостатков:

низкая эффективность использования сети (если среднестатистический человек произносит в минуту 12 – 20 слов, что соответствует примерно 30 бодам*, то пропускная способность каналов составляет от 2400 до 10^7 бод и более); проводя аналогию, американцы говорят, что такой способ подобен перекрытию одной полосы шоссе № 61 Нью-Йорк – Вашингтон и пропусканию по ней по одной машине в течение всего времени движения ее на этом участке;

на установление связи (соединения) также затрачивается время, за которое (при указанных пропускных способностях канала) может быть передано очень большое количество информации (хотя для телефонных разговоров это время может показаться незначительным);

высокий уровень шумов в коммутирующих АТС может вызывать сбои при передаче данных;

необходимость одинаковых (синхронных) устройств у абонентов, имеющих равные скорости приема-передачи.

---

* *Бод* – единица измерения скорости передачи в каналах связи. Названа в честь французского изобретателя телеграфного аппарата Эмиля Бодо. 1 бод ≅ 1 бит/с и зависит от методов кодирования в каналах.

Тем не менее, этот вид компьютерной связи использовался до последнего времени частными абонентами при подключении домашних ПК.

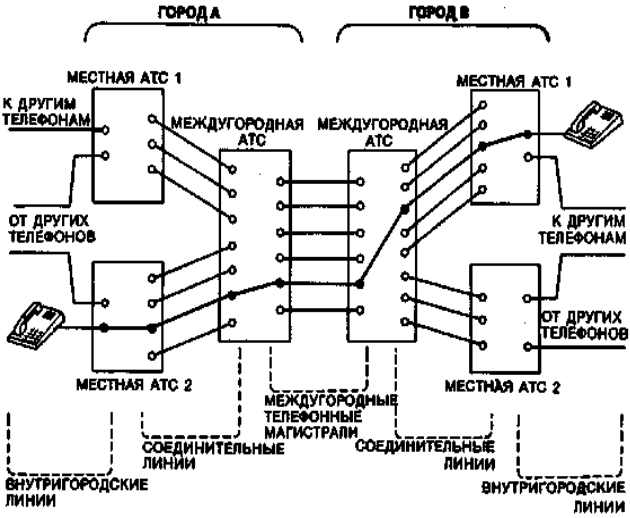


Рис. 4.9. Принципиальная схема коммутации цепей

2. *Принцип коммутации пакетов* представлен на рис. 4.10. Коммутация пакетов была изобретена специально для компьютерных сетей. На рис. 4.10, *а* компьютер передает сообщение. В первом узле коммутации пакетов (или на интерфейсе между компьютером и узлом) сообщение разбивается на пакеты каждый со своим адресом (см. рис. 4.10, *б*). Пакеты передаются по сети независимо. Каждый узел на трассе пакета выбирает дальнейшее направление его движения, учитывая информацию о потоках, неисправностях и т.п., полученную от соседних узлов. В результате пакеты могут пойти разными путями (см. рис. 4.10, *в*) и прибыть в точку назначения в неправильном порядке (см. рис. 4.10, *г*). Узел коммутации в точке назначения восстанавливает порядок перед доставкой их принимающему компьютеру (см. рис. 4.10, *д*).

При коммутации пакетов информация, посылаемая одним из компьютеров другому, не передается непрерывным потоком по отведенной для нее цепи. Вместо этого компьютер каждого пользова-

теля подключается к узлу (через коммутируемые телефонные линии) – мощному компьютеру, который разбивает поступающее сообщение на пакеты одинаковой длины, каждому из которых предшествует небольшая адресная информация (адрес назначения, адрес посылки, порядковый номер пакета и т.п.). Затем пакеты поступают в сеть, в которой все узлы скоммутированы (соединены). Сеть действует как быстродействующий вариант почты, пакеты самого разного назначения могут использовать одни и те же линии. При этом:

1) нет задержки при коммутации, так как таковая отсутствует; уменьшается полная задержка сообщений и объем памяти, так как пакет может быть передан, как только он поступил, до получения всего сообщения;

упрощается управление буферной памятью, так как размеры пакета фиксированы;

при обнаружении ошибки требуется повторить передачу только одного пакета, а не всего сообщения;

короткие порции приводят к более эффективному использованию пропускной способности, временные промежутки между пакетами могут использоваться другими сообщениями;

отказ одной линии (обрыв) не приводит к нарушению связи, так как маршрутизаторы (в узлах) отправят сообщения по другим направлениям;

2) нет избыточности пропускной способности, так как нет отдельных резервируемых линий;

Небольшая часть пропускной способности расходуется на маршрутизацию. Кратчайший маршрут по сети определяется совместно узлами в соответствии с распределенным алгоритмом маршрутизации. В каждом узле оценивается расстояние в некоторой «метрике» до всех точек назначения с учетом повреждений линий, загруженности узлов и т.п.

Пакеты одного и того же сообщения могут попасть в точку назначения различными путями, поэтому порядок их прибытия может отличаться от порядка отправления. Узел назначения выстраивает относящиеся к нему пакеты в правильном порядке. С точки зрения пользователя все выглядит так, как будто ему выделена целая цепь. При этом способе передачи говорят о «виртуальной цепи», или «виртуальном канале». В случае, когда порядок следова-

ния не важен или сообщение размещается в одном пакете, протокол передачи значительно упрощается (повышается производительность канала) и говорят о дейтаграммном протоколе. Дейтаграмма (или датаграмма) в данном случае – синоним пакета.

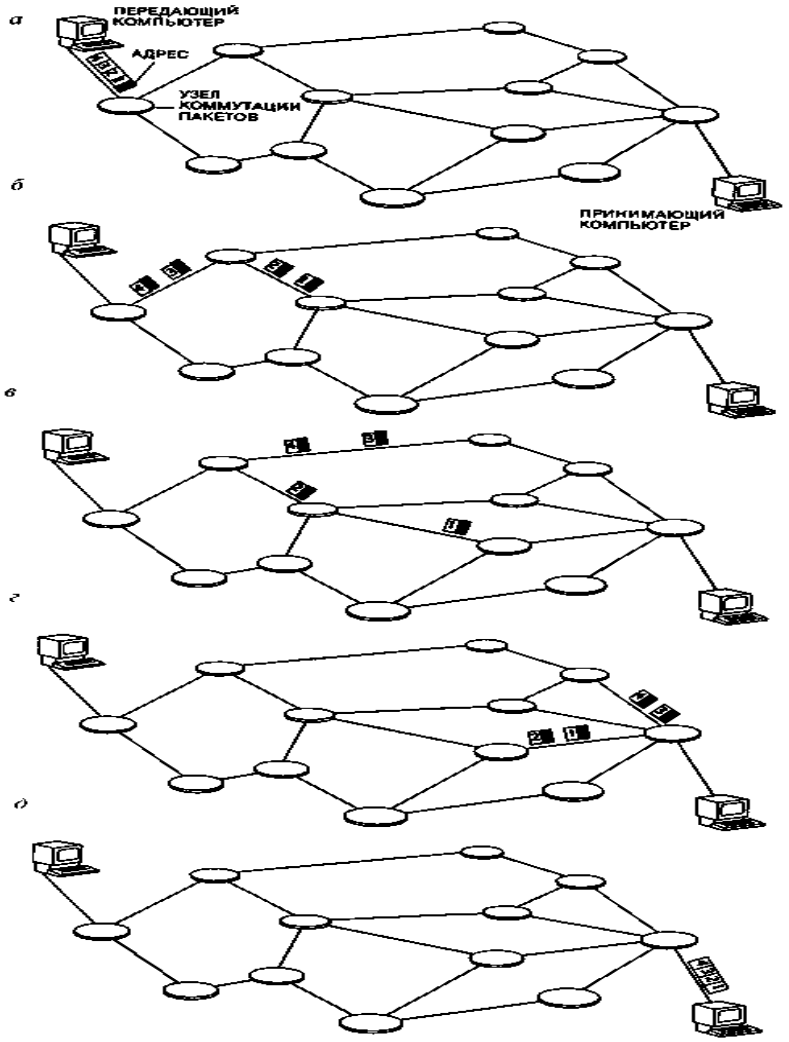


Рис. 4.10. Принцип коммутации пакетов

3. *Метод селекции пакетов* представлен на рис. 4.11.

Пакеты рассылаются всем абонентам без исключения. Адресат отбирает адресованные ему пакеты, а остальные игнорирует. Одной из первых сетей такого типа была созданная в 70-х годах профессором Гавайского университета Абрамсоном сеть Алоха.

Передающая среда, называемая моноканалом, – конус пространства с вершиной в пассивной отражающей антенне геостационарного спутника (израсходовавшего запасы питания). В зоне приема отраженных сигналов расположены университеты на Гавайских островах, в Японии, США, Австралии. Абоненты имеют радиоприемник и радиопередатчик с остронаправленной на спутник антенной.

Рис. 4.11. Метод селекции пакетов

Каждый из них может отправить на антенну-отражатель сообщение любому из четырех адресатов. Отразившись, оно придет на антенны всех абонентов, в том числе и на антенну пославшего его абонента. Так, на рисунке сообщение посылает абонент 1 для абонента 4; все, кроме 1 и 4, это проигнорируют, 4-й его примет, а первый проверит его правильность. Наложение двух сообщений будет проконтролировано отправляющим и принимающим сообщениями, и через случайный промежуток времени посылка будет

повторена. Сеть оказалась очень дешевой, а ее низкая эффективность в первоначальном варианте была значительно повышена за счет наложения дополнительных условий на прием-передачу сообщений. Сети такого типа получили довольно широкое распространение, в том числе и в отечественной практике. Метод селекции пакетов интересен и в связи с тем, что его принцип используется и в самой популярной в мире локальной вычислительной сети Ethernet.

За достаточно длительную историю своего развития (более 40 лет) ГВС претерпели значительные изменения, как в видах предлагаемых услуг, так и в их стоимости. Эти изменения стали возможны из-за быстрого роста пропускной способности каналов. Если в 60-х годах максимальная пропускная способность каналов ограничивалась 30 – 50 Кбит/с, то в 90-х годах вполне доступны каналы в 10 Мбит/с, существуют каналы со скоростями передачи в 1, 10 и более Гбит/с (на световодах).

Потребность в высокоскоростной передаче связана в значительной степени с изменением услуг, предлагаемых в сетях. Все в меньшей степени использование сетей определяется распределением ресурсов компьютеров, когда решающую роль играет вычислительная техника, и все большее значение приобретают информационные системы, где заглавную роль играют средства телекоммуникации. К примеру, очень высокая производительность каналов связана с мультимедийными приложениями (данные, звук, видео). Так, передача в черно-белом изображении на экран 1000x1000 пикселей (точек) с частотой 30 кадров в секунду (чтобы исключить мерцание экрана) создает поток данных в сети 30 Мбит/с. Для передачи цветного изображения (16 цветов) потребуется уже 120 Мбит/с, хотя и разработаны методы, позволяющие снизить требования к пропускной способности каналов для данных приложений. Общая потребность в повышении их производительности растет.

Чтобы удовлетворить новые требования корпораций в области вычислений и передачи данных, производители оборудования и поставщики услуг разработали ряд схем высокоскоростной передачи.

Среди большого количества каналов, применяемых в глобальных сетях, особо следует отметить выделенные (или арендуемые) каналы, в которых используются разнообразные методы доступа и виды передачи.

Среди схем с коммутацией (сеть) следует отметить выделенный канал.

Выделенным называется канал с фиксированной полосой пропускания или фиксированной пропускной способностью, постоянно соединяющий двух абонентов. Абонентами могут быть как отдельные устройства (компьютеры или терминалы), так и целые сети.

Выделенные каналы обычно арендуются у компаний – операторов территориальных сетей, хотя крупные корпорации могут прокладывать свои собственные выделенные каналы.

Выделенные каналы делятся на аналоговые и цифровые в зависимости от того, какого типа коммутационная аппаратура применена для постоянной коммутации абонентов – FDM или TDM. На аналоговых выделенных линиях для аппаратуры передачи данных физический и канальный протоколы жестко не определены. Отсутствие физического протокола приводит к тому, что пропускная способность аналоговых каналов зависит от пропускной способности модемов, которые использует пользователь канала. Модем собственно и устанавливает нужный ему протокол физического уровня для канала.

На цифровых выделенных линиях протокол физического уровня зафиксирован – он задан стандартом G.703.

На канальном уровне аналоговых и цифровых выделенных каналов обычно используется один из протоколов семейства HDLC или же более поздний протокол PPP, построенный на основе HDLC для связи многопротокольных сетей.

Среди некоммутируемых (выделенных) линий наибольшее распространение получили следующие:

аналоговые линии (самый дешевый вариант представляет собой аренду аналоговой витой пары; при использовании модемов, как правило, такой вариант позволяет передавать данные со скоростями от 4,8 до 56 Кбит/с);

служба цифровой передачи данных (высококачественные цифровые линии связи, для использования которых требуются специальные цифровые устройства (вместо модемов); такие линии дороже аналоговых, но позволяют передавать данные с более высокими скоростями;

линии Т-1 и Т-3 (на протяжении многих лет наиболее распространенной выделенной линией для высокоскоростной передачи

данных (в том числе и телефонных разговоров) была все еще популярная линия Т-1; если пропускной способности линии Т-2 недостаточно, можно арендовать линию Т-3);

frame relay (хотя frame relay (ретрансляция кадров) представляет собой коммутируемую сетевую технологию, протокол frame relay может работать на выделенной линии, обеспечивая удобный и гибкий мультиплексный метод передачи данных; для этого подхода потребителю требуется специальное оборудование);

SONET (в выделенных линиях с максимальными скоростями передачи данных используется технология SONET/SDH).

Модем. Модемы, подключенные к общественной телефонной сети, реализуют относительно недорогую, но и низкоскоростную схему передачи данных. Сами модемы недороги, а скорость передачи данных по телефонной линии считается разумной при умеренном времени установки соединения. В различных организациях локальные сети и частные телефонные сети оборудованы модемными пулами для предоставления недорогой услуге по передачи данных.

Стандарт X25. Является одним из наиболее старых типов сетей (применяется с 80-х годов) с коммутацией пакетов. Обеспечивает надежную передачу сообщения, используя принцип виртуальных цепей. Теоретически может поддерживать до 4096 виртуальных цепей в одном канале, практически – намного меньше. Хорошо работает и в физических каналах низкого качества, обеспечивая скорость передачи 1,2 – 64 Кбит/с.

ISDN. Обеспечивает как коммутацию каналов, так и коммутацию пакетов Ч.25 по 64-килобитным каналам B*. Могут предоставляться и более высокие скорости передачи данных. Как правило, тарифы за использование сети рассчитываются, исходя из длительности соединения, независимо от количества переданной информации.

Протокол Frame Relay (ретрансляция кадров), основанный на использовании кадров переменной длины, представляет разновидность коммутации каналов. Сетевой уровень у этого протокола отсутствует, а для повышения пропускной способности многие основные функции упрощены или устранены. Это высокоскоростной

* B-ISDN – широкополосная сеть ISDN (Broadband ISDN) с интерфейсом ATM.

протокол для поддержки цифровой сети общего доступа с интегрированным обслуживанием (B-ISDN). Он предназначен для передачи разных видов информации, в основном для передачи данных. Протокол не содержит компонентов для осуществления контроля прохождения и коррекции ошибок (которые имеются в X25), предполагая высоконадежный физический уровень передачи.

Технология ATM (Asynchronous Transfer Mode), или режим асинхронной передачи, – одна из самых последних разработок и в наибольшей степени отвечает современным приложениям, являясь разновидностью коммутации пакетов, при которой используются ячейки фиксированного размера (53 байта). Так же, как и в протоколе Frame Relay, сетевой уровень отсутствует, а многие основные функции еще в большей степени упрощены или устранены для повышения пропускной способности. Кроме традиционного (для X.25 и Frame Relay) понятия «виртуальные каналы», введено понятие *«виртуальные пути»* – связка виртуальных каналов с одинаковыми конечными точками, таким образом, все ячейки следующего по виртуальным каналам одного пути коммутируются одновременно. Эта технология включает в себя целый комплекс средств: физический интерфейс, технику высокоскоростной коммутации и передачи данных, унифицированную сетевую технологию и т.п. Сети такого типа способны передавать самые различные виды информации: речь, видео, текст. Скорость передачи зависит от конкретных физических каналов.

Скоростные характеристики перечисленных каналов коммутируемых сетей представлены в табл. 4.1.

Сделать выбор из различных вариантов выделенных линий и схем с коммутацией непросто, а с увеличением числа предлагаемых вариантов в 90-е годы такой выбор стал еще сложнее.

В заключение обсуждения наиболее распространенных каналов и используемых в глобальных сетях, несколько слов о технологиях xDSL и кабельных модемах. Технологии xDSL основаны на превращении абонентской линии обычной телефонной сети в цифровую, что и отражено в их названии (Digital Subscriber Line – цифровая абонентская линия). Существует достаточно много различных технологий, основанных на xDSL. Эти технологии рассчитаны на высокоскоростную передачу данных на коротком отрезке витой пары, соединяющей абонента с ближайшей АТС, т.е. решение проблемы «последней линии», отделяющей потребителя от поставщи-

ка услуг (высокоскоростных выделенных линий), особенно необходимых при доступе в Internet.

Общая идея заключается в том, что на обоих концах абонентской линии – на АТС и у абонента – устанавливаются разделительные фильтры (splitter). Низкочастотная (до 3,5 кГц) составляющая сигнала заводится на обычное телефонное оборудование (порт АТС и телефонный аппарат абонента), а высокоскоростная (выше 4 кГц) используется для передачи данных с помощью xDSL-модемов.

Таблица 4.1

Коммуникационные услуги и их характеристики

Некоммутируемая (выделенная) линия							
Размер, Кбайт/с	1 – 10	10 – 100	100 – 10^3	10^3 –10^4	10 – 10^5	10^5 – 10^6	10^6 – 10^7
Аналоговая линия	4,8 – 56						
Служба цифровой передачи данных	2,4 – 56						
T-1			1,54				
Frame Relay			1,54 – 44,736				
T-3					44,736		
SONET					51,84 – 2,488		
Коммуникация* (сеть)							
Модем	1,2 – 56						
Коммутация пакетов X.25	2,4 – 56						
ISDN		64 – 1,54					
ADSL		16 – 9					
Frame Relay			1,54 – 44,736				
ATM				25 – 155			

Поскольку физическая линия (пара проводов) между абонентом и АТС позволяет пропускать сигнал в полосе частот до 1 МГц, дос-

* Коммуникация (или коммуникационная сеть) представляет собой систему, которая соединяет некоторое число устройств и обеспечивает передачу данных между этими устройствами.

тижимые скорости передачи гораздо выше, чем предел в 56 Кбит/с, установленный и достигнутый для обычных модемов. Высокочастотная часть полосы пропускания сигнала может разделяться между встроенными каналами различным способом. При частотном разделении (FDM) часть спектра отдается на передачу в одном направлении, часть – в другом. При использовании эхоподавления (echo-cancellation) вся полоса используется для передачи в обе стороны, а каждое устройство при приеме из общего сигнала вычитает сигнал собственного передатчика. Пропускная способность может быть как симметричной, так и асимметричной. В случае подключения пользователя к сети Internet асимметрия выгодна, поскольку поток к абоненту (страницы текста, аудио- и видеопотоки) гораздо больше обратного (запросы URL).

Наибольшее распространение получила асимметрия ADSL (Asymmetric Digital Subscriber Line), где скорость к абоненту (downstream) до 6,1 Мбит/с, от абонента – 16-640 Кбит/с. Скорость передачи к абоненту кратна скорости каналов E1/T1 (2,048; 4,096 и 6,144 Мбит/с для базового канала E1 или 1,544; 3,088; 4,632 и 6,176 Мбит/с для базового канала T1). Достижимая скорость связана с длиной абонентской линии и ее качеством (сечение проводов, материал изоляции, шаг скрутки, однородность и т.п.). Минимальная скорость обеспечивается на линиях длиной до 5,5 км при диаметре провода 0,5 мм (24 AWG) и до 4,6 км при 0,4 мм (26 AWG). Скорость 6,1 Мбит/с достигается на линиях длиной до 3,7 км при диаметре провода 0,5 мм и до 2,7 км при 0,4 мм.

UADSL (Universal ADSL), она же DSL Line, – улучшенный вариант ADSL с меньшими скоростями (при длине линии до 3,5 км скорости 1,5 Мбит/с и 384 Кбит/с в разных направлениях; при длине до 5,5 км – 640 и 196 Кбит/с). Устройства просты в установке и относительно недороги.

RADSL (Rate Adaptive Digital Subscriber Line) – технология с адаптивным изменением скорости передачи в зависимости от качества линии.

HDSL (High Dana-Rate Digital Subscriber Line) – высокоскоростная технология, обеспечивающая скорости 1,536 или 2,048 Мбит/с в обоих направлениях. Протяженность линии – до 3,7 км, требует четырехпроводной линии.

SDSL (Single-Line Digital Subscriber Line) – симметричная высокоскоростная (1,536 или 2,048 Мбит/с), но на двухпроводной линии при длине до 3 км.

VDSL (Very High Dana-Rate Digital Subscriber Line) – очень высокоскоростная (до 56 Мбит/с), симметричная. Расстояние до 1,5 км. Технология весьма дорога, но рассчитана и на коллективное использование линий. После разделяющего фильтра на абонентской стороне может стоять одиночный модем (или концентратор), а может подключаться через специальную кабельную проводку (коаксиальный кабель или витая пара) и группа модемов, разделяя полосу пропускания предопределенным образом.

Для того чтобы использовать xDSL, провайдер (оператор связи) должен установить свое оборудование на территории АТС обслуживаемого абонента и соединить его с базовой сетью передачи данных каналом достаточной производительности. Конечно, возможны и частные случаи, когда с помощью xDSL объединяются локальные сети в зданиях, охваченных одной АТС. Установка модема ADSL или UADSL на стороне абонента практически не отличается от установки обычного модема. Но технологии xDSL позволяют одновременно и независимо использовать одну и ту же телефонную линию и для передачи данных, и для телефонных переговоров, чего не позволяют обычные модемы для коммутируемых линий.

Кабельные модемы предназначены для работы через сети кабельного телевидения (CATV), для которых используется широкополосный коаксиальный кабель с импедансом 75 Ом. Передача данных ведется параллельно с видеовещанием. Эти модемы к телефонным сетям непосредственного отношения не имеют, они используют кабельное хозяйство операторов услуг кабельного телевидения. Как и ADSL, кабельные модемы асимметричны: скорость к пользователю может достигать десятков мегабайт в секунду, от пользователя – значительно ниже. Кабельные модемы могут быть и симплексными – модем пользователя только принимает нисходящий (downstream) поток данных от модема оператора кабельного TV. При этом восходящий (upstream) поток данных от пользователя должен передаваться по иным каналам (например, ISDN или аналоговым модемам). Для упрощения структуры коммуникаций (но не оборудования) желательно оба потока передавать по одной и

той же кабельной сети. Развитием идеи раздельной передачи потоков является передача нисходящего потока по спутниковым каналам, но пока что это слишком дорогая технология. Кабельные модемы в основном предназначены для предоставления пользователям доступа к сети Internet с высокими скоростями получения информации.

4.5. Internet: краткая история, принципы построения, адресация, основные приложения

Internet с точки зрения технических средств – объединение сетей самого различного масштаба, состоящих из миллионов компьютеров, имеющих единые аппаратно-независимые протоколы среднего уровня (транспортные и сетевые адресные), использующих самые разнообразные каналы связи с различными протоколами канального и физического уровня.

С точки зрения пользователя, Internet – глобальное информационное пространство, значительно расширяющее традиционные средства телекоммуникаций. Информация в этом пространстве может сохраняться и по мере надобности многократно воспроизводиться.

Принято считать 1969 г. началом истории Internet, когда в США была создана сеть компьютеров ARPANET. Аналогично большинству современных технологий Internet начинался как военная программа, рожденная в недрах Министерства обороны США. Первоначальные замыслы по созданию компьютерных сетей абсолютно не предполагали современную глобализацию. Основная задача, которая решалась на начальном этапе, связана с системами раннего оповещения – средство обеспечения правительственной связи на случай ядерной войны. Решение этой задачи было возложено на Агентство передовых исследовательских проектов (Advanced Research Projects Agency – ARPA) Министерства обороны США. Чрезвычайно плодотворные идеи, заложенные и впоследствии реализованные в проекте, явились следствием популярности этих сетей. Принципиальным при разработке сети считалось то, что сеть не должна быть централизованной, состоять из отдельных сегментов (tatters – лохмотья), каждый узел сети является независимым от остальных узлов и самостоятельно отвечает за прием и передачу сообщений. В основу информационного обмена был положен

принцип коммутации пакетов. В работу над созданием проекта были включены ведущие университеты США (МТИ, КАЛТЕХ и др.). В 1969 г. в ARPA было принято решение объединить суперкомпьютеры оборонных и научных управляющих центров США в единую сеть под названием ARPANET. В 1969 г. эта сеть объединяла четыре компьютера, в 1971 г. – 14, в 1973 г. – 30. Затем процесс приобрел лавинообразный характер. Соразработчиками и узлами создаваемой коммуникационной инфраструктуры стали: Национальный научный фонд (NSF), Министерство энергетики (DOE), Министерство обороны (DOD), Агентство здравоохранения (HHS), Национальное аэрокосмическое агентство (NASA). С 1989 г. объединение этих сетей и принято называть Internet (Arpanet к этому времени вышел из этого объединения). К настоящему времени узлы Internet существуют на всех континентах, точную цифру узлов сетей назвать очень сложно, так как она постоянно изменяется. Европейская часть Internet носит название EUnet. Среди российских сетей крупнейшей является Relcom – Reliable Communications (надежные коммуникации) – полное название EUnet/Relcom. Достаточно популярны также следующие сети:

Sowam Teleport – коммерческая сеть, ориентированная исключительно на коммерческие организации;

FREEnet (The Network For Research Education and Engineering) – некоммерческая сеть, созданная для объединения сетей исследовательских организаций;

RADIO-MSU – тоже некоммерческая сеть, объединяющая физические центры академии наук;

Runet – сеть российских университетов.

GLASNET – неправительственная, некоммерческая сеть, спонсируемая США.

Количество сетей в России увеличивается. Одна из характерных особенностей сетей Internet та, что впервые был технически реализован принцип коммутации пакетов, весьма удачно разработаны в 1982 г. протоколы TCP/IP и их интеграция с операционной системой UNIX.

Несколько слов о протоколах в сетях Internet. Хотя стек протоколов Internet, изображенный на рис. 4.12, не полностью соответствует эталонной модели, расхождения с ней не очень велики.

Здесь следует отметить, что существует два вида стандартов – промышленные стандарты (например, те, которые разрабатывает ISO) и стандарты de facto, к которым можно отнести ряд стандартов ГВС Internet.

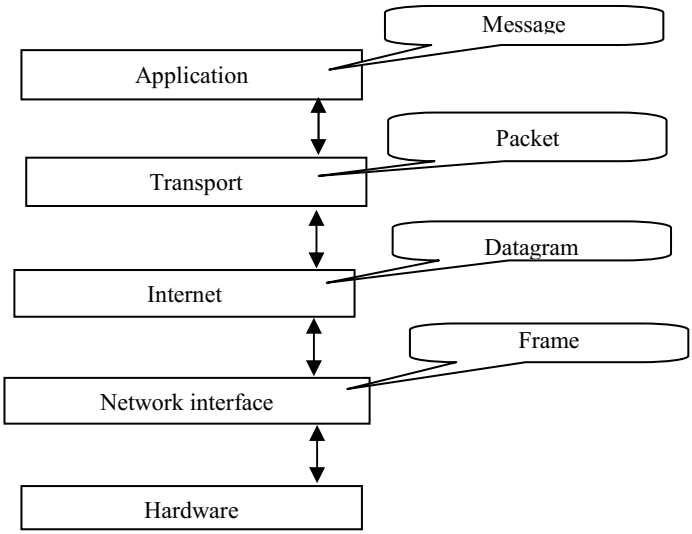


Рис. 4.12. Стек протоколов Internet

Первый уровень Hardware соответствует физическому уровню эталонной модели и так же, как и второй уровень Network interface, соответствующий канальному уровню, ответственному за методы доступа к каналу и логику передачи физической среды (упаковывая пакеты в формат, именуемый кадрами, – FRAME), не является характерным протоколом Internet. Основными называются протоколами TCP/IP третьего и четвертого уровней. Internet протокол (IP), обеспечивающий адресацию и маршрутизацию пакетов (Datagram), соответствует сетевому уровню эталонной модели. Он не гарантирует сохранение порядка и целостности потока пакетов и не различает логические объекты (процессы), порождающие поток информации.

Следующий уровень стека протоколов Transport, или TCP, включает в себя два протокола, также относящихся к основным протоколам Internet, не зависимым от физического канала передачи. Это протокол TCP (Transmission Control Protocol) – протокол

управления передачей, и протокол UDP (User Datagram Protocol) – дейтаграммный протокол передачи данных. Этот уровень соответствует двум уровням эталонной модели: транспортному и сеансовому. Протокол TCP, или протокол с установлением соединения, работает в режиме виртуального канала связи (разбивает сообщение на пакеты при передаче, собирает их в правильном порядке на приеме и отвечает за целостность потока данных) и обеспечивает правила проведения сеанса связи. Протокол UDP работает в дейтаграммном режиме, не контролируя порядок доставки пакетов и их целостность, но в отличие от IP способен различать приложения (которых на одном узле может быть несколько) и передавать информацию к нужным приложениям.

Верхний уровень стека протоколов Internet – протоколы приложений (Application), которые передают транспортному уровню сообщения (Message) в необходимом формате. Протоколы Application включают в себя представительный и прикладной уровни эталонной модели.

Таким образом, основу протоколов Internet составляют протокол сетевого уровня IP и протоколы транспортного уровня TCP и UDP, выбор которых связан с видом приложений и типом используемого канала связи.

Учитывая, что Internet объединяет десятки, а то и сотни миллионов компьютеров, выбор способов адресации является актуальной задачей, определяющей эффективность работы сети. *IP-адрес* – последовательность, состоящая из четырех байтов. Принято каждый байт этой последовательности записывать в виде десятичного числа, разделенного точками, например:

192.171.153.60

Каждое место подсоединения к сетевому интерфейсу имеет свой IP-адрес. Адрес состоит из двух частей: адреса сети и адреса хоста. Под *хостом* понимают любой компьютер или другое устройство (принтер, терминал и т.п.), включенные в сеть. Существует пять классов IP-адресов, рис. 4.13 (ID – Identification Data – идентификатор сети-хоста).

Разбивка сетей на классы A, B, C связана с разделением их по масштабности (глобальности). Количество сетей и хостов в каждом классе адресации представлено в табл. 4.2. При разработке IP-

адресов предполагалось, что эти классы будут использоваться по различному назначению. Классы D и E зарезервированы под специальные нужды. Учитывая, что самые мелкие сети (класс C) имеют возможное количество узлов 254, а в реальных физических сетях их бывает значительно меньше (например, сегмент тонкого Ethernet допускает только 32 узла), важным элементом адресного пространства Internet являются подсети. Схема разбиения адресного пространства сети на подсети представлена на рис. 4.14. Из этого рисунка видно, что один и тот же узел имеет два адреса, соответствующие адресам подсети (каждый интерфейс имеет свой адрес), которые он объединяет. Для адресации подсетей используется часть IP-адресов, закрепленная за номерами хостов.

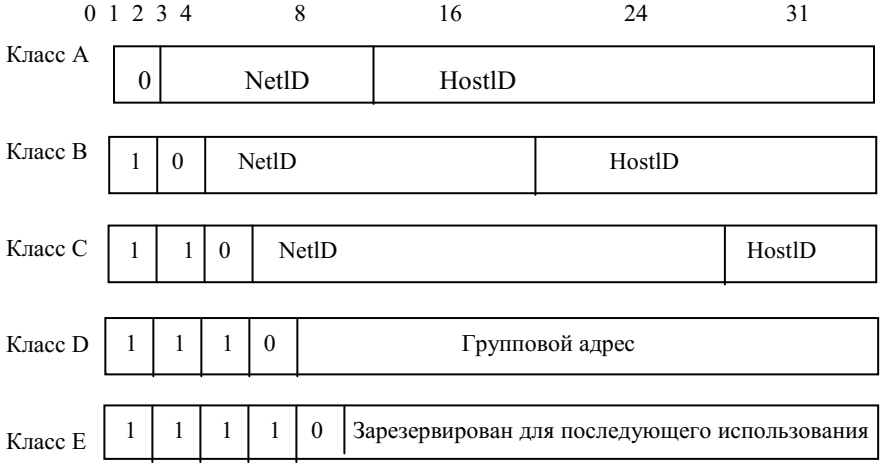


Рис. 4.13. Классы адресов Internet

Числовая адресация удобна для машинной обработки таблиц маршрутов, но она очень неудобна для использования человеком. Эта проблема, а также проблема эффективного распределения огромного количества адресов в Internet решается с помощью *доменной системы имен* (*DNS – Domain Name System*), имеющей иерархическую структуру. Каждому хосту присваивается символьное имя, отражающее наиболее характерные его черты.

Таблица 4.2

Количество сетей и хостов в различных классах адресов Internet

Класс	Диапазон значений первого октета	Возможное количество сетей	Возможное количество узлов
A	1 – 126	126	16777214
B	128 – 191	16382	65534
C	192 – 223	2097150	254
D	224 – 239	–	2^{28}
E	240 – 247	–	2^{27}

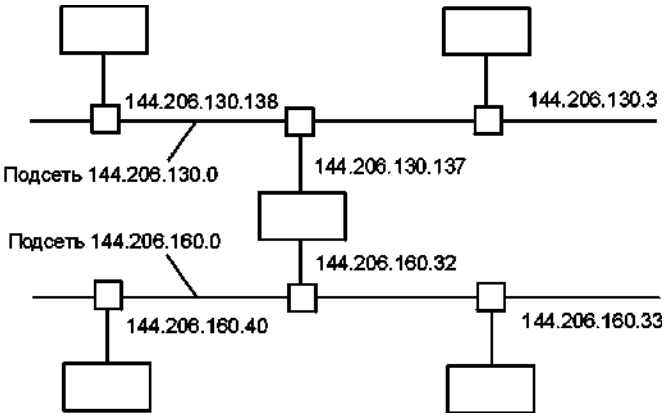


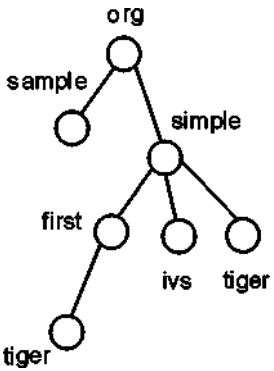
Рис. 4.14. Схема разбиения адресного пространства сети на подсети

Домен представляет собой группу сетей, находящихся под управлением одной организации, например компании или правительственного агентства. Домены имеют иерархическую структуру, т.е. они могут состоять из ряда подчиненных ему доменов. Имена доменов отражают эту иерархическую структуру.

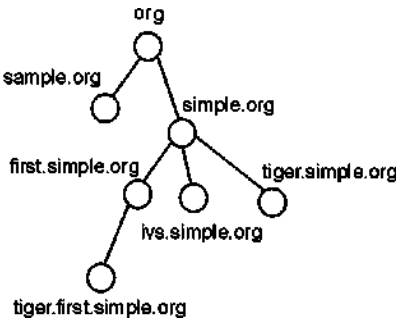
Иерархия доменных имен, аналогичная иерархии файлов, принятой в файловых системах, представлена на рис. 4.15.

Полный адрес формируется справа налево добавлением имен вложенных доменов, разделенных точкой. Для отображения доменных имен на адреса IP существует распределенная база данных

DNS, используя которую узлы могут преобразовывать доменные адреса в численные адреса IP. При этом каждый узел хранит только часть этой базы, определяющей адреса нижнего иерархического уровня. Этим и обеспечивается эффективность поиска.



Пример иерархической доменной системы имен



Полные доменные имена узлов

Рис. 4.15. Иерархия доменных имен

При создании Internet в сети было определено несколько *доменов верхнего уровня*, разделяющих доменные адреса по их принадлежности к различным организациям:

gov – правительственные организации;
com – коммерческие организации;
org – некоммерческие организации;
edu – исследовательские организации и университеты;
net – организации, занимающиеся сетевыми технологиями.

После выхода сети за пределы США в ней появились домены более высокого уровня, принадлежащие различным странам. Представим некоторые из них:

au – Австрия;
fr – Франция;
de – Германия;
ie – Ирландия;
il – Израиль;
it – Италия;

nz – Новая Зеландия;
ru – Российская Федерация;
us – США.

Когда говорят об использовании Internet, речь идет об использовании прикладных программ, поддерживаемых различными прикладными протоколами. Можно выделить две группы таких программ: к первым относятся телекоммуникационные программы, входящие в состав основных операционных систем (ОС) больших и малых компьютеров.

К таким приложениям ОС можно отнести следующие:

Telnet, или эмуляция терминала, обеспечивает подключение вашего терминала к удаленному компьютеру и использование его ресурсов;

FTP (File Transfer Protocol), обеспечивает прием и передачу файлов между компьютерами (протокол FTP также предоставляет серверу средства для идентификации клиента, хотя существует очень большое количество FTP-серверов с анонимным доступом для всех желающих);

E-mail – электронная почта, является одним из самых ранних и наиболее популярных применений Internet (существует очень большое количество почтовых программ с различными интерфейсами и функциональными возможностями).

Появление Internet привело к появлению множества новых приложений, связанных, прежде всего, с распределением информационных ресурсов этой глобальной сети. Среди распределенных информационно-поисковых систем, кроме традиционных, построенных на принципе иерархических каталогов американских систем Gopher и WAIS, следует особо отметить разработанную в 1989 г. сотрудником CERN* Тимом Бернс Ли так называемую всемирную паутину – поисковую информационную систему всемирной распределенной базы данных – распределенную гипертекстовую информационную систему World Wide Web (WWW). Области распространения этой системы стремительно расширяются. Разработана развитая система инструментальных средств подготовки гипертекстовой информации. Существует большое количество программ

* CERN (Conceil Europeenpourla Recherche Nucleaire) – Европейская организация по ядерным исследованиям.

(броузеров) с различными интеллектуальными возможностями для чтения гипертекстовой информации.

В заключение следует отметить, что Internet – не единственная ГВС такого типа.

Довольно широкое распространение получила ГВС BITNET (Because It's time NET-work), объединяющая более 1500 организаций из 52 стран. Европейская часть этой сети называется EARN (European Academic Research Network). Сеть построена на основе стандарта IBM – NJE (Network Job Entry), используемого в операционной системе мейнфреймов IBM – MVS. Одним из наиболее популярных ресурсов BITNET является система почтовых списков LISTSERV. В Internet имеются шлюзы для доступа к этим спискам.

4.6. Локальные вычислительные сети, разновидности протоколов канального уровня, технические средства, используемые ЛВС

Локальные вычислительные сети (ЛВС) начали использоваться с середины 70-х годов в результате падения цен на электронные компоненты, расширения возможностей терминальных устройств, появления интеллектуальных терминалов, а затем и персональных компьютеров. В результате количество различного оборудования, установленного в учреждениях, школах, университетах, заводах и т.п., возросло. Обладатели дешевых компьютеров начали поиски технически дешевых методов их соединения между собой. Это связано с применениями, требующими оперативного использования центрального банка данных (в сбербанках, бухгалтериях, учебных классах и т.п.), коллективным использованием дорогостоящей периферии, такой, как накопители большей емкости, высококачественная печать (лазерные принтеры) и многого другого.

Средства связи, применяемые в глобальных вычислительных сетях, нецелесообразно использовать в данном случае из-за того, что они ориентированы на передачу данных на большие расстояния. Для связи внутри одного здания можно использовать гораздо более дешевые и простые решения.

В то время, как глобальные сети могут быть общедоступными или частными, локальные сети, как правило, принадлежат той же организации, которая соединяет ими свое оборудование. Цель раз-

работчиков ЛВС заключается в нахождении простого и дешевого решения для объединения в вычислительную сеть нескольких десятков компьютеров, находящихся в одном здании. Для упрощения, а соответственно удешевления аппаратных и программных средств разработчики первых локальных сетей остановились на совместном использовании для всех компьютеров единой разделяемой среды (Shared media), используемой компьютерами в режиме разделения времени (TDM – Time Division Multiplexing).

При использовании в локальных сетях очень простых конфигураций (общая шина, кольцо) наряду с положительными имели место и отрицательные последствия, из которых наиболее неприятными были ограничения по производительности и надежности. Наличие только одного пути передачи информации, разделяемого всеми узлами сети, ограничивало пропускную способность сети пропускной способностью этого пути (которая делится, в среднем, на число компьютеров в сети), а надежность сети – надежностью этого пути. Поэтому по мере роста популярности локальных сетей и расширения сферы их применения наметилось движение к отказу от разделяемых сред передачи данных в локальных сетях и переходу к применению активных коммутаторов, в которых конечные узлы присоединяются к индивидуальным линиям связи. В чистом виде такой подход предлагается в технологии ATM, а в технологиях, носящих традиционные названия с приставкой Switched (коммутируемые), Switched Ethernet, Switched Token Ring обычно используется смешанный подход, сочетающий разделяемые и индивидуальные среды передачи данных. Чаще всего конечные узлы соединяются в небольшие разделяемые сегменты с помощью повторителей, а сегменты – друг с другом с помощью индивидуальных коммутируемых связей.

Таким образом, внутри базовых структур по-прежнему работают все те же протоколы разделяемых единственных сред передачи данных, которые были разработаны на начальном этапе появления ЛВС.

Изначально ЛВС имели следующие характерные особенности, отличающие их от ГВС:

работает только на ограниченной территории (как правило, это территория одного здания или рабочего участка, а ее протяженность от нескольких сотен метров до километра);

скорость передачи данных первоначально составляла 1 – 20 Мбит/с, в современных ЛВС она равна 100 – 200 Мбит/с, а в некоторых ЛВС доходит до $10^4 – 10^5$ Мбит/с;

основной передающей средой на начальном этапе являлись витая пара и коаксиальный кабель, а в последнее время все большее распространение получает оптический кабель, а также беспроводные каналы связи;

к основным топологиям, используемым в ЛВС (*топология* в контексте компьютерной сети означает способ соединения друг с другом оконечных систем или станций), относятся шина, кольцо и звезда, а также суперпозиция этих схем в виде дерева и т.п. (рис. 4.16);

использование метода коммутации пакетов при полном отказе (в отличие от ГВС) от коммутации каналов;

оперативность выполнения запросов – время прохождения запросов в ЛВС гораздо меньше ГВС.

При всем различии ЛВС и ГВС по мере развития этих сильно отличающихся на начальном этапе сетевых технологий происходит явное их сближение, которое уже привело к значительному их взаимопроникновению.

Одним из проявлений этого сближения является появление сетей масштаба большого города (MAN), занимающих промежуточное положение между локальными и глобальными сетями и имеющих ряд характерных особенностей как тех, так и других.

Появление новых технологий, использование оптической (цифровой) немодулированной передачи данных по оптоволоконным линиям в ГВС значительно повысило их надежность, упростило техническую реализацию, избавило от сложных и избыточных процедур обеспечения корректности передачи и позволило обеспечить скорости передачи данных, сравнимые и даже превышающие скорости современных ЛВС.

В то же самое время (как уже отмечалось) локальные сети перенимают у глобальных транспортные технологии. Все новые скоростные ЛВС поддерживают работу по индивидуальным линиям связи наряду с традиционными для ЛВС разделяемыми линиями.

В локальных сетях также большое внимание уделяется методам защиты информации от несанкционированного доступа. Такое внимание обусловлено тем, что ЛВС перестали быть изолирован-

ными, чаще всего они имеют выход в «большой мир» через глобальные связи.

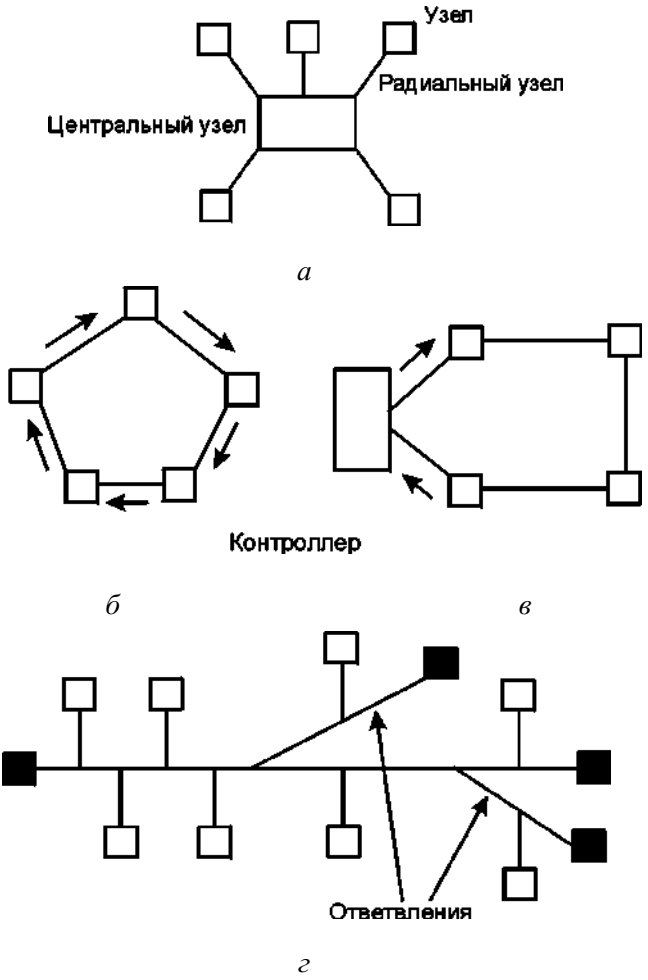


Рис. 4.16. Варианты топологии локальных вычислительных сетей

И, наконец, появляются технологии, изначально предназначенные для обоих видов сетей. Наиболее ярким представителем нового поколения технологий является технология ATM.

Процесс переноса служб и технологий из глобальных сетей в локальные приобрел такой массовый характер, что появился даже специальный термин *intranet-технологии* (intra – внутренний), обозначающий применение служб внешних (глобальных) сетей во внутренних (локальных).

Методы доступа в ЛВС. Методы доступа в ЛВС, входящие в состав протоколов канального уровня, определены рядом стандартов IEEE*. Наибольшее распространение получили ЛВС, использующие технологию Ethernet (802.3), реализованную в 1975 г. фирмой Xerox. Этот метод доступа опробован еще раньше в радиосети Алоха, о которой уже упоминалось.

В зависимости от типа физической среды (и топологии) стандарт IEEE 802.3 имеет различные модификации: 10 Base-5 – «толстый коаксиал»; 10 BASE-2 – «тонкий коаксиал» с шинной топологией; 10 Base-T – неэкранированная витая пара; 10 Base-F – оптоволокно; 100 Base-T, F – витая пара, оптоволокно с топологией «звезда».

При передаче двоичной информации по кабелю различных типов Ethernet обеспечивает пропускную способность 10 Мбит/с в полудуплексном режиме (за исключением 100 Base-T, F).

Принятый в 1995 г. стандарт Fast Ethernet обеспечивает скорость 100 Мбит/с, а принятый в 1998 г. стандарт Gigabit Ethernet предполагает скорости 1000 Мбит/с. Оба работают в дуплексном режиме.

В 2002 г. принят стандарт, поддерживающий скорость передачи 10 Гбит/с (10^4 Мбит), а в 2009 г. предполагается принятие стандарта 40 – 100 Гбит/с (некоторые фирмы уже в 2008 г. начали выпуск оборудования для подобных сетей Ethernet).

Все виды стандартов Ethernet используют один и тот же метод доступа (разделение среды) CSMA/CD (Carrier Sense Multiply Access with Collision Detection) – случайный множественный доступ с контролем несущей и разрешением коллизий. Принцип доступа аналогичен сети Алоха: используется один канал (моноканал) для всех станций, хотя в Ethernet в качестве канала служит кабель, который может использоваться всеми в любой момент, когда это

* IEEE (Institute of Electrical and Electronic Engineers – Институт инженеров по электротехнике и электронике) – организация, разрабатывающая и публикующая стандарты, непосредственно связанные с физическим уровнем и, в частности, с топологией сетей.

потребуется (случайным образом). Так, на рис. 4.17 три станции независимо, случайным образом обращаются к каналу. Но в отличие от простой сети Алоха станция, прежде чем занять канал для передачи своей информации, прослушивает его на предмет занятости. Таким образом, станция 2 занимает канал только после того, как станция 1 его освобождает. Однако при одновременном (или почти одновременном) обращении к каналу двух станций, когда сигнал не успевает распространиться по линии и станции не регистрируют занятости канала, возможно столкновение и искажение пакетов. В этом случае вступает в действие механизм разрешения столкновений (коллизий). Он обнаруживает столкновение и обрывает передачу. Это и представлено на рис. 4.17, где произошла коллизия между станциями 2 и 3. Обе станции ждут в течение короткого случайного интервала времени, а затем повторяют передачу. Этот метод доступа обеспечивает случайное временное уплотнение каналов и имеет очень высокую эффективность их использования.

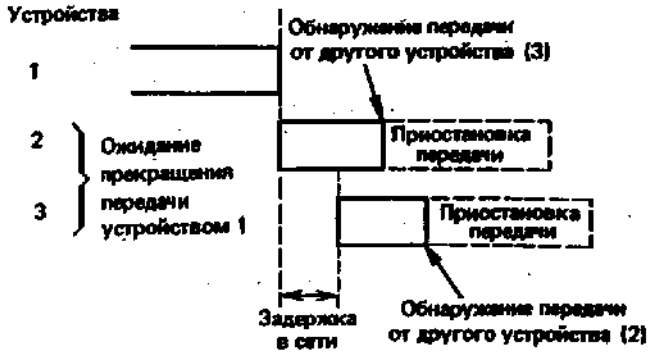


Рис. 4.17. Метод доступа CSMA/CD

В зависимости от типа спецификации количество рабочих станций в сети лежит в диапазоне от 30 – 10 Base-2 (тонкий Ethernet) до 1024 – 10 Base-T (неэкранированная витая пара), максимальное расстояние между узлами сети 925 м (10 Base-5, 10 Base-F – толстый коаксиальный или оптический кабель).

Широкое распространение получили также сети с *маркерным методом доступа* (Token Ring, ArcNet и FDDI). Принципиально они очень близки, так же как и в Ethernet они имеют общую, разде-

ляемую среду передачи данных, замкнутую, как правило, в кольцо (за исключением Arc Net) и представляющую общий ресурс. Метод доступа к среде не является случайным (как Ethernet) и основан на передаче станциям права на использование среды передачи в определенном порядке. Это право передается с помощью кадра (импульса) специального формата, называемого маркером или токеном (token). Среди сетей с подобным методом доступа наиболее распространена разработанная в 1984 г. фирмой IBM технология Token Ring, на ее основе в 1985 г. был принят стандарт 802.5. Принцип работы Token Ring представлен на рис. 4.18.

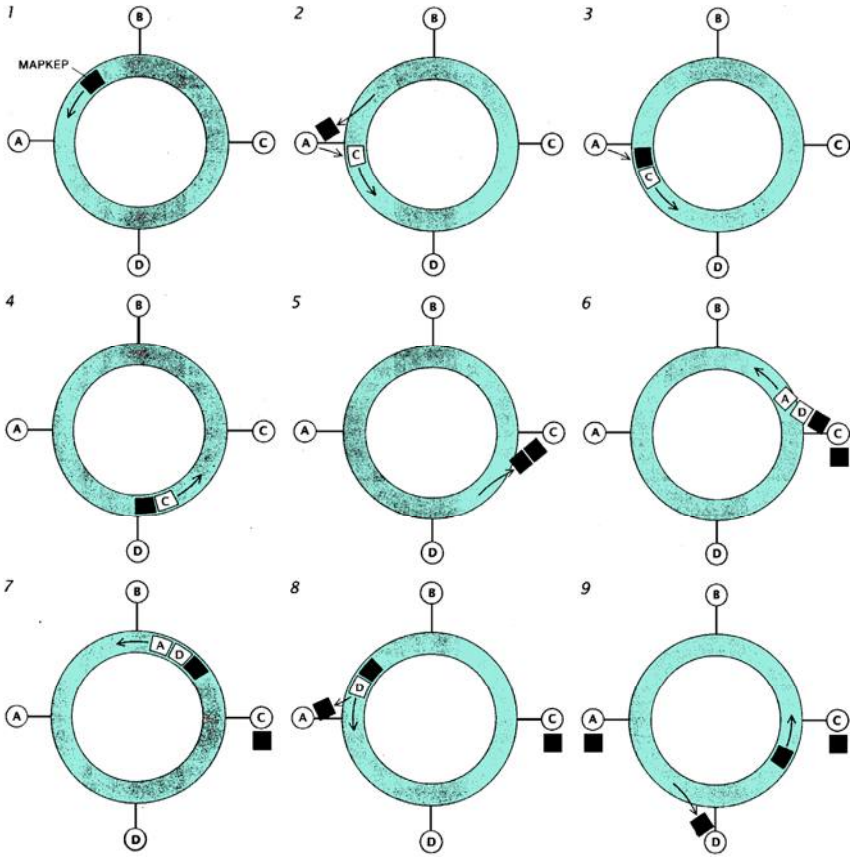


Рис. 4.18. Принцип работы ЛВС Token Ring

Маркерное кольцо – распространенный вариант локальной сети. Доступом к сети управляет циркулирующая по кольцу последовательность бит, называемая маркером. Чтобы послать сообщение, станция должна сначала дождаться прихода маркера, удалить маркер из кольца, направить в кольцо пакет с адресом (или последовательность пакетов) и в конце снова направить в кольцо маркер. Станции могут удалять адресованные им пакеты, сохраняя в кольце маркер. Показано, как станция A посылает сообщение станции C, которая получает его, а затем посылает сообщение станциям A и D.

В этих сетях скорость передачи достигает 16 Мбит/с, максимальное количество станций – 96, максимальная длина – 120 м.

С середины 90-х годов начался интенсивный переход от ЛВС начального уровня со скоростями 10 – 20 Мбит/с к ЛВС со скоростями 100 Мбит/с и выше. Структура стандартов IEEE 802.x представлена на рис. 4.19.

Первая высокоскоростная сеть на оптическом кабеле FDDI (Fiber Distributed Data Interface) была разработана в 1988 г. Это кольцевая сеть (двойное кольцо) протяженностью 100 км с маркерным методом доступа со скоростью до 100 Мбит/с. По своим размерам такие сети выходят за рамки общепринятых ЛВС, приближаясь к ГВС. На начальном этапе эта сеть не имела широкого распространения ввиду высокой стоимости сетевого оборудования, однако с появлением варианта этой сети на витой паре (CDDI), а также отработкой технологии оптических интерфейсов она активно внедряется.

Начиная с середины 90-х годов начался массовый переход традиционных ЛВС на скорости 100 Мбит/с. Это уже упоминавшийся стандарт Fast Ethernet, который по многим параметрам протоколов и технических средств совместим с начальным Ethernet, что облегчает переход к новой сети при сохранении ряда старых компонентов. Конкуренцию этой сети составляла технология 100 VG-AnyLAN, которая обеспечивала скорость 100 Мбит/с и поддерживала оба, самых распространенных протокола – Ethernet и Token Ring.

Тем не менее, большая часть локальных сетей поддерживает протокол Ethernet, который к 2008 г., как уже упоминалось ранее, достиг скорости передачи 40 – 100 Гбит/с. Зависимость максимальной длины Gigabit Ethernet от типа передающей среды представлена на рис. 4.20.

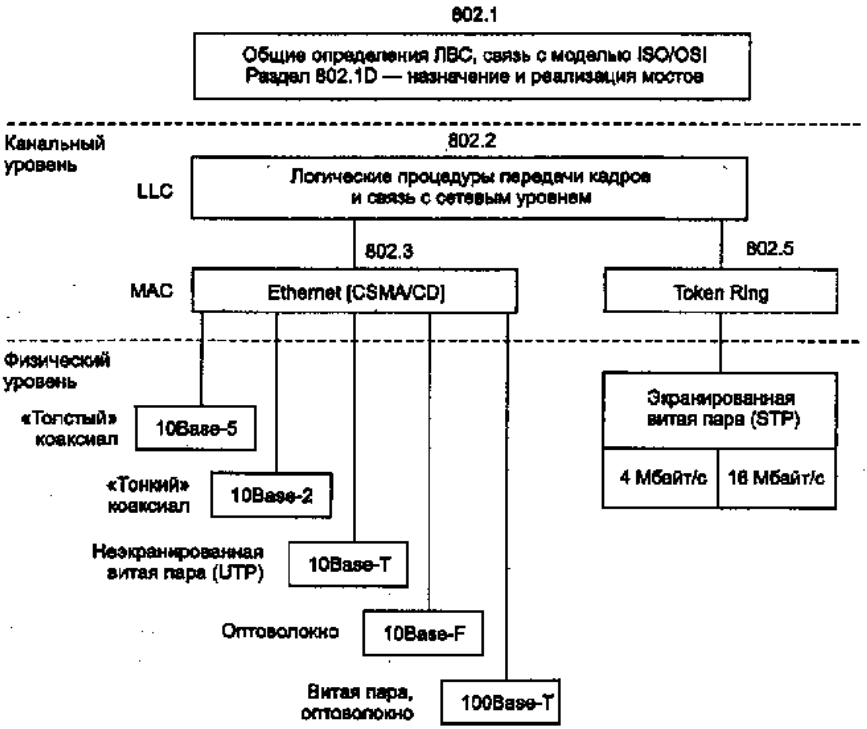


Рис. 4.19. Структура стандартов IEEE 802.x

Обсуждая производительность ЛВС, необходимо хотя бы кратко упомянуть о технологии Fibre Channel (FC), которая изначально использовалась в основном для подключения высокоскоростной периферии к большим компьютерам и передает данные по волоконнооптическому и медному кабелю с максимальными (по сравнению с обычными ЛВС) скоростями. Так в 90-х годах, когда максимальная скорость Ethernet (Fast Ethernet) была 100 Мбит/с, скорость канала FC доходила до 3 Гбит/с.

Одним из главных преимуществ FC является то, что он предоставляет ряд вариантов среды и скорости передачи, а также топологии сети. В табл. 4.3 приведены доступные комбинации физического носителя и скорости передачи данных для сетей FC (данные на 2000 г.).

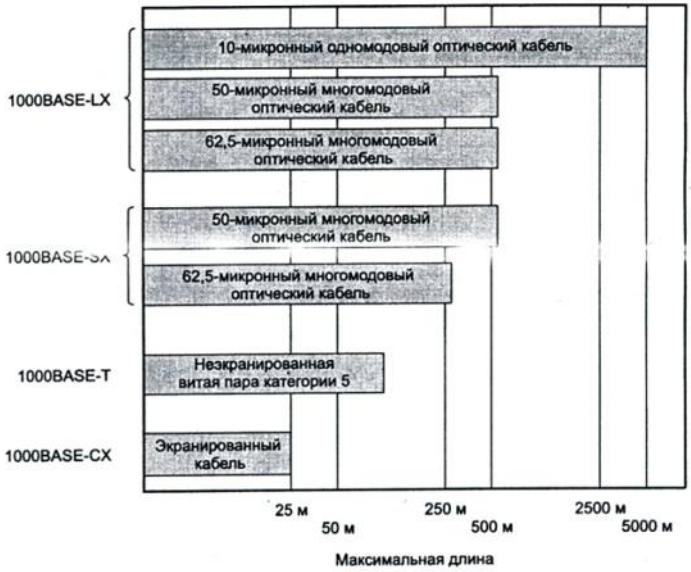


Рис. 4.20. Варианты для физических носителей для Gigabit Ethernet

Таблица 4.3

Максимальная длина физических носителей в сети Fibre Channel

Скорость передачи	100 Мбит/с	200 Мбит/с	400 Мбит/с	800 Мбит/с	1,6 Гбит/с	3,2 Гбит/с
Одномодовый оптический кабель	–	10 км	10 км	10 км	2 км	2 км
Многомодовый оптический кабель 50 мм	–	2 км	1 км	500 м	500 м	175 м
Многомодовый оптический кабель 62,5 мм	–	1 км	1 км	175 м	–	–
Коаксиальный видеокабель	100 м	100 м	71 м	50 м	–	–
Миниатюрный коаксиальный кабель	42 м	28 м	19 м	14 м	–	–
Экранированная витая пара	80 м	57 м	46 м	28 м	–	–
Twinax	93 м	66 м	46 м	33 м	–	–

Структура технических средств ЛВС. 1. Передающая среда, используемая в ЛВС, представлена тремя типами: витая пара, коаксиальный кабель и оптический кабель.

Витая пара – наиболее распространенный и дешевый вариант канала, традиционно используемый в телефонии. Этот вид каналов в наименьшей степени защищен от помех и возможностей несанкционированного доступа. Существует пять категорий кабеля на основе витой пары, различающихся по электротехническим и высокочастотным характеристикам. Так, витая пара пятой категории используется в высокоскоростных ЛВС. Данные кабели могут быть экранированными и неэкранированными. Стоимость высококачественного кабеля на основе витой пары достаточно высока.

Коаксиальный кабель используется, как правило, в сетях Ethernet, имеет лучшие высокочастотные характеристики и помехозащищенность по сравнению с витой парой, однако переход на быстрые протоколы Ethernet связан с переходом его на витую пару или оптический кабель.

Оптический кабель сравнительно новая и наиболее перспективная передающая среда, значительно превосходящая по своим коммуникационным свойствам, рассмотренным выше, другие передающие среды. В то же самое время она имеет достаточно сложную структуру и требует более детального описания.

Волоконно-оптические коммуникации были практически созданы в середине 60-х годов прошлого столетия на базе двух ключевых компонентов: создания твердотельных источников излучения света и получения чистого стекла. Эти работы были основаны на более ранних экспериментах, открывших принцип световода. В основе его – свойство неограниченного распространения света в воде и других средах.

Развитие этой технологии идет по пути создания более мощных источников излучения и уменьшения примесей в стекле (современное оптическое волокно в десять тысяч раз прозрачнее оконного стекла).

Другим фактором, повлиявшим на эволюцию волоконно-оптической линии связи (ВОЛС), стала разработка производительных приемников, принимающих световой сигнал в большом динамическом диапазоне.

ВОЛС осуществляется посредством трех главных компонентов: оптического кабеля, оптического трансивера (передатчика) и приемника оптического излучения. Сначала электрические сигналы поступают на вход трансивера, который преобразует их в световые

импульсы и направляет в волокно. Импульсы света принимаются на другом конце приемником и вновь преобразуются в электрический сигнал. Чем чаще передаются импульсы, тем больше пропускная способность канала.

Оптическое волокно состоит из сердцевины и оболочки с разными показателями преломления n_1 и n_2. Сердцевина имеет больший показатель преломления. Таким образом, световой сигнал благодаря внутреннему преломлению или отражению не покидает оптически более плотной среды (рис. 4.21). По диаметру сердцевины волокно подразделяется на одно- и многомодовое.

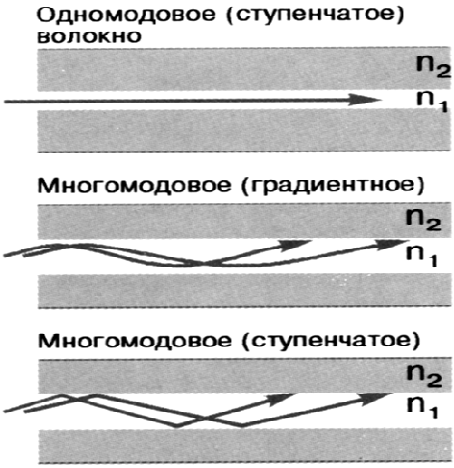


Рис. 4.21. Распространение света в волокне

В одномодовом волокне диаметр световодной жилы порядка 8 – 10 мкм, т.е. сравним с длиной световой волны. При такой геометрии в волокне может распространяться только один луч (одна мода). В многомодовом волокне размер световодной жилы порядка 50 – 60 мкм, что делает возможным распространение большого количества лучей (много мод). Многомодовое волокно, в свою очередь, подразделяется на градиентное, имеющее градиентный профиль показателя преломления световой области с максимумом на оси, и ступенчатое, имеющее постоянный показатель преломления сердцевины (табл. 4.4).

Имеются две основные характеристики оптоволокна, определяющие максимальное расстояние между станциями:

затухание определяется потерями на поглощение и на рассеяние излучения в оптическом волокне (затухание связано с длиной волны излучения, вводимого в волокно, потери на поглощение зависят от чистоты материала, потери на рассеяние определяются неоднородностями показателя преломления материала);

дисперсия характеризует зависимость скорости распространения сигнала от длины волны вводимого излучения (поскольку источники света (светодиоды или лазеры) излучают некоторый спектр длин волн, дисперсия приводит к уширению импульсов при распространении по волокну и, тем самым, порождает искажение сигналов).

Таблица 4.4

Основные характеристики волокна

Характеристика	Многомодовое		Одномодовое
Профиль показателя преломления	Ступенчатый	Градиентный	Ступенчатый
Длина волны излучения, мкм	0,85 – 1,3	0,85 – 1,3	1,3 – 1,55
Диаметр сердцевины, мкм	50 – 1000	50 – 60	4 – 10
Диаметр оболочки, мкм	125 – 1050	124 – 140	75 – 125
Композиционный состав серцевина/оболочка	Стекло/Стекло, стекло/пластик, пластик/пластик	Стекло/Стекло	Стекло/Стекло
Полоса пропускания, МГц · км	100	500	2000 и выше

Одномодовое волокно обладает самыми лучшими характеристиками по полосе пропускания и затуханию. Однако чтобы реализовать преимущества одномодового волокна, необходимо использовать дорогостоящие источники излучения и другое вспомога-

тельное оборудование. Само одномодовое волокно также существенно дороже многомодового*.

Многомодовое волокно более удобно при монтаже, на него рассчитаны доступные и дешевые излучатели, но оно обладает гораздо большим затуханием и меньшей полосой пропускания. В связи с этим многомодовое волокно вполне приемлемо для локальных сетей связи, но недостаточно для магистральных линий.

Оптические передатчики подразделяются на светоизлучающие диоды и лазеры. Первые – сравнительно дешевые, имеют большой срок службы и применяются в многомодовых линиях связи ввиду невысокой мощности излучения света и слабой фокусировки. Вторые лишены последних недостатков и используются в одномодовых линиях связи, однако их стоимость существенно выше.

В состав *приемного устройства* ВОЛС входят фотоприемник и электрическая цепь.

Среди основных преимуществ ВОЛС необходимо отметить следующие:

широкая полоса пропускания, которая для многомодового волокна более чем на порядок превышает полосу пропускания витой пары, а для одномодового это превышение более чем на два порядка;

---

* Если сравнивать по стоимости информационные сети на витой паре (медный провод) при длине сети не более 100 м и скорости передачи 1 Гбит и многомодовое волокно протяженностью 550 – 2000 м (в зависимости от типа волокна) при такой же скорости передачи, что стоимость одного рабочего места памяти приблизительно равна 100 дол, а на оптоволокне 160 – 180 дол. Стоимость активного оборудования (передатчики, приемники) для медного провода составляет приблизительно 60 дол, для оптоволокна 250 дол (данные на 2008 г.). Магистральные сети и сети с повышенными требованиями к качеству передающего изображения на одномодовом волокне стоят приблизительно в 50 – 100 раз выше (речь идет только о стоимости сетевого оборудования).

В оптоволоконных системах используются источники света двух типов: светоизлучающие диоды (Light-Emitting Diode, LED) и инжекционные лазерные диоды (Injection Laser Diode, ILD). Эти полупроводниковые устройства излучают свет при приложении к ним электрического поля. Светодиоды дешевле, работают в большем диапазоне температур, у них больший срок службы, но значительно дороже и применяются в многомодовых линиях ввиду невысокой мощности излучения света и слабой фокусировки. Инжекционные лазерные диоды работают по принципу лазера, они более эффективны и позволяют достичь больших скоростей передачи данных.

большие расстояния между станциями, так для одномодового волокна расстояние между станциями составляет 50 – 60 км;

высокая помехозащищенность – его нечувствительность к электрическим помехам, возможность прокладки линий вблизи мощных и высоковольтных электрофизических устройств;

гальваническая развязка элементов сети, т.е. волокно обладает изолирующим свойством, отсутствует потребность в заземлении;

взрыво- и пожаробезопасность – отсутствие искрообразования, позволяет использовать его на химических и нефтеперерабатывающих предприятиях при обслуживании технологических процессов повышенного риска.

Истинные возможности оптического волокна проявляются в том случае, если по одному волокну передается множество лучей с разными частотами. Это является одной из разновидностей частотного мультиплексирования (Frequency-Division Multiplexing, FDM), обычно, однако, используется термин *спектральное уплотнение*, или *мультиплексирование* по длинам волн (Wavelength-Division Multiplexing – WDM). В таком случае свет, распространяющийся по оптоволокну, состоит из лучей различных цветов, или длин волн, каждый луч передает данные отдельного канала. В 1997 г. специалисты Bell Laboratories продемонстрировали систему со спектральным уплотнением, работающую с сотней лучей, каждый из которых поддерживал скорость передачи данных 10 Гбит/с. Общая пропускная способность этой линии составляет 1 триллион битов в секунду (1 Тбит/с). В настоящее время доступны коммерческие системы с 80 каналами с пропускной способностью 10 Гбит/с.

2. Компьютеры, включенные в сеть, различаются по своему функциональному назначению:

рабочие станции – индивидуальные компьютеры пользователей;

серверы – компьютеры коллективного использования различного функционального назначения.

Ресурсы серверов доступны рабочим станциям и разделяются между ними. Наиболее распространенными ресурсами являются дисковые файлы, базы данных, прикладные программы, устройства печати и т.п.

3. Сетевые контроллеры (адаптеры) выполняют функции аппаратуры канала данных (АКД): обеспечение интерфейсных функ-

ций, буферизацию, формирование пакетов, параллельно-последовательные преобразования, кодирование/декодирование, доступ к кабелю и т.п.

4. В средства объединения сетей входят:

повторители (Repeater) – используются для физического соединения различных сегментов кабеля локальной сети с целью увеличения общей длины сети;

концентраторы (Hub) – применяются для объединения отдельных сегментов сетей в единую разделяемую среду;

коммутаторы (Switch) – могут применяться вместо концентратора для повышения производительности сетей (например, в Ethernet), переключая сообщение на нужный сегмент;

мосты (Bridge) – предназначаются для объединения или разъединения (с целью уменьшения трафика) сетей, имеющих одинаковые протоколы верхнего уровня (аппаратно-независимые), а также для объединения сетей с различной передающей средой;

маршрутизаторы (Router) – служат для определения оптимального маршрута сообщений между различными узлами сети и сетевыми сегментами;

шлюзы (Gateways) – обеспечивают работу сетей с различными протоколами верхнего уровня.

Возможны устройства, совмещающие несколько функций, например *гибридные маршрутизаторы* (brouter) – гибрид моста и обычного маршрутизатора.

Тип ЛВС определяется, главным образом, методами доступа (или протоколами нижнего уровня). На среднем уровне выбор протокола зависит в основном от применяемой операционной системы. Так, в сетях, работающих с ОС UNIX, используется рассмотренный ранее TCP/IP. В популярной, сравнительно недавно, сетевой ОС Novell Net Ware применяются протоколы IPX/SPX, а в NETBIOS, как правило, – сетевые операционные системы фирмы Microsoft.

IPX (Internetwork Packet Exchange) – *протокол межсетевой передачи пакетов,* соответствует транспортному уровню OSI.

SPX (Sequenced Packet Exchange) – *протокол последовательного обмена пакетами*, соответствует сетевому и сеансовому уровню OSI.

NETBIOS (Network Basic Input/Output System) – *протокол сетевой базовой системы ввода-вывода*, разработанный фирмой IBM.

В заключение следует отметить, что «жизнь» вычислительных сетей очень динамична, а их классификация неоднозначна. Изначальные представления, что глобальные сети – медленные и используют традиционные телекоммуникации (средства связи), а локальные – значительно быстрее, используя выделенные линии и единую среду передачи данных, изменились. Об этом свидетельствует появление высокоскоростных глобальных сетей с включением выделенных каналов, а с другой стороны, значительное усложнение локальных сетей с использованием характерных протоколов глобальных сетей и разработка новых технологий, изначально предназначенных для обоих видов сетей. К таким технологиям относится, прежде всего, АТМ, которая может служить основой не только локальных и глобальных компьютерных сетей, но и телефонных сетей, а также широковещательных видеосетей, объединяя все виды трафика в одной транспортной сети.

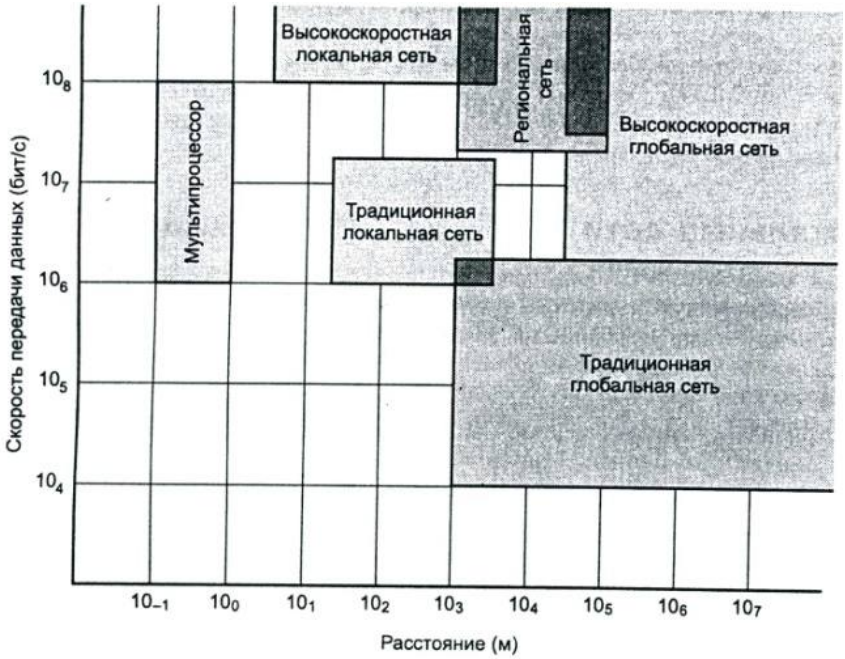


Рис. 4.22. Сравнение мультиплексорных систем, а также локальных, региональных и глобальных сетей

Кроме того, появились так называемые региональные сети, занимающие нишу между глобальными и локальными. В основном, рынок региональных сетей состоит из клиентов, которым необходима высокоскоростная передача данных на средние расстояния. Региональная сеть должна предоставлять требуемую пропускную способность с меньшими издержками и большей эффективностью, чем местная телефонная компания, используя метод коллективного использования высокоскоростного физического носителя, применяемого в локальных сетях.

И, наконец, очень специфическую, но типичную локальную сеть представляют суперкомпьютеры, о которых говорилось ранее и использующих быстрые коммуникационные каналы, традиционные для ЛВС.

Ориентировочные характеристики рассмотренных выше (традиционных) вычислительных сетей представлены на рис. 4.22.

Особое место среди вычислительных сетей занимают так называемые беспроводные сети, которые будут рассмотрены в дальнейшем.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Назовите четыре типа информации, передаваемой по сетям. Почему за последнее время выросла популярность оптического волокна?

2. Укажите основные принципы, достоинства и недостатки централизованной и распределенной обработки данных.

3. Определите функциональное назначение основных типов коммуникационного оборудования – повторителей, концентраторов, мостов, коммутаторов, маршрутизаторов, шлюзов.

4. Укажите основные методы повышения пропускной способности каналов передачи данных. В чем разница между мультиплексированием и сжатием (компрессией) информации?

5. Что такое протокол в мире коммуникаций? Какова разница между протоколом и интерфейсом? Каковы причины многоуровневого подхода к сетевым протоколам? Укажите различия между протоколами TCP/IP и эталонной моделью.

6. Укажите разницу между аналоговыми и цифровыми данными, сигналами и каналами. Какие типы данных можно передавать по аналоговым и цифровым каналам?

7. Укажите среды, используемые для передачи данных. В чем преимущества и недостатки разновидности сред, используемых глобальными и локальными сетями передачи данных?

8. Назовите методы доступа в локальных и глобальных сетях передачи данных, их достоинства и недостатки. Какова структура канала связи?

ГЛАВА 5. ЛИНГВИСТИЧЕСКИЕ СРЕДСТВА КОМПЬЮТЕРОВ

5.1. Основные тенденции в развитии языков программирования

Под *языками программирования* понимается система формальных обозначений для точного описания абстрактных структур данных и алгоритмов программ. Так же, как и любые языки, языки программирования имеют алфавит, словарь, синтаксис (правила конструирования выражений), семантику – разъяснение смысла конструкций из слов и выражений.

Внешняя форма языка программирования, т.е. запись текста программы, устанавливается с помощью синтаксиса языка, который определяет формальный язык. Синтаксис описывается с помощью определенных грамматических правил. Задание формального языка хотя и достаточно для установления синтаксической корректности, однако недостаточно для того, чтобы понять ее функцию и способ действия программы, которые определяются путем задания семантики, через установление значения отдельных языковых элементов.

Следует различать язык программирования и его реализацию. Реализация языка – программа, которая преобразует запись высокого уровня (более приспособленную для восприятия человеком) в последовательность машинных команд компьютера и входит в состав программного обеспечения компьютера.

Появление компьютеров и интенсивное развитие их технических возможностей сопровождается развитием их лингвистического обеспечения. Среди основных причин, стимулирующих развитие языков программирования, следует отметить следующие:

необходимость повышения эффективности труда программистов, уменьшение уровня абстрагирования – приближение методов программирования к приложениям;

развитие приложений – внедрение компьютеров в самые различные сферы человеческой деятельности;

значительное увеличение ресурсов компьютеров и их производительности;

нарастающее увеличение сложности и размерности задач, решаемых на компьютерах.

Разработка языков программирования началась практически одновременно с появлением вычислительных машин. Это связано с тем, что программирование в машинных кодах – рутинный длительный процесс, который сдерживал внедрение вычислительной техники, значительно снижая эффективность ее использования.

В 1945 г. немецкий математик Конрад Цузе изобрел систему записи, которую назвал «Планкалкюль» (что означает программное исчисление), но язык не был реализован, тем не менее, многие развитые в нем идеи нашли отражение в современных языках. История развития традиционных языков высокого уровня, т.е. языков, ориентированных не на коды компьютеров, а на алгоритм решения задачи, представлена на рис. 5.1.

В 1949 г. Дж. Мочли, одним из создателей первого универсального компьютера ЭНИАК, был разработан Short code – Ассемблер, реализованный на этой машине.

Несомненно, наибольшее влияние на все последующие языки программирования оказал Фортран (FORTRAN – FOR(mula) TRANS(lator)), разработанный Дж. Бэкусом и его коллегами из фирмы IBM в 1954 – 1957 гг. и реализованный на компьютере IBM 704. В те годы язык был встречен скептически, однако разработка эффективного транслятора для самых популярных (по тем временам) в мире компьютеров позволило языку завоевать всеобщее признание у программистов, занимающихся научными расчетами. Приблизительно в то же самое время Грейс Мюррей Хоппер – офицер ВМФ США (из фирмы UNIVAC) – разработала более простой язык коммерческих задач Флоу-Матик – прообраз Кобола. Очень важное место среди языков занимает Алгол (ALGO(rythmic)L(anguage)), первая версия которого была разработана в Европе международным комитетом на основе идей, заложенных в Фортране и Планкалкюле. Алгол хотя и не приобрел такой популярности, как Фортран, но оказал очень большое влияние на дальнейшее развитие языков. В начале 60-х годов появился Лисп и была разработана нотация APL.

Стремительное размножение языков встревожило программистов, поэтому было решено разработать единый язык для всех. Однако попытки создать такой язык закончились неудачей.

Частичный успех языка PL/1, разработанного фирмой IBM, показал, что всеобъемлющий язык должен быть труден в изучении и

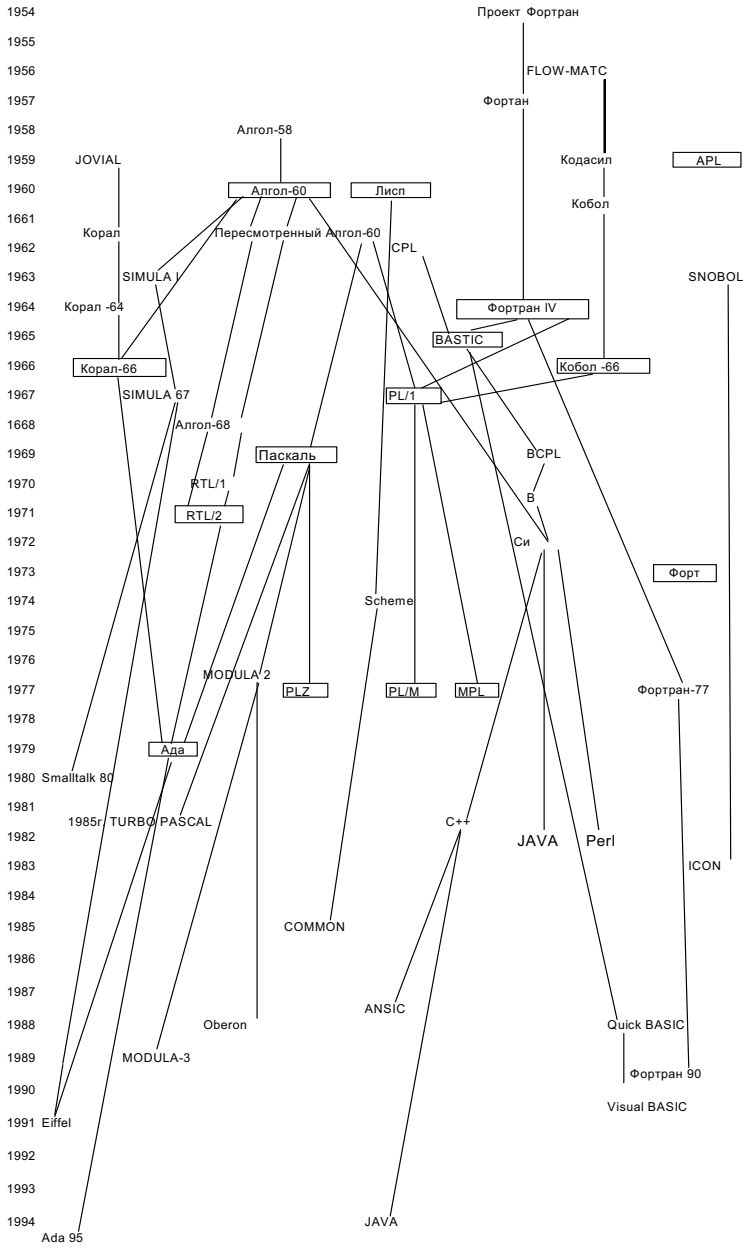


Рис. 5.1. История развития языков программирования

громоздок в реализации. Было признано, что необходимость в новых языках для новых областей будет появляться всегда. Все языки, о которых говорилось выше, можно определить как процедурные (или предписывающие). Программа на таком языке указывает, как получить результат и что для этого необходимо сделать. Еще одно изначальное название этих языков – языки высокого уровня или алгоритмические. Несколько слов об уровне языка, который определяется степенью абстрагирования от решаемой задачи (или как говорят от приложений). Самым нижним уровнем языков являются машинные коды. Языки высокого уровня ориентированы на программирование алгоритмов. Но алгоритм – также некоторая абстракция, удаленная от конкретного приложения и требующая адаптации к конкретной задаче.

Стремление подняться на новый более высокий уровень программирования и приблизиться к приложениям связано с так называемыми непроцедурными или описательными языками, которые можно назвать языками сверхвысокого уровня, и они приобретают все большее значение. Описательная программа констатирует, какой результат желателен, не указывая, как этого достичь. В программе скорее формулируются отношения, а не последовательность вычислений. Таким образом, программист освобождается от обязанности разрабатывать шаги алгоритма и определять их порядок. Одним из разновидностей таких языков является Пролог PRO(gramming) in LOG(ic), представляющий собой усовершенствование языка Лисп (LISP – L(ist) I(nformation) S(ymbol) P(rocessing)), а также Визикальк и Мультиплан, получившие распространение на персональных компьютерах.

Некоторые из этих языков называют также дескриптивными. Программа на дескриптивном языке состоит из двух частей: базы данных и цели. База данных содержит конкретные факты о решаемой задаче и основные принципы, касающиеся соотношений между этими фактами. Цель – общее описание решения. Определяя цель, программист, по существу, просит компьютер скомбинировать факты и соотношения так, чтобы можно было доказать, что цель истинна. Дескриптивные языки особенно популярны в области разработки экспертных систем.

Другая тенденция в развитии языков программирования – рост интереса к системам записи, называемым объектно-ориентирован-

ными языками, например Симула 67 (SIMULA – SIMU(lation) LA(nguage)). Кроме этого появление многопроцессорных систем привело к необходимости разработки языков параллельных вычислений. Примером таких языков могут служить Оккам, Компед (Computer Parallel) и др. Развитие языков тесно связано с развитием технических средств, с появлением новых информационных технологий, совершенствованием программных интерфейсов. Это относится к средствам программирования графических программных оболочек, программирования сетевых приложений, совершенствованию лингвистических возможностей операционных систем и т.п.

В настоящее время насчитывается несколько тысяч языков и, тем не менее, продолжается разработка новых. Это, по-видимому, связано с тем, что нет хорошего способа для определения качества языка. При всем многообразии языков широко используются лишь некоторые. От возникновения языка до его «успешной стандартизации» (стандартизация языков программирования проводится организацией ANSI – American National Standard Institute) проходит примерно 10 лет, поэтому программисты предпочитают старые языки, несмотря на постоянную критику со стороны энтузиастов-разработчиков новых языков. Это связано с тем большим заделом программного обеспечения, который существует у традиционных языков, а также новыми их версиями, в которых продолжается дальнейшее совершенствование с учетом критических замечаний (не всегда объективных). Поэтому характерной тенденцией последнего времени в развитии традиционных языков высокого уровня является не столько создание новых языков, сколько включение новых функциональных возможностей в традиционные языки – использование элементов машинной графики, объектная ориентация, создание графических программных интерфейсов средства распараллеливания и т.п. Проблемы параллельного программирования, использования виртуальной памяти в значительной степени решаются не только на уровне прикладного программирования, но также средствами системных программ (операционных систем, трансляторов и т.п.).

Среди очень большого количества языков, разработанных к настоящему времени, очень важное значение имеют языки, которые хотя и не получили широкой применительной практики, но оказали сильное влияние на все дальнейшее развитие лингвистического

обеспечения информатики. К таким языкам можно отнести: ALGOL (международный комитет), Симула 67 (Кристен Нигаард, Оле Йохан Дал), Small Talk (Ален Кэй), Modula, Pascal (Object Pascal), Oberon (Никлаус Вирт).

В целом, сравнение языков очень сложная задача, поскольку существует 2570 различных возможностей или свойств для сравнения, к которым добавляются 12 характеристик трансляторов. Следовательно, не существует самого лучшего языка программирования, так же как не существует самого лучшего естественного языка. Выбор языка определяется целями применения и индивидуальностью программиста. Как сказал король Карл V [17]: «Я разговариваю по-испански с богом, по-итальянски с женщинами, по-французски с мужчинами и по-немецки со своей лошадью».

В заключение необходимо отметить, что проблемы лингвистического обеспечения (как и программирования) сдерживают внедрение средств вычислительной техники и успехи в решении этих проблем значительно скромнее темпов развития технических средств.

5.2. Классификация и краткая характеристика современных языков программирования

Все языки программирования можно сгруппировать по некоторым признакам в ряд подгрупп (с определенной степенью условности). На рис. 5.2 представлен один из вариантов такого разбиения.

Из рисунка видно, что на сегодняшний день языки можно разделить на две большие подгруппы: машинно-ориентированные и машинно-независимые. К первым относят языки, ориентированные на систему команд конкретного типа ЭВМ, и, естественно, программы, написанные на этих языках, не могут выполняться на машинах с иной системой команд. Их основу составляют машинные языки. Программирование в кодах машины, при котором каждая команда представляется цепочкой из нулей и единиц, а на программиста возлагается задача расположения своей программы в адресах компьютера – сложный утомительный процесс. Программа, загруженная в память, выполняется без всякой предварительной обработки, так как вся работа по переводу алгоритма в программу, готовую к исполнению (исполнимую программу), осуществляется «вручную» человеком.

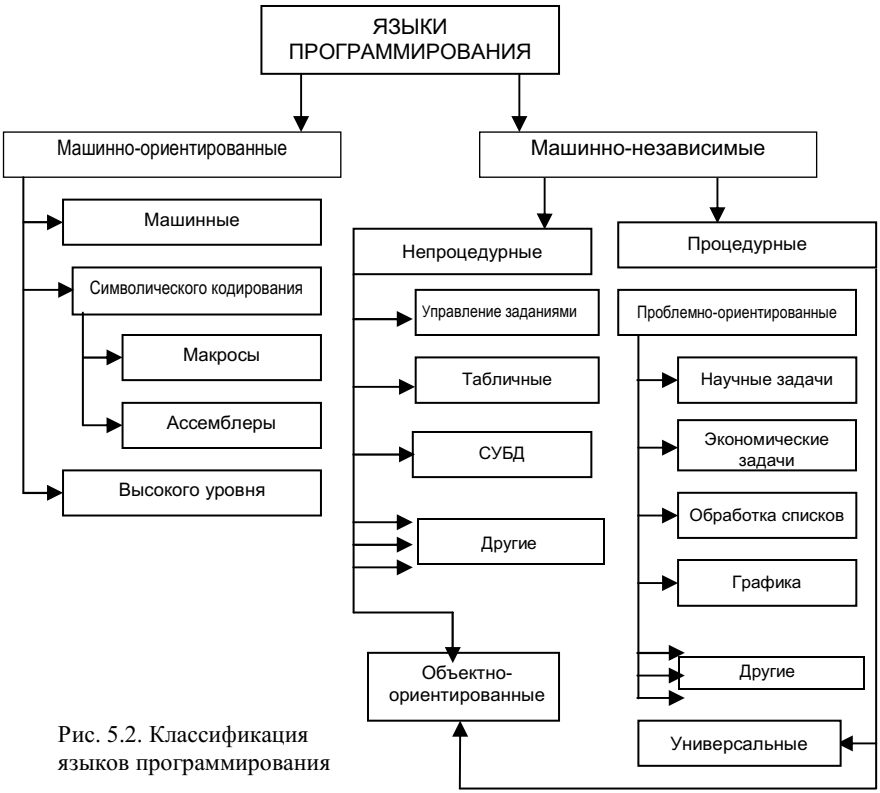


Рис. 5.2. Классификация языков программирования

Программирование в машинных кодах в настоящее время не используется программистами (за исключением некоторых узкоспециализированных приложений). Однако программирование на Ассемблере (Assembly System) практикуется достаточно широко, поскольку все системное программирование, программирование нестандартных периферийных устройств, программирование процессов реального времени и т.п. в той или иной мере связано с использованием Ассемблера. Язык Ассемблера – первый шаг по автоматизации программирования, избавлению программиста от мелочей двоичного кодирования. Использование символов, отражающих суть машинной команды (например, LOAD – загрузить, ADD – сложить), вместо соответствующих им кодов команд, а также

запись программ без привязки к конкретным адресам ЭВМ значительно облегчают работу программиста. При этом часть работы возлагается на компьютер, который с помощью программы транслятор-ассемблер переводит текст программы на машинный язык и определяет его место в поле оперативной памяти. Дальнейшее совершенствование Асемблеров связано с появлением макроассемблеров (или макросов), позволяющих писать более компактные программы. Это объясняется тем, что в макроассемблерах часто повторяющиеся последовательности команд Ассемблера объединяются и им присваиваются новые символы. Транслятор с языка Ассемблера является необходимым атрибутом любого современного компьютера, так как программирование на этом языке позволяет в полной мере использовать ресурсы вычислительных систем. В то же время использование этого языка предполагает доскональное изучение технических средств и архитектурных особенностей программируемой машины.

Появление языков высокого уровня вызвано стремлением упростить процедуру программирования, создать средства, отражающие структуру алгоритма задачи и не требующие от программиста детального изучения компьютера. Разработка машинно-зависимых систем автоматизации экспериментов, изготовление программ реального времени, зависящих от окружающей среды и времени, а также программирование нестандартных периферийных устройств означает наивысший уровень разработки программ. Тестирование изготовленных систем методически менее уточнено и трудно реализуемо. В то же время ошибки, остающиеся в готовых системах, вызывают очень тяжелые последствия. Машинная зависимость языков программирования реального времени высокого уровня хотя и затрудняет мобильность («переносимость») программ, но, безусловно, улучшает характеристики выполнения. В то же время мобильность программ управления не является актуальной задачей, так как эти системы аппаратно ориентируются на отдельные классы ЭВМ.

Если во многих применениях сравнительно рано были созданы языки высокого уровня, каждый из которых стал мировым стандартом в своей области, то ни одному из языков реального времени высокого уровня не удалось стать таковым. В Великобритании необходимость в языке программирования задач в реальном масштабе времени привела к появлению языка Корал (сокращение от «язык про-

граммирования для реализации радиолокационных систем»), построенного на идеях Алгола. Корал – один из немногих языков данного типа, на который имеется стандартное определение.

В США использовался Фортран реального времени (разработка университета Purdue), во Франции – язык Прокол (также на основе Фортрана), в Германии – Пеарл. Программирование задач реального времени для микропроцессорных систем привело к появлению таких языков высокого уровня, как PL/M, PLZ, MPL, предназначенных для семейств микропроцессоров фирм Intel, Zilog и Motorola соответственно. У этих языков было много приверженцев и они получили высокую оценку пользователей. Особое место в списке языков реального времени занимает Форт: он не имеет аналогов в машинно-независимых языках и позволяет сочетать программирование на языках высокого и низкого уровней.

Машинно-ориентированные языки высокого уровня применялись и при системном программировании – разработке операционных систем и инструментов программной поддержки компьютерных систем. Так, фирма IBM для этих целей использовала языки PLS (диалект PL/1), DEC-BLISS (на базе ALGOL), Burroughs – Extended ALGOL. В настоящее время в качестве основного инструмента для создания операционных систем используется язык С.

Следующую категорию составляют машинно-независимые языки высокого уровня. Они, в свою очередь, могут быть разделены на непроцедурные (дескриптивные или описательные) и процедурные. Процедурные языки, представляющие основу языков современного программирования, составляют наиболее солидную группу, и ими пользуется наибольшее число программистов для решения различных прикладных задач, связанных с расчетами, проектированием, экономикой и т.п. Процедурные языки можно условно разделить на проблемно-ориентированные и универсальные. К первым относятся такие традиционные языки научного и инженерного программирования, как Фортран и Алгол.

К этой же категории можно отнести Паскаль, который вначале применялся как язык для обучения программированию, но благодаря появлению микросистем стал одним из наиболее популярных языков решения инженерных задач. В экономических приложениях своеобразным стандартом являлся Кобол (Common Business Oriented Language).

Специализация языков связана с развитием технических средств, расширением разновидностей приложений и появлением новых информационных технологий. Так, совершенствование языков научно-технических приложений привело к появлению языков машинной графики, совершенствование программных интерфейсов связано со средствами разработки графических интерфейсов в среде Windows. Развитие глобальных вычислительных сетей привело к появлению средств разработки сетевых приложений.

Среди основных направлений специализации языков можно выделить следующие.

Языки подготовки сценариев. Первый такой язык – Sh (Shell – оболочка), изначально состоявший из набора команд, интерпретируемых как вызов системных программ, выполняющих служебные функции (например, управление файлами) и их простую фильтрацию. Затем на этой основе были добавлены переменные операторы потока управления, функции и т.п., в результате получили универсальный язык программирования, разработанный в Bell Lab., – AWK.

Разработанный Лари Уоллом язык Perl изначально составлял комбинацию Sh и AWK, затем получил значительное развитие и сейчас является достаточно мощным, выходящим за рамки сценариев, особенно в связи с развитием WWW, языком. Он очень удобен для программирования интерфейса CGI (Common Gatway Intertace – общий шлюзовой интерфейс).

Программирование в среде Windows. Создание графических оболочек связано с такими инструментальными средствами, как Turbo Vision, VB, VBA, VC++, Delphi, .NET Vision.

Программирование в компьютерных сетях. Программирование связано с рядом специфических требований, предъявляемых к разрабатываемым программам, а следовательно, и к инструментальным средам. Основные из этих требований:

платформенная независимость;

защита информации от несанкционированного доступа;

обеспечение высокого быстродействия.

Для программирования активных Web-серверов в наибольшей степени удовлетворяют перечисленным требованиям язык Java (C-), разработанный фирмой Sun, а также платформа .NET(DOT NET), включающая язык C# фирмы Microsoft.

Для разметки гипертекста, создания Web-сценариев (Web-дизайна) применяются: HTML(H(yper) T(ext) M(arkup) L(anguage)), XTML, Perl, Java Script, VB Script, используемые для программирования со стороны клиента, для программирования серверной части сценариев используется ASP (Active Server Pages).

Основные языки для работы с данными. Таким языком в рамках стандарта ANSI является SQL (Structured Query Language – язык структурированных запросов).

Среди узкоспециализированных языков, которых за последние 50 лет появилось великое множество, можно отметить: APG, используемый для генерации деловых отчетов; APT, созданный для управления программируемыми устройствами; GPSS, разработанный для моделирования больших систем (включая большие информационные системы) и т.п.

Однако узкоспециализированные языки, как правило, вызывают неприятие у большинства программистов. Поэтому основной подход к расширению возможностей программирования на сегодняшний день связан с включением в традиционные, наиболее популярные языки дополнительных средств, расширяющих их функции. Таким образом, исходный язык становится некоторым подмножеством группы новых языков.

Практически все языки научно-технического применения обеспечены средствами программирования графического вывода. Общим для всех языков являются средства программирования в среде Windows (Visual Basic, Delphi, Visual C т.п.), создание новых типов данных – объектно-ориентированное программирование.

К универсальным языкам на начальном этапе относили, прежде всего, PL/I (Programming Language I), CPL (Combined PL), APL, которые, хотя и используются, не получили широкого распространения.

Основным, всеми признанным де-факто, стандартом универсального языка стал разработанный в недрах Bell Laboratory крупнейшего американского концерна AT&T язык Си (C). Примечательно, что разработчики (отнюдь, не представляющие ведущие компьютерные фирмы) не ставили своей целью разработку универсального языка. Они создавали инструментальные средства для написания переносимой на различные компьютеры операционной системы UNIX (Кен Томпсон и Денис Ритчи). На первом этапе этот язык считался языком системного программирования. В 1980 г.

Бьярн Страуструп разработал объектно-ориентированное расширение этого языка – Си ++.

В дальнейшем различными фирмами созданы средства программирования графических интерфейсов в рамках языка Си. Язык сетевого программирования Java также создан на основе языка Си.

В 1979 г. было объявлено о языке Ада, разработанном для министерства обороны США с его задачами реального времени. Однако до настоящего времени этот язык не вошел в практику широкого применения.

Необходимо отметить определенную условность приведенной классификации. Машинно-независимые языки не отвечают такому названию в полной мере, так как при реализации языка на конкретных компьютерах фирмы вводят дополнительные возможности, выходящие за рамки стандартов на тот или иной язык. Однако программа, написанная в рамках стандартных определений, как правило, может выполняться на компьютерах любого типа, имеющих соответствующий транслятор.

Следует отметить, что чем ближе язык к прикладным задачам пользователей, тем большую работу должен выполнять компьютер по его реализации, т.е. при повышении эффективности работы программиста понижается эффективность работы компьютера, так как один и тот же алгоритм, закодированный в машинных кодах и написанный на языке высокого уровня, использует различный ресурс компьютера. Во втором случае программы, естественно, выполняются медленнее и занимают значительно больше оперативной памяти.

Однако некоторые непроцедурные языки возникли задолго до появления персональных компьютеров и связаны со стремлением исключить рутину процедурного программирования и приблизить язык к приложениям. И хотя эти языки не получили массового распространения, они включают ряд новых идей, полезных для дальнейшего совершенствования лингвистического обеспечения информатики.

К таким языкам относятся функциональные языки (другое название – *аппликативные*). Типичным представителем этих языков является LISP (L(ist) I(nformation) S(ymbol) P(rocessing)) – язык обработки списков, разработанный Дж. Маккарти в 1961 г. для символьной обработки. Потребность в языках, работающих с символами, ощущалась в лингвистике, философии, математике. У психологов необходимость в таких средствах связана с моделированием

человеческого мышления, у лингвистов – с обработкой текстов на естественном языке, у математиков – это язык доказательства теорем, дифференцирование, интегрирование и т.п. Кстати, такие математические пакеты, как Macsyma Reduce, Scratchpad, разработаны с помощью LISP.

К непроцедурным языкам относятся и языки логического программирования, которые являются декларативными. Разработка программы на этих языках заключается в представлении программы в виде символьной логики и использовании для получения результата процесса логического вывода (алгебра Буля – исчисление предиктов, см. гл. 1).

Логическое программирование является не процедурным, а декларативным. Это означает, что в них указывается лишь описание желаемого результата, а не детальная процедура его получения. Одним из представителей таких языков является разработанный на базе LISP язык PROLOG – PRO(gramming) in LOG(ic).

Оба указанных выше языка используются при разработке систем искусственного интеллекта.

Развитие непроцедурных языков связано со стремлением разработчиков исключить прикладное программирование, т.е. применение традиционных языков программирования, которые не являются естественными языками программирования ни для компьютеров, ни для человека, так как они ориентированы на алгоритмы и требуют адаптации и к компьютеру, и к пользователю компьютера.

Непроцедурные языки в последнее время очень широко распространены. К ним относятся ставшие уже традиционными языки управления заданиями (средствами общения с компьютером), входящие в состав операционных систем, и средства подготовки текстов. Интерес к персональным компьютерам и их интенсивное внедрение в деловую практику самых разнообразных организаций связан с появлением электронных таблиц и локальных баз данных (значительно уменьшивших интерес к Коболу), в состав которых также входят непроцедурные лингвистические средства. Математические пакеты, системы автоматизированного проектирования, экспертные системы и т.п. также имеют лингвистические средства такого типа.

Сегодня в полной мере задачу повышения уровня языка решить не удается, поскольку конкретные приложения пользователей значительно шире возможностей, предоставленных стандартными

средствами. Практически все непроцедурные языки имеют средства традиционного программирования. Так, пакеты прикладных программ, разработанные фирмой Microsoft, имеют в своем составе Visual Basic. В то же самое время в языки LISP и Prolog при их практическом применении включаются элементы традиционных (императивных) языков.

5.3. Структура и сравнительные характеристики процедурно-ориентированных (императивных) языков программирования

Важнейшим из факторов, влияющих на разработку языков программирования, является архитектура компьютера. Большинство популярных языков последних 40 лет разрабатывалось на основе архитектуры, которая используется практически во всех современных компьютерах, названной по имени одного из ее авторов Джона фон Неймана и описанной в гл. 2. Эти языки программирования называются императивными.

Главным элементом императивных языков программирования, основанных на неймановской архитектуре компьютера, являются переменные, которые моделируют ячейки памяти: операторы присваивания, основанные на операциях пересылки данных, а также итеративная форма повторений, являющаяся наиболее эффективным методом в этой архитектуре. Операнды выражений передаются из памяти в процессор, а результат вычисления выражений возвращается в ячейку памяти, представляемую левой частью оператора присваивания.

Поскольку команды и данные хранятся в соседних ячейках, то операции на компьютере выполняются быстро. Стремление повысить уровень программирования, исключив влияние архитектуры и приблизив его к конечным приложениям с помощью входных языков пакетов прикладных программ, функциональных языков или в языках логического программирования, в большинстве случаев реализовано и получило широкую применительную практику. Однако использование этих средств в полной мере потребовало включение в их состав основных элементов императивных языков.

В связи с этим по-прежнему актуально понимание основных принципов и структуры этих языков. Именно эти языки являются основным инструментом большинства программистов, занимающихся прикладным программированием и использующих либо

FORTRAN в рамках научно-технических задач, либо C со всеми его расширениями (C^{++}, C builder, C# и т.п.), который позволяет опуститься на более низкий уровень и осуществлять как прикладное, так и системное программирование. Определенной популярностью пользуется и Pascal (особенно в рамках среды Delphi).

Несмотря на постоянное развитие и определенные различия в императивных языках (иногда существенных), определяющих сферу их приложения, они имеют ряд общих принципов, которые и будут рассмотрены далее.

Так же, как и любые языки, о чем уже говорилось, языки программирования имеют алфавит, словарь, способ конструирования выражений (синтаксис), а также разъяснение смысла конструкций из слов и выражений (семантику). В отличие от других языков назначение языка программирования состоит в хранении и передаче упорядоченной последовательности инструкций или алгоритмов. Словарь их значительно меньше, чем у естественных, а выражения точны и однозначны.

Все популярные языки программирования используют латинский алфавит, ряд специальных символов, которые имеются на клавиатуре, а также английские слова и выражения (real, integer, if, then, else, do и т.п.), представляющие собой ключевые, или зарезервированные слова. Существуют некоторые различия в ключевых и зарезервированных словах.

Ключевые слова (Keyword) используются в Фортране и имеют свое функциональное предназначение, находясь, как правило, в неисполняемой части программы (описательной). Так, указание REAL L в начальной части программы указывает, что значение переменной L – действительное. В то же самое время слово «REAL» в исполняемой части может рассматриваться в некоторых языках как обычный идентификатор, например для обозначения арифметического выражения.

Зарезервированные слова (reserved word) могут использоваться только в контексте, предусмотренном правилами (семантикой) языка.

Ряд языков (например, ADA) могут содержать так называемые *предопределенные слова*, значения которых могут переопределяться пользователями.

Программы оперируют с объектами данных. *Объект данных* – комбинация данных, атрибутов, описывающих их свойства, и методов, раскрывающих их поведение.

Объектам данных, а также программам или отдельным их частям и т.п. сопоставляются имена (или идентификаторы), правила образования которых несколько различны в разных языках. Все языки имеют достаточно традиционный набор объектов данных – типов данных, встроенных в трансляторы языков. Все типы данных разделяются на две категории – на константы, не изменяемые в процессе работы программы, оформленные по определенным правилам, и переменные. Например, Фортраном константы определяются следующим образом:

```
PARAMETER (MILE=5280, FOOT=12, METER=39.36)
           имя   значение
```

а в Паскале –

```
CONST
MILE=5280;
FOOT=12;
METER=39.36;
```

Наиболее характерными типами данных, включаемых в рассматриваемые языки, являются следующие.

1. Элементарные типы данных – типы данных, не определяемые через другие типы и существующие практически во всех императивных языках:

числовые типы:

целые – Integer, действительные – REAL (с фиксированной и плавающей точкой);

комплексные (COMPLEX) и с двойной точностью (DOUBLE PRECISION) – только в Фортране;

булевские, или *логические*, *типы* – простейшие из всех типов, они имеют, прежде всего, значения – TRUE и FALSE (0, 1 для C);

символьные типы (литерные или CHAR), запоминаются с помощью цифрового кодирования, ввиду недостаточности восьмибитного кодирования в ASCII-кодах для всемирного обмена разработан 16-разрядный Unicode, в котором содержится большинство символов языков мира (первым языком, использующим Unicode, является Java).

2. Массивы (array, dimension), структурированные типы данных – однородное множество данных, в котором каждый элемент идентифицируется его положением по отношению к первому элементу.

К менее распространенным типам, которые используются в ряде языков, можно отнести: порядковые типы, символьные строки (Character strings), запись (record), множества (set type), указатели (Pointer) и т.п.

Все языки имеют три типа выражений.

Арифметические выражения задают порядок действий над элементами данных и состоят из операндов (константы, переменные и т.п.), круглых скобок и знаков операций, традиционных (с очень небольшими отличиями) для всех языков +, -, *, /. Для Фортрана и Бейсика в стандарте предусмотрены знаки возведения в степень – **, ^. Арифметические операции могут быть унарными (unary), содержащими один операнд – «-U», бинарными (binary), содержащими два операнда – a + b, и тернарными (ternary, три операнда) в языках C, C^{++} и Java.

Логические выражения (булевские) состоят из операндов (констант, переменных, элементов массивов и т.п.), логических операций (not, and, or и т.п.) и операций отношения, которые приводятся ниже для языков Бейсик, Фортран и Паскаль. Операции отношения выполняют сравнение двух операндов, а с помощью логических операций составляются более сложные логические выражения. Результатом выполнения логических выражений являются значения True или False.

Символьные выражения (литерные) порождают значения, имеющие литерный тип данных.

Разделители – элементы любого языка программирования, предназначенные для разделения отдельных элементов выражений внутри строки, для разделения строк, отдельных фрагментов программ и т.п. В качестве разделителей используются пробелы, точки, запятые, двоеточия, точки с запятой, разнообразные скобки и т.п.

Операторы в алгоритмических языках могут быть:

а) простыми (наиболее характерным оператором этого вида является оператор присваивания, он предписывает выполнить выражение, заданное в его правой части, и присвоить результат переменной, идентификатор которой расположен в левой части);

б) ввода-вывода, в качестве которых наиболее часто используются термины read, write, print;

в) структурными, или составными, которые представляют собой структуры, построенные из других операторов по строго определенным правилам (к таким операторам относятся условные операторы и операторы повтора).

Ниже представлены разновидности структурных операторов и логических отношений для языков Бейсик, Фортран и Паскаль.

Условные операторы:

Бейсик	Фортран	Паскаль
=	.eq.	=
<>	.ne.	<>
>	.gt.	>
<	.lt.	<
>=	.ge.	>=
<=	.le.	<=
and	.and.	and
or	.or.	or

Операторы цикла:

```
Бейсик                                  Фортран                          Паскаль
10 if i>1 then i=0                      if (i  .gt.  1) i=0              if i>1 then i:=0;
20 if i>1 then i=0 else i=–1              if (i  .gt.  1) then             if i>1
                                          i=0                                i:=0
                                          else                                 else
                                          i=–1                                 i:=–1;
30 if i>1 then i=0: j=2 else i=–1       if (i  .gt. 1) then              if  i>1 then
                                            j=0                            begin
                                            j=2                                 i:=0;
                                          else                                  j:=2
                                          i= –1                            end
                                                                             end if
                                                                             else
                                                                               i:=–1
```

Циклы:

```
Бейсик                                  Фортран                          Паскаль
10 for  i = 1 to n                        do 10 i=1, n                   for i:=1 to n begin
20       x = x+1                                x=x+1                       x:=x+1
30 next i                                10 continue                     end;
10 while i >10                             do 10 idum = 1,10000          while i < 10 do
20          i=i+1                          if (i  .ge.  10) go to 100    begin
30 wend                                      i =i+1                            i:= i + 1
                                         10 continue                     end;
                                        100 continue
10 while idum = 0                          do 10 idum = 1,10000          repeat
20          i = i + 1                      i = i + 1                             i:= i  + 1
30 if i>=10 then goto 50                   if (i  .ge.  10) go to 100    until i >= 10;
40 end                                   10 continue
50 'имитация цикла repeat – until'      100 continue
```

Указанные языковые конструкции являются общими для всех традиционных (императивных) языков программирования. На этом уровне различия в языках нельзя считать принципиальными. В качестве примера использования перечисленных средств программирования (как начального уровня) представлены тексты программ на языках FORTRAN, PASCAL и C, и блок-схема программы определения максимального и минимального элементов массива (рис. 5.3).

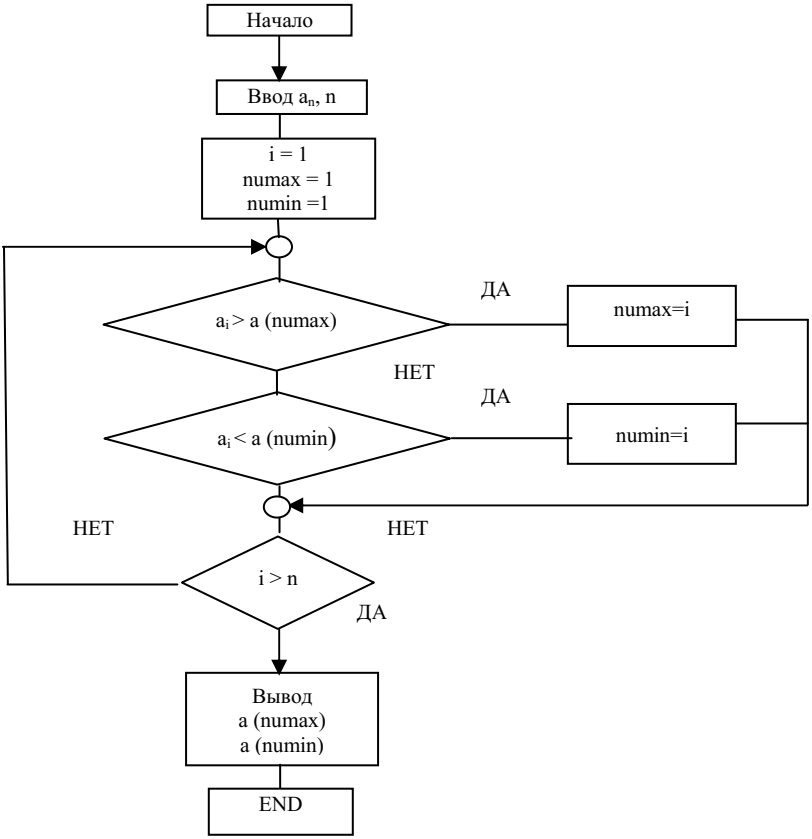


Рис. 5.3. Блок-схема программы, определяющей максимальный и минимальный элементы массива

Фортран:

```
PROGRAM MINMAX
real a(100)
integer n, numax, numin, i
print  *,' input n'
read *,n
do 10 i= 1, n
print *,  'input a( '.i.')'
read *,a(i)
10  continue
numax=1
numin=1
do 20 i=2, n
if (a(i).gt.a(numax) ) numax= i
  if (a(i).lt.a(numin ) ) numin=i
20  continue
      print *, ' max{a(i)}=',a(numax),' min{a(i)}=' ,a(numin)
      end
```

Паскаль:

```
program extrem( input, output);
var
n , i , nuimax, numin : integer ;
 a: array [ 1..100 ] of real;
begin
writeln( ' input n ') ;readln(n) ;
for i:=1 to n do
begin
writeln( ' input a('.j.' ) ');readIn (a [ i ] );
end ;
numax: = 1; numin : = 1;
for i: =2 to n do
begin
  if a[ i ]> a[numax] then numax:=i;
if a[ i ]< a[numin] then numin :=i;
end;
writeln(' max{a(i)}=',a[numax],' min{a(i)}=' ,a[numin]);
end.
```

Си:

```
# include<stdio.h>
int n , i ,numax, numin ;
float, a[100];
main( )
```

```
{
   printf("input n\n" ) ;
   scanf(″ %d ″, &n );
   for( i = 0; i <n; i++)
   {
          printf("input a(%d)\n", i ); scanf("%g",&a[i ]);
   }
numax=numin=0;
for( i=1;i<n;i++)
{
    if (a[ i ] > a[numax])numax=i ;
    if (a[ i ] < a[numin] )numin=i;
}
printf( "max{a( i ) }=%g  min{a( i)}=%g″ ,a[numax] ,a[numin] ) ;
}
```

В основе создаваемых языков, как правило, лежит некоторая основополагающая идея или, как говорят теоретики программирования, *парадигма*.

Рассмотренные средства составляют начальный уровень программирования, основная идея которого характеризуется как *хаотическое программирование*, типичным представителем этого направления был Бейсик. Программа на Бейсике представляла единый текст с большим количеством переходов к помеченным участкам (с помощью меток), и работа велась в режиме интерпретации (построчного выполнения). Однако появление сложных технических систем приводит к увеличению объема обрабатываемых данных, повышению сложности обработки, значительному увеличению размеров программ и появлению программных комплексов. Увеличение размеров программ довольно быстро выявило ограниченность этого подхода, связанного с невозможностью распараллеливания работы между программистами и сложностью диагностики ошибок.

Первый шаг в решении этой задачи был сделан разработчиками Фортрана, что связано с развитием так называемого процедурно-ориентированного, или модульного, программирования. Концепция этого направления в развитии программирования была предложена М. Уилксом в 1957 г. и заключалась она в следующем.

Программу можно составлять из отдельных автономных функциональных модулей, транслируемых независимо, которые могут вызываться многократно и использоваться различными програм-

мами (рис. 5.4). Модули, собранные по определенной тематике – математические, графические, экономические и т.п., объединяются в библиотеки.

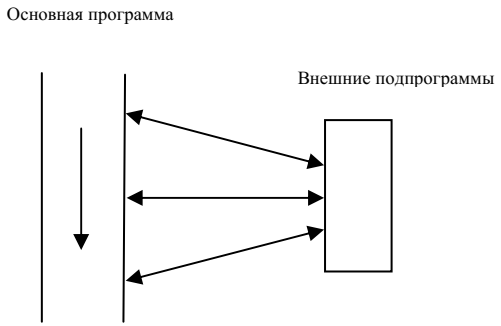


Рис. 5.4. Схема работы программы, многократно использующей внешний модуль

На первом этапе такие модули в объектном виде включались в виде внутренних библиотек в транслятор языка и подключались на уровне компоновки в исполнимую программу. К ним относились продпрограммы – функции для вычисления наиболее распространенных математических функций (тригонометрических, логарифмов, MIN, MAX и т.п.). Затем средства разработки этих модулей были включены в стандарт языка, и разными фирмами было разработано большое количество внешних подпрограмм для различных приложений. Так, фирма IBM разработала библиотеку научных подпрограмм на Фортране, в которую были включены практически все разделы вычислительной математики. В России была разработана библиотека графических подпрограмм на Фортране (ГРАФОР). Кроме этого в различных организациях было создано большое количество узкоспециализированных библиотек.

Такие модули существуют отдельно от транслятора в виде автономных библиотек как в объектном виде на магнитных носителях, так и в исходном виде, оформленных в различных форматах. Таким образом, средствами Фортрана пользователю предоставлялась возможность по-новому заниматься организацией своих программ.

Такой подход позволяет:

1) сократить текст основной программы путем включения дополнительных (внешних) модулей на уровне компоновки (иногда многократного включения одних и тех же модулей на различных фазах выполнения программы);

2) распараллеливать выполнение проекта, распределяя работу между программистами;

3) структурировать программу, облегчая ее тестирование и поиск ошибок.

Этот подход при проектировании сложных систем называется *восходящим проектированием* (или проектированием снизу вверх). В этом случае из базовых элементов (из подпрограмм) строится весь программный комплекс.

Фортран включает три структуры, использующие указанные принципы.

Первой структурой является оператор функции, которая, правда, не полностью отвечает вышеприведенным свойствам и не является внешней, а определяется в рамках единого текста программы и транслируется совместно с основной программой. Основное назначение ее заключается в следующем. Если в программе используется длинное арифметическое (или какое-то другое) выражение, которое по ходу программы необходимо многократно вычислять с различными аргументами, то в разделе описаний необходимо дать его формальное определение:

Name(arg) = EXPRESSION

где Name – имя оператора функции; arg – список формальных параметров; EXPRESSION – само выражение.

Например, в программе необходимо многократно вычислять выражение:

$$ax^2 + by^2 + cxy + d$$

Его можно оформить, как оператор функции:

```
QUAD(X,Y)=A*X**2+B*Y**2+C*X*Y+D
```

В последующем это выражение можно вычислять, не переписывая его, а подставляя в необходимые места только его имя (естественно, определив его в разделе описаний) и значения аргументов (X,Y), которые заменяют их формальные значения в определении оператора функции.

Например:

```
RESULT=X4+200*QUAD(X1,Y1)+300*QUAD(X2,Y2)+400*QU
AD(X3,Y3)
```

Однако если вычислительная процедура занимает несколько строчек, то необходимо применять другие методы.

В Фортране на этот случай предусмотрены две структуры, которые обеспечивают оформление сложных программных фрагментов, имеющих высокую степень автономности, возможность подключения их на уровне компоновки к различным программным комплексам (соответственно, эти модули должны располагаться в пределах доступа к ним редактора связи).

Первый модуль – подпрограмма функции, которая определяется следующим образом:

t FUNCTION f(a_1,a_2…a_n)

t – тип функции, может быть: Integer, Real, Double Precision, Complex, Logical; по умолчанию Real или Integer определяется первой буквой имени функции; FUNCTION – ключевое слово; f – имя функции; a_1,a_2, …, a_n – имена формальных параметров.

Структура этой подпрограммы следующая:

```
FUNCTION Name(a1, a2, …, an)
[исполнительная
 часть          ]

Name=Result
Return
End
```

После выполнения основной (исполнимой) части имени функции присваивается значение полученного результата ее работы (Result). Оператор Return передает результат в вызывающую программу (End – указание транслятору о завершении текста подпрограммы при трансляции).

При вызове подпрограммы функции вызывающая (главная) программа должна заменить (передать) формальные параметры подпрограммы на фактические, при которых необходимо проводить вычисления. Вызов осуществляется указанием имени подпрограммы функции и перечислением фактических параметров (перемен-

ных), указанных в скобках (аналогично вызову оператора функции).

Последовательность и тип передаваемых параметров из главной программы (которые, естественно, определяются до начала обращения к подпрограмме) должны соответствовать последовательности и типу формальных параметров, указанных в определении подпрограммы функции. Необходимо отметить, что имена и метки в подпрограммах функциях локализованы и не связаны с именами и метками в вызывающей программе.

В качестве примера рассмотрим подпрограмму функции для скалярного произведения двух векторов X и Y.

Алгоритм скалярного произведения:

$$S = \sum_{i=1}^{n} X_i \cdot Yi$$

```
FUNCTION SKALAR(X,Y,N)
DIMENSION X(100),Y(100)
s=0.0
do 1 i=1,N
s=s+X(i)*Y(i)
1        Continue
SKALAR=s
RETURN
END
```

В качестве фактических параметров подпрограммы функции могут выступать: массивы, элементы массива, выражения (арифметические, логические), имя стандартной функции, подпрограммы функции или подпрограммы. Характерной особенностью подпрограммы функции является то, что она вычисляет единственное значение и возвращает управление в вызывающую ее программу вместе с этим значением.

Существует еще один вид модулей более универсального назначения, который в Фортране называется SUBROUTINE, в Паскале – PROCEDURE (в отличие от SUBROUTINE подпрограмма PROCEDURE является внутренним программным модулем, доступ к которому возможен только из самой вызывающей программы, в состав которой он входит). В языке С, хотя все модули называются

функциями, которые вызываются из главной программы (main) с возможностями SUBROUTINE и PROCEDURE.

В подпрограммах типа SUBROUTINE снимается ограничение подпрограммы функции, которая обеспечивает единственный результат, присваиваемый имени функции. Это вызвано тем, что ряд задач связан с необходимостью получения большего количества выходных результатов (например, графические процедуры, матричные вычисления, решения систем линейных уравнений, дифференциальных уравнений и т.п.).

Программа SUBROUTINE оформляется следующим образом:

SUBROUTINE S(a_1, a_2, …, a_n)
[Тело подпрограммы
Исполнительная часть]
RETURN
END

где SUBROUTINE – ключевое слово; S – имя подпрограммы (не имеет типа и никак не связано с входными и выходными параметрами); a_1, a_2, …, a_n – формальные параметры, используемые при работе подпрограммы, включая и выходные параметры (результаты вычислений). Формальные и фактические параметры должны быть согласованы между собой (так же, как и в подпрограмме функции) по типу, количеству и порядку следования. Естественно выходные (вычисляемые) параметры фигурируют в главной программе только в виде имен (не имеющих значений до начала работы модуля). RETURN и END – выполняют те же функции, что и в подпрограмме функции.

В качестве примера использования SUBROUTINE рассмотрим математический алгоритм перемножения матрицы на вектор:

```
                SUBROUTINE MATRIX(A,X,Y,N)
              DIMENSION F(50,50),X(50),Y(50)
              DO 2 I=1,N
              s=0.0
              DO1 I=1,N
          1   s=s+A(I,J)*X(j)
          2   Y(I)=s
              END
RETURN
END
```

В данном случае значения всех параметров передаются из главной программы, и в результате работы этой SUBROUTINE в главную программу возвращается вычисленное значение вектора Y(I).

Вызов подпрограммы SUBROUTINE осуществляется следующим оператором:

$$\text{CALL METRIX}(a_1, a_2, \ldots, a_n)$$

что также отличает ее от программы функции. Такой способ вызова характерен только для Фортрана и отличен от вызова таких подпрограмм в других языках (где эти вызовы осуществляются по имени подпрограммы).

Следующее направление в развитии языков программирования (парадигма программирования) обычно связывают с так называемым *структурным программированием,* причиной которого также явилось продолжающееся увеличение сложности разрабатываемых систем и увеличение их размера. Сама идея структурного программирования также не оригинальна и является следствием блочно-иерархического подхода, применяемого при разработке сложных технических систем. Суть этого подхода заключается в расчленении задачи большой размерности на иерархические уровни-блоки, которые, в свою очередь, могут быть расчленены на еще меньшие составляющие, вплоть до базовых элементов (не подлежащих расчленению, принцип суперпозиции). На каждом уровне применяются свои понятия о входных и выходных параметрах и базовых элементах. Этот способ проектирования называется «*проектирование сверху вниз*» и на сегодняшний день является основным подходом к проектированию сложных технических систем.

Появление парадигмы структурного программирования обычно связывается с разработкой Никлаусом Виртом языка Pascal в 1972 г. Основное назначение этого языка изначально связано с обучением студентов программированию. Следует отметить, что Вирт – одна из ключевых фигур в создании программистской идеологии. Он разработал также такие языки, как Modula, Oberon, Object Pascal. Справедливости ради необходимо отметить, что большую популярность Pascal приобрел благодаря очень удачной среде Turbo Pascal, разработанной фирмой Borland. При этом стандарт «оригинального» языка значительно расширен фирмой. Что же касается структурного программирования по отношению к языку

Pascal, то здесь можно выделить несколько основных элементов, отношение к которым не всегда однозначно, хотя принципы блочности, иерархичности и декомпозиции явно просматриваются в его структуре.

1. При организации программы Pascal придерживается более строгого подхода. Обязательной является описательная (неисполняемая) часть, которая оформляется с помощью специальных синтаксических структур с описанием всех используемых системных модулей и встроенных процедур и функций, а также с описанием всех используемых параметров с именами переменных, констант, массивов и т.п. (т.е. принятые в Фортране умолчания не допускаются).

2. Включаются в текст программы структуры FUNCTION и PROCEDURE, функционально являющиеся полным аналогом FUNCTION и SUBROUTINE, но в отличие от Фортрана они являются внутренними подпрограммами, транслируемыми вместе с основной программой, и недоступны извне.

3. Используются блочные конструкции begin…end, придающие автономность отдельным частям текста программы, продолжение выполнения которой возможно только после завершения выполнения участка, заключенного в эти скобочные конструкции.

4. Применяются структурные составные операторы (циклические и ветвящиеся), что связано с отказом от использования меток для указания завершающей строки структуры. Эта особенность и предполагает, что всю программу можно представить состоящей из автономных блоков, на которые можно разделить описание большой размерности.

Однако необходимо отметить и некоторые отклонения от основной (генеральной) линии – структурного программирования, особенно при дальнейшем развитии языка и введении дополнительных функциональных возможностей.

Так, если основная критика Фортрана связана с оператором безусловного перехода (go to) и наличием меток, что приводит в больших программах к «*паутине*» переходов, затрудняя чтение программы и поиск ошибок (нарушает структурированностью), то, тем не менее, в Pascal эти средства также были введены, поскольку они в ряде случаев оказались очень удобны. Локализация имен и

меток, наличие глобальных имен также является отходом от основной (генеральной) линии Pascal.

Кроме этого, начиная с определенных версий Turbo Pascal (фирмы Borland), введены автономные (отдельно транслируемые) модули UNIT (по образу и подобию Фортрана). В связи с этим разговоры о структурированности уменьшились, особенно в связи с переходом на парадигму объектно-ориентированного программирования и развитием CASE-технологии. Концепция объектно-ориентированного программирования представляет современную парадигму традиционного программирования.

Несмотря на замысловатую терминологию и всеобщий интерес к этой технологии программирования, она является естественным продолжением хорошо известных принципов, которые уже давно внедрены на системном уровне. Причины развития этих принципов те же самые, что и развития процедурно-ориентированного и структурированного программирования – постоянно увеличивающаяся сложность задач и постоянный рост объема программ (проектирование сложных технических систем – СТС).

В основе объектно-ориентированного программирования находится понятие абстракции, которое изначально использовали языки программирования, однако в прежние времена эта технология не была в ходу. В определенном смысле любая программа высокого уровня является абстракцией, так как она абстрагируется от непосредственно исполнимой программы. В современных языках программирования используется два основных вида абстракции: абстракция процесса и абстракция данных, которые и составляют идеологическую основу объектно-ориентированного программирования.

Абстракция процесса – одно из наиболее старых понятий в области разработки языков программирования. Даже язык Plankalkul поддерживал абстракцию процесса. Все подпрограммы являются абстракцией процессов, поскольку они определяют способ, с помощью которого программа устанавливает, что необходимо выполнить некоторый процесс без уточнения деталей того, как именно это можно сделать (т.е. сама реализация этого процесса скрыта от пользователя в вызывающей ЦП программе).

Например, если в программе нужно упорядочить массив числовых данных некоторого типа, обычно используется подпрограмма, выполняющая сортировку. В основной программе, когда требуется выполнить сортировку, помещается оператор следующего вида:

SORT_int(list,list_len)

Этот вызов является абстракцией реального процесса сортировки (алгоритм которого в вызывающей программе не определен и не зависит от алгоритма в вызываемой программе). Единственным существенным моментом является имя вызываемой подпрограммы (SORT_int), а также указание (передачи в программу SORT_int) массива подлежащего сортировке (list) и имя отсортированного массива, который возвращается в основную программу (list_len).

Абстракция процесса – ключевое понятие в программировании. Возможность абстрагироваться от многочисленных деталей алгоритма, который выполняется подпрограммой, позволяет читать и понимать большие программы (в десятки тысяч строк).

Эволюция абстракции данных сопровождается эволюцией абстракции процесса, поскольку неотъемлемой и главной частью любой абстракции данных являются операции, которые определяются как абстракции процесса. Абстракция данных составляет основу (парадигму) информационно-ориентированного программирования (в противоположность процедурному). Эта парадигма стала популярной в 80-х годах прошлого века. Языки, поддерживающие информационное программирование, часто называются объектными (object_based). Они составляют основу объектно-ориентированного программирования. Появление абстракции данных связано с так называемой *инкапсуляцией* (скрытая реализация*).

Инкапсуляция – способ объединения в единое целое подпрограмм и данных, которые они обрабатывают. Такое объединение предоставляет пользователю возможность создавать собственные типы данных, аналогичных встроенным типам, используемыми

* Обращение к сложным подпрограммам в процедурно-ориентированном программировании связано с определением большого количества формальных параметров, согласованных по количеству, порядку следования и типу с параметрами формальными (большое количество ошибок возникает из-за этой несогласованности). В связи с этим пользователю, хотя и не требуется подробно разбираться в алгоритме, который используется в подпрограмме, нужно иметь явное представление о принципах его построения.

процедурными языками программирования (целые, действительные, логические, символьные и т.п.). В отличие от абстракции процесса, когда данные и методы, позволительные для их обработки, разделены (данные в главной программе, а методы в подпрограммах) и в процессе передачи очень часто возникают проблемы, инкапсуляция исключает эти трудности (поскольку ничего передавать не требуется).

Таким образом, абстрактный тип данных – инкапсуляция, которая содержит только представление данных одного конкретного типа и подпрограммы, которые выполняют операции с данными этого типа. Это средство против сложности, способ сделать большие и сложные программы более управляемыми.

Экземпляр абстрактного типа данных называется *объектом*.

Концепция объектно-ориентированного программирования (object-oriented) уходит корнями в язык SIMULA-67 (норвежцы – Кристен Нигаард и Оле Йохан Дал), но она не была полностью разработана, пока эволюция языка Smalltalk (американец Ален Лэй) не привела к появлению языка Smalltalk 80.

На сегодняшний день для всех традиционных языков программирования разработаны объектно-ориентированные диалекты (начиная с C^{++}, Ada, Pascal, Fortran) и т.п. Некоторые из современных языков программирования, разработанных для объектно-ориентированного программирования, не поддерживают другие парадигмы программирования, но продолжают использовать некоторые основные структуры императивных языков и внешне на них похожи. К таким языкам, например, относится Java. Кроме того, существует один полностью объектно-ориентированный язык, упомянутый ранее, – Smalltalk.

Объектно-ориентированное программирование имеет ряд важных свойств, развивающих информационно-ориентированное программирование, приводящее его из объектных языков к объектно-ориентированным. Объектно-ориентированные языки должны поддерживать три ключевых языковых свойства: абстрактные типы данных (которые уже обсуждались), наследование и динамическое связывание с так называемым полиморфизмом (последние свойства отсутствовали в концепции информационно-ориентированного программирования).

Наследование. В объектно-ориентированном программировании часть, общая для набора похожих типов данных, выделяется в новый тип.

Повторное использование разработанных программ является мощным средством повышения эффективности программирования и является сердцевиной объектно-ориентированного программирования. Наследование позволяет решить как проблемы модификации, возникающие в результате повторного использования абстрактного типа данных, так и проблемы организации программ. Если новый абстрактный тип данных может наследовать данные и функциональные свойства некоторого существующего типа, а также модифицировать некоторые из этих сущностей и добавлять новые, то повторное использование значительно облегчается и не связано с необходимостью вносить изменения в повторно используемый абстрактный объект.

Без механизма наследования модификации объекта могут быть трудновыполнимы и потребуют от программиста понимания части, если не всего целиком существующего кода. Кроме того, во многих случаях модификации влекут за собой изменения во всех программах клиента.

Предположим, что в программе есть абстрактный тип данных для массива целых чисел, включающих в себя операцию сортировки. После некоторого периода использования программа модифицируется и требует наличия не только абстрактного типа данных для массивов целых чисел с операцией сортировки, но и операции вычисления арифметического среднего для элементов объектов, представляющих собой массив. Поскольку структура массива скрыта в абстракционном типе данных (без наследования), этот тип должен быть модифицирован путем добавления новой операции в эту структуру (т.е. нужно разбираться в тексте существующей программы). При наличии наследования нет необходимости в модификации существующего типа: можно описать подкласс существующего типа, поддерживающий не только операции сортировки, но и операцию среднего арифметического.

Абстрактные типы данных с похожими свойствами в объектно-ориентированных языках обычно называются *классами* (classes).

Как и экземпляры абстрактных типов данных, экземпляры классов называются *объектами*. Класс, который определяется через

наследование от другого класса, называется *производным классом* (*derived class*), или *подклассом*.

Класс, от которого производится новый класс, называется *родительским классом* (*parent class*), или *суперклассом* (*super class*).

Программы, определяющие операции над объектами класса, называются *методами* (*methods*). Вызовы методов называются *сообщениями* (*message*). Весь набор методов объекта называется *протоколом сообщений* (*message protocol*), или *интерфейсом сообщений* объекта. Сообщение должно иметь, по крайней мере, две части: конкретный объект, которому оно должно быть послано, и имя определяющего необходимые действия над объектом.

И, наконец, третьим свойством объектно-ориентированного программирования является полиморфизм, который обеспечивается с помощью так называемого динамического связывания.

Полиморфизм (многозначность сообщений, или их неоднозначность) – свойство, когда одинаковые сообщения по-разному понимаются разными объектами, в зависимости от их класса. Так, одно и тоже сообщение звездочки (*) вызывает совершенно разные методы применительно к целым, действительным, комплексным числам или операциям с матрицами.

Связывание (*binding*) в общем смысле представляет собой процесс установления связи, аналогичной существующей между атрибутом и объектом или между операцией и символом. Связывание – важное понятие семантики языков программирования. Связывание называется *статическим* (*static*), если оно выполняется до выполнения программы и не меняется во время ее выполнения. Если связывание происходит во время выполнения программы или может меняться в ходе ее выполнения, то оно называется *динамическим*.

Применительно к объектам данных имеется в виду связывание типов данных и их методов. Во всех рассмотренных нами ранее типах данных, включаемых в стандарты языков, имелось в виду *статическое связывание*.

При статическом связывании код необходимых для работы программы библиотечных функций физически добавляется к коду объектных модулей для получения загрузочного модуля, который в результате получается очень большого размера.

При динамическом связывании в результирующем модуле проставляются лишь ссылки на код необходимых библиотечных

функций, если код реально добавляется только при его исполнении и освобождает память после завершения его работы.

Несмотря на то, что при динамическом связывании начальная загрузка выполняется несколько медленнее, оно обеспечивает большую гибкость при разработке программных продуктов (одна и та же программа может работать с различными типами данных) и является естественным дополнением полиморфизма.

Еще одно направление в развитии императивных языков связано с параллельным программированием. Вопрос о параллельном программировании возник достаточно давно в связи с появлением первых суперкомпьютеров и являлся занятием довольно узкого круга людей, занимающихся научными задачами в области военной проблематики. Однако в последнее время необходимость в параллельном программировании приобретает массовый характер. Этому способствует как непрерывное падение цен на параллельные компьютеры, так и постоянное усложнение задач, решаемых на них. Причем это не только традиционные задачи физики, химии, биологии, медицины и других наук, в которых переход от упрощенных моделей к реальным задачам сопряжен с количественным ростом объема вычислений, но также задачи, связанные с разработкой систем управления базами данных (СУБД), разнообразными Internet-серверами запросов (WWW, FTP, DNS и т.п.) и совместного использования данных (NFS, SMB* и др.), автоматизированными системами управления производством (АСУП) и технологическими процессами (АСУТП), поскольку в этих задачах требуется обеспечить обслуживание максимального количества запросов в единицу времени, а сложность самого запроса растет.

Можно сказать, что в настоящее время практически разработка всех крупных программных проектов как научной, так и коммерческой направленности либо содержит поддержку параллельной работы на компьютерах различного типа, либо эта поддержка запланирована.

Говоря о параллельном программировании, необходимо отметить, что даже на современных однопроцессорных компьютерах с помощью логического распараллеливания и расслоения памяти

---

* NFS – Network File System – сетевая файловая система.

SMB – Server Message Block – протокол верхнего уровня, выполняющий функции сеансового, представительского и прикладного уровней.

можно достичь существенного повышения производительности при правильном выборе алгоритма решения задач. Так, можно в 14 раз увеличить скорость перемножения матриц при выборе алгоритма перемножения, эффективно использующего кэш-память.

Для многопроцессорных компьютеров обычно рассматривается распараллеливание на уровне процессов. Под процессом понимается динамическая сущность программы, ее код в процессе выполнения, который имеет собственные области памяти под код и данные, собственное отображение виртуальной памяти на физическую может находиться в различных состояниях (которые будут обсуждаться в следующей главе).

Кроме процессов, возможно распараллеливание на уровне задач (потоков или нитей, thread) – это как бы одна из ветвей исполняемого процесса. Речь идет о распараллеливании внутри исполнимой программы. Задача разделяет с процессом как память под код и данные, так и отображение виртуальной памяти на физическую, а также имеет собственное состояние.

Реализация распараллеливания процессов и задач возможна как с помощью средств операционных систем, прежде всего различных версий UNIX, так и с помощью функций императивных языков программирования. Так, появившийся в 1990 г. интерфейс MPI (Message Passing Interfere), ставший фактически стандартом для программирования систем с распределенной памятью, приобрел большую популярность и теперь с успехом используется также в системах с общей памятью и разнообразных смешанных вычислительных установках. Он представляет собой набор утилит (вспомогательных программ) и библиотечных функций (для языков C/C^{++}, FORTRAN), позволяющих создавать и запускать приложения, работающие на параллельных вычислительных установках различной природы.

В последнее время большинство компиляторов языков Си, Си++ и Фортран стали поддерживать стандарт OpenMP, разработанный фирмой Intel специально для распараллеливания процессов. В отличие от MPI он реализуется не в виде подпрограмм, а добавляет в язык новые ключевые слова, благодаря которым распараллеливание осуществляется автоматически на этапе выполнения програм-

мы, причем это делается максимально эффективным для данной процессорной системы образом (при этом учитываются такие параметры, как, например, количество процессов в системе, доступных программе).

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. В чем различие между языком программирования и его реализацией? Каковы составные части языка – синтаксис и сематика, различия в использовании служебных слов (ключевые, зарезервированные, предопределенные слова)?

2. Чем определяется уровень языка? В чем особенности и к какому уровню можно отнести: машинные языки и ассемблеры, императивные языки, аппликативные языки, дескриптивные языки?

3. Назовите причины, стимулирующие развитие языков программирования и основные тенденции в их развитии.

4. Перечислите основные направления специализации языков программирования. Каковы причины расширения функциональности проблемно-ориентированных языков программирования?

5. Расскажите о модульном, структурном, объектно-ориентированном программировании. Назовите основные принципы, причины развития языков в этих направлениях.

ГЛАВА 6. ПРОГРАММНЫЕ СРЕДСТВА СОД

6.1. Общие сведения, классификация программного обеспечения и краткая характеристика отдельных частей

До 70-х годов изготовители ЭВМ, как правило, продавали лишь изделие одного вида – аппаратуру (Hard ware). Операционные системы (ОС), вспомогательные служебные программы, пакеты прикладных программ (ППП), эксплуатационная документация и учебные материалы часто поставлялись пользователям бесплатно. В начале 70-х годов IBM, разделив свои программные средства (soft ware) и аппаратуру, установила отдельные цены, хотя по-прежнему продолжала поставлять программное обеспечение бесплатно, что связано с ростом значимости программного обеспечения (ПО).

Все это привело к следующим последствиям:

появилось большое количество фирм, разрабатывающих и продающих ПО;

создается индустрия ПО;

пользователь получил возможность выбора ПО у конкурирующих фирм;

развился модульный принцип и появились стандарты в программировании;

повысилась ответственность поставщиков ЭВМ за ПО.

Наряду с традиционно продаваемым Soft ware, появились Firm ware – аппаратно-ориентированное ПО, поставляемое производителем ЭВМ (например, BIOS), Share ware – условно бесплатное ПО, Free ware – бесплатное ПО и т.п.

Для проведения обработки на ЭВМ алгоритм решения задачи необходимо написать в форме, понятной вычислительной машине. Языки программирования являются таким средством, с помощью которого можно «договориться» с вычислительной машиной путем разработки программ.

Программа – последовательность команд или операторов, которая после декодирования ее либо вычислительной машиной, либо вычислительной машиной и транслирующей программой может заставить эту машину выполнить некоторую работу. Программа одного и того же функционального назначения может иметь три различных состояния (или формата). *Исходная программа* – про-

грамма в кодах языка программирования, *объектная* (или полуготовая) – программа в кодах машины, прошедшая фазу трансляции, но выполнение которой невозможно по ряду причин. И, наконец, *рабочая программа* (или загрузочный модуль), которая может быть выполнена на ЭВМ.

Под программным обеспечением понимают группу взаимодействующих программ.

Любая программа имеет три стадии (или фазы) жизни (рис. 6.1).

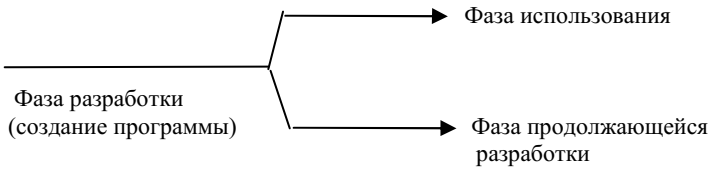


Рис. 6.1. Жизненный цикл программного обеспечения

На этапе продолжающейся разработки:

устраняются скрытые дефекты, не выявленные на этапе тестирования;

вносятся изменения в существующую версию, учитывающие опыт использования;

проводится адаптация программ к изменяющимся требованиям.

В общем объеме работ принято считать [8], что 33 % составляет фаза разработки, а 67 % – сопровождение (продолжающаяся разработка) и эксплуатация.

Сама фаза разработки предполагает прохождение шести обязательных этапов:

определение требований и заданий;

проектирование (алгоритмизация);

написание команд (кодирование – программирование);

компоновка;

тестирование;

документирование.

Разумеется, в данном случае имеется в виду, что программный продукт закончен и представляет собой промышленное изделие. Затраты на разработку программного обеспечения довольно высоки. Так, в США ежегодно тратится более 20 млрд дол. на эти цели.

Высокая стоимость программ объясняется низкой скоростью роста производительности труда программистов. Говоря о высокой стоимости программного обеспечения, очень часто ошибочно абсолютизируют соотношение между стоимостью программной и аппаратной частями системы. Утверждение о том, что программное обеспечение намного дороже аппаратного (обычно распространяемое программистами) справедливо в редких случаях при разработке больших по объему уникальных программных комплексов (например, в системах управления спутниками Земли). В большинстве тиражируемых систем ПО (например, ПК) это соотношение может быть прямо противоположным.

В заключение приведем высказывание одного из ведущих специалистов США по разработке программного обеспечения Дж. Фокса. Он сказал [18]: «Разрабатывать программное обеспечение с каждым годом становится все труднее и в то же самое время разрабатывать программное обеспечение с каждым годом становится легче». Это связано, прежде всего, с тем, что стремление к автоматизации труда программиста путем включения в этот процесс ЭВМ привело к созданию очень большого количества программ, освоение которых требует дополнительных и немалых усилий. Дадим краткий обзор существующих на сегодняшний день программных продуктов.

1. **Общесистемное ПО**, в котором иногда выделяют две части:

системное ПО – программы, выполняемые вместе с прикладными (сопровождают выполнение прикладных программ);

инструментальное ПО – программы, помогающие программистам и администрации создавать ПО, однако при исполнении разработанных программ они, как правило, не участвуют.

2. **Прикладное ПО** – программы, фактически выполняющие задачу пользователя. Имеется несколько способов организации этого ПО, находящихся в тесном взаимодействии. Например, позадачный подход, который представлен в виде отдельных программ, разрабатываемых под конкретные задачи пользователей и отражающих узкоспециализированные особенности этих задач. Этот подход применяется традиционно, когда технические требования не укладываются в рамках стандартных программных решений и требуют значительных временных и материальных ресурсов.

Библиотеки программ по специальностям – наборы программ по определенной тематике, описанные с соблюдением определенных стандартов в виде модулей типа функций или подпрограмм.

Они могут быть представлены как исходные тексты или объектные модули. Не имея самостоятельного значения, они предполагают наличие главной программы (предоставленной пользователем), из которой к ним происходит обращение и включение в состав этой программы редактором связи (Linkeditor).

Существует несколько уровней библиотечных программ. К ним относятся стандартные функции и подпрограммы, включаемые в состав систем программирования традиционных алгоритмических языков, а также разнообразные обрабатывающие функции непроцедурных языков (элементарные математические функции, подпрограммы графических построений, системные функции и т.п.). Существует ряд популярных библиотек подпрограмм в виде листингов исходных текстов и кодов, распространяемых в печатных изданиях или на различных носителях ВЗУ ЭВМ. К ним можно отнести библиотеки научных программ на Фортране, Паскале и т.п. Существуют и библиотечные программы конечного пользователя, разработанные с учетом специфики различных областей применения ПО. Библиотечные программы могут являться составной частью программного продукта, разрабатываемого при позадачном подходе.

Пакеты прикладных программ (ППП) – в отличие от библиотек самостоятельные автономные образования. В своем составе они имеют, кроме функциональных модулей, характерных для библиотек, управляющую программу (главную программу), выделенную базу данных и входной язык пакета (непроцедурный язык). Практически во всех ППП вводятся инструментальные средства традиционного программирования, что связано с невозможностью полностью удовлетворить потребности некоторых приложений стандартными средствами пакета. Как отмечалось ранее, в состав пакетов включается большое количество разнообразных библиотечных функций, отражающих специфику конкретных ППП.

Начиная с 1968 г. возникает мнение о кризисе программного обеспечения (Software Crisis), что связано с общим процессом проектирования, вызванным постоянным усложнением технических систем и появлением сложных технических систем (СТС), увеличением объема и сложности программных продуктов, значитель-

ным ростом ресурсов вычислительных систем, массовым внедрением параллельных архитектур.

Традиционный, позадачный способ проектирования ПО при создании крупных программных комплексов оказался тормозом в развитии ПО. В связи с этим аналогично системам автоматизированного проектирования (САПР) при проектировании СТС появились средства автоматизированного проектирования ПО, т.е. включение компьютера в процесс проектирования ПО. Эти средства получили названия CASE-технологии (Computer Aided SoftWare Engineering, т.е. создание ПО с помощью компьютера).

Эти средства реализуют блочно-иерархический подход при разработке ПО и так называемое проектирование сверху вниз.

Суть этого подхода давно и хорошо известна. Он связан с декомпозицией, т.е. с разбиением большой задачи на подзадачи и привлечением средств ЭВМ на всех шести этапах проектирования ПО, перечисленных выше, и использованием объектно-ориентированного подхода. К этим средствам относятся как текстовые и графические редакторы для изготовления описаний, блок-схем, графов и т.п. на верхних уровнях проектирования, так и средства генерации исполняемого кода на нижнем уровне – уровне базовых элементов.

К числу наиболее популярных элементов CASE-средств можно отнести унифицированный язык моделирования UML (United Modelling Language), имеющий четырехуровневую архитектуру и с 1997 г. являющийся стандартом в области визуального объектно-ориентированного моделирования.

6.2. Операционные системы, их эволюция, состав и функциональное назначение отдельных частей

Основу общесистемного ПО составляют ОС. Это – один из наиболее важных компонентов вычислительной системы независимо от класса ЭВМ, которая осуществляет интерфейсные функции между пользователем и аппаратурой. Кроме того, ОС обеспечивают управление ресурсами вычислительной системы (ЦП, ОП, УВВ). За время существования ЭВМ в эволюции ОС можно выделить несколько этапов, на каждом из которых решались задачи, актуальные для определенного времени.

В 40-е годы полное отсутствие элементов ОС привело к чрезвычайно низкой эффективности использования ЭВМ. Операторы должны были вручную (с помощью переключателей) загружать и выполнять программы пользователей. Время работы ЭВМ составляло незначительную часть по сравнению с подготовительными операциями. Хотя ресурсы ЭВМ полностью использовались при работе загружаемой программой, отсутствовали «накладные расходы» на программное сервисное обслуживание ЭВМ.

50-е годы ознаменовались появлением зачатков ОС, хотя отсутствовало само понятие этой структуры. Основная задача, которая решалась на этом этапе, – повышение эффективности использования ЭВМ, создание удобств для оператора по загрузке и выполнению задач пользователей. Был разработан пакетный режим обработки данных. Исходные тексты (или командные коды) программ готовились на автономных устройствах подготовки данных (на перфокартах или перфоленте (УПДК, УПДЛ)), после чего загружались в ЭВМ.

60-е – 70-е годы связаны с появлением многофункциональных операционных систем общего назначения для больших ЭВМ. Это громоздкие замкнутые системы, полностью загружаемые в ОП ЭВМ с круглосуточным режимом работы машины (поскольку загрузка была длительным и сложным процессом). При этом на работу ОС требовались значительные ресурсы ЭВМ. В это же время в связи с появлением таких технических средств, как схемы прерываний, генераторы истинного времени – таймеры и механизмы защиты памяти, были разработаны программные средства обеспечения многозадачного, многопользовательского режимов работы ЭВМ, так называемого режима разделения времени. Таким образом, было уделено внимание удобствам работы пользователей.

70-е – 80-е годы связаны с развитием модульного принципа построения ОС, их стандартизации и дальнейшим совершенствованием многопользовательских и многозадачных режимов. В сферу ОС входит организация структур с перекрытием организации систем виртуальной памяти. Этому способствовал прогресс технических средств, появление вначале мини-ЭВМ, а затем и персональных компьютеров.

90-е – 2000-е годы связаны с интенсивным внедрением сетевых ОС.

В состав ОС (рис. 6.2) входят программы, присутствующие в любой ОС универсального назначения.

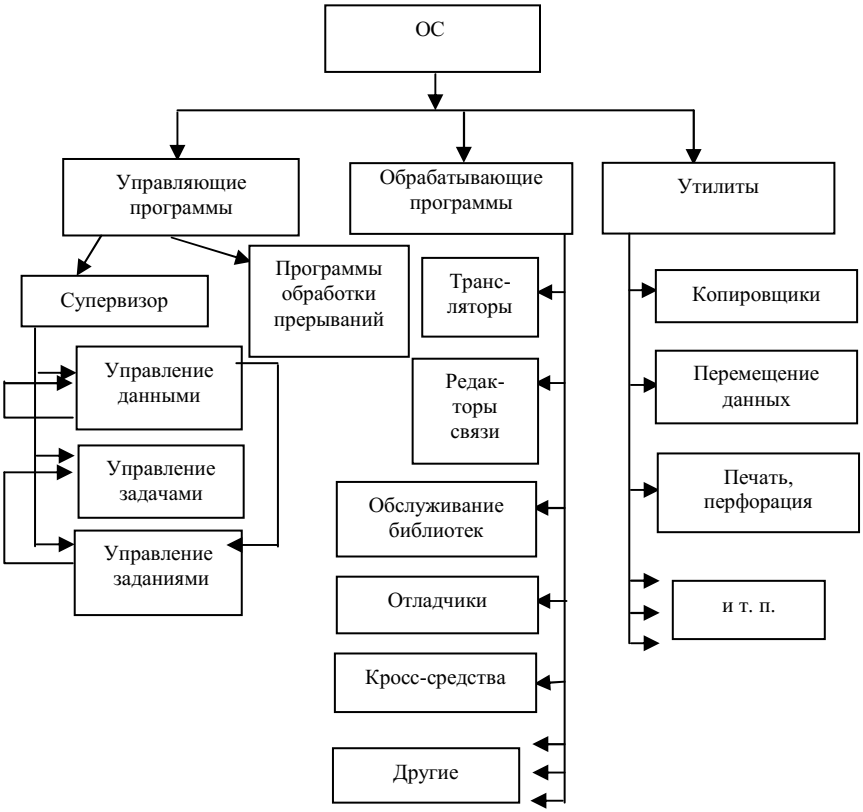


Рис. 6.2. Состав операционных систем

Управляющие программы. Ядро ОС составляют управляющие программы, с которых и началось создание ПО этого вида. Одно из важнейших понятий, связанных с управляющими программами, – понятие «процесс», под которым понимается любая задача (или несколько), находящаяся в ОП и обрабатываемая ЦП. Другими словами, это среда аппаратного или программного обеспечения, в

которой выполняется программа или происходит сеанс работы за терминалом.

Когда входите в систему, она порождает для Вас процесс. Процессу присваиваются атрибуты, связанные с Вашим разделом: ресурсы аппаратуры (например, место на диске), квота ОП памяти, привилегия (например, возможность доступа к файлам и т.п.).

Процессы могут порождать процессы, которые будут выполняться совместно.

Программы – супервизоры, осуществляющие планирование работы процессов, представлены в модуле «Управление задачами». Качественный скачок от сравнительно простых управляющих программ к современным ОС произошел с появлением мультипрограммной обработки задач (см. гл. 2). Программная поддержка этого режима может осуществляться двумя путями.

1. Переключение по событию происходит, когда ОС не определяет момент смены текущей задачи. Этот путь может быть указан активной задачей, «добровольно» приостанавливающей свое выполнение с помощью специального обращения к операционной системе (например, с запросом ввода-вывода). Поскольку задача переходит затем в ожидание результата операции, операционная система может передать управление процессом другой ожидающей задаче. Этот метод требует организации системы приоритетов для указания задачи (из очереди), которой необходимо передать управление. *Приоритет* – число, приписанное ОС каждому процессу и задаче. Чем больше это число, тем важнее этот процесс или задача.

2. Метод разделения времени (*time-sharing*) основан на использовании внешнего сигнала, поступающего на процессор через равные интервалы времени. Обычно происходит аппаратное прерывание от внешнего таймера, при этом операционная система прекращает выполнение текущей задачи и сохраняет ее параметры в момент остановки. Планировщик анализирует список ожидающих задач и передает управление выбранной задаче.

На рис. 6.3 представлены изменения состояний процесса при реализации мультипрограммного режима работы.

В любой момент времени активным (т.е. использующим ЦП) может быть только один процесс. В режиме вытесняющей многозадачности при передаче управления процессу пользователя программа управления задачами устанавливает интервальный таймер,

задавая квант времени, являющийся максимальным количеством времени ЦП, на которое процесс получает управление. После этого процесс из состояния активности переходит в состояние готовности. Затем, согласно стратегии планирования, выбирается следующий процесс, находящийся в готовности. Может оказаться, что активный процесс, не использовав полностью предоставленного ему кванта времени, будет ожидать некоторого события, например завершения операции ввода-вывода. В этом случае активный процесс блокируется, а какой-то новый процесс активизируется. Когда же ожидаемое событие наступает, соответствующий заблокированный процесс приводится в состояние готовности и может снова стать кандидатом на обслуживание.

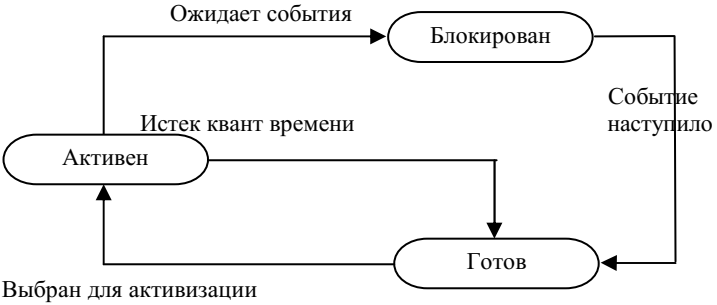


Рис. 6.3. Изменение состояния процесса

Режим невытесняющей многозадачности исключает прерывание по таймеру (в схеме рис. 6.3 исключается цепь «Истек квант времени»).

Программы управления заданиями обеспечивают прием и интерпретацию директив так называемого языка управления заданиями – инициирование и завершение заданий.

Имеется несколько типов заданий, представляющих команды ОС:

получение информации о системе;

манипулирование файлами;

команды запуска программ на выполнение;

командные процедуры пакетной обработки;

команды, связанные с разработкой программ;

распределение ресурсов памяти и внешних устройств.

Программы управления данными, входящие в состав супервизора, обслуживают так называемый физический уровень ввода-вывода. Они обеспечивают закрепление конкретных физических устройств ввода-вывода за логическими именами, набор аппаратно-ориентированных программ-драйверов, включают управление потоками данных между ОП и конкретными внешними устройствами.

Прерывание (*interrupt*) – сигнал, заставляющий менять порядок исполнения потока команд. Возникновение подобных сигналов может быть обусловлено самыми разнообразными событиями. На рис. 6.4 представлена последовательность событий, происходящих в ответ на сигнал прерывания.

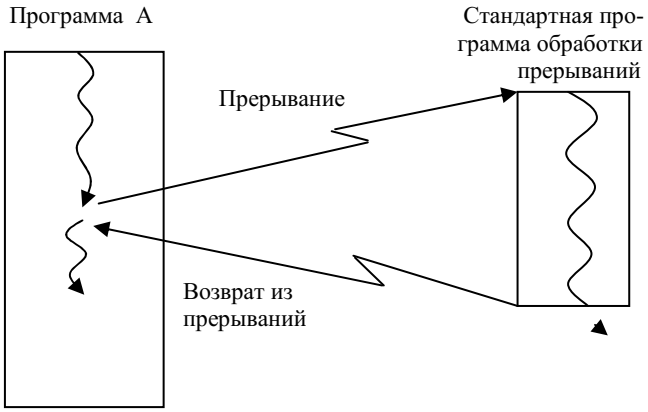


Рис. 6.4. Основная концепция обработки прерываний

Если в момент поступления от некоторого источника сигнала прерывания программа А находится в процессе исполнения, управление автоматически передается блоку программ обработки прерываний. Эти программы предназначены для выполнения некоторых действий в ответ на условие, вызвавшее прерывание. После завершения обработки управление может быть снова передано в ту точку программы А, где ее выполнение было прервано.

Следует иметь в виду, что возникновение и обработка прерываний могут быть совершенно не связаны с прерванной программой А. Можно выделить четыре класса прерываний.

Первый класс – прерывания, возникающие при выполнении ЦП команд вызова супервизора.

Второй класс – программные прерывания, возникающие при появлении некоторой ситуации, такой, как деление на нуль или попытка выполнить неправильную машинную команду и т.п.

Третий класс – прерывание по таймеру, которое обсуждалось выше.

И, наконец, четвертый – главный вид прерываний – прерывание по вводу-выводу.

Управляющая часть ОС обычно называется монитором и является резидентной, т.е. постоянно находится в ОП ЭВМ.

Следующие две подгруппы программ, входящие в ОС, – это обрабатывающие и утилиты (или вспомогательные программы).

Программы-утилиты. Обычно они хранятся в дисковой памяти и вызываются по мере необходимости.

Обрабатывающие программы. Они составляют основу инструментального обеспечения ЭВМ (см. рис. 6.2). Кратко рассмотрим их назначение и разновидности. Они отражают три состояния одной и той же программы, о которой уже упоминалось: фаза подготовки на машинный носитель (диск или ленту) текста программы на языке программирования (вид S – source); фаза трансляции и перевода написанной программы в коды ЭВМ (вид O – object); фаза компоновки и сборки готовой программы (вид B – binary) с помощью компоновщика (или редактора связей).

Язык программирования и транслятор, реализующий его на конкретной ЭВМ, обычно объединяют под общим названием – система программирования.

Трансляторы могут быть представлены в следующих вариантах:

ассемблеры;

компиляторы;

интерпретаторы;

трансляторы, использующие промежуточный код;

кросс-трансляторы.

Ассемблеры – трансляторы с языков Ассемблер. Хотя у каждого типа ЭВМ имеется свой язык, все разновидности трансляторов с этих языков имеют единое название – Ассемблер.

Компиляторы – наиболее распространенный вид транслятора. Схема его работы представлена на рис. 6.5, *а*.

После создания текста программы (исходного модуля) на машинном носителе с помощью текстового редактора EDIT (1-й этап –- EDIT) он загружается в ОП и последовательно, строчка за строчкой, обрабатывается компилятором (COMPILE). Как правило, компиляторы имеют синтаксическую диагностику, поэтому этот процесс (EDIT-COMPILE) носит итерационный характер.

После завершения синтаксической отладки (2-й этап) создается объектный модуль (О) этой программы, который на следующем этапе 3 обрабатывается редактором связи (или компоновщиком). Компоновщик включает в программу необходимые библиотечные модули и в результате работы этой программы создается исполнимая программа (В). Таким образом, в результате работы трех обрабатывающих программ имеются три программы одного и того же функционального назначения в различных форматах. Как правило, по завершении каждой стадии программы перечисленных форматов копируются на диск, а соответствующие обрабатывающие программы также удаляются из ОП. В связи с этим, чтобы запустить изготовленную программу, ее необходимо загрузить (этап 4 – LOAD) в ОП и произвести ее запуск (этап 5 – RUN).

В отличие от компилятора *интерпретатор* обрабатывает по одному предложению программы, выполняя все четыре фазы, (рис. 6.5, *б*). Разница, в некотором смысле, подобна разнице между переводчиком литературы и переводчиком устной речи. Этот режим характерен для диалоговых языков и, безусловно, более удобен с точки зрения алгоритмической отладки программ и оперативного получения результатов. Но для выполнения интерпретируемой программы интерпретатор должен находиться в оперативной памяти вместе с выполняемой программой.

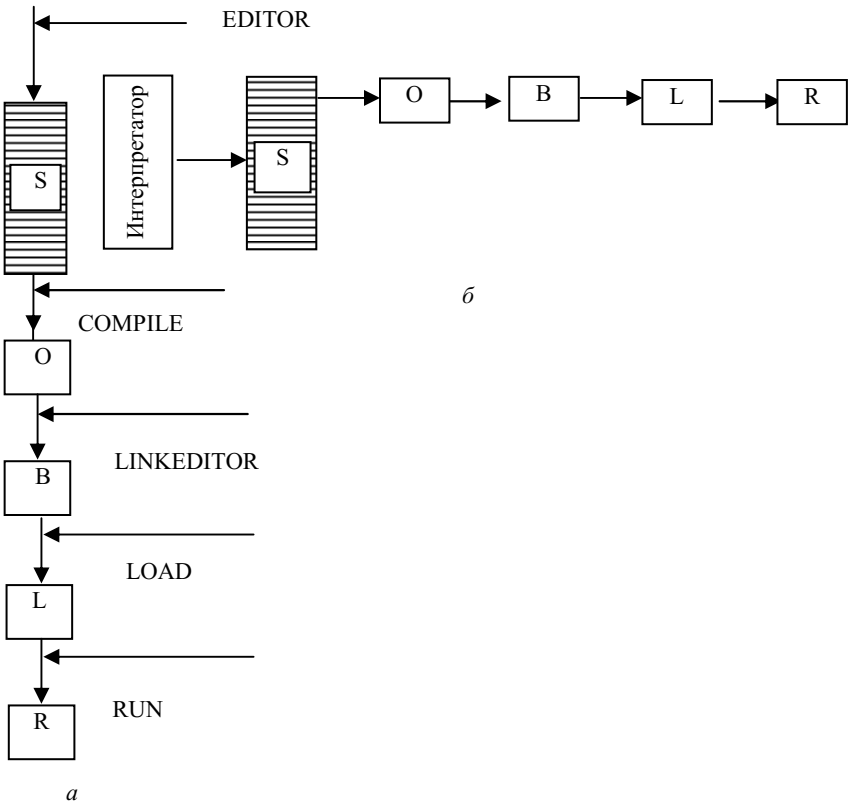


Рис. 6.5. Схема работы компилятора (*а*) и интерпретатора (*б*)

В принципе, большинство языков программирования могут быть как интерпретируемыми, так и компилируемыми. Однако традиционно интерпретируются сравнительно простые языки, например Бейсик (хотя существуют и версии компиляторов этого языка). Традиционно компилируемые языки имеют средства алгоритмической отладки (отладчики или трассировщики), позволяющие проводить среди множества функций и пошаговую отладку программ. Этот режим эквивалентен интерпретации программы. Основным преимуществом компилятора является скорость выполнения программ, поэтому большинство языков программирования, которые используют для создания сложных пакетов программ, применяют этот режим.

Транслятор, использующий промежуточный код был впервые разработан в университете Сан Диего (штат Калифорния) в рамках ОС UCSD p-System. Цель разработки – предложить студентам простой инструмент, используемый на различных типах компьютеров. Система использовала «псевдокод» – p-код, который является чем-то средним между машинным и исходным языками. Он распознавался любым микропроцессором при условии, если последний имел соответствующий интерпретатор. Любая программа в рамках этой системы запускалась на различных типах компьютеров, таких, как Apple, DEC-PDP11, Osborn.

Однако эта система являлась значительно более медленной, поскольку она должна была «переводить» команды, которые поступают на микропроцессор. Она имела хождение в университетских кругах, но практически не использовалась в профессиональных задачах.

Интерес к промежуточному коду значительно возрос в связи с сетевыми приложениями, требующими платформенной независимости запускаемых приложений. Практически все современные инструментальные средства сетевого программирования, такие, как Java, C# и т.п., используют промежуточный код, который имеет общее название байт-код. Microsoft использует промежуточный код MSIL (Microsoft intermediate Language). Одна из первых использовала этот подход фирма Sun при разработке языка сетевого программирования Java. Язык Java использует так называемые аплеты – небольшие программы, запускаемые на Web-клиенте при обнаружении последним вызова в HTML-программе отображаемой страницы. При вызове аплета его промежуточная форма загружается клиентом с сервера и интерпретируется. Результат работы отображается в виде Web-страницы. Аплеты работают под управлением броузера и не зависят от конкретного типа компьютера и его ОС. При этом браузер (Hot Java) пишется на языке Java.

Кроме всего прочего, аплеты обеспечивают невозможность воздействия компьютерных вирусов. Обеспечение высокого уровня безопасности достигается за счет ограничения объема операций, которые может выполнить аплет. Он не может читать и писать файлы пользователя.

Кросс-системы, устанавливаемые на больших и мини-ЭВМ, используются при программировании микросистем с ограниченными

инструментальными возможностями. Так, кросс-компиляторы позволяют проводить отладку ПО для микроЭВМ с применением расширенных возможностей больших универсальных ЭВМ.

Редакторы связи (LINKEDITOR), или компоновщики, у которых основная задача – сборка программы готовой к исполнению, подключают к основной программе внешние модули как системных библиотек, включенных в состав соответствующих трансляторов, так и внешних библиотек, включая модули, разработанные самим пользователем.

По аналогии со связыванием внутри объектов (атрибутов данных и методов), о которых говорилось в предыдущей главе, компоновка (или связывание) при подготовке исполнимой программы может быть статической, когда код необходимых для работы программы библиотечных функций физически добавляется к коду оттранслированной программы для получения загрузочного модуля, что является традиционным способом сборки. При этом, как уже указывалось (при обсуждении связывания объектов), связывание может быть динамическим, когда в результирующем загрузочном модуле поставляются лишь ссылки на код необходимых библиотечных функций, сам код реально добавляется к загрузочному модулю только при его исполнении (на время его исполнения). При статическом связывании (как и в случае с объектами) загружаемая (исполнимая) программа получается очень большого размера. Поэтому подавляющее большинство современных ОС использует динамическое связывание, несмотря на то, что при этом начальная загрузка процесса на исполнение происходит медленнее, чем при статическом связывании из-за необходимости поиска и загрузки кода необходимых библиотечных функций.

К обрабатывающим программам относятся также программы обслуживания библиотек и их профилактики. Эти программы представляют собой логический уровень ввода-вывода, непосредственно обращенный к пользователю.

Широкое распространение на мини-ЭВМ и ПК, имеющих жесткие диски большой емкости, получили так называемые файловые системы, представляющие собой совокупность каталогов и файлов, хранимых на внешних носителях.

Файл – поименованная область на внешнем носителе (диске или магнитной ленте), в которой может храниться исходный текст про-

граммы, программы в машинном коде, готовые к исполнению, или данные для ее работы. В файлах могут содержаться любые текстовые документы и числовые данные, закодированная табличная графическая и любая другая информация. Имя файла, как правило, содержит расширение, указывающее на его тип (исходный текст, объектный модуль, исполнимая программа, документ и т.п.).

Каталог (*директория*) – оглавление, это файл, содержащий информацию о файлах (имя, расширение, размеры в байтах, дату и время создания и т.д.), и другие каталоги, называемые подкаталогами (подчиненные каталоги). Так, образуется иерархическая древовидная структура, в основании которой находится корневой каталог. Такая структура дает возможность выделить:

в отдельные группы логически однородные файлы;

отдельное пространство на диске для каждого из пользователей данной ЭВМ;

все файлы, относящиеся к какой-либо подсистеме, текстовому редактору, транслятору языка и т.п.

Кроме перечисленного, каталоги обеспечивают защиту файлов и их быструю визуализацию, исключив работу с очень большим списком в случае единого описания.

Термин «файловая система» – совокупность каталогов и файлов, хранящихся на внешних носителях компьютера. Физически файлы подразделяются на: обычные, каталоги, архивированные, Hidden (скрытые), защищенные (R – чтение, read; W – запись, write; E – исполнение, Execute; D – удаление, Delete). Логически они подразделяются на: текстовые (txt, For, Pas, C, ASS и т.п.), графические, объектные (obj), исполнимые (exe, svs, com). Основные манипуляции, производимые с файлами, следующие: open – открыть (подготовить к обращению), close – закрыть (запретить дальнейшее обращение), create – создать (обеспечить формирование нового файла), destroy – разрушить (удалить файл), copy – копировать (создать еще один экземпляр файла), rename – переименовать (изменить имя файла), list – вывести содержимое файла.

Утилиты (вспомогательные программы) предназначены для копирования, перемещения, печати данных. Эта группа программ позволяет организовать удобства при всевозможных обменах данными.

Перечисленные программы ОС могут поглощать значительные ресурсы ЭВМ. Если управляющие программы загружаются и практически постоянно находятся в ОП во время работы ЭВМ, то в зависимости от размещения оставшейся части ОС подразделяются на: ПЛОС – перфоленточные ОС; ДОС – дисковые ОС; ОС – размещение в ОП.

Первый тип практически вышел из употребления в настоящее время. ДОС применяются в мини-, микроЭВМ, персональных компьютерах и рабочих станциях, ОС – характерны для больших ЭВМ.

По способу взаимодействия с пользователем различные ОС обеспечивают три режима работы:

1) режим пакетной обработки, наиболее эффективный с точки зрения использования ресурсов ЭВМ, но не очень удобный для пользователя;

2) режим диалога с пользователем:

однопользовательский, при котором один программист полностью монополизирует ЭВМ (приемлемо для ПК);

многопользовательский, или режим разделения времени (характерен для ЭВМ и ПК большой мощности);

дистанционный (или режим теледоступа в системах коллективного пользования);

3) режим реального времени, при котором время выполнения программы на ЭВМ обусловливается внешними обстоятельствами (этот режим связан с автоматизацией различных технологических процессов, управлением различными объектами, системами наведения и т.п.).

ОС составляют основу общесистемного ПО, но, отнюдь, не исчерпывают его полностью.

Необходимой составной частью любой универсальной ЭВМ являются системы обеспечения нормального функционирования или комплексы программ технического обслуживания, в которые входят:

средства генерации ОС;

контрольно-наладочные тесты;

комплекты диагностических программ.

Существует очень большое количество типов ОС с различными функциональными возможностями, которые обычно связаны с техническими возможностями ЭВМ. Среди них можно выделить од-

нопользовательские однозадачные; ОС пакетной обработки; однопользовательские многозадачные; многопользовательские; многозадачные или ОС с разделением времени; ОС реального времени; ОС, ориентированные на работу с базами данных и т.п.

Как правило, операционные системы имеют открытую модульную структуру и по мере необходимости в них могут включаться дополнительные функции.

Несмотря на большое разнообразие ОС, создаваемых различными фирмами, имеется несколько типов этого программного обеспечения, которые получили наибольшее распространение в мире. Именно поэтому проведем краткий обзор этих систем.

6.3. Разновидности построения ОС для различных СОД, сетевые операционные системы

Большое разнообразие и различие функциональных возможностей ОС больших и малых ЭВМ 70-х – 80-х годов объясняется различием в уровне технических средств этих машин. Значительное увеличение объемов ОП во всех без исключения типах ЭВМ, появление ВЗУ с малым временем доступа и практически неограниченной емкостью позволило, следуя принципу открытых систем, разработать единые требования к ОС:

расширяемость (модульность) – написание кода таким образом, чтобы можно было легко внести дополнения и изменения, не нарушая целостности системы;

переносимость – перенос кода с аппаратной платформы одного типа на аппаратную платформу другого типа;

надежность и отказоустойчивость – защищенность системы как от внутренних, так и от внешних ошибок;

совместимость – ОС должна иметь средства для выполнения прикладных программ, написанных для других ОС;

безопасность – наличие в ОС средств защиты от несанкционированного доступа;

производительность – хорошее быстродействие и время реакции.

Эти общие принципы так же, как и развитие технических средств, приводят к сближению ОС для различных типов ЭВМ. Более того появляются ОС, используемые на большинстве совре-

менных ЭВМ. Типичным примером такой ОС является UNIX. Тем не менее, имеются существенные отличия ОС однопроцессорных, многомашинных и многопроцессорных ЭВМ, не говоря уже о сетевых ОС.

Рассмотрим некоторые принципиальные особенности перечисленных ОС.

Однопроцессорные ОС имеют традиционный набор модулей. Управляющие программы являются резидентными и постоянно находятся в ОП. Обрабатывающие программы и утилиты, как правило, располагаются на системных ВЗУ (как и программы пользователей) и по мере необходимости загружаются в ОП. Значительным достижением в развитии локальных ОС явилась реализация конструкции виртуальной памяти и виртуальной машины. Эти концепции были реализованы в конце 70-х годов в ОС фирмы DEC VMS (Virtual Memory System) для ЭВМ VAX и MVS (Multiple Virtual Storage) фирмы IBM для ЭВМ IBM 360. На сегодняшний день практически все ОС (в том числе и для настольных систем) включают элементы виртуальной организации памяти, которая позволяет использование прикладными процессами практически неограниченного объема памяти.

Существует два наиболее распространенных способа реализации виртуальной памяти: страничная (блоки постоянной длины) и сегментная (блоки переменной длины – логические). Эти блоки располагаются во внешней памяти и по мере необходимости загружаются в ОП. Перевод виртуальных адресов (на дисках) в реальные (ОП) называется динамическим преобразованием адресов и выполняется системой прозрачно (невидимо) для пользователя.

ОС для мультипроцессорных систем определяются их конфигурацией.

На рис. 6.6 представлена мультипроцессорная (многомашинная) система со слабо связанными процессорами, которая иногда называется кластерной сетью. Каждый процессор имеет свою собственную ОС с функциями однопроцессорной системы. Некоторое отличие таких ОС от обычных однопроцессорных – возможность посылки запросов и возможность обмена между процессорами.

На рис. 6.7 представлены два возможных способа организации ОС в истинно мультипроцессорных системах (или системах с сильно связанными процессорами). В организации главный-

подчиненный (см. рис. 6.7, *а*) все процессоры могут использовать общие ресурсы ЭВМ – ОП и ВУ. Однако распределением ресурсов занимается только один процессор, что может вызвать его перегрузку и несбалансированность использования ресурсов. Но, тем не менее, организация ОС сравнительно проста и близка к однопроцессорной.

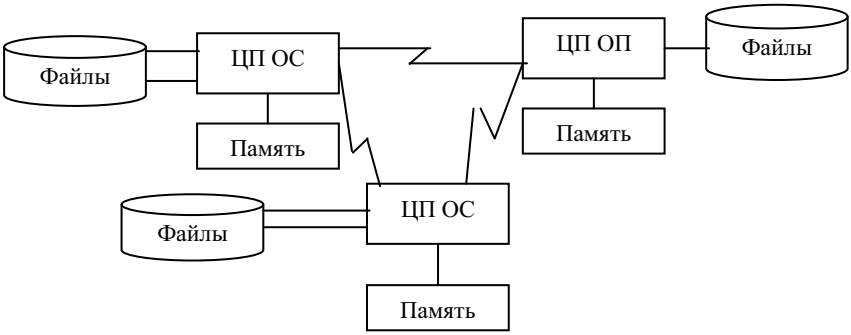


Рис. 6.6. Мультипроцессорные системы со слабо связанными процессорами

В симметричных ОС (см. рис. 6.7, *б*) различные части ОС могут выполняться разными процессорами. Поэтому разработки подобных систем значительно сложнее, однако эффективность использования ресурсов в этой системе может оказаться выше.

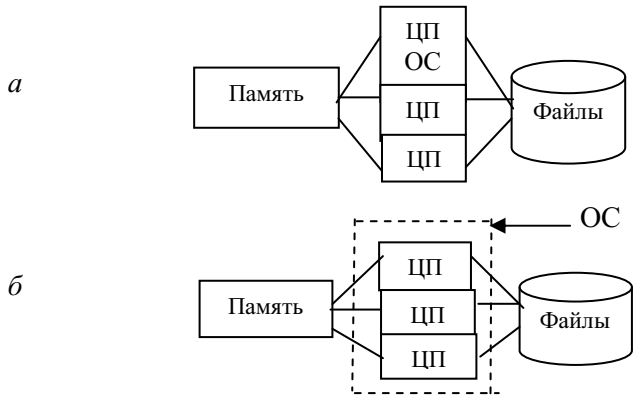


Рис. 6.7. Мультипроцессорные системы с сильно связанными процессорами с обработкой: *а* – типа главный-подчиненный; *б* – симметричной

Сетевые ОС получили широкое распространение в связи с повышением распространения локальных вычислительных сетей. Их функциональные возможности значительно больше, чем у автономных ОС.

В сетевой ОС можно выделить:

средства управления локальными ресурсами компьютера – все, что входит в традиционный состав автономной ОС, рассмотренной выше;

средства предоставления собственных ресурсов и услуг в общее пользование – серверная часть ОС (сервер); сюда входят блокировка файлов и записей, необходимая для совместного использования многими абонентами, обработки запросов удаленного доступа, управления очередями и т.п.;

средства запроса доступа к удаленным ресурсам и их использование – клиентская часть (сюда входят распознавание и перенаправление в сеть запросов к удаленным ресурсам, прием ответов от серверов и преобразование их в локальный формат и многое другое);

коммуникационные средства ОС, с помощью которых происходит обмен сообщениями в сети – адресация, буферизация, маршрутизация и т.п.

В зависимости от функций, возлагаемых на конкретный компьютер, в его ОС может отсутствовать либо клиентская, либо серверная части.

Определенными сетевыми функциональными возможностями обладают все современные локальные ОС. Даже в такой скромной по своим возможностям MS-DOS, начиная с третьей версии появились такие встроенные функции, как блокировка файлов и записей, необходимые для совместного доступа к файлам.

Однако более эффективным представляется путь разработки ОС, изначально предназначенной для работы в сети. Сетевые функции таких ОС глубоко встроены в основные модули системы, что обеспечивает логическую стройность, простоту эксплуатации и модификации, а также высокую производительность.

На практике сложились два основных подхода к построению сетевых ОС в зависимости от распределения функций между компьютерами сети. Сетевые ОС, а следовательно, и сети делятся на два класса: одноранговые сети и сети с выделенными серверами.

Если выполнение каких-либо серверных функций является основным назначением компьютера (например, предоставление файлов в общее пользование или организация совместного доступа факса, или предоставление всем пользователям сети возможности запуска на данном компьютере своих приложений), то такой компьютер называется выделенным сервером. В зависимости от того, какой ресурс сервера является разделенным, он называется файл-сервером, факс-сервером, принт-сервером, сервером приложений и т.п.

На выделенных серверах устанавливаются ОС, специально организованные для выполнения тех или иных серверных функций. Поэтому в сетях с выделенными серверами чаще всего используются сетевые ОС, отличающиеся возможностями серверных частей. Например, сетевая ОС Novell NetWare имеет серверный вариант, оптимизированный для работы в качестве файл-сервера, а также варианты оболочек для рабочих станций с различными локальными ОС (DOS, Mac OS, OS/2, UNIX, Windows). Другой пример ОС, ориентированной на построение сети с выделенным сервером, – ОС Windows NT. В отличие от NetWare оба варианта данной сетевой ОС – Windows NT Server (для выделенных серверов) и Windows NT Workstation (для рабочей станции) могут поддерживать функции и клиента, и сервера. Но серверный вариант Windows NT имеет больше возможностей для предоставления ресурсов своего компьютера другим пользователям сети (выполняет более широкий набор функций, поддерживает большое количество одновременных соединений с клиентами, имеет развитые средства защиты). В качестве локальных ОС могут использоваться также DOS, OS/2, Mac OS и др.

Несимметричность ОС в сетях с выделенным сервером сопровождается, как правило, несимметричностью аппаратуры. Для выделенных серверов используются более мощные компьютеры, рассмотренные в гл. 3, а также мейнфреймы и даже суперЭВМ.

В одноранговых сетях все компьютеры равны в правах доступа к ресурсам друг друга: каждый пользователь может объявить какой-либо ресурс своего компьютера разделяемым. В таких сетях на всех компьютерах устанавливается одна ОС, которая предоставляет всем компьютерам в сети равные возможности. Одноранговые сети могут быть построены, например, на базе ОС LANtastic, Personal Ware, Windows for Workstation, Windows NT Workstation.

Одноранговые сети проще в организации и эксплуатации, однако они применяются, в основном, для объединения небольших групп пользователей, не предъявляющих больших требований к объемам хранимой информации, ее защищенности от несанкционированного доступа и скорости доступа. При повышенных требованиях к этим характеристикам более подходящими являются сети с выделенными серверами, которые в ряде случаев обеспечивают наиболее эффективный режим работы сети, называемый клиент-сервер. В этом режиме работа сети распределяется между машиной клиента (который формирует запрос на обработку и принимает его результаты) и сервером, производящим требуемую обработку значительно быстрее ввиду значительно больших, чем у клиента, ресурсов и производительности.

При этом значительно уменьшается загрузка сети, по сравнению с работой в режиме файл-сервера, так как не требуется пересылка клиенту самих обрабатывающих программ и больших файлов данных для использования их клиентом при обработке на собственной ЭВМ.

Обеспечение защиты информации от несанкционированного доступа и контроль за ним являются неотъемлемой частью сетевых ОС.

Основы стандартов в области безопасности сетевых ОС сформулированы в «Критериях оценки безопасности компьютерных систем», разработанных в 1983 г. Национальным центром компьютерной безопасности США (NCSC – National Computer Security Center).

В соответствии с иерархией надежности систем, нижний уровень помечается как D, а верхний – как A:

в класс D попадают системы, оценка которых выявила их несоответствие требованиям всех других классов;

уровень C делится на два подуровня – уровень C1, предохраняющий данные от ошибок пользователей, но не от действий злоумышленников, и более строгий уровень – C2 (на уровне C2 должны присутствовать средства секретного входа и контроля перед тем, как им будет разрешен доступ к системе, средства учета и наблюдения (auditilly) обеспечивают возможность обнаружить и зафиксировать важные события, связанные с безопасностью или любые попытки создать, получить доступ или удалить системные ресурсы и т.п.);

системы уровня B основаны на помеченных данных и распределении пользователей по категориям, т.е. реализуется мандатный

контроль доступа (каждому пользователю присваивается рейтинг защиты и он может получать доступ к данным только в соответствии с этим рейтингом);

уровень А является самым высоким уровнем безопасности – он требует в дополнение ко всем требованиям уровня В выполнения формального, математически обоснованного доказательства соответствия системы требованиям безопасности.

В нашей стране существует аналогичный нормативный документ: «Средства вычислительной техники. Защита от несанкционированного доступа к информации. Показатели защищенности от несанкционированного доступа к информации», согласно которому средства вычислительной техники подразделяются на шесть классов (первый класс – наиболее защищенный).

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что следует понимать под программой и программным обеспечением? В чем различие Saft Ware, Firm Ware, Share Ware, Free Ware? Какие три состояния (формата) может иметь программа одного и того же функционального назначения?

2. Расскажите о жизненном цикле программного обеспечения, назначении каждой фазы. Перечислите основные этапы фазы разработки.

3. Расскажите о разновидности программного обеспечения (ПО). Перечислите функциональные особенности различных видов ПО и способы организации прикладного ПО.

4. Какие основные функции выполняет операционная система (ОС)? Назовите ее составные части, место размещения и способы загрузки.

5. Перечислите разновидности трансляторов. Укажите области их применения. Какова роль компоновщиков при подготовке программ? Что имеется в виду под динамическим связыванием?

6. Что такое файловая система? Назовите ее основные функции, разновидности файлов и их атрибуты.

7. Расскажите об особенностях ОС различных вычислительных систем. Перечислите основные функции сетевых ОС и их разновидности.

ГЛАВА 7. ОРГАНИЗАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В ПЕРОНАЛЬНЫХ КОМПЬЮТЕРАХ IBM-КЛОНА

7.1. Общая структура программного обеспечения

Все компоненты программного обеспечения, описанные в гл. 6, присутствуют и в структуре ПО ПК. Однако система программного обеспечения этого вида ЭВМ имеет ряд специфических особенностей, отражающих широкий спектр применений этих машин и их массовое внедрение в самые разнообразные сферы человеческой деятельности, досуга и развлечений.

История развития ПО персональных компьютеров IBM-клона началась с августа 1981 г., когда появилась операционная система PC-DOS. За сравнительно небольшой период времени различными фирмами для этих систем разработано огромное количество программного обеспечения разнообразного функционального назначения. Темпы развития ПО ПК IBM значительно выше темпов создания ПО всех предыдущих типов ЭВМ.

Массовое внедрение ПК и большое количество фирм, специализирующихся на разработке ПО для них, привели, с одной стороны, к значительному снижению цен на программный продукт (относительно ПО больших и мини-ЭВМ), с другой стороны, стремительное совершенствование технических средств имело следствием увеличение объема и сложности программ, появление сервисных программ, создающих так называемую «дружественную» среду для пользователя. Так, если первые ОС для ПК IBM требовали 30 – 50 Кбайт на диске, занимая незначительную часть ОП объема в 1 Мбайт, в 2000 г. требовалось ОП не менее 32 – 64 Мбайт и размещения на диске не менее 1 Гбайта, то после 2000 г. – ОП более 1 Гбайта, НЖМД более 100 Гбайт.

Все это приводит к удорожанию ПО ПК и создает определенные трудности его адаптации пользователями в связи с необходимостью изучения описаний большого объема и быстро изменяющихся версий различных программных продуктов.

Структура ПО может быть условно представлена в виде трех различных функциональных частей:

системное ПО;

инструментальное ПО;

прикладное ПО (или ППП).

К системному ПО относятся операционные системы, в состав которых входят управляющие части традиционной ОС (см. гл. 6), системные утилиты, а также сетевое ПО. Этот вид программного обеспечения носит универсальный характер, не связан с конкретным применением ПК и выполняет традиционные функции, характерные для системного ядра ОС: планирование и управление задачами, осуществление посреднических функций между физическим уровнем аппаратуры и логическим уровнем пользователя, управления вводом-выводом и т.п.

Утилиты (вспомогательные или служебные программы) облегчают работу пользователя при проверке, наладке и настройке ПК. Существует два альтернативных направления использования таких программ: интеграция с ОС и автономное функционирование. Первое, наиболее значительное направление составляют так называемые системные утилиты, среди которых можно условно выделить следующие классы:

сервисные утилиты, предназначенные для эффективной работы с периферийными устройствами и файлами в среде ОС;

тестовые утилиты для оценки и диагностики параметров аппаратных и программных средств;

утилиты загрузки шрифтов национального языка, в качестве дополнительных (к английскому) при работе с устройствами ввода-вывода (клавиатура, принтер и т.п.);

интерфейсные оболочки данной ОС;

антивирусы и т.д.

Инструментальное ПО выделено из состава ОС ввиду особенностей применений ПК, большая часть которых работает на уровне приложений (ППП) и не нуждается в подобных средствах. Однако для разработчиков ПО основными инструментальными средствами являются средства поддержки процесса программирования для ПК, наиболее распространенных в настоящее время.

Инструментальные средства ПК имеют ряд особенностей, связанных с созданием средоориентированных систем программирования.

Наблюдается устойчивая тенденция интегрировать компилятор и средства поддержки программирования, содержащие редактор текстов, макропроцессор, редактор связи (Linker), библиотекарь,

отладчик (Debugger), компоновщик, инструктор (Help), в единую систему, представляющую собой единый объектно-ориентированный командный интерфейс в виде меню и окон для манипулирования всеми перечисленными средствами.

Первой по этому плодотворному пути начала работать фирма Borland с семейством Turbo-реализаций наиболее популярных языков программирования. Успех такого подхода обеспечил ей заметную конкурентоспособность по отношению к мощному сопернику – фирме Microsoft, которая в своих последующих версиях также следует по пути создания интегрированных сред. Реализация языков в средах, как правило, содержит большое количество расширений (по сравнению со стандартными определениями ANSI). В трансляторы включается большое количество дополнительных функций и библиотек, значительно расширяющих возможности языков.

Следует отметить, что инструментальные системы ПК при всех их достоинствах связаны с ограничениями переносимости разработанного ПО на различные платформы.

Популярность и повсеместное распространение ПК определяется прикладным ПО.

Разработка пакетов прикладных программ разного назначения значительно изменила характер человеческой деятельности в самых различных сферах: от научных и проектных организаций до банков, бухгалтерий, отделов кадров, офисов и т.п.

Одно из распространенных применений ПК – *подготовка текстов*. Диапазон текстов простирается от простых деловых писем, статей и отчетов до газет, журналов и монографий, целиком подготавливаемых на ПК. Поэтому диапазон ППП этого типа может варьироваться от простых текстовых редакторов до комплексных программных систем оперативной полиграфии, позволяющих комбинировать в едином документе текстовые и графические образы.

Табличные процессоры, или электронные таблицы, – одно из самых первых и массовых применений ПК. В основе табличного процессора лежит механизм потоковых вычислений над элементами двумерной матрицы.

Быстрое совершенствование технических характеристик ПК сделало возможным создание и эксплуатацию *систем управления*

базами данных (СУБД), составляющих основу автоматизированных систем управления (АСУ).

Преодоление кризиса в проектировании сложных технических систем непосредственно связано с разработкой *систем автоматизированного проектирования*, элементами которого являются ППП для различных направлений техники и технологии.

Большое разнообразие *математических пакетов,* реализующих как методы вычислительной математики, так и символьные вычисления, активно используется в сфере научно-технических приложений.

Проблемы искусственного интеллекта представлены в виде *экспертных систем*.

Одно из направлений в развитии ППП – создание интегрированных сред, объединяющих наиболее распространенные приложения в рамках единых, стандартных для среды интерфейсов, обеспечивающих высокую степень взаимодействия между приложениями. Сюда можно отнести одну из первых таких разработок фирмы Microsoft Works, а также наиболее популярную в последнее время среду Office. Основу этих приложений составляют редакторы текстов, электронные таблицы и СУБД.

7.2. Краткая характеристика ОС, применяемых в ПК

До последнего времени для ПК IBM-клона применялись три типа локальных ОС:

однопользовательская однозадачная DOS;

однопользовательские многозадачные OS/2, Windows 95, 98, 2000, ...;

многопользовательская многозадачная UNIX.

Начальная ОС для ПК – DOS, которая появилась в 1982 г. и до 1988 г. была единственной ОС для этого типа ЭВМ и в некотором смысле основой при разработке последующих ОС. Последующие операционные системы (OS/2, Windows) представляют собой дальнейшее развитие DOS, недостатки которой довольно быстро выявились и связаны с опережающим развитием технических средств ПК. Все эти ОС в определенной степени возникли в недрах фирмы Microsoft и IBM имеют ряд общих черт, характерных для дисковых ОС.

1. Обеспечение автоматического запуска ОС с помощью комплекса программ BIOS (Basic Input/Output System), хранящихся в постоянном запоминающем устройстве (ПЗУ или ППЗУ) и представляющих так называемое аппаратно-ориентированное ПО (Firm ware), поставляемое изготовителем ЭВМ. BIOS начинает работать после включения питания ПК с проверки комплектности ЭВМ (проверки ОП, наличие НЖМД и НГМД, наличие клавиатуры) и сообщения о неисправностях. Если с комплектацией все в порядке, то BIOS завершает свою работу считыванием из специальной (системной) области диска программы-загрузчика, которая и осуществляет дальнейшее размещение ОС в ОП и ее запуск.

2. Физическая организация файловой системы во всех указанных ОС имеет общие принципы. Поверхность жесткого диска рассматривается как трехмерная матрица, измерениями которой являются номера поверхности, цилиндра и сектора. Под цилиндром понимается совокупность всех дорожек, принадлежащих разным поверхностям и находящихся на равном удалении от оси вращения. Данные о том, в каком месте диска записан тот или иной файл, хранятся в системной области диска в специальных таблицах размещения файлов (FAT-таблицах (File Allocation Table)).

Наименьшей физической единицей хранения данных является сектор. Размер сектора равен 512 байт. В операционных системах DOS, OS/2, Windows 95 для адресации файлов используются 16-разрядные поля (FAT 16). Такое адресное пространство не позволяет адресовать каждый сектор для дисков, объем которых превышает 32 Мбайт. В связи с этим группы секторов условно объединяют в кластеры. Кластер – наименьшая единица адресации данных на диске. Размер кластера в отличие от размера сектора не фиксирован и зависит от емкости диска. Так, для дисков объемом 1 – 2 Гбайта длина кластера составляет 32 Кбайта (64 сектора). Это приводит к нерациональному расходу рабочего пространства диска, поскольку любой файл (даже очень маленький) полностью окупирует целый кластер. Даже если файл достаточно велик, в его конце может образоваться остаток, расходующий целый кластер, занимая при этом незначительную часть его. В результате потери, связанные с неэффективностью файловой системы, могут составлять 25 – 40 % полной емкости диска.

Частично эта проблема решается разбиением физического пространства диска на ряд разделов – логических дисков, каждый из которых имеет свою FAT-таблицу и представляется для пользователя как автономный, независимый диск.

Еще более радикальное решение – переход на FAT-таблицы с 32-разрядным адресным полем (FAT 32). Такой переход начался с ОС Windows 98. Для дисков до 8 Гбайт эта система обеспечивает размер кластера 4 Кбайт (восемь секторов).

3. Логическая организация функции файловой системы достаточно традиционна. Для пользователя она представляется в виде иерархической, древовидной структуры. Метод именования файлов заимствован из системы UNIX, полное имя файла состоит из цепочки имен всех охватывающих каталогов (которые в ОС Windows 95, 98, 2000, ... называются папками), начиная от корня файловой системы, завершаемой локальным именем файла.

Каталог является файлом. В качестве разделителя используется левый слэш (в UNIX – правый слэш). Для сокращенного именования, как и в UNIX, введено понятие текущего каталога.

К функциям обслуживания файловой структуры относятся операции, происходящие под управлением ОС:

создание файлов и присваивание им имен;

создание каталогов (папок) и присваивание им имен;

переименование файлов и каталогов (папок);

копирование и перемещение файлов между дисками компьютера и между каталогами (папками) одного диска;

удаление файлов и каталогов (папок);

навигация по файловой структуре с целью доступа к заданному файлу, каталогу (папке);

управление атрибутами файлов.

Кроме имен и расширений имен файлов ОС хранит для каждого файла дату его создания (изменения) и несколько флаговых величин, называемых атрибутами файла. Атрибуты – дополнительные параметры, определяющие свойства файлов. ОС позволяет их контролировать и изменять. Состояние атрибутов учитывается при проведении автоматических операций с файлами.

Основных атрибутов четыре:

1) только для чтения (Read only);

2) скрытый (Hidden);

3) системный (System);

4) архивный (Archive).

Рассмотрим некоторые особенности ОС, применяемых для ПК IBM PC-клона.

DOS. Как уже упоминалось, первая 16-разрядная ОС для ПК IBM, имевшая колоссальный успех, была разработана в 1981 г. сотрудником фирмы Seattle Computer Products Тимом Петерсоном, который выполнял работу в рамках фирмы Microsoft. Эта ОС явилась развитием своего логического предшественника – очень популярной (по тем временам) операционной системы CP/M, применяемой для 8-разрядных микроЭВМ. При поставке DOS фирмой-разработчиком она называлась MS-DOS, при поставке фирмой IBM – PC-DOS. Множество лицензий на MS-DOS были затем проданы другим изготовителям и имели собственные названия: так, фирма Digital Research (разработчик CP/M) разработала DR-DOS.

DOS являлась единственной ОС, применяемой для ПК IBM PC вплоть до 1987 г. и фактически основной до 1995 г., поскольку появившаяся в 1987 г. OS/2 имела ограниченное распространение. За этот период выпущено шесть версий ОС: от MS-DOS 1.0 (1981 г.) до MS-DOS 6.0 (1993 г.). Продолжающаяся в этот период разработка DOS связана, прежде всего, с адаптацией к интенсивно развивающимся техническим средствам ПК, постоянно увеличивающимся объемом ОП, появлением новых типов НГМД и НЖМД, поддержкой CD-ROM, поддержкой средств сетевого взаимодействия и т.п. Очень важным элементом в совершенствовании DOS явилось создание полноэкранного, алфавитно-цифрового (Norton Commander), а затем и графического пользовательского интерфейса, последняя версия которого в виде оболочки Windows 3.1 явилась промежуточным этапом к переходу на 32-разрядные ОС фирмы Microsoft.

Независимо от версий, работу DOS обеспечивают следующие три компонента:

базовая система ввода-вывода (BIOS, или RIOS), записанная в ПЗУ программа (программа поставляется производителями ЭВМ и выполняет функцию обеспечения нормального функционирования: проверку комплектности, тестирование различных устройств ПК, инициализацию загрузки DOS и ряд других функций, эта программа не является файлом и фактически не входит в состав DOS);

ядро DOS, содержащее в виде скрытых файлов: блок первоначальной загрузки (bootstrap), располагаемый в первом секторе системного диска (считываемый с помощью BIOS) и обеспечивающий загрузку в память машины операционной системы – IO.SYS (или IBMBIO.COM), так называемую базовую систему ввода-вывода с программным обслуживанием периферийных устройств (расширение и исправление BIOS), а также файл MSDOS.SYS (IBMDOS.COM), содержащий программы управления файлами, памятью, запуском программ и т.д. (этот файл иногда называется модулем управления прерываниями, поскольку обращение к программным средствам (функциям DOS) из программы пользователя осуществляется с помощью программных прерываний);

файл COMMAND.COM, или командный процессор (интерпретатор), в функции которого входит прием, проверка команд, вводимых пользователем с клавиатуры, и их выполнение (существует три категории команд DOS – резидентные или внутренние, постоянно находящиеся в ОП, полурезидентные, которые могут (по мере необходимости) «затираться» прикладными программами, и внешние, вызываемые по мере необходимости из внешней памяти).

Подробное описание MS-DOS различных версий с перечнем команд и правилами их использования широко представлено в различной литературе.

Появившаяся в 1981 г. для IBM PC операционная система DOS предоставляла небывалые по тем временам возможности в мире микрокомпьютеров (если учесть что ОП, которая даже для миникомпьютеров по тем временам традиционно равнялась 64 Кбайт, была превышена на порядок, а объем дисковой памяти от сотен килобайт был доведен до десятков мегабайт). Казалось, что ОП в 640 Кбайт хватит на несколько десятков лет. Однако прогресс технических средств превзошел все мыслимые прогнозы.

Уже появление в 1984 г. процессора Intel-80286, обеспечивающего адресацию 16 Мбайт ОП, аппаратную поддержку механизма виртуальной памяти для доступа к 1 Гбайту внешней памяти, аппаратную поддержку механизма мультизадачного режима с возможностью взаимоизоляции ресурсов и взаимодействия задач (процессов), выявило ограниченные возможности 16-разрядной DOS в эффективной поддержке новых технических средств. DOS обеспечивает прямую адресацию не более 1 Мбайта и абсолютно не подго-

товлена к мультизадачному режиму работы. Дальнейшее развитие DOS связано с попыткой преодолеть эти ограничения, однако изначально заложенные программы значительно снижали эффективность использования новых ресурсов МП, связанных с многозадачностью и использованием оперативной и внешней памяти большого объема.

UNIX. Среди многопользовательских ОС UNIX занимает особое место. Предназначенная для профессионалов в области вычислительной техники эта мощная и совершенная система имеет рациональный и упорядоченный синтаксис. Она значительно сложнее чем DOS и ее производные, но и возможности ее значительно превышают возможности DOS.

UNIX – рекордсмен-долгожитель среди всех существовавших и всех существующих ОС.

Появление UNIX было вполне закономерным. В конце 60-х годов у всех крупных конструкторов компьютеров имелись мощные многозадачные, многопользовательские ОС в результате их непрерывного совершенствования. Они постоянно модернизировались по мере развития технических средств и появления новых приложений. После многих лет усовершенствований функциональные возможности этих систем значительно изменились, однако их исходная структура сохранилась и становилась все большим препятствием на пути модернизации всей системы. К тому же написанные на Ассемблере эти ОС могли использоваться только на одном типе компьютеров.

В 1969 г. Томсон и Ричи – два программиста фирмы Bell Laboratories, филиала могущественной корпорации American Telephon & Telegraph (AT&T) решили создать для внутренних нужд своей фирмы совершенную ОС, простую в использовании, одним словом «дружескую», и как можно меньше зависящую от типа ЭВМ, на которой она будет работать. Специально для этих целей Денис Ричи разработал язык С. Так и появилась ОС UNIX. Первоначально эта система разрабатывалась для больших ЭВМ. Внедрение UNIX в ПК связано с новыми возможностями и увеличением мощности микропроцессоров, начиная с модели 80386 и выше. Наибольшее распространение эти ОС получили в классе 32- и 64-разрядных рабочих станций, серверов и суперсерверов. С учетом заметной тен-

денции к слиянию этих классов ЭВМ движение системы UNIX на «смежную» территорию ПК выглядит закономерным.

В чем же причины популярности UNIX?

1. Код системы на 80 – 90 % написан на языке С (≈ 10000 строк на С и 1000 строк ASS), что сделало ее простой для понимания, изменений и переноса на другие платформы. По оценкам одного из создателей UNIX Дениса Ричи, система на языке С имела на 20 – 40 % больший размер, а производительность ее была на 20 % ниже аналогичной системы, написанной на Ассемблере. Однако ясность и переносимость (а в результате – и открытость системы) сыграли решающую роль в ее популярности. Система легко расширяема и настраиваема.

2. UNIX – многозадачная многопользовательская система с широким спектром услуг. Один мощный сервер может обслуживать запросы большого количества пользователей, выполняя различные функции: работать как вычислительный сервер, обслуживать сотни пользователей как сервер баз данных, как сетевой сервер, поддерживающий важнейшие приложения в сети (telnet, ftp, электронную почту, службу имен DNS и т.д.), может использоваться и как сетевой маршрутизатор.

3. Наличие стандартов (несмотря на многообразие версий UNIX основой всего семейства являются принципиально одинаковая архитектура и ряд стандартных интерфейсов) позволяет обслуживать различные версии без большого труда опытным администраторам, пользователям переход на другою версию и вовсе может оказаться незаметным.

4. Используется единая, легко обслуживаемая файловая система (основные концепции которой в дальнейшем использовались и в ОС ПК). Файловая система – не только доступ к файлам, хранящимся на диске. Через унифицированный интерфейс файловой системы осуществляется доступ к терминалам, принтерам, магнитным лентам, сети и даже памяти.

5. Семейство протоколов TCP/IP, стандартизованных в 1983 г. и являющихся основными протоколами ГВС Internet, было разработано в 70-х годах специально для ОС UNIX.

6. Имеется простой, но мощный модульный пользовательский интерфейс, содержащий в своем распоряжении набор утилит, каж-

дая из которых решает узкую специализированную задачу и при помощи которых можно конструировать сложные комплексы.

7. Обладает высокой надежностью и защищенностью системы как от случайных ошибок, так и от несанкционированного доступа.

8. Имеется очень большое количество приложений, в том числе свободно распространяемых, начиная от простейших текстовых редакторов и заканчивая мощными системами управления базами данных.

UNIX, как и любая многопользовательская система, имеет ряд характерных особенностей, отличных от традиционных локальных ОС ПК, связанных с различными уровнями защиты системы.

7.3. Инструментальное программное обеспечение ПК

На ПК, совместимых с IBM PC, реализованы практически все широко распространенные языки программирования, а многие из них имеют несколько независимых реализаций. Следует отметить ряд специфических особенностей инструментального обеспечения ПК. Как уже отмечалось, реализация языков – интегрированная среда, в которой кроме традиционных программ трансляторов и компоновщиков имеются обширные средства поддержки процесса разработки ПО. Эти среды имеют современный графический пользовательский интерфейс, удобные диагностические средства и многое другое.

Фирмы-разработчики, как правило, вводят дополнительные возможности для расширения функций тех или иных языков, значительно выходящих за рамки стандартов ANSI. По этим принципам разрабатываемые языки ПК нельзя считать машинно-независимыми в традиционном смысле этого понятия. Даже одни и те же языки, реализованные различными фирмами в различных средах, не являются совместимыми на одной и той же стандартной платформе ПК.

Однако такая несовместимость не является столь актуальной, как это было прежде. Массовое распространение и простота инсталляции любой требуемой среды значительно снижает актуальность несовместимости.

Следует, тем не менее, отметить, что освоение современных инструментальных средств, а также особенностей различных версий

одной и той же среды составляет немалую «головную боль» для пользователей.

Как и прежде, на сегодняшний день наиболее распространенными являются два семейства языков, выпускаемых фирмами Microsoft и Borland.

Среди первых версий семейство языков фирмы Microsoft имело:

макроассемблер MASM;

MS Fortran;

MS C;

MS Pascal;

MS Basic.

Стартовое семейство языков программирования фирмы Borland имело:

макроассемблер TASM;

Turbo C;

Turbo C++;

Turbo Basic;

Turbo Prolog;

Borland C++.

Из многочисленных менее популярных других языков программирования можно назвать:

Mulisp фирмы Soft.house;

Modula 2 фирмы Logitech;

Ada фирмы Artek;

Smalltalk фирмы Digitalk;

Ada фирмы Alsys;

MProlog фирмы Logicware;

Java фирмы Sun Microsystems.

Наиболее популярны и интенсивно развиваются традиционные языки программирования Бейсик, Паскаль, Си, разрабатываемые как фирмой Microsoft, так и фирмой Borland, а также поддерживаемый, в основном, фирмой Microsoft FORTRAN, который по-прежнему остается основным языком научного программирования благодаря большому количеству библиотек научных расчетов.

Динамика развития этих языков для ПК IBM и традиционные их функциональные возможности сближаются, а разговоры о модульности, структурированности и т.п. как о преимуществах некоторых из них (например, Паскаль) уже теряют смысл.

Все эти старые традиционные языки приобрели объектные черты, т.е. стали объектно-ориентированными. Первая объектно-ориентированная надстройка появилась у языка С. Затем как Microsoft, так и Borland взяли за основу Object Pascal от Apple (разработанный Никлаусом Виртом) и выпустили свои первые версии языков Quick Pascal и Turbo-Pascal. Значительное развитие получила инструментальная среда MS FORTRAN фирмы Microsoft, начальные версии которой имели скромные возможности. Microsoft FORTRAN Power Station наряду с полноценной реализацией объектно-ориентированного стандарта Фортран-95 имеет широкий набор графических библиотек, удобную инструментальную среду и т.п. Характерным расширением этих языков для ПК IBM являлась их так называемая «визуальность», т.е. появление встроенных объектов для программирования пользовательских интерфейсов в графических средах.

Первой и очень удачной реализацией «визуальных» языков явился разработанный Microsoft язык Visual Basic. Для создания более сложных программных продуктов этой же фирмой был создан Visual C ++, однако более популярными оказались версии инструментальных сред, разработанные фирмой Borland.

Интегрированные среды на основе Delphi и C++ Builder, разработанные фирмой Borland, представляют новый уровень технологии разработки ПО. Общим для обеих этих сред является то, что средства объектно-ориентированного и визуального программирования дополнены средствами доступа к наиболее распространенным (как централизованным, так и распределенным) базам данных, включая продукт фирмы Borland-InterBase. Так, в состав Delphi, основу которого составляет Turbo-Pascal, входят следующие компоненты:

мощная библиотека визуального программирования;

встроенный компилятор Object Pascal;

локальная версия Inter Base и многое другое.

Borland Database Engine (BDE) – машина базы данных Borland, организующая доступ к локальным таблицам в форматах Paradox, dBase, InterBase, а также через модуль SQL-LINK (SQL-Structured Query Language – язык структурированных запросов) доступ к SQL-серверам, таким, как Informix, Oracle, Sybase, MS-SQL и др.

C++ Builder – аналогичная интегрированная инструментальная среда для разработки более сложных приложений. Так же, как и в

Delphi, язык C++ дополнен средствами разработки графических интерфейсов и так же, как и в среде Delphi, с помощью BDE организована связь с наиболее распространенными локальными и распределенными базами данных.

Развитие сетевых информационных технологий привело в рамках инструментальных средств ПО ПК к дальнейшему совершенствованию средств сетевого программирования. Основным языком сетевого программирования является C++, сетевые вариации которого зачастую связаны с его упрощением и включением специфических функций.

Достаточно распространен разработанный на фирме SUN язык Java, в то же время Microsoft предлагает C# (Си-шарп) – простой объектно-ориентированный язык, напоминающий C++ и Java. Трансляторы Java и C# используют промежуточный код и считаются платформенно-независимыми.

Для создания гипертекстовой информации используется язык HTML и его дальнейшее развитие в виде языка XML. Совершенствование этого инструментария связано с созданием языков дескрипторного типа – VB Script (Visual Basic Script), JavaScript и т.п.

7.4. Пакеты прикладных программ

Разработка ППП является одним из основных направлений развития ПО ПК. Именно благодаря этому виду прикладного программного обеспечения ПК приобрели всеобщую популярность и стали необходимым инструментом, используемым в самых различных областях человеческой деятельности от бухгалтерии и отдела кадров до производственных и научных подразделений, спортивных мероприятий и развлекательных центров. При этом пользователь получил необходимый универсальный компьютер, не требующий знаний программирования и его разработки в большинстве массовых приложений. Это его выгодно отличало от ЭВМ предыдущих поколений и позволило многократно увеличить эффективность использования компьютерных технологий.

Широкий спектр ППП можно условно разделить на три иерархических уровня по степени распространения и частоте использования.

На нижнем уровне находятся пакеты универсального применения, не связанные с конкретными приложениями – областями применения.

Именно с этими пакетами связано массовое внедрение персональных компьютеров в самые разнообразные сферы деятельности человека.

На этом уровне, по-видимому, самыми популярными и наиболее часто используемыми являются ППП для подготовки текстов (технологии обработки текстовой информации). Диапазон применения таких ППП широк, а их разновидности определяются сферой применения от сравнительно простых текстовых редакторов (text – editor), которыми оснащаются (и оснащались ранее до появления персональных компьютеров) любая ОС и различные инструментальные среды для общения с пользователем и подготовки текстов программ с помощью полноэкранных редакторов до редакционно-издательских систем. Однако в современном представлении основными ППП для подготовки текстов являются так называемые текстовые процессоры (word – processor), запоминающие промежуточное положения, применяемые для подготовки статей, отчетов и т.п. и получившие наиболее широкое распространение. В рамках программного обеспечения персональных компьютеров их было разработано довольно много. Текстовые процессоры иногда разделяют на редакторы общего назначения (MS Word, Word Perfect и др.), редакторы научных текстов (Chi Writer, Tex и др.). По мере повышения производительности ПК функциональность и их эффективность повышаются.

Среди основных функций, которыми обладают все редакторы подобного класса, следующие:

создавать – означает набирать на клавиатуре текст, который при этом отражается на мониторе;

хранить – означает работу с файлами – сохранение на системном диске в виде файла, считывание с диска (файла) и его просмотр с перемещением (вверх-вниз, влево-вправо), копирование на сменные носители и т.п.;

редактировать, т.е. среди традиционных функций всех текстовых процессов: строчное редактирование (вставка символов, их замена, удаление и т.п.), работа с группой строк (пометка фрагмента текста с размещением его в буфере и вставка в любое место текста или его удаление, «резка и склейка» строк и т.п.), быстрое перемещение курсора, контекстный поиск – по заданным фрагментам текста, орфографический и синтаксический контроль;

вывод на печать – обычно редакторы могут работать со многими шрифтами и алфавитами (при печатании документов возможен выбор

принтера, жирности шрифта, подчеркивание слов, форматирование таблиц и т.п.).

Более сложные функции, характерные для «продвинутых» редакторов, следующие:

расположение текста в колонки;

включение графической информации;

сборка оглавления;

поддержка ссылок на страницы и примечание;

деловая и иллюстративная графика;

математические функции;

контроль опечаток, словарь и т.п.

По-существу, компьютер превратился в супераналог пишущей машинки. Правда, следует отметить сложность освоения мощных редакторов и их низкую производительность, требующую определенного опыта и постоянного использования. В связи с этим при подготовке текстов программ обычно используются простые редакторы, встроенные в соответствующие инструментальные среды.

Наиболее высокий уровень ППП по подготовке текстов представляют редакционно-издательские системы, их сравнительно немного и они представляют специализированные системы, выходящие за пределы рассматриваемого класса ППП.

Иногда к редакторам текста относят также Web-редакторы (упоминавшийся ранее HTML), применяемые для подготовки Web-страниц Internet, а также программы Браузеры, предназначенные для просмотра электронных документов, выполненных в формате HTML. Однако эти средства не являются ППП рассматриваемого класса и относятся к инструментальным средствам компьютеров.

Особую группу ППП ПК составляют графические редакторы, ориентированные на растровую и векторную графику, с помощью которых можно создавать новые изображения, полученные с помощью сканера, видеокамеры, цифровой фотоаппаратуры и других средств. Однако они также не относятся к рассматриваемому классу ППП ПК.

К универсальным ППП ПК следует отнести также электронные таблицы. Табличные процессы или электронные таблицы (ЭТ) являются (наряду с текстовыми процессорами) одними из наиболее популярных и массовых применений ПК. Они нашли применение в экономических, конторских, управленческих делах, а иногда и в научных исследованиях. По мнению ряда авторов, именно этот вид программ-

ного обеспечения ПК положил основу компьютерной революции, поскольку впервые пользователь, не владеющий искусством программирования смог воспользоваться всеми преимуществами компьютера.

В основе ЭТ лежит двумерная матрица. Каждая ячейка матрицы имеет свой уникальный номер, состоящий из имени столбца и номера строки, на пересечении которых она располагается. Столбцы именуются буквами латинского алфавита (A, B, … Z, AA, AB, … AZ), строки нумеруются числами (1, 2, 3, …).

Первоначально таблица служила простым вместилищем информации, используемой программой, затем возникла идея помещать в таблице одновременно с данными также формулы, позволяющие автоматически выполнять расчеты при изменении введенных данных, постепенно набор формул расширялся, вводились правила форматирования и обработки данных, т.е. появились так называемые средоориентированные языки программирования.

Первая программа ЭТ появилась в 1979 г. под названием Visi Calc (от Visible Calculator – видимый калькулятор) разработки фирмы Software Arts для компьютеров Apple II. В 1985 г. Microsoft выпустила пакет Excel для компьютера Macintosh, а в 1986 г. для компьютеров IBM PC.

За прошедший период ЭТ успешно разрабатывались и совершенствовались рядом фирм и по своим принципам и возможностям отличались не очень сильно.

Все они имеют графический, оконный (Windows) пользовательский интерфейс, развитые средства программирования, обширные экранные возможности.

Кроме набора стандартных арифметических, логических функций, имеют встроенные экономические, статистические и др. функции (Excel – до 131 функции), имеют прекрасные средства графической обработки данных и очень удобные возможности вывода на печать. Язык макрокоманд облегчает управление таблицей. В таблицы обычно включаются некоторые функции управления, традиционные для СУБД – поиск и сортировка данных в таблицах в заданных пользователем пределах. Практически во всех таких пакетах включены средства традиционного программирования. Так, Excel использует стандартный Visual Basic.

Предельный размер таблиц (количество столбцов и строк) у различных фирм различен. Так, Excel (Microsoft) имеет 256 столбцов и 16384 строк, Quattro (Borland) 8192 столбца и 256 строки.

ЭТ находятся в постоянном развитии, их возможности постоянно увеличиваются, следуя за ростом ресурсов современных ПК.

Кроме электронных таблиц к универсальным пакетам прикладных программ, безусловно, следует отнести системы управления базами данных (СУБД). Эти ППП занимают очень важное место, поскольку представляют основу всех систем автоматизированного управления (АСУ).

Возросшие возможности ПК позволили реализовать эти пакеты, ориентированные прежде на большие компьютеры, что позволило массовому внедрению АСУ.

Подробное описание этих пакетов будет рассмотрено в следующих главах данного учебного пособия.

Весьма логичный шаг в развитии рассмотренных выше универсальных ППП – их объединение в интегрированные пакеты, что облегчает их взаимодействие, в частности перемещение данных (импорт данных), а одинаковый формат данных и идентичный интерфейс уменьшает потери времени на их освоение. Лидером создания интегрированных пакетов является фирма Microsoft. Распространяемый интегрированный пакет этой фирмы Office включает текстовый редактор Word, электронную таблицу Excel, СУБД Access, средства создания презентаций Power Pointer и ряд других структур.

Следующий уровень ППП ограничен рамками (хотя и достаточно широкими) определенной сферы применения.

Так, кризис в проектировании, разразившийся в конце 60-х, начале 70-х годов прошлого столетия вызвал необходимость разработки ППП для систем автоматизированного проектирования (САПР). В связи с этим различными фирмами были успешно разработаны и непрерывно совершенствуются некоторые пакеты для различных отраслей промышленности. Так, для потребностей машиностроения фирмой Avtodesk был разработан пакет AutoCAD (CAD – Computer Aided Design – разработка с помощью компьютера). В электронной промышленности созданы и активно используются такие пакеты, как P-CAD, Or CAD, Micro Cap, Electronics Workbench и т.п.

Широкое распространение получили математические пакеты, которые с определенной степенью условности можно назвать как САПР для математиков.

Изначально эти пакеты функционально разделялись на пакеты символьной математики, наиболее популярные из которых Mathematica и Maple V и пакеты численной математики, среди которых в отечественной практике наиболее популярны MATHCAD и MATLAB. Однако по мере развития и те, и другие пакеты стали (различными способами) включать функции, не характерные на начальном этапе развития универсального пакета MATHCAD, предназначенного для массового пользователя самой различной квалификации [6]. Пакет имеет ряд последовательных версий, первая из которых разработана в середине 80-х годов; две из последних версий – MATHCAD 8 PRO и MATHCAD 2000 PRO (MATHCAD PRO – сокращение от слов MATHematics for Computer Aided Design PROfessional). Упрощенная версия MATHCAD 2000 Standard пригодна для большинства пользователей и учебных целей, а расширенная версия MATHCAD 2000 Premium – для профессиональных математиков и ученых. MATHCAD – интегрированный пакет, т.е. он может объединяться с другими математическими и графическими системами (пакетами) для решения сложных задач. Имеет удобный пользовательский интерфейс со многими окнами, кнопками, мышью, средствами управления курсором, вид которого можно изменять, и другими элементами. Решение описывается с помощью привычных математических знаков (сумм, интегралов, корней и т.д.) и формул.

Встроенный в пакет мощный символьный процессор позволяет выполнять многие символьные операции: дифференцирование, интегрирование, суммирование, разложение в ряды, упрощение формул, решение алгебраических и дифференциальных уравнений, преобразования Фурье, Лапласа и др. Имеется много встроенных функций и готовых для вычисления математических методов по обработке данных, статистике, регрессионному анализу, решению нелинейных уравнений, оптимизации, преобразованию Фурье и др. Богатые возможности пакета MATHCAD позволяют с его помощью готовить отчеты, статьи, диссертации, проекты, книги и другие материалы по многим темам.

Не менее популярным является пакет прикладных программ MATLAB (сокращение от слов MATrix LABoratory), фактически

представляющий собой объединение нескольких пакетов в единую систему. Первоначальная версия системы разработана в начале 80-х годов, затем был создан ряд все более мощных версий (некоторые из последних версий – MATLAB 5.3, MATLAB 6.0, MATLAB 7.3). Первая версия была написана на языке Фортран, более поздние версии на С. Имеется специальная упрощенная, «студенческая», версия системы «The Student Edition of MATLAB» с меньшими возможностями, но зато пригодная для недорогих компьютеров с ограниченными ресурсами.

Первоначально пакет MATLAB, как видно из названия, предназначался, в основном, для решения задач линейной алгебры и других задач, в которых все величины удобно представлять в виде матриц и векторов. Но в результате многих усовершенствований и переработок современный пакет MATLAB пригоден для решения широкого спектра задач в самых различных областях науки, техники, образования. Характерная особенность пакета – все величины рассматриваются как прямоугольные числовые массивы (скаляры, векторы, матрицы), возможно с комплексными элементами, для которых не требуется указывать размерность явно. Символьные вычисления также возможны в рамках этого пакета благодаря интерфейсу с популярным пакетом символьной математики Maple.

Система MATLAB активно используется в учебной и научно-исследовательской работе многих высших учебных заведений, а ее изучение включено в учебные планы большого числа ведущих университетов мира. Система широко применяется также в научно-исследовательских, конструкторских и проектных организациях, на крупных предприятиях и других хозяйственных объектах.

К этой же группе ППП можно, по-видимому, отнести проблемно- и предметно-ориентированные экспертные системы – одно из наиболее перспективных направлений в области искусственного интеллекта, а также ряд других ППП достаточно широкого применения.

И, наконец, последний уровень ППП составляют узкоспециализированные пакеты, не имеющие массового спроса, рассчитанные на решение частных, но часто весьма важных задач в различных областях науки и техники. К таким пакетам можно отнести ряд программных комплексов по проектированию различных электрофизических установок, установок ядерно-физического комплекса и др.

7.5. Программное обеспечение компьютерной графики

Программные средства работы с компьютерной графикой называют *графическими редакторами* (graphics editor). Существует два типа графических редакторов:

1) редакторы для создания новых изображений;

2) редакторы для улучшения уже готовых изображений, полученных с помощью сканера, видеокамеры, цифрового фотоаппарата и других средств.

Рассмотрим сначала графические редакторы для работы с растровой графикой. Существует несколько десятков форматов, в которых сохраняются растровые изображения. Для ОС Windows универсальным считается формат без сжатия информации Windows Bitmap, имеющий расширение .bmp. Для Web-документов в сети Internet более удобны форматы, обеспечивающие сжатие информации с целью сокращения объема файлов. Одним из таких форматов является формат JPEG (расширение .jpg), сохраняющий данные с огромной степенью сжатия, но за счет потери некоторой, относительно небольшой, части информации. Также используемый в Internet формат GIF (расширение .gif) обеспечивает наивысшее уплотнение без потери информации. В полиграфии распространенным является специальный формат TIFF (расширение .tif), дающий неплохую степень сжатия и открывающий некоторые дополнительные возможности при печати изображений.

Из графических редакторов первого типа для растровых изображений очень популярным и относительно простым является однооконный (для работы на экране монитора открывается единственное окно) редактор Paint, включенный в программное обеспечение Windows 95. Редактор Paint представляет собой OLE-сервер (см. разд. 10.9) и созданные в нем изображения можно вставлять в документы таких универсальных приложений Windows, как MS Word, MS Excel, MS PowerPoint, и даже в базы данных MS Access. Для создания и редактирования изображений Paint представляет богатый набор инструментов (палитры цветов, кисть, распылитель, ластики для стирания, «карандаши», ножницы для вырезания фрагментов) и средств работы с этими инструментами. Paint позволяет создавать довольно сложные черно-белые или цветные рисунки, схемы, чертежи, хотя и не обладающие высокими худо-

жественными или инженерно-техническими качествами. В силу своей простоты и доступности Paint часто используется в качестве первой ступени при обучении и овладении более сложными средствами компьютерной графики.

К редакторам первого типа относятся также Painter компании Fractal Design, Free Hand компании Macromedia и Fauve Matisse. Редактор Painter обладает широкими возможностями средств рисования и работы с цветом, позволяя, в частности, имитировать различные инструменты (кисти, карандаши, перо, уголь) и материалы (акварель, масло, тушь). Последние версии редактора Free Hand также содержат разнообразные средства редактирования изображений и текста, включая многоцветную градиентную заливку, библиотеку спецэффектов и др.

К графическим редакторам второго типа относятся, например, редакторы Adobe Photoshop, Photostyler, Picture Publisher, наиболее популярными из которых являются редакторы Adobe Photoshop фирмы Adobe, фактически считающиеся стандартом в этой области. Из множества средств обработки готовых изображений отметим такие средства, как улучшение яркости и контраста, повышение четкости, *цветовая коррекция* (изменение яркости и контрастности в различных цветовых каналах), *отмывка* (изменение яркости фрагментов), *обтравка* (вырезание отдельных фрагментов и их последующее улучшение с возвратом, «вклеиванием», на прежнее место), *набивка* (восстановление утраченных элементов изображения путем копирования сохранившихся фрагментов), *растушевка* (сглаживание границ), *монтаж* (компоновка изображения из фрагментов одного или нескольких изображений). Интересным средством обработки изображений являются так называемые *фильтры* – программные средства преобразования изображений с целью улучшения их качества или художественной выразительности. С помощью фильтров можно повысить четкость изображения, придать фотографии вид карандашного или угольного рисунка, барельефа, гравюры, мозаики; выполнить стилизацию изображения, например имитировать изображение на ткани, бумаге, металле и других основах.

Для выполнения этих преобразований графические редакторы предоставляют пользователю специальные инструментальные палитры в виде диалоговых окон с различными панелями (наборами)

инструментов, указанных пиктограммами. Так, в редакторе Adobe Photoshop 4.0 имеется десять таких палитр. Основное отличие палитр от обычных диалоговых окон ОС Windows – возможность перекомпоновки рабочей среды пользователем путем перемещения палитр на экране и монтирования новых палитр.

Рассмотрим теперь графические редакторы, применяемые для работы с векторной графикой, когда изображение – чертеж, схема, диаграмма, но не рисунок. Наиболее известными из таких редакторов являются Adobe Illustrator 7.0 (по-видимому, самый лучший); Macromedia Freehand 8.0, имеющий дружественный интерфейс и рекомендуемый для начинающих пользователей, и Corel Draw (версии от 5.0 до 8.0), исторически применяемый в компьютерах IBM PC, – очень богатый по своим возможностям и позволяющий создавать изображения, подобные художественным, однако более сложный в изучении и использовании и с менее удобным пользовательским интерфейсом, чем первые два редактора. В последнее время Corel Draw применяется меньше, поскольку редакторы Adobe Illustrator и Macromedia Freehand стали шире использоваться и в компьютерах IBM PC.

Элементами векторной графики в графических редакторах служат линии, контуры, объекты. Эти элементы можно группировать, комбинировать, объединять, заливать различными способами, используя многочисленные меню и инструменты, обычно кодируемые пиктограммами. Вместе с рисунками можно создавать и тексты, причем не только строчные, но и фигурные, расположенные вдоль заданных кривых или в заданных контурах. Можно также видоизменять символы и шрифты, создавая необычные надписи, обладающие художественной выразительностью, например логотипы (краткие наименования) предприятий и фирменные стили для использования в объявлениях, рекламах, проспектах, а также для создания оригинальных электронных документов и Web-страниц в сети Internet. Отметим еще, что растровые изображения можно преобразовывать в векторные, а затем дорабатывать, улучшать с помощью редакторов векторной графики, и наоборот, векторные изображения преобразовывать в растровые с целью последующего редактирования, улучшения с помощью, например, такого мощного средства, как фильтры редактора Photoshop.

Одним из перспективных приложений средств компьютерной графики становятся в последнее время так называемые *настольные типографии* (desktop publisher) для печати малотиражных изданий, реклам, извещений, объявлений, листовок, а также *настольные издательские системы,* применяемые для оформления (верстки) документов, предназначенных для полиграфических изданий. Наиболее известными из настольных издательских систем являются QuarkXPress и PageMaker. Процесс верстки документа состоит в оформлении текста и взаимного расположения текста и иллюстраций на основе оконной технологии. Цель верстки – создание оригинала-макета, пригодного для последующего размножения документа полиграфическими средствами. Работа с настольными издательскими системами является объектно-ориентированной, объектами работы служат блоки текста, рисунки и стандартные элементы оформления (линии, рамки и т.п.), причем блоки текста и рисунки могут быть подготовлены заранее с помощью текстовых и графических редакторов. Для работы пользователю настольной издательской системы предоставляется набор действий, оформленных как меню, панель инструментов, панель размеров и панель макета документа. Для хранения наборов объектов, созданных пользователями, имеются библиотеки, которые можно пополнять в ходе работы. Из библиотек можно извлекать копии текстовых и графических объектов для использования в верстке. Имеется также широкий набор средств для работы с цветом.

В заключение этого раздела заметим, что техника квалифицированной работы с компьютерной графикой и настольными издательскими системами требует большого опыта ввиду многообразия имеющихся средств и манипуляций с ними.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. В чем особенности структуры ПО ПК? Какие части ОС необходимы всем компьютерам? Какие части определяются областью их применения? Назовите разновидности вспомогательных программ (утилит), разработанных для ОС ПК.

2. Назовите разновидности ОС, применяемых в ПК. В чем основные ограничения DOS? Перечислите основные компоненты, характерные для всех дисковых операционных систем ПК.

3. Расскажите об особенностях физической организации файловых систем ПК. В чем причины перехода от таблицы FAT 16 к таблице FAT 32?

4. Каковы основные направления развития OS/2, функциональные возможности, совместимость с DOS приложениями? Назовите основные достоинства системы управления файлами HPFS.

5. Перечислите основные причины популярности ОС UNIX. Расскажите о системах приоритетов ОС UNIX, защите файлов, направлениях использования этой ОС.

6. Расскажите об особенностях инструментального программного обеспечения ПК, структуре и составе инструментальных сред, интегрированных средах фирмы Borland – Delphi, C^{++}Builder.

ГЛАВА 8. ОРГАНИЗАЦИЯ, ХРАНЕНИЕ И ПОИСК ДАННЫХ

8.1. Структуры данных

В наиболее общем понимании *данными* считаются все факты и идеи, представленные в формализованном виде, позволяющем их передавать и обрабатывать с помощью некоторых процессов и соответствующих технических средств. В более узком понимании данными можно считать любой набор знаков, независимо от их смысла. Во введении указывалось, что количество информации измеряется в битах. Применительно к данным используются также термины: символ, слово, запись и файл. *Символ* представляет собой любой знак, допустимый в избранном алфавите, и занимает несколько бит (обычно один байт). *Слово* состоит из нескольких символов, длина слова ограничивается разрядностью компьютера и составляет обычно 2 – 8 байт. *Запись* состоит из нескольких слов. Записью, например, может быть информация на перфокарте, строка на экране дисплея или на бумаге. *Файл* (file – досье, картотека) – это рассматриваемый как единое целое поименованный набор однотипных по структуре и использованию записей.

Организацию данных называют *структурой данных.* Различают линейные и нелинейные структуры. *Линейные* – структуры данных, в которых переход от одного элемента данных к другому не зависит от каких-либо логических условий, т.е. в линейных структурах используются лишь безусловные связи элементов; *нелинейные* – структуры данных, в которых переход от одного элемента данных к другому может зависеть от выполнения некоторого логического условия, т.е. в нелинейных структурах могут использоваться условные связи элементов.

К линейным структурам относятся массивы, таблицы, ряды и списки. *Массив* представляет собой упорядоченный набор элементов, взаимно-однозначно связанных с индексами; *индексы* – упорядоченный набор целых чисел, определяющий позицию каждого элемента массива. Простейшими типами массивов являются прямоугольные массивы. *Прямоугольным* называется массив, каждый индекс которого меняется с постоянным шагом от некоторой постоянной нижней границы до постоянной верхней. Одномерный прямоугольный массив называют *вектором,* двумерный – *матрицей.* В

непрямоугольных массивах нижние и верхние границы индексов, а также шаг изменения индексов сами могут являться функциями индексов. Так, в непрямоугольном двумерном массиве нижняя и верхняя границы номера столбца могут зависеть от номера строки.

Таблица – набор элементов, снабженных некоторым ключом (признаком). Поиск элементов в таблице осуществляется по значению ключа. Например, для математических таблиц различных функций ключом служит аргумент функции, для таблицы членов какого-либо коллектива – фамилия, имя, отчество. При трансляции программ транслятор составляет таблицу имен, в этом случае ключом служит идентификатор, а информацией – тип величины и ее адрес в памяти. Прямоугольные массивы и таблицы иногда называют *линейными прямоугольными структурами.*

К структурам типа *ряда* относятся строки, очереди и стеки. *Строка* представляет собой одномерный набор элементов данных с последовательным доступом. Доступ к какому-либо элементу возможен только с одного из концов строки путем последовательного перебора всех элементов, предшествующих искомому. Элементами строки могут быть не только символы какого-либо алфавита, но и более сложные структуры, в том числе другие строки (такие элементы называют подстроками). Так, для транслятора исходная программа представляется строкой, каждый элемент которой – строка текста программы.

Очереди и *стеки* – это динамически изменяемые упорядоченные наборы элементов. Новые элементы в очередях и стеках всегда добавляются к одному и тому же концу набора – «входному концу». Очередь функционирует по принципу «первым пришел – первым ушел», при этом новые элементы добавляются к входному концу очереди, а удаляются с другого конца – выходного, по мере ввода и удаления данные перемещаются в очереди от входа к выходу (рис. 8.1, *а*).

Стек функционирует по принципу «последним пришел – первым ушел», при этом удаление элементов производится со входного конца (рис. 8.1, *б*). При добавлении в стек нового элемента данных все ранее загруженные элементы сдвигаются на одну позицию в глубину стека, а при удалении элемента данных сдвиг производится на одну позицию ко входу в стек. По такому принципу используется, например, стопка бумаги (листы добавляются и удаля-

ются сверху) или магазин автомата (верхний патрон выстреливается первым), поэтому иногда стек называют магазином. Очереди и стеки обычно организуются аппаратными (схемными) средствами как очень быстрые запоминающие устройства ограниченной емкости с безадресным обращением. Например, очереди могут использоваться для запоминания запросов центрального процессора на обслуживание терминалов, стеки – при трансляции скобочных выражений, обработке вложенных циклов, вычислениях по рекуррентным формулам.

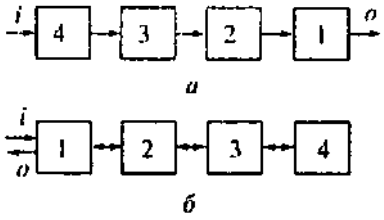


Рис. 8.1. Очередь (а) и стек (б): *i* – вход, *o* – выход

Список представляет собой набор элементов, каждый из которых содержит два поля: *информационное* с элементом данных (или адресом элемента данных в памяти компьютера) и *адресное*, в котором указывается адрес следующего элемента списка. Если в адресном поле содержится 0, то этот элемент списка является последним. Список снабжается *заголовком,* в котором указывается адрес первого элемента списка. Перебор элементов списка всегда начинается с заголовка, а затем продолжается по ссылочным адресам связи. Такие простейшие списки называются *однонаправленными* (рис. 8.2, *а*). Элементы более сложных *двунаправленных* списков содержат в адресном поле два адреса: адрес следующего элемента и адрес предыдущего (рис. 8.2, *б*). В *кольцевых* списках в адресном поле последнего элемента указывается адрес первого элемента (рис. 8.2, *в*).

В информационном поле элементов списка может содержаться значительный объем данных, например имя, фамилия, отчество, год рождения, место работы, образование, должность, стаж, адрес и т.п. Наконец, структура самих элементов списка может быть сложной, например они могут быть таблицами или списками. Так, список сту-

дентов вуза можно представить в виде списка факультетов, каждый элемент которого представляет собой список номеров групп, а каждый элемент списка номеров групп, в свою очередь, является списком группы. Такие многоуровневые списки называют *иерархическими* (рис. 8.2, *г*). Отметим еще *ассоциативные* списки, которые организуются на едином наборе информационных данных, но каждый элемент списка содержит в адресном поле много ссылочных адресов, образующих сложные переплетающиеся связи. Так, по анкетным данным сотрудников некоторой организации можно составить списки по специальности, году рождения, стажу, образованию, количеству детей и т.д.

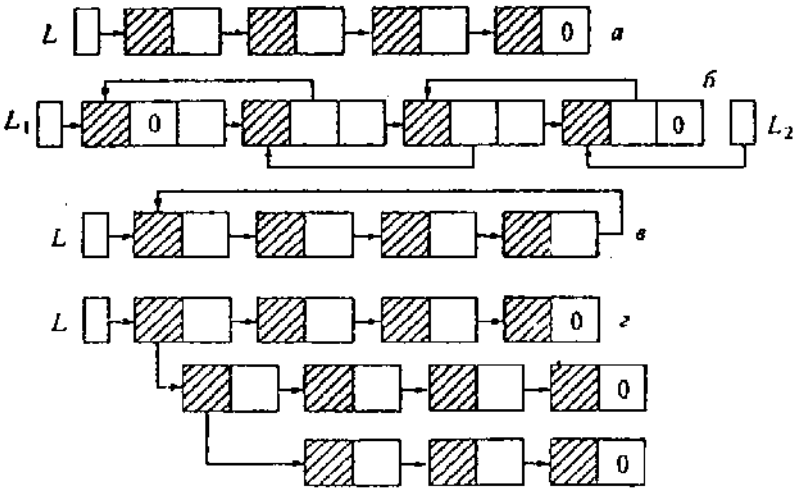


Рис. 8.2. Списки: *а* — однонаправленный; *б* – двунаправленный; *в* – кольцевой; *г* – иерархический; L, L_1, L_2 – заголовки списков; информационное поле заштриховано

К нелинейным структурам данных относятся графы, деревья, сплетения и сети фреймов. *Граф* – конечное множество точек (*вершин*) и отрезков линий, каждый из которых соединяет две вершины и называется *ребром.* Если на ребре указано направление (стрелкой), то это *ориентированное ребро* или *дуга*. Граф, содержащий только ребра, называется *неориентированным* (рис. 8.3, *а*); граф, содержащий только дуги, – *ориентированным* (рис. 8.3, *б*).

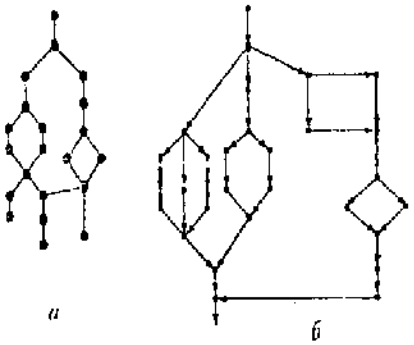


Рис. 8.3. Графы: *а* – неориентированный; *б* – ориентированный

Если вершины ребра (дуги) совпадают, то ребро (дуга) становится *петлей.* Любая последовательность смежных ребер на графе образует *маршрут*. Маршрут называется *цепью,* если все его ребра различны, и *простой цепью*, если все его вершины различны. Замкнутая цепь является *циклом*, а замкнутая простая цепь – *простым циклом*. В ориентированном графе простую цепь называют *путем*, а простой цикл – *контуром*. Граф считается *связным*, если любая пара его вершин соединяется маршрутом. *Концевая* вершина является концом (или началом) единственного ребра (дуги).

Деревом называется связный неориентированный граф, не содержащий циклов, в том числе петель (рис. 8.4, *а*).

Дерево с одной выделенной вершиной – *корневое дерево*, а выделенная вершина – *корень*. *Дерево, растущее из вершины* v_0, – это ориентированный граф, который является, без учета ориентации, корневым деревом с корнем v_0, причем для любой вершины v существует единственная цепь, соединяющая v_0 с v и являющаяся ориентированным путем из v_0 в v (рис. 8.4, *б*). Концевые вершины корневого дерева иногда называют *листьями* (за исключением корня, если корень также является концевой вершиной). Обычно корневые деревья изображают так, что вершины простых цепей, ведущих от корня к листьям, располагаются на уровнях, причем на первом уровне располагается корень. Так, дерево на рис. 8.4, *б* имеет шесть уровней вершин. Если какую-либо вершину корневого дерева v_1

принять за новый корень, то множество вершин и ребер, составляющих дерево с этим корнем, образует *поддерево* (рис. 8.4, *в*). Конечное или счетное множество деревьев составляет *лес*.

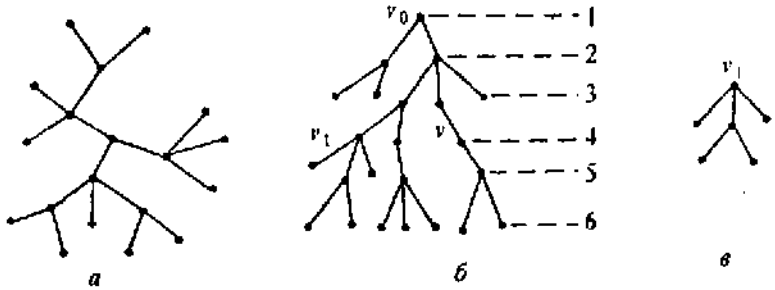


Рис. 8.4. Деревья: *а* – дерево; *б* – растущее дерево с корнем v_0; 1, 2, …, 6 – уровни вершин; *в* – поддерево растущего дерева

В *структурах данных типа графов* и *деревьев* вершинам отвечают элементы данных, а соединяющим вершины отрезкам линий – связи между этими элементами. Наконец, под *сплетениями* понимаются такие структуры данных, в которых на одном и том же наборе данных (вершин) построены различные графы и деревья.

Существенно более сложными нелинейными структурами данных являются сети фреймов. *Сеть фреймов* представляет собой граф, каждая вершина которого является фреймом. *Фрейм,* в свою очередь, представляет собой ориентированный граф (или растущее корневое, т.е. ориентированное дерево), вершины которого располагаются на нескольких уровнях, причем на самом верхнем уровне имеется единственная вершина (корень дерева), фиксирующая имя данного фрейма. *Именем* фрейма обычно служит название какого-либо понятия (объекта, процесса, явления) некоторой предметной области, данные о котором содержит фрейм, например «болт» (деталь), «сталь» (материал), «сверление» (процесс), «трение» (явление) и т.п. Сеть верхних вершин с именами фреймов отражает связь различных понятий предметной области, а уровень этих вершин называют уровнем фреймов.

Однако кроме *уровня фреймов*, назначение которого – указание понятий предметной области и их связей, фрейм имеет более глу-

бокие уровни, на которых раскрываются соответствующие понятия и хранятся идентификаторы специальных процедур для работы с фреймами. Эти процедуры автоматически инициализируются, если содержимое фрейма подвергается каким-либо изменениям, и поддерживают непротиворечивость данных всей сети фреймов. В отличие от классического графа, в котором все вершины равноправны и непосредственно доступны пользователю, в сети фреймов пользователь имеет непосредственный доступ лишь к вершинам самого верхнего уровня – уровня фреймов, тогда как к вершинам более глубоких уровней отсутствует непосредственный пользовательский доступ.

На рис. 8.5 показана типичная структура фрейма с пятью уровнями. За уровнем фреймов (1) следует *уровень слотов* (2).

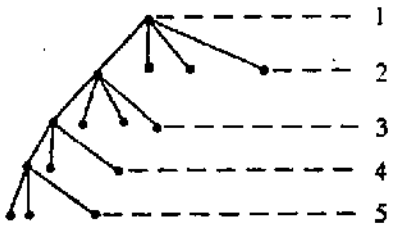


Рис. 8.5. Структура фрейма: 1 – уровень фреймов; 2 – уровень слотов; 3 – уровень фасетов; 4 – уровень данных; 5 – уровень комментариев

В вершинах этого уровня фиксируются имена *слотов.* Обычно это названия свойств, характеристик того понятия, которое описывает данный фрейм. Например, если имя фрейма – «болт», то именами слотов могут быть «тип», «материал», «покрытие», «резьба», «размеры», «головка», «применение», «цена» и т.п. Каждый слот, как и фрейм, представляет собой иерархическую структуру, но с четырьмя уровнями.

Под уровнем слотов располагается *уровень фасетов* (3). *Фасеты* позволяют задавать модификации свойств, описываемых слотами. Такими модификациями могут быть:

1) явное задание характеристики;
2) задание характеристики по умолчанию;
3) действия при добавлении данных;
4) действия при удалении данных.

Из этих четырех фасетов, показанных на рис. 8.5, первые два фасета указывают способ задания описательной части слота.

Например, для фрейма «болт» характеристика «тип» по умолчанию может иметь значение «крепежный», а явно заданная – «установочный»; характеристика «материал» по умолчанию – «сталь», а явно заданная – «бронза» и т.д. Третий и четвертый фасеты составляют процедурную часть слота: они указывают, какие действия и где следует выполнить при изменениях описательной части слота для сохранения непротиворечивости всей сети данных.

Под уровнем фасетов размещается *уровень собственно данных* (4), на котором хранятся фактические данные, идентификаторы процедур и т.п. В качестве иллюстрации рассмотрим два примера:

1) *имя фрейма* – болт;
   имя слота – материал;
   фасет – значение по умолчанию;
   значение данного – сталь.

2) *имя фрейма* – болт;
   имя слота – покрытие;
   фасет – явное задание;
   значение данного – хромированный.

Наконец, самый нижний уровень – *уровень комментариев* (5), имеющих фиксированную структуру и помогающих при обработке фреймов. Как показывает опыт, гибкая структура сетей фреймов удобна для хранения динамически изменяемых систем данных, но для работы с ними требуются более сложные программные средства.

8.2. Типичные виды работ с данными

Основными операциями со строками являются:

последовательный перебор символов строки;

включение нового символа (группы символов) в заданное место строки;

исключение символа (группы символов) из заданного места;

сцепление строк;

разделение строки на несколько строк;

поиск вхождения заданной подстроки в строку и определение места этого вхождения.

Такие языки, как Алгол и Фортран, плохо подходят для выполнения этих операций, поэтому используют специальные языки: Лисп, Снобол и др. Строки часто хранятся в памяти компьютера в

виде векторов с последовательным расположением символов. При этом операции со строками могут быть сопряжены с необходимостью перемещения в памяти компьютера большого числа символов и значительными затратами машинного времени. Поэтому строки часто оформляют в виде списков. В этом случае требуется дополнительное расходование машинной памяти, но зато включение и исключение символов, сцепление и разделение строк выполняются очень быстро путем изменения адресов связи в списках.

Основной операцией с таблицами является выбор из таблиц. Ограничимся случаем таблиц функций $y = y(x)$ одной действительной переменной x, полагая, что значения функции

$$y_j = y(x_j), \quad j = 0, 1, \ldots, n, \tag{8.1}$$

упорядочены по возрастанию аргумента (ключа), т.е.

$$x_{j+1} > x_j, \quad h_j = x_{j+1} - x_j > 0, \quad j = 0, 1, \ldots, n-1, \tag{8.2}$$

где h_j – шаг аргумента x. *Выбором из таблицы* будем называть нахождение такого промежутка, для которого

$$x_j \le x < x_{j+1} \tag{8.3}$$

или, что то же самое, нахождение такого значения индекса j, для которого выполняется неравенство (8.3).

В случае *таблиц с постоянным шагом*, когда $h_j = h = \text{const}$, индекс j легко находится по формуле

$$j = [(x - x_0)/h], \tag{8.4}$$

(квадратные скобки означают выделение целой части числа). *Регулярными* называют таблицы, в которых

$$j = f(x_j), \tag{8.5}$$

где $f(x)$ – монотонно возрастающая функция x. Для регулярных таблиц также несложно найти

$$j = [f(x)]. \tag{8.6}$$

Пусть, например,

$$x_j = x_0 \mathrm{e}^{qj}, \; j = \frac{1}{q} \ln\left(\frac{x_j}{x_0}\right) \quad (j = 0, 1, 2, \ldots). \tag{8.7}$$

Здесь $q > 0$ – некоторая константа. Тогда для произвольного $x > x_0$ находим

$$j = \left[\frac{1}{q} \ln\left(\frac{x}{x_0} \right) \right]. \qquad (8.8)$$

Существенно сложнее осуществляется выбор из *нерегулярных* таблиц с произвольным переменным шагом. Некоторое упрощение выбора из таких таблиц достигается в случае «почти регулярных» таблиц и таблиц с почти постоянным шагом. *Почти регулярными* называют таблицы, для которых существует такая монотонно возрастающая функция $f(x)$, что

$$\left| f(x_j) - j \right| \le k \ , \ 0 < k << n, \qquad (8.9)$$

где n – длина таблицы, т.е. узел j отстоит от

$$\tilde{j} = [f(x_j)] \qquad (8.10)$$

не более чем на $k << n$ узлов. Считают, что таблица имеет *почти постоянный шаг*, если

$$\left| x_j - (x_0 + jh) \right| \le kh \ , \quad h = (x_n - x_0)/n \ , \quad 0 \le k << n, \qquad (8.11)$$

где h – средний шаг таблицы. Здесь также искомый узел j отстоит от

$$\tilde{j} = [(x - x_0)/h] \qquad (8.12)$$

не более, чем на k узлов. В обоих случаях сначала находят $\tilde{j}$ по формулам (8.10) или (8.12), а затем – искомый узел j прямым просмотром узлов x_j с номерами от $\tilde{j} - k$ до $\tilde{j} + k$.

В общем случае *таблиц с произвольным шагом* скорость выбора определяется числом m просмотренных значений ключа (аргумента). Наиболее медленным является *прямой перебор* аргументов с начала таблицы (*метод шестеренки*), требующий в случае равномерно перемешанных значений аргумента в среднем $m/2$ просмотров. Если требуемые значения x упорядочены монотонно, то ускорение выбора достигается в *методе скользящего начала*. В этом методе запоминается последнее найденное j и при следующем об-

ращении просматривается таблица с началом в узле x_j, т.е. просматриваются узлы x_j, x_{j+1}, x_{j+2}, …, x_n.

В более быстром *методе двухступенчатого перебора* таблица предварительно разбивается на сегменты с граничными узлами x_0, x_k, x_{2k}, … ($k << n$). Далее просмотром граничных узлов находится сегмент, в который попадает данное значение x, а затем просматривается каждый узел этого сегмента. Наконец, наиболее быстрым является *метод половинного деления*, в котором сначала определяется, в какую половину таблицы попал данный x, затем эта половина опять делится пополам и т.д. За k делений первоначальный объем таблицы n уменьшается до $n / 2^k$. Процесс деления прекращается, когда остается таблица из одного узла. Можно показать, что число просмотров, а следовательно, и число арифметических операций в этом методе имеет порядок

$$M = O(\log_2 n). \qquad (8.13)$$

Часто встречается такой вид обработки информации, как упорядочение (сортировка) данных, например упорядочение списка фамилий по алфавиту или таблицы по возрастанию (убыванию) ключа. Ограничимся здесь лишь алгоритмами сортировки в оперативной памяти, называемой *внутренней сортировкой*, оставляя в стороне более сложные алгоритмы *внешней сортировки*, т.е. сортировки данных на внешних носителях (магнитных дисках и лентах). Экономичность многочисленных алгоритмов внутренней сортировки можно сопоставлять по запросам машинных ресурсов – числу операций M и объему рабочей области памяти C. Число операций M определяется, главным образом, числом сравнений ключей m. Теоретически доказано, что минимальное число сравнений

$$\min m = an \log_2 n, \quad a = \text{const}, \qquad (8.14)$$

где n – число сортируемых элементов; a – постоянная, не зависящая от n. Следовательно, идеальный алгоритм должен иметь

$$M = O(n \log_2 n), \quad C = 0. \qquad (8.15)$$

Практически, однако, для уменьшения M требуется увеличение C.

Далее для определенности будем полагать, что массив чисел $A(j)$ (j = 1, 2, …, n) требуется упорядочить по возрастанию. В простей-

шем методе сортировки (иногда называемом «*методом пузырька*») просматривается массив $A(j)$ и последовательно упорядочиваются пары элементов $[A(j), A(j + 1), (j = 1, 2, \ldots, n - 1)]$. Сортировка считается завершенной, если при просмотре всего массива не потребовалось ни одного перемещения элементов. В наихудшем случае, когда первоначально массив $A(j)$ упорядочен по убыванию, требуется $m = n^2$ сравнений элементов (ключей); в наилучшем случае, когда $A(j)$ уже упорядочен по возрастанию, значение $m \approx n$, а в среднем $m \approx 0{,}5n^2$. Практически метод применим лишь для небольших n или для «почти упорядоченных» массивов с малыми отклонениями от монотонности.

В другом элементарном методе сначала отыскивается минимальный элемент массива $\{(A(1), A(2), \ldots, A(n)\}$, который затем обменивается позициями с $A(1)$. После этого процесс повторяется для массива $\{(A(2), A(3), ..., A(n)\}$, имеющего $n - 1$ элементов и т.д. Число сравнений $m \approx 0{,}5n^2$ также достаточно велико.

Метод вставки основан на пополнении уже упорядоченного массива или его части новыми элементами. Пусть, например, часть массива $A(1), A(2), \ldots, A(j)$ уже упорядочена. Для пополнения его новым элементом $A(j + 1)$ отыскивается такая позиция i, что $A(i) \leq \leq A(j + 1) < A(i + 1)$; сюда вставляется элемент $A(j + 1)$, а элементы $A(i + 1), A(i + 2), ..., A(j)$ сдвигаются на одну позицию вправо (вниз). Для упорядочения всего массива методом вставки требуется в среднем $m \approx 0{,}25n^2$ сравнений и много сдвигов. (Сдвиги можно исключить, используя списковую организацию данных.)

Рассмотрим теперь *объединение двух упорядоченных массивов A* и *B*. Для определенности будем полагать, что каждый массив содержит n элементов. Просматривая массив A, начиная с $A(1)$, находим позицию j, в которую требуется вставить элемент $B(1)$. После вставки $B(1)$ продолжаем просмотр массива A с позиции $j + 1$ для нахождения позиции вставки $B(2)$ и т.д. Как видно, для объединения массивов потребуется $m = n$ сравнений, т.е. объединение можно выполнить очень быстро. Сдвиги элементов при объединении можно исключить, если дополнительно отвести для объединенного массива рабочую область памяти объемом $C = 2n$.

Метод вставки и объединение массивов используются в *сортировке Шелла*. В этом методе весь массив A разбивается на сегменты объемом n_1 и каждый сегмент упорядочивается методом вставки. Затем пары сегментов объединяются в сегменты объемом $2n_1$, $4n_1$ и т.д. Среднее число сравнений для метода Шелла

$$m \approx 0{,}5n^{3/2}. \qquad (8.16)$$

Все рассмотренные выше методы сортировки реализуются без дополнительных затрат памяти: $C \approx 0$. Рассмотрим теперь *древовидную сортировку* (название связано с тем, что сортировка может быть представлена в виде дерева). В этом алгоритме из массива A случайным образом извлекается некоторый элемент $A(i)$. Затем элементы массива перераспределяются по двум сегментам так, что в первом сегменте оказываются элементы $A(j) \le A(i)$, а во втором – элементы $A(j) > A(i)$. Затем аналогично перераспределяются элементы в каждом сегменте и т.д. Здесь уже $C \neq 0$, так как требуется запоминать границы образующихся сегментов, но число операций M близко к минимальному (8.15).

Минимальное число сравнений (8.14) достигается в *методе слияния (алгоритм фон Неймана)*. Сначала упорядочиваются пары соседних элементов, затем четверки, восьмерки и т.д. При этом объем массива n может не равняться целой степени числа 2. В этом методе $C = n$, но за счет некоторого увеличения числа операций можно уменьшить C на порядок и более.

Наконец, в алгоритме *двухступенчатой сортировки* при первом просмотре массива находятся границы изменения элементов $A_{\min}$, $A_{\max}$. Далее диапазон $[A_{\min}, A_{\max}]$ делится на некоторое число K интервалов с границами

$$A_i = A_{\min} + (A_{\max} - A_{\min})i / K, \quad i = 0, 1, 2, \ldots, K, \qquad (8.17)$$

и при втором просмотре массива A находятся числа элементов $A(j)$ в каждом интервале, а следовательно, и границы интервалов после упорядочения. При третьем просмотре массива все элементы $A(j)$ перераспределяются по своим интервалам. На этом первая ступень сортировки завершается. На второй ступени каждый интервал упорядочивается методом пузырька. Наибольшая экономия

достигается, если элементы $A(j)$ равномерно заполняют диапазон $[A_{\min}, A_{\max}]$, тогда

$$m \approx 3n + K(n/K)^2 \approx n(3 + n/K), \quad C \approx K. \qquad (8.18)$$

(Во многих задачах приемлемые результаты получаются уже после первой ступени сортировки.)

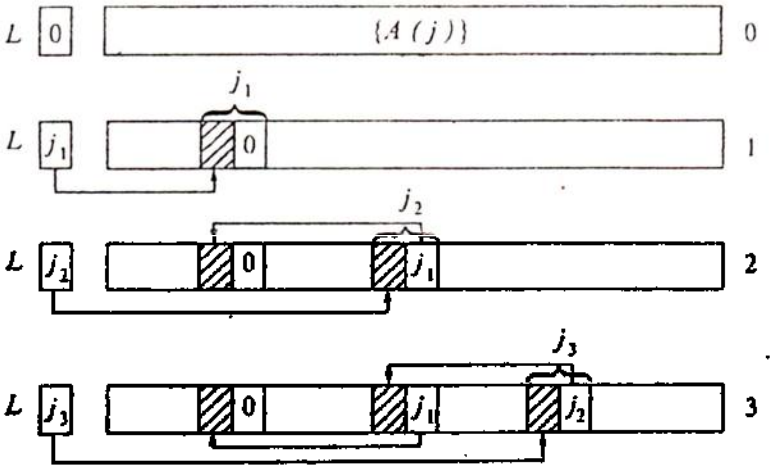


Рис. 8.6. Последовательность составления списка: 0 – до начала составления списка; 1 – после нахождения первого элемента; 2 – после нахождения второго элемента; 3 – после нахождения третьего элемента (L – заголовок списка, информационные поля заштрихованы)

Рассмотрим теперь алгоритм составления списков, ограничиваясь для простоты однонаправленными списками. Пусть из некоторого множества элементов $\{C(j);\ j = 1, 2, \ldots, n\}$ надо составить список элементов с заданным значением ключа: например, по списку персонала некоторого предприятия составить список инженерно-технических работников. Каждый элемент $C(j)$, как уже указывалось, имеет информационное поле $D(j)$ и адресное поле $A(j)$. Перед составлением списка выделяется ячейка памяти L, называемая заголовком списка. Вначале полагают $L = 0$, а в дальнейшем в L будет храниться адрес j_1 первого элемента списка. Просматривая затем множество $\{A(j)\}$, последовательно находят элементы,

которые должны быть включены в список. После нахождения очередного такого элемента заголовок списка L переносится в адресное поле $A(j)$ элемента, а адрес найденного элемента j – в заголовок L. При таком способе составления первый найденный элемент будет в списке последним, а последний найденный элемент – первым (рис.8.6). Чтение списка начинается с заголовка и продолжается по адресам связи до нулевого адреса; нулевой заголовок означает, что список пуст.

Если же после элемента $C(j)$ списка требуется вставить новый элемент $C(i)$, достаточно: 1) адресное поле $A(j)$ переместить в адресное поле $A(i)$, 2) положить $A(j) = i$ (рис. 8.7).

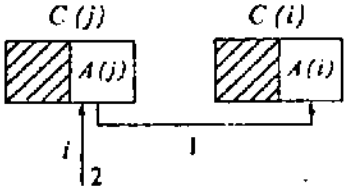


Рис. 8.7. Вставка нового элемента в список

Ценной особенностью списков является, как видно, отсутствие перемещений полей данных, которые могут иметь весьма значительный объем.

8.3. Организация хранения данных

Основными единицами хранения данных в компьютерах считаются файлы, поэтому такую организацию хранения часто именуют *файловой*. Файловая организация данных обычно обеспечивается лингвистической поддержкой: многие языки программирования имеют операторы работы с файлами.

Каждый файл снабжается специальной меткой в виде *управляющей записи*, содержащей имя файла и его характеристику. Обычно файлы постоянно или временно хранятся во внешней памяти компьютера. Каждый файл, в свою очередь, состоит из множества записей. *Запись* характеризуется длиной и форматом и представляет собой логически самостоятельную единицу – минимальный и недели-

мый элемент данных, уже не имеющий имени. Операционная система составляет *каталог имен* всех файлов с указанием адресов их хранения. С помощью этого каталога можно обратиться к конкретному файлу или отдельной записи в файле, указывая имя файла и дополнительную информацию о требуемой записи.

Физический носитель информации во внешней памяти считается *томом*. Например, отдельный магнитный диск или магнитная лента рассматриваются как том. В начале каждого тома обычно помещается метка, позволяющая его идентифицировать, а на магнитных дисках имеется еще и оглавление с указанием информации об областях хранения информации в томе.

Очень часто используемый тип организации хранения данных в компьютерах – библиотечный. Поскольку *библиотечный набор данных* обычно доступен многим пользователям, он называется разделяемым (partitioned data set). Библиотечные наборы данных размещаются только на устройствах с произвольным доступом (магнитных дисках), строятся в виде совокупности поименованных разделов и снабжаются оглавлением – справочником с указанием имени, адреса и размера каждого раздела. Отметим, что загрузка программ в оперативную память по имени возможна только для библиотечной организации. В частности, загрузочные модули программ до их выполнения хранятся в библиотеках, и каждый загрузочный модуль составляет один раздел библиотеки.

Различают системные, личные и временные библиотеки. *Системные библиотеки* служат для хранения системных программ ОС, трансляторов, редактора связей; в них может также храниться часть прикладных программ. Общесистемная библиотека всегда доступна всем пользователям, а системные библиотеки различных трансляторов обычно заказываются при постановке задач на трансляцию и счет.

Личные библиотеки служат для хранения программных (числовых, символьных) наборов данных пользователей и могут быть *каталогизированными* и *некаталогизированными* (сохраняемыми). Для использования каталогизированной личной библиотеки указывается ее имя, для некаталогизированной – имя и местонахождение.

Временные библиотеки создаются только для выполнения того задания, в котором они созданы; например, во временных библиотеках хранятся загрузочные модули программ перед их выполнением.

8.4. Базы данных

База данных – именованная совокупность данных, адекватно отображающих состояние объектов и их взаимосвязей в некоторой предметной области и организованных таким образом, что данные могут использоваться для решения многих задач многими пользователями. Предметная область в общем случае складывается из множества реальных объектов, обладающих некоторым набором свойств, – *атрибутов*. Отображению в базе данных подлежат лишь существенные атрибуты, несущественными можно пренебречь.

Чтобы различать объекты в базе данных, каждый объект снабжается идентификатором, который его однозначно определяет. Такой идентификатор называется *уникальным*, или *первичным ключом*. В качестве первичного ключа используется какой-либо атрибут или комбинация атрибутов (например, табельный номер рабочего или его фамилия, имя, отчество). Все объекты разбиваются на группы – типы объектов с одинаковыми атрибутами. Обозначение множества объектов каждого типа является *вторичным ключом*. Все объекты этого множества имеют одинаковый вторичный ключ.

Между объектами предметной области могут существовать два типа связей: *обязательные* и *факультативные*. Так, между объектами типов «экзамен» и «предмет» (учебная дисциплина) существует обязательная связь «проводится», так как каждый экзамен проводится по некоторому предмету. В то же время для типов «предмет» и «экзамен» связь «имеет» является факультативной, так как экзамены существуют не по всем предметам. Возможны не только парные связи, но и связи любой размерности, например тройные связи между типами объектов «завод», «продукция», «потребитель». Каждый экземпляр такой связи указывает для некоторого завода один из видов производимой им продукции и ее потребителя.

Структура предметной области в целом представляется в виде множества типов объектов и множества связей между ними. Структура базы данных должна отражать структуру предметной области так, что каждому объекту предметной области соответствует некоторый объект базы данных, а связи между объектами предметной области – связь между соответствующими объектами базы данных. Типы объектов предметной области, индивидуальные экземпляры объектов и связи могут изменяться со временем: могут появляться

или исключаться отдельные типы, устанавливаться или разрушаться связи, появляться или исключаться индивидуальные экземпляры объектов и их связи, изменяться значения атрибутов объектов. Соответственно, должна обновляться (актуализироваться) и база данных, так что каждому состоянию предметной области отвечает адекватное состояние базы данных. Иногда, например, в экономических базах данных, кроме текущего состояния предметной области сохраняется некоторая его предыстория.

Существуют централизованные и распределенные базы данных. *Централизованная база данных* хранится в памяти одного компьютера или одной вычислительной системы. Если этот компьютер (вычислительная система) входит в состав вычислительной сети, то к ее базе данных могут иметь доступ пользователи и других компьютеров сети. Централизованные базы данных часто используются в локальных сетях персональных компьютеров. *Распределенная база данных* состоит из нескольких частей, которые хранятся в памяти различных компьютеров вычислительной сети. При этом допускается частичное дублирование данных, но специальное программное обеспечение позволяет пользователю рассматривать распределенную базу данных как единое целое. Вся информация о размещении данных в распределенной базе содержится в *системном справочнике*, части которого также могут храниться в памяти различных компьютеров сети. Управление такой базой представляет собой сложную задачу.

Пользователями базы данных являются прикладные программисты, а также специалисты в конкретной предметной области; последних иногда называют *конечными пользователями*. Одно из важных требований к базе данных – *интегрированность*, которая понимается как эффективность работы с ней в среднем по всем категориям пользователей, а не для отдельных пользователей. При реализации требования интегрированности должны учитываться приоритеты пользователей (или задач), частота решения отдельных задач, запросы вычислительных ресурсов и другие факторы, в частности минимизация избыточности (дублирования) данных, поскольку избыточность данных легко может нарушать их непротиворечивость. Построение такой интегрированной базы данных обычно осуществляется с использованием машинного эксперимента, т.е. моделирования работы с базой данных на компьютере.

Функционирование базы данных как коллективного информационного ресурса многих пользователей обеспечивает специальная большая программа – *система управления базой данных* (СУБД), предназначенная для создания коллективной базы данных и ее актуализации, загрузки данных, организации эффективного доступа пользователей к данным и управления доступом, сохранения независимости и целостности данных, рационального использования вычислительных ресурсов и других целей. Аналогично тому, как доступ пользователей к ресурсам компьютера возможен только через ее ОС, доступ пользователей к базе данных возможен только через СУБД.

Различают СУБД общего назначения (универсальные) и специализированные. *Универсальные* СУБД пригодны для любых предметных областей и любых пользователей и имеют запас функциональных средств, который позволяет производить настройку на работу в конкретных условиях применения с целью повышения эффективности использования. При такой настройке могут выбираться наиболее эффективные методы доступа, модифицироваться организация хранения данных, изменяться структура хранимых данных и их размещение для увеличения производительности работы или экономии памяти. *Специализированные* СУБД предназначены для высокоэффективного конкретного применения. Поскольку разработка СУБД требует больших затрат труда и времени, специализированные СУБД создаются лишь при крайней необходимости. Дополнительная специфическая задача системы управления распределенной базой данных – организация работы базы как единого целого.

Выполняемая СУБД функция обеспечения независимости данных заключается в сохранении логической и физической независимости. При этом под *логической независимостью* понимается возможность изменять логическое представление данных, не меняя их физическое представление в среде хранения. Благодаря этому пользователи могут иметь различные логические взгляды на одну и ту же физическую базу данных. Аналогично, под *физической независимостью* данных понимается возможность изменять их физическое представление (хранение), не меняя логическое представление. Независимость данных означает также, что при различных модификациях, расширениях и реорганизациях базы данных не требуется вносить изменения в уже существующие прикладные программы, предназначенные для обработки данных.

Функция СУБД по обеспечению целостности данных также состоит в поддержании логической и физической целостности. *Логическая целостность*, под которой понимается непротиворечивость данных, должна поддерживаться при каждом пополнении или изменении данных и обеспечивается блокировкой ввода неверных данных, проверкой правильности обработки данных, сохранением данных при внешних прерываниях и согласованием данных в базе при всех изменениях. Так, при исключении в базе данных какого-либо предприятия в связи с его ликвидацией должна быть исключена и вся производимая им продукция, а также пересмотрен список потребителей этой продукции; при исключении какого-либо поставщика должны быть исключены и все его будущие поставки и т.п. Поэтому сохранение целостности накладывает жесткие ограничения на допустимые изменения в базе данных, которые называются *ограничениями целостности*. Различают статические ограничения целостности, имеющиеся для всех состояний базы данных, и динамические, определяющие лишь возможные переходы из одного состояния в другое.

Обеспечение *физической целостности* данных заключается в защите данных от разрушения при сбоях (отказах) аппаратуры и подразумевает автоматическое восстановление разрушенных данных путем использования их дубликатов.

С поддержанием целостности данных тесно связана проблема *управления доступом* к данным, т.е. ограничением доступа для определенных категорий пользователей. Например, для некоторых категорий пользователей может запрещаться изменение или пополнение данных; некоторые данные могут выдаваться не всем пользователям и т.д. Для управления доступом используются пароли – *замки управления доступом*.

Сложной задачей является организация параллельного доступа к данным. Необходимо, в частности, исключить случаи, когда один из пользователей обращается к данным, которые в это же время изменяются другим пользователем, поскольку при этом может нарушаться логическая целостность данных. Тогда используется блокировка изменяемых информационных ресурсов и специальный комплекс мер по сохранению целостности данных, называемый *транзакцией*. При этом, однако, возможно возникновение тупико-

вой ситуации, когда две транзакции одновременно блокируют друг у друга ресурсы; имеются способы преодоления этого тупика.

Взаимодействие прикладных программ и пользователей с СУБД называется *манипулированием данными*. Множества структур данных, ограничений целостности и операций манипулирования данными составляют *модель данных*. Обычно каждая СУБД поддерживает какую-либо одну модель данных, хотя существуют и мультимодельные СУБД, которые могут поддерживать несколько моделей. В настоящее время наибольшее применение находят иерархическая, сетевая и реляционная модели данных.

Иерархическая модель данных представляет собой множество деревьев, каждое из которых отображает иерархию объектов предметной области относительно одного из объектов, помещенного в корне дерева. Так, для предметной области «вуз» могут быть построены деревья: «вуз – факультеты – кафедры – студенческие группы – студенты», «вуз – корпуса – этажи – помещения – занятость помещений», «кафедра – занимаемые помещения – преподаватели – учебные дисциплины – группы – виды занятий – учебные часы и виды контроля успеваемости», «учебная дисциплина – кафедры – преподаватели – группы – виды занятий – учебные часы и виды контроля успеваемости» и т.д. Иерархическая модель данных отличается простотой, но обладает большой информационной избыточностью. Так, из приведенного выше примера видно, что одна и та же информация (кафедры, преподаватели, группы и др.) будет содержаться во многих деревьях. Большая избыточность информации приводит к излишней загрузке памяти ЭВМ и опасности нарушения целостности данных при изменениях в предметной области.

Вынужденная информационная избыточность иерархической модели данных объясняется тем, что она отображает лишь связи типа «один ко многим» и не отображает часто встречающиеся в реальности связи типа «многие ко многим». Этот недостаток устраняется в *сетевой модели данных*, отображающей объекты предметной области и их взаимосвязи в виде графа: объектам отвечают вершины графа, а связям – ребра. При этом ребра в общем случае могут быть кратными; например, вершины «кафедра» и «корпус» могут иметь две связи, одна из которых означает, что кафедра размещается в корпусе, а другая – что кафедра проводит занятия в корпусе. В целом сеть (граф) обычно отличается большой слож-

ностью, а движение («*навигация*») по сети регламентируется специальными правилами.

Реляционная модель данных, предложенная в 1970 г. американским ученым Коддом, основана на представлении данных в виде отношений между ними (relation – отношение), изображаемых двумерными прямоугольными таблицами. Кодд показал, что в виде двумерных таблиц можно представить любые, самые общие отношения между данными, в том числе и такие, которые изображаются деревьями (иерархическая модель) или графами (сетевая модель). Однако представление данных таблицами является более естественным и лучше понимается пользователями. Каждая строка таблицы содержит значения (x_1, x_2, …, x_n) свойств (атрибутов), которыми обладает некоторый объект данного типа; здесь n – число столбцов в таблице, называемое *степенью отношения*. Набор (x_1, x_2, …, x_n) называют *кортежем*, а множество значений одного атрибута, отвечающее столбцу таблицы, – *доменом.* Простейшим примером отношений могут служить таблицы учебных планов, помещаемые в студенческих календарных планах.

Можно дать строгое определение отношения следующим образом. Пусть имеются множества значений S_1, S_2, …, S_n (не обязательно различных). Тогда *отношением* R над множествами S_1, S_2, …, S_n называют множество таких кортежей (x_1, x_2, …, x_n) из n элементов, что в каждом кортеже

$$x_1 \in S_1,\ x_2 \in S_2,\ \ldots,\ x_n \in S_n\,. \qquad (8.19)$$

Для описания отношений и манипулирования ими в реляционной модели данных существует строгий математический аппарат, основанный на *реляционном исчислении* и *реляционной алгебре*. Это позволяет сжато и точно формулировать запросы пользователей; однако практическая реализация реляционной модели данных оказывается сложнее иерархической и сетевой.

Работа с данными в базах данных обеспечивается языковыми средствами в виде языка определения данных и языка манипулирования данными. *Язык определения данных* служит для описания типовых структур данных и правил их композиции, присваивания имен данным, учреждения ключей доступа и правил проверки достоверности данных. *Язык манипулирования данными* – совокуп-

ность средств взаимодействия прикладных программ с СУБД. Обычно указанные языки надстраиваются над некоторым базовым языком программирования. Многие основополагающие идеи организации баз данных и работы с ними, в том числе языковые средства, были заложены в так называемой *кодасиловской модели баз данных* (CODASYL, Conference on Data System Language, КОДАСИЛ – постоянно действующая конференция по языкам обработки данных).

Базы данных, в свою очередь, являются важнейшей составной частью современных автоматизированных систем управления и автоматизированных информационных систем.

8.5. Системы управления базами данных

В настоящее время существует много систем управления базами данных различных типов. Для персональных компьютеров широкое распространение получили так называемые *dBASE-подобные* реляционные СУБД семейств dBASE, FoxPro и Clipper, имеющие много разновидностей (версий). В этих СУБД формируются одинаковые рабочие файлы с расширением .dbf, каждый из которых представляет собой двумерную таблицу с фиксированным числом столбцов (полей) и переменным числом строк (записей). Достоинствами dBASE-подобных СУБД являются простота и наглядность использования (например, файлы .dbf на диске точно совпадают с таблицами на бумаге), но в целом системы этого типа сейчас считаются устаревшими, поскольку в них нельзя реализовать многие возможности реляционных баз данных и пользователям приходится их пополнять самостоятельно. Довольно популярны также различные версии реляционной СУБД Paradox фирмы ANSA Software, СУБД dB-VISTA фирмы Raima Corp для сетевой модели данных и др. Первоначальные версии указанных выше СУБД предназначались для работы в среде MS DOS, однако почти все они были в дальнейшем усовершенствованы и имеются версии для ОС Windows.

Из более современных систем большое распространение получили реляционные СУБД Approach фирмы Lotus, Paradox фирмы Borland и в особенности Access, включенная в пакет прикладных программ MS Office и намного превосходящая по своим возможностям СУБД dBASE. Эти СУБД могут поддерживать OLE-

технологию и манипулировать не только текстовой информацией, но также звуковой и видеоинформацией (печатными или телевизионными изображениями, фрагментами кинофильмов и т.п.). Так, СУБД Access имеет «дружелюбный» (к пользователю) интерфейс, легко изучается и может функционировать на изолированных персональных компьютерах или в локальных вычислительных сетях; позволяет проектировать базовые объекты информационных сетей, имеющие вид двумерных таблиц с различными типами данных, включая поля объектов OLE, устанавливать связи между объектами, поддерживать целостность данных; удобно осуществлять ввод, сохранение, просмотр, модификацию, сортировку, выборку, фильтрацию и другие манипуляции данными; формировать запросы (о данных) и отчеты (формы представления данных для вывода на печать); выполнять многие другие действия с использованием многооконной технологии, раскрывающихся (ниспадающих) меню и панелей инструментов.

При работе с СУБД Access пользователь может легко готовить собственные приложения на языках Access Basic или VBA (Visual Basic for Applications). Произвольный запрос в СУБД Access можно создать с помощью универсального языка SQL (аббревиатура от слов Structured Query Language), предназначенного, в основном, для профессионалов, однако ввиду сложности этого языка часто можно ограничиться более простым механизмом, который называется QBE (Query By Example – запрос по образцу) и состоит в заполнении по определенным правилам одного из специальных «бланков QBE», выдаваемого системой и имеющего вид небольшой таблицы. Запросы, в частности, позволяют выполнять различные групповые операции над данными, возможно, с использованием различных условий и функций алгебры логики.

Описанные выше СУБД иногда называют «настольными», подчеркивая этим их предназначение для ведения относительно небольших баз данных, которыми могут пользоваться отдельные абоненты или небольшие коллективы. Для мощных производственных информационных систем с большим числом пользователей лучше подходит более совершенная модель обработки данных, называемая «клиент-сервер». *Технология клиент-сервер*, или иначе *архитектура клиент-сервер*, составляет сейчас основное направление развития многопользовательских информационных систем и предназначена

для преодоления трудностей, возникающих вследствие практически бесконтрольного и массового манипулирования данными в базе данных одновременно многими абонентами. Одним из примеров таких трудностей являются транзакции, указанные в разд. 8.4. Основополагающий принцип технологии «клиент-сервер» – разделение важнейших функций манипулирования данными между программами-приложениями на абонентских терминалах («клиентах») и СУБД, установленной на мощном центральном компьютере информационной системы – *сервере баз данных*. Технология «клиент-сервер» наилучшим образом обеспечивает высокую производительность и надежность информационной системы, в том числе целостность данных, защиту от несанкционированного доступа, мобильность (возможность использования в информационных системах с различными техническими средствами, или, как говорят, с различными платформами), простоту установки и сопровождения, эффективность и несложность разработки приложений, «дружелюбный» пользовательский интерфейс [55]. «Клиенту» информационной системы с такой технологией достаточно лишь корректно сформулировать запрос и придать требуемую форму выданному сервером результату, все остальные действия выполняет сервер.

Архитектура «клиент-сервер» является открытой; это означает, что функциональное наполнение, программное обеспечение, состав и численность клиентов информационной системы могут легко изменяться в ходе эксплуатации. Ряд фирм выпускает коммерческую продукцию для архитектуры «клиент-сервер», среди которой можно отметить серверы баз данных Oracle Server фирмы Oracle и SQLBase фирмы Centura, а также средства разработки приложений Visual Basic фирмы Microsoft и PowerBuilder фирмы Powersoft. Заметим еще, что принцип разделения функций и программного обеспечения между клиентом (приложением) и сервером оказывается плодотворным не только для сетей компьютеров, но и для изолированного компьютера, на котором может находиться как приложение, так и сервер.

8.6. Автоматизированные информационные системы

К массовым задачам обработки данных относятся ввод, хранение, отбор, группирование, сортировка данных; генерирование отчетов

различной формы; систематизация и агрегирование данных. Типичными приложениями, в которых возникают указанные задачи, являются проблемы экономики, управление производством, снабжение, транспорт, складское хозяйство, статистическая отчетность, инвентаризация, архивы, библиотечное дело, финансы и банки, кадры и др. Массовость и трудоемкость решения таких задач требует применения автоматизированных информационных систем.

Автоматизированная информационная система (АИС) – функционирующий на основе компьютеров и других технических средств информатики комплекс для сбора, хранения, актуализации (обновления) и обработки информации. Основу комплекса АИС составляют технические средства (компьютер или сеть компьютеров с периферийным оборудованием), программное обеспечение, языковые средства и информационные ресурсы. Обычно выделяют два крупных класса АИС – документальные и фактографические, хотя встречаются и комбинированные системы. *Документальные* АИС предназначены для работы с документами на естественном языке, такими, как книги, статьи, отчеты, труды конференций, патенты, диссертации, авторефераты, архивные материалы. *Фактографические* АИС, в отличие от документальных, оперируют непосредственно фактическими сведениями из соответствующей предметной области, которые извлекаются из документов и представляются в виде специальных форматизованных записей, созданных средствами технологии баз данных. Наконец, *комбинированные документально-фактографические* АИС с помощью общих технических и программных средств могут обрабатывать как документальную, так и фактографическую информацию, причем их информационные ресурсы могут быть как раздельными (*кооперированные* АИС), так и частично или полностью совмещенными (*интегрированные* АИС).

Наиболее распространены *информационно-поисковые системы* (ИПС) для накопления и поиска информации по запросам пользователей (абонентов). Как и АИС, существуют *документальные* и *фактографические* ИПС. ИПС содержат программные средства, поисковый информационный массив данных (документов, сведений), отражающих достигнутый уровень знаний о предметной области, и средства поддержки информационного языка. По мере эволюции знаний о предметной области изменяется и поисковый информационный массив, причем возможные изменения также

регламентируются ограничениями целостности данных. Отображаемая в ИПС структура предметной области и ограничения целостности составляют информационно-логическую, или, короче, *инфологическую модель предметной* области.

Программные средства ИПС оформляются в виде пакетов управляющих и прикладных программ и служат для организации и хранения информации, организации взаимодействия пользователей с системой, обработки пользовательских запросов и выдачи результатов поиска, обеспечения эффективного использования ресурсов компьютера и надежного функционирования системы при одновременной работе многих абонентов. Наиболее продуктивной технологией работы современных ИПС является технология «клиент – сервер», описанная в разд. 8.5.

Поисковый массив данных документальной ИПС составляет ее базу данных. Обычно он не содержит непосредственно текстов документов. Так, автоматизированные библиографические системы, являющиеся наиболее распространенными ИПС, оперируют библиографическими данными, или, реже, рефератами и аннотациями. В фактографических ИПС сведения о предметной области в виде записей на специальном информационном языке также сохраняются в базе данных или нескольких базах данных. В дальнейшем, для определенности, ограничимся документальными ИПС.

Основная функция ИПС – выявление элементов информационного поискового массива, которые отвечают на запрос, предъявленный системе. Запрос составляется пользователем на естественном языке, а затем автоматически преобразуется в *поисковый образ запроса* – некоторое формализованное предписание. Для поиска используются записанные на информационном языке *поисковые образы документов* в виде формализованных объектов, отражающих содержание документов, сохраняемых в базе данных. Поисковые образы обычно формируются при вводе документов в базу данных специалистом в соответствующей предметной области. Для этого осуществляется *индексирование документов*: снабжение их ключевыми словами, рубриками и другими указателями. Возможно также и автоматическое индексирование документов средствами самой ИПС.

При обработке запроса поисковый образ запроса сравнивается с поисковыми образами документов по некоторому критерию, который называют *критерием смыслового соответствия*. Документы,

удовлетворяющие этому критерию, считаются *релевантными* запросу и в ответ на поступивший запрос выдаются сведения обо всех релевантных документах. Правильная реализация критерия смыслового соответствия – одна из наиболее сложных проблем в работе ИПС. При этом надо исключить как неполноту поиска, так и «информационный шум», под которым понимаются случайные погрешности поиска, вследствие которых может не выдаваться часть релевантных документов или, наоборот, выдаются лишние, нерелевантные документы. Основные причины указанных недостатков поиска – неудовлетворительное индексирование документов, слабые языковые и программные средства.

Информационные языки ИПС, называемые также *информационно-поисковыми языками*, реализуются на основе некоторого подмножества естественного языка. Для преодоления трудностей, связанных с интерпретацией высказываний на естественном языке, информационно-поисковые языки строятся как языки *дескрипторного типа*. Такие языки основаны на использовании *тезаурусов* – специальных словарей, описывающих лексические единицы (слова и устойчивые словосочетания) информационно-поискового языка, называемые *дескрипторами*, и смысловые связи между ними. Тезаурусы, следовательно, предназначены для поиска слов по их смыслу.

В качестве дескриптора могут быть выбраны ключевое слово, словосочетание или цифровой код типа кодов УДК (универсальная десятичная классификация в библиографии), указывающий раздел предметной области. Тезаурус позволяет находить нужные дескрипторы и правильно индексировать документы, однозначно переводить тексты с естественного языка на дескрипторный информационно-поисковый язык, составлять поисковые образы документов и запросов. Типичный информационно-поисковый тезаурус имеет три раздела:

1) алфавитный список дескрипторов;

2) указатель отношений между дескрипторами;

3) руководство по переводу ключевых слов и словосочетаний естественного языка на дескрипторный информационно-поисковый язык.

Различают два типа отношений между словами естественного или дескрипторного языка: парадигматические и синтагматические отношения. *Парадигматические* (иммаментные, внутренне присущие) отношения между словами (понятиями) существуют всегда, независимо от контекста. Парадигматические отношения, в свою

очередь, делят на *отношения подчинения*, отвечающие связям вида «класс – подкласс» или «целое – часть» (например, «жидкость» – «нефть»), и *ассоциативные отношения*, включающие все другие имманентные связи (например, «жидкость» – «текучесть»). *Синтагматическими* называют ситуативные отношения, возникающие между словами лишь в определенном контексте, т.е. в определенной ситуации, объединяющей обозначенные словами объекты, процессы, явления (например, «ускорение электрона» – электрон есть объект ускорения, «столкновение ядер» и т.п.). Тип отношений учитывается при организации поиска.

Поскольку состав и отношения объектов предметной области все время меняются, активно функционирующая АИС должна находиться в процессе постоянного обновления, который также должен быть автоматизирован. Создание и широкое использование разнообразных АИС – важнейшая задача информатики.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Символ, слово, запись, файл, что понимается под этими терминами в информационных системах?
2. Укажите два сновных типа структур данных, используемых при разработке информационных систем. В чем их принципиальное различие?
3. Дайте определение различным видам линейных структур данных и укажите на их характерные особенности.
4. Перечислите основные виды нелинейных структур данных и отобразите их структуру.
5. Рассмотрите основные виды работ с данными типа строка и таблица.
6. Перечислите основные способы организации хранения данных.
7. Укажите основные модели данных, используемых в СУБД. В чем их достоинства и недостатки?
8. Рассмотрите основные функции СУБД. Расскажите о физической и логической организации СУБД.
9. Перечислите разновидности СУБД и различные области их применения.

ГЛАВА 9. ТЕХНОЛОГИЯ ОБРАБОТКИ ТЕКСТОВОЙ И ГРАФИЧЕСКОЙ ИНФОРМАЦИИ В ПК IBM-КЛОНА

9.1. Функции текстовых редакторов

Инструментальные программы для подготовки текстов программ, документов, описаний называются *текстовыми редакторами* (text-editor). Мощные текстовые редакторы с расширенным спектром функций называют *текстовыми процессорами* (word-processor). Некоторые текстовые процессоры могут работать не только с текстами, но и с изображениями, например редактировать иллюстрированные документы.

Основные функции текстовых редакторов и процессоров:

работа с файлами – сохранение текста на магнитном диске в виде файла, считывание текста (файла) с диска, копирование в редактируемый текст любого количества строк из другого файла, имеющегося на диске (для надежности редактируемый файл должен периодически записываться на диск по команде пользователя или автоматически);

показ текста на экране – текст или его фрагмент можно показывать в специальном окне объемом 15 – 25 строк, организованном на экране монитора; текст на экране можно передвигать вверх-вниз, влево-вправо с помощью клавиш, помеченных стрелками, а также быстро заменять фрагмент текста на другой фрагмент по номеру строки (некоторые редакторы позволяют организовывать на экране несколько окон с различными файлами или с различными частями одного файла);

вывод на печать (если в составе компьютера имеется принтер); обычно в редакторы, работающие со многими шрифтами и алфавитами, встроена функция вывода на печать, хотя некоторые простые редакторы сами на печать не выводят и требуется сначала записать текст на магнитный диск, выйти из редактора и вывести файл на печать средствами операционной системы;

вставка символов и строк в места, указанные курсором (при этом текст раздвигается);

перемещение части текста, помеченного соответствующим образом (обычно помечаются начало и конец фрагмента), на другое

место, указанное курсором, или дублирование части текста в другом месте;

удаление символов и строк, указанных курсором и помеченных соответствующим образом, при этом текст сжимается (обычно редакторы позволяют также восстанавливать ошибочно удаленные фрагменты текста);

контекстный поиск – поиск строки по заданному фрагменту текста;

выравнивание ширины – выравниваются правый край, левый край или «по центру строки» путем вставки дополнительных пробелов (переносы слов при этом не делаются);

перенос слов, простейшие редакторы не используют перенос слов и если слово не помещается в строке, то оно целиком переносится на следующую строку; более мощные редакторы, «понимающие» грамматику языка текста, могут выполнять переносы слов (это удобно при подготовке текста к типографской печати); некоторые редакторы можно «обучать» переносу слов;

резка и склейка строк (можно отделить часть строки и перенести в следующую строку и, наоборот, строку «подклеить» к предыдущей строке);

замена одного фрагмента на другой (можно произвести замену одного фрагмента на другой, например имя «Pascal» заменить во всем тексте на «Basic», автоматически или полуавтоматически (с контролем); можно заменить прописные буквы на строчные, один год на другой и т.п.);

вставка заготовок (можно вставлять заранее заготовленные фрагменты (имена, служебные слова языков программирования) в предварительно помеченные места текста);

орфографический и синтаксический контроль текста с указанием цветом или подчеркиванием мест ошибок или непонятных редактору слов и выражений («обучаемые» редакторы можно научить понимать эти слова и выражения в дальнейшем).

9.2. Сравнительные характеристики распространенных текстовых редакторов

Текстовые редакторы разделяются на три группы:

1) *редакторы общего назначения* (сюда относятся, например, редакторы «Лексикон», MS Word, WordPerfect и др.);

2) *редакторы научных текстов* (ChiWriter, TeX и др.);

3) *редакторы исходных текстов программ* (например, Multi-Edit или встроенные редакторы систем программирования для языков Бейсик, Фортран, Си и др.).

По сравнению с редакторами общего назначения специализированные редакторы позволяют легче готовить и редактировать соответствующие тексты: например, редакторы научных текстов удобнее для набора математических или химических формул.

Многооконный редактор «Лексикон» – один из простейших текстовых редакторов отечественной разработки для несложных документов на русском или английском языке. Ранее редактор «Лексикон» работал в MS DOS, последние версии редактора предназначены для ОС Windows. Имеются так называемый «базовый вариант» редактора и более мощный профессиональный вариант для работы с пакетом MS Office в ОС Windows. «Лексикон» позволяет открыть на экране монитора до десяти окон, в каждом из которых может редактироваться свой документ.

В редакторе «Лексикон» предусмотрены следующие возможности:

просмотр и исправление текста;

автоматическое форматирование абзацев;

автоматическое разбиение текста на страницы;

перемещение, выделение, удаление, вставка фрагментов текста;

создание оглавлений;

использование подчеркиваний, курсива, полужирного шрифта;

одновременное редактирование нескольких документов и обмен материалами между окнами;

работа в графическом режиме;

печать текста на принтерах разных типов;

печать верхних и нижних индексов, греческих букв, математических символов;

просмотр на экране вида напечатанного материала до печати;

набор текста в несколько колонок;

автоматическое сохранение редактируемого документа через несколько строк (страниц);

проверка орфографии;

контекстный поиск.

В последнее время редактор «Лексикон» быстро вытесняется более мощными (но и более сложными) редакторами.

Текстовый редактор MS DOS Editor может применяться только для редактирования текстов файлов канонического формата, состоящих лишь из букв и цифр и не имеющих изображений. Редактор может вызываться из MS DOS и Windows и по своим возможностям близок к «Лексикону», но имеет более удобный интерфейс.

Текстовый редактор Windows 95 NotePad («Блокнот») – простейший редактор, входящий в Windows 95. Имеет ограниченные возможности, но очень простой для изучения и использования. Среди других средств редактирования, используемых в ОС Windows, отметим текстовые процессоры Word Pad, Word 6.0, Word 95 (Word 7.0), Word 97 (Word 8.0), Word 2000 (Word 9.0). Многооконные процессоры Word 95, 97, предназначенные для ввода, редактирования и форматирования текстов, могут работать одновременно с несколькими документами, каждый из которых отображается в своем окне. Можно менять размеры окон, а также вид и начертание шрифтов, оформлять текст в несколько столбцов, выполнять многие другие операции.

Рассмотрим, например, многооконный текстовый процессор MS Word 2002 (одна из последних версий редактора MS Word) – один из самых совершенных, входит в пакет MS Office 2002 и имеет сотни операций над текстовой и графической информацией. Как и ОС Windows, построен по оконной технологии с использованием ниспадающих меню и пиктограмм. Обладает универсальностью, позволяет достаточно быстро и с высоким качеством готовить практически любые документы: от деловой записки или письма до макета книги. Имеет множество технологических приемов редактирования текстово-графической информации, широкий набор шрифтов разной формы и размера, проверку орфографии и грамматики, автоматический перенос слов. Можно делать ссылки в тексте (это позволяют не все редакторы), работать с готовыми шаблонами документов, отключать часть средств или модифицировать используемые средства для ускорения работы. В текст можно включать рисунки, которые можно редактировать, изменяя масштаб, форму, качество изображения; формировать «обтекание» рисунка текстом, использовать рисунок в качестве фона. Позволяет вставлять в текст таблицы различного размера и сложности, поворачивать текст на странице, вставлять номера страниц и многое другое. Благодаря этим возможностям с помощью редактора Word может выполнять-

ся верстка документов, как это делается в некоторых типографиях. Имеются средства, облегчающие подготовку документов для сети Internet.

Усовершенствованием рассмотренных выше редакторов Word является более удобный в работе текстовый процессор WordPerfect, имеющий ряд версий. Редактор контролирует правописание, вычерчивает отрезки линий, оформляет рамки, формирует ссылки, оглавления, указатели, имеет много других возможностей.

Универсальность процессоров Word сопряжена, однако, и с некоторыми недостатками, к которым относятся трудности освоения и относительно низкая производительность при редактировании простых текстов. Поэтому для подготовки простых однородных текстов иногда целесообразно сначала воспользоваться несложными редакторами типа MS DOS Editor или «Лексикон», затем отформатировать полученный текст для Word и уже с помощью Word улучшить его в изобразительном отношении. При наличии в тексте большого количества сложных математических или химических формул более производительным может быть, например, редактор ChiWriter. Наконец, редакторы Word не подходят для подготовки такой сложной полиграфической продукции, как атласы, высококачественные рисунки (копии фотографий), иллюстрированные альбомы.

9.3. Типы и характеристики компьютерной графики

Современные компьютеры все шире применяются для построения изображений (рисунков), используемых в научных исследованиях; для наглядного представления результатов; в конструкторских разработках, тренажерах, компьютерных играх; в инженерном, издательском, рекламном деле и других областях. Компьютерная графика служит основой *анимации*, под которой понимается изменение вида, формы, размеров, расположения объектов на экране, создающее эффект мультипликации. Различают три основных типа компьютерной графики: растровая, векторная и фрактальная. Обычно особо выделяют еще трехмерную (3D – three-dimensional) графику как средство построения объемных изображений. По цветности различают черно-белую и цветную компьютерную графику, а по областям применения – инженерную, научную, дело-

вую, игровую (развлекательную) компьютерную графику, компьютерную полиграфию и другие типы.

В *растровой графике* изображение строится как множество точек, так называемых пикселов. *Пиксел* (сокращение от слов picture cell – элемент изображения) представляет собой единицу измерения разрешения экрана (монитора) или печатного изображения и соответствует отдельной светящейся точке, цветом и яркостью которой можно управлять. Растр экрана монитора с диагональю 20 – 21" может содержать от 0,3 до 3 млн пикселов. Поскольку изображение может быть цветным, для кодирования одного пиксела может потребоваться до трех байт информации. На весь экран, следовательно, может потребоваться от 1 до 10 Мбайт, т.е. весьма значительный объем, но изображение, тем не менее, может быть довольно грубым. Качество изображений принято оценивать по числу пикселов на 1" длины. Единицу такого измерения называют dpi – dots per inch. Для газетных иллюстраций достаточно около 70 dpi, для полноцветной полиграфической печати 200-300 dpi, для фотоэкспонирующих устройств профессионального класса 2500 dpi, тогда как экран монитора обычно обеспечивает лишь несколько десятков (например, 70) dpi и расстояние между соседними точками около 0,25 мм, что недостаточно для получения изображений высокого качества.

При растрировании изображения на него как бы накладывается сетка линий, разбивающая его на квадратные ячейки. Число линий на дюйм Lpi (Lpi – lines per inch) называется *линиатурой.* Для лазерных принтеров рекомендуемая линиатура составляет 65 – 100, для газет 65 – 85, для книг и журналов 85 – 133, для художественных и рекламных работ 133 – 300. Интенсивность тона – *светлота,* определяется числом точек (пикселов) в ячейке растра. Для человеческого глаза рекомендуется 256 уровней тона, т.е. в ячейке должно помещаться до $16 \cdot 16 = 256$ пикселов. Для изменения уровня тона можно также изменять размеры пикселов; максимальный размер пиксела равен, очевидно, размеру ячейки растра.

Растровая графика позволяет строить изображения очень высокого качества, но, как видно из приведенных оценок, для этого требуется очень большой объем компьютерной памяти (например, для журнальной иллюстрации – до 130 Мбайт и более). Помимо больших запросов на память, недостатком растровой графики являются

трудности увеличения изображения для анализа его деталей. Поскольку при увеличении объем запасенной информации сохраняется, то без принятия специальных мер увеличение изображения приводит лишь к *пикселизации* – укрупнению отдельных пикселов с сохранением их численности. Изображение становится все более «зернистым» (фотографический термин), грубым, искаженным. Сглаживание пикселизации представляет собой самостоятельную проблему.

Если основным элементом растровой графики является точка, то в *векторной графике* основным элементом является линия (прямая или кривая). Объем памяти, требуемый для хранения линии, не зависит от ее длины, так как в памяти компьютера линия представляется формулой с несколькими параметрами, а не точками. Обычно ограничиваются линиями не выше третьего порядка, для построения которых достаточно иметь не более девяти коэффициентов и абсциссы двух концевых точек. При изменении размера линии меняются лишь параметры, а объем памяти сохраняется. Линии имеют свойства: форму, цвет, толщину, тип (сплошные, штриховые, пунктирные и т.п.). Для сохранения одной линии достаточно 20 – 30 байт оперативной памяти. Замкнутые линии имеют свойство заполнения. Заполнение описанного линией контура может быть выполнено цветом (несколькими цветами) или текстурой (узором). Концы линии – узлы, которые также обладают свойствами, например, могут быть точками, кружками, квадратиками, различными фигурками. На экран линия по-прежнему выводится точками (таковы особенности монитора), но координаты этих точек вычисляются, а не хранятся в памяти компьютера.

Линия – элементарный, простейший объект векторной графики. Простейшие объекты могут объединяться в более сложные, например плоские и объемные, фигуры. Типичные объекты сохраняются в памяти компьютера. Векторная графика позволяет легко увеличивать изображение или его фрагменты, например план дома или квартиры, чертеж механизма или схемы с сохранением их качеств: можно поворачивать изображения, совмещать их, изменять угол зрения, совершать другие манипуляции. При этом используются некоторые математические основы векторной графики. Даже достаточно сложные композиции, содержащие тысячи объектов, расходуют лишь десятки и сотни кбайт памяти. Векторная графика

мало пригодна для создания художественных изображений и обычно применяется в оформительских, чертежных, проектно-конструкторских работах, системах автоматизированного проектирования (например, архитектурного проектирования) и аналогичных приложениях.

Фрактальная графика, как и векторная, также вычисляемая, но в памяти компьютера не сохраняются никакие объекты, кроме их формул. Изображение строится согласно уравнению или системе уравнений. Меняя коэффициенты (параметры) уравнений, можно получить другое изображение. Характерная особенность фрактальной графики – наследование свойств. Например, фрактальный треугольник (точнее, его формулы) – простейший фрактальный объект. Можно построить треугольник другого размера с сохранением свойств исходного (например, равносторонний треугольник). Таким путем можно строить изображения необычного вида: декоративные узоры, орнаменты, имеющие очертания снежинок, кристаллов, листьев, сложных геометрических фигур.

Трехмерная графика широко применяется в таких областях, как научные расчеты, инженерное проектирование, моделирование физических процессов и технических объектов, а также в обучающих системах и «индустрии развлечений (игр)». Для создания модели трехмерного объекта используются *геометрические примитивы* (куб, параллелепипед, шар, эллипсоид, конус и др.) и гладкие поверхности, описываемые кусочно-гладкими бикубическими полиномами. Вид поверхности задается сеткой расположенных в пространстве опорных точек. Участки поверхности между опорными точками – границы объекта, которые обладают различными свойствами и могут быть гладкими, шероховатыми, прозрачными, непрозрачными, зеркальными и т.п. В соответствии с этими свойствами поверхности закрашиваются тем или иным способом. Движение объектов и анимация воспроизводятся движением геометрических примитивов и опорных точек по заданным законам. Для построения трехмерных изображений и анимации используется достаточно сложное алгоритмическое и программное обеспечение.

Цветность изображения характеризуется цветовой моделью и цветовым разрешением. Под *цветовой моделью* понимают способ разделения цвета на основные компоненты. В наиболее простой цветовой модели, используемой в мониторах и цветных телевизо-

рах, любой цвет считается состоящим из трех основных компонентов: красного, зеленого и синего цветов, смешанных в определенной пропорции. Совмещение трех основных компонентов в равной пропорции дает белый цвет. В такой модели цвет ячейки растра можно изобразить вектором, исходящим из начала координат в пространстве трех основных цветов. При этом проекции вектора дают относительный вклад основных цветов, а его модуль – интенсивность цвета. К трем основным цветам обычно добавляют для удобства еще черный цвет (цвет экрана). Имеются и другие цветовые модели.

Под *цветовым разрешением*, или *глубиной цвета*, понимается метод кодирования цветовой информации. И от него зависит, сколько цветов на экране может воспроизводиться одновременно. Таблица данных, в которой хранится информация о том, каким кодом закодирован тот или иной цвет, именуется *цветовой палитрой.* Если на кодирование цвета отводится 1 бит информации, изображение будет двухцветным (черно-белым); один байт информации позволяет закодировать 256 цветов, два байта – 65536 цветов (режим High Color), три байта – около 16,5 млн цветов (режим True Color). В последнем случае для кодирования каждого из трех основных цветов отводится один байт информации. При работе во всемирной сети Internet используется так называемая *безопасная палитра*, содержащая всего 216 цветов и жестко задающая их коды, а поэтому пригодная для любых компьютеров, подключенных к сети, в том числе не совместимых с IBM PC.

9.4. Программное обеспечение компьютерной графики

Программные средства работы с компьютерной графикой называют *графическими редакторами* (graphics editor). Существуют два типа графических редакторов:

1) редакторы для создания новых изображений;

2) редакторы для улучшения уже готовых изображений, полученных с помощью сканера, видеокамеры, цифрового фотоаппарата и других средств.

Рассмотрим сначала графические редакторы для работы с растровой графикой. Существует несколько десятков форматов, в ко-

торых сохраняются растровые изображения. Универсальным при работе с ОС Windows считается формат без сжатия информации Windows Bitmap, имеющий расширение .bmp. Для Web-документов в сети Internet более удобны форматы, обеспечивающие сжатие информации с целью сокращения объема файлов. Одним из таких форматов является формат JPEG (расширение .jpg), сохраняющий данные с огромной степенью сжатия, но за счет потери некоторой, относительно небольшой части информации. Также используемый в Internet формат GIF (расширение .gif) обеспечивает наивысшее уплотнение без потери информации. В полиграфии распространенным является специальный формат TIFF (расширение .tif), дающий неплохую степень сжатия и открывающий некоторые дополнительные возможности при печати изображений.

Из графических редакторов первого типа для растровых изображений очень популярен и относительно прост однооконный (на экране монитора открывается единственное окно) редактор Paint, входящий в программное обеспечение Windows. Редактор Paint представляет собой OLE-сервер, и созданные в нем изображения можно вставлять в документы таких универсальных приложений Windows, как MS Word, MS Excel, MS PowerPoint и даже в базы данных MS Access. Для создания и редактирования изображений Paint представляет богатый набор инструментов (палитры цветов, кисть, распылитель, ластики для стирания, «карандаши», ножницы для вырезания фрагментов) и средств работы с этими инструментами. Paint позволяет создавать довольно сложные черно-белые или цветные рисунки, схемы, чертежи, хотя и не обладающие высокими художественными или инженерно-техническими качествами. В силу своей простоты и доступности Paint часто используется в качестве первой ступени при обучении и при овладении более сложными средствами компьютерной графики.

К редакторам первого типа относятся также Painter компании Fractal Design, Free Hand компании Macromedia и Fauve Matisse. Редактор Painter обладает широкими возможностями средств рисования и работы с цветом, позволяя, в частности, имитировать различные инструменты (кисти, карандаши, перо, уголь) и материалы

(акварель, масло, тушь). Последние версии редактора Free Hand также содержат разнообразные средства редактирования изображений и текста, включая многоцветную градиентную заливку, библиотеку спецэффектов и др.

К графическим редакторам второго типа относятся, например, редакторы Adobe Photoshop, Photostyler, Picture Publisher, из них наиболее популярны редакторы Adobe Photoshop фирмы Adobe, фактически считающиеся стандартом в этой области. Из множества средств обработки готовых изображений отметим такие средства, как улучшение яркости и контраста, повышение четкости, *цветовая коррекция* (изменение яркости и контрастности в различных цветовых каналах), *отмывка* (изменение яркости фрагментов), *обтравка* (вырезание отдельных фрагментов и их последующее улучшение с возвратом, «вклеиванием», на прежнее место), *набивка* (восстановление утраченных элементов изображения путем копирования сохранившихся фрагментов), *растушевка* (сглаживание границ), *монтаж* (компоновка изображения из фрагментов одного или нескольких изображений). Интересным средством обработки изображений являются *фильтры* — программные средства преобразования изображений с целью улучшения их качества или художественной выразительности. С помощью фильтров можно повысить четкость изображения, придать фотографии вид карандашного или угольного рисунка, барельефа, гравюры, мозаики; выполнить стилизацию изображения, например имитировать изображение на ткани, бумаге, металле и других основах.

При выполнении этих преобразований графические редакторы предоставляют пользователю специальные инструментальные палитры в виде диалоговых окон с различными панелями (наборами) инструментов, указанных пиктограммами. Так, в редакторе Adobe Photoshop 4.0 имеется десять таких палитр. Основное отличие палитр от обычных диалоговых окон ОС Windows – возможность перекомпоновки рабочей среды пользователем путем перемещения палитр на экране и монтирования новых палитр.

Рассмотрим теперь графические редакторы, применяемые для работы с векторной графикой, когда изображение – чертеж, схема, диаграмма, но не рисунок.

Элементами векторной графики в графических редакторах служат линии, контуры, объекты. Эти элементы можно группировать, комбинировать, объединять, заливать различными способами, используя многочисленные меню и инструменты, обычно кодируемые пиктограммами. Вместе с рисунками можно создавать и тексты, причем не только строчные, но и фигурные, расположенные вдоль заданных кривых или в заданных контурах. Можно также видоизменять символы и шрифты, создавая необычные надписи, обладающие художественной выразительностью, например логотипы (краткие наименования) предприятий и фирменные стили для использования в объявлениях, рекламах, проспектах, а также для создания оригинальных электронных документов и Web-страниц в сети Internet. Отметим еще, что растровые изображения можно преобразовывать в векторные, а затем дорабатывать, улучшать с помощью редакторов векторной графики и, наоборот, векторные изображения преобразовывать в растровые с целью последующего редактирования, улучшения с помощью, например, такого мощного средства, как фильтры редактора Photoshop.

Одним из перспективных приложений средств компьютерной графики становятся в последнее время *настольные типографии* (desktop publisher) для печати малотиражных изданий, реклам, извещений, объявлений, листовок, а также *настольные издательские системы,* применяемые для оформления (верстки) документов, предназначенных для полиграфических изданий. Наиболее известными из настольных издательских систем являются QuarkXPress и PageMaker. Процесс верстки документа состоит в оформлении текста и взаимного расположения текста и иллюстраций на основе оконной технологии. Цель верстки – создание оригинала-макета, пригодного для последующего размножения документа полиграфическими средствами. Работа с настольными издательскими системами является объектно-ориентированной, объектами работы служат блоки текста, рисунки и стандартные элементы оформления (линии,

рамки и т.п.), причем блоки текста и рисунки могут быть подготовлены заранее с помощью текстовых и графических редакторов. Пользователю настольной издательской системы предоставляется набор действий, оформленных как меню, панель инструментов, панель размеров и панель макета документа. Для хранения наборов объектов, созданных пользователями, имеются библиотеки, которые можно пополнять в ходе работы. Из библиотек можно извлекать копии текстовых и графических объектов, используемых в верстке. Имеется также широкий набор средств для работы с цветом.

В заключение отметим, что техника квалифицированной работы с компьютерной графикой и настольными издательскими системами требует большого опыта ввиду многообразия имеющихся средств и манипуляций с ними.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Перечислите традиционные функции редакторов текста и поясните их назначение.
2. Рассмотрите общность и различие текстовых редакторов, получивших распространение для различных приложений.
3. Укажите главные направления использования машинной графики и основные виды реализаций графических изображений для различных областей применения.
4. Приведите основные характеристики растровой графики, определяющие качество изображения и цветовую палитру.
5. Опишите основные принципы реализации векторной графики, укажите основные достоинства такого вида отображения графических элементов и области его приложения.
6. Что такое фрактальная графика и в чем ее отличие от векторной?
7. Перечислите основные типы графических редакторов и области их применения.

ГЛАВА 10. ОФИСНЫЕ ТЕХНОЛОГИИ ИНФОРМАТИКИ

10.1. Виды офисных работ с использованием компьютеров

Компьютерная техника и компьютерные технологии широко используются для выполнения офисных работ. Основными видами офисных работ с применением компьютеров являются следующие:

автоматизация ввода документов в компьютер (называемая иногда преобразованием документов в электронную форму) с помощью сканеров различных типов и средств телекоммуникаций;

автоматическое распознавание текста (работа, тесно связанная с предыдущей); распознавание текста необходимо потому, что текст, введенный сканером в компьютер, представляет собой лишь набор разноцветных точек, а вовсе не осмысленный текстовый документ;

создание и обработка текстовых, графических и текстово-графических документов, к которым относятся письма, отчеты, объявления, приглашения, расписания, прейскуранты, бланки, рекламы;

финансовые и коммерческие расчеты и связанные с этим таблицы, сводки, деловые и бухгалтерские операции (приходные и расходные накладные, составление баланса, учет фондов, платежные требования и поручения, расчет зарплаты, кассовые операции) и т.п.;

управление работой персонала – составление расписаний и графиков работы сотрудников, учет кадров, подготовка приказов, совещаний, деловых встреч;

автоматический перевод деловых документов с одного языка на другой;

регистрация телефонных звонков, почтовых поступлений и отправлений;

использование электронной почты для рассылки и получения текстовой и графической информации, поиск существующих и подготовка новых материалов в сети Internet, в частности подготовка и модификация Web-страниц;

ведение различных баз данных;

организация электронного документооборота между отдельными сотрудниками, отделами, фирмами, в том числе зарубежными;

организация и поддержка электронных архивов;

подготовка презентаций на различные темы.

Для выполнения такого количества работ требуются, естественно, разнообразные аппаратные и программные средства и обученный персонал. Ряд офисных работ может выполняться с использованием мощного пакета программ Microsoft Office (например, MS Office 2002), содержащего приложения Word, Excel, Access, PowerPoint, Office Binder, Outlook, Internet Explorer. К офисным работам примыкают также некоторые виды работ, выполняемые с использованием экспертных систем.

10.2. Электронные таблицы

Многие данные в торговле, коммерции, бухгалтерских расчетах, снабжении, планировании и контроле удобно представлять в виде таблиц. Эффективным средством автоматизации работы с такими данными могут служить *электронные таблицы* (соответствующий английский термин – spredsheet) – структурные модели представления и обработки данных. В разное время был предложен ряд программ для работы с электронными таблицами – Lotus, SuperCalc, QuattroPro, но наибольшую известность приобрела, пожалуй, программа Excel, написанная средствами языка Visual Basic, имеющая ряд версий и включенная в пакет MS Office.

Объектами обработки в программе Excel являются документы в виде файлов с расширением .xls, называемые *рабочими книгами* (workbook). Каждая рабочая книга может содержать от 1 до 255 *рабочих листов* (worksheet) и на каждом листе размещается одна электронная таблица. Как и любая таблица, электронная таблица представляется в виде строк и столбцов. Строки нумеруются числами 1, 2, ... (по возрастанию), столбцы – латинскими буквами A, B, ... (в алфавитном порядке). В Excel допускается до 65536 строк и до 256 столбцов. Поскольку для 256 столбцов букв алфавита не хватает, допускаются обозначения из двух букв, т.е. после Z следуют AA, AB, ..., AZ, BA,... и т.д. На пересечении строк и столбцов располагаются ячейки таблицы. Адрес ячейки состоит из номера столбца и номера строки, например A2, B4, C6. Возможна также нумерация ячеек только числами по их расположению в таблице. При этом перед номером строки ставится буква R (от слова row – строка), а перед номером столбца – буква C (от слова column – столбец), например R1C1 – ячейка в левом верхнем углу

таблицы. Но существует и возможность присвоить отдельным ячейкам оригинальные имена, например «Income» (доход), «Demand» (спрос) и т.п. На экране монитора, помимо таблицы, имеются строка заголовка, строка меню, панели инструментов и *строка формул,* которая является характерной особенностью программы Excel. Ширину столбцов и высоту строк можно менять.

Ячейка таблицы может содержать текст, число или формулу. Признаком формулы является то, что она начинается со знака равенства. Excel допускает хранение в ячейке до 32000 символов, т.е. текст или формула могут быть достаточно длинными. При этом на экране отображается лишь начало длинной формулы, а остальная часть обрезается, но в строке формул вся введенная в ячейку формула будет выдана полностью. Формулы задают действия над числами в ячейках с использованием знаков арифметических операций +, –, * (умножение), / (деление), ** (возведение в степень) и широкого набора (до 400) функций. Например, квадратный корень из суммы чисел в ячейках A3 и B3 записывается в виде = SQRT(A3+B3), сумма чисел в ячейках от F2 до F6 обозначается краткой записью = СУММ(F2 : F6), среднее значение чисел в ячейках от D2 до D50 записывает как = СРЗНАЧ (D2 : D50) и т.д. Допускается использование случайных чисел и логических операций с условием «ЕСЛИ», описывающих выполнение вычислений в зависимости от некоторого условия. Набор функций может пополняться пользователем.

При изменении исходных значений в ячейках таблицы автоматически вычисляются все формулы. Соответственно, существуют два режима отображения электронных таблиц: режим отображения формул и режим отображения значений (рис. 10.1 и 10.2).

Магазин № 1

	A	B	C	D	E	F
1	Продукт	Цена	Поставлено	Продано	Осталось	Выручка
2	Сахар	14	200	100	= C2 – D2	= B2 * D2
3	Творог	18	100	90	= C3 – D3	= B3 * D3
4	Сыр	90	150	80	= C4 – D4	= B4 * D4

Рис. 10.1. Фрагмент электронной таблицы в режиме отображения формул

Магазин № 1

	A	B	C	D	E	F
1	Продукт	Цена	Поставлено	Продано	Осталось	Выручка
2	Сахар	14	200	100	100	1400
3	Творог	18	100	90	10	1620
4	Сыр	90	150	80	70	7200

Рис. 10.2. Фрагмент электронной таблицы в режиме отображения значений

Программа Excel позволяет работать с очень многими таблицами, размещенными на листах рабочих книг. Excel – многооконная программа. Одно окно на экране можно разделить на два или четыре подокна и одновременно работать с разными таблицами или с несколькими фрагментами одной и той же таблицы. Имеется много операций с таблицами и их фрагментами: удаление, вставка, перемещение, сортировка по заданным критериям. Можно выделять фрагменты таблицы произвольного размера и выполнять операции с этими фрагментами, например находить сумму, среднее значение, максимум и минимум по фрагментам. Excel может работать с числами, денежными величинами в разных валютах и процентами.

Электронные таблицы Excel применимы для ведения простейших баз данных реляционного типа, хотя открывающиеся здесь возможности существенно меньше, чем в программе Access. Программа Excel позволяет также решать некоторые (относительно несложные) задачи поиска оптимальных решений различных проблем при наличии дополнительных условий, например задачу максимизации прибыли от продажи товаров при ограничениях на продажную цену и оплату труда.

Результаты вычислений могут представляться в виде таблиц, а также в виде черно-белых или цветных графиков, стандартных или нестандартных диаграмм разнообразной формы (круговых, линейчатых, столбчатых, кольцевых, лепестковых, точечных) и масштаба. Предусмотрены средства контроля вводимой информации по определенным критериям и средства защиты информации. Имеются многочисленные средства улучшения внешнего вида таблиц. Программы, выполняющие столь разнообразные действия с таблицами, иногда называются табличными процессорами.

Богатые возможности Excel позволяют активно использовать эту программу в бухгалтериях, магазинах, биржах, складах, банках, брокерских конторах и многих других организациях для планирования, анализа и прогноза деятельности.

10.3. Автоматическое распознавание текстов

После ввода документа с помощью сканера в памяти компьютера создается *графический образ* документа, представляющий собой, как уже отмечалось, всего лишь набор разноцветных точек. Поэтому требуется распознавание текста. Современные программы распознавания достаточно уверенно распознают даже весьма вычурные тексты на одном или нескольких языках, иногда даже рукописные. Из программ, способных распознавать тексты на русском языке, наиболее известна программа Fine Reader отечественной разработки. Она может распознавать тексты на русском, английском, немецком, французском, украинском и многих других языках, а также смешанные двуязычные тексты; работать с разными моделями сканеров и с пакетами документов или многостраничными документами; объединять сканирование и распознавание в единую операцию; способна к обучению для распознавания сложных или неразборчивых шрифтов. Программа Fine Reader также распознает бланки, т.е. отформатированные документы, в поля которых внесены данные (например, анкеты, бюллетени для голосования, опросные листы). Бланки состоят из постоянной части, в которой содержится информация, используемая при заполнении бланка, и переменной части, в которую в ходе заполнения заносятся данные. Часто приходится последовательно обрабатывать многие тысячи однотипных бланков (результаты голосования или опроса).

В программе используется оконная технология, причем на экране отображается строка меню, ряд панелей инструментов и рабочая область документа. Преобразование исходного бумажного документа осуществляется в три этапа:

1) сканирование, в результате которого изображение преобразуется в цифровую форму;

2) требуемая сегментация текста, так как исходный текст может иметь сложный формат, например располагаться в несколько колонок,

перемежаться рисунками и подписями к ним; поэтому для распознавания необходимо выделить сегменты (блоки) с цельными фрагментами однородного текста; сегментация может выполняться автоматически или полуавтоматически (с «подсказкой» пользователя);

3) собственно распознавание, которое обычно выполняется автоматически, если не требуется обучение.

Распознаваемый текст отображается в отдельном окне на экране монитора и может редактироваться и форматироваться независимо от исходного текста. Цветом будут выделены те символы, которые программа считает неоднозначно опознанными. Полученный в результате распознавания текст можно сохранить в виде форматированного или неформатированного документа.

10.4. Автоматизированный перевод документов

Средства автоматизированного перевода можно разделить на две группы: электронные словари и программы перевода. *Электронные словари* служат для перевода отдельных слов и позволяют немедленно получить требуемое слово на одном или нескольких других языках. В настоящее время электронные словари выпускаются в самом различном исполнении и могут устанавливаться на стационарных и переносных компьютерах, вплоть до электронных блокнотов карманного формата. Электронные словари в переносных компьютерах обладают довольно значительным словарным запасом и дают возможность прямого и обратного перевода на ряд языков, например с русского языка на английский, немецкий, французский языки, и обратно. Карманные электронные словари-блокноты удобны для туристов и лиц, профессионально общающихся с иностранцами: гидов-переводчиков, работников фирм, персонала гостиниц и т.п. Электронные словари, установленные на стационарных компьютерах, могут разделяться по областям знаний (технические, физические, химические, математические, военные словари, словари по вычислительной технике и др.), обычно дают наборы значений и примеры употребления отдельных слов и используются переводчиками, которые большую часть перевода выполняют самостоятельно.

Программы перевода предназначены для перевода текста полностью, хотя получение перевода высокого качества, в особенности

художественной литературы, пока находится за пределами возможностей современных компьютеров. Однако перевод деловых, научных, технических документов может быть выполнен вполне удовлетворительно и требовать лишь относительно небольшой доработки. Качество перевода зависит от полноты используемых словарей и грамматических правил. Распространенной программой перевода является Promt, имеющая несколько версий и предназначенная для перевода с английского, немецкого и французского языков на русский, и обратно.

Программой Promt используются при переводе словари трех типов: генеральный словарь, специализированные словари и пользовательский словарь. *Генеральный словарь* содержит основную лексику соответствующего языка и всегда используется при переводе. *Специализированные словари* разделяются по областям знаний и подключаются к переводу в соответствии с тематикой документа. *Пользовательский словарь* составляется и пополняется пользователем, содержит слова и словосочетания, отсутствующие в других словарях, а также дает более точный перевод специальных терминов и речевых оборотов по профилю документа. При переводе поиск слов всегда начинается с пользовательского словаря, продолжается в специализированных словарях и заканчивается генеральным словарем. Слова, специально выделенные пользователем в тексте, могут переводиться побуквенно, например имена собственные. Некоторые указанные пользователем (зарезервированные) слова (например, Windows, HTML, служебные слова языков программирования типа begin, end, do и др.) или целые фрагменты текста (например, программы на языках программирования) сохраняются без перевода. Может использоваться общий пополняемый список всех зарезервированных слов.

В исходном тексте возможны слова или фрагменты, не поддающиеся переводу из-за опечаток, плохого качества печати документа, неправильного распознавания при вводе в компьютер, наличия собственных имен или слов, отсутствующих в словарях. Поэтому при переводе как исходный текст, так и перевод нуждаются в редактировании. Для работы программа Promt, помимо меню и панелей инструментов, открывает на экране монитора три окна (рис. 10.3): окно исходного документа, окно перевода и окно, называемое информационной панелью для ввода информации о перево-

димом документе и специальных настройках, например указаний о побуквенном переводе или о сохранении слов и фрагментов текста без перевода.

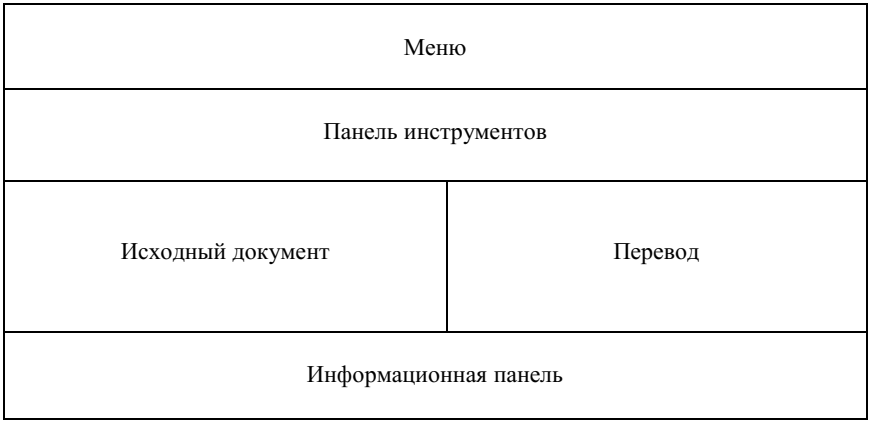


Рис. 10.3. Экран монитора в программе Promt

В программе Promt имеются также средства быстрого автоматического перевода без активного участия пользователя, дающие возможность оперативно выполнять приближенный перевод отдельного текста или группы текстов, позволяющий понять их смысл. Быстрый перевод последовательности файлов, организованной в виде очереди, осуществляется с помощью приложения File Translator, а быстрый перевод отдельного неформатированного текста – с помощью приложения Qtrans (от слов quick translator) программы Promt. Средства быстрого перевода особенно удобны для оперативного перевода документов, поступающих по сети Internet. Для быстрого перевода Web-страниц в сети Internet предназначено приложение WebView программы Promt, близкое по своим возможностям к программе Internet Explorer.

10.5. Подготовка презентаций

В офисной технологии *презентацией* называют набор цветных слайдов с текстовой и графической информацией на некоторую конкретную тему, предназначенных для демонстрации в реклам-

ных или ознакомительных целях определенной категории зрителей – потенциальных покупателей, клиентов или заказчиков, акционеров, туристов, читателей, работников средств массовой информации. Презентации в этом понимании могут также использоваться в учебном процессе, для демонстрации достижений и в докладах на научно-технических конференциях, семинарах, совещаниях. Набор подготовленных слайдов составляет слайд-фильм (диафильм), который может демонстрироваться на большом экране с помощью слайд-проектора (диапроектора), управляемого компьютером или докладчиком. Демонстрационные материалы могут готовиться и на обычной кинопленке. По аналогии с кинофильмами слайды часто называют кадрами. Использование современных средств мультимедиа позволяет также готовить презентации с применением более совершенных звуковых и изобразительных средств, включая фрагменты кинофильмов или мультфильмов.

В пакете MS Office имеется программа PowerPoint для подготовки и редактирования произвольных презентаций. Подготовленная этой программой презентация сохраняется в виде файла с расширением .ppt. Программа PowerPoint выполняет две группы операций: 1) создание и редактирование отдельных слайдов; 2) монтаж презентации (слайд-фильма) из готовых слайдов. PowerPoint – многооконная программа, т.е. на экране монитора можно одновременно открыть несколько окон, в каждом из которых редактируется один из файлов с расширением .ppt. Для каждого слайда сначала готовится приближенный макет, который затем дорабатывается (редактируется) в содержательном и изобразительном отношении. Элементы макетов делятся на две категории: тексты и объекты (графика, звук и т.п.).

В приложении к программе, называемом Microsoft Clip Art Gallery, имеется большая коллекция готовых изображений на различные темы, которые можно вставлять в создаваемые слайды. Можно также вставлять в слайды дополнительные тексты, фигуры, собственные рисунки, диаграммы из программы Excel и др.; вызывать программу проверки орфографии; применять средства мультимедиа и анимации при подготовке презентаций для получения движущихся изображений; пользоваться готовыми шаблонами презентаций (слайд-фильмов) с рекомендуемым дизайном, которые хранятся в файлах с расширением .pot. Набор этих шаблонов мож-

но пополнять. Подготовленная презентация записывается на дискету или оформляется в виде набора диапозитивов.

Упрощенной версией программы PowerPoint является программа MS PowerPoint Viewer, которая дает возможность лишь демонстрировать уже готовые слайд-фильмы. Этой программой удобно пользоваться, например, если надо организовать показ слайд-фильма с другого компьютера, на котором отсутствует программа PowerPoint. При демонстрации смонтированных слайд-фильмов наибольшие возможности открываются, если компьютер снабжен средствами воспроизведения изображений с экрана монитора на большом экране. Используя такую технику, можно демонстрировать кадры «с наплывами» снизу, сверху, слева, справа; переставлять или пропускать отдельные кадры; удерживать кадр на экране заранее установленное время; изменять последовательность кадров; показывать в обратном порядке; автоматически повторять слайд-фильм; выполнять другие манипуляции. На время демонстрации можно превратить указатель мыши в специальный «карандаш», писать и рисовать им на экране, т.е. на демонстрируемом слайде (слайд в это время покоится); сделанные надписи и рисунки по окончании демонстрации на слайде не сохраняются. Каждый кадр, демонстрируемый на экране, можно сопровождать пустой страницей, например для комментария к демонстрации (если требуется – с использованием «карандаша»). Часть указанных возможностей сохраняется и при демонстрации презентации, оформленной в виде набора диапозитивов, с помощью подключенного к компьютеру диапроектора.

10.6. Электронные календари и скоросшиватели

В офисной работе всегда используются те или иные настольные календари, еженедельники, блокноты для заметок, в которых указываются дела, встречи, мероприятия, запланированные на дни недели, месяца, года. Этим целям может служить электронный календарь-еженедельник, реализованный в виде программы Outlook («перспектива») в пакете MS Office. Эта программа позволяет составлять и оперативно вести как личные, так и коллективные электронные календари, в которых детально распланированы дни и часы всех мероприятий на многие годы (до 4500 года!). Листы электронного кален-

даря можно перелистывать как в обычном календаре или сразу обращаться к конкретной дате. На экране монитора можно показывать различные виды календаря: на неделю, месяц, год.

Программа Outlook максимально облегчает составление деловых расписаний. При назначении времени планируемого мероприятия программа предупреждает, если это время уже занято, не допуская таким образом противоречивых или некорректных сроков. При планировании деятельности принимаются также во внимание праздничные дни, под которыми понимаются как общегосударственные праздники многих стран, так и коллективные (юбилеи), личные (дни рождения), семейные (годовщины различных событий). Запланированные мероприятия легко переносить или отменять. Программа способна работать как фоновая, когда компьютер используется для других целей. В таком режиме программа может за некоторое заранее заданное время напоминать о времени, тематике и месте проведения намеченного мероприятия.

Программа применима для планирования регламента и расписания совещаний со многими участниками; предупреждает о регулярных встречах по расписанию; облегчает ведение переписки, в том числе с помощью электронной почты; обеспечивает получение сообщений и своевременное отправление ответов на них, рассылку поздравлений указанным адресатам к соответствующим датам и выполнение многих других функций, типичных для секретариата. С помощью программы можно создать, вести учет и планирование текущих задач в деловой деятельности, разделив их по различным категориям – «папкам», например, выделив папки «Контакты», «Поздравления», «Поставки», «Подарки», «Заметки», «Учебные занятия», «Техника», «Транспорт», «Финансы», «Повышение квалификации» и т.п.; отображать на экране задачи текущего дня, недели, месяца, года; удалять или переносить в архив выполненные задачи. При использовании программы несколькими пользователями программа позволяет обеспечить защиту информации от несанкционированного доступа, в частности информация о личных встречах недоступна всем другим пользователям. Для улучшения наглядности информацию на экране можно сортировать и группировать.

Особое место в электронном календаре занимают «события», под которыми понимаются мероприятия, продолжающиеся не менее суток. Среди них выделяются регулярные, например ежегод-

ные, и нерегулярные (сессии, командировки) события. Программа наблюдает за тем, чтобы запланированные мероприятия были отложены или отменены в соответствии с событиями.

Некоторое неудобство электронного календаря – меньшая портативность по сравнению с обычными еженедельниками. Для частичной компенсации этого недостатка предусмотрена возможность печати страниц электронного календаря на листах заданного формата, которые можно сброшюровать и всегда иметь с собой.

В канцелярской работе, как известно, часто используются скоросшиватели различных типов или подшивки документов, позволяющие собрать вместе различные документы по определенной тематике. В выполнении этой задачи может быть полезной программа Binder, имеющаяся в пакете MS Office. Программа позволяет составлять и редактировать подшивки различных документов – файлов, называемых разделами, часть которых уже может быть создана программами Word, Excel, PowerPoint; работать с любым разделом непосредственно в окне подшивки, оформить все разделы в едином стиле, пронумеровать страницы подшивки, составить ее оглавление; некоторые разделы сделать скрытыми; отдельные разделы или всю подшивку вывести на печать и т.п.

Следует заметить, что использование программы Binder возможно лишь в персональных компьютерах класса не ниже Pentium 100 с емкостью оперативной памяти не менее 32 Мбайт.

10.7. Технология офисной работы в сети Internet

В деятельности современного офиса обычно широко используются возможности, открываемые глобальной сетью Internet. Основное назначение Internet – предоставление доступа к коллективным ресурсам и обеспечение обмена информацией между пользователями (абонентами) сети. *Ресурсами* Internet считаются серверы, страницы, сайты, файлы и каталоги. *Серверы* или, точнее, *сетевые серверы* – это узлы сети в виде мощных компьютеров, обеспечивающих передачу информации в сети с использованием хранящихся в этих компьютерах специальных программ, которые часто также называют серверами или *программами-серверами.* Владельцы серверов (*провайдеры*) за определенную плату обеспечивают абонентов услугами сети. Пространство «Всемирной паутины» WWW

составляют отдельные документы – *Web-страницы.* Используется также термин «страница». *Страница* – это особый объект, выполненный в коде HTML (сокращение от слов Hyper Text Markup Language), размещенный на Web-странице и передаваемый абоненту вместе с ней. Совокупность тематически объединенных Web-страниц, принадлежащих некоторому абоненту и размещенных на каком-либо сервере, составляет *сайт* (site – участок). Для адресации ресурсов используется *унифицированный указатель на ресурс*, сокращенно URL (Unified Resource Locator).

Важнейшими видами офисной работы в сети Internet являются:

целенаправленное движение по сети (*навигация*);

поиск информации в сети по различным запросам и ее просмотр;

использование ресурсов сети;

сохранение найденной информации путем печати или копирования на сетевой компьютер;

размещение в сети собственной информации;

использование средств электронной почты.

Для выполнения этой работы предназначена однооконная программа Internet Explorer, включенная в пакет MS Office и представляющая собой широко распространенный браузер. Первоначально функции *браузеров* (браузер – обозреватель, навигатор), ограничивались просмотром электронных документов в сети Internet, выполненных в формате HTML. Современные браузеры, к которым относится и Internet Explorer, способны воспроизводить не только текст, но также музыку и человеческую речь, обеспечивают прослушивание радиопередач в сети Internet, обслуживание электронной почты и телеконференций, в том числе и видеоконференций. В терминологии Internet *телеконференция* или *служба новостей* – это средство распространения тематической информации, не предназначенной для конкретного адресата, хранящейся на специально выделенных «серверах новостей» и доступной всем пользователям этих серверов. Пользователи могут отправлять в телеконференции свои отклики или новые сообщения, которые также должны иметь общий характер, поскольку они общедоступны.

Технология работы с программой Internet Explorer близка к технологии работы в ОС Windows. Навигация в сети Internet состоит в переходе к той или иной странице. Примерами страниц Internet могут быть Web-страницы «паутины» WWW и каталоги файлов FTP.

В программе Internet Explorer предусмотрены три способа перехода к другой странице:

1) путем явного указания адреса страницы;

2) по ссылке;

3) по гипертекстовой ссылке, сокращенно – гиперссылке.

Явное указание адреса некоторых типов файлов можно выполнить подобно указанию адреса в ОС Windows или выбором адреса из раскрывающегося списка адресов, а также с помощью протокола HTTP (Hyper Text Transfer Protocole), причем имя этого протокола можно и не указывать, оно принимается по умолчанию. Переходы по ссылкам организуются с использованием одного из трех списков:

списка особых, специальных ссылок;

списка адресов просмотренных ранее страниц;

списка избранных ссылок, составляемых пользователем.

Наконец, переходы по гиперссылкам, которые выделяются в текстах документов цветом, осуществляются с помощью курсора и мыши, как, например, в FAR Maneger.

Во «Всемирной паутине» содержатся многие миллионы самых различных документов с текстовой, графической, аудио- и видеоинформацией. Поиск нужных документов в сети может оказаться весьма трудоемким и дорогостоящим. Программа Internet Explorer сама не занимается поиском, она лишь принимает от пользователя соответствующим образом оформленный запрос, обрабатывает его и передает одной из информационно-поисковых систем, среди которых отметим системы Alta Vista, Infoseek, Lycos, Excite, «Рэмблер». Основой запроса являются ключевые слова (под которыми понимаются слова или словосочетания, отражающие содержание документа) и индексы, например индексы УДК – универсальной десятичной классификации. Эффективность поиска в сети в решающей степени определяется умением квалифицированно составить поисковый запрос, что в свою очередь зависит от опыта и эрудиции пользователя.

Собственная информация размещается в сети Internet как гипертекст, оформленный в виде сайта. *Гипертекстом* в Internet называют документ с взаимосвязанной информацией, который, помимо обычной текстовой и графической информации, содержит гиперссылки на другие гипертекстовые документы, причем эти гиперссылки встроены в текстовую или графическую информацию данного доку-

мента средствами языка HTML. Поскольку гиперссылки возможны не только на обычный текст, но также на графическую, звуковую или видеоинформацию, вместо термина «гипертекст» иногда в расширенном понимании используется термин «гипермедиа».

Все шире сейчас в офисах используется электронная почта, применяемая для отправления и получения сообщений (документов) с помощью сети Internet. Для пользования почтой абонент сети должен зарегистрироваться у провайдера и получить у него электронный адрес. После этого на сервере провайдера создается почтовый ящик абонента для накопления входящих и исходящих сообщений. Основные операции с электронной почтой – создание сообщений, доставка почты и чтение входящих (полученных) сообщений. Почтовая программа абонента создает для работы четыре папки с входящими, исходящими, отправленными и удаленными сообщениями соответственно.

Каждые сообщения электронной почты содержат заголовок, текст письма и, возможно, вложенные (присоединенные) файлы. В заголовке помимо служебной информации имеются поля для указания адреса отправителя (From:), даты (Date:), адреса получателя (To:) и темы сообщения (Subject:), причем отправитель указывает лишь адрес получателя и тему сообщения, а остальные поля заполняются системой автоматически. Часто используемые адреса можно поместить в «адресную книгу», а затем выбирать их из этой книги. Текст письма набирается с клавиатуры, как обычный текстовый файл в ОС Windows.

Одно из важнейших достоинств электронной почты – возможность переслать с письмом в качестве вложения (Attachment) произвольный файл. Вложенные файлы объемом свыше 20 – 30 Кбайт целесообразно предварительно упаковать (архивировать) для снижения времени и стоимости пересылки. Файлы слишком большого объема (свыше 2 – 3 Мбайт) пересылать не рекомендуется. При чтении письма, доставленного электронной почтой, иногда может потребоваться его перекодировка из одного кода в другой, например из ASCII в ДКОИ-8. Как текст полученного письма, так и вложенные в него файлы можно сохранить в компьютере получателя и напечатать.

Помимо браузера Internet Explorer следует отметить также более ранний браузер Netscape Navigator с близкими возможностями, а также относительно простой браузер Lynx.

10.8. Экспертные системы

Экспертные системы составляют одно из наиболее перспективных направлений в области искусственного интеллекта. Цель этого направления – разработка методов и программ, позволяющих получать результаты, не уступающие по качеству и эффективности результатам, которые смог бы получить эксперт-человек при решении особенно трудных для экспертизы, так называемых неформализованных задач. *Неформализованными* считаются задачи, обладающие хотя бы одной из следующих характеристик:

невозможно задание в числовой форме;

нет четко определенной целевой функции;

задача алгоритмически неразрешима или решение невозможно из-за ограниченности ресурсов компьютера.

Типичные свойства неформализованных задач – ошибочные, неоднозначные, неполные, противоречивые исходные данные и знания о проблемной области; большое число параметров, т.е. большая размерность пространства параметров, в котором отыскивается точка оптимума; динамическое изменение данных и знаний в ходе решения задачи.

Экспертные системы применимы для интерпретации, предсказания, диагностики, планирования, конструирования, контроля, отладки, инструктажа, управления в таких областях, как финансы, медицина, космос, связь, образование, энергетика, некоторые виды промышленности. Решение задач, всегда имеющих практически важное значение, в экспертных системах обычно находится эвристически с использованием символьных вычислений. Как метод получения решения, так и само решение должны быть понятны пользователю и могут быть объяснены экспертной системой.

Основная операция, применяемая при выработке новых знаний в экспертных системах, называется продукцией. *Продукция* состоит в исключении в начале заданного слова некоторого числа символов и добавлении в конце оставшегося слова некоторого числа других символов. Например, исключая в слове АСТ символы (буквы) АС и добавляя символы АВ, получаем слово ТАВ. Любой текст можно рассматривать как слово (хотя и достаточно длинное), поэтому располагая продукционной системой, можно строить новые тексты. *Продукционная система* PS в общем случае представляется в виде

$$PS = <R, B, I>.$$

Здесь R – *рабочая память*, которая в экспертных системах понимается как база исходных и текущих данных; B – *база знаний* для хранения долгосрочных данных, описывающих рассматриваемую область, и правил преобразования данных, т.е. множество допустимых продукций; I – *интерпретатор* («решатель»), реализующий циклический процесс вывода решений.

Базу B можно пополнять. Интерпретатор I можно формально представить в виде

$$I = <V, S, K, W>,$$

где V – процесс выбора из B и R для очередного цикла работы интерпретатора; S – процесс сопоставления правил вывода и данных; K – процесс разрешения конфликтов, т.е. процесс планирования – нахождения продукции, которая будет применяться в следующем цикле; W – процесс выполнения выбранных правил.

Мощность экспертной системы определяется, главным образом, мощностью базы знаний, которые обычно являются эвристическими, экспериментальными, правдоподобными. Эксперт ведет диалог с системой на удобном языке и может приобретать новые знания, а система может объяснить свои рассуждения.

Различают статические и динамические экспертные системы. *Статические экспертные системы* предназначены для решения задач, в которых можно не учитывать изменение окружающего мира за время решения. Статическая система (рис. 10.4) помимо базы знаний, рабочей памяти и интерпретатора имеет *компонент приобретения знаний*, через который пользователь наполняет систему знаниями; *объяснительный компонент*, поясняющий пользователю, как было получено или почему не получено решение, какие знания использовались; *диалоговый компонент*, организующий дружественный интерфейс с пользователем. В динамической системе (рис. 10.5) дополнительно имеются устройства сопряжения с внешним миром и подсистема его моделирования.

Экспертные системы работают в одном из двух режимах. Вначале *в режиме приобретения знаний* эксперт наполняет систему знаниями о проблемной области, которые включают исходные данные и правила преобразования данных (допустимые продукции). Затем в *режиме консультации* осуществляется собственно

решение задачи, причем система объясняет пользователю как получен результат.

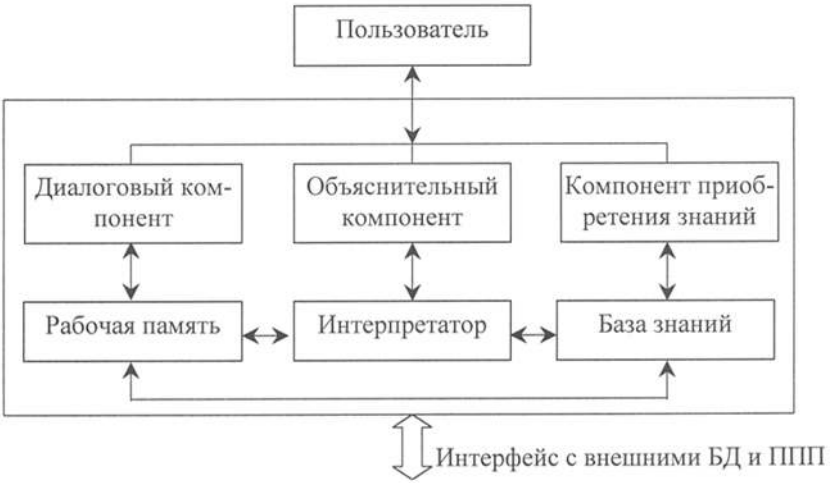


Рис. 10.4. Архитектура статической экспертной системы

Существуют *изолированные экспертные системы*, не взаимодействующие с другими программными средствами (базами данных, пакетами прикладных программ, электронными таблицами), и *интегрированные*, в которых такие взаимодействия имеются. В *централизованных экспертных системах* аппаратные средства состоят из единственного компьютера и терминалов, а в *распределенных* системах используются ресурсы нескольких (иногда многих) компьютеров. *Мобильные экспертные системы* могут устанавливаться на компьютерах разных типов, а *немобильные* – на компьютерах одного типа. По затратам ресурсов экспертные системы делятся на *малые* (могут функционировать на персональных компьютерах, используются обычно для обучения или исследования возможностей системы), *средние* (устанавливаются на рабочих станциях и охватывают все приложения систем) и *большие* (на рабочих станциях и больших компьютерах, обычно имеют доступ к огромным базам данных). Особо можно выделить *символьные экспертные системы*, которые предназначены для исследовательских работ и устанавливаются на компьютерах с символьными вычисле-

ниями и языками типа Пролог, Лисп и т.п. По приложениям выделяют *проблемно-ориентированные экспертные системы*, ориентированные на некоторые классы задач (управление, планирование, прогнозирование и т.п.), и *предметно-ориентированные* – для различных предметных областей (банки, биржи, поиск неисправностей в разных технических средствах и др.).

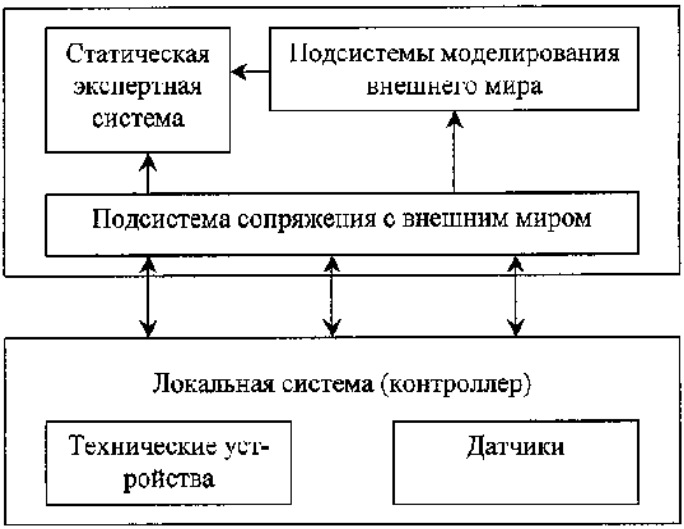


Рис. 10.5. Архитектура динамической экспертной системы

Динамические экспертные системы *реального времени* – это особый тип систем, имеющих эффективную связь с внешними системами и базами данных и обычно применяемых для управления какими-либо процессами. Архитектура экспертной системы реального времени показана на рис. 10.6.

Входящая в систему *машина вывода* выполняет все рассуждения с использованием средств ускорения рассуждений и осуществляет реакции на непредвиденные обстоятельства – аварии, истощение ресурсов, сбои аппаратуры.

В настоящее время в продаже уже имеется довольно много экспертных систем различных фирм, в том числе для персональных компьютеров. Выпускаются статические, динамические, символьные экспертные системы и системы реального времени. Можно

отметить, например, интегрированную экспертную систему реального времени G2 фирмы Gensym Corp. (USA), систему TDC Expert фирмы Honeywell (USA), систему Promass фирмы Unibit (UK), систему ILOG Rules фирмы ILOG (France). Стоимость выпускаемых экспертных систем пока довольно велика. Работы в области создания экспертных систем активно продолжаются.

Рис. 10.6. Архитектура экспертной системы реального времени

10.9. OLE-технология подготовки составных документов

Документы, в которых объединяются объекты разного происхождения и типа (например, текст, изображение, звук), принято называть *составными.* Для создания и обработки составных документов целесообразно использовать универсальную технологию, называемую *OLE-технологией* (аббревиатура от слов Object Linking and Embedding – связывание и внедрение объектов) и позволяющую комбинировать при создании документов средства разных приложений [55, 56].

Объект OLE (OLE-объект) – произвольный элемент, созданный средствами какого-либо приложения, который можно связать с документом другого приложения или внедрить (поместить) в другое приложение. Под *внедрением* объектов понимается создание комплексного (составного) документа, содержащего два или более автономных объектов. Обычный способ внедрения – это импорт объекта из уже готового файла, в котором хранится этот объект, например импорт рисунка из документа графического редактора, звукозаписи или видеозаписи в документ, создаваемый текстовым редактором. При этом не только получается составной документ, но и

появляется возможность редактировать внедренный объект средствами «породившего» его приложения, не меняя оригинал. Внедрить можно как целиком некоторый документ, так и его фрагмент (это достоинство метода внедрения), однако объем составного документа возрастает на величину объема внедренных объектов.

Поместить объект в какой-либо документ можно и другим путем – с помощью связывания. При *связывании* сам объект остается на своем месте, а в документ вставляется лишь ссылка – указатель на местоположение этого объекта. Когда при просмотре составного документа пользователь дойдет до вставленного указателя, произойдет обращение по адресу объекта и объект отобразится в документе. При связывании объектов объем составного документа практически не увеличивается, так как ссылки занимают мало места в памяти. Более того одни и те же объекты можно связывать со многими составными документами, получая огромную экономию памяти компьютера по сравнению с методом внедрения. Но поскольку адресуются только документы в целом, а не их фрагменты, связанные документы будут передаваться в составной документ полностью. Кроме того, при использовании метода связывания необходимо строго следить за тем, чтобы все связанные объекты постоянно хранились на своих местах (в своих папках), так как при их перемещении вставленные ссылки окажутся неверными. Также при передаче кому-либо (например, «заказчику») составного документа связанные с ним объекты могут остаться на «родном» компьютере, и переданный документ окажется испорченным. Такие явления называют *разрывом связей* и надежные меры по сохранению связей – необходимое условие применения метода связывания.

Метод связывания удобно применять, в частности, в офисной технологии для вставки в составные документы различных бланков или реквизитов, которые во многих организациях используются огромным числом сотрудников различных подразделений. Тогда при изменении формы какого-либо бланка или реквизитов достаточно изменить один единственный объект-оригинал и во все документы, с которыми связан этот объект, будут автоматически передаваться бланк новой формы или новые реквизиты.

Роль OLE-технологии особенно велика в информационных системах. Так, досье с текстовыми биографическими данными о криминальных элементах можно дополнить с помощью OLE-технологии их цветными фотографиями, отпечатками пальцев, фонограм-

мами голосов, видеозаписями с их «участием»; рекламную афишу можно украсить фотографиями артистов, кадрами из спектаклей или видеоклипами и т.п.

Приложение, средствами которого создается OLE-объект, называется *OLE-сервером*, а приложение, принимающее OLE-объект, – *OLE-клиен*том. Например, при вставке рисунка в текстовый документ графический редактор является OLE-сервером, а текстовый редактор – OLE-клиентом. Некоторые программы могут быть только OLE-серверами (например, графический редактор Paint), другие – только OLE-клиентами (например, информационные системы), тогда как наиболее мощные и универсальные приложения могут выполнять функции как OLE-серверов, так и OLE-клиентов (например, Word, Excel). Средства использования OLE-технологии имеются в новых версиях ОС Windows и работающих в этой системе приложениях. В приложениях MS Office вместо внедрения некоторого объекта можно также вставить обозначающий его значок-пиктограмму, который реализует обращение.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Перечислите основные виды офисных работ с использованием компьютера.
2. Опишите структуру электронной таблицы (ЭТ), укажите области возможного ее применения.
3. Укажите, какие виды обработки данных возможны в ЭТ и какие виды графического представления данных включаются в состав большинства ЭТ.
4. Опишите основные этапы распознавания графической и текстовой информации.
5. Что понимается под презентацией офисной технологии? Для каких целей может быть использован подготовленный таким способом документ? Какая программа Microsoft Office реализует эту возможность?
6. Что понимается под экспертной системой, на какие задачи она ориентирована?
7. Что такое OLE-технологии, для каких целей она может быть полезна. Приведите пример применения OLE-технологии в информационных системах.

ГЛАВА 11. ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

11.1. Проблемы информационной безопасности

Проблема информационной безопасности возникла достаточно давно и имеет глубокие исторические корни. До сравнительно недавнего времени методы защиты информации были в исключительной компетенции спецслужб, обеспечивающих безопасность страны. Однако новые технологии измерения, передачи, обработки и хранения информации значительно расширили сферы деятельности людей, нуждающихся в защите информации, привели к развитию и распространению новых методов несанкционированного доступа к информации и, как следствие, к интенсивному развитию нового научного направления – «*информационная безопасность*». Все это связано, прежде всего, с появлением систем обработки данных на базе компьютеров, а также с бурным развитием систем передачи данных.

Можно выделить некоторые причины, которые и привели к необходимости как разработки новых методов защиты информации, так и к дальнейшему развитию традиционных.

Первые системы коллективного пользования ЭВМ, а затем объединение их в глобальные и локальные сети, технологии открытых систем уже на первом этапе выявили потребность в защите информации от случайных ошибок операторов, сбоев в аппаратуре, электропитании и т.п.

Стремительный рост емкости внешних запоминающих устройств и высокая эффективность их использования в системах автоматизированного управления привели к созданию банков (баз) данных колоссальной емкости и высокой стоимости, одновременно создавая проблемы их защиты, как от разнообразных случайностей, так и от несанкционированного доступа.

Современные информационные системы составляют техническую основу органов управления государственной власти, промышленных предприятий и научно-исследовательских организаций, учреждений кредитно-финансовой сферы, банков и т.п.

Сегодня, когда компьютер прочно вошел в наш быт, мы все чаще вынуждены доверять ему свои секреты (финансовые, промыш-

ленные, медицинские и др.), и в связи с этим вопросы защиты информации приобретают всеобъемлющий характер.

Кроме чисто технических задач разработки средств защиты информации имеются нормативно-технические, организационно-правовые, юридические и другие аспекты. Основные задачи, рассматриваемые специалистами по информационной безопасности (а также публикации на эту тему), связаны с обеспечением безопасности использования глобальных и локальных сетей, с проблемами глобальной вычислительной сети Интернет, «хакерами», «вирусами» и т.п.

Резюмируя сказанное, можно выделить технические, организационные и правовые меры обеспечения информационной безопасности и предотвращения компьютерных преступлений.

К техническим мерам относятся:

защита от несанкционированного доступа;

резервирование особо важных компонентов подсистем;

организация вычислительных сетей с перераспределением ресурсов при временном нарушении работоспособности какой-либо части сети;

создание устройств обнаружения и тушения пожаров;

создание устройств обнаружения утечек воды;

техническая защита от хищений, саботажа, диверсий, взрывов;

дублирование электропитания;

надежные запирающие устройства;

устройства сигнализации о различных опасностях.

К организационным мерам относятся:

надежная охрана;

подбор надежного персонала;

правильная организация работы персонала;

предусмотренный план восстановления работы информационного центра после сбоя;

организация обслуживания и контроля работы компьютерного центра лицами, не заинтересованными в сокрытии преступлений;

создание средств защиты информации от любых лиц, включая и руководящий персонал;

предусмотренные меры административной и уголовной ответственности за нарушение правил работы;

правильный выбор местонахождения информационного центра с дорогостоящим техническим и программным обеспечением.

К правовым мерам относятся:

разработка уголовных норм ответственности за компьютерные преступления;

защита авторских прав программистов;

усовершенствование уголовного и гражданского законодательства;

усовершенствование судопроизводства по компьютерным преступлениям;

общественный контроль за разработчиками компьютерных систем;

принятие ряда международных соглашений, касающихся информационной безопасности.

Это очень крупная научно-техническая проблема и, естественно, мы не можем осветить ее в полной мере и коснемся только основных понятий, определений и средств защиты, доступных пользователю, причем начнем с наиболее близкой проблемы рядового пользователя – с сохранения информации, не связанного с несанкционированным доступом.

11.2. Защита информации от непреднамеренных воздействий

Рассмотрим некоторые приемы и методы, связанные с защитой информации от случайных ошибок или некомпетентности пользователей, а также от сбоев аппаратуры, в частности из-за помех в электросети, т.е. причин возможной потери информации, не связанных с несанкционированным доступом и происками злоумышленников. Потеря файлов, а также крах системы вполне возможны и без внешних, корыстных помыслов. В связи с этим во всех операционных системах предусматриваются простейшие средства профилактики. Так, при удалении файлов, как правило, требуется дополнительное подтверждение, а удаленный файл, как правило, при необходимости может быть восстановлен, поскольку определенное время он хранится в специальном буфере («корзина для мусора»).

Для того чтобы обезопасить себя от неприятных последствий (связанных с вышеперечисленными инцидентами), приводящих к потере данных на сервере или рабочих станциях, которые могут представлять большую ценность, так как являются результатом больших трудовых затрат, необходимо выполнение определенных мероприятий. Существует три основных способа защиты от таких воздействий – резервное копирование данных, избыточное дублирование и установка специализированных устройств защиты от нарушений в системе электропитания.

Резервное копирование данных. Методы, используемые для резервного копирования, зависят от их объема, важности информации и динамики ее изменения. Если говорить о носителях, применимых для хранения резервных копий, то дискеты годятся лишь в частных случаях для небольших объемов информации и личных архивов пользователей. В большинстве случаев используются либо накопители на магнитной ленте (стримеры), либо магнитно-оптические устройства, либо оптические типа WORM или WARM. Независимо от типа устройства для резервного копирования необходимо систематически проводить копирование данных во избежание их потери. Выбор конкретного способа зависит от того, как часто изменяются данные, какую ценность они представляют и как много времени потребуется для этой процедуры. Существуют следующие способы резервного копирования.

Случайный. При таком подходе производится случайное копирование отдельных файлов. Метод является наименее надежным, так как если обнаружится, что копия не самая новая, приходится проделать весь объем работы от момента изготовления этой резервной копии. Еще хуже, если носитель, на котором находится резервная копия, окажется поврежденным. Однако это лучше, чем ничего.

Серьезный. Резервные копии производятся регулярно и для их изготовления используются два набора носителей.

Профессиональный. Этот метод используют вычислительные центры с дорогостоящим оборудованием и большими компьютерами. В нем используются три копии данных на трех наборах носителей (для надежности иногда используются по два экземпляра для каждого из наборов). При работе поочередно используется каждый

из наборов. Этот метод иногда называют схемой «сын – отец – дед».

Избыточность данных. Резервирование также подразумевает избыточность данных. С точки зрения подлинности лучше иметь два средних размера файловых сервера в локальной сети, чем один большой. Тогда в случае выхода из строя одного из них можно временно продолжать работать с другим. Конечно же, при этом на втором сервере должны находиться резервные копии рабочих файлов.

Несмотря на то, что системы хранения данных, основанные на магнитных дисках, производятся уже 40 лет, массовое производство отказоустойчивых систем началось совсем недавно. Дисковые массивы с избыточностью данных, которые принято называть RAID (redundant arrays of inexpensive disks – избыточный массив недорогих дисков) были представлены исследователями (Петтерсон, Гибсон и Катц) из Калифорнийского университета в Беркли в 1987 г. Но широкое распространение RAID-системы получили только тогда, когда диски, которые подходят для использования в избыточных массивах, стали доступны и достаточно производительны. Со времени представления официального доклада о RAID в 1988 г. исследования в сфере избыточных дисковых массивов начали бурно развиваться, в попытке обеспечить широкий спектр решений на основе компромисса «цена – производительность – надежность».

С аббревиатурой RAID в свое время случился казус. Дело в том, что недорогими дисками во время написания статьи назывались все диски, которые использовались в персональных компьютерах, в противовес дорогим дискам для мейнфреймов. Но для использования в массивах RAID пришлось использовать достаточно дорогостоящую аппаратуру по сравнению с другой комплектацией персональных компьютеров, поэтому RAID начали расшифровывать как redundant array of independent disks – избыточный массив независимых дисков.

Производители файловых серверов, учитывая необходимость избыточности данных, предлагают модели с дисковыми массивами – системами НЖМД, в которых информация зеркально дублирована на различных дисководах. Естественно, что избыточность данных ни в коей мере не заменяет необходимость резервного копирования.

Защита от помех в электросети. Сбои электропитания всегда происходят неожиданно. В момент сбоя электросети практически любая программа может в какой-то степени испортить файл, с которым она работала. Для защиты от таких ситуаций необходимо использовать источники бесперебойного питания (UPS – Uninterruptable Power System) файл-серверов. Нет необходимости подключать к UPS рабочие станции, поскольку производители UPS нормируют их по максимальной мощности подключенных к ним приборов, так что не следует превышать эту величину.

После пропадания напряжения в электросети батареи UPS обеспечивают работоспособность сервера в последующие десять минут – время, достаточное для того, чтобы завершить работу и успеть сохранить рабочие файлы. Кроме того, UPS защищает файл-сервер от скачков напряжения в электросети.

Существуют и более дешевые системы дежурного питания (SPS – Standby Power System) вместе с фильтром напряжения сети, которые защищают оборудование от кратковременных исчезновений электроэнергии в электросети, выбросов и помех.

В современных критически важных серверах принято за правило устанавливать особые Redundant-блоки питания, которые предоставляют системному администратору возможность подключать сервер одновременно к двум источникам питания (например, к двум электрическим розеткам, подключенным к двум независимым электрическим группам, или одновременно к электрической розетке и к UPS). При падении напряжения на одном из подключений сервер продолжает работать на другом. Это намного повышает надежность питания сервера и позволяет производить плановую замену UPS без остановки сервера.

11.3. О некоторых понятиях компьютерной безопасности

Термины «информационная безопасность» (information security) и «безопасность сети» (network security) в широком смысле относятся к секретности, т.е. гарантии того, что информация и службы, имеющиеся в сети, не будут доступны для несанкционированного использования. Безопасность подразумевает механизм защиты, гарантирующий невозможность несанкционированного доступа к вычислительным ресурсам, шпионажа или перехвата сообщений, а

также работу служб. Конечно, нельзя гарантировать абсолютную безопасность сети, так же как нельзя гарантировать полную защищенность материальных ценностей.

Обеспечение информационной безопасности требует охраны как физических, так и виртуальных ресурсов. К физическим устройствам можно отнести такие пассивные устройства для хранения информации, как жесткие диски и компакт-диски, и такие активные устройства, как компьютеры пользователей. В сетевом окружении понятие физической безопасности относится к кабелям, мостам, маршрутизаторам и т.д. Хотя о физической безопасности упоминается очень редко, она часто играет важную роль при планировании полной безопасности, а меры по ее обеспечению достаточно традиционны и хорошо известны.

Обеспечение безопасности такого виртуального ресурса, как информация, обычно связывают с тремя основными понятиями компьютерной безопасности.

1. Угроза безопасности компьютерной системы – потенциально возможное происшествие, которое может оказать нежелательное воздействие на саму систему, а также на информацию, хранящуюся в ней.

Обычно выделяют три вида угроз.

Угроза раскрытия заключается в том, что информация становится известной нежелательным лицам. Иногда вместо слова «раскрытие» используют термины «кража» или «утечка».

Угроза целостности включает себя любое умышленное изменение (модификацию или даже удаление) данных, хранящихся в вычислительной системе или передаваемых из одной системы в другую. Обычно считается, что угрозе раскрытия подвержены в большей степени государственные структуры, а угрозе целостности – деловые или коммерческие.

Угроза отказов обслуживания возможна всякий раз, когда в результате определенных действий блокируется доступ к некоторому ресурсу вычислительной системы. Блокирование может быть постоянным (чтобы запрашиваемый ресурс никогда не был получен) или может вызвать только задержку, достаточно долгую для того, чтобы он стал бесполезным. В таких случаях говорится, что ресурс исчерпан. В локальных вычислительных системах наиболее частыми являются угрозы раскрытия и целостности информации, а в глобальных на первое место выходит угроза отказа от обслуживания.

2. Уязвимость компьютерной системы – некоторые ее неудачные характеристики, которые дают возможность возникновения угрозы. Именно из-за уязвимости в системе происходят нежелательные явления.

3. Атака на компьютерную систему – третье основополагающее понятие компьютерной безопасности. Это действие, предпринимаемое злоумышленником, которое заключается в поиске той или иной уязвимости. Таким образом, атака – реализация угрозы. К сетевым системам наряду с обычными (локальными) системами, осуществляемыми в пределах одной компьютерной системы, применим специфический вид атак, обусловленный распределенностью ресурсов и информации в пространстве – так называемые «сетевые (или удаленные) атаки». Они характеризуются, во-первых, тем, что злоумышленник находится за тысячи километров от атакуемого объекта, и, во-вторых, тем, что нападению может подвергаться не конкретный компьютер, а информация, передающаяся по сетевым соединениям. С развитием локальных и глобальных сетей именно удаленные атаки считаются лидирующими по количеству попыток и по успешности их применения, поэтому обеспечение безопасности с точки зрения противостояния сетевым атакам приобретает первостепенное значение.

Под *удаленной атакой* обычно понимается информационное воздействие на распределенную вычислительную систему, программно осуществляемое по каналам связи. Это определение охватывает как удаленные атаки на информационную инфраструктуру и протоколы сети, так и удаленные атаки на операционные системы и приложения. Под инфрастуктурой сети понимается сложившаяся система организации между объектами сети и используемые в сети сервисные службы. А под операционными системами и приложениями – все программное обеспечение, работающее на удаленном компьютере, которое тем или иным образом обеспечивает сетевое взаимодействие.

Хотя информационная безопасность охватывает множество способов защиты, но основными из них являются следующие [3].

Целостность данных. Безопасная система должна защитить информацию от несанкционированного изменения и повреждения.

*Доступность данных***.** Система должна гарантировать, что несанкционированный пользователь не может помешать заданному доступу к данным.

Секретность и конфиденциальность. Система не должна позволять несанкционированным пользователям создавать копии данных во время их передачи по сети, а также анализировать их содержимое в том случае, если копии все-таки сделаны.

Авторизация. Меры информационной безопасности должны быть избирательными, учитывающими классификацию людей и ресурсов по различным категориям.

Аутентификация. Система должна позволять двум взаимодействующим между собой объектам проверить подлинность друг друга.

Запрещение повторного использования. Чтобы посторонние не могли перехватывать копии с целью их дальнейшего использования, система не должна обрабатывать копии повторно переданных пакетов данных.

Говоря об основных понятиях информационной безопасности, необходимо определиться и с людьми, так или иначе связанными с проблемами компьютерного взлома, так называемыми хакерами. Общественное мнение специалистов по отношении деятельности хакеров не однозначно, оно либо сугубо негативное (хакеры – преступники), либо достаточно позитивное (хакеры – «санитары леса»). В действительности, эта деятельность имеет как положительную сторону, так и сугубо отрицательную, и эти две стороны четко разграничены. В связи с этим некоторые авторы предлагают разделить всех профессионалов, связанных с информационной безопасностью, на хакеров (hackers) и кракеров (crackers). И те, и другие во многом занимаются решением одних и тех же задач – поиском уязвимости в вычислительных системах и осуществлением на них атак («взломом»).

Кракеры – специалисты, способные снять защиту от копирования с лицензионного программного обеспечения. В современном компьютерном андеграунде кракерами обычно называют взломщиков ПО, в то время как *хакерами* именуют людей, специализирующихся на взломе защиты отдельных компьютеров и распределенных систем.

Принципиальное различие между хакерами и кракерами состоит в целях, которые они преследуют. Основная цель хакера состоит в том, чтобы, исследуя вычислительную систему, обнаруживать слабые места (уязвимость) в ее системе безопасности и информиро-

вать пользователя и разработчиков системы с целью устранения найденных уязвимостей. Другая задача хакера – проанализировать существующую систему, сформулировать необходимые требования и условия повышения уровня ее защищенности. Задача кракера состоит в непосредственном осуществлении взлома системы с целью получения несанкционированного доступа к чужой информации для кражи, подмены или объявления факта взлома. Среди основных *целей кракеров* следует отметить следующие:

получить доступ к важной информации, закрытой по тем или иным соображениям от использования посторонними лицами;

получить доступ к ресурсам чужой системы (процессора, внешней памяти и т.п.), в этом случае владелец системы ничего не теряет, за исключением времени занятости процессора и части дискового пространства, но, возможно, и приобретает достаточно дорогое программное обеспечение;

нарушить работоспособность хоста, без реализации угрозы раскрытия (это может быть достаточно опасным, если хост обеспечивает бесперебойное обслуживание клиентов);

создать плацдарм для осуществления вышеназванных целей, но для атаки на другой компьютер с целью переадресовать корыстные цели на чужой компьютер;

отладить механизм атак на другие системы, используя Ваш компьютер в качестве пробного.

Мотивы кракеров низменны, но их состав неоднороден. Существует даже их классификация, в соответствии с которой кракеров разделяют на следующие категории:

вандалы – самые известные (благодаря распространению вирусов) и самая малочисленная часть кракеров (их основная цель – взломать систему для ее дальнейшего разрушения, это специалисты в написании вирусов и их разновидностей под названием «троянских коней», весь компьютерный мир ненавидит вандалов лютой ненавистью, эта стадия «кракерства» характерна для новичков и быстро проходит, если кракер продолжает совершенствоваться);

«шутники» – наиболее безобидная часть кракеров (основная цель «шутников» – известность, достигаемая путем взлома компьютерных систем и внедрения туда различных эффектов, выражающих их неудовлетворенное чувств юмора; к «шутникам» также можно отнести создателей вирусов с различными визуально-

звуковыми эффектами («музыкодрожание» или переворачивание экрана и т.п.); «шутники», как правило, не наносят существенного ущерба компьютерным системам и администраторам; все их действия – либо невинные шалости, либо рекламные акции профессионалов);

взломщики – профессиональные кракеры, пользующиеся почетом и уважением в этой среде (их основная задача – взлом компьютерной системы с серьезными целями, например с целью кражи или подмены хранящейся в системе информации; как правило, для того чтобы осуществить взлом, необходимо пройти три основных стадии):

исследование вычислительной системы с выявлением в ней изъянов (уязвимости);

разработка программной реализации атаки;

непосредственное осуществление атаки.

Настоящим профессионалом можно считать того кракера, который для достижения своей цели проходит все три стадии. В принципе, работа взломщиков – обычное воровство.

Однако в нашем отечестве, где находящееся у пользователей программное обеспечение в преобладающей части является пиратским, т.е. украденным не без помощи тех же взломщиков, отношение к ним не столь категорично.

В связи с этим, если не ограничиваться рассмотрением хакеров и кракеров с позиций распределенных систем, то следует отметить, что самая многочисленная категория кракеров занимается снятием защиты с коммерческих версий программных продуктов, изготовлением регистрационных ключей (registration key) для условно бесплатных программ (shareware) и т.п.

Говоря о хакерах, следует еще отметить, что в последнее время сформировался устойчивый *миф об их всемогуществе и полной незащищенности компьютерных систем*. Действительно, современные вычислительные системы общего назначения имеют серьезные проблемы с безопасностью. Но речь идет именно о системах общего назначения. Там же, где требуется обработка критической информации и обеспечение высшего уровня защиты (например, в военной области, атомной энергетике и т.п.), используются специализированные защищенные вычислительные системы, которые изолированы от сетей общего назначения физически и не допускают

несанкционированного удаленного доступа извне. В то же самое время любая уважающая себя организация, будь то ЦРУ, АНБ, НАСА, имеет свои www- или FTP-серверы, находящиеся в открытой сети и доступные всем, и кракеры в этом случае проникали именно в них.

Другим, еще более устойчивым мифом, является *миф о всеобщей беззащитности банковских систем*. Действительно, в отличие от вычислительных систем стратегического назначения банки вынуждены для обеспечения удобства и быстродействия работы с клиентами предоставлять им возможность удаленного доступа из сетей общего пользования к своим банковским вычислительным системам. Однако для связи в этом случае используются защищенные криптопротоколы и разнообразные системы сетевой защиты, и к тому же предоставление клиенту возможности удаленного доступа, отнюдь, не означает, что клиент может получить доступ непосредственно к внутренней банковской сети.

По мнению специалистов, зарубежные банковские вычислительные системы являются наиболее защищенными вслед за системами стратегического назначения. В обоих случаях речь идет о несанкционированном удаленном доступом извне. В том случае, если нанести ущерб системам вознамерится кракер из состава персонала защищенной системы, трудно судить об успехе его попыток. Известен, например, случай на Игналинской АЭС, когда местный системный программист внедрил в вычислительную систему программу («троянского коня»), которая едва не привела к аварии на станции. Как утверждают статистики, нарушение безопасности системы собственным персоналом составляет около 90 % от общего числа нарушений. Таким образом, даже критические вычислительные системы нельзя считать неуязвимыми, но реализовывать на них успешную удаленную атаку практически невозможно.

Аналогичный вывод можно сделать и по отношению к банковским системам. В качестве примера можно привести дело российского программиста Левина, вскрывшего City Bank и наделавшего много шума в прессе. Однако вряд ли ему удалось это сделать благодаря своим выдающимся программистским способностям. Наиболее убедительной является версия, что у него все же были сообщники в этом банке, которые предоставили ему входное имя и пароль. Неявным подтверждением этого факта является то, что он

не смог скрыть своих «следов» и довольно быстро был обнаружен правоохранительными органами.

Как утверждают некоторые авторы, ни одного подтвержденного факта целенаправленного взлома с помощью программных средств (а не с помощью подкупа и т.п.) указанных выше систем ни в России, ни за рубежом пока обнаружить не удалось.

11.4. Основные методы защиты от удаленных и локальных атак

Говоря об удаленных атаках, следует отметить, что защита от них взаимосвязана с методами доступа и использованными пользователем ресурсами глобальной сети. Сети являются общедоступными. Удаленный доступ к этим ресурсам может осуществляться анонимно любым неавторизированным пользователем. Примером неавторизированного доступа является подключение к www- или FTP-серверам. В этом случае, если трафик пользователя будет перехвачен, пройдет через сегмент атакующего, то последний не получит ничего, кроме общедоступной информации, т.е. отпадает забота о защите информации. Если же планируется авторизированный доступ к удаленным ресурсам, то следует обратить на эту проблему особое внимание.

Методы защиты связаны также с используемой пользователем операционной системой, имея при этом в виду: собирается ли пользователь разрешать удаленный доступ из сети к своим ресурсам. Если нет, то пользователь должен использовать чисто «клиентскую» ОС (например, Windows 98 или NT Workstation). Удаленный доступ к данной системе в принципе невозможен, что, безусловно, повышает ее безопасность (хотя и не гарантирует ее полностью). Естественно, все ограничения, связанные с безопасностью, ухудшают функциональность системы. В связи с этим существует такая аксиома безопасности: «Принципы доступности, удобства, быстродействия и функциональности вычислительной системы антагонистичны принципам ее безопасности. Чем более удобна, быстра и многофункциональна вычислительная система, тем она менее безопасна». Естественно, полная изоляция компьютера от глобальной сети путем отключения разъема или создания выделенной линии связи обеспечивает абсолютную безопасность

от удаленных атак, однако полностью исключает функциональные возможности сетей и поэтому бессмысленна. Основная же цель комплексной защиты информации – обеспечение максимальных функциональных возможностей при достаточной защищенности системы. Среди разнообразных мер по защите от удаленных атак наиболее простыми и дешевыми являются административные меры. Например, как можно защититься от анализа сетевого трафика злоумышленником, если известно, что с помощью программного прослушивания можно перехватить любую информацию, которой обмениваются удаленные пользователи, когда по каналу передаются нешифрованные сообщения? Также известно, что базовые прикладные протоколы удаленного доступа TELNET и FTP не предусматривают элементарную защиту передаваемых по сети идентификаторов (имен) и аутентификаторов (паролей).

Поэтому администраторы сетей могут запретить использовать эти базовые протоколы для предоставления авторизированного доступа к ресурсам своих систем. При необходимости можно рекомендовать средства защиты этих протоколов.

Определенную опасность представляет использование так называемого протокола ARP. Этот протокол осуществляет поиск и сопоставление IP-адреса с адресом конкретной локальной сети (например, Ethernet) и направление следования именно по этому адресу. Послав специально сформированный ARP-пакет, злоумышленник может заставить окружающие его компьютеры посылать информацию не на маршрутизатор, а на его компьютер, прослушивая таким образом весь проходящий через сеть трафик. Чтобы устранить эту неприятность, связанную с отсутствием у ОС каждого хоста необходимой информации о соответствующих IP- и Ethernet-адресах остальных хостов внутри данного сегмента сети, сетевой администратор создает статическую ARP-таблицу в виде файла, куда вносится необходимая информация об адресах. Данный файл устанавливается на каждый хост внутри сетевого сегмента, и, следовательно, у сетевой ОС отпадает необходимость использования удаленного ARP-поиска.

Известна также уязвимость адресной службы DNS, что позволяет кракеру получить глобальный контроль над соединениями путем навязывания ложного маршрута через хост кракера – ложный DNS-сервер. Это приводит к катастрофическим последствиям для

огромного числа пользователей. Защита от ложного DNS-сервера – достаточно сложная задача, однако и в этом случае могут быть предложены административные методы, которые могут предотвратить установление такого глобального контроля или защитить подобную удаленную систему. Разработаны административные меры от навязывания ложного маршрута, защиты от отказа в обслуживании и от других причин нарушения безопасности информации.

Признавая важность административных мер защиты от удаленных атак, тем не менее, следует считать, что основную роль играют все же программно-аппаратные методы защиты. Центральным элементом в комплексе программно-аппаратных методов является криптография. В течение многих лет криптография использовалась исключительно в военных целях с привлечением и внедрением фундаментальных научных исследований (достаточно вспомнить работы А. Тьюринга и их практическую реализацию в крупном вычислительном комплексе в Великобритании в период Второй мировой войны). Агентство национальной безопасности АНБ Соединенных Штатов Америки и его аналоги в бывшем Советском Союзе, Англии, Франции, Израиле и других странах затратили миллиарды долларов на разработку криптографических методов, пытаясь обеспечить безопасность собственных линий связи и одновременно – взломать все чужие. Как правило, работы этих ведомств по данной тематике носили секретный характер. В последние 20 – 30 лет наблюдается бурный рост несекретных академических исследований в области криптографии. Сегодня современная компьютерная криптография широко практикуется и за стенами военных ведомств, что, безусловно, связано с расширением сферы деятельности людей, в которых возникает потребность в криптографических методах защиты информации. Это научное направление имеет очень серьезное теоретическое (математическое, алгоритмическое) обоснование и связано с работами таких выдающихся ученых, как упоминавшийся выше Алан Тьюринг, Клод Шеннон, Давид Кан и др. В открытой печати уже появилось довольно много публикаций, учебников и солидных монографий по криптографии [11], [12], [13].

Конкретная реализация методов криптографии связана с разработкой аппаратных и программных средств, так, например, мейнфреймы фирмы IBM, начиная с 90-х годов оснащаются криптографическими процессорами, обеспечивающими шифрование и

дешифрование сообщений с минимальной дополнительной нагрузкой на центральный процессор. Эти спецпроцессоры могут обрабатывать большие объемы данных и обеспечивать высокий уровень защищенности вычислительных систем. Важную роль в реализации криптографических методов играет разработка генераторов «истинно случайных» чисел и другие исследования.

В качестве примера применения криптографических методов рассмотрим процедуру защиты IP-протокола в глобальной сети Интернет. Обеспечить безопасность глобальной сети Интернет особенно трудно, поскольку дейтаграммы, передающиеся от отправителя до конечного получателя, проходят через несколько промежуточных сетей и маршрутизаторов, не контролируемых ни отправителем, ни получателем. Таким образом, поскольку дейтаграммы могут быть перехвачены без ведома отправителя, их содержимому нельзя доверять. Например, рассмотрим сервер, который использует процедуру аутентификации источника для проверки того, что запросы поступают от авторизированных клиентов. Процедура аутентификации источника требует, чтобы сервер при получении каждой дейтаграммы проверял IP-адрес отправителя, и принимал запросы только от компьютеров, адреса которых перечислены в специальном списке. Данный вид аутентификации обеспечивает слабую защиту, поскольку ее можно легко обойти. В частности, один из промежуточных маршрутизаторов может контролировать трафик проходящих через него дейтаграмм и фиксировать IP-адреса авторизированных клиентов, которые могут быть перехвачены любым злоумышленником, контролирующим этот маршрутизатор. Затем этот злоумышленник может выступить в роли авторизированного клиента.

Группа IETF (Internet Engineering Task Forse) – инженерная группа, входящая в структуру архитектурного совета Интернет, разработала набор протоколов, которые обеспечивают безопасную связь в глобальной сети. Все вместе они называются семейством протоколов IPsec (IP security или защитным протоколом IP). В этих протоколах аутентификация и шифрование данных выполняются на уровне протокола IP.

Методы криптографии, применяемые при разработке разнообразных криптопротоколов, составляют основу программных методов защиты информации в сетях. В то же время они являются составной частью так называемой *методики Firewall*, являющейся

сейчас основой программно-аппаратных средств осуществления сетевой политики безопасности в IP-сетях и реализующих следующие функции.

Многоуровневая фильтрация сетевого трафика. Фильтрация обычно происходит на четырех уровнях OSI:

канальном (Ethernet);

сетевом (IP);

транспортном (TCP, UDP);

прикладном (FTP, TELNET, HTTP, SMTP и т.д.).

Фильтрация сетевого трафика является основной функцией системы Firewall и позволяет администратору безопасности сети централизованно осуществлять необходимую сетевую политику в выделенном сегменте IP-сети. Настроив для этого соответствующим образом Firewall, можно разрешить или запретить пользователям как доступ из внешней сети к соответствующим службам хостов или к хостам, находящимся в защищенном сегменте, так и доступ пользователей из внутренней сети к соответствующим ресурсам внешней сети.

Proxy – схема с дополнительной идентификацией и аутентификацией пользователей. Смысл proxy-схемы заключается в создании соединения с конечным адресатом через промежуточный сервер, называемый proxy-сервером (proxy – полномочный).

Создание приватных сетей с виртуальными IP-адресами. Если администратор безопасности считает целесообразным скрыть истинную топологию своей внутренней IP-сети, он может использовать proxy-сервер для отделения своей внутренней приватной сети со своими внутренними виртуальными IP-адресами, которые, очевидно, непригодны для внешней адресации. При этом proxy-сервер должен осуществлять связь с абонентами из внешней сети со своего настоящего IP-адреса. Эти же схемы применяются и в том случае, если для создания IP-сети выделено недостаточное количество IP-адресов. Основным аппаратным компонентом для реализации методики управления доступом к объединенной сети является специализированное устройство, называемое *брандмауэром*, ассоциируемое с термином Firewall (термин брандмауэр позаимствован из строительства, где он обозначает толстую несгораемую стену, благодаря которой секция строения становится непроницаемой для огня). Обычно брандмауэр устанавливается между внутренней

сетью организации и каналом, ведущим к внешним сетям (например, к глобальной сети Интернет), брандмауэры разделяют объединенную сеть на две области, которые неофициально называются внутренней и внешней. Хотя сама идея брандмауэра проста, ее реализация усложняется множеством факторов. Один из них – внутренняя сеть организации может иметь несколько внешних соединений. При этом необходимо сформировать периметр безопасности (security perimeter), установив брандмауэр на каждое внешнее соединение. Чтобы гарантировать эффективность периметра безопасности, во всех брандмауэрах должны использоваться одинаковые ограничения доступа. В противном случае злоумышленники могут обойти ограничения, наложенные одним брандмауэром, и зайти в объединенную сеть через другой.

Существует несколько способов реализации брандмауэров [10]. Выбор способа зависит от того, какое количество внешних каналов существует в организации. В большинстве случаев каждый барьер в брандмауэре реализуется на основе маршрутизатора, содержащего фильтр пакетов. Чтобы брандмауэр не замедлял работу сети, его аппаратное и программное обеспечение должно быть оптимизировано на решение конкретной задачи. Решению этой задачи способствует и тот факт, что в большинство коммерческих маршрутизаторов включен быстродействующий механизм фильтрации пакетов, который выполняет основную часть работы.

На практике, как правило, возникает необходимость создать безопасный брандмауэр, который предотвратит нежелательный доступ извне, и в то же время позволит пользователям внутренней сети получить доступ к внешним службам. При этом необходимо выработать специальный механизм безопасности. В общем случае организация может обеспечить доступ к внешним службам только через защищенный компьютер. Поэтому обычно с каждым брандмауэром связывают один защищенный компьютер и устанавливают на этом компьютере набор шлюзов уровня приложения. Для того чтобы такой компьютер мог служить в качестве безопасного канала связи, его степень защиты должна быть очень высока. Поэтому такой компьютер часто называют бастионным узлом. На бастионном узле запускаются службы, которые организация хочет сделать видимыми извне, а также proxy-серверы, которые позволяют внутренней сети получить доступ к внешним серверам. В брандмауэре

также может использоваться так называемая «тупиковая сеть», которая позволяет изолировать внешний трафик от внутреннего. К этой сети подключаются брандмауэры, а также бастионный узел.

Все брандмауэры можно разделить на три типа:

пакетные фильтры (packet filter);

серверы прикладного уровня (application gateways);

серверы уровня соединения (circuit gateways).

Все типы могут одновременно встретиться в одном брандмауэре.

Пакетные фильтры. Брандмауэры с пакетными фильтрами принимают решение о том, пропускать пакет или отбросить, просматривая IP-адреса, флажки или номера TCP-портов в заголовке этого пакета. IP-адрес и номер порта – информация сетевого и транспортного уровней соответственно, но пакетные фильтры используют и информацию прикладного уровня, так как все стандартные сервисы в TCP/IP ассоциируются с определенным номером порта.

Серверы прикладного уровня. Брандмауэры с серверами прикладного уровня используют серверы конкретных сервисов – TELNET, FTP и т.д. (proxy server), запускаемые на брандмауэре и пропускающие через себя весь трафик, относящийся к данному сервису. Таким образом, между клиентом и сервером образуются два соединения: от клиента до брандмауэра и от брандмауэра до места назначения.

Использование серверов прикладного уровня позволяет решить важную задачу – скрыть от внешних пользователей структуру локальной сети, включая информацию в заголовках почтовых пакетов или службы доменных имен (DNS). Другим положительным качеством является возможность аутентификации на пользовательском уровне.

При описании правил доступа используются такие параметры, как название сервиса, имя пользователя, допустимый временной диапазон использования сервиса, компьютеры, с которых можно пользоваться сервисом, схемы аутентификации. Серверы протоколов прикладного уровня позволяют обеспечить наиболее высокий уровень защиты – взаимодействие с внешним миром реализуется через небольшое число прикладных программ, полностью контролирующих весь входящий и выходящий трафик.

Серверы уровня соединения. Сервер уровня соединения представляет собой транслятор TCP-соединения. Пользователь образует соединение с определенным портом на брандмауэре, после чего последний производит соединение с местом назначения по другую сторону от брандмауэра. Во время сеанса этот транслятор копирует байты в обоих направлениях, действуя как провод.

Как правило, пункт назначения задается заранее, тогда как источников может быть много (соединение типа «один со многими»). Используя различные порты, можно создавать различные конфигурации. Такой тип сервера позволяет создавать транслятор для любого определенного пользователем сервиса, базирующегося на TCP, осуществлять контроль доступа к этому сервису, сбор статистики по его использованию.

Говоря о программных средствах защиты информации, необходимо отметить, что конечной целью атаки кракера является определенный компьютер, с конкретной реализацией сетевых протоколов, с конкретной определенной системой. В связи с этим необходимо коснуться защиты операционных систем. Среди типичных атак, которым могут быть подвергнуты любые ОС, можно указать следующие [6]:

кража пароля (подглядывание за несколькими пользователями, получение из файла, кража носителей и т.п.);

подбор пароля (перебор возможных вариантов, включая оптимизированный перебор);

копирование «жестких» дисков компьютера;

сбор «мусора»: если средство ОС позволяют восстанавливать ранее удаленные объекты, злоумышленник может получить доступ к удаленным объектам (удаленных другими пользователями), просмотрев содержимое их мусорных корзин;

превышение полномочий: используя ошибки в программном обеспечении или администрировании ОС, злоумышленник получает полномочия, превышающие те, которые предоставлены ему согласно действующей политике безопасности;

отказ в обслуживании (целью этой атаки является частичный или полный вывод ОС из строя, как правило, с помощью вирусов).

Самой распространенной ОС в глобальной сети Интернет является ОС Unix, основными протоколами, определяющими сети Интернет, являются протоколы TCP/IP, которые были разработаны

для ОС Unix. Не менее 90 % мощных Интернет-узлов работают под управлением этой ОС и различных ее диалектов. Основные концепции Unix разрабатывались в конце 60-х – начале 70-х годов прошлого столетия, когда не было никакой теории компьютерной безопасности, и никто не подозревал, о тех крупных неприятностях, которые возникнут по мере развития сетевых технологий.

Современные сетевые операционные системы оказываются в заведомо более выгодном положении, поскольку они разрабатывались с учетом ошибок Unix и современной ситуации с безопасностью сетей. Однако это вовсе не говорит об их большей безопасности.

За долгий срок жизни Unix исследователями написаны, а администраторами изучены сотни статей и книг относительно механизмов безопасности Unix и способов их нарушения. Все это позволяет предположить, что никаких сюрпризов Unix больше не преподнесет.

С новыми ОС ситуация прямо противоположная. И хотя в них заложены концепции, согласующиеся с современным состоянием теории безопасности, у них очень малый срок эксплуатации. Они активно исследуются хакерами и кракерами, и, несмотря на опыт Unix, начинают проходить тот же самый путь и совершать те же самые ошибки в обеспечении безопасности.

В дополнение к гл. 6 следует добавить несколько замечаний о структуре средств информационной безопасности ОС Unix и наиболее слабых ее местах. Как известно, изначально Unix была ориентирована на централизованные вычисления в системах коллективного пользования как многозадачная, многопользовательская ОС. Пользователи системы разделялись на группы, в зависимости от прав доступа (или привилегий):

суперпользователь (root), имеющий неограниченные права;

обычный пользователь, имеющий права в рамках своего идентификатора (UID, user ID) и членство в группе (GID, group ID) – права и ограничения устанавливаются для него суперпользователем.

По мере развития ОС и использования Unix-машин в качестве серверов в глобальных сетях среди обычных пользователей выделялись так называемые *специальные пользователи*. Они, как правило, имеют зарегистрированные имена (guest, bin, uucp и т.п.) и но-

мера UID и GID. Прав у этого пользователя еще меньше, чем у обычного. Их устанавливает суперпользователь для работы с конкретными приложениями. Одним из интересных примеров специального пользователя является анонимный пользователь FTP, который так и называется anonymous, или ftp.

И, наконец, есть категория так называемых *псевдопользователей*, не имеющих никаких прав и не идентифицируемых системой. Но они могут подключаться к системе с помощью так называемых *программ-демонов* (в современной терминологии серверов), в частности, используя средства электронной почты *e-mail*. От этого пользователя не требуется аутентификации, учет по нему также не ведется.

Именно две последние категории пользователей (особенно последняя) и являются причиной основных неприятностей в ОС Unix с точки зрения информационной безопасности. Среди основных причин уязвимости Unix принято считать наличие демонов; механизм SUID/SGID; излишнее доверие (поскольку в не столь давние времена создатели делали систему «под себя», не подозревая, насколько теснее и опаснее станет компьютерный мир через несколько лет); «люки».

Механизм SUID/SGID позволяет пользователю запускать некоторые программы с правами администратора системы, например для обеспечения возможности изменения собственного пароля, поскольку файл с паролями, разумеется, не доступен рядовым пользователям.

Также слабым местом Unix-систем являются «люки». Люком, или «черным входом» (backdoor), часто называют оставленную разработчиком недокументированную возможность взаимодействия с системой (чаще всего – входа в нее), например известный только разработчику универсальный «пароль». Люки оставляют в конечных программах вследствие ошибки, не убрав отладочный код, или вследствие необходимости продолжения отладки уже в реальной системе из-за ее высокой сложности, или же из корыстных интересов. Люки – любимый путь в удаленную систему не только у хакеров, но и у журналистов, и режиссеров вместе с подбором «главного» пароля перебором за минуту до взрыва, но в отличие от последнего способа люки реально существуют. Классический пример люка – это, конечно, отладочный режим в sendmail.

Учитывая динамику развития ОС Unix в настоящее время, можно сказать, что она является наиболее мощной и надежной, в том числе и с точки информационной безопасности системы.

На рынке достаточно много предложений средств защиты Интернет, однако по ряду параметров ни одно из них не может быть признано адекватным задачам защиты информации именно для Интернет. Например, достаточно криптостойкой и замечательной по своей идее является распространенная система PGP (Pritty good privacy). Однако, поскольку PGP обеспечивает шифрование файлов, она применима только там, где можно обойтись файловым обменом. Защитить, например, приложения «on-line» при помощи PGP затруднительно. Кроме того, уровень иерархии управления защиты PGP слишком высок: эту систему можно отнести к прикладному или представительскому уровням модели OSI. Стыковка защиты PGP с другими прикладными системами потребует также определенных усилий, если, конечно, вообще окажется осуществимой. Альтернативу таким «высокоуровневым» системам защиты среди традиционных решений составляют устройства защиты канального и физического уровня – скремблеры и канальные шифраторы. Они «невидимы» с прикладного уровня и в этом смысле совместимы со всеми приложениями. Однако такие системы имеют ограниченную совместимость с различным каналообразующим оборудованием и физическими средами передачи данных. Это, как правило, не сетевые устройства, способные распознавать топологию сети и обеспечить связь из конца в конец через многие промежуточные узлы, а «двухточечные» системы, работающие на концах защищаемой линии и поэтому вносящие значительную аппаратную избыточность. И, конечно же, на таких устройствах невозможно построить систему защиты в рамках такой сети, как Интернет, уже хотя бы потому, что невозможно обеспечить их повсеместное распространение (вследствие высокой цены) и всеобщую аппаратную совместимость.

В заключение этого краткого обсуждения методов защиты информации следует отметить, что люди представляют собой наиболее уязвимое звено в любой системе безопасности. Служащий фирмы либо по злому умыслу, либо по неосторожности, либо, не зная принятой в организации стратегии, может поставить под угрозу самую современную систему безопасности. В изучении методов

защиты информации сложилось даже целое направление – социальная инженерия, связанная со злоупотреблением доверия пользователей, например, как одним из наиболее эффективных методов получить информацию у ничего не подозревающих людей, особенно в больших организациях, где многие пользователи не знают персонал своих компьютерных подразделений «в лицо», общаясь, в основном, по телефону. По определению самих хакеров «социальная инженерия» – термин, используемый взломщиками и хакерами для обозначения несанкционированного доступа другим способом, чем взлом программного обеспечения; цель – обмануть людей для получения паролей к системе или иной информации, которая поможет нарушить безопасность системы. Классическое мошенничество включает звонки по телефону, электронную почту, разговоры по Интернет в «реальном времени», обыкновенную почту, личные встречи и т.п.

11.5. Компьютерные вирусы

Проблема «вирусов» и «вирусной безопасности» возникла достаточно давно. Первые исследования саморазмножающихся искусственных технических и программных конструкций проводились в середине прошлого столетия в работах фон Неймана, Винера и других ученых. Было дано определение и проведен математический анализ конечных автоматов, в том числе и самовоспроизводящихся. Термин «компьютерный вирус» появился позднее – официально считается, что его впервые употребил сотрудник Лехайского университета (США) Фред Коэн в 1984 г. на Седьмой конференции по безопасности информации, проходившей в США. Идеи вирусов были изложены широкой публике еще в 1983 г. известным разработчиком ОС Unix Кэном Томпсоном в одной из своих лекций.

Одной из самых известных практических реализаций чисто теоретических работ фон Неймана и других известных ученых, явилась программа Worm («червь»), созданная осенью 1988 г. студентом выпускного курса Корнельского университета Робертом Морисом, который занимался в Bell Laboratories программным обеспечением безопасности Unix. Запущенный на сетевой машине «червь» искал в сети Интернет машины с серверами и использовал их для воссоздания себя в большом количестве копий. Такое дейст-

вие «червя» стало возможным в результате использования ошибки в программе («демоне») Unix fingerd. Вирус распространялся с поразительной скоростью и появлялся в самых различных районах США. Через пять часов было поражено пять систем, через двое суток – шесть тысяч. По самым скромным оценкам вирус Мориса стоил свыше 8 млн ч потери доступа и свыше миллиона часов прямых потерь на восстановление работоспособности систем. Общая стоимость этих затрат оценивается в 98 млн дол. Ущерб был бы еще большим, если бы вирус изначально создавался с разрушительными целями.

Этот крупнейший инцидент в области компьютерной безопасности доказал (не теоретически, а практически) возможность создания саморепродуцирующихся программ, дал толчок к появлению целой отрасли компьютерной безопасности – компьютерной вирусологии, а также выявил необходимость разработчиков Unix более серьезно заняться безопасностью этой ОС. К тому времени уже существовали единичные вирусы и на персональных компьютерах – саморепродуцирующиеся в пределах одного компьютера (видимо, поэтому сетевые вирусы стали называться «червями»).

Так что же такое компьютерный вирус? По определению, данному одним из известных отечественных специалистов Евгением Касперским, компьютерным вирусом называется программа, которая может создавать свои копии и внедряться в файлы и системные области компьютера, вычислительной сети и т.п. При этом копии сохраняют способность дальнейшего распространения. Другими словами, *компьютерный вирус* – небольшая программа (средний размер – 700 байт), написанная на языке Assembler и выполняющая разрушительные для ОС действия. Следует отметить, что такие программы как «бомбы» и «троянские кони» также приводят к неприятностям в системах, но отличаются от вирусов, так как не обладают свойством саморазмножения.

Название «вирус» распространилось ввиду явного сходства с биологическим прототипом. Суть воздействия биологического вируса сводится к нарушению информации, содержащейся в генетическом коде клетки. Посредством небольшого изменения фрагмента ДНК и РНК он захватывает управление жизненным процессом клетки. Таким образом, вирус обеспечивает себе возможность сво-

бодно и неограниченно размножаться. Это часто приводит к трагическим последствиям.

Если компьютерную систему сопоставить с живым организмом, а отдельные программы – с клетками, то получим полную аналогию. Компьютерный вирус разрушает информацию, содержащуюся в коде программы. Он перехватывает контроль над компьютерной системой путем замены небольшого фрагмента программы, что позволяет ему неограниченно размножать свой код. Так же, как и биологический аналог, компьютерные вирусы:

представляют опасность для той системы, на которой они паразитируют;

быстро размножаются, легко распространяются на большие расстояния;

проявляют себя не сразу;

имеют «латентный период», во время которого вирус продолжает распространяться в компьютерной системе;

играют важную роль в борьбе с «заболеваниями».

Биологическая аналогия оказывается настолько глубокой, что в литературе, посвященной компьютерным вирусам, широко используются и другие медицинские термины: «заболевание», «вакцина», «лечение», «карантин» и др., что иногда приводит к недоразумениям, когда забывается, что компьютерный вирус является обычной программой для компьютерной системы, которая имеет своего создателя.

Вирусы можно разделить на классы по следующим признакам:

по среде обитания вируса;

по способу заражения среды обитания;

по деструктивным возможностям.

По среде обитания различают, прежде всего, *сетевые вирусы*, или *вирусы-черви* (*worm*), которые распространяются в компьютерной сети. Проникая в память компьютера, они вычисляют сетевые адреса других машин и по этим адресам рассылают свои копии.

Файловые вирусы являются наиболее распространенным типом и обладают наибольшей инфицирующей способностью. Объектом поражения файловых вирусов являются исполняемые файлы, драйверы устройств и файлы ОС. По способу заражения файловые вирусы делятся на *резидентные* и *нерезидентные.*

Нерезидентный файловый вирус при запуске пораженной программы ищет первую «жертву» – незараженный файл в текущей директории, и дописывает к ней свое тело, а затем передает управление запущенной программе. Нерезидентные вирусы не заражают память компьютера и сохраняют активность ограниченное время.

Резидентные вирусы находятся в памяти компьютера, оставляя в оперативной памяти свою резидентную часть, которая перехватывает обращение ОС к объектам заражения и внедряется в них. Резидентные вирусы находятся в памяти и остаются активными вплоть до выключения компьютера или его перезагрузки.

Загрузочные вирусы внедряются в загрузочный сектор системного диска, проникая в компьютер при загрузке зараженной дискеты. При идентифицировании диска вирус в большинстве случаев переносит оригинальный boot-сектор в какой-либо другой сектор диска, а сам записывается на его место. В результате при загрузке с зараженного диска вместо настоящего boot-сектора будет выполнен программный код вируса, который при первой возможности делает свое «черное дело». В настоящее время этот вид вирусов практически обречен, так как практически все машины имеют защиту boot-сектора.

Вирусы всех типов могут распространяться по сети. По своим деструктивным возможностям «троянский» компонент вируса обычно разделяют на:

безвредные вирусы, никак не влияющие на работу компьютера, кроме изменения свободной памяти на диске в результате своего размножения;

неопасные вирусы, влияние которых ограничивается уменьшением объема свободной памяти на диске и графическими, звуковыми и прочими эффектами, к которым относится, например, выдача букв или проигрывание какой-нибудь мелодии в определенное время;

опасные вирусы, которые могут привести к серьезным сбоям в работе компьютера;

очень опасные вирусы, которые могут привести к потере программ; уничтожить данные; стереть необходимую для работы компьютера информацию, записанную в системных областях памяти; и даже способствовать ускоренному износу движущихся частей диска.

Вирусы-«черви» (*worm*) – вирусы, которые распространяются в компьютерной сети и, так же как и вирусы-«спутники», не изменяют файлы или секторы на дисках. Они проникают в память компьютера из компьютерной сети, вычисляют сетевые адреса других компьютеров и рассылают по этим адресам свои копии. Такие вирусы иногда создают рабочие файлы на дисках системы, но могут вообще не обращаться к ресурсам компьютера (за исключением оперативной памяти).

Большинство вопросов связано с термином «полиморфный вирус». Этот вид компьютерных вирусов представляется на сегодняшний день наиболее опасным. *Полиморфные вирусы* – вирусы, модифицирующие свой код в зараженных программах таким образом, что два экземпляра одного и того же вируса могут не совпадать ни в одном бите. Такие вирусы не только шифруют свой код, но и содержат код генерации шифровщика и расшифровщика, что отличает их от обычных шифровальных вирусов, которые также могут шифровать участки своего кода, но имеют при этом постоянный код шифровальщика и расшифровщика. Полиморфные вирусы – вирусы с самомодифицирующимися расшифровщиками. Цель такого шифрования: имея зараженный и оригинальный файлы, вы все равно не сможете проанализировать его код с помощью обычного дизассемблирования. Этот код зашифрован и представляет собой бессмысленный набор команд. Расшифровка производится самим вирусом уже непосредственно во время выполнения. При этом возможны варианты: он может расшифровать себя всего сразу, а может выполнять такую расшифровку «по ходу дела», может вновь шифровать уже отработавшие участки. Все это делается с целью затруднения анализа кода.

Отметим еще *макровирусы*. Это особая разновидность вирусов, которые поражают документы в прикладных программах, имеющие расширение .doc, например документы, созданные текстовым процессором MS Word и выполняющие макрокоманды. Если открыть файл документа в окне, происходит заражение.

Абсолютных гарантий антивирусной безопасности не имеется, даже при наличии самых наилучших антивирусных программ. Однако путем соблюдения определенных правил профилактики (так называемой компьютерной гигиены), можно снизить до минимума риск заражения компьютеров.

Необходимо регулярно делать резервные копии файлов, с которыми ведется работа, на внешний носитель.

Следует покупать дистрибутивные копии программного обеспечения только у официальных продавцов.

Не следует запускать непроверенные антивирусные программы, полученные из сомнительных источников.

При лечении дисков следует использовать заведомо «чистую» операционную систему.

Необходимо иметь в виду, что очень часто вирусы переносятся с игровыми программами, с которыми следует быть предельно осторожным. В заключение следует отметить, что, кроме вирусов, существует другой вид программ, представляющих опасность для вычислительных систем, о которых ранее упоминалось. Это так называемые «троянские» программы. Такие программы не способны самостоятельно размножаться, и их распространение основано целиком на добровольном копировании. При запуске такой программы она, выполняя внешне безобидные действия, одновременно портит данные в компьютере. «Троянские программы» распространяются значительно медленнее, чем вирусы, поскольку, уничтожив систему, они погибают сами. Как правило, их маскируют под игровые программы или широко известные пакеты.

11.6. Криптографические методы защиты информации

Поскольку в основе всех автоматизированных сетевых и компьютерных систем безопасности находится криптография, пользователю любого уровня полезно иметь хотя бы общее представление об этом направлении развития вычислительных систем.

Слово «криптография» греческого происхождения и означает тайнопись, в современной терминологии – шифрование, которое заключается в преобразовании открытого текста или данных в нечитаемую форму при помощи обратимых математических операций.

До изобретения в 70-х годах XIX в. шифрования с так называемым открытым ключом традиционное шифрование (иногда называемое шифрование с обычным ключом) было единственным типом шифрования. Этот тип шифрования использовался для секретного взаимодействия очень многими людьми и группами людей, начиная с Юлия Цезаря и заканчивая немецкими подводными лод-

ками, а также современными дипломатами, военными, бизнесменами и т.п. Традиционное шифрование до сих пор остается самым популярным, используя алгоритм, при котором шифрование и дешифрование осуществляется при помощи одного и того же ключа (метода преобразования текста).

Традиционную схему шифрования образуют пять составляющих.

Открытый текст – исходное сообщение или данные, которые подаются на вход алгоритма.

Алгоритм шифрования – различные способы преобразования текста.

Секретный ключ подается также на вход алгоритма шифрования наряду с открытым текстом. Конкретные операции замены и перестановки, выполняемые алгоритмом, зависят от ключа.

Зашифрованный текст – искаженное сообщение, формируемое на выходе алгоритма. Зашифрованный текст представляет собой функцию открытого текста и секретного ключа. Если закодировать одно и то же сообщение с помощью двух разных ключей, получим два разных варианта зашифрованного текста.

Алгоритм дешифрования – алгоритм шифрования, работающий в реверсивном режиме. Он принимает на вход зашифрованный текст и секретный ключ, а на выходе генерирует оригинальный открытый текст.

Для обеспечения безопасности традиционного шифрования необходимо выполнение двух условий:

нужен надежный алгоритм шифрования, секретный ключ которого не должен быть доступен злоумышленнику;

отправитель и получатель должны получать копии секретного ключа безопасным образом и обеспечить надежное его хранение.

Существует два основных способа атаки на традиционную схему шифрования, первый из которых называется криптоанализом, он опирается на природу алгоритма, а также на определенные сведения об общих характеристиках открытого текста, включая примеры перехваченных пар открытого и зашифрованного текстов. Цель заключается в выявлении ключа шифрования. Второй метод называется «методом грубой силы» или решением задачи «в лоб». Он включает полный перебор всех возможных вариантов ключа к зашифрованному сообщению, пока не будет получен осмысленный

вариант открытого текста. Это требует зачастую вычислительных систем высокой производительности и зависит от размерности ключа шифрования, которая может изменяться в широких пределах.

Существует ряд стандартов шифрования. Так, Национальный институт стандартов и технологий США (NIST) принял следующие стандарты:

DES (Data Encryption Standard – стандарт шифрования данных), в котором открытый текст разбивается на блоки по 64 бит, каждый из которых кодируется с помощью 56-разрядного ключа;

тройной алгоритм TDEA (Tryple DEA), представляющий троекратное шифрование методом DES, разработанный в связи с потенциальной уязвимостью DES, принят как стандарт ANSI для использования в финансовых приложениях при трех различных ключах, эффективная длина ключа составляет 68 бит;

алгоритм шифрования AES (Advanced Encryption Standard – передовой стандарт шифрования), был разработан как альтернатива TDEA, который при очень высокой надежности и эффективности работал очень медленно и имел очень короткие блоки обработки (что взаимосвязано).

Шифрование с открытым ключом впервые было предложено в 1976 г. Диффи (Diffie) и Хеллманом (Hellman) и представляет прорыв в криптографии, первый за последнюю тысячу лет. Эта разновидность криптографической системы, в которой шифрование и дешифрование осуществляется при помощи двух различных ключей, один из которых называется открытым, а второй закрытым, т.е. шифрование становится асимметричным. В основе алгоритмов шифрования открытым ключом лежат математические функции, а не простые операции с последовательностью битов, характерные для симметричного шифрования.

В схему шифрования с открытым ключом входит шесть составляющих:

открытый текст – исходное сообщение или данные, которые подаются на вход алгоритма;

алгоритм шифрования, представляющий различные способы преобразования текста;

открытые и закрытые ключи, которые выбирают таким образом, что один применяется для шифрования, а другой для дешифрова-

ния (конкретные преобразования, осуществляемые с открытым текстом, зависят от используемого открытого или закрытого ключей);

зашифрованный текст – искаженное сообщение, формируемое на выходе алгоритма (зашифрованный текст представляет собой функцию открытого текста и ключа, если зашифровать одно и тоже сообщение с помощью двух различных ключей, получим два различных варианта зашифрованного текста);

алгоритм дешифрования (включается, когда на его вход поступает зашифрованный текст и второй ключ пары, а на выходе генерируется оригинальный текст).

Процесс работает независимо от того, в каком порядке действует пара ключей. Открытый ключ публикуется для использования всеми желающими, тогда как закрытый ключ известен только владельцу. Одним из основных достоинств этого метода является то, что вычислительными методами трудно определить закрытый ключ по открытому.

Очень удачной аналогией этого способа кодирования является почтовый ящик. Шифрование открытым ключом аналогично опусканию письма в почтовый ящик – может сделать кто угодно, просто открыв паз и опустив письмо в почтовый ящик. Расшифрование с закрытым ключом при этом подобно извлечению почты из почтового ящика. Открыть его гораздо сложнее. Обычно если у вас есть ключи от почтового ящика, извлечь письмо не трудно.

Один из наиболее популярных алгоритмов при шифровании открытым ключом является алгоритм RSA (Rivest – Shamir – Adleman), предложенный в 1977 г. сотрудником Массачусетского технологического института. В этом алгоритме открытый и зашифрованный тексты рассматриваются как целые числа в диапазоне от 0 до $n - 1$. Для шифрования в сети используются арифметические операции деления по модулю. Алгоритм шифрования основан на сложности разложения большого числа на множители.

Однако при всех достоинствах шифрования открытым ключом не следует считать, что он является универсальной технологией, с появлением которой традиционное шифрование можно считать устаревшим. Современные схемы шифрования открытым ключом требуют очень больших объемов вычислений, что вряд ли позволит отказаться в обозримом будущем от традиционных методов шиф-

рования. К тому же распространение ключей в данном методе шифрования, отнюдь, не представляет тривиальную задачу. Следует также отметить, что защита от криптоанализа при шифровании открытым ключом не всегда лучше, чем у традиционного шифрования. В связи с этим оба эти метода применяются на практике, дополняя друг друга.

Важным криптографическим дополнением, подтверждающим достоверность передаваемой зашифрованной информации, является использование так называемой цифровой подписи, которая, с одной стороны, обеспечивает аутентификацию пользователей, а с другой стороны – исключает отказ посылающего от своего сообщения.

Следует отметить, что подпись от руки издавна пользуется для доказательства авторских прав или согласия с документом. Среди наиболее важных моментов подписи следующие:

подпись достоверна, убеждает получателя, что человек, подписавший документ, сделал это сознательно;

подпись не поддельна, она доказывает, что именно указанный человек подписал документ;

подпись невозможно использовать повторно, она часть документа, ее невозможно нанести на другой документ;

подписанный документ изменить невозможно;

от подписи нельзя отречься.

Хотя все эти утверждения не бесспорны, однако действия мошенников в традиционно подписываемых документах затруднены, и они рискуют быть разоблаченными.

Однако реализация электронной подписи и передача ее в сетях требует специальной защиты, так как ситуация оказывается гораздо сложнее. Во-первых, компьютерные файлы легко копируются. Даже если подпись человека – графическое изображение, подписи от руки подделать нелегко, можно без труда «вырезать» подлинную подпись из этого документа и вставить ее в другой. Таким образом, просто наличие в документе такой подписи ничего не означает. Во-вторых, компьютерные файлы легко изменить уже после подписания документа, не оставив ни малейшего следа изменения.

В связи с этим механизм цифровой подписи, реализуемый криптографическими методами, состоит из двух процессов:

формирование подписи блока данных – при передаче блока данных используется не только шифрование открытым ключом, но

также еще один алгоритм, называемый защитной функцией хэширования, которая определяется из передаваемого сообщения;

проверка подписи в принятом блоке данных.

Любое изменение сообщения злоумышленником приводит к изменению хэш-кода и недостоверности подписи при дешифровке. Фактически к сообщению добавляются данные, играющие роль подписи, поскольку они шифруются закрытым ключом отправляющего сообщение, что является гарантией подлинности и целостности документа.

Одним из самых ранних примеров использования цифровой подписи было упрощение проверки соблюдения договоров о ядерных испытаниях. Соединенные Штаты, Советский Союз разрешали друг другу разместить за границей сейсмографы для мониторинга ядерных испытаний. Проблема заключалась в том, что каждая сторона хотела быть уверенной в том, что страна, в которой размещены приборы, не подделывает их показаний. В свою очередь, страна, в которой размещались сейсмографы, искала гарантий, что приборы посылают только ту информацию, которая нужна для мониторинга и испытаний.

Эта проблема была решена с помощью цифровых подписей.

Страна, на территории которой стоял сейсмограф, может читать, но не изменять данные сейсмографа, а наблюдающая сторона знает, что данные не подделываются. Таким образом, комбинируя цифровые подписи и криптографию открытым ключом можно создать протокол, сочетающий надежное шифрование с достоверностью цифровых подписей.

Еще один вопрос, который неизбежно возникает при внедрении криптографических устройств, – вопрос о месте их размещения. Имеется два варианта решения этого вопроса: шифрование в линиях и сквозное шифрование.

При шифровании в линиях каждая уязвимая линия оборудуется на обоих концах шифрующими устройствами. Таким образом, может быть защищен весь трафик во всех линиях связи, хотя в большой сети для этого потребуется очень много таких устройств.

Один из основных недостатков такого подхода связан с тем, что на каждом промежуточном коммутаторе виртуального канала сообщение должно быть дешифровано и зашифровано заново с тем, чтобы определить адрес необходимого направления пакета. Таким

образом, сообщение оказывается уязвимым на каждом коммутаторе, поскольку в общественных сетях с коммутацией пакетов пользователь не может контролировать безопасность каждого узла. Кроме того, шифрование в линиях при существующих на сегодня скоростях передачи данных находится на пределе технических возможностей, что ограничивает применение данного варианта шифрования.

При сквозном шифровании криптографический процесс выполняется на двух оконечных системах. В зашифрованном виде данные пересылаются по сети без каких-либо изменений, вплоть до хоста, которому предназначаются эти данные. Такой подход должен вроде бы защищать от атак на сетевые коммутаторы, однако у данного подхода также есть слабое место. Для того, чтобы коммутаторы могли направлять пакеты по нужному адресу, заголовок письма не должен быть зашифрован. Таким образом, при сквозном шифровании пользовательские данные надежно защищены, однако маршрут данных не защищен, так как заголовок пакетов не шифруется.

Для повышения уровня защиты возникает необходимость совместного использования обоих методов – сквозного шифрования с включением элементов шифрования в линиях. В заключение следует сказать, что шифрование, состоящее в сокрытии содержания сообщения путем приведения его в нечитаемый вид, является наиболее важным автоматизированным средством обеспечения сетевой безопасности.

Традиционное шифрование и шифрование с открытым ключом часто комбинируются для предоставления услуг, связанных с обеспечением безопасности. Передаваемые данные шифруются традиционными методами, часто с помощью одноразового сеансового ключа. Сеансовый ключ может передаваться доверенным центром распространения ключей в зашифрованном, по алгоритму шифрования с открытым ключом, виде.

Шифрование с открытым ключом также может использоваться для создания цифровых подписей с целью аутентификации источников сообщений.

11.7. Бизнес в Интернете

В конце 1999 г. сразу два автомобильных гиганта Ford и General Motors объявили о своем намерении перевести большую часть своих коммерческих операций во Всемирную паутину и объединить поставщиков, деловых партнеров и покупателей во всем мире на виртуальных рынках. Это были первые примеры того, как крупные компании собираются воспользоваться Интернетом, чтобы оказаться в центре новой электронной коммерческой системы, способной изменять сам способ ведения бизнеса. При этом появляется возможность преобразовать всю традиционную цепочку поставки, соединяющую несколько последовательных операций. Взамен создается некоторый треугольник, в котором Интернет помогает образовать динамическое информационное пространство между поставщиком и потребителем. Интернет-коммерция не только позволяет получить огромный выигрыш в производительности, но также обеспечивает непосредственное сотрудничество, повышая гибкость и ответственность всех партнеров. А когда использовать web-системы начинают крупные фирмы, на работу через Интернет начинают переходить десятки тысяч деловых партнеров. Для поддержки web-коммерции необходимо применение технологий интранета и экстранета. Соответствующая сетевая инфраструктура должна поддерживать три типа деловой активности.

Внутрикорпоративный бизнес. В основе внутренней деловой активности лежат технологии интранета. Интранет должен охватывать не только сотрудников центрального офиса, но также филиалы и удаленных пользователей. Ключевое требование состоит в предоставлении удаленным сотрудникам того же уровня обслуживания, что и локальным сотрудникам и в одновременном обеспечении того же уровня безопасности, как во внутренней сетевой инфраструктуре.

Бизнес между производителями. Большинство решений, поддерживающих операции между производителями, построены по технологии безопасного экстранета. В большинстве случаев эту функцию выполняет корпоративный брандмауэр.

Бизнес между производителем и покупателем. Операции между производителем и потребителем представляют собой наиболее заметный тип электронной коммерции (для потребителя), знакомый среднему web-пользователю. Этот тип коммерции основан на ис-

пользовании корпоративного web-сайта, с которым соединяются потребители для совершения покупок и получения поддержки.

Финансовые операции в сети (или, как их принято называть в данном контексте, транзакции) между крупными потребителями и поставщиками уже существуют. Так, успешно используются такие эффективные платежные системы, как EFT (Electronic Funds Transfer – электронный перевод денежных средств), EDI (Electronic Data Interchange – электронный обмен данными). Для установления финансовых отношений существуют готовые стандартные методы проверки кредитоспособности и других параметров транзакций, мало чем отличающихся от методов проверки неэлектронных транзакций, однако транзакции между случайным продавцом и покупателем, не имевших дело друг с другом, ставят новые проблемы.

Интернет представляет собой огромный потенциальный рынок. Число пользователей и компаний, имеющих доступ в Интернет, стремительно растет и все эти пользователи обладают графическими браузерами. Его потенциальные коммерческие возможности очевидны. Сегодня практически вся коммерческая реклама содержит ссылки на web-сайты. Такие ссылки содержатся на упаковке товаров. Однако Интернет-коммерция отличается от традиционной коммерции. Такие достоинства Интернета, как общедоступность, популярность и скорость обслуживания, оборачиваются серьезной проблемой, связанной с потенциальными угрозами, которые обсуждались в предыдущих разделах. Такие крупные фирмы, как AT&T, IBM и Microsoft, на собственном опыте убедились, как, столкнувшись с феноменом всемирной паутины, тщательно выверенные рыночные планы могут стать бесполезными за очень короткое время. И, с другой стороны, жесткая анонимность и недоступность транзакций в финансовой сфере обладает определенной противоречивостью. Отслеживание транзакций бывает необходимо для борьбы с уклонением от налогов, отмыванием преступных денег, выкупом заложников, а также для борьбы с финансированием террористической деятельности.

В реализации Интернет-коммерции существенным являются новые технологические успехи в области связи, шифрования и платежных систем. Наиболее универсальное решение проблемы защиты транзакций заключается в использовании протокола, функционирующего между протоколом транспортного уровня (TCP) и при-

ложением. Наиболее известным применением такого протокола является протокол SSL (Secure Sockets Layer – слой защитных сокетов) и основанный на нем Интернет-стандарт, известный как TLS (Transport Layer Security – безопасность транспортного уровня). При этом возможны два варианта реализации. Для полной универсальности протоколов SSL (стандарт TLS) может быть реализован как часть низлежащего стека протоколов и, таким образом, быть прозрачным для приложений. Альтернативный подход заключается во внедрении SSL в конкретные пакеты. Например, web-браузер Netscape и Microsoft поставляются с встроенной поддержкой SSL, и большая часть web-серверов также поддерживает этот протокол. Работа протокола SSL прозрачна для пользователя. Механизм безопасности реализован поверх базовой службы TCP/IP.

Этот протокол обеспечивает защиту транзакций с помощью различных алгоритмов шифрования как открытым ключом на начальном этапе клиента и сервера (во время процедуры так называемого рукопожатия), так и шифрования симметричным ключом пересылаемых данных.

Очень важной задачей использования Интернета, взаимосвязанной с проблемой защиты транзакций, является задача создания эффективных электронных платежных систем.

Для электронной коммерции было предложено много электронных платежных систем. Среди наиболее распространенных систем является система SET (Secure Electronic Transaction – безопасная электронная транзакция), предложенная компаниями MasterCard и Visa для операции с банковскими картами в Интернете. Эта система позволяет в режиме подключения (on-line) к сети выполнять стандартные банковские операции. Другая система более привлекательная в ряде отношений (однако менее распространенная по определенным причинам) – так называемая система «электронных денег», которую предполагается реализовать в проекте Digit Cash, разработанная Дэвидом Чаумом.

Модель электронной торговли с помощью Master Card и Visa можно представить в виде следующей схемы.

1. Посещение магазина и изучение продаваемых товаров – пользователь ищет интересующий его товар при помощи инструментальных средств (например, браузера), подобно тому, как сегодня используется реклама и телефон.

2. Выбор продавца и товара – с помощью полученной информации владелец банковской карточки выбирает нужный ему товар и продавца.

3. Заказ и переговоры – владелец банковской карточки заполняет электронный бланк заказа и просматривает его для проверки заказанных товаров и цен на них. В этот момент на основании заказанных товаров и других обстоятельств обсуждаются скидки и другие параметры оплаты и стоимости.

4. Выбор платежного средства – владелец банковской карты и продавец договариваются о средствах платежа.

5. Авторизация и передача платежа – владелец банковской карты инициирует процедуру оплаты и с этого момента в дело вступает система SET.

6. Подтверждение и отчет – получив от владельца банковской карты заказ на товар, продавец запрашивает у системы SET авторизацию. Кроме того, продавец посылает владельцу банковской карты подтверждение и отчет о состоянии заказа.

7. Доставка товара – получив авторизацию, продавец отправляет владельцу банковской карты заказанные им товары.

8. Покрытие расходов продавца – отправив товар заказчику, продавец обращается к системе SET, инициируя запрос на возмещение расходов.

В системе SET для защиты конфиденциальности финансовой информации используются как методы шифрования симметричным ключом, так и методы шифрования открытым ключом. Представляется аутентификация с использованием цифровых подписей и гарантии с помощью алгоритмов кэширования.

Как следует из схемы взаимодействия между продавцом и покупателем, кроме этих двух участников существует третья сторона – финансовое учреждение (некоторый посредник), как у покупателя, так и у продавца, которая и осуществляет необходимые финансовые операции. При этом покупатель удален от своих финансовых средств, т.е. наличных денег, что не всегда удобно.

Предложенная Дэвидом Чаумом электронная платежная система исключает недостатки системы SET. Оплата в этой системе осуществляется в автономном режиме, пользователю предоставляется возможность совершить покупки без установки соединения с третьей стороной, представляющей пользователя или продавца для

авторизации или ратификации. В идеале электронные деньги должны обладать рядом качеств, такими, как конфиденциальность, переносимость, делимость и т.п.

В этой системе у каждого пользователя электронных денег есть счет в финансовом учреждении, поддерживающем эту систему. На жестком диске рабочей станции каждого покупателя также есть «бумажник» Digit Cash, содержащий маркеры (электронные «монетки»), вычитаемые из суммы на его счету. Эти маркеры представляют собой пакеты данных, создаваемые алгоритмом обслуживания цифровых денег. Использование электронных денег происходит по следующей схеме: после того как пользователь запускает свою программу электронных денег, она работает на его персональном компьютере в фоновом режиме. В небольшом окне можно видеть, сколько электронных «монеток» хранится на жестком диске пользователя. Это окно оснащено кнопками для обращения к основным функциям системы.

Чтобы использовать электронные деньги для совершения покупок, пользователю нужны электронные монетки, хранящиеся на жестком диске. Перечисление денежного счета в банке на Ваш жесткий диск во многом напоминает работу банкомата. После щелчка на соответствующей кнопке панели инструментов появляется диалоговое окно. В этом окне пользователь указывает количество снимаемых со счета или помещаемых на счет денег.

Деньги могут быть потрачены двумя способами. Оплата может быть потребована кем-то, либо пользователь может инициировать оплату сам. В первом случае пользователь просто объявляет о своем согласии или несогласии платить. Чтобы инициировать оплату, пользователь щелкает мышью на значке оплаты и заполняет формы в диалоговом окне, которые используются для востребования оплаты. На них размещаются поля для ввода, идентификатора счета, денежной суммы, описание и пр.

У получателя денег есть выбор: перевести электронные деньги на свой банковский счет или записать их на свой жесткий диск в виде «монеток». Получатель указывает, что он желает получить оплату, а также в каком виде он хочет ее получить.

Пользователь системы с помощью панели инструментов может вызвать диалоговое окно, отображающее различные параметры счета пользователя, включая оплаченные заказы и полученные платежи.

Чтобы все это работало, а также обеспечивалась безопасность и конфиденциальность обеих сторон, необходимо применение сложных криптографических методов, некоторые из которых были изобретены и запатентованы Дэвидом Чаумом, основателем компании Digit Cash.

В заключение следует отметить, что для электронной коммерции было предложено множество электронных платежных систем. Нет никакого сомнения в том, что большая часть этих предложений окажется несостоятельными, и от них придется отказаться. Останется, вероятнее всего, несколько систем, так как у разных приложений могу быть различные требования.

11.8. Правовое обеспечение информационной безопасности

К правовым мерам обеспечения информационной безопасности относится: разработка норм, устанавливающих ответственность за компьютерные преступления; защита авторских прав программистов; совершенствование уголовного и гражданского законодательства, а также судопроизводства. К правовым мерам следует отнести также общественный контроль за разработчиками компьютерных систем и принятие соответствующих международных соглашений. До недавнего времени в Российской Федерации отсутствовала возможность эффективной борьбы с компьютерными преступлениями, так как данные преступления не могли считаться противозаконными, поскольку они не квалифицировались уголовным законодательством. До 1 января 1997 г. на уровне действующего законодательства России можно было считать удовлетворительно урегулированной лишь охрану авторских прав разработчиков программного обеспечения и, частично, защиту информации в рамках государственной тайны, но не были отражены права граждан на доступ к информации и защита информации, непосредственно связанные с компьютерными преступлениями.

Частично указанные проблемы были решены после введения в действие с 1 января 1997 г. нового Уголовного кодекса (УК), принятого Государственной Думой 24 мая 1996 г. В новом УК ответственность за компьютерные преступления устанавливают ст. ст. 272, 273 и 274. Ст. 272 нового УК устанавливает ответственность за неправомерный доступ к компьютерной информации (на машинном носителе в компьютере или сети компьютеров), если это привело либо к

уничтожению, блокированию, модификации или копированию информации, либо к нарушению работы вычислительной системы. (Под блокированием понимается такое воздействие на компьютер или компьютерную систему, которое повлекло временную или постоянную невозможность выполнять какие-либо операции над информацией.) Эта статья защищает право владельца на неприкосновенность информации в системе. Владельцем информационной системы может быть любое лицо, правомерно пользующееся услугами по обработке информации как собственник вычислительной системы или как лицо, которое приобрело право ее использования. Преступное деяние, ответственность за которое предусмотрено ст. 272, состоит в неправомерном доступе к охраняемой законом компьютерной информации, который всегда имеет характер совершения определенных действий и может выражаться в проникновении в компьютерную систему путем использования специальных технических или программных средств, позволяющих преодолеть установленные системой защиты; незаконного применения действующих паролей или маскировки под законного пользователя для проникновения в компьютер, хищения носителей информации (при условии, что были приняты меры их охраны), если это повлекло к уничтожению или блокированию информации. (Доступ считается правомерным, если он разрешен правообладателем, собственником информации или системы. Неправомерным является доступ, если лицо не имеет права доступа, либо имеет право на доступ, но осуществляет его с нарушением установленного порядка.) Для наступления уголовной ответственности обязательно должна существовать причинная связь между несанкционированным доступом к информации и наступлением предусмотренных ст. 272 последствий, тогда как случайное временное совпадение неправомерного доступа и сбоя в вычислительной системе, повлекшего указанные последствия, не влечет уголовной ответственности.

Неправомерный доступ к компьютерной информации должен осуществляться умышленно, т.е. совершая это преступление, лицо сознает, что неправомерно вторгается в компьютерную систему, предвидит возможность или неизбежность указанных в законе последствий, желает и сознательно допускает их наступление или относится к ним безразлично. Следовательно, с субъективной стороны преступление по ст. 272 характеризуется наличием прямого или кос-

венного умысла. Мотивы и цели данного преступления могут быть самыми разными: корыстными, направленными на причинение вреда (из хулиганских, конкурентных или иных побуждений) или проверку своих профессиональных способностей, и др. Поскольку мотив и цель преступления в ст. 272 не учитываются, она может применяться к всевозможным компьютерным посягательствам.

Ст. 272 УК состоит из двух частей. В первой части наиболее серьезное наказание преступника состоит в лишении свободы сроком до двух лет. Часть вторая указывает в качестве признаков, усиливающих уголовную ответственность, совершение преступления группой лиц или с использованием преступником своего служебного положения, а равно имеющим доступ к информационной системе, и допускает вынесение приговора сроком до пяти лет. При этом не имеет значения местонахождение объекта преступления (например, банка, к информации которого осуществлен неправомерный доступ в преступных целях), который может быть и зарубежным. По уголовному законодательству субъектами компьютерных преступлений могут быть лишь лица, достигшие 16-летнего возраста.

Ст. 272 УК не регулирует ситуации, когда неправомерный доступ к информации происходит по неосторожности, поскольку при расследовании обстоятельств доступа зачастую крайне трудно доказать преступный умысел. Так, при переходах по ссылкам от одного компьютера к другому в сети Интернет, связывающей миллионы компьютеров, можно легко попасть в защищаемую информационную зону какого-либо компьютера, даже не замечая этого (хотя целью могут быть и преступные посягательства).

Ст. 273 УК предусматривает уголовную ответственность за создание, использование и распространение вредоносных программ для компьютеров или модификацию программного обеспечения, заведомо приводящее к несанкционированному уничтожению, блокированию, модификации, копированию информации или к нарушению работы информационных систем. Статья защищает права владельца компьютерной системы на неприкосновенность хранящейся в ней информации. Вредоносными считаются любые программы, специально разработанные для нарушения нормального функционирования других компьютерных программ. Под нормальным функционированием понимается выполнение операций, для которых эти программы предназначены и которые определены

в документации на программу. Наиболее распространенные вредоносные программы – компьютерные вирусы, логические бомбы, а также программы, известные как «троянский конь». Для привлечения к ответственности по ст. 273 необязательно наступление каких-либо нежелательных последствий для владельца информации, достаточен сам факт создания вредоносных программ или внесение изменений в уже существующие программы, заведомо приводящих к указанным в статье последствиям. Использованием программ считается их выпуск в свет, воспроизведение, распространение и другие действия по введению в оборот. Использование программ может осуществляться путем записи в память компьютера или на материальный носитель, распространения по сетям или путем иной передачи другим пользователям.

Уголовная ответственность по ст. 273 возникает уже в результате создания вредоносных программ, независимо от их фактического использования. Даже наличие исходных текстов программ является основанием для привлечения к ответственности. Исключение составляет деятельность организаций, разрабатывающих средства противодействия вредоносным программам и имеющих соответствующие лицензии. Статья состоит из двух частей, различающихся признаком отношения преступника к совершаемым действиям. Ч. 1 предусматривает преступления, совершенные умышленно, с сознанием того, что создание, использование или распространение вредоносных программ заведомо должно привести к нарушению неприкосновенности информации. При этом ответственность наступает независимо от целей и мотивов посягательства, которые могут быть вполне позитивными (например, охрана личных прав граждан, борьба с техногенными опасностями, защита окружающей среды и т.п.). Максимальное наказание по первой части – лишение свободы сроком до трех лет. По ч. 2 дополнительный квалифицирующий признак – наступление тяжких последствий по неосторожности. В этом случае лицо сознает, что создает, использует или распространяет вредоносную программу или ее носители и предвидит возможность наступления серьезных последствий, но без достаточных оснований рассчитывает их предотвратить, или не предвидит этих последствий, хотя как высококвалифицированный программист мог и был обязан их предусмотреть. Поскольку последствия могут быть очень тяжкими (смерть или вред здоровью человека, опасность военной или

иной катастрофы, транспортные происшествия), максимальное наказание по ч. 2 – семь лет лишения свободы.

Отметим, что в законе не говорится о степени нанесенного вреда в отличие от краж, когда различаются просто кража, кража в крупном размере и кража в особо крупном размере. Здесь устанавливается лишь факт преступления, а размер ущерба влияет лишь на оценку его тяжести и меру ответственности.

Наконец, ст. 274 УК устанавливает ответственность за нарушение правил эксплуатации компьютеров, компьютерных систем или сетей лицом, имеющим доступ к ним, повлекшее уничтожение, блокирование или модификацию охраняемой законом информации, если это деяние причинило существенный вред. Эта статья защищает интересы владельца компьютерной системы в отношении ее правильной эксплуатации и распространяется только на локальные вычислительные сети организаций. К глобальным вычислительным сетям, например к таким, как Интернет, эта статья неприменима. Под охраняемой законом информацией понимается информация, для которой в специальных законах установлен режим ее правовой защиты. Между фактом нарушения правил эксплуатации и наступившим существенным вредом должна быть обязательно установлена причинная связь и полностью доказано, что наступившие вредные последствия являются результатом именно нарушения правил. Оценку нанесенного вреда устанавливает суд, исходя из обстоятельств дела, причем считается, что существенный вред менее значителен, чем тяжкие последствия.

Субъект этой статьи – лицо, в силу своих должностных обязанностей имеющее доступ к компьютерной системе и обязанное соблюдать установленные для них технические правила. Согласно ч. 1 статьи он должен совершать свои деяния умышленно; сознавать, что нарушает правила эксплуатации; предвидеть возможность или неизбежность неправомерного воздействия на информацию и причинение существенного вреда, желать или сознательно допускать причинение такого вреда или относиться к его наступлению безразлично. Наиболее строгое наказание в этом случае – лишение права занимать определенные должности или заниматься определенной деятельностью на срок до пяти лет, или ограничение свободы на срок до двух лет. Ч. 2 ст. 274 предусматривает ответственность за те же деяния, не имевшие умысла, но повлекшие по неос-

торожности тяжкие последствия, например за установку инфицированной программы без антивирусной проверки, что повлекло за собой серьезные последствия (крупный финансовый ущерб, транспортные происшествия, утрата важных архивов, нарушение работы системы жизнеобеспечения в больнице и др.). Мера наказания за это преступление устанавливается судом в зависимости от наступивших последствий, максимальное наказание – лишение свободы на срок до четырех лет.

Как видно, рассмотренные статьи УК не охватывают все виды компьютерных преступлений, разнообразие которых увеличивается вместе с прогрессом в области компьютерной техники и ее использованием. Кроме того, некоторые формулировки статей допускают неоднозначное истолкование, например в определении злостного умысла. Поэтому в дальнейшем возможно пополнение и усовершенствование этих статей.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. В чем причины, связанные с необходимостью разработки новых методов защиты информации и дальнейшему развитию традиционных? На каких направлениях использования информационных систем эти методы особенно актуальны?

2. Какие непреднамеренные воздействия могут привести к потере информации, и каким образом осуществляется резервное копирование? Расскажите о защите от помех в электросетях. Что такое избыточность данных как метода защиты информации от непреднамеренных воздействий?

3. Определите, что означают следующие требования к методам защиты информации: целостность данных; доступность данных; секретность и конфедициальность; авторизация; аутентификация; запрещение повторного использования?

4. Какие методы защиты от локальных атак знаете?

5. Что такое криптография? Опишите традиционную схему шифрования открытым ключом. Назовите основные методы атаки на традиционную схему.

6. Расскажите, что такое электронная подпись. Назовите основные причины необходимости в ее специальной защите. Перечислите основные методы, применяемые для этих целей. Расскажите о защитной функции хэширования.

ГЛАВА 12. БЕСПРОВОДНИЕ СИСТЕМЫ И СЕТИ ПЕРЕДАЧИ ДАННЫХ

12.1. Беспроводные среды

По мере того как электронные информационные системы затрагивают все большие аспекты деятельности человека, необходимость привязываться к этим системам проводами в ряде случаев значительно ограничивает сферу их применения.

Беспроводная связь становится необходимой в следующих случаях:

требуется мобильная связь;

необходимо установить связь в местности, где прокладка кабеля сложна или совсем невозможна;

коммуникационные системы требуют оперативной установки;

требуются минимальные затраты на установку коммуникационной системы;

требуются широковещательная рассылка одной и той же информации большому количеству получателей;

имеется интенсивное развитие беспроводных средств связи (сотовая, спутниковая и т.п.) и массовое внедрение ее среди широкого круга потребителей, в частности мобильной телефонии, которая явилась предпосылкой использования их в системах передачи данных и вычислительных сетях.

Справедливости ради необходимо отметить, что по сравнению с проводной связью беспроводная, на ряду с очевидными преимуществами, обладает и определенными недостатками:

беспроводная связь функционирует не в столь контролируемой среде, как проводная и по этому она в большей степени подвержена потерям сигнала, шумам и подслушиванию;

скорость передачи данных в беспроводных системах меньше, чем в проводных;

в случае с проводной связью значительно проще многократно использовать несущие частоты, чем в случае беспроводной.

Однако преимущества беспроводной связи в большинстве случаев значительно превышают ее ограничения, на что указывает практика ее внедрения в системах передачи данных, включая вычислительные сети.

Можно выделить две категории беспроводных сетей – беспроводные локальные вычислительные сети (ЛВС) и беспроводные глобальные вычислительные сети (ГВС). Они существенно различаются как по функциональным возможностям, так и по используемым программно-техническим средствам. Хотя основная часть данной главы будет посвящена беспроводным ГВС, для полноты картины необходимо немного остановиться и на ЛВС.

В беспроводных ЛВС используется два типа средств связи. Большая часть этих сетей использует радиосвязь (в различном, но достаточно ограниченном диапазоне частот). Однако свою экологическую нишу имеют и сети, использующие для передачи информации инфракрасное излучение. Этот вид связи в ряде случаев имеет определенные преимущества перед радиосвязью. Такие каналы имеют высокую пропускную способность, обеспечивают высокую степень защиты информации, позволяют избежать сложностей, связанных с защитой организма человека от нежелательного воздействия радиоволн, к тому же организация и технические средства такой связи имеют невысокую стоимость. Однако ограниченный радиус действия (примерно 24,5 м), низкая надежность и неспособность проникать сквозь непрозрачные объекты ограничивают применение этого вида связи.

Значительно большее распространение в ЛВС получили средства радиосвязи. Радиус действия аппаратуры, используемой в этих сетях и выпускаемой различными фирмами, изменяется от 25 до 30000 м. Различия в радиусе действия связаны с совершенно различными техническими решениями, ориентированными на различные структуры ЛВС. Возможны две схемы организации радиосвязи в таких сетях – двухточечная (point-to-point), предназначенная для связи на значительные расстояния между удаленными группами локальных сетей, и многоточечная схема (point-to-multipoint) для организации различных протоколов локальных сетей (Ethernet, Token Ring).

Одной из фирм (отнюдь, не единственной), занимающейся производством средств для локальных компьютерных радиосетей, является всемирно известная Motorola. Она предлагает различные продукты, отличающиеся дальностью взаимодействия. Они предназначаются как для передачи данных внутри здания (point-to-multipoint), так и между зданиями в зоне прямой видимости на частоте 18 ГГц (point-to-point). При этом для поддержки устойчивой

радиосвязи используется принцип сотовой телефонии, основанный на использовании ретрансляторов. Радиус действия двухточечной связи доходит до 2,1 км. Наиболее продвинутые продукты, присутствующие на рынке беспроводных сетей, до последнего времени принадлежали компании Windata.

Когда беспроводные сети только появились, возникла некоторая эйфория – казалось, что они смогут заменить кабельные сети во всех случаях жизни. Вскоре выяснилось, что это совсем не так, поскольку они (на начальном этапе своего развития) были существенно дороже и медленнее кабельных, и это направление ЛВС чуть было не потерпело фиаско. Потом пришло понимание, что радиосвязь и кабельная сеть вполне могут сосуществовать, у них есть свои независимые экологические ниши.

Однако значительно большее влияние оказали беспроводные средства связи на развитие ГВС, являясь важной составной частью этих сетей. В связи с этим в дальнейшем рассматриваются различные аспекты беспроводной связи применительно к такого рода вычислительным сетям.

Передача данных в беспроводных средах достигается с помощью антенн: работая на передачу, антенна излучает электромагнитную энергию в среду распространения. При приеме антенна получает электромагнитную энергию из окружающей среды.

Тип среды, распространение радиоволн, определяет тип приемопередающих устройств и технические характеристики радиолиний. Можно выделить две основные среды, используемые в радиолиниях – атмосфера Земли и космическое пространство. Атмосфера Земли, с точки зрения передачи электромагнитных волн, также неоднородна. С определенной степенью условности она разделяется на три слоя: тропосферу с высотой слоя 10 – 12 км, стратосферу от 10 до 50 км и ионосферу от 50 до 400 км. Около 80 % массы атмосферы сосредоточено в тропосфере и около 20 % в стратосфере. Плотность атмосферы в ионосфере крайне мала, граница между ионосферой и космическим пространством является условным понятием, так как следы атмосферы встречаются даже на высотах более 400 км. Считается, что плотные слои атмосферы заканчиваются на высоте около 120 км.

Для передачи во всех перечисленных средах используются электромагнитные волны в очень широком диапазоне частот (длин

волн). В соответствии с классификацией, установленной международным регламентом связи, выделяются следующие диапазоны:

сверхдлинные (мириаметровые) или очень низкие частоты (ОНЧ) – 10 – 100 км, 3 – 30 кГц;

длинные (километровые) или низкие частоты (НЧ) – 1 – 10 км, 30 – 300 кГц;

средние (гектометровые) или средние частоты (СЧ) – 100 – 1000 м, 300 – 3000 кГц;

короткие (декаметровые) или высокие частоты (ВЧ) – 10 – 100 м, 3 – 30 МГц;

ультракороткие (метровые) или очень высокие частоты (ОВЧ) – 1 – 10 м, 30 – 300 МГц;

дециметровые или ультравысокие частоты (УВЧ) – 10 – 100 см, 300 – 3000 МГц;

сантиметровые или сверхвысокие частоты (СВЧ) – 1 – 10 см, 3 – 30 ГГц;

миллиметровые или крайне высокие частоты (КВЧ) – 1 – 10 мм, 30 – 300 ГГц;

децимиллиметровые или гипервысокие частоты (ГВЧ) – 0,1 – 1 мм, 300 – 3000 ГГц.

Иногда диапазон 30 МГц – 1 ГГц называют радиодиапазоном, а частоты в диапазоне от 2 до 40 ГГц (1 ГГц = 10^9 Гц) – микроволновым диапазоном.

Если в проводных системах передачи сигнал распространяется строго в направлении прокладки кабеля, то в беспроводных системах ситуация значительно отличается.

Различают два типа беспроводной связи: однонаправленную и всенаправленную. В однонаправленных системах сигнал распространяется в одном направлении, передающая система излучает однонаправленный электромагнитный луч. В связи с этим приемник и передающая система должны находиться строго на одной прямой. При всенаправленной передаче сигнал распространяется во всех направлениях, его можно принять любыми системами. В общем случае с ростом частоты сигнала его легче сфокусировать в однонаправленный луч.

Радиоволны, излучаемые передающей антенной, прежде чем попасть в приемную антенну, проходят в общем случае сложный

путь. На значение напряженности поля в точке приема оказывает влияние множества факторов. Основные у них следующие:

отражение электромагнитных волн от поверхности Земли;

преломление (отражение) в ионизированных слоях атмосферы (ионосфере);

рассеивание на диэлектрических неоднородностях нижних слоев атмосферы (тропосферы);

дифракция сферической выпуклости Земли.

Кроме того, напряженность поля в точке приема зависит от длины волны и освещенности земной атмосферы Солнцем и других факторов.

Как и в любой другой системе передачи, основным источником потерь в беспроводных системах является затухание. Однако потери, к примеру, в микроволновых беспроводных системах меньше, чем при передаче по витой паре и коаксиальному кабелю (при этих же частотах). Это позволяет увеличить расстояние между усилителями или повторителями, обычно расстояние для этого диапазона длин волн составляет 10 – 100 км. Правда, затухание возрастает во время дождя. Влияние дождя заметно при частотах выше 10 ГГц. Другой источник нарушений – помехи. С ростом каналов передачи возникает перекрытие зон передачи. В этом случае есть опасность помех, поэтому распределение частотных диапазонов должно строго контролироваться.

12.2. Системы радиосвязи

Существует довольно большое количество разновидностей радиосвязи: *радиорелейные прямой видимости и тропосферные, спутниковые, различных уровней, ионосферные* и т.п. Однако все эти разновидности имеют в своей структуре функционально одинаковые устройства. Для обеспечения односторонней связи в пункте, из которого ведется передача сигнала, размещается радиопередающее устройство, содержащее радиопередатчик и передающую антенну, а в пункте, в котором ведется прием сигналов, – радиоприемное устройство, содержащее приемную антенну и радиоприемник. Антенны подключаются к приемопередающему устройству при помощи фидерных трактов.

Для двухстороннего обмена сигналами нужно иметь два комплекта оборудования. Двухсторонняя связь может быть *симплексной и дуплексной*. При симплексной связи прием и передача ведутся поочередно и оба передатчика работают на одной частоте. При дуплексной радиосвязи передача и прием осуществляются одновременно на разных частотах.

В качестве функциональных узлов в состав радиопередатчика входят генератор несущей частоты и модулятор, а также много другого оборудования: источники питания, средства охлаждения, автоматического и дистанционного управления, сигнализации, защиты, блокировки и т.п. По назначению радиопередающие устройства делятся на *маломощные* (до 100 Вт), *средней мощности* (от 100 до 10000 Вт), *мощные* (от 10 до 500 кВт) и *сверхмощные* (свыше 500 кВт).

Задачи радиоприемного устройства сводятся к выделению полезного радиосигнала (очень малой мощности) из множества других сигналов и возможных помех, а также воспроизведению (восстановлению) передаваемого сообщения. Основными показателями радиоприемных устройств являются: диапазон рабочих частот, в пределах которых устройство может работать; чувствительность (мера способности обеспечивать прием слабых сигналов); избирательность (способность устройства отличить полезный сигнал от радиопомехи); помехоустойчивость (противодействие мешающему действию помех) и т.п. По назначению радиоприемные устройства подразделяются на *радиовещательные* (обычно называются радиоприемниками), *телевизионные, профессиональные* (магистральные радиоприемные устройства), *специальные* (радиолокационные, навигационные, самолетные) и т.д.

Особую роль в радиосвязи занимают антенно-фидерные устройства. Антенна представляет собой элемент сопряжения между передающим или приемным оборудованием и средой распространения радиоволн. Имеется несколько разновидностей антенн, применяемых в различных диапазонах длин волн. Тип антенны также зависит от направления применения приемопередающих устройств. Антенны, состоящие из проводов небольшого поперечного сечения по сравнению с длиной и продольными размерами (кратными длине волны), называются проволочными и используются для длинноволновой части электромагнитного излучения. Антенны, имеющие вид

поверхностей и излучающие через свой раскрыв – апертуру, называются апертурными. Иногда их называют дифракционными, рефлекторными, зеркальными. Электрические токи таких антенн протекают по проводящим поверхностям, имеющим размеры, соизмеримые с длиной волны или много большие ее. Такие антенны применяются для коротковолновой части электромагнитного излучения.

Среди наиболее важных характеристик антенны следует отметить: входное сопротивление, определяемое, как отношение напряжения высокой частоты на ее зажимах к току питания; коэффициент полезного действия – отношение мощности, излучаемой антенной, к мощности, подводимой к ней. Не вся мощность, подводимая к антенне, излучается в окружающее пространство, часть ее расходуется как на нагревание самой антенны, так и на находящиеся вблизи предметы. Электромагнитные волны излучаются антенной в различных направлениях неравномерно. Создание антенн, излучающих электромагнитные волны равномерно во всех направлениях, крайне сложно и практически недостижимо. Распределение в пространстве напряженности электромагнитного поля, создаваемого антенной, определяется очень важным фактором – амплитудной характеристикой направленности. Различные применения антенн предполагают прямо противоположные требования к диаграммам направленности. Так, для организации сети звукового вещания используются кило- и гектометровые радиоволны, которые распространяются с помощью антенн, устанавливаемых в центре зоны обслуживания, и поэтому должны создавать ненаправленное излучение вдоль поверхности Земли, т.е. иметь диаграмму направленности в горизонтальной плоскости в виде окружности. Таким условиям отвечают антенны-мачты, антенны-башни. Их высота обычно 150 – 250 м, а в некоторых случаях достигают 350 – 500 м, что значительно увеличивает зону обслуживания.

Для радиосвязи и радиовещания на значительные расстояния (тысячи километров) используются дециметровые волны. Особенность их распространения такова, что антенны должны сформировать направленное излучение с максимумом излучения под некоторым углом к поверхности Земли. Самыми распространенными типами передающих антенн, отвечающих этим требованиям, являются проволочные антенны: вибраторные, ромбические, синфазные в виде решетки из вибраторов, возбужденных определенным образом.

Диапазон метровых радиоволн используется главным образом для организации телевизионного и звукового вещания, а также для связи с подвижными объектами в пределах определенной зоны обслуживания. Передающие антенны, как правило, должны создавать ненаправленное излучение в горизонтальной плоскости. С другой стороны, в диапазоне деци-, сантиметровых и более коротких волн, применяемых для организации так называемой радиорелейной связи, антенны, устанавливаемые на линиях, должны обладать высокой направленностью, их диаграммы направленности должны иметь «игольчатую форму». Наиболее распространены в этом диапазоне апертурные (зеркальные) антенны.

Антенна – устройство обратимое. Если антенна хорошо излучает радиоволны, то она хорошо их и принимает. В связи с этим приемные антенны аналогичны передающим для соответствующих диапазонов длин волн.

Электрическая цепь и вспомогательные устройства, с помощью которых энергия радиочастотного сигнала подводится от радиопередатчика к антенне или от антенны к радиоприемнику, называются фидером. Передающие антенны длинноволнового диапазона радиоволн соединяются с радиопередатчиком с помощью многопроводных коаксиальных фидеров, в диапазоне метровых волн используется коаксиальный кабель. На более коротких волнах фидер выполняется в виде полой металлической трубы-волновода прямоугольного, эллиптического или круглого сечения.

Рассмотрим теперь основные типы каналов, осуществляющих связь на большие расстояния и располагающихся в земной атмосфере (спутниковая связь будет рассмотрена отдельно).

Основным видом передачи в этом случае является так называемая *радиорелейная система передачи*, в которой сигнал электросвязи передается с помощью наземных ретрансляционных станций. На частотах ОВЧ- и СВЧ-диапазонов надежная радиорелейная связь с низким уровнем помех может быть получена только в условиях прямой видимости между антеннами, излучающими радиоволны. Типичные расстояния составляют 40 – 50 км при высотах башен и мачт, на которых устанавливаются антенны, около 100 м. При этом для волн указанного диапазона частот верхние слои атмосферы прозрачны и не могут использоваться для передачи. Невозможность свободного распространения радиоволн на большие

расстояния позволяет избежать взаимных помех между радиорелейными системами внутри одной страны и разных стран. Кроме того, в указанных диапазонах практически отсутствуют атмосферные и промышленные помехи. Антенны могут работать в режиме передачи и приема, причем для одновременной передачи в противоположных направлениях с использованием двух частот. Аналоговые радиорелейные системы этих диапазонов частот предназначены, в основном, для передачи многоканальных телефонных сигналов в аналоговой форме и данных с низкой и средней скоростями, а также сигналов телевидения. Цифровые радиорелейные системы используются для организации цифровых трактов передачи сигналов со скоростями от 2 до 155 Мбит/с. Большинство станций радиорелейных систем являются промежуточными радиостанциями, играющими роль активных ретрансляторов.

Существует довольно большое количество отечественных как аналоговых, так и цифровых радиорелейных систем передачи, использующих различные частоты указанного диапазона. Число каналов в различных типах систем лежит в диапазоне 300 – 1920 для аналоговых систем и от 15 – 240 для цифровых.

Тропосферные радиорелейные системы передачи связаны с волнами диапазона 0,3 – 5 ГГц, которые способны рассеиваться на локальных неоднородностях, вызванных различными физическими процессами, происходящими в этой нижней части атмосферы (рис. 12.1).

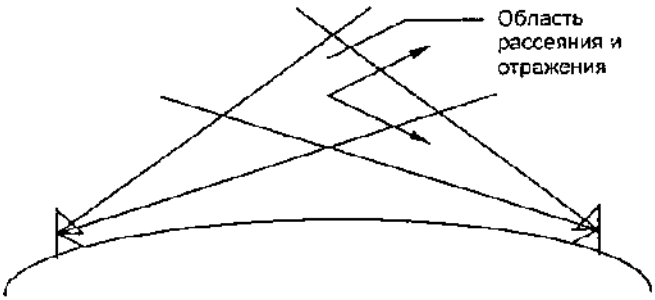


Рис. 12.1. Иллюстрация тропосферной радиосвязи

Рассеянные волны могут быть приняты направленной приемной антенной на значительном удалении от передающей, учитывая, что неоднородности находятся на значительной высоте. Это дает возможность разнести станции на расстояния 200 – 400 км друг от друга, что значительно больше расстояния прямой видимости. Как отмечалось раньше, для линий выше рассмотренного диапазона частот (прямой видимости) тропосфера прозрачна.

Линии на основе радиорелейных систем передачи строятся, как правило, в трудно доступных и удаленных районах.

Значительное расстояние между станциями, безусловно, выгодно при организации протяженных линий, поскольку требуется меньшее число станций. Однако за счет глубоких замираний из-за неустойчивости пространственно-временной структуры тропосферы и крайне малой мощности радиосигнала в точке приема организация хорошего качества каналов затруднительна. Количество одновременно работающих каналов для сравнительно малого ассортимента аппаратуры, обеспечивающих этот вид передачи, невелико (от 12 до 120).

Существуют еще две системы передачи, обеспечивающие еще большую протяженность от приемника до передатчика, так как в качестве среды передачи они используют ионосферу – самую удаленную часть атмосферы.

Так, для декаметровой волны (3 – 30 МГц) происходит поворот ее траектории из-за неоднородностей диэлектрических свойств профиля ионосферы (в отличие от рассеяния волн в тропосфере на частотах 3 – 5 ГГц). Траектория распространения радиоволн от одной точки поверхности Земли к другой с одним отражением от ионосферы называется ионосферным скачком. Расстояние между пунктами приема и передачи, измеренное вдоль поверхности Земли, составляет около 2000 км. Траектория распространения радиоволн может быть образована несколькими ионосферными скачками. Условия распространения радиоволн, а следовательно, и качество радиосвязи зависят от состояния ионосферы, определяемого временем года, суток и циклом солнечной активности, что не позволяет организовать больше одного-двух телефонных или нескольких телеграфных каналов.

Метровые волны (40 – 70 МГц) могут также использовать ионосферу в качестве среды передачи, правда, в отличие от предыдуще-

го случая метровые волны используют рассеяние на неоднородностях ионосферы. Это аналогично тропосферной радиорелейной передаче, но на гораздо большей высоте. Придельная дальность связи в этом случае составляет 2000 – 3000 км. При ионосферном рассеивании в пункт приема приходит ничтожная часть излучаемой энергии, что вынуждает использовать мощные радиопередатчики и большие по размеру антенны. Рассеяние волн происходит на ионизированных следах, сгораемых в ионосфере мелких космических частиц – метеоров, непрерывно проникающих в атмосферу Земли. Время прохождения радиосигналов при метеорной связи составляет только 2 – 4 ч в сутки. Такие системы позволяют организовать с удовлетворительным качеством не более трех телефонных каналов. Обычно с помощью таких радиосистем организуется передача телеграфных сигналов, причем таких, для которых задержка не играет существенной роли. Они используются для дублирования ионосферных систем на декаметровых волнах в полярных широтах.

12.3. Спутниковая система связи

23 апреля 1965 г. был запущен на высокую эллиптическую орбиту спутник связи «Молния 1», который ознаменовал становление в нашей стране спутниковой радиосвязи. Почти одновременно в США был запущен на геостационарную орбиту первый спутник коммерческой связи Intelsat-1. Таким образом, была реализована заманчивая идея резкого увеличения дальности радиосвязи благодаря размещению ретранслятора высоко над поверхностью Земли, что позволило обеспечить одновременную радиовидимость (по аналогии с прямой радиовидимостью наземных радиорелейных ретрансляторов, обеспечивающих связь всего на 50 км) радиостанций, расположенных в разных точках обширной территории.

Спутник связи представляет по существу микроволновую ретрансляционную станцию. Он используется для связи двух или более наземных передатчиков-приемников, называемых земными, или наземными, станциями. Спутник получает сигнал в одной полосе частоты (восходящая линия), усиливает или повторяет сигнал и передает его на другой частоте (нисходящая линия). Одна орбитальная станция может работать с рядом частотных диапазонов,

называемых каналами транспондера или просто транспондерами. На рис. 12.2 изображены две спутниковые системы связи.

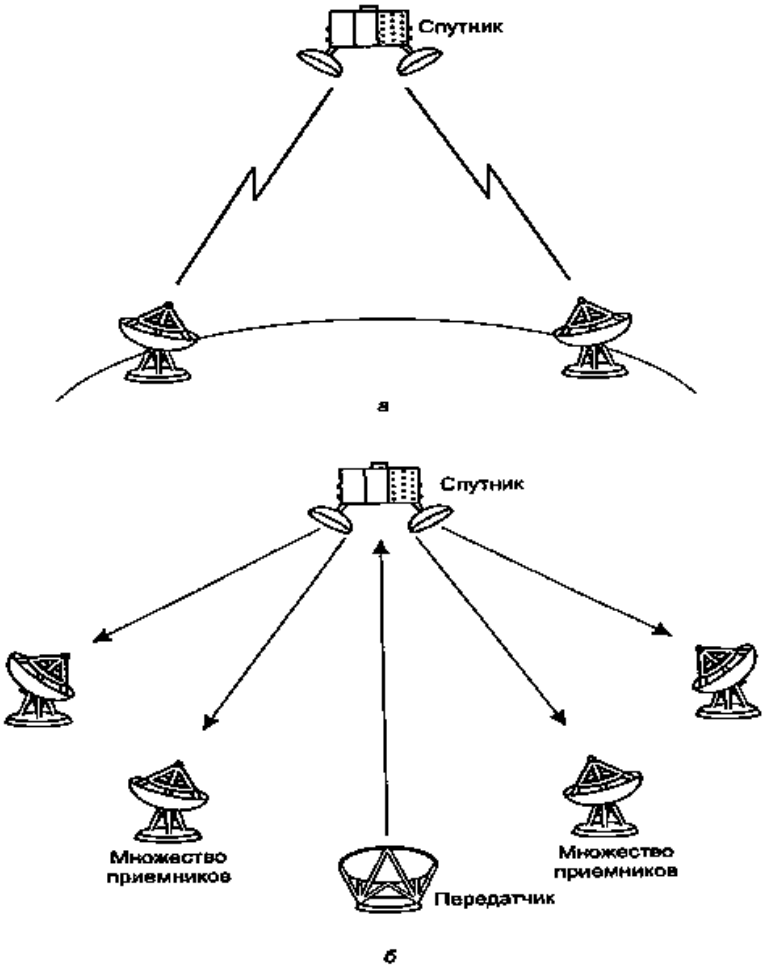


Рис. 12.2. Системы спутниковой связи: *а* – двухточечный спутниковый канал связи; *б* – широковещательная рассылка через спутник

В первой из них спутник применяется для двухточечной связи между наземными станциями, во второй спутник используется для передачи от наземной антенны к нескольким наземным приемникам (так называемая широковещательная рассылка).

Оптимальный диапазон спутниковой передачи составляет от 1 до 10 ГГц. На частоте ниже 1 ГГц существенное влияние оказывает шум от естественных источников (галактико-солнечный, земной, атмосферный шумы), а также искусственные помехи от различных электронных устройств. При частотах свыше 10 ГГц сигнал значительно затухает из-за атмосферного поглощения и осадков.

Спутник связи – революционная технология, сравнимая по важности с оптическим волокном. Среди наиболее важных применений спутниковой связи можно указать: передачу телевизионных сигналов, междугороднюю-международную связь, частные коммерческие сети.

Конфигурация систем спутниковой связи зависит от типа искусственного спутника Земли, вида связи и параметров земных станций.

Основной характеристикой спутника является орбита, на которой он находится. По степени удаленности от Земли они различаются на геостационарные – 35838 км, низкие околоземные орбиты – от 320 до 1100 км и средние околоземные – 8000 – 1200 км (рис. 12.3).

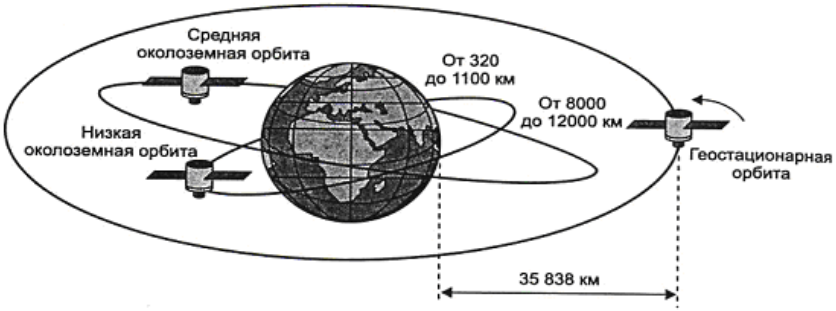


Рис. 12.3. Орбиты спутников (не в масштабе)

Еще одной разновидностью орбит, используемых спутниками связи «Молния», в определенной степени оптимальных в обеспечении надежной коммуникации в России, является высокая эллиптическая орбита с периодом обращения 12 ч, наклонением 63°, высотой апогея в северном полушарии в 40000 км.

Движение спутника в области апогея замедляется, при этом длительность радиовидимости составляет 6 – 8 ч. Преимуществом данного типа орбит является большой размер зоны обслуживания при охвате большей части северного полушария. К недостаткам можно отнести необходимость слежения антенн за медленно дрейфующим спутником и их переориентированию с заходящего спутника на восходящий.

Уникальной является геостационарная орбита – круговая орбита, расположенная в плоскости экватора с периодом обращения 24 ч. Орбита синхронна с вращением Земли, поэтому спутник оказывается неподвижным относительно земной поверхности. Безусловным достоинством геостационарных спутников является большая зона обслуживания, составляющая около трети земной поверхности. Трех спутников достаточно для обеспечения почти глобальной связи, антенны на земных станциях не требуют системы слежения. Однако в северных широтах спутник виден под малым углом к горизонту и вовсе не виден в приполярных областях. К тому же в виду большой удаленности спутника сигнал сильно ослабевает и имеет вполне ощутимую, даже при телефонных разговорах, задержку (до 0,5 с) при прохождении в оба конца.

Большую популярность приобрели спутники, использующие низковысотные круговые орбиты, плоскости которых наклонены (под разными углами) к плоскости экватора и имеющие ряд важных достоинств. Запуск таких спутников осуществляется с недорогих пусковых установок, нет проблем с ослаблением сигнала (что важно для мобильных и персональных терминалов) и его задержкой. Однако скорость вращения спутника относительно поверхности Земли достаточно велика: в результате длительность сеанса от восхода спутника до его захода не превышает нескольких десятков минут, поэтому требуется много спутников для глобальной связи. Так, компанией Motorolla была предложена в 1987 г. система Iridium, которая была введена в строй в 1988 г. и состояла из 66 спутников (системой Iridium владеет международная консорциум Iridium Inc). Правда, малая зона обслуживания одним спутником имеет и положительную сторону, поскольку позволяет экономить частотный диапазон (зона связи отдельных спутников находится на значительном удалении, что позволяет повторно использовать одни и те же частоты для различных абонентов, а это значительно по-

вышает эффективность использования каналов связи). К числу недостатков низкоорбитальных спутников следует отнести влияние эффекта Доплера – изменение частоты принимаемого сигнала от скорости движущегося объекта.

В зависимости от назначения системы спутниковой связи и типа наземных станций регламентом Международного союза электросвязи различаются следующие службы:

фиксированная спутниковая служба для связи между станциями, расположенными в определенных фиксированных пунктах, а также распределение телевизионных программ;

подвижная спутниковая служба для связи между подвижными станциями, размещенными на транспортных средствах (самолетах, морских судах, автомобилях и т.д.);

радиовещательная спутниковая служба для непосредственной передачи радио- и телевизионных программ на терминалы, находящиеся у абонентов.

На начальном этапе фиксированная спутниковая служба развивалась в направлении создании систем магистральной связи с применением крупных наземных станций с диаметром зеркала антенн порядка 12 – 30 м. В настоящее время функционирует более 50 таких систем. Это отечественные системы «Молния 3», «Радуга», «Горизонт». Международные системы: Intelsat и Eutelsat. Развитие таких систем идет в направлении увеличения срока службы искусственных спутников, повышения точности их удержания на орбите, разработки и совершенствования многолучевых антенн, а также возможности работы на антенны земных станций малого диаметра (1,2 – 2,4 м) – система VSAT.

В силу международного характера работы транспорта для его управления создаются международные системы глобальной спутниковой связи, например система морской спутниковой связи Inmarsat, которая введена в действие с 1982 г. Она содержит геостационарные спутники, расположенные над Атлантическим, Индийским и Тихим океанами; береговые станции, установленные на различных континентах; разветвленную сеть судовых станций различных стандартов. В настоящее время системой Inmarsat пользуются более 15 тыс. судов. В рамках организации Inmarsat решаются проблемы создания системы авиационной спутниковой связи.

В последнее время в связи с технологическими успехами в микроэлектронике, разработке лазерных линий межспутниковой связи и т.п. большой интерес проявляется к использованию легких низколетящих искусственных спутников Земли, что оказывается экономически целесообразно. Примером такой системы служит, упоминавшийся ранее международный проект Iridium, возглавляемый фирмой Motorola. Эту систему предполагалось использовать и для поддержки системы сотовой связи стандарта GSM. Однако пока эта система не имеет коммерческого успеха, и ее существование остается под вопросом.

Международный телекоммуникационный союз утвердил международный стандарт спутникового TV вещания в диапазоне 12 ГГц (НТВ-12). В планах зафиксированы точки стояния спутников, номера частотных каналов, параметров бортовой передающей аппаратуры для различных стран. Так, для спутников бывшего СССР выделено несколько точек стояния: 22°; 44°; 74° и 140° восточной долготы. Для непосредственного телевизионного вещания используется более 100 спутников, среди которых TV-SAT-1; TV-SAT-2; TDF-2; TELEX и др.

Для широкого внедрения непосредственного телевизионного вещания требуются многопрограммные спутники с несколькими десятками программ с тем, чтобы, приобретая сравнительно дорогое приемное оборудование, абонент (зритель) мог бы значительно расширить свой телевизионный выбор. В этой связи актуальны работы в области цифрового сжатия телевизионных изображений, позволяющие передавать в одном частотном стволе до 6 – 10 программ одновременно.

12.4. Системы подвижной радиосвязи

В настоящее время доминирующее положение на рынке подвижной радиосвязи занимают:

профессиональные (частные) системы подвижной радиосвязи (PMR – Professional Mobil Radio, PAMR – Public Access Mobile Radio);

системы персонального вызова (Paging System);

системы сотовой подвижной радиосвязи (Cellar Radio System);

системы беспроводных телефонов.

Безусловно, наибольший интерес представляют системы сотовой связи, которые буквально произвели революцию в телефонии, приобрели массовый характер, активно включаясь в системы передачи данных и ГВС. Однако чтобы понять причины появления этих систем и их принципиальные особенности, необходимо, хотя бы кратко, остановиться на профессиональных (частных) системах подвижной связи, которые, можно сказать, были прародителями сотовых систем, поскольку исторически они появились первыми. Часть этих систем, обеспечивающих взаимодействия с телефонными сетями общего пользования, получила названия частных (PAMR), а не обеспечивающих такого взаимодействия – профессиональных (PMR), т.е. обеспечивающих связью замкнутую группу абонентов.

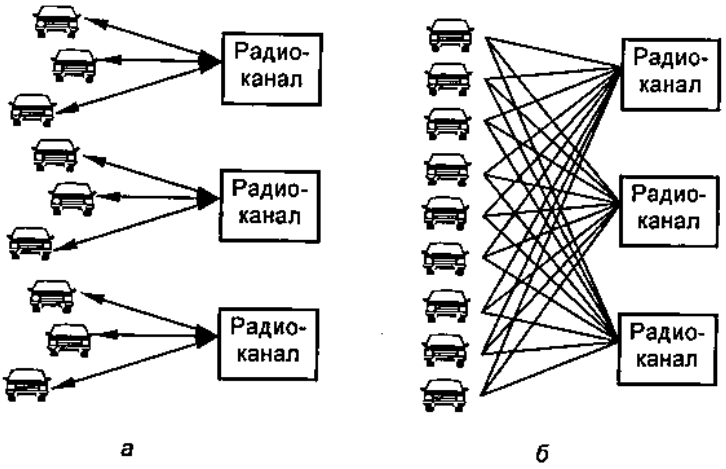


Рис. 12.4. Структура профессиональных (частных) систем подвижной радиосвязи

В первых профессиональных системах передатчик и приемник проектировались для работы на определенной фиксированной частоте, каждый радиоканал был закреплен за сравнительно небольшой группой абонентов (рис. 12.4, *а*). Если число абонентов превышало возможности одного канала, образовывали другую группу, за которой закрепляли другой радиоканал.

В системе с общедоступным пучком каналов (транкинговые системы – рис. 12.4, *б*) всем абонентам доступна группа каналов. При поступлении вызова за парой абонентов закрепляется один из свободных каналов. После отбоя канал освобождается и может быть предоставлен любой другой паре абонентов, технически это реализуется различными способами.

Пропускная способность системы с общедоступным пучком каналов существенно выше пропускной способности системы с закрепленными каналами.

Сети профессиональной радиосвязи проектируются по аналогии с вещательными системами: достаточно мощный передатчик работает через высоко подвешенную антенну, охватывая территорию в пределах прямой видимости радиусом 40 – 50 км. При этом на площади обслуживания в 5 – 8 тыс. кв. км абонентам может быть доступно несколько десятков радиоканалов.

На транкинговом принципе действия, начиная с 60-х годов прошлого столетия, был создан ряд как отечественных, так и зарубежных систем подвижной связи, которые модернизируются и используются по настоящее время, применяя различные стандарты.

Общей тенденцией развития профессиональных систем подвижной радиосвязи является переход от аналоговых корпоративных и национальных стандартов к цифровым международным стандартам с обеспечением конфиденциальности связи и роуминга абонентов. Внедрение общеевропейского стандарта на транкинговые системы подвижной радиосвязи TETRA началось в Европе в конце 90-х годов первоначально в интересах служб безопасности, полиции и охраны границ.

Однако эффективность транкинговых систем с радиальной структурой сети оказывается недостаточной для удовлетворения массового спроса на услуги подвижной связи в густонаселенных районах. Так, в Москве с ее 10-миллионным населением обеспечение только 0,1 % жителей подвижной связью при стандартных условиях качества обслуживания (средняя длительность переговоров 1,5 мин, вероятность блокировки 5 %) потребует выделения 250 радиоканалов.

Проблему организации подвижной связи для густонаселенных районов удалось решить путем построения подвижной связи по

сотовому принципу. Этот принцип является основой современной всемирной мобильной телефонии.

Система сотовой подвижной радиосвязи использует большое число маломощных передатчиков, которые предназначены для обслуживания только сравнительно небольшой зоны (радиусом 1 – 2 км). Эти небольшие зоны покрытия называют сотами. Такая организация предполагает, что все имеющиеся в распоряжении частотные каналы могут повторно использоваться в каждой ячейке сотовой структуры. Тогда требуемые для 0,1 % жителей Москвы 250 каналов можно получить, например, разделением обслуживаемой территории радиусом 50 км на 25 ячеек радиусом 10 км с организацией в каждой ячейке по 10 радиоканалов с одним и тем же набором частот.

Из-за недопустимого большого уровня взаимных помех ячейки с одинаковым набором частот необходимо перемежать буферными ячейками с другим набором частот. Группа ячеек с одинаковым набором частот называется кластером.

Мобильные телефонные системы можно разделить на поколения. Системы первого поколения основаны на передаче аналогового сигнала путем частотной модуляции. Они были рассчитаны в основном на обслуживание абонентов в рамках национальных границ.

Из-за растущей популярности систем первого поколения и, соответственно, усиления конкурентной борьбы за доступные частоты возникала потребность в других системах, более эффективно использующих спектр сигнала. Эта задача была решена системами второго поколения, в которых применялись цифровые технологии и методы множественного доступа с временным разделением (TDMA – Time Division Multiple Access) или множественный доступ с кодовым разделением (CDMA – Code Division Multiple Access). Третье поколение мобильных телефонных систем призвано объединить многочисленные системы второго поколения единым стандартом.

Рассмотрим принцип работы и структуру системы сотовой связи на примере наиболее простой, разработанной в США в 1987 г. системы первого поколения AMPS (Advanced Mobile Phone Service – перспективная мобильная телефонная связь), в которой используются все основные функциональные блоки любой системы сотовой связи следующих поколений. На рис. 12.5 представлена схема

функционирования системы AMPS, разработанная компанией AT&T в 80-х годах и в дальнейшем получившая распространение в Северной и Южной Америке, Австралии и Китае.

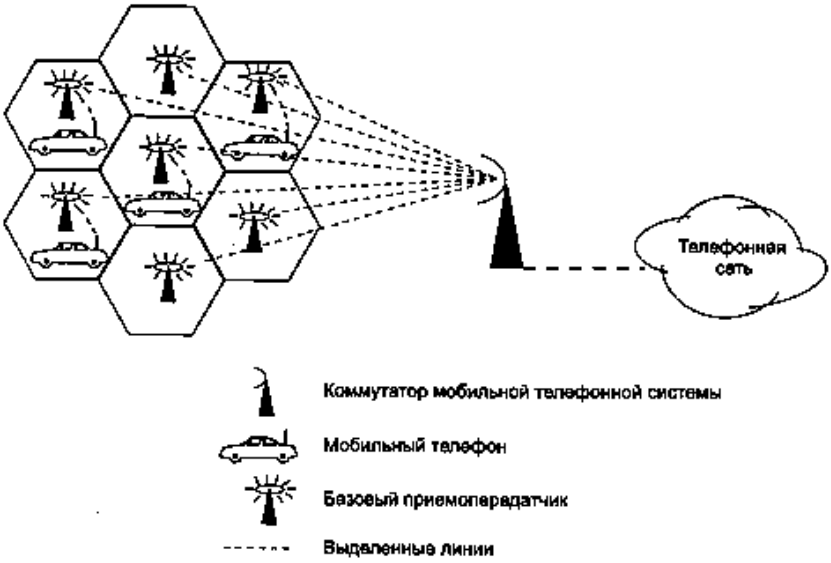


Рис. 12.5. Схема функционирования системы AMPS

Под систему AMPS (в Северной Америке) выделено два частотных диапазона шириной по 25 МГц. Один для передачи от базовой станции к мобильному телефону (869-894), другой для передачи от мобильного телефона к базовой станции (824-849). Каждый из этих диапазонов поделен пополам по 12,5 МГц и предоставляется двум различным операторам (с целью конкуренции на рынке присутствуют два оператора связи), т.е. каждому оператору выделяется по 12,5 МГц в каждом направлении. Эта полоса частот разбивается на каналы, каждому каналу выделяется полоса частот 30 кГц, таким образом, весь диапазон разделяется на 416 каналов. Двадцать один канал требуется под управление, в результате для телефонных разговоров остается 395 каналов. В большинстве случаев 395 каналов явно недостаточно. В связи с этим в системе AMPS реализована одна из основополагающих идей сотовой связи – многократное использование одних и тех же частот, т.е. их пространственное рас-

пределение. В качестве идеализированного примера многократного использования частоты на рис. 12.5 представлено гексагональное мозаичное разбиение территории на соты. В каждой соте установлен один приемопередатчик (базовая станция). Мощность сигнала станции тщательно контролируется, чтобы обеспечить устойчивую связь в пределах соты и в то же время максимально снизить интерференцию от телефонных сигналов в соседних сотах. Как правило, каждой соте назначается от 10 до 50 частот в зависимости от ожидаемого трафика. Между сотами с одинаковыми частотами располагаются промежуточные соты (использующие другой диапазон частот). Возможны различные варианты решения вопроса о числе промежуточных сот и схемы распределения частот. Со временем, по мере того как разработанной схемой начинает пользоваться все большее количество клиентов, количество частот, выделяемых одной соте, может оказаться недостаточным. В этом случае прежние соты могут дополнительно разбиваться на соты меньшего размера. Как правило, используются соты с изначальным размером в 6,5 – 13 км. Соты меньшего размера могут быть разбиты на еще более мелкие. Однако размер сот в 1,5 км близок к минимальному. На практике распределение трафика и топографические характеристики не являются однородными. Ячейки небольших размеров требуются только в центральной части города со значительной плотностью абонентов. Ближе к окраинам плотность снижается, и размеры ячеек могут увеличиваться. Расщепление ячеек может производиться достаточно гибко как в пространстве, так и во времени. Правда, использование сравнительно небольших сот усугубляет проблему поддержания непрерывной связи.

При движении по непрерывному маршруту объект системы в течение одного сеанса связи может миновать несколько ячеек. В этом случае непрерывность связи обеспечивается способностью системы автоматически передавать связь с объектом тем базовым станциям, в зоне действия которых он оказывается в данный момент. Кроме того, для сот меньшего размера необходимо снизить мощность сигнала внутри соты.

Благодаря непрерывным изменениям уровней сигналов, поступающих в центр коммутации подвижной связи базовых станций, ближайших к движущемуся объекту, система может определить момент пересечения границы двух ячеек и переключить разговор-

ный канал из первой ячейки во вторую в течение достаточно малого промежутка времени, не приводящего к нарушению непрерывности разговора. Такая процедура, получившая название эстафетной передачи (handover), требует весьма сложного алгоритма определения именно той ячейки из нескольких соседних, куда перемещается объект, а также быстродействующих алгоритмов и схемотехнических решений, обеспечивающих освобождение канала в первой ячейке и поиск свободного канала с восстановлением по нему связи во второй ячейке. Хорошо управляемая система способна поддерживать несколько десятков тысяч одновременных телефонных разговоров.

У каждого мобильного телефона стандарта AMPS есть три идентификатора. *Электронный порядковый номер* представляет 32-разрядный аппаратный идентификатор, назначаемый мобильному телефону его производителем. *Системный идентификационный номер* представляется собой 15-разрядный код, идентифицирующий системного оператора, к которому относится мобильный телефон. С его помощью система может выявлять мобильные телефоны, работающие за пределами обслуживания системного оператора, абонентом которого является владелец данного телефона. Работу мобильного телефона за пределами зоны обслуживания своего системного оператора называют «роумингом» (от английского слова rouming – бродяжничество). Для мобильных телефонов, работающих за пределами обслуживания своего системного оператора, должны выполняться специальные процедуры авторизации. Наконец, у мобильного телефона есть *мобильный идентификационный номер*, являющийся 34-разрядным идентификатором, представляющим 10-значный номер мобильного телефона.

Базовый приемопередатчик поддерживает дуплексную связь с мобильным устройством. Все базовые приемопередатчики соединены с коммутатором мобильной телефонной системы, как правило, радиоканалом СВЧ-диапазона или кабелем.

Коммутатор мобильной телефонной системы (Mobile Switching Center, MSC) представляет собой управляющий и контролирующий элемент сети AMPS. Он решает следующие задачи:

осуществляет коммутирующие функции в мобильной сети (например, передачу мобильного телефона от одной соты к другой);

координирует обратную связь;

занимается сбором данных для учета средств на счету абонента;

тестирует систему и следит за ней;

соединяет мобильную сеть с общедоступной кабельной телефонной сетью.

Системы второго поколения проектировались для создания крупномасштабных сетей с учетом обеспечения международного роуминга – автоматического обслуживания абонентов, приехавших со своими терминалами в другую страну.

Система GSM (Global System for Mobile Communication) – одна из самых быстроразвивающихся телефонных технологий создавалась как единая технология второго поколения для Европы, что должно было привести к использованию одинаковых телефонных аппаратов на всем континенте.

Эта технология оказалась крайне успешной и в настоящее время является самым популярным стандартом в мире. Впервые система GSM появилась в Европе в 1991 г. Теперь сходные системы развернуты в Северной и Южной Америке, Азии, Северной Африке, на Ближнем Востоке, в Австралии.

Такие системы второго поколения, как GSM, создавались на основе систем первого поколения и сохранили многие характеристики этих систем.

Новым замечательным элементом в системе GSM является модуль идентификации абонента (Subscriber Identity Module, SIM) или SIM-карта. Это переносное устройство в виде смарт-карты, хранящее идентификационный номер абонента, идентификаторы сетей, которыми разрешено пользоваться абоненту, ключи шифрования, а также другую информацию, специфическую для абонента. У абонентских устройств системы GSM нет характерных, отличающих их друг от друга особенностей до тех пор, пока в них не вставлена SIM-карта. Таким образом, абоненту нужно всего лишь носить с собой SIM-карту, чтобы пользоваться самыми различными абонентским устройствами в разных странах. Для этого ему достаточно просто вставить SIM-карту в устройство.

Передаваемые данные в GSM шифруются, что существенно увеличивает степень конфедициальности разговора. В AMPS, напротив, телефонные разговоры легко подслушать с помощью всеволнового приемника. Помимо классических услуг телефонной связи система GSM может поддерживать передачи данных и изо-

бражений. В основе услуг лежит модель ISDN, поддерживаются скорости передачи до 9,6 Кбит/с.

В общих чертах система GSM аналогична структуре AMPS, на рис. 12.6 показаны основные элементы GSM и ее типичная топология.

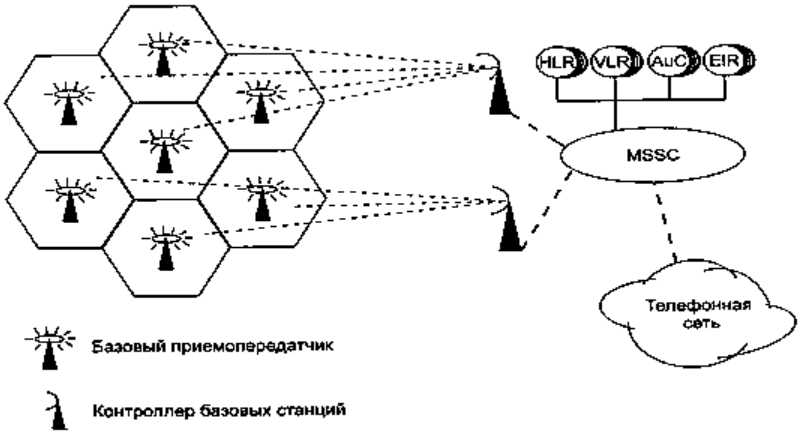


Рис. 12.6. Схема GSM

Четырьмя базовыми элементами системами являются: абонент (телефон), базовый приемопередатчик, контроллер базовых станций и коммутатор мобильных услуг (Mobile Services Switching Center – MSSC).

В отличие от централизованного управления, характерного для систем первого поколения, в системе стандарта GSM принят принцип распределенного управления между центром коммутации подвижной связи, базовыми станциями и подвижными терминалами. В течение всего сеанса связи подвижные терминалы измеряют уровни сигналов от соседних базовых станций и сообщают результаты измерения обслуживающей их базовой станции. Последняя определяет необходимость эстафетной передачи и транслирует информацию о наиболее предпочтительной новой ячейке для обслуживания подвижного объекта к системному контроллеру центра коммутации подвижной связи. Благодаря такому алгоритму распределенного управления большая часть работы выполняется не системным контролером, а базовыми станциями и подвижными термина-

лами, что позволяет избежать перегрузки центрального звена и упростить процедуру эстафетной передачи.

Контроллеры базовых станций могут располагаться либо в непосредственной близости от базовых приемопередатчиков, либо около коммутатора мобильных услуг. В том случае, когда базовый приемопередатчик и контроллер базовых станций располагаются вместе, эту пару часто называют надсистемой базовой станции. Хотя контроллер базовых станций выполняет множество таких функций, как обслуживание переходов из одной соты в другую, управление уровнем мощности сигнала и выделение частот абонентам, сердцем системы является коммутатор мобильных услуг. Он предоставляет интерфейс с общественной кабельной телефонной сетью и учитывает средства, потраченные абонентом на телефонные разговоры. Хранящиеся в коммутаторах данные располагаются в четырех базах данных.

В базе данных HLR (Home Location Register – реестр абонентов) хранится информация (как постоянная, так и временная) о каждом абоненте системы (т.е. об абоненте, телефонный номер которого относится к коммутатору). Кроме того, у каждого коммутатора имеется база данных VLR (Visitor Location Register – реестр расположения посетителей), в которой хранится информация об абонентах, находящихся в данный момент в районе обслуживания коммутатора. В ней фиксируется, активен ли абонент в данный момент, а также другие параметры абонента. Когда GSM получает вызов своего абонента, она по телефонному номеру находит домашний коммутатор абонента. Затем этот коммутатор по своей базе данных HLR находит информацию о текущем физическом расположении вызываемого абонента. Если же звонок поступает на коммутатор абонента мобильного телефона, используется база данных VLR. Даже если абонент находится в зоне обслуживания своего домашнего коммутатора, сведения о нем помещаются в базу данных VLR для простоты и надежности.

База данных AuC (Authentication Center – центр аутентификации) – база, в которой располагаются ключи аутентификации и шифрования всех абонентов, зарегистрированных как в базе данных HLR, так и в базе данных VLR. Центр аутентификации контролирует доступ к пользовательским данным, а также управляет процедурами аутентификации, когда абонент подключается к сети.

В базе данных EIR (Equipment Identity Register – реестр оборудования) хранятся сведения об оборудовании мобильных станций. Она также используется для обеспечения безопасности (например, с ее помощью блокируются звонки с украденных телефонов, а также предотвращается использование сети станций, не прошедших аутентификации).

Основным стимулом перехода систем сотовой связи первого поколения к системам второго поколения была необходимость экономии частотного диапазона. Системы первого поколения оказались исключительно успешными, поэтому количество абонентов росло экспоненциально. Однако рост систем сотовой телефонии был ограничен возможностями частотного диапазона. Существует четыре основных метода разделения частотного диапазона между активными пользователями. Одним из них является пространственное мультиплексирование (Space Division Multiplexing – SDM), подразумевающее использование одного и того же диапазона в двух пространственно удаленных друг от друга. Именно эта идея лежит в основе сотовой связи. Одна и та же частота может использоваться в двух достаточно удаленных сотах так, чтобы сигналы не могли интерферировать друг с другом (как это отмечалось ранее).

Дальнейшее развитие систем сотовой связи связано с частотным мультиплексированием (Frequency – Division Multiplexing, TDM) и кодовым мультиплексированием (Code Division Multiplexing, CDM), о принципах которых говорилось в гл. 3 применительно к кабельным системам передачи данных, локальным и глобальным вычислительным системам.

Таким образом, основные принципы характерные для сотовой связи, можно сформулировать следующим образом:

использование маломощных передатчиков с радиопокрытием небольших по размеру ячеек;

повторное использование частот в пределах одной зоны обслуживания;

поэтапное увеличение пропускной способности за счет расщепления ячеек;

обеспечение непрерывной связи в процессе перемещения объекта от ячейки к ячейке.

Все это привело в начале 80-х годов к созданию и массовому внедрению во всем мире сотовых систем подвижной радиосвязи.

Дальнейшее развитие систем беспроводной связи связано с созданием систем третьего поколения. Целью систем беспроводной связи третьего поколения является представление скорости передачи данных, достаточной для поддержки мультимедиа, передачи данных и видео в дополнение к обычной телефонной связи. Так, международный союз мобильных телекоммуникаций определил следующие возможности систем третьего поколения:

качество телефонной связи, сопоставимое с общедоступной коммутируемой телефонной сетью;

скорость передачи данных от 144 до 384 Кбит/с для пользователей, использующих различные средства передвижения (в зависимости от скорости передвижения);

поддержка скорости передачи данных 2,048 Мбит/с для офисного использования пользователем;

поддержка услуг, предоставляемых системами коммутации, как пакетов, так и каналов;

адаптивный интерфейс с Интернетом;

более эффективное использование всего доступного частотного диапазона;

поддержка широкого спектра оборудования;

гибкость, позволяющая внедрять новые услуги и технологии.

Одной из движущих сил современной коммуникационной технологии является тенденция к объединению универсальных персональных средств связи и универсального доступа. Первая концепция означает способность пользователя легко идентифицировать себя и задействовать любую систему связи (подразумевается как наземных, так и спутниковых услуг) в пределах одного государства, континента или даже глобально с помощью всего одной учетной записи.

Вторая концепция означает способность в любом окружении для получения информационных услуг применять один и тот же терминал. Становится очевидно, что революция в области персональных компьютеров существенным образом затрагивает беспроводную связь. Так, система GSM с ее модулем идентификации абонента представляет собой большой шаг в этом направлении.

Среди технологий под общим названием PCS (Personal Communication Services – персональные коммуникационные услуги) можно назвать американскую и японскую системы, а также европей-

скую систему DECT (Digital European Cordless Telephone – европейский стандарт цифровой беспроводной связи). В этих системах могут использоваться низкоорбитальные и геостационарные спутники, а также наземные антенны.

Технология развивается стремительно. Среди характерных возможностей систем третьего поколения имеется уже реализованная возможность доступа к услугам ГВС – Интернет.

12.5. Беспроводное подключение узлов в локальных сетях

Беспроводная (wireless) связь в локальных сетях осуществляется в инфракрасном и радиочастотном диапазоне электромагнитных волн. Для организации беспроводного подключения узлов к сети требуется два компонента: точка доступа и беспроводной адаптер. Точка доступа представляет стационарное устройство, подключаемое к проводной локальной сети и имеющее приемопередатчик. Беспроводной адаптер, как правило, исполняется в виде карты расширения компьютера также снабженной приемопередатчиком. Выпускаются и внешние беспроводные приемопередатчики, подключаемые к сетевой карте компьютера (ETHERNET). Расстояние, на котором поддерживается связь между адаптером и точкой доступа, зависит от технологии связи, размеров антенн, особенностей электромагнитной обстановки в помещении. Для инфракрасной связи диапазон не превышает единиц метров; для радиосвязи, в зависимости от технологии, может быть в пределах 30 – 150 м, а при использовании внешних антенн может достигать нескольких километров.

Одна точка может обслуживать несколько мобильных узлов, но, естественно, чем больше узлов, тем меньше эффективная скорость передачи каждого из них. Если одной точки доступа недостаточно для охвата необходимой территории, устанавливается несколько точек и организуется роуминг (определение точки, обслуживающей конкретного мобильного абонента). Таким образом, формируется микросотовая система связи. Функции роуминга выполняются средствами точек доступа.

В 1997 г. был принят стандарт для беспроводных сетей IEEE 802.11 (в народе получивший название Wi-Fi), в котором определя-

ется несколько вариантов технологии физического и логического уровней.

Потребность в беспроводном подключении узлов к локальной сети возникает в ряде случаев использования локальных сетей на объектах здравоохранения, образования, торговли, а также для объединения локальных сетей, расположенных в разных зданиях.

Беспроводная связь требуется мобильным (в пределах офиса) пользователям, использующим в основном блокнотные ПК с интерфейсом PC Card, применяют ее и для подключения стационарных пользователей, расположенных вне зоны охвата кабельной сети.

Использование радиодоступа обостряет вопросы безопасности, поскольку среда передачи становится неконтролируемой и наряду с ростом потребности беспроводных сетей повышается интерес к средствам из взлома и это не единственное слабое место этих сетей:

- радиосигналы значительно ослабевают с расстоянием существенно быстрее, чем сигналы в любых кабельных каналах;

- радиоволны отражаются от стен, мебели и оборудования и идут от передатчика к приемнику сразу несколькими путями (в результате интерференции волн в точке приема могут возникать замирания сигнала);

- сигнал искажается радиопомехами от различного электрооборудования (микроволновые печи, стартеры автомобильных двигателей, лампы дневного света и т.п.);

- беспроводные радиосигналы принципиально уступают оптоволоконным, Ethernet и ADSL линиям по скорости и безопасности. В обычных ЛВС (как уже отмечалось) скорость передачи может достигать от 1 до 100 Гбайт/с, а в соответствии IEEE 802.11 b скорость беспроводных ЛВС не превышает 11 Мбайт/с. Оборудование, в котором реализованы новые протоколы IEEE 802.11 a и 11 g, позволяет реализовать 54 Мбайт/с, а вновь разрабатываемый стандарт предусматривает передачу данных со скоростью 108 Мбайт/с. Однако названные цифры отражают чрезмерно оптимистичный взгляд на возможности Wi-Fi. На самом деле в силу ряда причин скорость приходится снижать, так даже по протоколу IEEE 802.11 b скорость может падать до 1 Мбайт/с.

В связи с перечисленными трудностями отношение к беспроводным радиосетям до 2000 г. было сдержанным, однако в настоя-

щее время ситуация резко изменилась. Значительно возросшая потребность в ресурсах Internet, значительное увеличение мобильных пользователей ПК и т.п. привели к необходимости дальнейшего совершенствования связи, не привязанной к проводам.

Разработка версии Smart Wi-Fi, в которой предусмотрены меры борьбы с перегрузкой, разрешение конфликтов с помощью протокола Ethernet (CSMA/CA), борьба с помехами и эхоподавление, благодаря грамотному проектированию расположения станций, а также адаптация сети при изменении условий прохождения сигнала значительно изменило отношение к Wi-Fi.

К беспроводным сетям Wi-Fi стандартов IEEE 802.11 a, b, g все уже давно привыкли, в офисах эти сети используются наряду с проводными сетями, более того уже есть провайдеры Wi-Fi доступа в Internet для домашних пользователей и корпоративных клиентов.

Wi-Fi – не единственные сети, использующие беспроводные радиосигналы. К сетям радиочастотного диапазона относятся также региональные сети WiMAX. В отличие от Wi-Fi сетей они основаны на стандарте IEEE 802.16 d, принятом в 2004 г. для скоростного беспроводного обмена данными со стационарными пользователями на расстояниях до 50 км. Следует отметить, что максимальный радиус охвата Wi-Fi сетей не превышает нескольких сотен метров. Станции WiMAX передают сигналы большой мощности, которая зависит от используемого радиодиапазона. Скорость передачи данных в сетях WiMAX достигает 75 Мбайт/с, правда полоса пропускания распределяется по множеству пользователей.

Несмотря на первоначальное назначение технологии, разрабатывается стандарт для мобильных устройств IEEE 802.16 e, который известен как mobile WiMAX.

Беспроводные сети, соответствующие этому стандарту, будут предоставлять Интернет-доступ для ноутбуков и карманных компьютеров в радиусе нескольких километров.

Пока этот стандарт еще не принят, поэтому не ясно найдет ли эта технология массовое развитие и вряд ли WiMAX и Wi-Fi будут конкурировать, скорей всего каждая из этих технологий займет свою экономическую нишу (по аналогии с кабельными локальными и региональными сетями).

Безусловно, отсутствие «привязки» к проводу – важнейшее достоинство радиосетей, а решение ряда сложных технологических

проблем значительно увеличивает их возможности массового внедрения. Правда, разработки этих сетей увеличивают один «неприятный» вопрос, который все чаще поднимается при обсуждении сотовой связи, – вопрос электромагнитной совместимости и влияние слабых электромагнитных полей на организм человека. И, несмотря на заверения экспертов по данной проблеме (связанных с компаниями сотовой связи) об абсолютной безопасности подобных полей, существует и прямо противоположное мнение других экспертов (не связанных с компаниями сотовой связи) с некоторыми статистическими данными. Истина, по-видимому, как всегда, где-то посредине.

Еще один вид беспроводной связи, который в последнее время начинает широко применяться, – это использование световых пучков, несущих кодированные сигналы. Такие сети на основе светодиодов называют оптическими, потому что передача данных в них осуществляется видимыми или невидимыми (инфракрасными) световыми волнами.

В существующих оптических беспроводных системах используется крайне маломощное инфракрасное излучение (ИК). ИК-пучки уже достаточно давно используются в пультах дистанционного управления (ПДУ) телевизорами.

Оптические системы позволяют связать беспроводные цифровые устройства с информационным портом в помещении, который, в свою очередь, может быть связан с любой кабельной широкополосной сетью. Эта технология ограничивается отдельным помещением, внутри которого имеется ряд преимуществ перед радиоканалом:

во-первых, безынтерфереционные элементы с фокусировкой обеспечивают почти неограниченную ширину полосы для большого числа пользователей; деструктивная интерференция невозможна (в отличие от радиоволн), поскольку чувствительные элементы в активной зоне фотодиода поглощают волны независимо и затем усредняют поступающую информацию, взаимное уничтожение исключается;

во-вторых, она информационно безопасна, поскольку свет в отличие от радиоволн не проходит сквозь стены;

в-третьих, фотоны в отличие от радиоволн одинаковой частоты не создают взаимных помех (ширина полосы ограничивается толь-

ко максимальной скоростью, с которой устройства способны принимать поступающие данные);

в-четвертых, скорость передачи ИК-систем достигает гигабит в секунду (в радиосистемах, как отмечалось ранее, скорость достигает мегабит в секунду).

Исследователи занимаются концепцией оптической связи внутри помещений с начала 1980 г., когда была построена первая работоспособная системы. Ее развитие задержалось на десятилетие, поскольку к этому времени еще не появились потребности в беспроводном доступе к таким мультимедийным веб-услугам как поиск в Интернете, видеоконференции, телепередачи и кино по заказу (Интернет находится в младенческом состоянии), а Wi-Fi, да и WIMAX были мало пригодны для таких целей. Благодаря поразительному росту Интернета в последние годы ситуация кардинально изменилась и оптический вид связи оказался востребованным. Наряду с проблемой «последней линии» – трудности передачи широкополосной информации от высокоскоростных общенациональных сетей стационарным пользователям (о которой писали ранее) – возникала похожая проблема «последних футов» – сложности передачи широкополосного трафика от оконечного устройства стационарной сетевой структуры мобильным пользователям в помещениях. И эта проблема вполне успешно может решаться с помощью ИК-систем.

Хотя не все так просто. Оптические каналы связи работают лучше всего, когда передатчик направлен прямо на приемник как в привычных ПДУ для телевизоров и цифровых камер. Но такая система неприемлема для связи между всеми работниками офиса или передачи информации людям в местах их скопления. Нужен полный обхват помещения. Оптические сети должны распределять несущее информацию излучение по всему пространству помещения. Для этих целей были разработаны устройства, в которых несколько копий данных передается в виде сетки узконаправленных ИК-лучей, заполняющих весь объем помещения в результате отражения от различных поверхностей в нем. Маломощные лучи, несущие одну и ту же информацию, соединяют порт доступа, подключенный к базовой структуре высокочастотной передачи данных, со всеми цифровыми устройствами в помещениях оборудованных ИК-приемником. В пространстве, заполненном множеством коди-

рованных лучей, пользователь может передвигаться по всему помещению, оставаясь все время на связи с системой, при потере контакта с одними лучами автоматически происходит подключение к другим. Сетка узких лучей работает с высокой скоростью передачи информации – до 1 Гбит/с.

ИК-системы оптической беспроводной связи будут, вероятнее всего, со временем вытеснены системами на основе светодиодов, излучающих белый свет, которые позволяют получить еще более широкую полосу и ряд других преимуществ. Правда, для внедрения беспроводных сетей на основе белых светодиодов разработчикам предстоит решить ряд проблем.

И в заключение необходимо отметить, что внедрению оптической беспроводной связи способствует грядущая технология обеспечения доступа стационарным пользователям доступа на «последней миле» – широкополосная связь по силовым линиям (Broad band over Power Lines, BoPL), в которой широкополосные данные передаются по существующим силовым линиям среднего или низкого напряжения к настенным розеткам в помещениях. Подобные услуги уже предлагаются в США, Испании, Нидерландах, Норвегии, Швеции, Южной Корее и Японии. Для использования широкополосного «моста» в оптических беспроводных приложениях нужно только включить в розетку электросети недорогой адаптер, способный передавать данные с помощью встроенного в него ИК-передатчика цифровым устройствам в помещении, имеющим ИК-приемник.

Какое бы излучение – ИК или видимое – не было выбрано, операторы беспроводных цифровых устройств получат новую возможность «въехать в будущее на широкополосной волне». Оптическая беспроводная технология хорошо подходит для создания моста, обеспечивающего широкополосный цифровой доступ на всей площади нашего жилого или рабочего пространства, занимая свою экологическую нишу в локальных сетях.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Перечислите направления применения беспроводных систем передачи данных, основные достоинства и слабые стороны беспроводных систем.

2. Каковы особенности применения беспроводных средств связи в локальных вычислительных сетях (ЛВС)? Перечислите разновидности и функциональные возможности радиосвязи. Расскажите об использовании инфракрасного излучения для передачи информации в ЛВС.

3. Перечислите основные типы радиосвязи, используемые в ГВС. Как зависит частотный диапазон от используемого типа радиосвязи?

4. В чем особенность спутниковой системы связи? Какие орбиты спутников используются для передачи данных? Назовите их сравнительные характеристики.

5. Перечислите основные системы подвижной радиосвязи, особенности сотовой связи, различия в системах AMPS и GSM.

СПИСОК ТЕРМИНОВ И ОПРЕДЕЛЕНИЙ

Алгоритм шифрования – различные способы преобразования текста.

Аналоговые данные – физическая величина, которая может изменяться в непрерывном диапазоне значений. Величина прямо пропорциональна данным или является их функцией.

Аналоговый сигнал – непрерывно изменяющаяся электромагнитная волна, распространяющаяся в различных средах.

Асинхронный метод передачи (или стартстопный) – посимвольный режим передачи с контролем начала и конца символа, имеет низкую скорость и малую эффективность.

Ассоциативная память – память, поиск информации в которой осуществляется одновременно во всех ячейках памяти по ее содержанию (содержимому), ассоциативному признаку, что позволяет в некоторых случаях существенно ускорить поиск и обработку данных.

Атрибуты – дополнительные параметры, определяющие свойства файлов.

*Брандмауэр - о*сновной аппаратный компонент для реализации методики управления доступом к объединенной сети.

Вандалы – специалисты в написании компьютерных вирусов и их разновидностей под названием «троянских коней».

Вектор – упорядоченный массив независимых чисел.

Взломщики – профессиональные кракеры.

Вирусы-«черви» (*worm*) – вирусы, которые распространяются в компьютерной сети и не изменяют файлы или секторы на дисках.

Временное мультиплексирование – разделение системы передачи на два или более канала путем поочередного подключения общей линии к разным информационным каналам.

Выделенный канал – канал с фиксированной полосой пропускания или фиксированной пропускной способностью, постоянно соединяющий двух абонентов.

Глобальные или всемирные сети (GAN – Global Area Network) – сети, которые предназначены для связи абонентов, расположенных в различных географических регионах, и включают в себя множество сетей различной протяженности – материковые (WAN – Wide Area Network), региональные или городские ГВС (MAN – Metropol-

itan Area Network), а также локальные (LAN – Local Area Network), кампусная, объединяющая локальные сети близко расположенных зданий (CAN – Campus Area Network).

Грид (GRID) – перспективное направление развития информационных технологий (ИТ). Термин «грид» применяется по аналогии с понятием «Power grid» – система, интегрирующая мощности электрических сетей в единое «хранилище» энергии, откуда она перераспределяется вне зависимости от ее источника.

Зашифрованный текст – искаженное сообщение, формируемое на выходе алгоритма.

Инкапсуляция – способ объединения в единое целое подпрограмм и данных, которые они обрабатывают.

Интерфейс (или средства сопряжения) – совокупность средств, обеспечивающих логические, электрические и конструктивные условия совместимости ЦП и функциональных устройств в точках сопряжения и их взаимодействия.

Информатика – наука о законах и методах измерения, передачи, обработки и хранения информации с использованием математических, технических и программных средств.

Исходная программа – программа в кодах языка программирования.

Каталог (*директория*) – оглавление, или файл, содержащий информацию о файлах (имя, расширение, размеры в байтах, дату время создания и т.д.).

Классы (classes) – абстрактные типы данных с похожими свойствами в объектно-ориентированных языках.

Кластер – сложный программно-аппаратный комплекс.

Компьютерный вирус – небольшая программа (средний размер – 700 байт), написанная на языке Assembler и выполняющая разрушительные для ОС действия.

Корпоративная сеть – сеть, получившаяся в результате объединения нескольких, как правило, разнообразных сетей.

Кракеры – специалисты, способные снять защиту от копирования с лицензионного программного обеспечения.

Линии интерфейса – электрические цепи, являющиеся составными физическими связями интерфейса.

Магистраль – совокупность всех шин интерфейса.

Макровирусы – особая разновидность вирусов, которые поражают документы в прикладных программах, имеющие расширение .doc, например документы, созданные текстовым процессором MS Word и выполняющие макрокоманды.

Мейнфрейм – синоним понятия «большой универсальный компьютер».

Микропроцессор (МП) – изготовленный при помощи технологии высокой степени интеграции прибор, который способен выполнять под действием программного управления функции центрального процессора компьютера.

Носитель – пакет дисковых пластин, вращающихся на одной оси.

Объект данных – комбинация данных, атрибутов, описывающих их свойства, и методов, раскрывающих их поведение.

Объектная (или полуготовая) – программа в кодах машины, прошедшая фазу трансляции, но выполнение которой невозможно по ряду причин.

Оперативная (основная Main Memory) *память* (ОП)*, или оперативное запоминающее устройство* (ОЗУ) – пассивное устройство, являющееся непосредственным местом хранения программ и обрабатываемых по этим программам данных.

Открытая система – система, которая состоит из компонентов, взаимодействующих друг с другом через стандартные интерфейсы.

Открытый текст – исходное сообщение или данные, которые подаются на вход алгоритма.

Персональный компьютер (ПК) – небольшой компьютер, основой которого служит МП, т.е. микрокомпьютер.

Пиксель (сокращенно от слов picture cell) – элемент изображения, представляющий собой единицу измерения разрешения экрана (монитора) или печатного изображения и соответствующий отдельной светящейся точке, цветом и яркостью которой можно управлять.

Платформа – программно-технические условия, в рамках которых работает (или не работает) данная прикладная программа.

Позиционер – устройство, наводящее головку на нужную дорожку.

Полиморфизм – многозначность сообщений.

Полиморфные вирусы – вирусы, модифицирующие свой код в зараженных программах таким образом, что два экземпляра одного и того же вируса могут не совпадать ни в одном бите.

Прерывание (*interrupt*) – сигнал, заставляющий менять порядок исполнения потока команд.

Программа – последовательность команд или операторов, которая после декодирования ее либо вычислительной машиной, либо вычислительной машиной и транслирующей программой может заставить эту машину выполнить некоторую работу.

Протокол – строго заданная процедура или совокупность правил, регламентирующая способ выполнения определенного класса функций.

Рабочая программа (или загрузочный модуль) – программа, которая может быть выполнена на ЭВМ.

Рабочие станции – индивидуальные компьютеры пользователей.

Разделители – элементы любого языка программирования, предназначенные для разделения отдельных элементов выражений внутри строки, для разделения строк, отдельных фрагментов программ и т.п.

Редакторы связи (LINKEDITOR), или компоновщики – редакторы, у которых основная задача – сборка программы, готовой к исполнению, подключающиеся к основной программе внешние модули как системных библиотек, включенных в состав соответствующих трансляторов, так и внешних библиотек, включая модули, разработанные самим пользователем.

Роуминг (от английского слова rouming) – бродяжничество.

Сервер – компьютер, обеспечивающий обслуживание пользователей сети: разделяемый доступ к процессорам, дискам, файлам, принтеру, системе электронной почты и т.д.

Синхронный метод передачи – метод, осуществляющий объединение большого количества символов или байт в отдельные блоки – кадры, которые передаются без задержек между восьмибитными элементами.

Стековая память (или магазинная) – одномерная память, представляющая набор ячеек, где могут быть записаны фрагменты программ, считывание которых осуществляется по указателю в порядке, обратном записи.

Стек коммуникационных протоколов – иерархический набор протоколов, достаточный для организации взаимодействия узлов в сети.

Стык – место соединения устройств передачи сигналов данных, входящих в систему передачи данных.

Суперкомпьютеры – один из наиболее динамично развивающихся классов компьютеров, имеющих обширные и очень важные области применения.

Транспьютеры – необычный вид RISC-процессоров. Слово «транспьютер» – синтез двух слов: транзистор + компьютер.

Троянские программы – программы, не способные самостоятельно размножаться, и их распространение основано целиком на добровольном копировании.

Уплотнение информации, или мультиплексирование, – функция, позволяющая двум или более источникам данных совместно использовать общую среду передачи данных таким образом, что каждый получает собственный канал передачи данных.

Утилиты (вспомогательные программы) предназначены для копирования, перемещения, печати, перфорации данных.

Файл – поименованная область на внешнем носителе (диске или магнитной ленте), в которой может храниться исходный текст программы, программы в машинном коде, готовые к исполнению, или данные для ее работы.

Файловая система – совокупность каталогов и файлов, хранящихся на внешних носителях компьютера.

Хакерами именуют людей, специализирующихся на взломе защиты отдельных компьютеров и распределенных систем.

Центральный процессор (CPU – central processing unit) – основная часть («мозг») компьютера, непосредственно осуществляющая процесс обработки данных и управляющая ее работой.

Цифровой сигнал – дискретный (разрывной) сигнал, такой, как последовательность импульсов напряжения.

Цифровые данные – данные, которые принимают дискретные значения – текст, целые числа, двоичные данные.

Частотное мультиплексирование – разделение системы передачи на два или более канала путем разделения всей доступной полосы частот на более узкие полосы, каждая из которых образует отдельный канал.

Шина – совокупность линий, сгруппированных по функциональному назначению (шина адреса, шина команд, шина данных, шина состояния и т.п.).

Шифрование – преобразование открытого текста или данных в нечитаемую форму при помощи обратимых математических операций.

Шутники – наиболее безобидная часть кракеров, к ним можно отнести создателей компьютерных вирусов с различными визуально-звуковыми эффектами.

CERN (Conceil Europeenpourla Recherche Nucleaire) – Европейская организация по ядерным исследованиям.

CISC (Complex Instruction Set Computing) – архитектура с полным набором машинных команд.

EDI (Electronic Data Interchange) – электронный обмен данными.

EFT (Electronic Funds Transfer) – электронный перевод денежных средств.

Internet – объединение сетей самого различного масштаба, состоящих из миллионов компьютеров, имеющих единые аппаратно-независимые протоколы среднего уровня (транспортные и сетевые адресные), использующих самые разнообразные каналы связи с различными протоколами канального и физического уровня.

Proxy – схема с дополнительной идентификацией и аутентификацией пользователей.

RISC (Reduced Instruction Set Computing) – архитектура с упрощенным набором команд.

СПИСОК ЛИТЕРАТУРЫ

1. Бауэр Ф.Л., Гооз Г. Информатика. Вводный курс: Пер. с нем. – М.: Мир, 1976 (2-е изд., перераб., 1990).
2. Омельченко Л., Федоров А. Windows Millenium. Самоучитель. – СПб.: BHV, 2000.
3. Немнюгин С.А. Turbo-Pascal: учебник. – СПб.: Питер, 2000.
4. Информатика, базовый курс: учебное пособие для втузов / Под ред. С.В. Симоновича. – СПб.: Питер, 2000.
5. Шафрин Ю.А. Информационные технологии: учебное пособие. – Ч. 1 и 2. – М.: Лаборатория базовых знаний, 2000.
6. Дьяконов В.П. MathCAD 812000: специальный справочник. – СПб.: Питер, 2000.
7. Бартеньев О.В. Графика OPENGL: программирование на Фортране. – М.: Диалог-МИФИ, 2000.
8. Лукин С.Н. Турбо-Паскаль 7.0: самоучитель для начинающих. – М.: Диалог-МИФИ, 2000.
9. Король В.И. Visual Basic 6.0, Visual Basic for Applications 6.0: справочник с примерами. – М.: Кудиц – образ, 2000.
10. Крамер Дуглас Э. Сети TCP/IP. Т. 1. Принципы. Протоколы и структура. – М., СПб., Киев, 2003.
11. Шнайер Брюс. Прикладная криптография. Протоколы. Алгоритмы, исходные тексты на языке Си. – М.: ТРИУМФ, 2003.
12. Милославская Н.Г., Толстой А.И. Интерсети: доступ в Интернет, защита. – М.: ЮНИТИ, 2000.
13. Милославская Н.Г., Толстой А.И. Интерсети: обнаружение вторжений. – М.: ЮНИТИ, 2001.
14. Информатика и математика для юристов: учебное пособие / Под ред. Х.А. Андриашина, С.А. Казанцева. – М.: ЮНИТИ, 2003.
15. Аверьянов Г.П., Рошаль А.С. Введение в информатику. – М.: МИФИ, 2002.
16. Криминалистика: учебник / Под ред. А.С. Волынского. – М.: Право и закон, 2002.
17. Баррон Д. Введение в языки программирования: Пер. с англ. – М.: Мир, 1980.
18. Фокс Дж. Программное обеспечение и его разработка: Пер. с англ. – М.: Мир, 1992.