

Министерство образования и науки Российской Федерации

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

ИНФОРМАТИКА

Учебное пособие

Томск
«Эль Контент»
2011

УДК 004(075.8)

ББК 32.97я73

И741

Тимченко С. В.

И741 Информатика : учебное пособие / С. В. Тимченко, С. В. Сметанин, И. Л. Артемов, А. В. Гураков, О. И. Абдалова, П. С. Мещеряков, П. А. Башкиров, Д. С. Шульц. — Томск: Эль Контент, 2011. — 160 с.

ISBN 978-5-4332-0009-8

В учебном пособии по дисциплине «Информатика» рассмотрены основные разделы информатики, определяющие современный уровень подготовки специалистов в системе высшего образования. В пособии изложены основы информатики, раскрыты технологии и инструменты сбора, обработки, хранения, поиска и передачи информации с использованием современного технического аппарата. Рассмотрены общие вопросы информационной безопасности.

Пособие подготовлено в соответствии с требованиями Государственного образовательного стандарта высшего профессионального образования.

Учебное пособие по дисциплине «Информатика» предназначено для студентов факультета дистанционного обучения ТУСУР.

УДК 004(075.8)

ББК 32.97я73

ISBN 978-5-4332-0009-8

- © Тимченко С. В., Сметанин С. В., Артемов И. Л., Гураков А. В., Абдалова О. И., Мещеряков П. С., Башкиров П. А., Шульц Д. С., 2011
- © Оформление.
ООО «Эль Контент», 2011

ОГЛАВЛЕНИЕ

Введение	6
1 Основные понятия информатики	9
1.1 Что такое информатика?	9
1.2 Информация	11
1.3 Свойства информации	13
1.4 Данные и методы их воспроизведения и обработки	16
1.5 Кодирование информации	19
1.5.1 Системы счисления	20
1.5.2 Единицы измерения данных	28
1.5.3 Кодирование текстовой информации	28
1.5.4 Кодирование графической информации	30
1.5.5 Кодирование звуковой информации	31
2 Вычислительная техника	34
2.1 История развития вычислительной техники	34
2.1.1 Первое поколение (1945–1955 гг.): электронные лампы	36
2.1.2 Второе поколение (1955–1965 гг.): транзисторы	37
2.1.3 Третье поколение (1965–1980 гг.): интегральные схемы	37
2.1.4 Четвертое поколение (с 1980 гг.): сверхбольшие интегральные схемы	37
2.2 Классификация компьютеров	38
2.3 Устройство компьютера	41
2.3.1 Архитектура ЭВМ	44
2.3.2 Центральный процессор	47
2.3.3 Память компьютера и ее виды	47
2.3.4 Аудио- и видеоадаптер	50
2.3.5 Видеосистема компьютера	51
2.4 Периферийные устройства	52
2.4.1 Принтеры	52
2.4.2 Сканеры	53
2.4.3 Модемы и факс-модемы	53
3 Программное обеспечение компьютера	55
3.1 Классификация программного обеспечения	56
3.2 Базовое программное обеспечение	57
3.3 Операционные системы	59
3.3.1 Представление данных в виде файлов и каталогов	59
3.3.2 Базовые понятия операционных систем	62
3.3.3 Виды операционных систем	63

3.3.4	Операционная система UNIX	65
3.3.5	Операционная система Windows	65
3.4	Служебные программы	66
3.4.1	Файловые менеджеры	66
3.4.2	Архиваторы	67
3.4.3	Стандартные утилиты Windows	69
3.5	Прикладное программное обеспечение	70
3.5.1	Текстовые редакторы и процессоры	70
3.5.2	Графические редакторы	73
3.5.3	Офисные интегрированные программные средства	74
4	Сети ЭВМ	77
4.1	Общие сведения о сетях ЭВМ	77
4.2	Назначение компьютерных сетей	78
4.3	Компоненты аппаратного и программного обеспечения сетей	79
4.4	Классификация сетей	79
4.4.1	По технологии передачи	79
4.4.2	По территориальной распространенности	80
4.4.3	По принадлежности	81
4.4.4	По способу управления	81
4.4.5	По топологии (способу организации связей)	83
4.4.6	По типу среды передачи	86
4.4.7	По скорости передачи	87
4.5	Глобальная сеть Интернет	87
4.5.1	Историческая справка	87
4.5.2	Способы доступа в Интернет	89
4.5.3	Структура и основные принципы работы Интернета	92
4.5.4	Сервисы сети Интернет	94
4.6	Выводы	98
5	Базы данных	99
5.1	Краткая история баз данных	100
5.1.1	Ранние модели баз данных	101
5.1.2	Недавняя история	101
5.2	Распределенные базы данных	102
5.2.1	Перспективы развития	103
5.2.2	Основные понятия реляционной модели	103
5.2.3	Отношение, схема отношения, кортеж	104
5.2.4	Тип данных и домены	105
5.2.5	Уникальность кортежей отношения	106
5.2.6	Отсутствие упорядоченности кортежей и атрибутов	106
5.2.7	Атомарность значений атрибутов, первая нормальная форма	107
5.2.8	Характеристика реляционной модели	108
5.2.9	Технология манипулирования данными в реляционной структуре	110
5.3	Нормализация отношений	111
5.3.1	Вторая нормальная форма	112

5.3.2	Третья нормальная форма	114
5.4	Итоги	116
6	Безопасность компьютерных систем	117
6.1	Понятие безопасности	117
6.2	Виды угроз	117
6.3	Злоумышленники	118
6.4	Основы криптографии	119
6.4.1	Шифрование с закрытым (симметричным) ключом	119
6.4.2	Шифрование с открытым (несимметричным) ключом	120
6.4.3	Цифровые подписи	120
6.5	Аутентификация пользователей	120
6.5.1	С использованием пароля	121
6.5.2	Совершенствование безопасности паролей	121
6.5.3	С использованием физического объекта	122
6.5.4	С использованием биометрических данных	122
6.6	Атаки системы изнутри	123
6.6.1	Троянские кони	123
6.6.2	Фальшивые программы регистрации	123
6.6.3	Логические бомбы	124
6.6.4	Потайные двери	124
6.7	Атаки системы извне	124
6.7.1	Как вирус причиняет ущерб	125
6.7.2	Как работает вирус	125
6.7.3	Разновидности вирусов	125
6.7.4	Как распространяются вирусы	127
6.8	Антивирусы	127
6.8.1	Сканеры	128
6.8.2	Проверка целостности	128
6.8.3	Проверка поведения	128
6.9	Предохранение от вирусов	129
6.10	Восстановление после вирусной атаки	129
6.11	Резюме	130
7	Языки программирования	132
7.1	Исторический обзор	132
7.2	Классификация языков программирования	138
7.3	Системы программирования	139
7.4	Какой язык программирования лучше?	141
	Заключение	144
	Литература	145
	Глоссарий	147
	Предметный указатель	156

ВВЕДЕНИЕ

Если из-за бедности моего ума что-нибудь написано неверно или неточно — а иногда мой ум посещают лукавые мысли, — Вы, которых господь благословил умом, чтобы понять это писание, когда читаете, благословляйте, а не проклинаяте, и с Вашей помощью мои ошибки будут исправлены, и, может быть, Господь избавит меня от геенны в тот день, который наступит, и скажет: «Да воздастся Вам по делам Вашим».

*Неизвестный автор.
«Приписка в октоихе¹», XIV век.*

В наше время от любого человека, независимо от его профессии и жизненных интересов, требуется больше самостоятельности, инициативы, профессионализма. Любой компетентный специалист должен не столько обладать большими знаниями, сколько уметь находить их и использовать, в том числе с помощью информационных и коммуникационных технологий (ИКТ). Познакомить с ИКТ и научить использовать их для решения различных проблем, и есть задача дисциплины «информатика».

История появления и развития информатики берет свое начало в 50-х годах XX столетия. Именно в этот период родилась наука — кибернетика. Новая дисциплина развивалась. Охватывала все больше направлений в науке и технике. Но со временем, совокупность научных направлений, которые она в себя включала, стали называть другим словом — «информатика».

В шестидесятых годах двадцатого столетия появилось два научных направления, которые назывались одним и тем же термином — «информатика». Одно из них пришло на смену дисциплине «Теория научной информации», которая занималась изучением структуры и общих свойств научной информации. Появление второго направления связано с развитием вычислительной техники.

Таким образом, длительное время просуществовали две разные научные дисциплины, для обозначения которых использовался один и тот же термин. Однако

¹Книга церковных песнопений.

обе науки имели большие и быстро увеличивающиеся области пересечения, что, конечно же, не могло не привести к их объединению.

В наше время информатика развивается в трех направлениях: техническая, социальная и биоинформатика. Без данного предмета не может обойтись ни среднее образование, ни высшее. При этом основной упор делается на изучение технической (прикладной) информатики, которая рассматривает принципы и методы функционирования технических средств: вычислительной техники, средств телекоммуникаций, организационной техники.

Первая часть курса «Информатика» разбита на семь частей.

В первой главе определяется понятие информации и ее свойства. Рассказываются о способах кодирования различной информации.

Во второй главе Вы познакомитесь с историей развития вычислительной техники, устройством персонального компьютера, видами и назначением периферийных устройств.

Третья глава посвящена описанию программного обеспечения, которое необходимо для работы компьютера.

Глава четвертая посвящена компьютерным сетям — их назначению, классификации. Здесь также рассмотрены основные принципы работы глобальной сети Интернет, рассказано про зарождение и становление данной сети, а также об ее сервисах.

В пятой главе рассказывается о том, что такое базы данных. Вводятся основные определения. Описывается программное обеспечение, необходимое для создания и управления БД.

Шестая глава посвящена вопросам защиты информации.

В седьмой главе описаны история появления языков программирования.

Курс «Информатика. Часть 1» изучается по пособиям:

Тимченко С.В. Информатика. Часть 1

Лазичев А. А. Гураков А. В. Информатика. Введение в Microsoft Office

Также к Вашим услугам презентации, видеоролики, электронный учебник на сайте www.fdo.tusur.ru/online

В течение семестра Вам будет необходимо освоить теоретический материал, выполнить две компьютерные контрольные работы и поучаствовать в семинаре. После усвоения теоретического материала и успешной сдачи контрольных работ, Вам предстоит сдать электронный экзамен.

Дисциплина «Информатика» является базовой. Для ее изучения достаточно знаний, опирающихся на материал школьных курсов информатики и математики.

Соглашения, принятые в книге

Для улучшения восприятия материала в данной книге используются пиктограммы и специальное выделение важной информации.



.....
Эта пиктограмма означает определение или новое понятие.
.....



.....
Эта пиктограмма означает внимание. Здесь выделена важная информация, требующая акцента на ней. Автор здесь может поделиться с читателем опытом, чтобы помочь избежать некоторых ошибок.
.....



.....
Эта пиктограмма означает цитату.
.....



.....
Контрольные вопросы по главе
.....

Глава 1

ОСНОВНЫЕ ПОНЯТИЯ ИНФОРМАТИКИ

В данной главе мы постараемся разобраться, что изучает наука информатика, что такое информация и какими свойствами она обладает. Рассмотрим основные способы кодирования данных различного типа.

1.1 Что такое информатика?

Информатика¹ — это основанная на использовании компьютерной техники дисциплина, изучающая структуру и общие свойства информации, а также закономерности и методы ее создания, хранения, поиска, преобразования, передачи и применения в различных сферах человеческой деятельности.

Термин «информатика» широко распространен в ряде стран Восточной Европы. В то же время, в большинстве стран Западной Европы и США распространен другой термин — «*Computer science*», что означает буквально «компьютерная наука».

Информатика — дисциплина развивающаяся. Это является очень существенным фактом. Если математикой, физикой или биологией человечество занимается несколько веков, то об информатике заговорили только в середине XX века, когда на смену механическим средствам обработки информации пришли электронные. Они позволяли во много раз сократить время сбора, обработки и передачи информации. Это явилось началом новой технологии — компьютерной. Можно считать, что родилась наука информатика.

Сам термин «информатика» появился в 1962 году практически одновременно во Франции (Ф. Дрейфус) и у нас в стране (А. А. Харкевич). В 1963 г. в журнале «Известия вузов. Электромеханика», №11 была опубликована статья Ф. Е. Темникова «Информатика». В ней была сделана попытка определить состав интеграль-

¹informatique (франц.) — происходит от двух французских слов information (информация) и automatique (автоматика).

ной науки об информации, как совокупность трех составных частей — теории информационных элементов, теории информационных процессов и теории информационных систем. Однако впоследствии закрепился французский (более узкий) вариант трактовки термина «информатика» (от французского *informatique*), как науки об ЭВМ и их применении.

Все же первоначально под информатикой у нас понимали науку, связанную, прежде всего, с научной или научно-технической информацией, и определяли как *«научную дисциплину, изучающую структуру и общие свойства научной информации, а также закономерности всех процессов научной коммуникации»*.

В современной информатике можно выделить три основных направления:

Техническая (и/или прикладная) информатика. Она изучает принципы и методы функционирования и построения технических средств информатики — вычислительной техники, средств телекоммуникаций, оргтехники, а также прикладные основы создания информационных технологий.

Социальная информатика. Изучает общие закономерности информационного взаимодействия в обществе, включая проблемы социальной коммуникации, формирования информационных ресурсов и информационного потенциала общества, информатизации общества, особенностей информационного общества. Здесь же рассматриваются междисциплинарные проблемы типа «информатика-искусство» (музыка, живопись, архитектура, кино) и «информатико-социокультурные системы» (психология, социология, юриспруденция, педагогика), экономические, правовые, психологические, этические аспекты информатики.

Биоинформатика. Рассматривает общие закономерности и особенности протекания информационных процессов в объектах биосферы (живых организмах и растениях).

Если рассматривать информатику как теоретическую и прикладную междисциплинарную науку, то в данном случае можно выделить восемь направлений:

Теоретическая информатика. Эта дисциплина тесно связана с математикой, поскольку использует ее методы для построения и изучения моделей обработки, передачи, приема и использования информации.

Кибернетика. Наука об общих закономерностях процессов управления и передачи информации в различных системах, будь то машины, живые организмы или общество. Свое рождение ведет с 1948 года, когда английский ученый-математик Норберт Винер опубликовал свою книгу «Кибернетика или управление и связь в животном и машине». В ней автор выдвинул идею, что системы управления в живых, неживых и искусственных организмах обладают общими чертами.

Программирование. В широком смысле — это процесс подготовки и составления программы деятельности, выполнение которой должно привести к определенным целям. В прикладной информатике программирование сводится к процессу подготовки задач для их решения с помощью компьютера.

Искусственный интеллект. Этот раздел информатики занимается вопросами имитации мышления человека с помощью компьютера. Начало исследований в области искусственного интеллекта связывают с работами Аллена Ньюэлла, Герберта Саймана и Клиффа Шоу, исследовавших процессы решения различных задач.

Информационные системы. Начало этому направлению положили исследования в области анализа научно-технической документации еще до появления ком-

пьютеров. Сейчас в рамках этого направления решается несколько основных задач: исследование способов представления и хранения информации; изучение потоков документов с целью их минимизации, стандартизации и эффективной обработки; создание информационно-поисковых систем и т. д.

Вычислительная техника. Вычислительная техника прошла те же исторические этапы эволюции, которые прошли и все прочие технические устройства: от ручных приспособлений к механическим устройствам и далее к гибким автоматическим системам. Современный компьютер — это прибор, без которого невозможно развитие современной информатики. С другой стороны, развитие и эффективное использование компьютеров невозможно без знания их архитектуры и принципов функционирования. Разработка современной элементной базы вычислительных машин, создание программного обеспечения требует знания теоретической информатики, программирования и других разделов информатики.

Общественная информатика. Современное общество можно назвать информационным. В этом разделе информатики изучаются взаимоотношения процессов информатизации и человека.

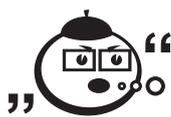
Информатика в природе. Это направление изучает информационные процессы, протекающие в биологических системах.

Как видим понятие информатики многозначно. Существует, по крайней мере, четыре содержательных понимания термина «информатика»: наука, отрасль экономики, сфера человеческой деятельности, технологии. Таким образом, информатика тесно связана с другими науками, прочно вошла во все сферы нашей жизни и является крайне необходимой для познания окружающего мира.

1.2 Информация

Термин «информация» происходит от латинского слова «**informatio**», что означает **сведения, разъяснения, изложение**. Несмотря на то, что понятие информации очень широко используется и в науке, и в повседневной жизни, его строгого научного определения не существует. По сей день разные научные дисциплины вводят это понятие по-разному.

Антропоцентрический (фактологический) подход состоит в том, что информация отождествляется с фактами или со сведениями. Этот подход в настоящее время используется наиболее широко и отражен в российском законодательстве.



.....
 «Под информацией понимаются сведения (сообщения, данные) независимо от формы их представления» (Федеральный Закон №149—ФЗ «Об информации, информационных технологиях и о защите информации» от 27.07.06 г. «Российская газета» №165 от 29.07.06 г.)

Основное достоинство фактологического подхода состоит в его простоте и понятности. Он очень хорошо согласуется с бытовым представлением об информации как об источнике знаний.

Основной недостаток антропоцентрического подхода заключается в том, что он в принципе не способен адекватно описать информационные процессы, действующие в живой природе и технике.

Техноцентрический подход состоит в том, что информацию отождествляют с данными. Этот подход нашел очень широкое распространение в технических дисциплинах. Например, нам часто встречаются упоминания о том, что «информация передается по компьютерным сетям», «информация обрабатывается компьютерами», «информация хранится в базах данных». Во всех этих случаях происходит подмена понятий. Дело в том, что по компьютерным сетям передаются только данные, компьютеры обрабатывают только данные, а в базах данных хранятся тоже только данные. Станут ли эти данные информацией и если да, то какой, зависит не только от данных, а и от многочисленных аппаратных, программных и естественных методов.

Вероятностный подход — модельный. Область применения данного подхода сильно ограничена. Он используется в информационных системах, протекающих в *закрытых системах*. Закрытыми называют системы, в которых источник и потребитель информации действуют по ранее согласованным правилам. Подключение нового источника или изменение правил в таких системах недопустимо.

Согласно вероятностному подходу, информация — это способ уменьшить неполноту знаний (неопределенность) приемника о состоянии источника.

Постараемся дать научное определение термину «информация», которое бы объединило все эти подходы. Для этого представим, что информация — это динамический объект, не существующий в природе сам по себе, а образующийся в результате взаимодействия данных и методов их представления и обработки.



.....
Информация — это продукт взаимодействия данных и методов их обработки, рассмотренный в контексте этого взаимодействия.

В данном определении важным является пояснение «...*рассмотренный в контексте этого взаимодействия*». Для более глубокого понимания важности этой фразы приведем пример.

Возьмите с полки любую книгу. В ней напечатан некоторый текст, содержащий определенные данные. Используя метод чтения, Вы можете получить определенную информацию из этой книги.

Однако можно закрыть глаза и просто потрогать книги, стоящие на полке. В этом случае Вы также получите некоторую информацию о толщине книги, структуре и типу ее обложки и т. д. (осязательный метод).



.....
Контекст — это логический (нематериальный) компонент, который влияет на содержание информации и обобщает условия и цели ее получения.

1.3 Свойства информации

Информация является динамическим объектом и, так же как всякий объект, обладает определенными свойствами.

Адекватность

Под адекватностью понимают степень соответствия информации, полученной потребителем, тому, что автор вложил в ее содержание (то есть в данные).

Так как мы считаем, информация есть результат взаимодействия данных и методов их обработки, то адекватность информации напрямую зависит от адекватности как данных, так и методов. Покажем это на примере. На рисунке 1.1 приведены некоторые данные, причем и в первом и во втором случае информация, заложенная в них, одинакова.

В первом случае, применив метод наблюдения и простого счета, получим информацию о том, что $8+8=16$. Данная информация вполне адекватна имеющимся данным.

$$8+8=16$$
$$8+8=10$$

Рис. 1.1 – Получение адекватной информации с использованием адекватных методов

Теперь воспользуемся такими же методами для получения информации во втором случае. Получается, что данные, а следовательно, и информация, не выглядят адекватными. Однако стоит сменить метод и перейти к другой системе счисления¹, мы вновь получим адекватную информацию.

Часто адекватность путают с другими свойствами информации, например с достоверностью. Рассмотрим заведомо ложное (недостоверное) объявление

Требуется бухгалтер. Режим работы: год через три.

Если поместить его в юмористический журнал (или газету), то его можно считать адекватным. Размещение его в другом, серьезном, печатном издании делает его неадекватным.

Примеры разного подхода к адекватности информации можно найти в законодательстве. Закон различает права свидетелей и подозреваемых. В то время как сообщение заведомо ложных данных подозреваемым считается адекватным поведением, те же действия со стороны свидетелей адекватными не являются и рассматриваются как правонарушение.

¹Подробнее об этом мы расскажем позднее.

Достоверность

Достоверность — это свойство информации не иметь скрытых ошибок, т. е. это характеристика ее неискаженности.

Достоверность информации напрямую зависит от достоверности данных и адекватности методов, используемых при их обработке.

Недостоверную информацию можно получить в результате использования изначально ложных данных (например, оставление на месте преступления предметов с чужими отпечатками пальцев), модифицированных данных (удаление отпечатков пальцев с орудия преступления) или данных, которые тяжело выявить на фоне других сигналов (предмет с отпечатками пальцев преступника несет многочисленные отпечатки пальцев посторонних лиц).

Используя неадекватные методы, можно также получить недостоверную информацию. Например, попытаться измерить массу молекулы водорода с помощью обыкновенных весов.

Кроме как закрытых систем, не существует абсолютно достоверной и абсолютно недостоверной информации. Всегда имеются некоторые информационные искажения, снижающие достоверность. Их называют — *информационный шум*.

Свойство достоверности информации имеет важное значение в тех случаях, когда ее используют для принятия решений. Недостоверная информация может привести к решениям, которые в дальнейшем будут иметь негативные последствия.

Полнота

Полнота информации — это относительная характеристика, определяющая количество информации, собранной об объекте или явлении.

Полнота тесно связана с объективностью. Чем полнее информация, собранная об объекте или явлении, тем выше ее потенциальная объективность.

Свойство полноты очень важно для принятий решений. Оно зависит как от полноты данных, так и от наличия необходимых методов их обработки.

С понятием полноты данных сталкиваются все, например, студенты, когда решают задачи. Если исходные данные к задаче неполные, найти верный ответ нелегко, а бывает, что и невозможно. Однако многим студентам очень хорошо знакомы случаи, когда данные полны, а верное решение никак не находится. Это говорит о том, что они не располагают нужными методами.

Объективность

Объективность — это характеристика информации, выражающая степень ее соответствия реальной жизни.

Понятие объективности является относительным и крайне редко бывает абсолютным. Информация связана как с данными, которые в большинстве своем объективны, так и с информационными методами, а вот они имеют субъективную природу.

Можно говорить о степени объективности информации, она будет тем выше, чем ниже субъективность методов, использованных для получения данной информации. Например, если сделать фотоснимок определенной местности и рисунок

той же местности, при изучении фотографии будет получена более объективная информация, чем при исследовании рисунка.

Актуальность

Актуальность информации — это свойство, характеризующие степень ее соответствия текущему моменту времени.

Информация может стареть. Это явление очень важно учитывать при принятии решений. Иногда достоверная и адекватная информация может привести к ошибке, если к тому времени является устаревшей.

Необходимость поиска (или разработки) адекватного метода для работы с данными может привести к такой задержке в получении информации, что она становится неактуальной и ненужной. Это свойство используется в современных системах шифрования и электронной подписи. Лица, не владеющие ключом (методом) для чтения данных, могут заняться его поиском. Однако это занимает столько много времени, что информация за время поиска теряет свою актуальность и становится бесполезной.

Доступность

Доступность информации — это обобщенное свойство, характеризующие доступность данных и доступность информационных методов, необходимых для воспроизведения этих данных.

Данное свойство характеризует возможность получить ту или иную информацию потребителем. Отсутствие доступа к данным или отсутствие адекватных методов для их обработки делают информацию недоступной.

Доступностью к информации можно управлять: повышать или понижать.

Повышение доступности общественной информации достигается путем развития и совершенствования средств связи (телеграф, телефон, радио, телевидение, Internet. . .).

Доступность научной информации повышается с развитием новых технических информационных методов. Сравните телескоп Галилео Галилея и современные оптические (а есть еще и радио- и рентгеновские) телескопы. Современные телескопы дают гораздо больше информации, хотя источник информации остается один и тот же — звездное небо.

Понижение доступности информации необходимо для сохранения конфиденциальной информации: личных данных, сведений о здоровье, финансовых счетах, а также имеющих отношение к государственной тайне.

Защита информации может осуществляться двумя способами: уменьшением доступности данных и снижением информационных методов. В первом случае используются физические, организационные и технические средства ограничения доступа к данным. Во втором снизить доступность информации позволяет шифрование данных.

1.4 Данные и методы их воспроизведения и обработки

Любое взаимодействие материальных объектов имеет энергетическую природу. Сигналы окружают нас на каждом шагу: солнечный свет, радиосигналы, электромагнитные волны и т. д. и даже колебания почвы во время землетрясений являются сигналами.

Как и все объекты материальной природы, сигналы не возникают из ничего и не пропадают бесследно. Их распространение в пространстве всегда завершается взаимодействием с веществом физических тел. Такое взаимодействие в информатике рассматривается как *регистрация* сигналов.

Результат регистрации сигналов информатика рассматривает как *данные*.

Данные всегда объективны. Их состав и структура существуют независимо от наблюдателя. Вместе с тем содержание объективных данных может интерпретироваться субъективно.

Данные не могут существовать без носителя. Свойства данных, такие как долговечность хранения, скорость перемещения и плотность размещения, определяются свойствами носителя.

Методы воспроизведения и обработки данных

В зависимости от физической природы данных их можно потрогать, послушать, посмотреть, попробовать на вкус и т. д. Другими словами, данные можно каким-либо образом воспроизвести. Для этого необходимо обладать адекватными методами их воспроизведения и обработки.

Используя различные методы, можно на основании одних и тех же данных получить различную информацию. Например, Вы получили записку, написанную от руки. Обладая знанием алфавита, Вы сможете прочесть текст и получить некоторую информацию. Криминалисты же используют другие методы и способны по почерку определить характер человека, написавшего записку, его настроение и душевное состояние в момент ее написания. По бумаге, на которой написана записка, можно определить не только ее происхождение, но и содержимое предыдущих страниц, если конечно таковые были.

В подавляющем числе случаев для получения информации из данных требуется применить не один, а множество взаимосвязанных методов.

Различают *естественные* и *технические* методы воспроизведения и обработки информации (рис. 1.2).

Естественные методы присущи человеку и другим объектам живой природы. Если говорим о человеке, то, прежде всего, к естественным относятся методы, основанные на его органах чувств (зрение, вкус, осязание, обоняние и слух).

Для того, чтобы работать с данными, не имеющими объективных аналогов в материальном мире, человек обладает таким естественным методом, как *логическое мышление*.

Прочие методы обработки данных, присущие человеку, основаны на особенностях его мышления: *воображение, анализ, сравнение, сопоставление, прогнозирование* и т. д.



Рис. 1.2 – Методы воспроизведения и обработки информации

Многие естественные методы обработки данных присущи и другим живым объектам. Например, метод генетического наследования.

Технические методы можно разделить на *аппаратные* и *программные*.

Технические методы используются в тех случаях, когда физическая природа данных не позволяет использовать для воспроизведения и обработки естественные методы. Например, человек не способен воспринимать электромагнитные волны, поэтому использует специальные устройства, позволяющие принять и преобразовать сигнал в понятную для человека форму.

Аппаратные методы — это всегда устройства (приборы). Их функция заключается в том, чтобы преобразовать сигналы из одной формы, недоступной для обработки естественными методами, в форму, доступную для них.

Широкое внедрение средств вычислительной техники позволяет автоматизировать обработку самых разных видов данных с помощью компьютеров. Компьютер — это прибор особого типа, в котором одновременно сочетаются аппаратные и программные методы обработки и представления информации.

Носителем данных может быть любое физическое тело. Регистрация (запись) данных осуществляется самыми различными способами, связанными с перемещением тел, изменением их формы и качества поверхности, изменением электрических, магнитных и оптических характеристик, изменением химического состава или характера химических связей и т. д.

Носители данных

На сегодня самым популярным носителем данных является бумага. Для записи данных мы, с помощью специальных устройств (ручка, карандаш и т. д.), наносим на бумагу некоторое вещество (чернила, графит и т. д.), имеющее иной коэффициент отражения в оптическом диапазоне длин волн (т. е. другого цвета).

В наше время мы часто сталкиваемся с тем, что данные имеют электронный вид. Поэтому, наравне с бумагой используются такие носители данных, как CD-диск (для регистрации изменяются оптические свойства), дискета, лента (изменяются магнитные свойства). Регистрация данных путем изменения химического состава широко используется в фотографии.

Любой носитель информации можно охарактеризовать двумя параметрами:

- *Разрешающая способность* — максимальное количество данных, которые можно записать в принятой для носителя единице измерения;
- *Динамический диапазон* — логарифмическое отношение интенсивности амплитуд максимального и минимального регистрируемого сигнала.

Эти параметры могут существенно повлиять на некоторые, свойства информации, такие, например, как полнота или доступность. Размещая некоторую базу данных на компакт-диске, намного легче обеспечить полноту информации, чем в аналогичной по назначению базе данных, помещенной на гибком магнитном диске. Информация, получаемая из книги, более доступна, чем информация на компакт-диске, так как не все обладают необходимыми для ее получения методом.

Операции с данными

С развитием научно-технического прогресса существенно увеличивается объем обрабатываемых данных. Это связано с усложнением управления производством и обществом. Так же очень быстро появляются и внедряются новые устройства для хранения и передачи данных. Весь этот объем данных требует быстрой и верной обработки, которая включает в себя множество операций. Выделим основные:

- *сбор данных* — накопление данных с целью обеспечения достаточной полноты информации для принятия решений;
- *формализация данных* — приведение данных, поступающих из разных источников, к одинаковой форме, чтобы сделать их сопоставимыми между собой, то есть повысить их уровень доступности;
- *фильтрация данных* — отсеивание «лишних» данных, в которых нет необходимости для принятия решений; при этом должен уменьшаться уровень «шума», а достоверность и адекватность данных должны возрасти;
- *сортировка данных* — упорядочение данных по заданному признаку с целью удобства использования; повышает доступность информации;
- *группировка данных* — объединение данных по заданному признаку с целью повышения удобства использования; повышает доступность информации;
- *архивация данных* — организация хранения данных в удобной и легкодоступной форме; служит для снижения экономических затрат на хранение данных и повышает общую надежность информационного процесса в целом;
- *защита данных* — комплекс мер, направленных на предотвращение утраты, воспроизведения и модификации данных;
- *транспортировка данных* — прием и передача данных между удаленными участниками информационного процесса; при этом источник данных в информатике принято называть *сервером*, а потребителя — *клиентом*;

- *преобразование данных* — перевод данных из одной формы в другую или из одной структуры в другую. Преобразование данных часто связано с изменением типа носителя, например книги можно хранить в обычной бумажной форме, но можно использовать для этого и электронную форму, и микрофотопленку.

Данный список далеко не полон. Он отражает только общий подход к обработке данных. На любом производстве существуют свои специфические операции. Поэтому полный список составить в принципе невозможно, да и не нужно. Важно понимать, что процедура обработки данных очень трудоемкая и в ней нельзя обойтись без некоторой, а то и полной автоматизации¹.

1.5 Кодирование информации

Перед записью информации на носитель требуется их определенным образом преобразовать в данные. Для этого используется операция *кодирования*.



.....
Кодирование — это управляемый процесс представления элементов информационных объектов элементами данных.

Обратите внимание на слово — *управляемый*. Оно говорит о том, что процесс кодирования не случайный и происходит по определенному закону с использованием адекватного носителю метода.

Различают три схемы кодирования: аналоговое, табличное и цифровое (рис. 1.3).



Рис. 1.3 – Классификация схем кодирования

При аналоговом кодировании происходит изменение физической природы данных. Делается это с целью увеличения плотности записи, надежности хранения, скорости перемещения и других свойств данных.

Примером аналогового кодирования может служить фотография, сделанная фотоаппаратом, использующим фотопленку.

¹Например, в связи с постоянным ростом числа студентов и увеличением компьютерных контрольных работ, в ФДО, с целью ускорить проверку и регистрацию зачетов, процедура анализа протоколов полностью автоматизирована.

Табличное кодирование основано на периодическом сравнении элементов сигнала с имеющимися модельными образцами (символами), собранными в специальные таблицы. В результате кодирования получается дискретная последовательность, которую называют *выборкой данных*.

Различают *таблично-символьное* и *таблично-цифровое* кодирование. В первом случае из таблицы берутся сами образцы, и выборка данных представляет собой последовательность символов. При таблично-цифровом кодировании выборка составляется из чисел, обозначающих местоположение символов в таблице кодирования.

Ярким примером таблично-символьного метода кодирования является — письменность. В ее основе лежит запись звуков с помощью специальных символов — букв.

При аналоговом методе сигнал регистрируется на протяжении всего интервала времени, на котором он существует. При цифровом методе кодирования сигнал измеряется через определенные моменты времени, в большинстве случаев — одинаковые, результат измерения записывается в числовом виде.

Цифровое кодирование наиболее эффективно, если для обработки данных используется ЭВМ, поэтому будем говорить в основном об этих способах, но сначала поговорим о цифрах.

1.5.1 Системы счисления

В вычислительной технике для записи, передачи, обработки и т. д. используются не привычные для нас цифры (1, 2, 3, ...) или слова, а двоичные цифры. Другими словами, используется иная система счисления.

В жизни мы часто имеем дело с числами. С помощью чисел можно закодировать практически любую информацию. Все числа состоят из символов, которые называются *цифрами*. Способ, с помощью которого из цифр составляют числа, называется — системой счисления.



.....
Система счисления (СС) — это совокупность приемов и правил записи чисел цифровыми знаками.

Все системы счисления можно разделить на две группы: позиционные и непозиционные.



.....
Непозиционными системами счисления (НСС) называют системы, в которых значение символа (цифры) не зависит от его положения в числе.

Непозиционные системы счисления очень широко использовались в древности. Но некоторые из них дошли до наших времен и не просто дошли, а широко используются. Например, римская система счисления. В данной системе используются следующие цифры: I — 1 (единица); V — 5 (пять); X — 10 (десять); L — 50 (пятьдесят); C — 100 (сто); D — 500 (пятьсот); M — тысяча.

Для записи чисел в римской системе счисления используется три правила:

- 1) каждый меньший знак, поставленный слева от большего, вычитается из него;
- 2) каждый меньший знак, поставленный справа от большего, прибавляется к нему;
- 3) одинаковые знаки, поставленные рядом, складываются.

Например, IV — четыре и VI — шесть. Обратите внимание, что, независимо от того, где находится символ «I», слева или справа от «V», его значение не изменяется.



.....
Позиционная система счисления (ПСС) — это система счисления, в которой значение символа зависит от его положения в ряду цифр, изображающих число.

Позиционные системы счисления называют так же *взвешенными*. Положение цифры в числе называют — *разрядом*. Все разряды нумеруются справа налево, начиная с нуля¹.

Основной параметр, характеризующий позиционную систему счисления, — ее *основание*.



.....
Основание ПСС — это количество символов, используемых в разрядах для отображения числа. Обозначается буквой — q .

Название системы счисления определяется ее основанием. Если основание равно двум, система называется двоичной, трем — троичной, десяти — десятичной и т. д.

В быту мы используем десятичную систему счисления (основание равно десяти). И используем для записи чисел десять цифр: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Рассмотрим любое десятичное число:

$$748_{10} = 7 \cdot 10^2 + 4 \cdot 10^1 + 8 \cdot 10^0$$

Как видно, число 748 состоит из трех разрядов²: в нулевом разряде стоит цифра восемь; в первом — четыре; во втором — семь. С правой стороны числа добавлен индекс — 10. Он обозначает основание системы счисления, в которой это число записано. Данный индекс обязателен, если имеется даже намек на то, что число может быть записано в другой, отличной от десятичной, системе счисления.

Из курса математики третьего (а может быть и второго) класса мы знаем, что цифра семь означает сотни, четыре — десятки и восемь — единицы. В результате

¹Для целых чисел. Для вещественных чисел нулевым считается крайний справа разряд в целой части.

²Над каждой цифрой стоит номер ее разряда. Сделано это только для удобства понимания материала. Однако если Вы только сейчас начинаете знакомиться с системами счисления, наш совет, во избежание ошибок, проставлять номер разряда над каждой цифрой, «карандашиком».

число можно представить в виде суммы цифр, умноженных на определенное число. Это число называется *весом разряда* и определяется по формуле:

$$p_i = \frac{q^i}{q^0},$$

где i — номер разряда.

Видно, что вес каждого разряда равен целой степени числа десять, т. е. основания данной системы счисления.

$${}^{210}_{748}_{10} = 7 \cdot 100 + 4 \cdot 10 + 8 \cdot 1 = 7 \cdot 10^2 + 4 \cdot 10^1 + 8 \cdot 10^0$$

Обратите внимание. Цифра восемь умножается на основание системы счисления в степени, равной номеру разряда, в котором располагается эта цифра. То же самое можно сказать и про остальные цифры.

Любое число в позиционной системе счисления с основанием q можно представить в виде суммы произведений цифр и основания системы счисления в степени, равной разряду этой цифры.

$$X_q = a_n q^n + a_{n-1} q^{n-1} + \dots + a_1 q^1 + a_0 q^0 = \sum_{i=0}^N a_i q^i, \quad (1.1)$$

где a_i — цифра, расположенная в разряде i ; N — номер старшего разряда.

Формула (1.1) очень важна при переводе чисел из любой системы счисления с основанием q в десятичную систему счисления.

Для представления данных в компьютерах и других электронных вычислительных устройствах используется двоичная система счисления. Из названия следует, что ее основание равно двум ($q = 2$), следовательно, для записи чисел используются только две цифры: 0 и 1.

Выбор двоичной системы счисления обусловлен двумя причинами. Первая причина аппаратная. Создать устройство, которое может находиться в двух устойчивых состояниях и быстро переключаться из одного состояния в другое, очень просто и дешево. Вторая причина заключается в том, что на основе двоичной системы счисления очень просто реализовать алгебру логики.

Однако у двоичной системы счисления есть существенный недостаток. Числа, записанные в этой системе, очень большие, и человеку очень трудно с ними работать и запоминать¹. Поэтому находят применение системы счисления с основанием больше двух. А чтобы переход от данной системы счисления к двоичной был наиболее прост, ее основание должно являться целой степенью двойки.

В соответствии с этими соображениями становится понятно, почему в компьютерной технике наряду с двоичной системой счисления используется шестнадцатеричная система счисления. Из названия следует, что основание системы равно шестнадцати, т. е. в разрядах числа может стоять одно из шестнадцати значений: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

В таблице, представленной ниже, приведены шестнадцать десятичных чисел и их двоичный и шестнадцатеричный коды.

¹Число 243_{10} в двоичной системе счисления выглядит следующим образом 11110011_2 , а в шестнадцатеричной — $F3_{16}$.

Таблица 1.1

Десятичная	Двоичная	Шестнадцатеричная
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Перевод чисел из системы счисления с основанием q в десятичную систему счисления

Перевести числа в десятичную систему счисления очень просто. Делается это с помощью формулы (1.1). То есть каждая цифра умножается на основание системы счисления q , которое возведено в степень, равную разряду, в котором находится эта цифра. Затем все произведения складываются. Разберем пример.

Пусть требуется перевести число 1011_2 в десятичную систему счисления. Это число записано в двоичной системе счисления, об этом говорит нижний правый индекс, следовательно, $q = 2$. Проставим над каждой цифрой соответствующий ей разряд¹.

$$\begin{array}{c} 3 \ 2 \ 1 \ 0 \\ 1011_2 \end{array}$$

Первую слева направо цифру умножаем на основание системы счисления в кубе ($1 \cdot 2^3 = 8$). Вторую — на основание в квадрате ($0 \cdot 2^2 = 0$), третью — на основание в первой степени ($1 \cdot 2^1 = 2$) и последнюю — на основание в нулевой степени ($1 \cdot 2^0 = 1$)². То, что получилось, складываем:

$$\begin{array}{c} 3 \ 2 \ 1 \ 0 \\ 1011_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 8 + 0 + 2 + 1 = 11_{10} \end{array}$$

Получили, что двоичное число 1011_2 соответствует десятичному числу 11_{10} (правильность решения можно проверить по таблице 1.1).

¹В принципе это можно сделать мысленно, но если Вы боитесь ошибиться, смело пишите сверху цифры.

²Любое число, возведенное в нулевую степень, равняется единице (кроме самого нуля).

Приведем еще несколько примеров.

- 1) Перевести число 10110110_2 в десятичную систему счисления.

$$\begin{array}{r} 76543210 \\ 10110110_2 = 1 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + \\ + 1 \cdot 2^1 + 0 \cdot 2^0 = 128 + 0 + 32 + 16 + 0 + 4 + 2 + 0 = 182_{10} \end{array}$$

- 2) Перевести число 123_{16} в десятичную систему счисления.

$$123_{16} = 1 \cdot 16^2 + 2 \cdot 16^1 + 3 \cdot 16^0 = 256 + 32 + 3 = 291_{10}$$

- 3) Перевести число 123_7 в десятичную систему счисления.

$$123_7 = 1 \cdot 7^2 + 2 \cdot 7^1 + 3 \cdot 7^0 = 49 + 14 + 3 = 66_{10}$$

- 4) Перевести число $1A1D_{16}$ в десятичную систему счисления.

$$\begin{array}{r} 3210 \\ 1A1D_{16} = 1 \cdot 16^3 + A \cdot 16^2 + 1 \cdot 16^1 + D \cdot 16^0 = \\ = 1 \cdot 4096 + 10 \cdot 256 + 1 \cdot 16 + 13 \cdot 1 = 6685_{10} \end{array}$$



.....
Внимание! Очень часто при решении таких задач студенты совершают следующую ошибку. Заменяют шестнадцатеричные цифры (в данном случае A и D) десятичным эквивалентом, а затем нумеруют разряды. Это неправильно! Сначала следует записать сумму, а уже затем производить замену.

- 5) Перевести число 1021_3 в десятичную систему счисления.

$$\begin{array}{r} 3210 \\ 1021_3 = 1 \cdot 3^3 + 0 \cdot 3^2 + 2 \cdot 3^1 + 1 \cdot 3^0 = 27 + 0 + 6 + 1 = 34_{10} \end{array}$$

Перевод чисел из десятичной системы счисления в любую другую систему счисления с основанием q

Существует простой способ перевода чисел из десятичной системы счисления в другие системы счисления с основанием q . Этот метод основан на делении с остатком. Ниже приведен алгоритм перевода.



.....
Шаг 1. Разделите число на основание новой системы счисления q . Остаток от деления запомните.

Шаг 2. Сравните частное от деления с основанием новой системы счисления q .

Шаг 3. Если частное от деления больше q , делим его на q . Остаток от деления запоминаем. Переходим ко второму шагу.

Шаг 4. Если частное от деления меньше q , записываем последнее частное и все остатки от деления, начиная с последнего.

Проиллюстрируем, как работает данный алгоритм на примере. Пусть требуется перевести число 13_{10} в двоичную систему счисления ($q = 2$) (рис. 1.4, а).

Делим число тринадцать на два. Получаем шесть и один в остатке. Остаток запоемним.

Частное меньше двух, следовательно, делим его на два. Получаем три и ноль в остатке. Ноль запоминаем.

Получили, что новое частное меньше двух, поэтому делим его на два. Получим один и один в остатке.

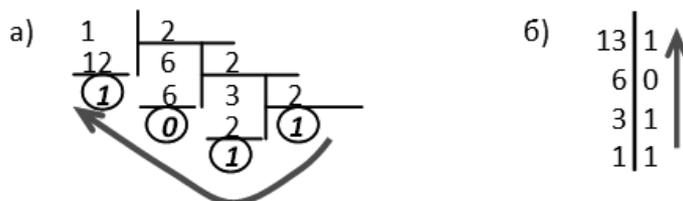


Рис. 1.4 – Перевод числа в двоичную систему счисления

Частное меньше двух. Теперь можно записать результат. Сначала записываем результат последнего деления — единицу (1). Справа от нее ставим значение остатка от последнего деления — единицу (11). Теперь справа добавим остаток от второго (предпоследнего) деления — ноль (110). И в завершение добавляем остаток от первого деления — единицу (1101).

Получили, что число 13_{10} в двоичной системе счисления записывается следующим образом — 1101_2 . Проверка:

$$1101_2 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 8 + 4 + 0 + 1 = 13_{10}$$

Число переведено правильно.

На рисунке 1.4, а показана традиционная запись деления — уголком. Такая форма записи не совсем удобна, особенно в тех случаях, когда требуется оформить решение задачи на компьютере, например в MS Word. Поэтому мы предлагаем вторую запись процедуры перевода числа, показанную на рисунке 1.4, б. Такая форма более компактная и понятная.

В левой части записывается исходное число. Затем делим его на два. Ниже записываем частное от деления, а справа остаток. Теперь делим частное и помещаем остаток в правой колонке, а новое частное — ниже предыдущего и т. д. Самый последний результат деления тот, что меньше основания $q = 2$, копируем в правый столбик. Результат перевода записан в правой колонке снизу вверх.

Приведем несколько примеров.

- 1) Перевести число 20_{10} в шестнадцатеричную систему счисления. Основание шестнадцатеричной системы счисления равно 16, поэтому делить следует на это число.

$$\begin{array}{r|l} 20 & 4 \\ 1 & 1 \end{array}$$

Ответ: $20_{10} = 14_{16}$

- 2) Перевести число 218_{10} в шестнадцатеричную и двоичную системы счисления.

$$\begin{array}{r|l}
 218 & 10 \text{ (A)} \\
 13 & 13 \text{ (D)} \\
 \hline
 218 & 0 \\
 109 & 1 \\
 54 & 0 \\
 27 & 1 \\
 13 & 1 \\
 6 & 0 \\
 3 & 1 \\
 1 & 1
 \end{array}$$

Ответ: $218_{10} = DA_{16} = 11011010_2$



.....
Обратите внимание! Число 218_{10} занимает три разряда. Для записи этого же числа в двоичной системе счисления требуется 8 разрядов, а в шестнадцатеричной — 2 разряда. То есть чем меньше основание системы счисления, тем больше требуется разрядов для записи числа.

- 3) Перевести число 1271_{10} в шестнадцатеричную и двоичную системы счисления.

$$\begin{array}{r|l}
 1271 & 1 \\
 635 & 1 \\
 317 & 1 \\
 158 & 0 \\
 79 & 1 \\
 39 & 1 \\
 19 & 1 \\
 9 & 1 \\
 4 & 0 \\
 2 & 0 \\
 1 & 1
 \end{array}
 \qquad
 \begin{array}{r|l}
 1271 & 7 \\
 79 & 15 \text{ (F)} \\
 4 & 4
 \end{array}$$

Ответ: $1271_{10} = 10011110111_2 = 4F7_{16}$

- 4) Перевести число 127_{10} в семеричную систему счисления. Основание семеричной системы счисления равно семи. Следовательно, необходимо делить на семь.

$$\begin{array}{r|l}
 127 & 1 \\
 18 & 4 \\
 2 & 2
 \end{array}$$

Ответ: $127_{10} = 241_7$

Перевод чисел из двоичной системы счисления в шестнадцатеричную систему счисления

Иногда требуется найти шестнадцатеричный код двоичного числа. В этом случае перевод осуществляется очень легко, всего в два этапа.

- 1) Разбиваем двоичное число на тетрады¹ справа налево. Если количество разрядов не кратно четырем, то можно слева к двоичному числу добавить нули.
- 2) Каждую тетраду заменить ее шестнадцатеричным эквивалентом. Для этого можно использовать данные из таблицы 1.1.

Приведем несколько примеров.

- 1) Перевести число 10111000110100_2 в шестнадцатеричную систему счисления. Разбиваем двоичное число на тетрады. Так как разрядов 14, слева добавим два нуля.

$$\underbrace{0010}_{2} \underbrace{1110}_{E} \underbrace{0011}_{3} \underbrace{0100}_{4}$$

Заменим каждую тетраду ее шестнадцатеричным эквивалентом.

$$\underbrace{0010}_{2} \underbrace{1110}_{E} \underbrace{0011}_{3} \underbrace{0100}_{4}$$

Ответ: $10111000110100_2 = 2E34_{16}$.

- 2) Перевести число 101100000010100_2 в шестнадцатеричную систему счисления. Разбиваем двоичное число на тетрады. Так как разрядов 15, слева добавим один нуль.

$$\underbrace{0101}_{5} \underbrace{1000}_{8} \underbrace{0001}_{1} \underbrace{0100}_{4}$$

Заменим каждую тетраду ее шестнадцатеричным эквивалентом.

$$\underbrace{0101}_{5} \underbrace{1000}_{8} \underbrace{0001}_{1} \underbrace{0100}_{4}$$

Ответ: $101100000010100_2 = 5814_{16}$.

Перевод чисел из шестнадцатеричной системы счисления в двоичную систему счисления

Для перевода чисел в двоичную систему счисления из шестнадцатеричной следует заменить каждую цифру тетрадой ее двоичного эквивалента. Для этого можно воспользоваться данными из таблицы 1.1.

Приведем несколько примеров.

- 1) Перевести число $9ED_{16}$ в двоичную систему счисления.

Используя данные таблицы 1.1, заменим: цифру 9 на число 1001_2 ; цифру E — на число 1110_2 ; цифру D — на число 1101_2 .

Ответ: $9ED_{16} = 10011101101_2$.

¹Тетрада группа из четырех символов.

2) Перевести число $A17D_{16}$ в двоичную систему счисления.

Используя данные из таблицы 1.1, заменим: цифру A на число 1010_2 ; цифру 1 — на число 0001_2 ; цифру 7 — на число 0111_2 ; цифру D — на число 1101_2 .

Ответ: $A17D_{16} = 1010000101111101_2$.

1.5.2 Единицы измерения данных

Данные, которые образуются при записи информации на электронные носители, хранятся в двоичном виде. Один разряд двоичного числа можно считать минимальным количеством данных. Он получил название — *бит*¹.

Бит очень мелкая единица, с которой не всегда легко работать, поэтому используются более крупные единицы. Наиболее устоявшейся единицей измерения данных в вычислительной технике используется *байт*. Байт состоит из восьми бит.

Используются и более крупные единицы измерения. *Килобайт* (Кбайт) = 1024 байт = 2^{10} байт. В физике и математике принято считать, что приставка *кило-* перед названием единицы измерения обозначает более крупную единицу, отличающуюся от исходной в тысячу раз. В информатике иной подход. Так как для хранения и обработки информации используется двоичная система счисления, то и масштабный множитель связан со степенью двойки. Об этом следует постоянно помнить. Однако не следует слишком увлекаться и переносить данный подход на другие объекты. Иначе в один прекрасный день вы будете считать, что в одном километре 1024 метра.

Количество обрабатываемой информации все время увеличивается и определяется тысячами и миллионами килобайт. Поэтому наравне с килобайтами используются более крупные единицы измерения:

мегабайт (Мбайт) = 1024 Кбайт = 2^{20} байт;

гигабайт (Гбайт) = 1024 Мбайт = 2^{30} байт;

терабайт (Тбайт) = 1024 Гбайт = 2^{40} байт.

1.5.3 Кодирование текстовой информации

Для кодирования текстовой информации используются специальные таблицы. Каждому символу алфавита соответствует определенное целое число. Чаще всего используются восьмиразрядные двоичные числа. С их помощью можно закодировать 256 символов. Этого вполне достаточно, чтобы выразить все символы как английского, так и русского алфавита, знаки препинания, цифры, знаки арифметических операций и некоторые специальные символы.

В 1968 году Национальным институтом стандартизации США был принят стандарт ASCII (American Standard Code for Information Interchange — стандартный код информационного обмена США). Система кодирования ASCII разделена на две страницы: базовую и расширенную. Базовая таблица определяет значения кодов от 0 до 127, расширенная — от 128 до 255.

Первые 32 кода относятся к управляющим. За ними не закреплены никакие символы языков, и предназначены они для управления различными устройствами.

¹Слово «*бит*» происходит от английского слова *bit*, которое является сокращением от словосочетания *binary digit* — двоичная цифра.

Аналогичные системы кодирования текстовых документов были разработаны и в других странах. Например, в 1974 году Государственный комитет по стандартизации утвердил стандарт ГОСТ 19768-74, согласно которому внедрялись сразу две схемы кодирования. Одна из них получила название ГОСТ-альтернативной. Впоследствии корпорация IBM опубликовала эту схему в своем корпоративном стандарте, и схема получила название CP 866 (Code page 866). Другая называлась ДКОИ (двоичный код обмена информацией) и стала основной для сетевых ЭВМ, работающих под управлением операционной системы UNIX. Сегодня данная схема называется КОИ-8 (код обмена информацией, восьмиразрядный). После распада СССР различают отдельные схемы кодирования для России (КОИ-8Р) и Украины (КОИ-8У).

Таблица 1.2 – Базовая таблица кодировки ASCII

32	пробел	48	0	64	@	80	P	96	`	112	p
33	!	49	1	65	A	81	Q	97	a	113	q
34	"	50	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	\$	52	4	68	D	84	T	100	d	116	t
37	%	53	5	69	E	85	U	101	e	117	u
38	&	54	6	70	F	86	V	102	f	118	v
39	'	55	7	71	G	87	W	103	g	119	w
40	(56	8	72	H	88	X	104	h	120	x
41)	57	9	73	I	89	Y	105	i	121	y
42	*	58	:	74	J	90	Z	106	j	122	z
43	+	59	;	75	K	91	[107	k	123	{
44	,	60	<	76	L	92		108	l	124	\
45	-	61	=	77	M	93]	109	m	125	}
46	.	62	>	78	N	94	^	110	n	126	~
47	/	63	?	79	O	95	_	111	o	127	неразрывный пробел

Наглядным примером корпоративного стандарта является схема кодирования Windows-1251, которую ввела в действие корпорация Microsoft. Никакими государственными или международными стандартами эта схема не поддерживается, однако является самой распространенной на компьютерах платформы IBM PC.

Формально для России имеет наивысший приоритет схема кодирования — ISO-8859, поскольку она утверждена Международным институтом стандартизации. Однако на практике документы, использующие эту схему, встречаются редко, особенно на компьютерах платформы IBM PC.

Существование множества таблиц кодирования сильно затрудняет информационный обмен. Приходится создавать программы, которые способны работать с различными кодировками. Это приводит к нерациональному использованию средств. Требуется создать универсальную кодовую таблицу, но это невозможно сделать, если кодировать символы восьмиразрядным кодом.

В конце 80-х годов был создан международный консорциум Unicode, который классифицировал национальные письменные системы и изучил их особенности.

В результате этой работы был разработан международный стандарт кодирования. В его основе лежат три положения:

- 1) Каждый символ имеет уникальное имя.
- 2) Каждый символ имеет уникальный номер, определяющий его позицию в таблице кодирования.
- 3) Каждый символ можно представить его позицией, выраженной 16-разрядным двоичным кодом.

1.5.4 Кодирование графической информации

Наравне с текстами человечество широко использует и графические формы представления информации: рисунки, картины, схемы и т. д. А с появлением электронных способов хранения, отображения и обработки информации возник вопрос кодирования графической информации.

Давайте возьмем очень сильное увеличительное стекло и рассмотрим черно-белую фотографию (рис. 1.5). Видим, что она состоит из мельчайших точек различных оттенков серого цвета. Этот характерный узор называют *растром*. Так же называют и математическую модель, которая используется для автоматизации работы с изображениями.

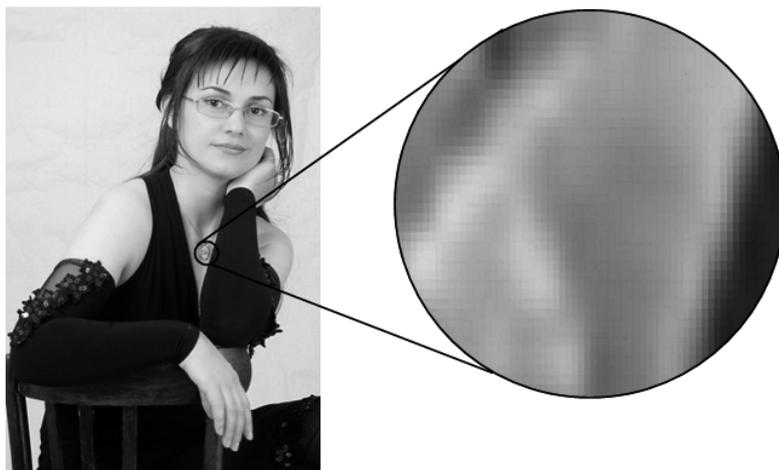


Рис. 1.5 – Растровый метод кодирования информации

В растровой компьютерной модели изображение считается:

- 1) прямоугольником с фиксированными размерами по ширине и высоте;
- 2) состоящим из регулярной последовательности цветных точек (пикселей¹).

Изображение, записанное в растровой модели, хранится как последовательность целых чисел, представляющих цвета отдельных точек в порядке развертывания прямоугольника слева направо и сверху вниз. Растровая модель — базовая для воспроизведения изображений. Какая бы модель ни использовалась для хранения рисунка, но на экране и принтере он всегда воспроизводится точками.

¹Слово «пиксел» происходит от английского словосочетания picture cell (ячейка изображения).

Преимущество растровой модели очевидно, если важнее всего точность воспроизведения цветовых оттенков, чем точность передачи формы. Каждая точка растра способна иметь свой цвет. Это позволяет создавать весьма сложные механизмы образования оттенков.

Вместе с тем разбиение изображения на точки приводит к искажению геометрических форм. Это очень хорошо заметно при увеличении изображения. Этот дефект называется *пикселизацией*.

Кроме растровой модели, используются *векторные* модели и модели *трехмерной графики*.

В векторной модели изображение представляется коллекцией независимых графических объектов, имеющих различимые свойства. Элементарным объектом векторного изображения является линия (кривая), обладающая различными свойствами: форма, местоположение, параметры контура, замкнутость контура и параметры внутренней заливки контура, если линия замкнута. Изображения, записанные в векторной модели, хранятся как таблицы свойств объектов. Благодаря этому они занимают очень мало места в памяти компьютера. Однако для воспроизведения и преобразования требуются весьма сложные и ресурсоемкие процедуры.

Векторную модель следует использовать в тех случаях, если важнее сохранить форму объекта. Так как компьютер хранит не само изображение, а коэффициенты алгебраического уравнения, качество изображения не зависит от масштаба увеличения.

Трехмерная графическая модель используется для отображения объемных фигур. Сначала объект представляется коллекцией граней. Каждая грань разбивается на треугольники. В свою очередь, каждый треугольник можно описать с помощью трех векторов, образующих его стороны. Каждый вектор описывается тремя числовыми значениями, выражающими его координаты относительно точки, принятой за начало отсчета.

Разбиение произвольной фигуры на треугольники называется *декомпозицией*. Адекватность трехмерной модели зависит от глубины декомпозиции.

Перед воспроизведением происходит перерасчет пространственной модели в плоское растровое экранное изображение. Этот процесс называется *визуализацией*.

Трехмерные модели позволяют передать не только сведения о цвете и форме объектов, но и об их взаимодействии в пространстве сцены. Полностью достоинства трехмерных моделей проявляются, когда изображение динамически меняется и зритель может управлять воспроизведением.

1.5.5 Кодирование звуковой информации

Звук — это упругие волны, распространяющиеся в воздушной среде и обладающие частотами от 16 Гц до 20 кГц. Чтобы произвести кодирование, в первую очередь звуковой сигнал преобразовывают в его электрический аналог. Для этого используется микрофон, на выходе которого получается непрерывный сигнал. Электрический сигнал преобразовывается в цифровой код с помощью аналого-цифрового преобразователя (АЦП). Во время оцифровки сигнала производится дискретизация сигнала по двум параметрам: времени и уровню.

Дискретизация сигнала во времени — это преобразование непрерывного аналогового сигнала в последовательность его значений в дискретные моменты време-

ни. Эти значения называются отсчетами или выборками. Выполняется следующим образом: весь период времени T разбивается на малые интервалы времени Δt , точками t_1, t_2, \dots, t_n . Интервал Δt выбирается таким образом, что за это время уровень сигнала меняется незначительно и с некоторым допущением может считаться постоянным. Величина $F_\delta = 1/\Delta t$ называется *частотой дискретизации*. Она измеряется в герцах (Гц) — количество измерений за одну секунду.

Дискретизация по уровню называется *квантованием*. Производится оно следующим образом: диапазон амплитуд отсчетов дискретного сигнала от U_{min} до U_{max} разбивается на 2^n интервалов. Величина получившегося интервала называется шагом квантования

$$\Delta = \frac{U_{max} - U_{min}}{2^n}. \quad (1.2)$$

Каждый уровень нумеруется целым числом, которое можно легко представить в двоичной системе счисления (n — разрядность таких чисел). Если амплитуда выборки сигнала в определенный момент времени попадает в промежуток $U_{i-1} \leq U \leq U_i$, то ему в соответствие ставится код i .

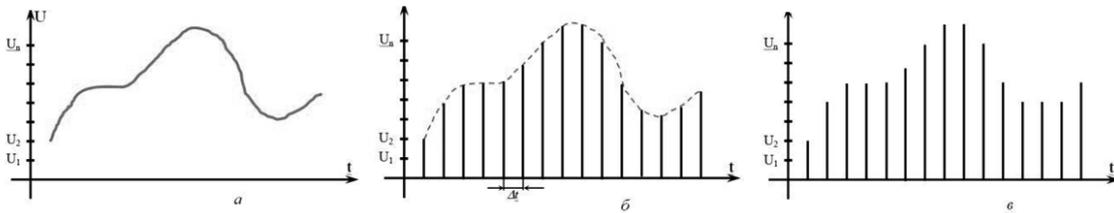


Рис. 1.6 – Аналоговый сигнал (а); дискретный сигнал (б); квантование отсчетов (в)

Оцифрованные отсчеты никогда не бывают точными. В результате неточного соответствия квантованного сигнала исходному возникает ошибка, которая приводит к появлению шума. Для уменьшения данной ошибки необходимо ответить на два вопроса:

- 1) какую выбрать частоту дискретизации;
- 2) каким должен быть шаг квантования.

Ответ на первый вопрос дает теорема Котельникова, согласно которой любой непрерывный сигнал, спектр которого ограничен верхней частотой f_s , может быть полностью восстановлен по его отсчетам, взятым с интервалом дискретизации меньше чем $1/(2f_s)$. Другими словами, частота дискретизации определяется выражением:

$$F_\delta \geq 2 \cdot f_s. \quad (1.3)$$

Вторая задача решается подбором числа уровней так, чтобы снизить уровень шума квантования. При этом не следует забывать о наличии внешних помех (например, помех в канале связи). Уменьшение шага квантования ниже уровня помех приводит к появлению ошибки. Поэтому нет смысла чрезмерно увеличивать число уровней квантования.



.....
Контрольные вопросы по главе 1
.....

- 1) Какие направления в информатике Вы знаете?
- 2) Перечислите свойства информации.
- 3) Какие системы счисления называют позиционными?
- 4) Какие, кроме перечисленных в методическом пособии, системы счисления Вы знаете?

Глава 2

ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

Вычислительная техника является важнейшим компонентом процесса вычислений и обработки данных. *Компьютер* (англ. computer — «вычислитель»), *ЭВМ* (электронная вычислительная машина) — машина для проведения вычислений, а также приема, переработки, хранения и выдачи информации по заранее определенному *алгоритму* (компьютерной *программе*).

Потребность в автоматизации обработки данных, в том числе вычислений, возникла еще несколько тысячелетий до н. э.

Первыми приспособлениями для вычислений были, вероятно, всем известные счетные палочки, которые и сегодня используются в начальных классах многих школ для обучения счету. Развиваясь, эти приспособления становились более сложными, например такими, как финикийские глиняные фигурки, также предназначенные для наглядного представления количества считаемых предметов, но помещаемые при этом в специальные контейнеры. Такими приспособлениями, похоже, пользовались торговцы и счетоводы того времени.

Постепенно из простейших приспособлений для счета рождались все более и более сложные устройства: абак (счеты), логарифмическая линейка, механический арифмометр, электронный компьютер.

2.1 История развития вычислительной техники

В период развития цифровых технологий разработаны компьютеры самых разных типов. Многие из них давно забыты, но другие оказали сильное влияние на развитие современных вычислительных систем. Далее приведен краткий обзор этапов развития вычислительных машин. Рассмотрим устройства для вычислений с автоматически выполняющимися заложенными в них программами.

1642 г. — «суммирующее устройство», первая счетная машина с хранимой программой построена французским ученым *Блезом Паскалем* (рисунок 2.1). Она была механической с ручным приводом и могла выполнять операции сложения и вычитания.

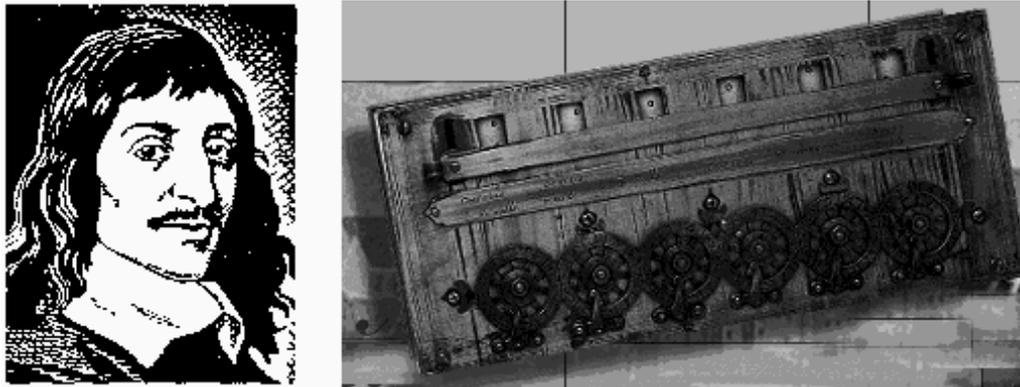


Рис. 2.1 – Блез Паскаль и его счетная машина

1672 г. — построена немецким математиком *Готфридом Лейбницем* механическая машина (калькулятор), которая могла делать также и операции умножения и деления.

Все эти устройства имели жесткую логику работы.

1834 г. — впервые разработана английским ученым *Чарльзом Беббиджем* машина, работающая по программе (рисунок 2.2).

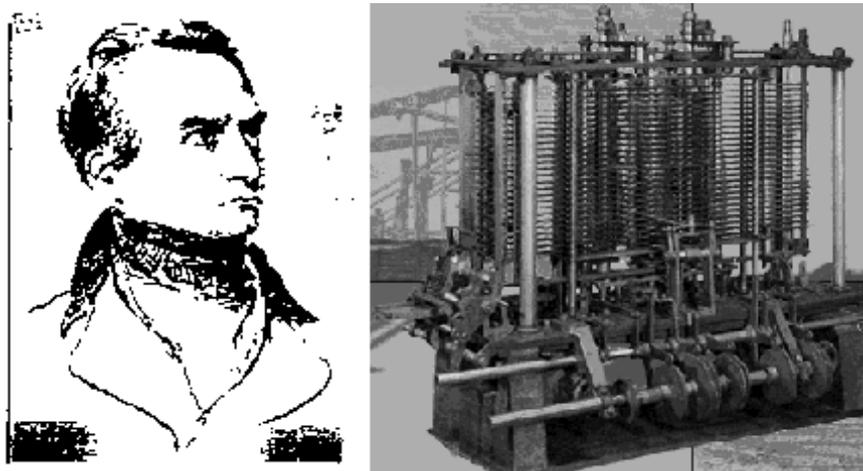


Рис. 2.2 – Чарльз Беббидж и его «аналитическая машина»

Стала возможной идея гибкой автоматизации механических устройств. Эта идея используется и в современных компьютерах. Суть этой идеи, реализованной им в «аналитической машине», состоит в разделении команд (программ) и данных.

Но, несмотря на революционность идей, эти устройства были механическими. Лишь век электричества и появление более совершенных технологий (использование электронных реле, электронных ламп, полупроводниковых элементов, микросхем) позволили воплотить давнюю мечту в реальность. Вот лишь некоторые, наиболее характерные примеры таких устройств.

В 1888 г. *Герман Холлерит* сконструировал первую счетную машину, использующую электрическое реле — табулятор.

В 1945 г. *Дж.Моучли* и *П.Эккерт* для упрощения процесса задания программ стали конструировать новую машину, которая могла бы хранить программу в своей памяти. Под их руководством создана первая действующая ЭВМ — *ENIAC* (рисунок 2.3).

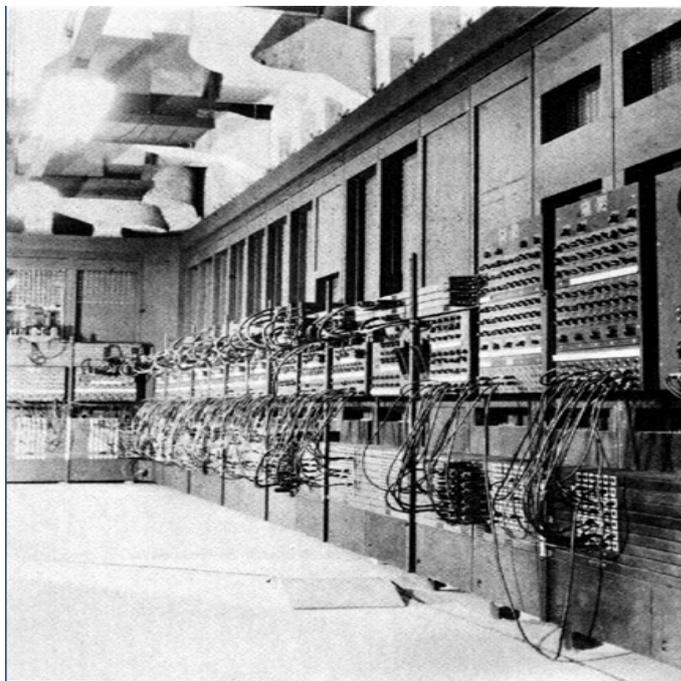


Рис. 2.3 – Первая в мире ЭВМ — ENIAC

Огромный вклад в теорию и практику создания электронной вычислительной техники на начальном этапе ее развития внес один из крупнейших американских математиков Джон фон Нейман. В историю науки навсегда вошли «принципы фон Неймана». Совокупность этих принципов породила классическую (фоннеймановскую) архитектуру ЭВМ. Один из важнейших принципов — принцип хранимой программы — требует, чтобы программа закладывалась в память машины так же, как в нее закладывается исходная информация. Первая ЭВМ с хранимой программой (EDSAC) была построена в Великобритании в 1949 г.

В истории вычислительной техники существует своеобразная периодизация ЭВМ по поколениям. В ее основу первоначально был положен физико-технологический принцип: машину относят к тому или иному поколению в зависимости от используемых в ней физических элементов или технологии их изготовления.

Границы поколений во времени размыты, так как в одно и то же время выпускались машины совершенно разного уровня. Когда приводят даты, относящиеся к поколениям, то, скорее всего, имеют в виду период промышленного производства; проектирование велось существенно раньше.

2.1.1 Первое поколение (1945–1955 гг.): электронные лампы

В основе базовой системы элементов этого поколения компьютеров лежали электронные лампы. Электронные лампы обеспечивали высокую скорость переключения логических элементов, что увеличивало скорость вычисления по срав-

нению с попытками создать вычислительную машину, базовый элемент которой был построен на основе электромеханического реле.

Вычислительные машины получались громоздкими, заполняющими целые комнаты, с десятками тысяч электронных ламп, но все равно они были в миллионы раз медленнее, чем даже самый дешевый современный персональный компьютер. В те времена каждую машину и разрабатывала, и строила, и программировала, и эксплуатировала, и поддерживала в рабочем состоянии одна команда. Все программирование выполнялось на машинном языке, управление основными функциями машины осуществлялось просто при помощи соединения коммутационных панелей проводами.

2.1.2 Второе поколение (1955–1965 гг.): транзисторы

В 1948 г. изобретены полупроводниковые приборы — транзисторы. Новая элементная база на основе транзисторов произвела революцию не только в производстве компьютеров, но и в радиоэлектронной промышленности в целом.

В отличие от электронных ламп транзисторы характеризовались малыми размерами, низким напряжением питания и малой потребляемой мощностью. Площадь, требуемая для размещения компьютера, снизилась до нескольких квадратных метров. Снижение стоимости компьютера увеличило число потенциальных пользователей компьютеров. Появились крупные фирмы по производству компьютеров широкого назначения: International Business Machines (IBM), Control Data Corporation (CDC), Digital Equipment Corporation (DEC) и др.

2.1.3 Третье поколение (1965–1980 гг.): интегральные схемы

В 1959 г. Роберт Нойс (будущий основатель фирмы Intel) изобрел метод, позволивший создать на одной пластинке и транзисторы, и все необходимые соединения между ними. Полученные электронные схемы стали называться интегральными схемами, или чипами. В 1970 г. фирма Intel начала продавать интегральные схемы памяти. Использование интегральных микросхем привело не только к резкому увеличению надежности ЭВМ, но и к снижению размеров, энергопотребления и стоимости (до 50 тыс. долларов). Уменьшение стоимости привело к разработке и практической реализации мощных вычислительных систем, использующих параллельную обработку: многопроцессорные и конвейерные вычислители.

2.1.4 Четвертое поколение (с 1980 гг.): сверхбольшие интегральные схемы

Теперь полупроводниковая схема содержала уже не набор нескольких логических элементов, из которых строились затем функциональные узлы компьютера, а целиком функциональные узлы. Это в первую очередь *процессор*, который, учитывая его размеры, получил название *микروпроцессор*, а также устройства управления внешними устройствами — *контроллеры* внешних устройств. Такие интегральные схемы получили название сначала *больших интегральных схем* (БИС), а затем и *сверхбольших интегральных схем* (СБИС).

Итогом такого развития микроэлектроники стало появление одноплатных ЭВМ, где на одной плате размещались несколько СБИС, содержащих все функциональные блоки компьютера. Ввиду того, что стоимость одноплатных ЭВМ стала еще более доступной, английские инженеры С. Джобс и С. Возняк собрали настольную вычислительную машину — *микрокомпьютер (Apple)* (рисунок 2.4).

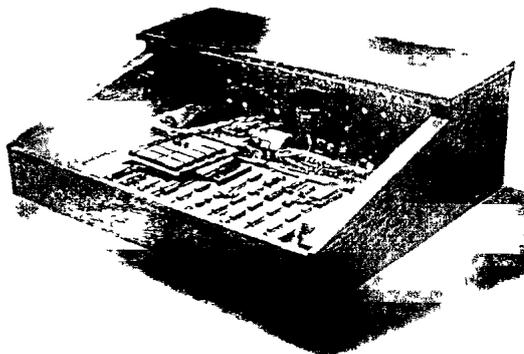


Рис. 2.4 – Первый персональный компьютер «Apple»

Совместно с фирмой *Intel*, разработавшей микропроцессорный комплект, и фирмой *Microsoft*, которая оснастила компьютер операционной системой MS DOS, крупная компания *IBM*, занимавшаяся выпуском мощных вычислительных систем, создала персональный компьютер *IBM PC*.

Дальнейшая классификация вычислительных систем по их принадлежности к различным поколениям весьма условна. В настоящее время элементная база микропроцессорных систем активно развивается, но в ее основе по-прежнему лежат СБИС. Некоторые специалисты выделяют пятое, шестое и последующие поколения как усовершенствование микроэлектронных технологий. Другие рассматривают последующие поколения как изменение структур обработки команд и данных внутри микропроцессора.

2.2 Классификация компьютеров

Массовость использования ПК, огромные рекламные усилия производителей и коммерсантов не должны заслонить тот факт, что, кроме ПК, есть и другие, многократно более мощные вычислительные системы. Всегда есть круг задач, для которых недостаточно существующих вычислительных мощностей и которые столь важны, что для их решения не жалко никаких средств. Это, например, может быть связано с обороноспособностью государства, решением сложнейших научно-технических задач, созданием и поддержкой гигантских банков данных. В настоящее время лишь немногие государства способны производить так называемые супер-ЭВМ — компьютеры, на фоне которых «персоналки» кажутся игрушками. Впрочем, сегодня ПК часто становится терминалом — конечным звеном в гигантских телекоммуникационных системах, в которых решением непосильных для ПК задач обработки информации занимаются более мощные ЭВМ.

Существуют различные классификации компьютерной техники, например следующая:

- по этапам развития (по поколениям);
- по архитектуре;
- по производительности;
- по условиям эксплуатации;
- по количеству процессоров;
- по потребительским свойствам и т. д.

Четких границ между классами компьютеров не существует. По мере совершенствования структур и технологии производства появляются новые классы компьютеров, границы существующих классов значительно изменяются.

Рассмотрим классификацию компьютеров по производительности и характеру использования:

- микрокомпьютеры, в том числе — персональные компьютеры;
- миникомпьютеры;
- мэйнфреймы (универсальные компьютеры);
- суперкомпьютеры;
- и другие.

Микрокомпьютеры — это компьютеры, в которых центральный процессор выполнен в виде микропроцессора.

Продвинутые модели микрокомпьютеров имеют несколько микропроцессоров. Производительность компьютера определяется не только характеристиками применяемого микропроцессора, но и емкостью оперативной памяти, типами периферийных устройств, качеством конструктивных решений и др.

Микрокомпьютеры представляют собой инструменты для решения разнообразных сложных задач. Их микропроцессоры с каждым годом увеличивают мощность, а периферийные устройства — эффективность.

Разновидность микрокомпьютера — *микромикроконтроллер*. Это основанное на микропроцессоре специализированное устройство, встраиваемое в систему управления или технологическую линию.

Персональные компьютеры (ПК) — это микрокомпьютеры универсального назначения, рассчитанные на одного пользователя и управляемые одним человеком.

В класс персональных компьютеров входят различные машины — от домашних и игровых с небольшой оперативной памятью, с памятью программы на кассетной ленте и обычным телевизором в качестве дисплея (80-е годы), до сверхсложных машин с мощным процессором, винчестерским накопителем емкостью в десятки Гигабайт, с цветными графическими устройствами высокого разрешения, средствами мультимедиа и другими дополнительными устройствами.

Миникомпьютерами и *суперминикомпьютерами* называются машины, конструктивно выполненные в одной стойке, т. е. занимающие объем порядка половины кубометра. Сейчас компьютеры этого класса вымирают, уступая место микрокомпьютерам.

Мэйнфреймы предназначены для решения широкого класса научно-технических задач и являются сложными и дорогими машинами. Их целесообразно применять в больших системах при наличии не менее 200–300 рабочих мест.

Десятки мэйнфреймов могут работать совместно под управлением одной операционной системы над выполнением единой задачи.

Суперкомпьютеры — это очень мощные компьютеры с производительностью свыше 100 мегафлопов (1 мегафлоп — миллион операций с плавающей точкой в секунду). Они называются *сверхбыстродействующими*. Эти машины представляют собой *многопроцессорные* и (или) *многомашинные* комплексы, работающие на общую память и общее поле внешних устройств. Различают суперкомпьютеры среднего класса, класса выше среднего и переднего края (high end).

Наиболее распространенные суперкомпьютеры — *массово-параллельные* компьютерные системы. Они имеют десятки тысяч процессоров, взаимодействующих через сложную, иерархически организованную систему памяти.

Суперкомпьютеры используются для решения сложных и больших научных задач (метеорология, гидродинамика и т. п.), в управлении, разведке, в качестве централизованных хранилищ информации и т. д.

На компьютерном рынке сложилось следующее разделение конфигураций персональных компьютеров.

Рабочая станция (Work Station) представляет собой мощный компьютер, основанный обычно на двухпроцессорной платформе, оснащенный максимальным объемом быстрой оперативной памяти, массивом жестких дисков и часто включенный в локальную сеть предприятия. В зависимости от решаемых задач рабочие станции бывают графическими, для научных расчетов или иного назначения.

Графическую рабочую станцию комплектуют 3D-видеокартой профессионального класса, устройствами оцифровки и захвата сигналов телевизионного формата, высокоточными сканерами и другим необходимым оборудованием.

Термин «рабочая станция» используется в нескольких, порой несовпадающих, смыслах. Так, рабочей станцией может быть мощная микро-ЭВМ, ориентированная на специализированные работы высокого профессионального уровня, которую нельзя отнести к персональным компьютерам хотя бы в силу очень высокой стоимости.

Настольный компьютер (Desktop) предусматривает самый обширный спектр возможных конфигураций как платформы, так и дополнительных устройств. Принято классифицировать настольные компьютеры по предназначению или по производительности. По назначению компьютеры подразделяют на *офисные*, *домашние*, *игровые*, *дизайнерские*. По производительности различают компьютеры начального уровня (Easy PC), среднего уровня (Mainstream), высшего класса (High End).

Ноутбук (Notebook) является переносным персональным компьютером. Помимо компактных габаритов, ноутбук отличается от настольного компьютера возможностью работы от аккумуляторов. Автономное функционирование обусловило высокие требования к режиму энергопотребления компонентов. Обычно в ноутбуках используют специальные модификации процессоров, графических чипсетов, жестких дисков с низким энергопотреблением и автоматическим регулированием производительности в зависимости от решаемой задачи.

Обычно ноутбуки классифицируют по размеру, диагонали дисплея и числу отдельных приводов (жесткий диск, дисковод CD-ROM, дисковод гибких дисков и др.).

Настольный ноутбук (DeskNote). Этот класс компьютеров возник и развился в 2002 году. Его отличие от ноутбуков заключается в отсутствии аккумуляторов (и, как следствие, невозможности автономной работы), использовании процессоров для обычных настольных ПК, а иногда и адаптеров 3D-графики высокого класса.

Планишетный ПК (Tablet PC) характеризуется наличием отдельного сенсорного дисплея с возможностью рукописного ввода и специального электронного пера. Некоторые модели комплектуются клавиатурой, трекболом, приводом CD-ROM, жестким диском.

Карманный ПК (Personal Digital Assistant, PDA) примыкает к товарной нише персональных компьютеров. Невысокая производительность, ограниченный набор программ и неудобный интерфейс пользователя сужают сферу применения КПК. Однако многие КПК позволяют подключаться к настольному компьютеру для переноса данных: телефонного справочника, записной книжки и прочих, позволяют читать литературные произведения в электронном виде, просматривать видео и т. д.

2.3 Устройство компьютера

Компьютер — это электронное устройство, предназначенное для работы с информацией, а именно осуществляет *введение, обработку, хранение, вывод и передачу информации*. Кроме того, ПК представляет собой единое двух составляющих — аппаратной и программной. Согласно определению компьютера его компоненты можно разделить на устройства, которые выполняют определенные функции, связанные с информацией (рисунок 2.5).



Рис. 2.5 – Компоненты компьютера

Персональный компьютер с технической точки зрения можно определить как единую систему, представляющую собой набор сменных компонентов, соединенных между собой стандартными интерфейсами. Компонентом здесь выступает отдельный узел (устройство), выполняющий определенную функцию в составе системы.

Интерфейсом называют стандарт присоединения компонентов к системе. В качестве такового служат разъемы, наборы микросхем, генерирующих стандартные сигналы, стандартный программный код. В компьютерной индустрии существует набор однотипных компонентов с разными функциональными возможностями, включаемых в систему по единому интерфейсу. Полное описание набора и характеристик устройств, составляющих данный компьютер, называется *конфигурацией ПК*.

Существует «минимальная» конфигурация ПК, т.е. минимальный набор устройств, без которых работа с ПК становится бессмысленной. Это: системный блок, монитор, клавиатура, мышь. Обычно под набором комплектующих, объединенных понятием «типовой персональный компьютер», понимают следующий их состав: корпус с блоком питания, системная (материнская) плата, процессор, оперативная память, видеоконтроллер, монитор, жесткий диск, клавиатура, манипулятор «мышь», звуковая карта, CD—DWD.

Как устроен компьютер?

Разнообразие современных компьютеров очень велико. Но их структуры основаны на общих логических принципах, позволяющих выделить в любом компьютере следующие главные устройства:

- память (запоминающее устройство (ЗУ));
- процессор, включающий в себя устройство управления (УУ) и арифметико-логическое устройство (АЛУ);
- устройство ввода;
- устройство вывода.

Эти устройства соединены *каналами связи*, по которым передается информация.

Основные устройства компьютера и связи между ними представлены на схеме (рисунок 2.6). Жирными стрелками показаны пути и направления движения информации, а простыми стрелками — пути и направления передачи управляющих сигналов.

Функции памяти:

- прием информации из других устройств;
- запоминание информации;
- выдача информации по запросу в другие устройства машины.

Функции процессора:

- обработка данных по заданной программе путем выполнения арифметических и логических операций;
- программное управление работой устройств компьютера.

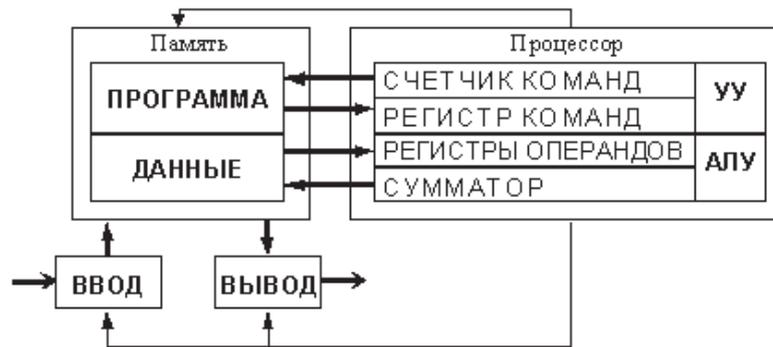


Рис. 2.6 – Общая схема компьютера

Та часть процессора, которая выполняет команды, называется *арифметико-логическим устройством (АЛУ)*, а другая его часть, выполняющая функции управления устройствами, называется *устройством управления (УУ)*. Обычно эти два устройства выделяются чисто условно, конструктивно они не разделены.

В составе процессора имеется ряд специализированных дополнительных ячеек памяти, называемых *регистрами*.

Регистр выполняет функцию кратковременного хранения числа или команды. Над содержимым некоторых регистров специальные электронные схемы могут выполнять некоторые манипуляции. Например, «вырезать» отдельные части команды для последующего их использования или выполнять определенные арифметические операции над числами.

Основным элементом регистра является электронная схема, называемая триггером, которая способна хранить одну двоичную цифру (разряд двоичного кода).

Регистр представляет собой совокупность триггеров, связанных друг с другом определенным образом общей системой управления. Существует несколько типов регистров, отличающихся видом выполняемых операций. Некоторые важные регистры имеют свои названия, например:

- сумматор — регистр АЛУ, участвующий в выполнении каждой операции;
- счетчик команд — регистр УУ, содержимое которого соответствует адресу очередной выполняемой команды; служит для автоматической выборки программы из последовательных ячеек памяти;
- регистр команд — регистр УУ для хранения кода команды на период времени, необходимый для ее выполнения. Часть его разрядов используется для хранения *кода операции*, остальные — для хранения *кодов адресов операндов*.

Как выполняется команда?

Выполнение команды можно проследить по схеме (рисунок 2.6). Как правило, этот процесс разбивается на следующие этапы:

- из ячейки памяти, адрес которой хранится в счетчике команд, выбирается очередная команда; содержимое счетчика команд при этом увеличивается на длину команды;

- выбранная команда передается в устройство управления на регистр команд;
- устройство управления расшифровывает адресное поле команды;
- по сигналам УУ операнды считываются из памяти и записываются в АЛУ на специальные регистры операндов;
- УУ расшифровывает код операции и выдает в АЛУ сигнал выполнить соответствующую операцию над данными;
- результат операции либо остается в процессоре, либо отправляется в память, если в команде был указан адрес результата;
- все предыдущие этапы повторяются до достижения команды «стоп».

При рассмотрении компьютерных устройств принято различать их архитектуру и структуру.

Архитектурой компьютера называется его описание на некотором общем уровне, включающее описание пользовательских возможностей программирования, системы команд, системы адресации, организации памяти и т. д. Архитектура определяет принципы действия, информационные связи и взаимное соединение основных логических узлов компьютера: процессора, оперативного ЗУ, внешних ЗУ и периферийных устройств. Общность архитектуры разных компьютеров обеспечивает их совместимость с точки зрения пользователя.

Структура компьютера — это совокупность его функциональных элементов и связей между ними. Элементами могут быть самые различные устройства — от основных логических узлов компьютера до простейших схем. Структура компьютера графически представляется в виде структурных схем, с помощью которых можно дать описание компьютера на любом уровне детализации.

2.3.1 Архитектура ЭВМ

Архитектура ЭВМ включает в себя как структуру, отражающую состав ПК, так и программно-математическое обеспечение. Структура ЭВМ — совокупность элементов и связей между ними. Основным принципом построения всех современных ЭВМ является программное управление.

Основы учения об архитектуре вычислительных машин были заложены Джоном фон Нейманом. Совокупность этих принципов породила классическую (фоннеймановскую) архитектуру ЭВМ. Фон Нейман не только выдвинул основополагающие принципы логического устройства ЭВМ, но и предложил ее структуру, представленную на рисунке 2.7.

Положения фон Неймана:

- Компьютер состоит из нескольких основных устройств (арифметико-логическое устройство, управляющее устройство, память, внешняя память, устройства ввода и вывода).
- Арифметико-логическое устройство — выполняет логические и арифметические действия, необходимые для переработки информации, хранящейся в памяти.
- Управляющее устройство — обеспечивает управление и контроль всех устройств компьютера (управляющие сигналы указаны пунктирными стрелками).

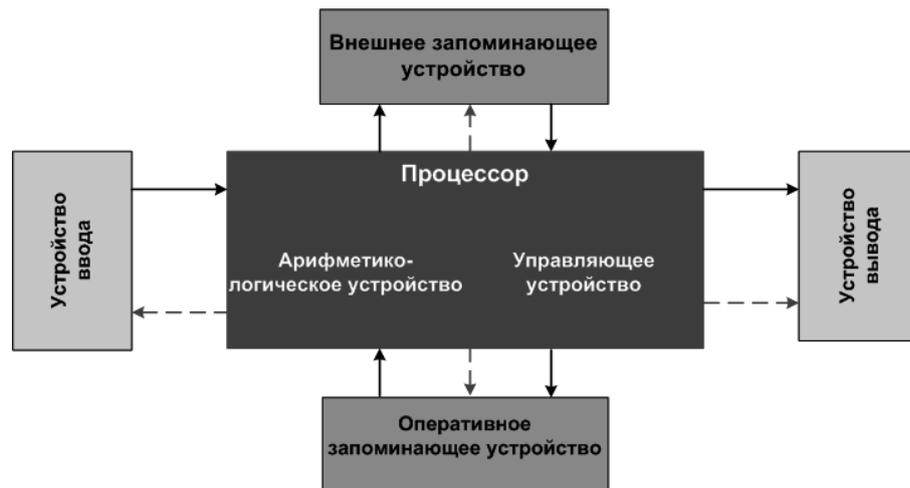


Рис. 2.7 – Структура ЭВМ

- Данные, которые хранятся в запоминающем устройстве, представлены в двоичной форме.
- Программа, которая задает работу компьютера, и данные хранятся в одном и том же запоминающем устройстве.
- Для ввода и вывода информации используются устройства ввода и вывода.

Один из важнейших принципов — принцип хранимой программы — требует, чтобы программа закладывалась в память машины так же, как в нее закладывается исходная информация.

Арифметико-логическое устройство и устройство управления в современных компьютерах образуют процессор ЭВМ. Процессор, который состоит из одной или нескольких больших интегральных схем, называется микропроцессором или микропроцессорным комплектом.

Процессор — функциональная часть ЭВМ, выполняющая основные операции по обработке данных и управлению работой других блоков. Процессор является преобразователем информации, поступающей из памяти и внешних устройств.

Запоминающие устройства обеспечивают хранение исходных и промежуточных данных, результатов вычислений, а также программ. Они включают: оперативные (ОЗУ), сверхоперативные (СОЗУ), постоянные (ПЗУ) и внешние (ВЗУ) запоминающие устройства.

Оперативные ЗУ хранят информацию, с которой компьютер работает непосредственно в данное время (резидентная часть операционной системы, прикладная программа, обрабатываемые данные). В СОЗУ хранятся наиболее часто используемые процессором данные. Только та информация, которая хранится в СОЗУ и ОЗУ, непосредственно доступна процессору.

Внешние запоминающие устройства (накопители на магнитных дисках, например жесткий диск или винчестер) с емкостью намного больше, чем ОЗУ, но с существенно более медленным доступом используются для длительного хранения больших объемов информации. Например, операционная система (ОС) хранится на жестком диске, но при запуске компьютера резидентная часть ОС загружается в ОЗУ и находится там до завершения сеанса работы ПК.

ПЗУ (постоянные запоминающие устройства) и ППЗУ (перепрограммируемые постоянные запоминающие устройства) предназначены для постоянного хранения информации, которая записывается туда при ее изготовлении, например ППЗУ для BIOS. В качестве устройства ввода информации служит, например, клавиатура. В качестве устройства вывода — дисплей, принтер и т. д.

В построенной по схеме фон Неймана ЭВМ происходит последовательное считывание команд из памяти и их выполнение. Номер (адрес) очередной ячейки памяти, из которой будет извлечена следующая команда программы, указывается специальным устройством — счетчиком команд в устройстве управления.

Наиболее распространены следующие архитектурные решения.

Классическая архитектура (архитектура фон Неймана) — одно арифметико-логическое устройство (АЛУ), через которое проходит поток данных, и одно устройство управления (УУ), через которое проходит поток команд, — программа (рисунк 2.7). Это однопроцессорный компьютер. К этому типу архитектуры относятся и архитектура персонального компьютера с общей шиной. Все функциональные блоки здесь связаны между собой общей шиной, называемой также *системной магистралью*.

Физически магистраль представляет собой многопроводную линию с гнездами для подключения электронных схем. Совокупность проводов магистрали разделяется на отдельные группы: шину адреса, шину данных и шину управления.

Периферийные устройства (принтер и др.) подключаются к аппаратуре компьютера через специальные контроллеры — устройства управления периферийными устройствами.

Контроллер — устройство, которое связывает периферийное оборудование или каналы связи с центральным процессором, освобождая процессор от непосредственного управления функционированием данного оборудования.

Многопроцессорная архитектура. Наличие в компьютере нескольких процессоров означает, что параллельно может быть организовано множество потоков данных и потоков команд. Таким образом, параллельно могут выполняться несколько фрагментов одной задачи. Структура такой машины, имеющей общую оперативную память и несколько процессоров, представлена на рисунке 2.8.

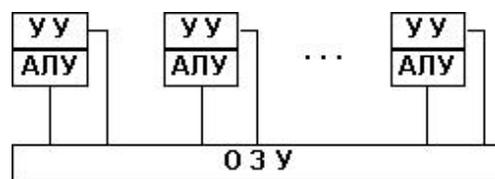


Рис. 2.8 – Архитектура многопроцессорного компьютера

Многомашинная вычислительная система. Здесь несколько процессоров, входящих в вычислительную систему, не имеют общей оперативной памяти, а имеют каждый свою (локальную). Каждый компьютер в многомашинной системе имеет классическую архитектуру, и такая система применяется достаточно широко. Однако эффект от применения такой вычислительной системы может быть получен только при решении задач, имеющих очень специальную структуру: она должна разбиваться на столько слабо связанных подзадач, сколько компьютеров в системе.

Архитектура с параллельными процессорами. Здесь несколько АЛУ работают под управлением одного УУ. Это означает, что множество данных может обрабатываться по одной программе — то есть по одному потоку команд. Высокое быстродействие такой архитектуры можно получить только на задачах, в которых одинаковые вычислительные операции выполняются одновременно на различных однотипных наборах данных. Структура таких компьютеров представлена на рисунке 2.9.

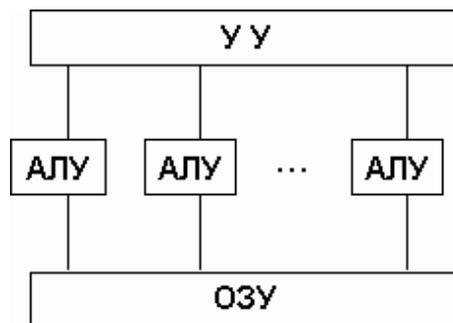


Рис. 2.9 – Архитектура с параллельным процессором

В современных машинах часто присутствуют элементы различных типов архитектурных решений. Существуют и такие архитектурные решения, которые радикально отличаются от рассмотренных выше.

2.3.2 Центральный процессор

Центральный процессор (CPU, от англ. Central Processing Unit) — это основной рабочий компонент компьютера, который выполняет арифметические и логические операции, заданные программой, управляет вычислительным процессом и координирует работу всех устройств компьютера.

Во время работы из-за использования больших тактовых частот процессор сильно нагревается, поэтому необходимо его охлаждать. Для эффективного охлаждения процессора в комплекте с ним поставляются вентилятор и радиатор. Радиатор представляет собой металлическую пластину (из меди или алюминия) с лопастями, благодаря которым повышается его площадь и соответственно теплоотдача. На радиатор крепится вентилятор, увеличивая охлаждение процессора.

Перечислим основные параметры CPU:

- тактовая частота работы процессора;
- разрядность обрабатываемых данных;
- степпинг (тип подключения к материнской плате и количество контактов);
- объем кэш — специальной быстрой памяти, предназначенной для повышения производительности процессора.

2.3.3 Память компьютера и ее виды

Память компьютера построена из двоичных запоминающих элементов — *битов*, объединенных в группы по 8 битов, которые называются *байтами*. (Единицы

измерения памяти совпадают с единицами измерения информации.) Все байты пронумерованы. Номер байта называется его *адресом*.

Байты могут объединяться в ячейки, которые называются также *словами*. Для каждого компьютера характерна определенная длина слова — два, четыре или восемь байтов. Это не исключает использования ячеек памяти другой длины (например, полуслово, двойное слово). Как правило, в одном машинном слове может быть представлено либо одно целое число, либо одна команда. Однако допускаются переменные форматы представления информации. Разбиение памяти на слова для четырехбайтовых компьютеров представлено в таблице 2.1:

Таблица 2.1 – Разбиение памяти на слова

Байт 0	Байт 1	Байт 2	Байт 3	Байт 4	Байт 5	Байт 6	Байт 7
ПОЛУСЛОВО		ПОЛУСЛОВО		ПОЛУСЛОВО		ПОЛУСЛОВО	
СЛОВО				СЛОВО			
ДВОЙНОЕ СЛОВО							

Широко используются и более крупные производные единицы объема памяти: *Килобайт*, *Мегабайт*, *Гигабайт* и др.

Различают два основных вида памяти — *внутреннюю* и *внешнюю*.

В состав внутренней памяти входят оперативная память, кэш-память и специальная память.

Оперативная память (ОЗУ, англ. RAM, Random Access Memory — память с произвольным доступом) — это быстрое запоминающее устройство, непосредственно связанное с процессором и предназначенное для записи, считывания и хранения выполняемых программ и данных, обрабатываемых этими программами.

Оперативная память используется только для временного хранения данных и программ, так как при выключении машины выключается все, что находилось в ОЗУ, пропадает. Доступ к элементам оперативной памяти прямой — это означает, что каждый байт памяти имеет свой индивидуальный адрес.

Модули памяти характеризуются такими параметрами, как объем, число микросхем, паспортная частота, время доступа к данным и число контактов.

Кэш (англ. cache), или сверхоперативная память, — ЗУ небольшого объема, которое используется при обмене данными между микропроцессором и оперативной памятью для компенсации разницы в скорости обработки информации процессором и несколько менее быстродействующей оперативной памятью.

Кэш-памятью управляет специальное устройство — контроллер, который, анализируя выполняемую программу, пытается предвидеть, какие данные и команды вероятнее всего понадобятся в ближайшее время процессору, и подкачивает их в кэш-память. При этом возможны как «попадания», так и «промахи». В случае попадания, то есть, если в кэш подкачаны нужные данные, извлечение их из памяти происходит без задержки. Если же требуемая информация в кэше отсутствует, то процессор считывает ее непосредственно из оперативной памяти. Соотношение числа попаданий и промахов определяет эффективность кэширования.

К устройствам **специальной памяти** относятся постоянная память (ROM), перепрограммируемая постоянная память (Flash Memory), память CMOS RAM, питаемая от батарейки, видеопамять и некоторые другие виды памяти.

Постоянная память (ПЗУ, англ. ROM, Read Only Memory — память только для чтения) — энергонезависимая память, используется для хранения данных, которые никогда не потребуют изменения. Содержание памяти специальным образом «зашивается» в устройстве при его изготовлении для постоянного хранения. Из ПЗУ можно только читать.

Перепрограммируемая постоянная память (Flash Memory) — энергонезависимая память, допускающая многократную перезапись своего содержимого с дискеты.

Прежде всего, в постоянную память записывают программу управления работой самого процессора. В ПЗУ находятся программы управления дисплеем, клавиатурой, принтером, внешней памятью, программы запуска и остановки компьютера, тестирования устройств.

Важнейшая микросхема постоянной или Flash-памяти — модуль BIOS. Роль BIOS двоякая: с одной стороны, это неотъемлемый элемент аппаратуры, а с другой стороны — важный модуль любой операционной системы.

BIOS (Basic Input/Output System — базовая система ввода-вывода) — совокупность программ, предназначенных для автоматического тестирования устройств после включения питания компьютера и загрузки операционной системы в оперативную память.

Разновидность постоянного ЗУ — CMOS RAM. *CMOS RAM* — это память с невысоким быстродействием и минимальным энергопотреблением от батарейки. Используется для хранения информации о конфигурации и составе оборудования компьютера, а также о режимах его работы.

Для хранения графической информации используется видеопамять.

Видеопамять (VRAM) — разновидность оперативного ЗУ, в котором хранятся закодированные изображения. Это ЗУ организовано так, что его содержимое доступно сразу двум устройствам — процессору и дисплею. Поэтому изображение на экране меняется одновременно с обновлением видеоданных в памяти.

Внешняя память (ВЗУ) предназначена для длительного хранения программ и данных, и целостность ее содержимого не зависит от того, включен или выключен компьютер. В отличие от оперативной памяти, внешняя память не имеет прямой связи с процессором. К внешней памяти компьютера можно отнести накопители:

- на жестких магнитных дисках;
- гибких магнитных дисках;
- компакт-дисках;
- магнито-оптических компакт-дисках;
- магнитной ленте (стримеры) и др.

Жесткий диск (винчестер) предназначен для постоянного хранения информации, используемой при работе с компьютером: программ операционной системы, часто используемых пакетов прикладных программ, текстовых документов, файлов базы данных и др. Главными параметрами жесткого диска являются его емкость, скорость считывания данных, время доступа и размер внутреннего буфера (кэш-памяти).

Накопитель на гибких магнитных дисках служит для чтения/записи информации на носители, называемые дискетами или флоппи-дисками. Сам накопитель

часто называют просто дисководом и обозначают аббревиатурой FDD (*Floppy Disk Drive*).

Компакт-диски (CD-ROM — Compact Disk Read-Only Memory) широко используются для хранения различного программного обеспечения в виде дистрибутивов, так как позволяют хранить большие объемы информации. Привод CD-ROM достаточно универсален: он может воспроизводить как обычные звуковые и видеодиски, так и диски, предназначенные специально для компьютера, содержащие, например, прикладные программы.

2.3.4 Аудио- и видеоадаптер

Аудиоадаптер (Sound Blaster или звуковая плата) это специальная электронная плата, которая позволяет записывать звук, воспроизводить его и создавать программными средствами с помощью микрофона, наушников, динамиков, встроенного синтезатора и другого оборудования.

Аудиоадаптер содержит в себе два преобразователя информации:

- *аналого-цифровой*, который преобразует непрерывные (то есть аналоговые) звуковые сигналы (речь, музыку, шум) в цифровой двоичный код и записывает его на магнитный носитель;
- *цифроаналоговый*, выполняющий обратное преобразование сохраненного в цифровом виде звука в аналоговый сигнал, который затем воспроизводится с помощью акустической системы, синтезатора звука или наушников.

Профессиональные звуковые платы позволяют выполнять сложную обработку звука, обеспечивают стереозвучание, имеют собственное ПЗУ с хранящимися в нем сотнями тембров звучаний различных музыкальных инструментов. Звуковые файлы обычно имеют очень большие размеры.

Видеоадаптер — это электронная плата, которая обрабатывает видеоданные (текст и графику) и управляет работой дисплея. Содержит видеопамять, регистры ввода вывода и модуль BIOS. Посылает в дисплей сигналы управления яркостью лучей и сигналы развертки изображения.

С увеличением числа приложений, использующих сложную графику и видео, наряду с традиционными видеоадаптерами широко используются разнообразные *устройства компьютерной обработки видеосигналов*:

- **Графические акселераторы (ускорители)** — специализированные графические сопроцессоры, увеличивающие эффективность видеосистемы. Их применение освобождает центральный процессор от большого объема операций с видеоданными, так как акселераторы самостоятельно вычисляют, какие пиксели отображать на экране и каковы их цвета.
- **Фрейм-грабберы**, которые позволяют отображать на экране компьютера видеосигнал от видеомagneтофона, камеры, лазерного проигрывателя и т. п., с тем чтобы захватить нужный кадр в память и впоследствии сохранить его в виде файла.
- **TV-тюнеры** — позволяют выбрать любую нужную телевизионную программу и отображать ее на экране в масштабируемом окне. Таким образом можно следить за ходом передачи, не прекращая работу.

2.3.5 Видеосистема компьютера

Видеосистема компьютера состоит из трех компонент:

- монитор (называемый также дисплеем);
- видеоадаптер;
- программное обеспечение (драйверы видеосистемы).

Видеоадаптер посылает в монитор сигналы управления яркостью лучей и синхросигналы строчной и кадровой разверток. Монитор преобразует эти сигналы в зрительные образы. А программные средства обрабатывают видеоизображения — выполняют кодирование и декодирование сигналов, координатные преобразования, сжатие изображений и др.

Монитор — устройство визуального отображения информации (в виде текста, таблиц, рисунков, чертежей и др.).

Монитор на базе электронно-лучевой трубки

Основной элемент дисплея — электронно-лучевая трубка. Ее передняя, обращенная к зрителю часть с внутренней стороны покрыта люминофором — специальным веществом, способным излучать свет при попадании на него быстрых электронов. Люминофор наносится в виде наборов точек трех основных цветов — красного, зеленого и синего. Эти цвета называют основными, потому что их сочетаниями (в различных пропорциях) можно представить любой цвет спектра.

Наборы точек люминофора располагаются по треугольным триадам. Триада образует **пиксел** — точку, из которых формируется изображение (англ. pixel — picture element, элемент картинки). Расстояние между центрами пикселей называется **точечным шагом монитора**. Это расстояние существенно влияет на четкость изображения. Чем меньше шаг, тем выше четкость.

Количество отображенных строк в секунду называется **строчной частотой развертки**. А частота, с которой меняются кадры изображения, называется **кадровой частотой развертки**.

Жидкокристаллические мониторы

Жидкие кристаллы — это особое состояние некоторых органических веществ, в котором они обладают текучестью и свойством образовывать пространственные структуры, подобные кристаллическим. Жидкие кристаллы могут изменять свою структуру и светооптические свойства под действием электрического напряжения. Меняя с помощью электрического поля ориентацию групп кристаллов и используя введенные в жидкокристаллический раствор вещества, способные излучать свет под воздействием электрического поля, можно создать высококачественные изображения, передающие более 15 миллионов цветовых оттенков.

Большинство ЖК-мониторов использует тонкую пленку из жидких кристаллов, помещенную между двумя стеклянными пластинами. Заряды передаются через так называемую пассивную матрицу — сетку невидимых нитей, горизонтальных и вертикальных, создавая в месте пересечения нитей точку изображения (несколько размытого из-за того, что заряды проникают в соседние области жидкости).

Активные матрицы вместо нитей используют прозрачный экран из транзисторов и обеспечивают яркое, практически не имеющее искажений изображение. Экран при этом разделен на независимые ячейки, каждая из которых состоит из четырех частей (для трех основных цветов и одна резервная). Количество таких ячеек по широте и высоте экрана называют разрешением экрана.

Сенсорный экран

Общение с компьютером осуществляется путем прикосновения пальцем к определенному месту чувствительного экрана. Этим выбирается необходимый режим из меню, показанного на экране монитора. (Меню — это выведенный на экран монитора список различных вариантов работы компьютера, по которому можно сделать конкретный выбор.) Сенсорными экранами оборудуют рабочие места операторов и диспетчеров, их используют в информационно-справочных системах и т. д.

2.4 Периферийные устройства

2.4.1 Принтеры

Принтер — печатающее устройство. Осуществляет вывод из компьютера закодированной информации в виде печатных копий текста или графики. Существуют тысячи наименований принтеров. Но основных видов принтеров три: матричные, лазерные и струйные.

Матричные принтеры используют комбинации маленьких штырьков, которые бьют по красящей ленте, благодаря чему на бумаге остается отпечаток символа. Каждый символ, печатаемый на принтере, формируется из набора 9, 18 или 24 игл, сформированных в виде вертикальной колонки. Недостатками этих недорогих принтеров являются их шумная работа и невысокое качество печати.

Лазерные принтеры работают примерно так же, как ксероксы. Компьютер формирует в своей памяти «образ» страницы текста и передает его принтеру. Информация о странице проецируется с помощью лазерного луча на вращающийся барабан со светочувствительным покрытием, меняющим электрические свойства в зависимости от освещенности. После засветки на барабан, находящийся под электрическим напряжением, наносится красящий порошок — **тонер**, частицы которого налипают на засвеченные участки поверхности барабана. Принтер с помощью специального горячего валика протягивает бумагу под барабаном; тонер переносится на бумагу и «вплавляется» в нее, оставляя стойкое высококачественное изображение.

Струйные принтеры генерируют символы в виде последовательности чернильных точек. Печатающая головка принтера имеет крошечные сопла, через которые на страницу выбрызгиваются быстросохнущие чернила. Эти принтеры требовательны к качеству бумаги. Цветные струйные принтеры создают цвета, комбинируя чернила четырех основных цветов — ярко-голубого, пурпурного, желтого и черного.

Принтер связан с компьютером посредством кабеля принтера, один конец которого вставляется своим разъемом в гнездо принтера, а другой — в порт принтера

компьютера. *Порт* — это разъем, через который можно соединить процессор компьютера с внешним устройством.

Каждый принтер обязательно имеет свой драйвер — программу, которая способна переводить (транслировать) стандартные команды печати компьютера в специальные команды, требующиеся для каждого принтера.

Плоттер (графопостроитель) — устройство, которое чертит графики, рисунки или диаграммы под управлением компьютера. Плоттеры используются для получения сложных конструкторских чертежей, архитектурных планов, географических и метеорологических карт, деловых схем. Плоттеры рисуют изображения с помощью пера.

Роликовые плоттеры прокручивают бумагу под пером, а планшетные плоттеры перемещают перо через всю поверхность горизонтально лежащей бумаги. Плоттеру, так же как и принтеру, обязательно нужна специальная программа — *драйвер*, позволяющая прикладным программам передавать ему инструкции: поднять и опустить перо, провести линию заданной толщины и т. п.

2.4.2 Сканеры

Сканер — устройство для ввода в компьютер графических изображений. Создает оцифрованное изображение документа и помещает его в память компьютера. Если принтеры выводят информацию из компьютера, то сканеры, наоборот, переносят информацию с бумажных документов в память компьютера. Существуют ручные сканеры, которые прокатывают по поверхности документа рукой, и планшетные сканеры, по внешнему виду напоминающие копировальные машины.

Если при помощи сканера вводится текст, компьютер воспринимает его как картинку, а не как последовательность символов. Для преобразования такого графического текста в обычный символьный формат используют программы оптического распознавания образов.

2.4.3 Модемы и факс-модемы

Модем — устройство, позволяющее компьютеру выходить на связь с другим компьютером посредством телефонных линий. Факс-модем — модем, позволяющий также принимать и посылать факсимильные сообщения.

По своему внешнему виду и месту установки модемы подразделяются на внутренние и внешние. Внутренние модемы представляют собой электронную плату, устанавливаемую непосредственно в компьютер, а внешние — автономное устройство, подсоединяемое к одному из портов. Основной параметр в работе модема — скорость передачи данных. Она измеряется в bps (бит в секунду).



Контрольные вопросы по главе 2

- 1) Какие выделяют основные этапы развития вычислительных машин?
- 2) Какие существуют классификации компьютерной техники?
- 3) Что такое конфигурация ПК?
- 4) Расскажите о составляющих общей схемы компьютера.
- 5) Приведите основные архитектурные решения компьютера.
- 6) Из каких компонентов состоит видеосистема компьютера?
- 7) Назовите известные вам периферийные устройства.

Глава 3

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ КОМПЬЮТЕРА

Ни компьютер в целом, ни его составные части не способны сами по себе обрабатывать данные. Управляют работой компьютера *программы*, которые имеют различные функции и назначение.



.....
Программа — это последовательность команд, которую выполняет компьютер в процессе обработки данных при решении какой-либо задачи. Каждая отдельная команда представляет собой элементарную инструкцию, предписывающую компьютеру выполнить ту или иную операцию.
.....

В современном мире существует около шести тысяч различных профессий, тысячи различных увлечений и большинство из них в настоящее время имеет какие-либо свои прикладные программные продукты. Причем прикладные программы невозможно использовать без комплексов программ, осуществляющих интерфейсные функции (посреднические между человеком и компьютером, аппаратным и программным обеспечением, между одновременно работающими программами).



.....
Совокупность программ, необходимых для обработки различных данных на компьютере, называется программным обеспечением (ПО) или Software.
.....

Программное обеспечение — это то, что «оживляет» компьютер. Без него компьютер превратится в грудку дорогостоящего, но бесполезного «железа». Сфера применения конкретного компьютера определяется его ПО.

3.1 Классификация программного обеспечения

Программное обеспечение можно разделить на несколько функциональных уровней, которые оно занимает в механизме управления устройствами. Среди этих уровней особую роль играют два уровня: самый нижний и самый верхний. Нижний уровень также называют **аппаратным**, а верхний — **пользовательским**.

Программы нижнего уровня управляют только устройствами. Как правило, это очень лаконичные программы, составленные из числовых кодов, адресованных контроллерам устройств.

Программы верхнего уровня, напротив, устройствами не управляют. Их основная задача — обмен данными с человеком и передача полученных от него управляющих воздействий программам нижних уровней. От программ верхнего уровня требуется не лаконичность, а комфортность работы человека. Хорошая программа предвидит характер развития событий, вовремя предупреждает оператора о возможных затруднениях, подсказывает ему эффективные действия, дает исчерпывающие справки в ответ на запросы.

Между ПО нижнего и верхнего уровней располагаются программные средства промежуточных уровней. Назначение этих средств — управление потоками данных, проходящих между пользовательским и аппаратным уровнями. Во многих случаях передача команд сверху вниз сопровождается их **трансляцией**, в ходе которой команда сначала распознается (интерпретируется), а затем заменяется новой командой (а чаще — группой команд), понятной программам нижележащего уровня.

Классифицируя программы по функциональному уровню, можно предложить следующую структуру программного обеспечения (рис. 3.1).

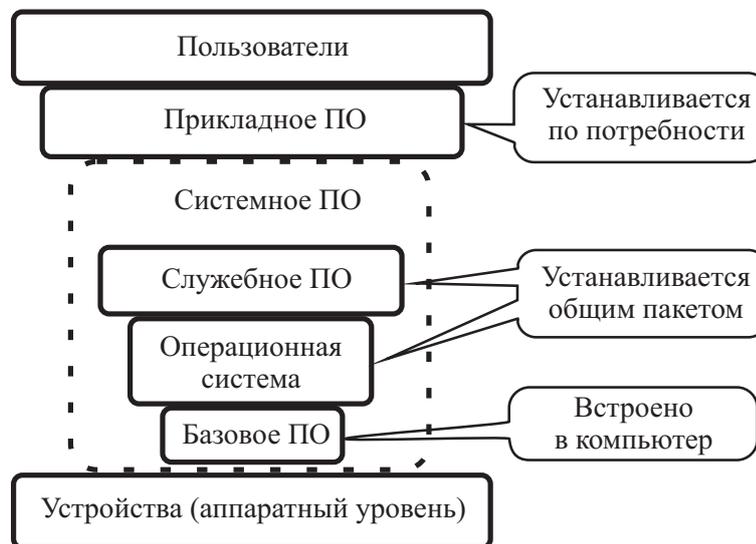


Рис. 3.1 – Архитектура ПО

Базовое ПО в архитектуре компьютера занимает особое положение. С одной стороны, его можно рассматривать как составную часть аппаратных средств, с другой стороны, оно является одним из программных модулей *операционной системы*.

Основу *системного ПО* составляют программы, входящие в *операционные системы* компьютеров. Задача таких программ — управление работой всех устройств

компьютерной системы и организация взаимодействия отдельных процессов, протекающих в компьютере во время выполнения программ. Сюда относятся и программы, обеспечивающие отображение данных на дисплее в удобном для пользователя виде, диалоговые программы для общения на ограниченном естественном языке, а также системы трансляции, переводящие на машинный язык программы, написанные на языках программирования.

Другой комплекс программ — *служебные*. Это различные сервисные программы, используемые при работе или техническом обслуживании компьютера, — редакторы, отладчики, диагностические программы, архиваторы, программы для борьбы с вирусами и другие вспомогательные программы. Данные программы облегчают пользователю взаимодействие с компьютером. К ним примыкают программы, обеспечивающие работу компьютеров в сети. Они реализуют сетевые протоколы обмена информацией между машинами, работу с распределенными базами данных, удаленную обработку данных.

Вся совокупность программ, образующих ту программную среду, в которой работает компьютер, называется *системным ПО*. И чем богаче системное ПО, тем продуктивнее становится работа на компьютере.

Прикладное ПО — самый большой по объему класс программ, это программы конечного пользователя. Прикладное ПО управляется системными программами и занимает высший функциональный уровень. Сама цель эксплуатации большинства компьютерных систем состоит в обеспечении условий для работы прикладных программ.

В социально-экономическом смысле прикладное ПО — это важное средство автоматизации производственной, научной и творческой деятельности. В информационном обществе уровень развития прикладного ПО напрямую связан с уровнем материально-технического оснащения общества.

Предложенная классификация ПО является в большой мере условной, так как в настоящее время программные продукты многих фирм стали объединять в себе программные элементы из разных классов. Например, операционная система Windows, являясь комплексом системных программ, в своем составе содержит блок служебных программ (дефрагментация, проверка, очистка диска и др.), а также текстовый процессор WordPad, графический редактор Paint, которые принадлежат классу прикладных программ.

3.2 Базовое программное обеспечение

Базовое ПО предназначено для непосредственного управления устройствами и характеризуется тем, что размещается внутри самих устройств. Так, в частности, к этой категории относятся программы *BIOS* (Basic Input/Output System — базовая система ввода/вывода).



.....
BIOS размещается в постоянном запоминающем устройстве материнской платы и отвечает за управление всеми ее компонентами.

Фактически BIOS является неотъемлемой составляющей системной платы и поэтому может быть отнесена к особой категории компьютерных компонентов, занимающих промежуточное положение между аппаратурой и ПО.

Раньше основным назначением BIOS была *поддержка функций ввода/вывода* за счет предоставления операционной системе интерфейса для взаимодействия с аппаратурой. В последнее время ее назначение и функции значительно расширились.

Второй важной функцией BIOS является *процедура тестирования всего установленного на материнской плате оборудования* (за исключением дополнительных плат расширения), проводимая после каждого включения компьютера. В процедуру тестирования входят:

- проверка работоспособности системы управления электропитанием;
- инициализация системных ресурсов и регистров микросхем;
- тестирование оперативной памяти;
- подключение клавиатуры;
- тестирование портов;
- инициализация контроллеров, определение и подключение жестких дисков.

В процессе инициализации и тестирования оборудования BIOS сравнивает данные системной конфигурации с информацией, хранящейся в CMOS — специальной энергозависимой памяти, расположенной на системной плате. Хранение данных в CMOS поддерживается специальной батарейкой, а информация обновляется всякий раз при изменении каких-либо настроек BIOS. Именно эта память хранит последние сведения о системных компонентах, текущую дату и время, а также пароль на вход в BIOS или загрузку операционной системы (если он установлен). При выходе из строя, повреждении или удалении батарейки все данные в CMOS-памяти обнуляются.

Третьей важной функцией BIOS является *загрузка операционной системы*. Определив тип устройства загрузки (жесткий диск, приводы гибких дисков, CD-ROM и др.), BIOS приступает к поиску программы — загрузчика операционной системы на носителе. Когда ответ получен, программа загрузки помещается в оперативную память, откуда и происходит загрузка системной конфигурации и драйверов устройств.

С появлением процессоров Pentium и материнских плат форм-фактора ATX (Advanced Technology Extended — расширенная продвинутая технология) BIOS стала выполнять функции управления потребляемой мощностью и включения/выключения источника питания.

Физически BIOS находится в энергонезависимой перепрограммируемой флэш-памяти, которая вставляется в специальную колодку на материнской плате (на этой микросхеме есть яркая голографическая наклейка с логотипом фирмы — разработчика ПО для BIOS).

Встроенное ПО имеют не только системные платы, но и модемы, цифровые камеры, принтеры, сканеры, а также многие другие устройства.

3.3 Операционные системы



.....
Операционная система (ОС) — это комплекс системных программ для организации взаимодействия пользователя с компьютером, управления другими программами и устройствами компьютера.

Операционная система обычно хранится во внешней памяти компьютера — на диске. При включении компьютера она считывается с дисковой памяти и размещается в оперативной памяти. Этот процесс называется **загрузкой ОС**.

Функции операционной системы:

- осуществление диалога с пользователем;
- ввод/вывод и управление данными;
- планирование и организация процесса обработки программ;
- распределение ресурсов (оперативной памяти, процессора, внешних устройств);
- запуск программ на выполнение;
- всевозможные вспомогательные операции обслуживания;
- передача информации между различными внутренними устройствами;
- программная поддержка работы периферийных устройств (дисплея, клавиатуры, дисковых накопителей, принтера и др.).

3.3.1 Представление данных в виде файлов и каталогов

Операционная система устраняет необходимость сложной работы непосредственно с дисками и предоставляет простой, ориентированный на работу с *файлами* интерфейс. Также ОС скрывает подробности работы с прерываниями, счетчиками времени, организацией памяти и другими низкоуровневыми элементами. В каждом случае процедура, предлагаемая ОС, намного проще и удобнее в обращении, чем те действия, которые требует выполнить основное оборудование. С точки зрения пользователя ОС выполняет функцию виртуальной машины, с которой проще и легче работать, чем непосредственно с аппаратным обеспечением, составляющим реальный компьютер.

Наиболее удобной для доступа к долговременным устройствам хранения информации оказалась система, при которой пользователь назначает для той или иной совокупности данных некоторое имя.



.....
Поименованная область данных на носителях информации называется файлом.

Точные правила именования файлов варьируются от системы к системе, но все современные ОС поддерживают использование в качестве имен файлов 8-символьные текстовые строки. Часто в именах файлов также разрешается использование цифр и специальных символов. Многие операционные системы поддерживают имена файлов длиной до 255 символов. В некоторых ОС различаются прописные и строчные символы, в других, таких, как MS-DOS, нет.

Во многих ОС имя файла может состоять из двух частей, разделенных точкой, например Program.exe. Часть имени файла после *последней* точки (точек в имени файла может быть и несколько) называется **расширением файла** и обычно означает тип файла. В некоторых операционных системах, например в UNIX, расширения файлов являются просто соглашениями, и ОС не требует от пользователя их строго придерживаться. Так, файл file.txt может быть текстовым файлом, но это скорее памятка пользователю, а не руководство к действию для UNIX. Система Windows, напротив, распознает расширения файлов и назначает каждому расширению определенную программу. Пользователи или процессы могут регистрировать расширения в Windows, указывая программу, создающую данное расширение. При двойном щелчке левой кнопки «мыши» на имени файла запускается программа, назначенная этому расширению, с именем файла в качестве параметра. Например, двойной щелчок «мышью» на имени текстового файла file.txt запускает текстовый редактор Блокнот, который открывает этот файл.

Большое количество файлов удобно упорядочивать, объединяя их в группы по некоторому общему признаку. Для этого используются *каталоги*.



.....
Каталог (в Windows папка) — это файл, в котором хранятся данные о местоположении на диске других каталогов и файлов, виртуально вложенных в него.

«Виртуально» означает, что для пользователя содержимое каталога отображается именно как вложенные в него объекты (каталоги и файлы), с которыми пользователю предоставляется возможность производить необходимые действия. При таком подходе каждый пользователь может создать столько каталогов, сколько ему нужно, группируя файлы естественным образом. В результате формируется иерархическая структура — **дерево каталогов** (рис. 3.2).



.....
Все каталоги и файлы находятся в основном каталоге, называемом корневым каталогом.

В иерархии каталогов файл указывается при помощи *полного имени*.



.....
Полное имя файла — это текстовая строка, состоящая из пути к файлу и имени самого файла.

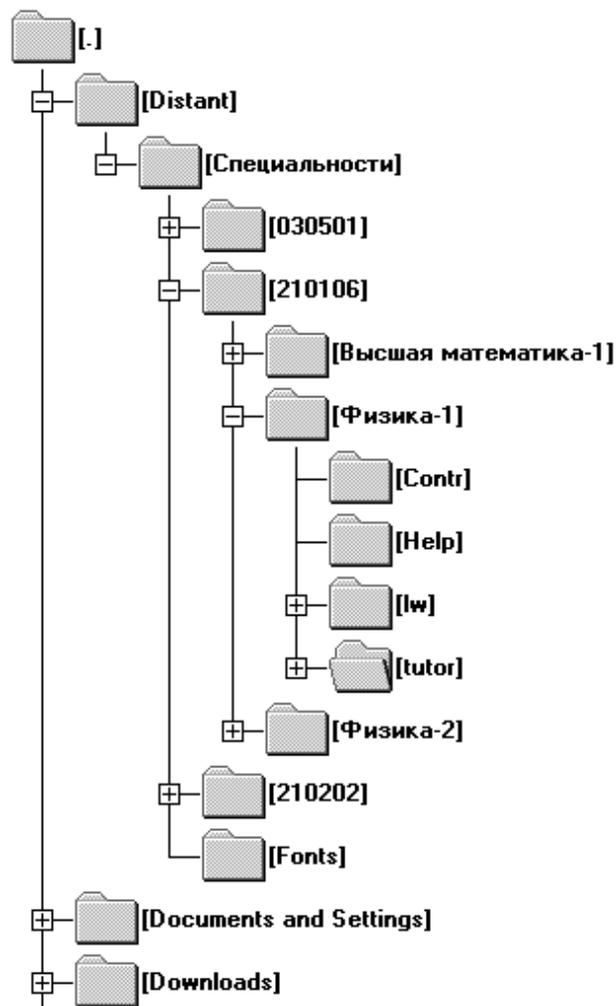


Рис. 3.2 – Дерево каталогов



.....
 Путь к файлу представляет собой последовательность имен каталогов от корневого до того, в котором содержится файл.

В качестве примера рассмотрим полное имя файла в ОС Windows:

C:\DISTANT\Специальности\210106\Физика-1\tutor\
 \Глава_1.4.1.doc

Обратной косой чертой «\» разделяются компоненты полного имени файла.

C:\DISTANT\Специальности\210106\Физика-1\tutor — это путь к файлу, где C: — корневая папка с именем диска, за которой через «\» следуют имена вложенных друг в друга папок вплоть до tutor.

Файл Глава1_4.1.doc находится в самой нижней папке указанной последовательности — tutor. Имя файла содержит расширение doc, связанное с текстовым процессором Word.

3.3.2 Базовые понятия операционных систем

Концепция, рассматривающая операционную систему, прежде всего как удобный интерфейс пользователя, — это взгляд сверху вниз. Альтернативный взгляд снизу вверх дает представление об ОС как о механизме управления всеми частями компьютера. Современные компьютеры состоят из процессоров, памяти, дисков, сетевого оборудования, принтеров и огромного количества других устройств. В соответствии со вторым подходом работа ОС заключается в обеспечении организованного и контролируемого распределения процессоров, памяти и устройств ввода/вывода между различными программами, состязаясь за право их использовать.

Файловая система является базовым понятием, которое поддерживается всеми ОС, и описанная выше модель представления данных в виде файлов и каталогов включена в нее.



.....
Файловая система (ФС) — это часть ОС, работающая с файлами, обеспечивающая хранение данных на дисках и доступ к ним.

Любая файловая система предназначена для хранения данных о физическом размещении частей файла. В ФС существует минимальная единица информации — **кластер**, размер которого является нижним пределом размера записываемой на носитель информации в рамках ФС. Не следует путать понятие кластера с понятием **сектора**, который является минимальной единицей информации со стороны аппаратного обеспечения. От ФС требуется четкое выполнение следующих действий:

- определение физического расположения частей файла;
- установление наличия свободного места и выделение его для вновь создаваемых файлов.

Еще одно ключевое понятие операционной системы — *процесс*.



.....
Процессом называют программу в момент ее выполнения.

С каждым процессом связывается его **адресное пространство** — список адресов в памяти от некоторого минимума до некоторого максимума. По этим адресам процесс может занести данные и прочесть их. Адресное пространство содержит саму программу и данные к ней.

Чтобы лучше разобраться в понятии процесса, проведем аналогию с системой, работающей в режиме разделения времени. Предположим, ОС решает остановить работу одного процесса и запустить другой, потому что первый израсходовал отведенную для него часть рабочего времени центрального процессора. Позже остановленный процесс должен быть запущен снова из того же состояния, в каком его остановили. Следовательно, все данные о процессе нужно где-либо сохранить. Так, процесс может иметь несколько одновременно открытых для чтения файлов. Связанный с каждым файлом указатель дает текущую позицию, т. е. номер бай-

та или записи, которые будут прочитаны после повторного запуска процесса. При временном прекращении действия процесса все указатели нужно сохранить так, чтобы команда чтения, выполненная после возобновления процесса, прочла правильные данные.

Все данные о каждом процессе хранятся в таблице ОС. Эта таблица называется **таблицей процессов** и представляет собой связанный список структур, по одной на каждый существующий в данный момент процесс.

В каждом компьютере есть оперативная память, используемая для хранения исполняемых программ. В простых ОС в конкретный момент времени в памяти может находиться только одна программа. Более сложные системы позволяют одновременно хранить в памяти несколько программ. Для того чтобы они не мешали друг другу, необходим защитный механизм. Этот механизм управляется ОС.

Другой важный, связанный с памятью вопрос — **управление адресным пространством процессов**. Обычно под каждый процесс отводится некоторое множество адресов, которые он может использовать. В простейшем случае, когда максимальная величина адресного пространства для процесса меньше оперативной памяти, процесс заполняет свое адресное пространство, и памяти хватает на то, чтобы содержать его целиком. Однако что произойдет, если адресное пространство процесса окажется больше, чем оперативная память компьютера, а процессу потребуется использовать его целиком? В этом случае используется метод, называемый **виртуальной памятью**, при котором ОС хранит часть адресов в оперативной памяти, а часть на диске и меняет их местами при необходимости. Управление памятью — важная функция операционной системы.

С появлением персональных компьютеров с их огромным разнообразием устройств ввода/вывода от ОС потребовалась программная поддержка этих устройств.



.....
Специальная программа для управления каждым устройством ввода/вывода, подключенным к компьютеру, называется драйвером устройства.

Каждый драйвер поддерживает один тип устройства или, максимум, класс близких устройств. Например, драйвер дисков может поддерживать различные диски, отличающиеся размерами и скоростями. Однако «мышь» и джойстик отличаются настолько сильно, что требуют использования различных драйверов.

Драйвер обычно пишется производителем устройства и распространяется вместе с устройством. Поскольку для каждой ОС требуются специальные драйверы, производители устройств обычно поставляют драйверы для нескольких наиболее популярных операционных систем.

3.3.3 Виды операционных систем

Операционные системы появились и развивались в процессе совершенствования аппаратного обеспечения компьютеров, поэтому эти события исторически тесно связаны. Развитие компьютеров привело к появлению огромного количества различных ОС, из которых далеко не все широко известны.

На самом верхнем уровне находятся *ОС для мэйнфреймов*. Эти огромные машины еще можно встретить в больших организациях. Мэйнфреймы отличаются от персональных компьютеров по своим возможностям ввода/вывода. Довольно часто встречаются мэйнфреймы с тысячами дисков и терабайтами данных. Мэйнфреймы выступают в виде мощных web-серверов и серверов крупных предприятий и корпораций. Операционные системы для мэйнфреймов в основном ориентированы на обработку множества одновременных заданий, большинству из которых требуется огромное количество операций ввода/вывода. Обычно они выполняют три вида операций: пакетную обработку, обработку транзакций (групповые операции) и разделение времени. При пакетной обработке выполняются стандартные задания пользователей, работающих в интерактивном режиме. Системы обработки транзакций управляют очень большим количеством запросов, например бронированием авиабилетов. Каждый отдельный запрос невелик, но система должна отвечать на сотни и тысячи запросов в секунду. Системы, работающие в режиме разделения времени, позволяют множеству удаленных пользователей одновременно выполнять свои задания на одной машине, например работать с большой базой данных. Все эти функции тесно связаны между собой, и ОС мэйнфрейма выполняет их все.

Уровнем ниже находятся *серверные ОС*. Серверы представляют собой или многопроцессорные компьютеры, или даже мэйнфреймы. Эти ОС одновременно обслуживают множество пользователей и позволяют им делить между собой программно-аппаратные ресурсы. Серверы также предоставляют возможность работы с печатающими устройствами, файлами или Интернетом. У Интернет-провайдеров обычно работают несколько серверов для того, чтобы поддерживать одновременный доступ к сети множества клиентов. На серверах хранятся страницы web-сайтов и обрабатываются входящие запросы. UNIX и Windows 2008 являются типичными серверными ОС.

Следующую категорию составляют *ОС для персональных компьютеров*. Их работа заключается в предоставлении удобного интерфейса для одного пользователя. Такие системы широко используются в повседневной работе. Основными ОС в этой категории являются системы на платформах Windows, Linux и ОС компьютера Macintosh.

Еще один вид ОС — это *системы реального времени*. Главным параметром таких систем является время. Например, в системах управления производством компьютеры, работающие в режиме реального времени, собирают данные о промышленном процессе и используют их для управления оборудованием. Такие процессы должны удовлетворять жестким временным требованиям. Если, например, по конвейеру передвигается автомобиль, то каждое действие должно быть осуществлено в строго определенный момент времени. Если сварочный робот сварит шов слишком рано или слишком поздно, то нанесет непоправимый вред изделию.

Встроенные ОС используются в карманных компьютерах и бытовой технике. Карманный компьютер — это маленький компьютер, помещающийся в кармане и выполняющий небольшой набор функций, например телефонной книжки и блокнота. Встроенные системы, управляющие работой устройств бытовой техники, обладают теми же характеристиками, что и системы реального времени.

Самые маленькие ОС работают на смарт-картах, представляющих собой устройство размером с кредитную карту и содержащих центральный процессор.

На такие ОС накладываются очень жесткие ограничения по мощности процессора и памяти. Некоторые из них могут управлять только одной операцией, например электронным платежом.

3.3.4 Операционная система UNIX

UNIX является многозадачной многопользовательской системой, способной обеспечить одновременную работу очень большого количества пользователей, каждый из которых может сразу решать несколько задач.

Файловая система UNIX — иерархическая, с файлами и каталогами. В отличие от Windows в UNIX все диски монтируются в единое дерево каталогов, начинающееся в одном корне. Отдельные файлы могут быть связаны с любым каталогом дерева.

UNIX — мощная и гибкая система. Она характеризуется небольшим количеством базовых элементов, которые можно комбинировать бесконечным числом способов, чтобы приспособить их для конкретного приложения. Ядро UNIX написано на языке высокого уровня C (Си), что позволяет программистам быстро переносить UNIX на другие аппаратные платформы, достаточно легко внося в эту ОС серьезные изменения и дополнения. Одно из основных правил UNIX заключается в том, что каждая программа должна выполнять всего одну функцию, но делать это хорошо.

В отличие от Windows с ее объектно-ориентированным подходом и графическим интерфейсом, который скрывает от пользователя практически все детали системы, UNIX иллюстрирует множество важных принципов построения ОС. Но именно поэтому система UNIX предназначена для опытных пользователей и требует от них определенных навыков в обращении с компьютером.

3.3.5 Операционная система Windows

Windows — многозадачная ОС, которая позволяет параллельно работать с несколькими задачами. Активная работа осуществляется с одной задачей, тогда как остальные программы выполняются в фоновом режиме.

Объектно-ориентированный подход, реализованный в Windows, предполагает, что первичными являются объекты, а обрабатывающие их инструменты вторичны. Человек выбирает нужный объект (окно, папку, файл и др.), а система предлагает на выбор возможные действия или автоматически предоставляет необходимые для обработки средства. Каждый объект Windows уникален, т. е. имеет свой оригинальный набор свойств. И с объектом можно выполнять различные действия, начиная с рабочего стола, у которого, например, можно изменить фоновый рисунок, и заканчивая символом, форма, размеры и оформление которого также могут быть изменены.

Система Windows наделена графическим интерфейсом, ориентированным на работу с объектами при помощи манипулятора «мышь». Обычно выделение объекта или запуск действия осуществляется левой кнопкой «мыши»; правая кнопка предназначена для вызова контекстного меню. Также возможно перемещать и копировать объекты, перетаскивая их «мышью» (способ **Drag&Drop** — перетащить

и оставить). Такая технология настолько упрощает работу в Windows, что ее основы постигаются пользователем на логически-ассоциативном уровне.

Важнейшее проявление интеграционных свойств Windows — это возможность объединять в одном файле объекты различной структуры: тексты, рисунки, таблицы и т. д. и в дальнейшем редактировать каждый объект с использованием средств той программы, которой он создавался. В Windows имеется **Буфер обмена** (часть оперативной памяти и специальная программа), с помощью которого можно перемещать или копировать объекты из одного места в другое, в том числе вставлять данные, созданные одной программой, в файл другой программы. Вставляемый объект становится частью файла-приемника. При этом вместе с объектом сохраняются сведения и о его программе, которая запускается двойным щелчком «мыши» на объекте. Между файлом-источником встраиваемого объекта и файлом-приемником устанавливается связь так, что при изменении данных в файле-источнике они автоматически обновляются в файле-приемнике. Описанные возможности предоставляет **OLE-технология** (Object Linking and Embedding — объектное связывание и встраивание), которую поддерживает большинство приложений Windows.

Современные системы Windows предполагают постоянную работу в глобальном пространстве Интернета. Для этого компьютер должен быть постоянно включен и иметь канал связи. Тогда он сможет самостоятельно подключиться к Интернету, принять поступившую почту и отправить подготовленные письма, получить информацию по оформленным подпискам, автоматически обновить свое ПО и т. п.

В настоящее время большинство персональных компьютеров в мире работают под управлением различных версий ОС Windows фирмы Microsoft.

3.4 Служебные программы



.....
Служебные программы (утилиты) — это программы, используемые при работе или техническом обслуживании компьютера для выполнения вспомогательных функций.

К утилитам относятся программы работы с файлами и каталогами (файловые менеджеры), сжатия информации (архиваторы), диагностирования системы и аппаратуры, восстановления поврежденных данных, оптимизации дискового пространства, антивирусные средства (см. главу 6) и т. п.

3.4.1 Файловые менеджеры

С момента появления программы Norton Commander для операционной системы MS DOS *файловые менеджеры* стали распространенным приложением на любом компьютере. Многие из них внешне напоминают знаменитого предка, но в плане функциональности ушли далеко вперед.

Файловые менеджеры, которые еще называются **оболочками системы**, предназначены для разнообразной работы с файлами: копирования, переноса, удаления, редактирования текстовых файлов, гибкого запуска программ.

Самыми популярными файл-менеджерами сегодня в России являются *FAR Manager* и *Total Commander* (бывший *Windows Commander*). Все перечисленные программы работают под управлением Windows.

FAR Manager — это Norton-подобный файл-менеджер, который может работать как в полноэкранном, так и в оконном режимах, поддерживает длинные имена файлов, корректно работает с русскими буквами, а встроенный редактор позволяет переключаться между DOS- и Windows-кодировками, окрашивает имена файлов в соответствии с их расширениями, что очень удобно при работе. Среди возможностей программы — определение размеров папок, вызов списка активных задач, передача файлов через FTP-клиент, управление сетевыми и подключенными к компьютеру принтерами, подсветка синтаксиса в исходных текстах программ, поиск и замена символов одновременно во множестве файлов с применением регулярных выражений, средства переименования групп файлов с возможностью использования сложных составных масок, проверка орфографии при обработке текста в редакторе FAR и многое другое. Программа поддерживает большинство известных архивных форматов и позволяет архивировать и разархивировать, просматривать, редактировать и запускать на выполнение файлы из архивов. Предусмотрены развитая система управления горячими клавишами и очень большой набор встроенных функций, которые существенно расширяют возможности FAR Manager, дополняя и модифицируя их.

Если FAR Manager выбирают те пользователи, которые привыкли работать с Norton Commander, то люди, начинавшие с Windows, предпочитают **Total Commander**. Он продолжает добрую традицию двухпанельных файловых менеджеров, но в большей степени ориентирован на Windows-интерфейс. Так же, как и у многих Windows-программ, интерфейс Total Commander легко настраивается, причем отрегулировать можно буквально все: цвета, наборы панелей, цвета для групп файлов, иконки папок и документов. Интерфейс Total Commander позволяет просматривать содержимое носителей с использованием настраиваемых закладок. В итоге в рамках одной панели файл-менеджера можно открыть несколько папок, каждая из которых будет представлена отдельной закладкой, а при переходе между закладками сохраняется состояние папки. Наряду со стандартными возможностями, характерными для данного класса приложений, программа обладает целым рядом особенностей, которые способны существенно ускорить навигацию по дискам и папкам. К таким особенностям относятся: запоминание часто используемых папок, история последних посещенных папок, быстрый поиск, в том числе внутри файлов, многофункциональные возможности настройки горячих клавиш, удобная панель инструментов с поддержкой технологии перетаскивания, удобная работа с комментариями к файлам и папкам и др. Поддержка встроенных функций позволяет реализовать в Total Commander многие полезные возможности, такие как диспетчер задач, редактор реестра, временная панель, управление сервисами, работа с сетевыми папками, карманным компьютером, Linux-разделами и пр.

3.4.2 Архиваторы

Вследствие неуклонно растущих объемов обрабатываемых данных особое значение приобретает их сжатие (архивирование) для долговременного хранения с целью уменьшения занимаемой памяти компьютера. Также архивирование рекомен-

дуются при отправке данных по электронной почте, так как чем меньше размер письма, тем больше вероятность, что оно дойдет без потерь пересылаемых данных.

Программы-архиваторы позволяют за счет применения специальных методов «упаковки» сжимать данные, т. е. создавать файлы меньшего размера, а также объединять несколько файлов в один архивный файл или, напротив, разбивать один файл большого размера на несколько томов-файлов задаваемого меньшего размера.

Популярным архиватором является *WinZip*. Объясняется это тем, что формат ZIP считается мировым стандартом архивирования и имеет самую длительную историю развития. Большинство архивов в Интернете также имеют формат ZIP. За ним следует многими предпочитаемый *WinRAR*. Все эти архиваторы работают под управлением Windows.

Возможностей **WinZip** вполне достаточно для того, чтобы обеспечить надежное и эффективное архивирование данных. Программа работает в двух режимах: классическом и режиме мастера, рассчитанном на новичков. Она ориентирована преимущественно на ZIP-архивы, но при этом поддерживает и другие популярные архивные форматы. В числе возможностей WinZip — поддержка технологии перетаскивания и полная интеграция с программой «Проводник»; создание самораспаковывающихся файлов; поддержка антивирусных программ; отправка архива по электронной почте и пр. Для защиты архива от несанкционированного доступа можно при его создании указать пароль. Кроме того, теперь WinZip позволяет обрабатывать файлы практически неограниченного размера.

WinRAR — признанный лидер среди программ-архиваторов по удобству и массовости применения, поддерживающий в настоящее время самый большой набор языков, включая русский. При этом формат RAR в большинстве случаев обеспечивает значительно лучшее сжатие, чем ZIP, особенно в режиме создания непрерывных архивов. Формат RAR оснащен несколькими очень важными функциями, отсутствующими у ZIP, к числу которых, например, можно отнести функцию добавления информации для восстановления, которая позволяет восстановить физически поврежденный файл, и функцию блокировки архивов для предотвращения случайной модификации особенно ценных данных. Формат RAR позволяет обрабатывать файлы практически неограниченного размера и не только предлагает оригинальные и эффективные алгоритмы для сжатия информации различных типов, но и автоматически применяет нужный алгоритм при сжатии файлов. В число его основных особенностей входят: полная поддержка архивов ZIP; управление архивами других форматов; наличие графической оболочки с поддержкой технологии перетаскивания; поддержка метода solid-архивирования, при котором может быть достигнута степень сжатия, на 10–50% превышающая ту, что дают обычные методы; поддержка многотомных архивов; создание самораспаковывающихся (SFX) обычных и многотомных архивов; возможность создания и использования томов для восстановления, позволяющих воссоздавать недостающие части многотомных архивов; отправка архива по электронной почте и пр. Очень удобна появившаяся в одной из последних версий архиватора возможность поиска файлов внутри архивов. Полезна новая команда «Просмотр архива на вирусы», обеспечивающая распаковку архива во временную папку и проверку ее содержимого установленным на компьютере антивирусом. А команда печати позволяет теперь распечатывать как

обычные, так и архивированные файлы. Программа поддерживает два варианта работы: классический и с помощью мастера.

3.4.3 Стандартные утилиты Windows

В комплект Windows входит пакет стандартных программ, который включает служебные программы (рис. 3.3). Рассмотрим некоторые из утилит системы Windows XP Home Edition.

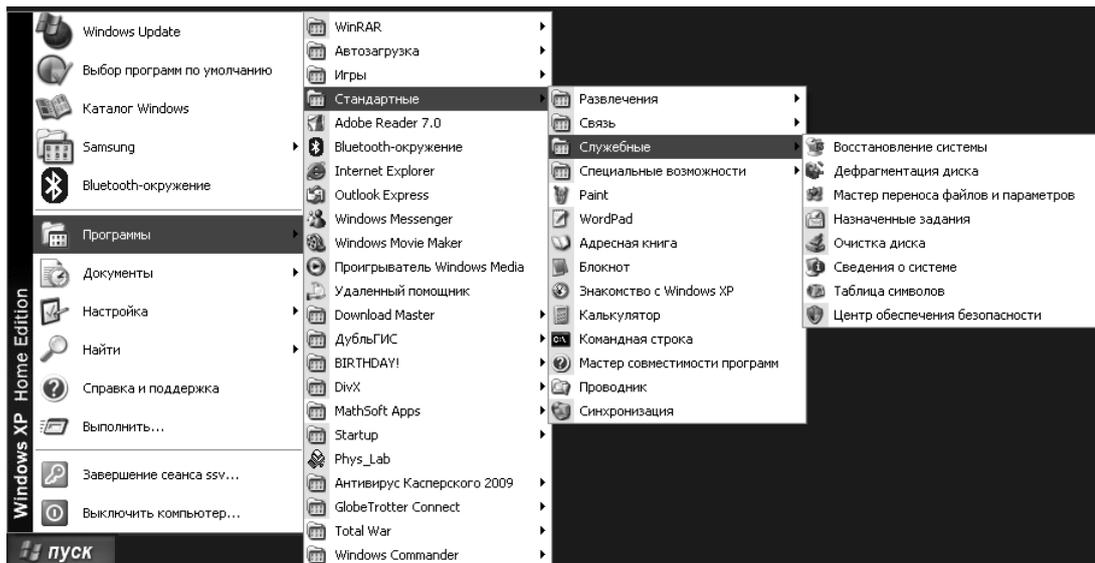


Рис. 3.3 – Содержимое папок Стандартные и Служебные в Главном меню кнопки Пуск

Восстановление системы предназначено для отмены изменений в конфигурации системы и восстановления ее параметров и производительности. Данная утилита позволяет вернуть конфигурацию компьютера в более раннее состояние, называемое контрольной точкой восстановления, без потери текущих данных, таких как документы, почта, избранные ссылки и журнал ссылок. Любые изменения конфигурации, выполняемые программой восстановления, обратимы.

Windows автоматически формирует контрольные точки восстановления, которые называются системными. Кроме этого, можно воспользоваться утилитой для создания собственных точек, что является полезным при установке нового программного обеспечения или внесении изменений в реестр.

Программа **Дефрагментация диска** анализирует размещение файлов и объединяет их отдельные части таким образом, чтобы каждый файл занимал единое непрерывное пространство. В результате чтение и запись файлов выполняются эффективнее. Собирая части файлов, программа дефрагментации также выделяет в единое целое свободное место на диске, что делает менее вероятной фрагментацию новых файлов.

Файл разбивается на фрагменты, если при записи нет свободного участка диска, где он поместился бы целиком. Такая ситуация возникает при активной работе, связанной с многочисленными удалениями и записями файлов. Сильно фрагментированные данные существенно уменьшают скорость работы диска.

До начала дефрагментации рекомендуется максимально очистить диск с помощью соответствующей стандартной утилиты.

Программа **Очистка диска** выполняет проверку потенциальных возможностей увеличения свободного места за счет чистки папок временных файлов (такие файлы создаются, например, при работе с Интернетом) и содержимого Корзины, хранящей удаленные файлы и папки. В предлагаемом программой списке надо оставить флажки на именах тех папок, которые можно очистить.

Состав и вид стандартных утилит в различных версиях ОС Windows немного отличаются друг от друга.

3.5 Прикладное программное обеспечение

В настоящее время везде, где требуется выполнить большие математические расчеты, или производится обработка больших объемов разнообразных данных, или требуется быстрый анализ ситуации с принятием управляющего решения, — компьютеры под управлением прикладных программных продуктов с успехом заменяют человека.



.....
Прикладное программное обеспечение предназначено для решения самых различных задач информационного характера в любой сфере человеческой деятельности.

Прикладное ПО объединяет в себе программы, широко используемые большинством пользователей персональных компьютеров, например текстовые редакторы, электронные таблицы, графические системы, игры, развлечения и многое другое. Также к прикладным относятся программы для решения задач профессионального характера в различных предметных областях.

Здесь необходимо упомянуть о такой разновидности ПО как инструментальные средства, которые более подробно освещаются в главе 7.

Процесс создания новых программ на языке машинных команд очень сложен и кропотлив, поэтому он низкопроизводителен. На практике большинство программ пишется на формальных языках программирования, которые более близки к математическому, следовательно, проще и производительней в работе. Рассматривая программирование как профессиональную деятельность, **инструментальное программное обеспечение** — комплексы программ для создания других программ, — можно отнести к прикладному ПО. Но зачастую инструментальное ПО выделяют в особый класс наравне с системным и прикладным программным обеспечением. Как уже говорилось выше, классификация программ является условной и в большой мере зависит от выбранной точки зрения.

3.5.1 Текстовые редакторы и процессоры

Со времен изобретения письменности текстовые документы используются для хранения и передачи данных. В широком смысле текстовым документом является

и короткая записка, и толстая иллюстрированная книга. Современные информационные технологии позволяют создавать текстовые документы, ориентированные не на печатную страницу, а на просмотр с экрана монитора. Такие документы называются **электронными** и могут включать в себя, кроме текстовых данных, таблиц, диаграмм, фотографий, еще и различные анимационные картинки, звуковые файлы, гиперссылки (т. е. ссылки на данные в собственном компьютере или в Интернете).

В связи с различными возможностями создания и оформления текстовых документов пользователю предоставляется многообразие программных продуктов по работе с этими документами. Условно их можно разделить на *текстовые редакторы* и *текстовые процессоры*.



.....
Текстовые редакторы — это программы для создания и редактирования текстовых документов. Редактированием называется комплекс операций по внутренней (смысловой) и внешней (оформительской) работе над текстом.

Текстовые редакторы обеспечивают основные возможности по подготовке небольших и несложных документов. Сюда входят следующие простые операции:

- ввод алфавитно-цифровой информации;
- перемещение по набранному тексту;
- вставка или удаление символов.

А также более сложные операции по работе с блоками (фрагментами) текста:

- выделение блока;
- удаление блока;
- копирование, перемещение, вставка блока;
- дополнительные сервисы (поиск фрагмента, поиск с заменой, печать документа и т. д.).

Большинство текстовых редакторов ориентировано на работу только с текстовыми файлами, среди которых могут быть тексты программ, написанные на различных языках, конфигурационные файлы, файлы настройки и др.



.....
Текстовые процессоры включают в себя инструменты редакторов и предоставляют более широкий круг возможностей форматирования (шрифты, таблицы, формулы), создания документов, содержащих данные разных типов (вставка графических, звуковых данных), создания электронных документов.

Типичным представителем этой группы является текстовый процессор Word.

Существует еще одна отдельная группа текстовых процессоров — это настольные издательские системы, которые автоматизируют процесс верстки полигра-

фических изданий. Издательские системы отличаются расширенными средствами управления взаимодействия текста с параметрами страницы и графическими объектами, но имеют более слабые возможности по автоматизации ввода и редактирования текста. Их целесообразно применять к документам, которые предварительно обработаны в обычных текстовых процессорах и графических редакторах.

Следует отметить, что растущие возможности текстовых процессоров постепенно приближают их к издательским системам, и такие процессоры, как Word, в состоянии обеспечить набор и распечатку не очень больших изданий.

Кратко рассмотрим стандартные программы: редактор *Блокнот* (NotePad) и процессор *WordPad*, которые встроены в ОС серии Windows (см. рис. 3.3).

Блокнот предназначен для создания небольших неформатированных текстов. Его файлы имеют распознаваемое Windows расширение — `.txt`. Скромные возможности определяют область применения программы как редактора для создания коротких записок, текстов управляющих и системных файлов.

Отдельные части текста в Блокноте не форматировуются, но можно отобразить весь текст в одном из перечисленных в меню шрифтов.



.....
 Набор из четырех основных шрифтов — обычный, *курсив*, **полужирный**, *полужирный курсив* — в совокупности называется гарнитурой шрифта. Каждая гарнитура имеет свое наименование, например, Arial (обычно выделяет заголовки), Times New Roman (общепринято для основного текста), Courier New и другие. Вертикальный размер шрифта (именно он понимается под размером шрифта) измеряется в пунктах, один пункт равен 1/72 дюйма — примерно 0.353 мм. Шрифт размером 10 пунктов, его называют десятый кегль, часто используется в книгах. Четырнадцатый кегль имели печатные машинки, поэтому этот размер шрифта часто применяется сейчас при составлении различных официальных документов.

Дальнейшее развитие возможностей текстового редактора получено в процессоре **WordPad**. Кроме строки меню в нем введены:

- две **панели инструментов** для быстрого доступа к часто используемым пунктам меню (одна называется **стандартная**, другая — **форматирование**);
- **линейка**, позволяющая визуально устанавливать отступы;
- **строка состояния**, дающая дополнительную информацию.

На панелях инструментов собраны **пиктограммы** (значки) наиболее часто выполняемых команд меню: создать, открыть, сохранить, печатать и т. д. Щелчок по пиктограмме равносильно выполнению команды меню. Краткую информацию о назначении пиктограммы можно получить с помощью **всплывающей подсказки**, которая появляется, если навести и задержать немного на значке курсор «мыши».

Попутно заметим, что различные способы выполнить одно и то же действие (через меню, панель инструментов, с помощью комбинаций клавиш, называемых «горячие клавиши») являются элементом гибкого интерфейса, а всплывающая подсказка — элемент дружественного интерфейса.

В документ WordPad можно делать вставки рисунков, формул, звуковых и видеофайлов, таблиц Excel, презентаций Power Point и других объектов. Возможность эта основана на OLE-технологии, позволяющей создавать комплексные документы из разных типов данных, обеспечивать совместную работу нескольких приложений при подготовке одного документа, копировать и переносить объекты между приложениями.

Несмотря на более широкий инструментарий процессора WordPad по сравнению с редактором Блокнот, по-настоящему популярным во всем мире является мощный текстовый процессор Word, который входит в состав интегрированного пакета офисных программ MS Office.

3.5.2 Графические редакторы



.....
Графические редакторы — это отдельные программы и аппаратно-программные комплексы, специализированные на создание и обработку различных изображений на экране монитора.
.....

Как уже было сказано в первой главе, все компьютерные изображения можно разделить на три класса. Поэтому из-за особенностей представления изображения для каждого класса приходится применять свой графический редактор. Разумеется, у них есть общие черты — возможности открывать и сохранять файлы в различных форматах, выбирать нужный цвет или оттенок, инструменты с одинаковыми названиями (карандаш, перо и т. д.) или функциями (выделение, перемещение, масштабирование и др.). Однако принципы реализации процессов рисования и редактирования различны, что обусловлено свойствами растровых и векторных изображений.

Есть тенденция к сближению редакторов векторной и растровой графики. Большинство современных векторных редакторов способны использовать растровые картинки в качестве фона, а то и переводить в векторный формат части изображения. Также обычно имеются средства редактирования загруженного фонового изображения. Кроме того, может осуществляться непосредственный перевод нарисованного векторного изображения в растровый формат и дальнейшее использование как нередатируемого растрового элемента. Причем все это помимо обычно имеющихся конвертеров из векторного формата в растровый с получением соответствующего файла.

В стандартные программы Windows (рис. 3.3) входит **Paint** — простейший растровый редактор, предназначенный для создания и редактирования изображений в основном формате Windows (BMP) и форматах Интернета (GIF и JPEG). Весь набор инструментов отображен на панели в левой части окна Paint и включает в себя следующее:

Выделение произвольной области — ограничение любого фрагмента рисунка.

Выделение — выбор некоторой прямоугольной области.

Выделенный фрагмент можно переносить, копировать, удалять или передавать его в другое приложение.

Ластик — стирание части рисунка. Удаленный участок закрашивается цветом фона.

Заливка — закрашивание выбранным цветом внутренней части замкнутой области. Для этого необходимо щелкнуть левой кнопкой «мыши» в любой точке внутри нужного участка. Если он незамкнутый, то закрасится весь рисунок.

Выбор цветов — определение цвета имеющегося элемента рисунка для дальнейшего рисования этим цветом. Такой быстрый выбор уже применявшегося цвета избавляет от необходимости его поиска в *палитре*.

Масштаб — на выбор двух-, шести-, восьмикратное увеличение/уменьшение рисунка.

Карандаш — рисование выбранным цветом следа курсора «мыши» толщиной один пиксель.

Кисть — действие похоже на карандаш, но можно менять форму кисти — различные кружки, квадратики, линии.

Распылитель — рисование с эффектом распыления краски. Густота распыления является задаваемым параметром.

Надпись — создание текстовой области для ввода каких-либо пояснений, комментариев. Размеры текстовой области можно изменять, захватывая и перемещая ее границы левой кнопкой «мыши». Форматирование надписи производится с помощью панели атрибутов текста.

Линия — изображение прямой линии (отрезка) выбранного цвета и толщины. Начало и конец отрезка — места, где соответственно была нажата, а затем отпущена левая кнопка «мыши».

Кривая — построение гладкой кривой линии. Сначала проводят прямую линию выбранного цвета и толщины, которую затем можно <i>дважды</i> изогнуть, захватывая в любых местах и перемещая в нужных направлениях левой кнопкой «мыши».

Прямоугольник — рисование прямоугольника задаваемого вида: только с рамкой, с рамкой и заполнением, только с заполнением. Перечисленные виды изображения имеются и у нижеследующих фигур.

Многоугольник. Сначала аналогично прямой линии намечается первая сторона многоугольника. Остальные стороны достраиваются программой автоматически к вершинам, которые указываются щелчком левой кнопки «мыши».

Эллипс и Скругленный прямоугольник — рисование эллипса (окружности) и прямоугольника с округленными вершинами соответственно.

В левой нижней части окна программы имеется **палитра** с 28 предлагаемыми цветами для линий и заливки.

Описанные инструменты редактора Paint позволяют создавать простые иллюстрации, в основном схемы, диаграммы и графики, которые можно встраивать в документы других приложений.

3.5.3 Офисные интегрированные программные средства

Каждая отдельная программа, предназначенная для решения какой-либо прикладной задачи, не всегда может в полной мере удовлетворить все запросы пользователя. Например, информацию, взятую из некоторой базы данных, бывает удобно

обработать с помощью электронных таблиц, а результаты, оформленные в виде наглядных таблиц и диаграмм, поместить в отчет, представляющий собой текстовый документ, который был создан в текстовом процессоре.



.....
Программные пакеты, образующие единую унифицированную среду для создания и обработки самых различных данных, называются интегрированными программными средствами.
.....

Наиболее распространенный продукт этого класса — пакет офисных программ **Microsoft Office**, в состав которого обычно входят:

- 1) текстовый процессор Word;
- 2) электронные таблицы Excel;
- 3) пакет подготовки и демонстрации презентаций PowerPoint;
- 4) система управления базами данных (СУБД) Access;
- 5) организатор и планировщик работы Outlook и др.

Перечисленные приложения тесно связаны как между собой, так и с ОС Windows. Это означает, что они имеют удобные возможности обмена любыми данными.

Перемещение объекта между приложениями производится с помощью операций копирования в буфер обмена Windows и последующей вставки в документ. Эти операции доступны посредством основного меню, панелей инструментов, «горячих клавиш» или контекстного меню, которое появляется при нажатии на правую кнопку «мыши». Наконец, объект можно переместить, просто захватив левой кнопкой «мыши» в одном документе и перетащив на подходящее место в другом (технология Drag&Drop).

Также возможно использование механизма связывания объекта и его копии в другом приложении — OLE-технология. В этом случае при изменении данных в документе-источнике они изменяются и в документе-приемнике. Объектное встраивание и связывание позволяет эффективно работать с составными документами, которые объединяют различные источники. Например, данные, взятые из СУБД Access, будут обновляться автоматически при изменении базы, и любой, кто захочет ознакомиться с таким документом, увидит самую свежую информацию. Или еще пример: обновление исходных данных в таблице Excel приведет к соответствующей модификации построенной на их основе диаграммы, а вместе с ней и той копии диаграммы, которая содержится в документе Word.

Семейство MS Office содержит набор инструментов, общих для всех приложений. К ним относятся механизмы проверки правописания и грамматики, средства для рисования, инструменты для создания красочных заголовков, редакторы диаграмм, математических формул, фотоизображений, библиотека картинок и т. д. Сюда же относятся и панели инструментов, и даже меню, которые являются стандартными элементами любого офисного приложения.

Работая в сетевых операционных системах, приложения MS Office поддерживают совместную групповую работу нескольких человек над общими документа-

ми. Существует возможность использования материалов, расположенных не только на локальном диске, но и на соседних компьютерах или на сервере сети. Например, электронной таблицей Excel или базой данных Access могут в одно и то же время пользоваться разные люди. Текстовый процессор Word дает возможность создавать документ, над разными частями которого могут одновременно работать несколько человек. Программа PowerPoint позволяет проводить видеоконференции, когда презентация сразу отображается на экранах всех участников конференции. Кроме того, все офисные программы поддерживают работу с электронной почтой. При работе над документом, требующим корректуры нескольких авторов, можно послать этот документ по почте в режиме последовательной рассылки. Как только один участник работы закончит вносить поправки, документ отправляется к следующему. После того, как документ обойдет всех указанных в рассылке людей, он возвращается к тому, кто его посылал. При этом все пометки и исправления будут отражены отдельным цветом для каждого участника совместной работы. Исправления можно просмотреть и решить, какие из них нужно принять, а какие отменить.

Пакет MS Office имеет широкий набор функций, рассчитанных на самые различные категории пользователей. Однако он не может содержать абсолютно все специфические инструменты, которые могут потребоваться. Поэтому в приложениях встроен универсальный язык программирования MS Visual Basic. Квалифицированный пользователь может запрограммировать собственные функции, выполняющие необходимые действия, и даже создать свое приложение, отвечающее требованиям конкретной организации или подразделения. Одна и та же программа на языке Visual Basic применима к объектам и документам всего офисного пакета.



Контрольные вопросы по главе 3

- 1) Расскажите о классификации ПО.
- 2) Каковы основные функции BIOS?
- 3) С каких точек зрения можно рассматривать ОС? Рассмотрите базовые понятия ОС.
- 4) Перечислите и охарактеризуйте виды ОС.
- 5) Опишите класс прикладного ПО. Приведите примеры ППО.
- 6) Перечислите преимущества интегрированных пакетов ПО.

Глава 4

СЕТИ ЭВМ

Эта глава посвящена компьютерным сетям — их назначению, классификации. Здесь рассмотрены основные принципы работы глобальной сети Интернет, рассказано про зарождение и становление данной сети, а также об ее сервисах.

4.1 Общие сведения о сетях ЭВМ

У людей всегда была потребность общаться друг с другом, то есть обмениваться информацией. В качестве примера первых информационных сетей можно привести телефонные сети. Благодаря техническому прогрессу расширились возможности общения (появились пейджеры, сотовая связь, Интернет и т. д). Так что же такое сеть? В самом общем случае *сеть — это некая система, позволяющая производить обмен информацией.*

Современные традиционные технологии нуждаются во все более совершенных средствах обработки информации. Поэтому потребности в таких средствах постоянно растут. Объединение компьютеров и средств коммуникации оказало существенное влияние на принципы организации компьютерных систем. Модель, в которой один компьютер выполнял всю необходимую работу по обработке данных, уступила место модели, представляющей собой большое количество отдельных, но связанных между собой компьютеров. Такие системы и называются *компьютерными сетями*. Два или более компьютера называются связанными между собой, если они могут обмениваться информацией.

Минимальный набор компонентов, составляющий базовую коммуникационную модель, состоит из источника, приемника, среды передачи, сообщения.

Источником и приемником могут быть два разговаривающих между собой человека. В сети источником и приемником могут быть персональный компьютер и главная ЭВМ или спутник и принимающая антенна.

Средой передачи или каналом может быть телефонная линия, кабель или воздух, по которому распространяется микроволновое излучение.

Сообщение представляет собой информацию, передаваемую от источника к приемнику.

К *преимуществам* использования сетей можно отнести:

- Быстрый обмен информацией между пользователями.
- Общий доступ к ресурсам.
- Оптимальное распределение нагрузки между несколькими компьютерами.
- Возможность резервирования для повышения устойчивости всей системы к отказам.
- Создание гибкой рабочей среды.

Компьютерная сеть включает все аппаратное и программное обеспечение, необходимое для подключения компьютеров и другого электронного оборудования к каналу, по которому они могут общаться друг с другом. Устройства, которые взаимодействуют с другими устройствами в сети, называются узлами, станциями или сетевыми устройствами. Число узлов может составлять от двух до многих тысяч. Современные коммуникационные технологии основаны на использовании информационных сетей.

4.2 Назначение компьютерных сетей

Для каких же целей используются компьютерные сети?

Первая цель — предоставление доступа к программам, оборудованию и особенно данным для любого пользователя сети. Это называется совместным использованием ресурсов.

Вторая цель — обеспечение высокой надежности при помощи альтернативных источников информации. Например, все файлы могут быть расположены на двух или трех машинах одновременно, так что, если одна из них недоступна по какой-либо причине, то используются другие копии. Возможность продолжать работу, несмотря на аппаратные проблемы, имеет большое значение для военных и банковских задач, воздушного транспорта, безопасности ядерного реактора и т. п.

Третья цель — экономия средств. Небольшие компьютеры обладают значительно лучшим соотношением цена-производительность, нежели большие. Это обстоятельство заставляет разработчиков создавать системы на основе модели «клиент-сервер». Обмен информацией в модели «клиент-сервер» обычно принимает форму запроса серверу на выполнение каких-либо действий. Сервер выполняет работу и отправляет ответ клиенту. Обычно в сети количество клиентов значительно больше числа используемых ими серверов.

Четвертая цель — масштабируемость, т. е. способность увеличивать производительность системы по мере роста нагрузки. В случае модели «клиент-сервер» новые клиенты и новые серверы могут добавляться по мере необходимости.

Пятая цель — ускорение передачи информации. Компьютерная сеть является мощным средством связи между удаленными друг от друга пользователями. Если один из них изменяет документ, находящийся на сервере, в режиме on-line, остальные могут немедленно увидеть эти изменения.

4.3 Компоненты аппаратного и программного обеспечения сетей

Рабочая станция — компьютер, подключенный к сети и работающий под управлением локальной операционной системы.

Серверы сети — выполняют функции управления распределением сетевых ресурсов и предоставления различного рода сервисных услуг.

Коммуникационные узлы — модемы, повторители, коммутаторы (мосты), маршрутизаторы, шлюзы и др.

Средства связи — коаксиальный кабель, витая пара, оптоволоконный кабель, телефон.

Программное обеспечение — для решения задач обработки информации, осуществления планирования и организации коллективного доступа к информационным ресурсам сети, динамического распределения этих ресурсов.

4.4 Классификация сетей

Вычислительные сети классифицируются по ряду признаков:

- технология передачи информации;
- размеры (территориальная распространенность);
- принадлежность;
- способ управления;
- по топологии (способу организации связей);
- тип среды передачи;
- скорость передачи информации.

4.4.1 По технологии передачи

Существуют два типа *технологии передачи*:

- широковещательные сети;
- сети с передачей от узла к узлу.

Широковещательные сети обладают единым каналом связи, совместно используемым всеми машинами сети. Короткие сообщения, называемые пакетами, посылаемые одной машиной, принимаются всеми машинами. Поле адреса в пакете указывает, кому направляется сообщение. При получении пакета машина проверяет его адресное поле. Если пакет адресован этой машине, она обрабатывает пакет. Пакеты, адресованные другим машинам, игнорируются.

Сети с передачей от узла к узлу состоят из большого количества соединенных пар машин. В такой сети пакету необходимо пройти через ряд промежуточных машин, чтобы добраться до пункта назначения. Часто при этом существует несколько возможных путей от источника к получателю. Обычно небольшие сети используют широковещательную передачу, тогда как в крупных сетях применяется передача от узла к узлу.

4.4.2 По территориальной распространенности

Другим критерием классификации сетей является их *размер (территориальная распространенность)*. Сети можно разделить на локальные, региональные, корпоративные и глобальные.

Локальные сети (ЛВС) (или LAN — Local Area Network) — это сети, охватывающие ограниченную территорию (обычно в пределах удаленности станций не более чем на несколько десятков или сотен метров друг от друга, реже на 1–2 км) и, следовательно, размещающиеся, как правило, в одном здании или на территории какой-либо организации. Их часто используют для предоставления совместного доступа компьютеров к ресурсам (например, принтерам) и обмена информацией. Отличительной чертой ЛВС является большая скорость передачи данных. Обычные ЛВС имеют пропускную способность канала связи от 10 до 100 Мбит/с, небольшую задержку — десятые доли мкс и низкий уровень ошибок.

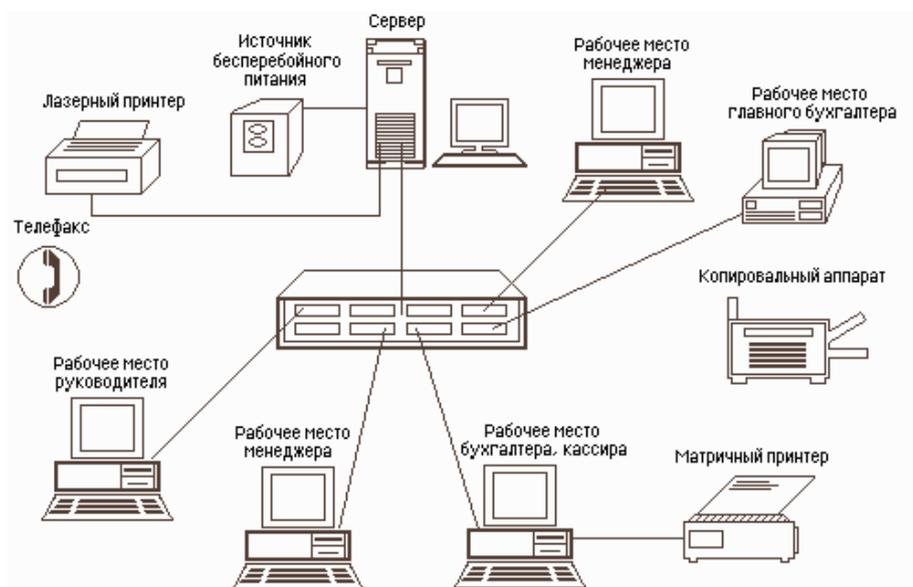


Рис. 4.1 – Офисная локальная сеть

Муниципальные или региональные сети (MAN — Metropolitan AN) являются увеличенными версиями локальных сетей и обычно используют схожие технологии. Такая сеть может объединять несколько предприятий, корпорации или город. Муниципальная сеть может поддерживать передачу цифровых данных, звука и включать в себя кабельное телевидение. Обычно муниципальная сеть не содержит переключающих элементов для переадресации пакетов во внешние линии, что упрощает структуру сети. Региональные сети разработаны для поддержки больших расстояний. Они могут использоваться для связывания нескольких ЛВС вместе в высокоскоростные интегрированные сетевые системы.

Глобальные сети (Wide AN или ГВС) охватывают значительную территорию, часто целую страну или даже континент. Они объединяют множество машин, предназначенных для выполнения приложений. Эти машины называются хостами. Хосты соединяются коммуникационными подсетями или просто подсетями. Задачей подсети является передача сообщений от хоста хосту, подобно тому, как телефон-

ная система переносит слова говорящего слушающему. То есть коммуникативный аспект сети — подсеть отделен от прикладного аспекта — хостов, что значительно упрощает структуру сети. Коммуникации по ГВС осуществляются посредством телефонных линий, спутниковой связи или наземных микроволновых систем. ГВС зачастую создаются путем объединения локальных вычислительных сетей (ЛВС) и региональных вычислительных сетей (РВС). Фактически объединение изолированных ЛВС и РВС в форму ГВС является современной тенденцией в области сетей. По сравнению с ЛВС большинство ГВС отличаются низкой скоростью передачи и более высоким уровнем ошибок. Новые технологии в области ГВС призваны разрешить эти проблемы.

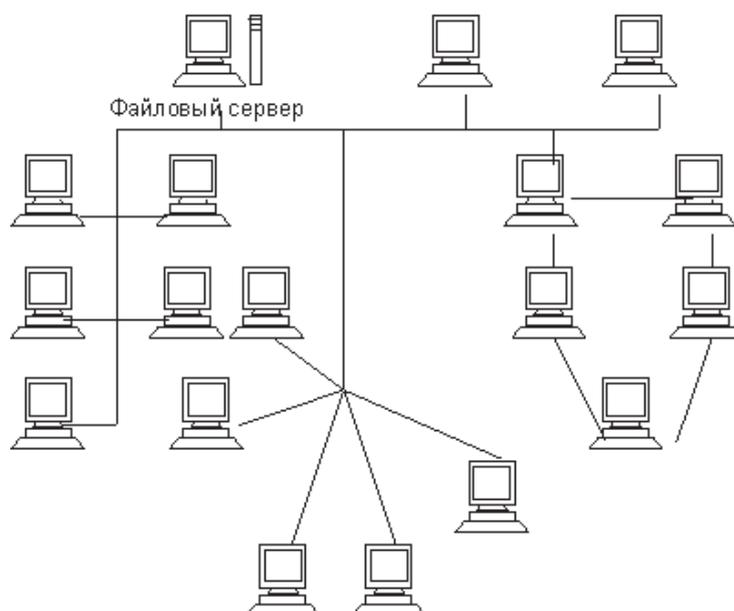


Рис. 4.2 – Глобальная компьютерная сеть

Корпоративные сети — совокупность связанных между собой ЛВС, охватывающих территорию, на которой размещено одно предприятие или учреждение в одном или нескольких близко расположенных зданиях. Локальные и корпоративные вычислительные сети — основной вид вычислительных сетей, используемых в системах автоматизированного проектирования (САПР).

4.4.3 По принадлежности

В зависимости от прав собственности сети могут быть сетями общего пользования (public) или частными (private). Среди сетей общего пользования выделяют телефонные сети ТфОП (PSTN — Public Switched Telephone Network) и сети передачи данных (PSDN — Private Switched Data Network).

4.4.4 По способу управления

По способу управления сети подразделяются на два типа: *одноранговые* и *на основе сервера*. Между этими двумя типами сетей существуют принципиальные

различия, которые определяют их разные возможности. Выбор типа сети зависит от многих факторов: размера предприятия и вида его деятельности, необходимого уровня безопасности, доступности административной поддержки, объема сетевого трафика, потребностей сетевых пользователей, финансовых возможностей.

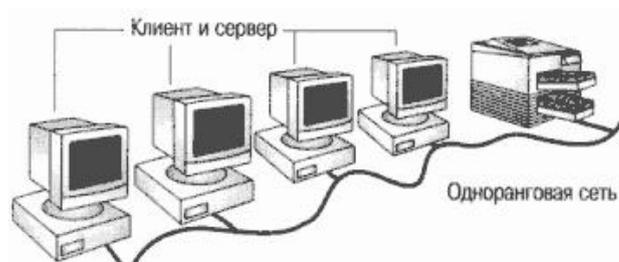


Рис. 4.3 – Одноранговая сеть

В *одноранговой сети* все компьютеры равноправны. Каждый компьютер функционирует и как клиент, и как сервер. Нет отдельного компьютера, ответственного за администрирование всей сети. Пользователи сами решают, какие ресурсы на своем компьютере сделать доступными в сети. Одноранговые сети, как правило, объединяют не более 10 компьютеров. Отсюда их другое название — *рабочие группы*. Одноранговые сети относительно просты, дешевле сетей на основе сервера, но требуют более мощных компьютеров. Требования к производительности и уровню защиты сетевого программного обеспечения (ПО) ниже, чем в сетях с выделенным сервером. Поддержка одноранговых сетей встроена во многие операционные системы (ОС), поэтому для организации одноранговой сети дополнительного ПО не требуется.

Если в сети более 10 компьютеров, то одноранговая сеть становится недостаточно производительной. Поэтому большинство сетей имеют другую конфигурацию — они работают на основе *выделенного сервера*.

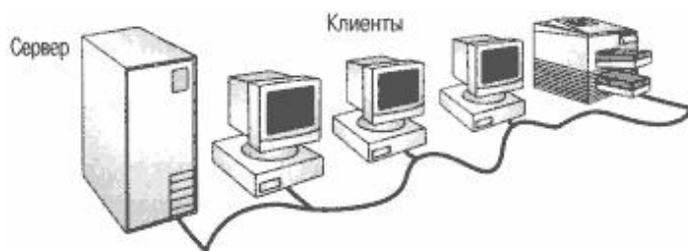


Рис. 4.4 – Сеть на основе сервера

Выделенным сервером называется такой компьютер, который функционирует только как сервер и не используется в качестве клиента или рабочей станции. Он специально оптимизирован для быстрой обработки запросов от сетевых клиентов и обеспечивает защиту файлов и каталогов. При этом клиентские запросы бывают самыми разными, начиная с простейшей проверки имени и пароля пользователя при входе в систему и заканчивая сложными поисковыми запросами к базам данных, на обработку которых даже современный многопроцессорный компьютер

может потратить несколько часов. *Сети на основе сервера* стали промышленным стандартом. Основным аргументом при выборе сети на основе сервера является защита данных. Проблемами безопасности занимается один администратор: он формирует единую политику безопасности и применяет ее в отношении каждого пользователя сети. Сети на основе сервера, в отличие от одноранговых сетей, способны поддерживать тысячи пользователей. При этом к характеристикам компьютеров и квалификации пользователей предъявляются более мягкие требования, чем в одноранговых сетях.

Существуют и *комбинированные типы сетей*, совмещающие лучшие качества одноранговых сетей и сетей на основе сервера. Операционные системы для сетей на основе сервера, например Microsoft Windows NT/2000/2003 Server, в этом случае отвечают за совместное использование основных приложений и данных. На компьютерах-клиентах могут выполняться любые операционные системы Microsoft Windows, которые будут управлять доступом к ресурсам выделенного сервера и в то же время предоставлять в совместное использование свои жесткие диски, а по мере необходимости разрешать доступ и к своим данным. Комбинированные сети — распространенный тип сетей, но для их правильной реализации и надежной защиты необходимы определенные знания и навыки планирования.

4.4.5 По топологии (способу организации связей)

Термин «*топология сети*» характеризует способ организации физических связей компьютеров и других сетевых компонентов. Выбор той или иной топологии влияет на состав необходимого сетевого оборудования, возможности расширения сети и способ управления сетью. Все сети строятся на основе базовых топологий: «шина», «звезда», «кольцо», «ячеистая». Сами по себе базовые топологии не сложны, однако на практике часто встречаются довольно сложные их комбинации.

Шина. Эту топологию (рис. 4.5) часто называют линейной шиной. Она наиболее простая из всех топологий и весьма распространенная. В ней используется один кабель, называемый магистралью или сегментом, вдоль которого подключены все компьютеры.

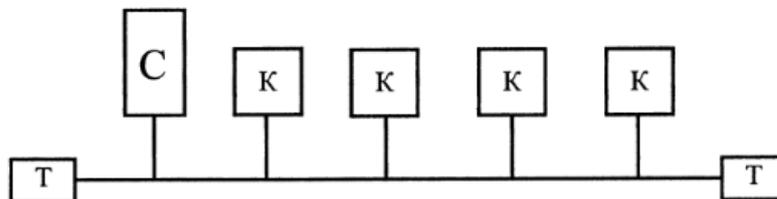


Рис. 4.5 – Топология «Шина»:
С — сервер, К — компьютер, Т — терминатор

В сети с топологией «шина» данные в виде электрических сигналов передаются всем компьютерам сети, но принимает их тот, адрес которого совпадает с адресом получателя, зашифрованным в этих сигналах. Причем в каждый момент времени передачу может вести только один компьютер. Поэтому производительность такой сети зависит от количества компьютеров, подключенных к шине. Чем

больше компьютеров, ожидающих передачи данных, тем медленнее сеть. На быстроедействие сети также влияют:

- тип аппаратного обеспечения сетевых компьютеров;
- частота, с которой компьютеры передают данные;
- тип работающих сетевых приложений;
- тип сетевого кабеля;
- расстояние между компьютерами в сети.

Шина — пассивная топология: компьютеры только слушают передаваемые по сети данные, но не перемещают их от отправителя к получателю. Поэтому выход одного или нескольких компьютеров из строя никак не сказывается на работе сети. Электрические сигналы распространяются по всему кабелю — от одного конца к другому. Сигналы, достигшие концов кабеля, отражаются от них. Возникает наложение сигналов, находящихся в разных фазах, и, как следствие, их искажение и ослабление. Поэтому сигналы, достигшие конца кабеля, следует погасить. Для гашения сигналов на концах кабеля устанавливают терминаторы. При разрыве кабеля или отсутствии терминаторов функционирование сети прекращается. Сеть падает.

Звезда. При топологии «звезда» (рис. 4.6) все компьютеры с помощью сегментов кабеля подключаются к центральному устройству, называемому *концентратором* (*hub*). Сигналы от передающего компьютера поступают через концентратор ко всем остальным.

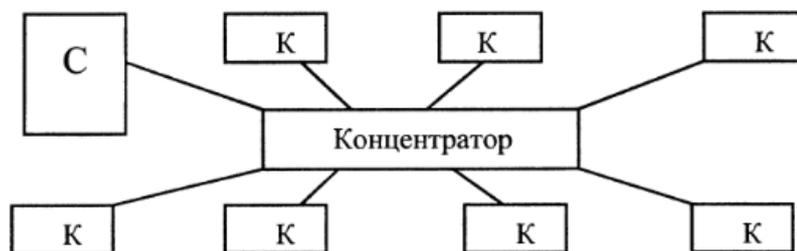


Рис. 4.6 – Топология «Звезда»

В настоящее время концентратор стал одним из стандартных компонентов сетей. В сетях с топологией «звезда» он, например, служит центральным узлом. Концентраторы делятся на активные и пассивные. Активные регенерируют и передают сигналы так же, как репитеры. Их называют многопортовыми повторителями. Обычно они имеют от 8 до 12 портов для подключения компьютеров. Активные концентраторы следует подключать к электрической сети. К пассивным концентраторам относятся монтажные или коммутирующие панели. Они просто пропускают через себя сигнал, не усиливая и не восстанавливая его. Пассивные концентраторы не надо подключать к электрической сети. Недостатки этой топологии: дополнительный расход кабеля, установка концентратора. Главное преимущество этой топологии перед шиной — более высокая надежность. Выход из строя одного или нескольких компьютеров на работу сети не влияет. Любые неприятности с кабелем касаются лишь того компьютера, к которому этот кабель присоединен, и только неисправность концентратора приводит к падению сети. Кроме того, concentra-

тор может играть роль интеллектуального фильтра информации, поступающей от узлов в сеть, и при необходимости блокировать запрещенные администратором передачи.

Кольцо. Компьютеры подключаются к кабелю, замкнутому в кольцо (рис. 4.7). Сигналы передаются по кольцу в одном направлении и проходят через каждый компьютер. В отличие от пассивной топологии «шина», здесь каждый компьютер выступает в роли репитера (повторителя), усиливая сигналы и передавая их следующему компьютеру. Поэтому выход из строя хотя бы одного компьютера приводит к падению сети.

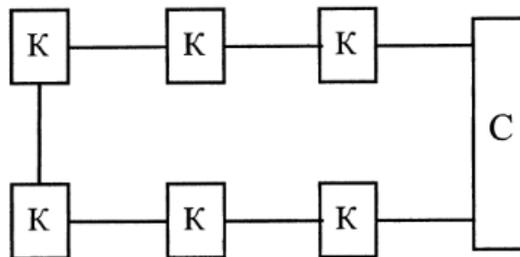


Рис. 4.7 – Топология «Кольцо»

Способ передачи данных по кольцу называется *передачей маркера*. Маркер (token) — это специальная последовательность бит, передающаяся по сети. В каждой сети существует только один маркер. Маркер передается по кольцу последовательно от одного компьютера к другому до тех пор, пока его не захватит тот компьютер, который хочет передать данные. Передающий компьютер добавляет к маркеру данные и адрес получателя, и отправляет его дальше по кольцу. Данные проходят через каждый компьютер, пока не окажутся у того, чей адрес совпадает с адресом получателя. Затем принимающий компьютер посылает передающему сообщение, в котором подтверждает факт приема. Получив подтверждение, передающий компьютер восстанавливает маркер и возвращает его в сеть.

Ячеистая топология. Сеть с ячеистой топологией обладает высокой избыточностью и надежностью, так как каждый компьютер в такой сети соединен с каждым другим отдельным кабелем (рис. 4.8).

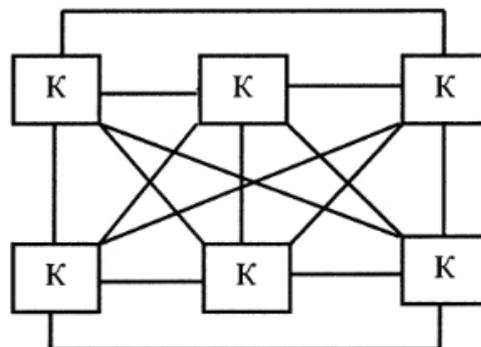


Рис. 4.8 – Ячеистая топология

Сигнал от компьютера-отправителя до компьютера-получателя может проходить по разным маршрутам, поэтому разрыв кабеля не сказывается на работоспособности сети. Основным недостатком — большие затраты на прокладку кабеля, что компенсируется высокой надежностью и простотой обслуживания. Ячеистая топология применяется в комбинации с другими топологиями при построении больших сетей.

Кроме базовых топологий существуют их комбинации — **комбинированные топологии**. Чаще всего используются две комбинированные топологии: «звезда-шина» и «звезда-кольцо». **«Звезда-шина»** — несколько сетей с топологией «звезда» объединяются при помощи магистральной линейной шины (к концентратору подключены компьютеры, а сами концентраторы соединены шиной). Выход из строя одного компьютера не сказывается на работе всей сети, а сбой в работе концентратора влечет за собой отсоединение от сети только подключенных к нему компьютеров и концентраторов. **«Звезда-кольцо»** — отличие состоит только в том, что концентраторы в «звезде-шине» соединяются магистральной линейной шиной, а в «звезде-кольце» концентраторы подсоединены к главному концентратору, внутри которого физически реализовано «кольцо».

4.4.6 По типу среды передачи

По типу среды передачи можно выделить сети, построенные на базе коаксиального кабеля, витой пары, оптоволокна.

Коаксиальный кабель до недавнего времени был самым распространенным. Недорогой, легкий, гибкий, удобный, безопасный и простой в установке. Существует два типа коаксиальных кабелей: тонкий (спецификация 10Base2) и толстый (спецификация 10Base5). Тонкий — гибкий, диаметр 0,64 см (0,25"). Прост в применении и подходит практически для любого типа сети. Подключается непосредственно к плате сетевого адаптера. Передает сигнал на 185 м практически без затухания. Волновое сопротивление — 50 ом. Толстый — жесткий, диаметр 1,27 см (0,5"). Его иногда называют *стандартный Ethernet* (первый кабель в популярной сетевой архитектуре). Жила толще, затухание меньше. Передает сигнал без затухания на 500 м. Используют в качестве магистрали, соединяющей несколько небольших сетей. Волновое сопротивление — 75 ом. Для подключения к толстому коаксиальному кабелю применяется специальное устройство — трансивер (transceiver — приемопередатчик). Он снабжен коннектором, который называется «вампир» или «пронзающий ответвитель». К сетевой плате трансивер подключается с помощью кабеля с разъемом. Для подключения тонкого коаксиального кабеля используются BNC-коннекторы (British Naval Connector). Применяются BNC-T-коннекторы для соединения сетевого кабеля с сетевой платой компьютера, BNC — баррел-коннекторы для сращивания двух отрезков кабеля, BNC-терминаторы для поглощения сигналов на обоих концах кабеля в сетях с топологией «шина».

Витая пара — это два перевитых изолированных медных провода. Несколько витых пар проводов часто помещают в одну защитную оболочку. Переплетение проводов позволяет избавиться от электрических помех, наводимых соседними проводами и другими внешними источниками, например двигателями, трансформаторами, мощными реле. Существует неэкранированная и экранированная витая пара. *Неэкранированная витая пара* (UTP) широко используется в ЛВС, макси-

мальная длина 100 м. UTP определена особым стандартом, в котором указаны нормативные характеристики кабелей для различных применений, что гарантирует единообразие продукции. *Экранированная витая пара (STP)* помещена в медную оплетку. Кроме того, пары проводов обмотаны фольгой. Поэтому STP меньше подвержена влиянию электрических помех и может передавать сигналы с более высокой скоростью и на большие расстояния. Преимущества витой пары — дешевизна, простота при подключении. Недостатки — нельзя использовать при передаче данных на большие расстояния с высокой скоростью.

В оптоволоконном кабеле цифровые данные распространяются по оптическим волокнам в виде модулированных световых импульсов. Это надежный способ передачи, так как электрические сигналы при этом не передаются. Следовательно, оптоволоконный кабель нельзя вскрыть и перехватить данные. Оптоволоконные линии предназначены для перемещения больших объемов данных на очень высоких скоростях, так как сигнал в них практически не затухает и не искажается. Оптоволокно передает сигналы только в одном направлении, поэтому кабель состоит из двух волокон с отдельными коннекторами: одно — для передачи, другое — для приема. Скорость передачи данных в настоящее время составляет от 100 Мбит/с. Между тем, получает все большее распространение скорость 1 Гбит/с, теоретически — до 200 Гбит/с. Расстояние — многие километры. Кабель не подвержен электрическим помехам. Существенным недостатком этой технологии является дороговизна и сложность в установке и подключении. Типичная оптическая сеть состоит из лазерного передатчика света, мультиплексора/демультиплексора для объединения оптических сигналов с разными длинами волн, усилителей оптических сигналов, демультиплексоров и приемников, преобразующих оптический сигнал обратно в электрический. Все эти компоненты обычно собираются вручную.

4.4.7 По скорости передачи

По скорости передачи информации сети подразделяют на низкоскоростные (до 10 Мбит/с), среднескоростные (до 100 Мбит/с) и высокоскоростные (свыше 100 Мбит/с).

Также существуют объединения двух и более сетей. Хорошо известным примером такого объединения является *Internet*. Размеры сетей являются важным классификационным фактором, поскольку в сетях различного размера применяется различная техника.

4.5 Глобальная сеть Интернет

4.5.1 Историческая справка

Прообраз сети Интернет начал создаваться в конце 60-х годов XX века. В 1969 году агентство перспективных исследований министерства обороны США начало работу над проектом связи компьютеров оборонных организаций. В результате выполнения данных проектов была создана сеть, которая явилась предтечей сети Internet и была названа ARPANET. При создании данной сети преследовалось несколько целей. Главная — это для поддержки научных исследований в военно-

промышленной сфере, в частности для исследования методов построения сетей, устойчивых к частичным повреждениям, получаемым, например, при бомбардировке авиацией и способных в таких условиях продолжать нормальное функционирование. Это требование дает ключ к пониманию принципов построения и структуры Internet. В модели ARPANET всегда была связь между компьютером-источником и компьютером-приемником (станцией назначения). С самого начала предполагалось, что связь в сети является ненадежной: любой ее сегмент может быть поврежден или уничтожен, но сеть должна была обеспечивать связь между уцелевшими компьютерами.

В 1974 г. была поставлена задача разработки универсального протокола передачи данных, которая была решена созданием протокола передачи данных и объединения сетей — Transmission Control Protocol/Internet Protocol (TCP/IP). В 1983 г. был осуществлен перевод ARPANet на TCP/IP.

Следующим этапом в развитии Интернета следует считать новую сеть NSFNET, разработанную по инициативе Национального Научного Фонда (National Science Foundation — NSF). В конце 80-х NSF создал пять суперкомпьютерных центров, сделав их доступными для использования в любых научных учреждениях. Было создано всего лишь пять центров, потому что они были очень дороги даже для богатой Америки. Именно поэтому их и следовало использовать кооперативно. Возникла проблема связи: требовался способ соединить эти центры и предоставить доступ к ним различным пользователям. Сначала была сделана попытка использовать коммуникации ARPANET, но это решение потерпело крах, столкнувшись с бюрократией оборонной отрасли и проблемой обеспечения персоналом.

Тогда NSF решил построить свою собственную сеть, основанную на IP-технологии ARPANET. Центры были соединены специальными телефонными линиями с пропускной способностью 56 Kbps. Однако было очевидно, что не стоит даже и пытаться соединить все университеты и исследовательские организации непосредственно с центрами, т. к. проложить такое количество кабеля — не только очень дорого, но практически невозможно. Поэтому решено было создавать сети по региональному принципу. В каждой части страны заинтересованные учреждения должны были соединиться со своими ближайшими соседями. Получившиеся цепочки подсоединялись к суперкомпьютеру в одной из своих точек, таким образом, суперкомпьютерные центры были соединены вместе. В такой топологии любой компьютер мог связаться с любым другим, передавая сообщения через соседей. Это решение было успешным, но настала пора, когда сеть уже более не справлялась с возросшими потребностями. Совместное использование суперкомпьютеров позволяло подключенным общинам использовать и множество других вещей, не относящихся к суперкомпьютерам. Неожиданно университеты, школы и другие организации осознали, что заимели под рукой море данных и мир пользователей. Поток сообщений в сети (трафик) нарастал все быстрее и быстрее, пока, в конце концов, не перегрузил управляющие сетью компьютеры и связывающие их телефонные линии. В 1987 г. контракт на управление и развитие сети был передан компании Merit Network Inc., которая занималась образовательной сетью Мичигана совместно с IBM и MCI. Старая физически сеть была заменена более быстрыми (примерно в 20 раз) телефонными линиями. Были заменены на более быстрые и сетевые управляющие машины. Одновременно с созданием центров в разных

частях США стали создаваться национальные сети в других странах. Затем они стали объединяться, в итоге в 90-х годах XX века сформировалась сеть Интернет в сегодняшнем виде.

Одним из наиболее важных событий в истории Интернета стала разработка так называемой всемирной паутины — среды — World Wide Web (WWW). История WWW началась в марте 1989 г, когда в Европейской лаборатории физики элементарных частиц (CERN, Швейцария, Женева) Тим Бернерс-Ли разработал технологию гипертекстовых документов. Суть состоит в том, что документ, к которому будет возможен доступ через Интернет, определенным образом форматируется с помощью гипертекстового языка. Информация может быть найдена в сети посредством так называемого универсального локатора ресурса (URL) и отображена с помощью навигационных программ-браузеров. Одной из первых наиболее удачных таких программ была программа Mosaic. В настоящее время таких программ множество (Internet Explorer, Mozilla Firefox, Opera). Появление WWW и программ-браузеров позволило работать в Интернете не только программистам, но и новичкам, которые имеют лишь начальные сведения о работе с оболочками типа Windows.

Сейчас Интернет объединяет тысячи сетей по всему миру. В России интернет появился относительно недавно. Бурный рост числа пользователей Интернета в России начался в 1996 году. Процесс совершенствования сети Интернет идет непрерывно, но большинство этих перестроек происходит незаметно для пользователей. Включив компьютер, вы не увидите объявления о том, что ближайшие полгода Интернет не будет доступен из-за модернизации. Наряду с ростом числа пользователей сети Интернет в России значительные изменения происходят в ее русской части. Если несколько лет назад вся информация в сети была представлена на английском языке, то сегодня можно найти самые разнообразные сведения на русском языке.

4.5.2 Способы доступа в Интернет.

Dial-Up-технология — это доступ по коммутируемой телефонной линии с помощью модема. Один из первых способов доступа в Интернет в России. Для этого достаточно иметь городской телефонный номер и модем. Скорость доступа при таком способе подключения не более 56 Кбит/с, а скорость передачи информации будет несколько ниже — около 33 кбит/с, что является существенным недостатком. Такая скорость сегодня мало кого устраивает. Еще один недостаток такого соединения заключается в том, что пока вы находитесь в сети Интернет, ваш телефон будет занят (до вас не смогут дозвониться). Какие же альтернативные технологии позволяют получить более высокую скорость доступа в Интернет?

Обычный телефон использует лишь низкочастотный диапазон линии. Однако провод телефонной линии способен передавать гораздо больше данных, если использовать более широкую полосу (полоса пропускания обычной телефонной линии 3400 кГц). Поэтому телефонную сеть, которая изначально предназначалась для передачи голосового сигнала, приспособили для высокоскоростной передачи цифровых данных.

DSL-технология (Digital Subscriber Line — цифровая абонентская линия) позволяет использовать более широкую полосу пропускания для передачи данных без

ущерба для использования телефонной линии по прямому назначению. Существует целое семейство технологий под общим названием xDSL, где приставка x указывает на конкретную спецификацию семейства DSL. Эта технология позволяет одновременно работать в Интернете и разговаривать по телефону. Скорость подключения по ней намного выше, чем при помощи обычного модема. DSL не требует прокладки новых проводов, так как использует уже имеющуюся телефонную линию. Одним из основных преимуществ технологии xDSL является высокоскоростной доступ в Интернет. При работе в Интернет основной поток информации идет из Сети к пользователю, а в Сеть передается гораздо меньший объем данных. Действительно, при просмотре Web-страниц в ответ на небольшой запрос пользователь получает из сети не только текст, но и изображения.

ADSL-технология (Asymmetrical DSL) позволяет передавать данные пользователю со скоростью, на порядок превышающей скорость передачи данных от пользователя. При этом сигнал от пользователя в Сеть передается на более низких частотах, чем сигнал из Сети к пользователю. Теоретически при этом можно иметь канал с пропускной способностью 1 Мбит/с в прямом направлении (в Сеть) и 8 Мбит/с — в обратном. При этом одна и та же линия может использоваться для передачи голоса и цифровых данных. По сравнению с коммутируемым доступом ADSL-линия работает, как минимум, на два порядка быстрее. Высокая скорость позволяет комфортно работать с Web-сайтами с мультимедийной информацией, быстро перекачивать большие файлы и полноценно использовать интерактивные приложения. В среднем скорость передачи данных с помощью данной технологии составляет от 128 кбит/сек до 24 мбит/сек.

Достоинства ADSL: легкость установки (используется уже имеющаяся телефонная линия), постоянный доступ в Интернет (пользователи ADSL не разделяют полосу пропускания с другими абонентами).

Недостаток ADSL: ограничения по дальности. Скорость передачи потока данных в обратном направлении существенно зависит от расстояния. Если при расстоянии 3 км можно получить скорость около 8 Мбит/с, то на расстоянии 5 км — только 1,5 Мбит/с.

Мобильный интернет — GPRS/EDGE/3G. В принципе это три разных поколения стандартов Интернет-соединения, но их объединяет одно, все они беспроводные и предоставляются сотовыми операторами. И для подключения к такому Интернету вам нужно иметь мобильный телефон с поддержкой GPRS/EDGE/3G (или всего вместе) и подключить услугу на своем тарифном плане. Сотовый телефон в таком случае можно подключить через дата-кабель (обычно входит в комплект телефона) либо через BlueTooth. Преимущество лежит на поверхности. Во-первых, не зависите ни от каких договоров (для подключения услуги GPRS-Интернет обычно достаточно поговорить с оператором службы поддержки по телефону). Во-вторых, это очень интересный вариант для тех, кто не имеет домашнего телефона. В-третьих, если у вас есть ноутбук, то вы можете пользоваться Интернетом по всей зоне покрытия вашего оператора. Теперь ложечка дегтя в эту бочку меда: уровень цен на услуги Интернета сотовых операторов несколько выше, чем услуги провайдеров проводного телефона, и порой для пользователя данный способ является невыгодным. К тому же скорость соединения GPRS хоть и выше по стандарту, порой она опускается до уровня dial-up соединения, это связано с тем, что

большинство операторов устанавливают приоритет голосовым вызовам и линия перегружается, но с приходом 3G эта проблема будет решена.

Что же такое 3G? **3G** — это мобильная связь 3-го поколения. Представляет собой общепринятое обозначение нескольких стандартов мобильной связи, имеющих ряд отличительных признаков. В частности, мобильные сети 3-го поколения обеспечивают возможность одновременной передачи голосовых (обычный телефонный звонок) и не голосовых данных (загрузка файлов, обмен электронной почтой). Немаловажная роль в 3G-услугах отведена скоростному Интернет-доступу, видеотелефонии и мобильному ТВ. Правда, последние два сервиса пока не достигли ожидаемой популярности.

Спутниковый интернет. Отличается большой стоимостью оборудования и высокой скоростью передачи данных, сравнимой со скоростью в выделенном канале. Оптимальный способ организовать работу в интернете на высоких скоростях и при больших объемах входящего трафика вдали от населенных пунктов. К этому надо добавить проблему организации запросного (так называемого «земного») канала по одному из вышеуказанных вариантов связи. В этом случае стоимость входящего трафика получается очень низкой. Это делает данный вид связи экономически более выгодным при больших объемах информации. Данный вид выхода в сеть в основном используют коммерческие организации. Но спутниковый интернет также доступен и для индивидуальных пользователей, однако их число не так велико по сравнению с тем же ADSL.

Выделенный канал Интернет. В просторечии «выделенка». Это один из возможных способов подключения к сети Интернет. На сегодняшний день получает все большее распространение среди пользователей. Доступ в Интернет по выделенному каналу выгоден тем пользователям, которым необходимо долго находиться в сети. Соединение с сетью может осуществляться при помощи оптоволоконного кабеля или через модем ADSL. Выделенный канал обладает значительным преимуществом по сравнению с соединением через коммутируемую городскую телефонную сеть. Соединение по «выделенке» происходит напрямую с персонального компьютера. Выделенный канал доступа имеет неоспоримые преимущества, например:

- свободная телефонная городская линия;
- быстрый дозвон и соединение с сетью;
- более высокая скорость передачи данных;
- постоянная устойчивая связь (в то время как при использовании телефонной линии вполне может произойти перегрузка сети, снижение скорости загрузки страниц, вплоть до обрыва сигнала).

Выделенные каналы стали появляться, когда службы АТС начали выделение своих линий связи для подключения сетевого оборудования ПК в общие вычислительные сети. Выделенный канал — это проводная линия в одном случае или радиоканал в другом. Связь с сетью осуществляется через два модема: один — у пользователя (без привлечения телефона), другой — у провайдера. При выделенном подключении к сети Интернет с помощью оптоволоконного кабеля необходимо связать отдельный компьютер с модемом провайдера. Для этого нужно только вставить кабель в компьютер или ноутбук. Скорость передачи данных по выделен-

ным сетям достигает до 100 мбит/сек. «Выделенка» по оптоволоконным кабелям — наиболее перспективное направление в развитии технологии связи.

Wi-Fi-интернет. Это современная беспроводная технология подключения компьютера к сети. Услуга незаменима для домашних пользователей, которые имеют ноутбук, КПК, мобильный телефон или другое оборудование, оснащенное WiFi-адаптером, а также хотят связать несколько домашних стационарных либо мобильных компьютеров в сеть и не желают прокладывать десятки метров кабеля между ними. Точка доступа Wi-Fi — это просто устройство, и чтобы получить именно Интернет посредством Wi-Fi, для начала необходимо подключиться к какому-нибудь провайдеру, а потом на это соединение установить беспроводное оборудование (в точке доступа будет установлен радиомодуль, который выполняет функции приема-передачи данных. Такой же модуль устанавливается в компьютер или ноутбук). В этом случае у Вас будет беспроводный доступ в Интернет.

Преимущества доступа в Сеть с использованием данной технологии очевидны:

- нет необходимости постоянно быть на одном месте;
- можно находиться в любой точке помещения, попадающего в зону действия точки доступа, не нужно разводить десятки метров кабеля;
- удобно использовать в публичных местах (гостиницы, библиотеки, кафе, холлы учебных заведений)

4.5.3 Структура и основные принципы работы Интернета

Как и любая другая сеть, Интернет состоит из множества компьютеров, соединенных между собой линиями связи, и установленного на этих компьютерах программного обеспечения. Тип программного обеспечения определяется идеологией, называемой *клиент/сервером*, которая составляет основу всех сервисов Интернета. Каждая операция в сети Интернет состоит из взаимодействия трех элементов: клиента, сервера, сети Интернет, которая передает запрос клиента серверу и ответ сервера клиенту.

Под понятием «клиент» подразумевают программы, с помощью которых пользователь обращается к какому-либо сервису сети Интернет.

Понятие «сервер» имеет несколько значений. Это может быть программа, предоставляющая клиентам различные данные, или компьютер, на котором выполняется эта программа.

Пользователи Интернета подключаются к сети через компьютеры специальных организаций, которые называются *провайдерами*. Компьютеры, подключенные к Интернету, называются *узлами*. Изучение принципов передачи информации в сети связано с такими понятиями, как *протокол* и *адрес*.

Различают два типа протоколов: базовые и прикладные. *Базовые протоколы* отвечают за физическую пересылку сообщений между компьютерами в сети Интернет. Это протоколы IP и TCP. *TCP-протокол* делит длинные сообщения на несколько пакетов, каждый из которых затем помещается в TCP-конверт и передается *IP-протоколу*. К каждому полученному TCP-конверту протокол IP добавляет информацию, по которой можно определить адреса отправителя и получателя. Это аналогично помещению адреса на бумажный почтовый конверт. Для каждого поступающего пакета маршрутизатор, через который проходит пакет, по данным

IP-адреса определяет, кому из ближайших соседей необходимо переслать данный пакет, чтобы он быстрее оказался у получателя, т. е. принимает решение об оптимальном пути следования очередного пакета. В конечном итоге TCP-модуль адресата собирает и распаковывает IP-конверты, затем распаковывает TCP-конверты и помещает данные в нужной последовательности. Если чего-либо не достает, он требует переслать этот пакет снова. Пакеты не только теряются, но и могут искажаться при передаче из-за наличия помех на линиях связи. TCP решает и эту проблему. В конце концов, информация собирается в нужном порядке и полностью восстанавливается. Таким образом, протокол IP осуществляет перемещение данных в сети, а протокол TCP обеспечивает надежную доставку данных, используя систему кодов, исправляющих ошибки. Так как протоколы TCP и IP тесно взаимосвязаны, их часто объединяют, говоря, что в Интернете базовым является **протокол TCP/IP**.

Прикладными называют протоколы более высокого уровня, они отвечают за функционирование специализированных служб. Например, протокол HTTP служит для передачи гипертекстовых сообщений, протокол FTP — для передачи файлов, SMTP — для передачи электронной почты.

Каждому компьютеру, подключенному к Интернету, присваивается идентификационный номер, который называется **IP-адресом**. IP-адрес состоит из разделенных точками четырех чисел, каждое из которых не превышает 256. Например: 193.27.61.137 или 87.104.65.77. Для облегчения запоминания IP-адрес обычно выражают рядом чисел в десятичной системе счисления, разделенных точками. Но компьютеры хранят его в бинарной форме. Например, тот же IP-адрес в двоичном коде будет выглядеть так: 11000001. 00011011. 00111101. 10001001.

На ранней стадии своего развития Интернет состоял из небольшого количества компьютеров, объединенных модемами и телефонными линиями. Тогда пользователи могли установить соединение с компьютером, набрав цифровой адрес, например 163.25.51.132. Это было удобно, пока компьютеров было мало. По мере увеличения их количества цифровые имена стали заменять текстовыми, потому что текстовое имя проще запомнить, чем цифровое. Возникла проблема автоматизации этого процесса, и в 1983 г. в Висконсинском университете США была создана так называемая DNS-система (Domain Name System) — *система имен доменов*, которая автоматически устанавливала соответствие между текстовыми именами и IP-адресами. Вместо чисел была предложена ставшая сегодня для нас привычной запись типа `www.yandex.ru`

В основе системы доменных имен лежит иерархический принцип. В именах домены отделяются друг от друга точками, например: `addressx.msk.ru`, `addressy.spb.ru`, `gorod.tomsk.ru`. В имени может быть различное количество доменов, но обычно их не больше пяти. Так в приведенном последнем примере самым старшим является домен *ru*, ему подчинен домен *tomsk*, который в свою очередь имеет в подчинении домен *gorod*.

Система имен доменов имеет четко выраженный региональный характер. Обычно последним элементом в списке является двухсимвольный код страны. Например: *ru* — Россия, *ua* — Украина, *kz* — Казахстан, *de* — Германия, *jp* — Япония, *au* — Австралия, *be* — Бельгия и т. д. Это так называемые географические домены верхнего уровня.

Помимо географического признака используется организационный признак, в соответствии с которым существуют следующие доменные имена первого уровня:

- com — коммерческие предприятия,
- edu — образовательные учреждения,
- gov — государственные учреждения,
- mil — военные организации,
- net — сетевые образования,
- org — учреждения других организаций и сетевых ресурсов.

Примеры:

www.microsoft.com — домен верхнего уровня com указывает на принадлежность к коммерческой организации, а в качестве имени домена второго уровня (располагается левее домена первого уровня) указано название корпорации Microsoft.

www.ttu.tomsk.ru — домен верхнего уровня ru указывает на то, что адрес принадлежит российской части Интернет, tomsk — определяет город, следующий уровень — домен конкретного предприятия (в данном случае трамвайно-троллейбусное управление).

При работе в Интернете используются не просто доменные имена, а универсальные указатели ресурсов URL (Uniform Resource Locator). *URL-адрес* — это адрес любого ресурса в Интернете с указанием того, с помощью какого протокола к нему следует обращаться. Иными словами, в указателе кроме собственного адреса имеются сведения, какую программу следует запустить на сервере и к какому файлу следует обратиться. Понятие «сервер» и «сайт» очень часто путают. Вообще-то сайт считается более мелким элементом, и на одном сервере могут «проживать» тысячи независимых сайтов. Адреса сайтов состоят из нескольких важных элементов.

Например:

http:// — это протокол передачи гипертекстовых документов (Hypertext Transfer Protocol), по которому обеспечивается доставка документа с Web-сервера, указывает браузеру, что для доступа к ресурсу применяется данный сетевой протокол. Вообще-то чаще всего он нужен только для проформы — при наборе адреса его чаще всего опускают, и адрес приобретает более компактный вид, например: www.mail.ru

ftp:// — указатель на доступ через FTP (File Transfer Protocol — протокол передачи файлов);

file:// указатель на файл.

В общем виде URL записывается так: протокол://адрес сервера/путь/имя файла
Рассмотрим, например такой адрес: http://www.tusur.ru/ru/faculties

Здесь: http — как уже было сказано выше, это протокол, который служит для работы с гипертекстовыми документами; www.tusur.ru — адрес сервера, где www — узел, на котором размещен сайт, tusur и ru — имена доменов второго и третьего уровня; ru — каталог (путь) к искомому файлу; faculties — имя файла.

4.5.4 Сервисы сети Интернет

Обычно пользователи идентифицируют Интернет со службой WWW (*World Wide Web* — *Всемирная паутина*). Но это далеко не так, ибо WWW — одна из мно-

гочисленных служб Интернета, это только часть услуг, которыми располагает Интернет.

WWW построена на технологии, в основу которой положен гипертекст, то есть текст со ссылками. В настоящее время можно говорить не о гипертекстовой, а о гипермедийной среде, т.е. сеть перестала быть просто текстовой, в ней появилось огромное количество графики, музыки и т.д. Сайты строятся по новой технологии, например Flash, т.е. не содержат текста как такового. Перемещение от сайта к сайту осуществляется с помощью гиперссылок. Как было сказано выше, для адресации в WWW используется URL. Помимо WWW, с помощью Интернета можно воспользоваться электронной почтой, FTP-сервисом, телеконференциями и другими услугами.

Электронная почта. История развития электронной почты берет начало с 1965 года. Именно тогда появилась программа под названием MAIL, которая работала в операционной системе CTSS. Авторами программы стали Том Ван Влек и Ноэль Моррис. 26 марта 1976 года было отправлено первое электронное письмо Елизаветы II — королевы Англии. Практически каждый пользователь сегодня пользуется «смайлами». Но далеко не все знают, что днем рождения смайлика следует считать 12 апреля 1979 года. 1994 год стал годом рождения рекламных рассылок по электронной почте. Уже позднее подобные «засоряющие» почтовые ящики материалы окрестят «спамом» и будут придумывать способы блокировать их. Электронное письмо, как и обычное, содержит адреса отправителя и получателя. В него можно вложить графическое изображение или иной файл — точно так же, как в конверт с письмом можно положить открытку или фотографию. На него можно поставить электронную подпись, которая играет ту же роль, что и подпись в обычном письме. Однако служба e-mail давно обошла по популярности традиционную почту: ежегодно в мире рассылается более 600 миллиардов электронных писем. Чем же вызвана такая популярность? Для ответа на этот вопрос перечислим достоинства электронной почты.

В отличие от телефонного звонка электронная почта может быть прочитана в удобное время, что особенно важно с учетом разницы во времени между часовыми поясами. Следует также отметить демократичность электронной почты: отправляя электронное письмо хоть самому президенту, вы не рискуете отвлечь его от текущих дел. К удобствам электронной почты следует также отнести возможность рассылки писем сразу большому количеству получателей, высокую скорость доставки, удобство пересылки вложенных файлов. Хранение писем в базе данных почтового клиента позволяет осуществлять быстрый поиск и сортировку почтовых отправлений. Адрес электронной почты имеет формат: имя_пользователя@имя_домена, например: Ivanov@mail.ru. Часть слева от значка @ — это имя почтового ящика на сервере, из которого владелец этого адреса забирает письма (в данном примере — Ivanov). Часть справа от значка @ называется доменом и указывает на местонахождение этого почтового ящика. Сам значок @ называется «собака». Электронный адрес индивидуален и неповторим. Нужно отметить, что носителем адреса электронной почты является вовсе не конечный пункт доставки, т.е. не адрес вашего домашнего компьютера, а адрес сервера, на котором вы будете получать почту. Электронная почта построена по принципу клиент-серверной архитектуры. Пользователь общается с клиентской программой,

которая, в свою очередь, общается с почтовым сервером. Очевидно, что процедуры отправки и получения почты требуют разной степени идентификации личности, поэтому существуют и два разных протокола — на отправку и на прием писем. Для передачи писем используются протокол SMTP (Simple Mail Transfer Protocol — простой протокол пересылки почты) и соответственно SMTP-серверы. Чаще всего отправка почты происходит с почтового сервера вашего провайдера, хотя, в принципе, это не обязательно. Обычно SMTP-серверы не требуют идентификации, поэтому вы можете отправить письмо с любого такого сервера. Для приема почтовых сообщений в настоящее время наиболее часто используется протокол POP3 (Post Office Protocol — протокол почтового офиса), который контролирует право пользователя забирать почту из ящика и поэтому требует предоставления имени пользователя и пароля.

FTP-сервис — (File Transfer Protocol — протокол передачи файлов) — позволяет получать и передавать файлы с одного компьютера на другой с использованием TCP-соединений. Для чего он нужен? Проблема обмена данными, когда нужно оперативно передать коллегам или друзьям какие-то объемные материалы, знакома многим пользователям. Скажем, вам регулярно необходимо предоставлять рабочие материалы (презентации, изображения, видеоролики) другим сотрудникам для обсуждения. Материалы очень объемные, по почте их не отправишь. Как поступить в таком случае? Вариантов множество - можно воспользоваться услугами фото- или видеохостинговых сервисов, разместить файлы в онлайн-хранилище либо обратиться к файлообменным сервисам. Но есть и другой вариант — создать свой FTP-сервер, который позволит сделать обмен данными более быстрым, безопасным и удобным. С помощью FTP-сервера можно будет не только открывать доступ к определенным папкам на собственном компьютере, но и гибко управлять объемами трафика, а также списками доступных файлов и пользователей. При этом пользователям, для которых предназначены материалы, также станет гораздо удобнее их получить, ведь они смогут использовать для скачивания файлов привычные FTP-клиенты, обеспечивающие «докачку» и умеющие проводить скачивание в несколько потоков.

ICQ. Это специальная программа, называемая в просторечии «аська». При ее установке вам присваивается уникальный идентификационный номер, которым вы можете делиться со своими друзьями и знакомыми. Стоит вам войти в Сеть, запустить данную программу, ICQ посылает на свой сервер сигнал «объект номер такой-то вошел в сеть». В тот же самый момент сервер пересылает этот сигнал вашим знакомым из вашего «контакт-листа». Работая в ICQ, чаще всего пользователи обмениваются текстовыми сообщениями. Поэтому неудивительно, что «аську» многие считают «Интернет-пейджером». Но ICQ — это больше, чем пейджер. Скорее — универсальный коммуникатор. Работая в ICQ, вы можете отправить через ее собственный сервер какой-либо файл (документ, фотографию, песню и т.д.) любому человеку из вашего «контакт-листа». Через какое-то время после появления ICQ стали появляться похожие программы, например QIP. Это программа с поддержкой ICQ-номеров, обеспечивающая возможность обмена сообщениями с пользователями ICQ. Благодаря расширенным встроенным возможностям поиска, программу можно использовать разными способами. Данное приложение абсолютно бесплатно (впрочем, как и ICQ, и остальные похожие программы), для ее запуска достаточно иметь учетную запись ICQ.

Skype. Последнее время данный сервис набирает все большую популярность. Skype — это программа, позволяющая общаться через сеть Интернет со своими коллегами, друзьями, родственниками по всему миру. Программа разработана компанией Skype Limited. Программа позволяет:

- Вести индивидуальную переписку, обмениваться сообщениями с коллегами по работе, так же как и в других программах вида ICQ, QIP или Jabber.
- Если у вас есть микрофон и наушники, вы можете дозвониться до друга из соседнего города и просто разговаривать.
- Если у вас есть веб-камера, микрофон и наушники, то вы можете видеть собеседника, слышать его и отвечать. В данном случае информация передается через сеть Интернет, и вы платите не за минуты, а за количество отправленных и полученных мегабайт.

Программа Skype также позволяет звонить вашему партнеру на мобильный и стационарный телефон, отправлять смс-сообщения. При этом вы не встаете из-за компьютера, выбираете выгодный для вас тариф и платите меньше. С помощью Skype вы можете дозвониться в любую точку планеты, при этом звонки со Skype на Skype будут совершенно бесплатны (платите только за трафик). Отличие Skype от других программ (ICQ, QIP, Jabber) состоит в том, что вы можете разговаривать в чате как с одним человеком, так и сразу с несколькими десятками людей, тех, кого вы пригласите в свой чат. Так же как и в других программах, Skype позволяет пересылать файлы, вести записную книгу, получать новости, заходить на другие конференции, тема которых вам интересна. Разработчики программы Skype позаботились о большинстве пользователей, программа работает на самых известных платформах: Windows, Linux, Mac OS X, Pocket PC.

Электронные покупки. Первые примеры электронной торговли корнями уходят в банковские операции и процесс, получивший название «телеграфные переводы». В этом процессе покупатель уполномочивал банк сделать телеграфный перевод на счет в другом банке. Такой процесс включал в себя перемещение денежных средств через страну или на другой край света и обычно использовался крупными предприятиями и другими самостоятельными коммерческими организациями. На смену телеграфным переводам пришли электронные средства передачи финансовой информации — процесс, в ходе которого один банк посылал телекс другому, поручая ему перевести денежные средства. При этом банки уведомляли друг друга о переводе телеграммами, телексами или по электронной почте.

В настоящее время существует множество *электронных платежных систем*, которые обеспечивают проведение финансовых расчетов между участниками системы (лицами, которые открыли свой счет в системе) в режиме реального времени. Примеры самых популярных платежных систем: WebMoney, Яндекс.Деньги. Также существуют множество платежных систем в регионах (например, томская региональная платежная система RegPlat). Зарегистрировавшись в данных системах, вы получаете собственный электронный кошелек, пополнять который можно, например, с помощью банковских карт международных платежных систем, таких как Visa, MasterCard, либо через терминал. После того, как в электронном кошельке появятся деньги, вы можете совершать электронные покупки. Через сеть Интернет можно осуществить различные покупки: авиа- и железнодорожные билеты, лекарства, продукты, бытовую технику и так далее.

4.6 Выводы

Компьютеры — важная часть сегодняшнего мира, а компьютерные сети серьезно облегчают нашу жизнь, ускоряя работу и делая отдых более интересным. Практически сразу после появления ЭВМ возник вопрос о налаживании взаимодействия компьютеров друг с другом, чтобы более эффективно обрабатывать информацию, использовать программные и аппаратные ресурсы. Появились и первые сети, в то время объединявшие только большие ЭВМ в крупных компьютерных центрах. Однако настоящий «сетевой бум» начался после появления персональных компьютеров, быстро ставших доступными широкому кругу пользователей — сначала на работе, а затем и дома. Компьютеры стали объединять в локальные сети, а локальные сети соединять друг с другом, подключать к региональным и глобальным сетям. В результате сотни миллионов компьютеров в мире были объединены в сети, и более миллиарда пользователей получили возможность взаимодействовать друг с другом. Сегодня можно с уверенностью сказать, что компьютерные сети стали неотъемлемой частью нашей жизни, а область их применения охватывает буквально все сферы человеческой деятельности. Без компьютерных сетей человечеству уже никак не обойтись.



Контрольные вопросы по главе 4

- 1) Для каких целей используются компьютерные сети?
- 2) Перечислить признаки классификации сетей.
- 3) Перечислить способы доступа в Интернет.
- 4) Какие преимущества имеет выделенный канал?
- 5) В чем заключается суть Dial-Up-технологии доступа в Интернет?

Глава 5

БАЗЫ ДАННЫХ

Базы данных всегда были важнейшей темой при изучении информационных систем. Однако в последние годы всплеск популярности Интернета и бурное развитие новых технологий для Интернета сделали знание технологии баз данных для многих одним из актуальнейших путей карьеры. Технологии баз данных увеличили Интернет-приложения далеко от простых брошюрных публикаций, которые характеризовали ранние приложения. В то же время Интернет-технология обеспечивает пользователям стандартизированные и доступные средства публикации содержимого баз данных. Правда, ни одна из этих новых разработок не отменяет необходимости в классических приложениях баз данных, которые появились еще до развития Интернета для нужд бизнеса. Это только расширяет важность знания баз данных.

Многие студенты считают этот предмет приятным и интересным, даже несмотря на его сложность. Проектирование и разработка базы данных требуют и искусства, и умения. Понимание пользовательских требований и перевод их в эффективный проект базы данных можно назвать творческим процессом. Преобразование этих проектов в физические базы данных с помощью функционально полных и высокопроизводительных приложений — инженерный процесс. Оба процесса полны сложностей и приятных интеллектуальных головоломок.

Цель базы данных — помочь людям и организациям вести учет определенных вещей. На первый взгляд, эта цель кажется скромной, и вы, возможно, удивитесь, зачем нам нужна такая сложная технология и целый курс, посвященный этому предмету. Большинство из нас может вспомнить ситуации, в которых нам требуется отслеживать некоторые вещи. Мы, например, составляем список дел, которые нужно сделать на этой неделе, список покупок в магазине, список расходов для налоговой декларации и так далее. Почему не делать то же самое для информационных систем?

На самых ранних стадиях развития информационных технологий использовались списки — набитые на перфокарте и написанные на магнитной ленте. Со временем, однако, стало ясно, что только немногие проблемы можно решить с помощью таких списков.

5.1 Краткая история баз данных

В табл. 5.1 сведена история развития технологии баз данных. До середины 1960-х годов почти все компьютерные хранилища данных были на магнитных лентах. Поскольку лента может обрабатываться только последовательно, данные должны были храниться в виде списков (или последовательных файлов, как они назывались). Однако, как вы узнали в начале этой главы, хранение даже простейших данных в таком формате чревато большими проблемами.

Таблица 5.1

Период	Технология	Примечания
До 1968 г.	Обработка файлов	Предшествовала обработке баз данных. Данные хранились в виде списков. Характер обработки определялся всеобщим использованием в качестве носителя магнитной ленты
1968–1980 гг.	Иерархические и сетевые модели	Эра обработки нереляционных баз данных. Выдающейся иерархической моделью данных была DL/I фирмы IBM. Первая СУБД называлась IMS
1980 г. – нас. время	Реляционная модель данных	Реляционная модель данных впервые была опубликована в 1970 году. Реализовываться в коммерческих приложениях начала в 1980 году. IBM выпустила DB2, среди других продуктов выделяется Oracle. Реляционный язык SQL стал промышленным стандартом
1982 г.	Первые СУБД для микрокомпьютеров	Фирма Ashton-Tate разработала dBase, Microrim – R:Base, а Borland – Paradox
1985 г.	Развитие интереса к объектно-ориентированным СУБД	С развитием объектно-ориентированного программирования были предложены ООСУБД. Коммерческий успех их невелик, в первую очередь потому, что преимущества не оправдывают перевод миллиардов байтов данных организаций в новый формат. Продолжают развиваться и сейчас
1991 г.	Компания Microsoft выпускает Access	Персональная СУБД, созданная как элемент Windows. Постепенно вытеснила с рынка все другие персональные СУБД

продолжение на следующей странице

Таблица 5.1 – Продолжение

Период	Технология	Примечания
1995 г.	Первые приложения баз данных для Интернета	Базы данных стали ключевым компонентом Интернет-приложений. Популярность Интернета существенно повысила необходимость в базах данных и требования к ним
1997 г.	Применение XML к обработке баз данных	Использование XML решило проблемы, которые долго стояли перед базами данных. Ведущие производители стали интегрировать XML в свои СУБД

5.1.1 Ранние модели баз данных

С коммерческим успехом хранилищ на дисках в середине 1960-х стало возможным получение непоследовательного, или прямого, доступа к записям. Базы данных стали разрабатываться по-другому. Изначально стали успешными две конкурирующие архитектуры, или модели. Корпорация IBM разработала и внедрила DL/I (Data Language One, язык данных один), который моделировал данные в базах данных в форме иерархий, или деревьев (см. рис. 5.1, а). Эта модель, которая была разработана совместно с промышленными предприятиями, легко могла использоваться для поддержки данных, таких как сметы материалов и списки деталей, но для общих целей мало подходила. Представление неиерархических сетевых данных (рис. 5.1, б) было громоздким.

После этого CODASYL, группа, которая разрабатывала стандарты для языка COBOL, в 1970 году создала модель под названием DBTG (Data Base Task Group, группа задач баз данных). Модель DBTG была готова к представлению как иерархических, так и сетевых данных. Один раз эта модель предлагалась в качестве национального стандарта, но не была принята в первую очередь из-за своей сложности. Однако это была основа для ряда коммерчески успешных СУБД в семидесятых и восьмидесятых годах прошлого века. Наиболее успешным был продукт корпорации Cullinane под названием IDMS.

5.1.2 Недавняя история

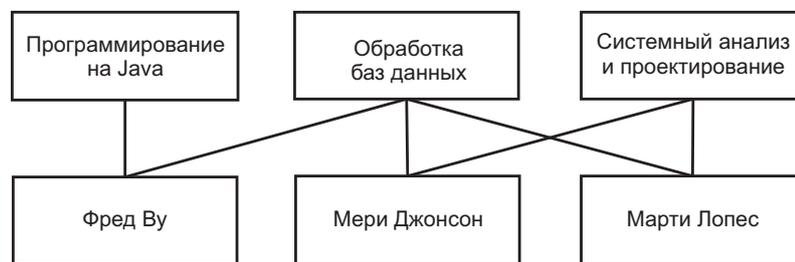
В 1991 году Microsoft выпустила Access, который на несколько лет вытеснил с рынка все остальные СУБД. Частично это произошло благодаря тому, что Access был интегрирован в Microsoft Office, и Microsoft смогла использовать свое влияние на рынке и монополию в связи с Windows для смещения других продуктов. Правда, Microsoft нужно отдать справедливость: Access — суперпродукт. Он доминирует на рынке, потому что это легкая в использовании и сильная СУБД.

Как все знают, использование Интернета распространилось в середине девяностых годов. Но немногие знают, что именно это сильно повысило значение и важность технологии баз данных. Как только ранние статические веб-страницы



Замечание: каждый узел имеет не более одного родительского узла

а



Замечание: узлы могут иметь и по несколько родительских узлов

б

Рис. 5.1

уступили дорогу динамическим, как только стали иметь успех компании типа Amazon.com и как только большие организации начали использовать Интернет для публикации своих данных, все большее и большее количество сайтов стало зависеть от баз данных. Эта тенденция продолжается и по сей день.

Наконец, в последние годы появился и стал широко использоваться язык XML, который представляет собой технологию для поддержки веб-сайтов, но был расширен для проведения важных решений, связанных с базами данных.

5.2 Распределенные базы данных

С увеличением количества локальных сетей все больше информации стало передаваться между компьютерами. Остро встал вопрос согласованности данных, хранящихся и обрабатываемых в разных местах, но логически друг с другом связанных. Успешное решение этих задач приводит к появлению распределенных баз данных, сохраняющих все преимущества настольных СУБД и в тоже время позволяющих организовывать параллельную обработку информации и поддержку целостности баз данных.

К основным особенностям этого этапа можно отнести:

- 1) полную поддержку реляционной модели данных;
- 2) работу на различных операционных платформах;

- 3) развитую систему администрирования баз данных с реализацией общей концепции средств защиты данных;
- 4) решение проблем обработки транзакций и пользовательских запросов на высоком техническом уровне;
- 5) возможность подключения к базам клиентских приложений, разработанных с использованием настольных СУБД. Развитые средства экспорта и импорта данных;
- 6) разработку ряда стандартов в рамках языков описания и манипулирования данными, начиная с SQL89, SQL92, SQL99 и технологий по обмену данными между различными СУБД, к которым можно отнести и протокол ODBC (Open Data Base Connectivity), предложенный фирмой Microsoft;
- 7) начало на данном этапе работ, связанных с концепцией объектно-ориентированных БД — ООБД.

Представителями этого этапа можно считать все серверные базы данных, такие как, Oracle, MS SQL Server, Informix, DB2, SQL Base и другие современные серверы баз данных, которых в настоящий момент насчитывается несколько десятков.

5.2.1 Перспективы развития

С появлением новой технологии доступа к данным — Интранет — появляются и новые системы хранения данных. Основное отличие использования Интранета от технологии «клиент-сервер» состоит в том, что отпадает необходимость применения специализированного клиентского программного обеспечения. Для работы с удаленной базой данных используется стандартный браузер Интернет, например Microsoft Internet Explorer или Netscape Navigator, и для конечного пользователя процесс обращения к данным происходит аналогично скольжению по Всемирной Паутине. Однако алгоритмически сложные задачи рекомендуется реализовывать в архитектуре «клиент-сервер» с разработкой специального клиентского программного обеспечения.

У каждого из вышеперечисленных подходов к работе с данными есть свои достоинства и свои недостатки, которые и определяют область применения того или иного метода, и в настоящее время все подходы широко используются.

5.2.2 Основные понятия реляционной модели

Реляционная модель основана на теории отношений (Э. Ф. Кодд, 1970 г.). При проектировании БД применяются строгие методы, построенные на нормализации отношений.

Наиболее важными характеристиками реляционной модели являются следующие:

- реляционная модель описывает данные с их естественной структурой, не добавляя каких-либо дополнительных структур, необходимых для машинного представления или для целей реализации;
- модель обеспечивает математическую основу для интеграции выводимости, избыточности и непротиворечивости отношений;

- реляционная модель позволяет добиться реальной независимости данных от их физического представления, связей между данными и способов реализации, связанных с эффективностью и подобными заботами.

В начале 80-х годов такие гиганты информационной индустрии, как Oracle Corporation, Ingres Corp., IBM, и более мелкие организации предложили ряд систем, наглядно демонстрирующих возможность применения реляционной модели для разработки БД, хранящих информацию о любой предметной области, а также возможность реализации на их основе гибких пользовательских приложений. Единственным ограничением для широкомасштабного внедрения реляционных СУБД являлась низкая производительность средств вычислительной техники. Появление и стремительное распространение персональных компьютеров, а также простота управления реляционной БД способствовали быстрому расширению рынка реляционных СУБД и признанию таких систем разработчиками и пользователями приложений.

5.2.3 Отношение, схема отношения, кортеж

Основными понятиями структурной части БД, строящихся на реляционных моделях, являются: *отношение*, *тип данных*, *домен*, *атрибут*, *кортеж*, *первичный ключ*. Смысл этих понятий наглядно поясним на примере отношения СТУДЕНТЫ (см. рис. 5.2).

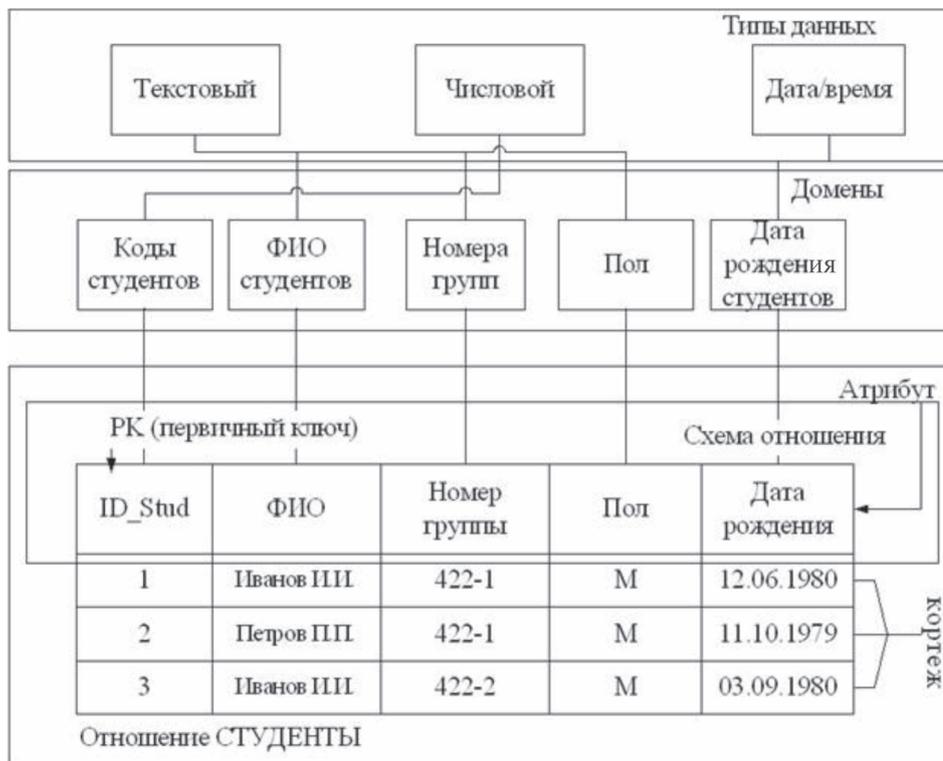


Рис. 5.2

В реляционной модели данные представляются в виде плоских таблиц, называемых **отношениями**.

Столбец таблицы называется **атрибутом** или **полем**, строка — **кортежем** отношения.

Схема отношения состоит из названий атрибутов и типов данных, на которых определены эти атрибуты.

Схема отношения есть конечное множество имен атрибутов, которым ставится в соответствие определенный тип данных (или домен, если СУБД поддерживает это понятие).

Степень схемы отношения есть мощность этого множества. Степень или арность отношения СТУДЕНТЫ равна пяти, т. е. это отношение является 5-арным.



.....
 Таким образом, схема БД есть набор схем отношений.

Отношение есть линейная структура данных, состоящая из множества кортежей, соответствующих одной схеме отношения.

Схему отношения называют заголовком, а совокупность кортежей отношения — телом отношения. Кортеж отношения (запись) описывает часть экземпляра объекта предметной области, или, если объект предметной области характеризуется одним отношением, в одном кортеже отражается полная характеристика экземпляра объекта.

Таким образом, реляционная база данных состоит из набора взаимосвязанных отношений, имена которых совпадают с именами схем отношений в схеме БД. При проектировании базы данных сначала определяют схемы отношений, после чего заносят данные. В некоторых СУБД после определения схемы отношения нельзя ни удалить, ни переименовать ни один из его атрибутов. Однако можно удалять отношения, менять их названия, менять типы данных атрибутов. Структурное изменение схем отношений БД называют также эволюцией базы данных.

5.2.4 Тип данных и домены

Типам данных в реляционной модели можно сопоставить типы данных, используемых в языках программирования. Все атрибуты в отношении должны быть определенного типа. Выделяют следующие типы данных, хранящихся в реляционных БД:

- символьные (текстовые);
- числовые;
- логические;
- дата/время.

В некоторых СУБД введены дополнительные типы данных, например в СУБД MS ACCESS используется тип данных объекта OLE — в полях такого типа можно хранить графические изображения, файлы документов и электронные таблицы, а также другие подобные объекты.



.....
 Домен — есть множество допустимых значений атрибута определенного типа.

Это понятие характерно для баз данных и аналогично подтипам в языках программирования высокого уровня, домен может быть определен на основе конкретного типа данных.

Домен определяется заданием некоторого базового типа данных, к которому относятся элементы домена, и произвольного логического выражения, применяемого к элементу типа данных. Если вычисление этого логического выражения дает результат «истина», то элемент данных является элементом домена.

Например, домен «Дата рождения» в нашем примере определен на базовом типе дата/время, но в число его значений могут входить только даты, которые могут отображать только дату без отображения времени. Атрибут ПОЛ текстового типа, определенный на одноименном домене, может принимать только два значения: М или Ж.

В СУБД, использующих понятие домена, атрибуты отношения считаются сравнимыми в том и только том случае, если эти атрибуты определены на одном домене.

5.2.5 Уникальность кортежей отношения

Отношениям как основной единице построения реляционных БД присущи определенные свойства и правила заполнения тела отношения сведениями об экземплярах объекта предметной области.

Главное отличие реляционного отношения от плоской таблицы заключается в том, что отношение (в классическом его понимании) не может иметь дубликатов кортежей.

Поскольку отношение есть множество кортежей, а каждое множество не должно включать одинаковых элементов, то уже этого достаточно для обеспечения уникальности кортежей. Обеспечить это требование возможно с помощью первичного ключа — набора атрибутов, значения которых однозначно определяют кортеж отношения.

В современных СУБД для обеспечения свойства уникальности записей в каждой таблице БД дополнительно вводится идентификатор записи (инкремент, счетчик и т. д.) — атрибут, значение которого является уникальным для каждой записи отношения БД. Наличие такого идентификатора записи необходимо в связи с возможным отсутствием первичного ключа в некоторых создаваемых разработчиками отношениях. Обычно же в состав первичного ключа включается минимальный набор атрибутов отношения, являющегося идентификатором объекта. Наличие первичного ключа необходимо для определения связей между отношениями и, как следствие, для обеспечения целостности данных.

5.2.6 Отсутствие упорядоченности кортежей и атрибутов

В каком бы порядке ни хранились данные в отношении, их смысл не будет изменяться. Действительно, с помощью языков манипулирования данными при организации запросов всегда можно указать тот или иной порядок сортировки данных для результирующего набора данных.

Хотя некоторые реляционные СУБД позволяют обеспечить доступ к атрибуту отношения по порядковому номеру, который присваивается атрибуту в схеме отношения в этих СУБД, атрибуты в большинстве своем не упорядочены, и для ссылки

на значение атрибута обычно (а в языках манипулирования данными — обязательно) используется имя атрибута.

Обеспечение этих свойств позволяет, с одной стороны, СУБД хранить данные в произвольном порядке, с другой стороны, обеспечить разработчику и пользователю возможность манипулировать данными в отношениях без нарушения структурной целостности системы.

5.2.7 Атомарность значений атрибутов, первая нормальная форма

Одним из главных свойств отношений является соблюдение принципа нормализации, иначе говоря, каждое отношение в БД должно удовлетворять первой нормальной форме (1-NF).

Отношение находится в первой нормальной форме (нормализовано по 1-NF) тогда и только тогда, когда значения его атрибутов являются атомарными.

Другими словами, значения атрибутов в 1-NF не содержат множества значений, иными словами, значением атрибута отношения не может быть какое-либо отношение; значениями атрибутов не являются составные данные. Каждое отношение в 1-NF является особым случаем ненормализованного отношения, но каждое ненормализованное отношение не находится в 1-NF. На нижеследующем рисунке приведен пример ненормализованного отношения ГРУППЫ. Можно сказать, что здесь мы имеем бинарное отношение, где значением атрибута СТУДЕНТЫ является отношение.

№ группы	Студенты				
	№ зачетной книжки	ФИО	Пол	Место рождения	Дата рождения
412-1	1992412-11	Карасев А.А.	М	г. Чита	27.08.75
412-2	2002412-02	Красников И.И.	М	г. Бийск	12.02.83
432-1	1992432-11	Данилов О.В.	М	г. Алматы	27.08.75
	1992432-12	Раевский А.И.	М	г. Бишкек	20.05.75
421-1	2002421-01	Иванков И.С.	Ж	г. Томск	11.10.85
	2002421-02	Авдеев Н.В.	М	г. Омск	01.04.84

Рис. 5.3

На следующем рисунке 5.4 отношение СТУДЕНТЫ является нормализованным вариантом отношения ГРУППЫ. В этом отношении на пересечении столбца и строки находится только одно значение:

В современных реляционных СУБД допускается хранение в полях таблиц сложных объектов (полей типа OLE), что, однако, не противоречит принципу атомарности данных, поскольку в данных полях содержится либо ссылка на внешний файл объекта, либо непосредственно сам OLE-объект, являющийся неделимой информационной единицей.

Что же касается составных данных, то возможность хранения в одном поле перечислимой информации типа «белый, синий, черный» остается на совести разработчика БД — либо принимается тезис о необходимости четкого описания объекта

№ зачетной книжки	ФИО студента	Пол	Место рождения	Дата рождения	№ группы
1992412-11	Карасев А.А.	М	г. Чита	27.08.75	412-1
2002412-02	Красников И.И.	М	г. Бийск	12.02.83	412-2
1992432-11	Данилов О.В.	М	г. Алматы	27.08.75	432-1
1992432-12	Раевский А.И.	М	г. Бишкек	20.05.75	
2002421-01	Иванков И.С.	Ж	г. Томск	11.10.85	421-1
2002421-02	Авдеев Н.В.	М	г. Омск	01.04.84	

Рис. 5.4

предметной области для обеспечения возможности манипулирования информацией, представленной в этом поле, что влечет необходимость дальнейшей нормализации; либо эта информация принимается как сопроводительная и имеет статус примечания.

5.2.8 Характеристика реляционной модели

К. Дейт в 1998 г. дает следующее определение:



.....
Реляционная модель состоит из трех частей, описывающих разные аспекты реляционного подхода: структурной, манипуляционной и целостной.

Единственной структурой данных, используемой в реляционных БД, является нормализованное n -арное отношение — это характеристика структурной части.

В манипуляционной части модели утверждаются два фундаментальных механизма манипулирования реляционными БД — *реляционная алгебра* и *реляционное исчисление*, на основе которых строятся все известные *реляционные языки управления БД*.

Что касается целостной части реляционной модели данных, то для нее фиксируются два базовых требования, характерные для любой реляционной СУБД: первое требование, согласно которому отношение должно обладать первичным ключом для обеспечения уникальности записей, называется требованием целостности сущностей; второе — требованием целостности по ссылкам.

При описании сложных объектов предметной области обойтись одним отношением, соблюдая принцип нормализации, бывает очень сложно, а зачастую и практически невозможно. Таким образом, один объект может быть описан в нескольких отношениях. Поясним смысл требования целостности по сущностям на примере. Отобразим в реляционной БД сущность ФАКУЛЬТЕТ, содержащую информацию о группах — НОМЕР ГРУППЫ, КОЛИЧЕСТВО СТУДЕНТОВ — и студентах факультета, а именно: № ЗАЧЕТНОЙ КНИЖКИ, ФИО СТУДЕНТА, ПОЛ, МЕСТО РОЖДЕНИЯ, ДАТА РОЖДЕНИЯ.

В результате проектирования БД получим два отношения СТУДЕНТЫ и ГРУППЫ со схемами, представленными на рисунке:



Рис. 5.5

Атрибут № ГРУППЫ появляется в отношении СТУДЕНТЫ для обеспечения возможности восстановить сущность ФАКУЛЬТЕТ. Значение атрибута НОМЕР ГРУППЫ отношения СТУДЕНТЫ должно соответствовать значению атрибута НОМЕР ГРУППЫ в каком-либо кортеже отношения ГРУППЫ. Атрибут № ГРУППЫ в отношении СТУДЕНТЫ называется внешним ключом. Значения такого атрибута отношения однозначно характеризуют сущности, представленные кортежами другого отношения, т. е. соответствуют значению его первичного ключа.

Таким образом, отношение, в котором на каком-либо атрибуте (может быть составным) определен внешний ключ, ссылается на отношение, в котором соответствующий атрибут является первичным ключом. Говорят также, что отношения связаны по некоторому ключу.

Требование целостности по ссылкам (требование внешнего ключа) состоит в том, что для каждого значения внешнего ключа, появляющегося в ссылающемся отношении, должен найтись кортеж с таким же значением первичного ключа в отношении, на которое ведет ссылка, либо значение внешнего ключа должно быть неопределенным, т. е. ни на что не указывать. Для нашего примера это означает, что если для студента указан номер группы, то эта группа должна существовать. Ограничения целостности сущностей и ограничения по ссылкам должны поддерживаться в большинстве современных СУБД.

Для обеспечения ограничения по ссылкам при добавлении и изменении данных ссылающегося отношения необходимо обеспечить проверку на ввод значений внешнего ключа. При изменении значения первичного ключа в отношениях необходимо изменять соответствующие значения внешних ключей. В некоторых СУБД эта процедура производится автоматически. Такой принцип получил название — «каскадное обновление данных».

При удалении кортежа из отношения, на которое ведет ссылка, существуют три подхода, поддерживающих целостность по ссылкам.

Первый подход заключается в том, что запрещается производить удаление кортежа, на который существуют ссылки (т. е. сначала нужно либо удалить ссылающиеся кортежи, либо соответствующим образом изменить значения их внешнего ключа).

Во втором подходе при удалении кортежа, на который имеются ссылки, во всех ссылающихся кортежах значение внешнего ключа автоматически становится неопределенным.

Третий подход (каскадное удаление) состоит в том, что при удалении кортежа из отношения, на которое ведет ссылка, из ссылающегося отношения автоматически удаляются все ссылающиеся кортежи.

При разработке структуры БД целостность данных обеспечивается либо на уровне СУБД, либо на уровне прикладной программы. Для реляционных СУБД, поддерживающих домены, существует понятие целостности доменов. Обеспечение механизма целостности доменов гарантирует, что все значения некоторого атрибута принадлежат множеству допустимых значений. Реализация механизма целостности доменов осуществляется с помощью предварительного задания характеристик домена в описательной части БД.

Соблюдение целостности БД является одним из условий обеспечения качественного хранения информации, а также гарантией отсутствия противоречивых данных, что позволяет постоянно актуализировать эти данные и использовать их многократно и без потерь.

5.2.9 Технология манипулирования данными в реляционной структуре

Согласно К. Дейту для манипуляционной составляющей определяются два базовых механизма манипулирования реляционными данными — реляционная алгебра (основана на теории множеств) и реляционное исчисление (основано на исчислении предикатов первого порядка и разделено на два типа — исчисление доменов и исчисление предикатов).

Все современные языки манипулирования данными основаны на этих понятиях, а так как реляционная алгебра и реляционное исчисление замкнуты относительно понятия отношения, то любое выражение или формула могут быть представлены как отношения, что позволяет использовать их в других реляционных выражениях или формулах.

Применение реляционной алгебры и реляционного исчисления дает возможность интерпретировать сложные пользовательские запросы в виде простых предложений на языке манипулирования данными. По этой причине эти механизмы включены в реляционную модель данных.

Конкретный язык манипулирования реляционными БД называется реляционно полным, если любой запрос, выражаемый с помощью одного выражения реляционной алгебры или одной формулы реляционного исчисления, может быть выражен с помощью одного оператора этого языка.

В принципе для любого выражения реляционной алгебры можно построить формулу реляционного исчисления, приводящую к тому же результату, что и реляционное выражение, и наоборот. Таким образом, эти два понятия являются эквивалентными. Для облегчения технической реализации пользовательских запросов в реляционной модели данных имеют место оба механизма манипулирования данными.

5.3 Нормализация отношений

При проектировании базы данных разработчику необходимо определить концептуальное и соответствующее ему физическое представление, после чего происходит создание внешних представлений с учетом потребностей прикладных программ, использующих спроектированную БД. При проектировании БД выделяют две стадии проектирования БД — логическое и физическое проектирование.

Логическое проектирование БД ставит своей целью представление реальной предметной области в абстрактных моделях таким образом, чтобы эти модели данных максимально отражали в себе объекты выбранной предметной области.

Физическое проектирование БД предполагает создание структуры данных, определенной для конкретных СУБД, на основании спроектированной логической модели предметной области, а также реализацию дополнительных элементов БД (триггеров, индексов и т. д.).

При проектировании реляционных БД трудно представить какие-либо общие решения по физическому проектированию. Ограничения в каждом конкретном случае накладывает СУБД, в которой предполагается создавать базу данных.

Таким образом, абстрагируясь от конкретных СУБД, будем считать, что при проектировании реляционной БД необходимо определить отношения, составляющие БД, их атрибуты, а также ключи и связи между отношениями.

Основной направляющей при проектировании реляционных БД является технология нормализации отношений, направленная на обеспечение безызбыточного и бесконфликтного хранения информации в отношениях БД. При использовании принципа нормализации процесс проектирования БД производится методом последовательных приближений к требуемому набору схем отношений. Нормализации подвергаются отношения, каждое из которых содержит характеристики объектов предметной области, при этом на каждом следующем шаге проектирования (нормализации) с помощью декомпозиции отношений достигается такой набор схем отношений, что каждая следующая нормальная форма (НФ или NF) обладает лучшими свойствами, чем предыдущая.

При нормализации отношений каждой NF соответствует свой набор ограничений, тогда справедливо утверждение, что отношение находится в какой-либо нормальной форме, если удовлетворяет характерному ей набору ограничений. Так, ограничением первой нормальной формы (1NF), как мы отмечали ранее, является условие атомарности атрибутов отношения. И поскольку в основе реляционной модели лежит отношение, находящееся в 1NF, в дальнейшем будем подразумевать, что все рассматриваемые нами отношения уже находятся в 1NF или, как говорят, нормализованы по 1NF.

Процесс нормализации позволяет разработчику БД глубже понять семантику атрибутов при проектировании структуры БД и их взаимосвязи, а также облегчает проведение анализа предметной области.

За время развития технологии проектирования реляционных БД были выделены следующие нормальные формы:

- 1) первая нормальная форма (1NF);
- 2) вторая нормальная форма (2NF);

- 3) третья нормальная форма (3NF);
- 4) нормальная форма Бойса-Кодда (BCNF);
- 5) четвертая нормальная форма (4NF);
- 6) пятая нормальная форма, или нормальная форма проекции-соединения (5NF или PJ/NF).

Обычно на практике применение находят только первые три нормальные формы.

Для атрибутов отношений реляционных БД справедливо определение функциональной зависимости: так, в отношении R атрибут Y функционально зависит от атрибута X в том и только в том случае, если каждому значению X соответствует одно значение Y . Схематично функциональную зависимость атрибута Y от атрибута X изображают так:

$$R.X \rightarrow R.Y \quad \text{или} \quad R(X \rightarrow Y).$$

Из этого определения следуют определения полной и транзитивной функциональной зависимости.

Функциональная зависимость атрибута Y от атрибута X называется полной, если атрибут Y не зависит функционально от любого точного подмножества X (следует учесть, что атрибуты X и Y могут быть составными).

Функциональная зависимость $X \rightarrow Y$ называется транзитивной, если имеется такой атрибут Z , что существуют функциональные зависимости $X \rightarrow Z$ и $Z \rightarrow Y$ и при этом отсутствует функциональная зависимость $Z \rightarrow X$.

В отношении R атрибуты взаимно независимы, если ни один из этих атрибутов функционально не зависит от других.

5.3.1 Вторая нормальная форма

Рассмотрим процесс нормализации отношения по 2NF, т. е. приведем отношение, показанное на рисунке 5.6, находящееся в первой нормальной форме, ко второй нормальной форме.

Схема отношения УСПЕВАЕМОСТЬ: (№ ЗАЧЕТНОЙ КНИЖКИ, ФИО СТУДЕНТА, МЕСТО РОЖДЕНИЯ, ДАТА РОЖДЕНИЯ, КУРС, СРЕДНИЙ БАЛЛ). Первичным ключом отношения является совокупность атрибутов № ЗАЧЕТНОЙ КНИЖКИ, КУРС.

Можно выявить следующие функциональные зависимости:

№ зачетной книжки \Rightarrow ФИО студента

№ зачетной книжки \Rightarrow Место рождения

№ зачетной книжки \Rightarrow Дата рождения

№ зачетной книжки, Курс \Rightarrow Средний балл

Первичный ключ данного отношения состоит из атрибутов № ЗАЧЕТНОЙ КНИЖКИ, КУРС, однако в нашем случае атрибуты ФИО СТУДЕНТА, МЕСТО РОЖДЕНИЯ и ДАТА РОЖДЕНИЯ функционально зависят только от части первичного ключа — № ЗАЧЕТНОЙ КНИЖКИ. Таким образом, при работе с таким ненормализованным отношением невозможно обеспечить корректную работу по выполнению операции вставки нового кортежа при занесении данных о студентах,

Отношение УСПЕВАЕМОСТЬ

№ зачетной книжки	ФИО студента	Место рождения	Дата рождения	Курс	Средний балл
1992412-11	Карасев А.А.	г. Чита	27.08.75	1	4,5
1992412-11	Карасев А.А.	г. Чита	27.08.75	2	4,1
1992412-11	Карасев А.А.	г. Чита	27.08.75	3	5
1992412-11	Карасев А.А.	г. Чита	27.08.75	4	4,8
1992432-11	Данилов О.В.	г. Алматы	27.08.75	1	4,6
1992432-11	Данилов О.В.	г. Алматы	27.08.75	2	4,4
1992432-11	Данилов О.В.	г. Алматы	27.08.75	3	4,2
1992432-11	Данилов О.В.	г. Алматы	27.08.75	4	4,3
1992432-12	Раевский А.П.	г. Бишкек	20.05.75	1	3,8
⋮					
1992432-12	Раевский А.П.	г. Бишкек	20.05.75		

Рис. 5.6

сведений об успеваемости которых еще нет, т. к. первичный ключ не может содержать неопределенное значение (в нашем случае часть ключа КУРС не определена). При удалении записи из отношения мы теряем связь конкретного студента с его успеваемостью за конкретный курс. Аналогично мы получим неверный результат при выполнении подсчета общего количества студентов. Такие неприятные явления называются аномалиями схемы отношения или коллизиями. Эти недостатки реляционных отношений устраняются путем нормализации по 2NF.

Отношение R находится во второй нормальной форме (2NF) тогда и только тогда, когда отношение находится в первой нормальной форме и каждый его неключевой атрибут полностью зависит от первичного ключа; или, что тоже справедливо, отношение, находящееся во второй нормальной форме, не содержит атрибутов, зависящих от части ключа.

Приведение данного отношения к 2NF заключается в разбиении (декомпозиции) на два отношения, удовлетворяющих соответствующим требованиям нормализации. Можно произвести следующую декомпозицию отношения УСПЕВАЕМОСТЬ в два отношения СТУДЕНТЫ и УСПЕВАЕМОСТЬ СТУДЕНТОВ (см. рис. 5.7).

Первичным ключом отношения СТУДЕНТЫ является атрибут № ЗАЧЕТНОЙ КНИЖКИ.

Можно выявить следующие функциональные зависимости:

№ зачетной книжки \Rightarrow *ФИО студента*

№ зачетной книжки \Rightarrow *Место рождения*

№ зачетной книжки \Rightarrow *Дата рождения*

Первичным ключом отношения Успеваемость студентов являются атрибуты № зачетной книжки, Курс.

В этом отношении существует одна функциональная зависимость: № ЗАЧЕТНОЙ КНИЖКИ, КУРС \Rightarrow СРЕДНИЙ БАЛЛ.

СТУДЕНТЫ

№ зачетной книжки	ФИО студента	Место рождения	Дата рождения
1992412-11	Карасев А.А.	г. Чита	27.08.75
1992432-11	Данилов О.В.	г. Алматы	27.08.75
1992432-12	Раевский А.И.	г. Бишкек	20.05.75

УСПЕВАЕМОСТЬ СТУДЕНТОВ

№ зачетной книжки	Курс	Средний балл
1992412-11	1	4,5
1992412-11	2	4,1
1992412-11	3	5
1992412-11	4	4,8
1992432-11	1	4,6
1992432-11	2	4,4
1992432-11	3	4,2
1992432-11	4	4,3
1992432-12	1	3,8
1992432-12	4	5

Рис. 5.7

Каждое из этих двух отношений находится в 2NF, и в них устранены отмеченные выше аномалии (легко проверить, что все указанные операции выполняются без проблем).

В отношении, помимо одного первичного ключа, могут находиться атрибуты, по значению которых также можно однозначно определить записи. Такие атрибуты называются альтернативными ключами. Если допустить наличие нескольких ключей, то определение 2NF примет следующий вид:



.....
отношение R находится во второй нормальной форме (2NF) в том и только в том случае, когда оно находится в 1NF и каждый его неключевой атрибут полностью зависит от каждого ключа этого отношения.

5.3.2 Третья нормальная форма

Добавим в отношение СТУДЕНТЫ два атрибута: № ГРУППЫ и ФИО КУРАТОРА (см. рис. 5.8).

Первичным ключом отношения СТУДЕНТЫ является атрибут № ЗАЧЕТНОЙ КНИЖКИ. Отношение находится во второй нормальной форме, поскольку отсутствуют зависимости атрибутов от части первичного ключа.

Функциональная зависимость № ЗАЧЕТНОЙ КНИЖКИ \Rightarrow ФИО КУРАТОРА транзитивная, поскольку является следствием функциональных зависимостей № ЗАЧЕТНОЙ КНИЖКИ \Rightarrow № ГРУППЫ и № ГРУППЫ \Rightarrow ФИО КУРАТОРА. Другими словами, ФИО КУРАТОРА является характеристикой не студента, а группы, в которой он проходит обучение.

№ зачетной книжки	ФИО куратора	Дата рождения	Место рождения	№ группы	ФИО куратора
1992412-11	Карасев А.А.	27.08.75	г. Чита	412-1	Самойлов С.С.
1992432-11	Данилов О.В.	27.08.75	г. Алматы	432-1	Авдеев Р.М.
1992432-12	Раевский А.И.	20.05.75	г. Бишкек	432-1	Авдеев Р.М.
⋮					
1992432-22	Глазов О.А.	04.07.75	г. Киров	432-1	Авдеев Р.М.

Рис. 5.8

Легко заметить, что в рассматриваемом отношении существуют следующие аномалии включения и удаления. В результате выполнения операции включения мы не сможем занести в базу данных информацию о кураторе группы до тех пор, пока в этой группе не появится хотя бы один студент, поскольку первичный ключ не может содержать неопределенное значение, однако на практике еще до составления списков групп уже известны их кураторы. При удалении кортежа, описывающего последнего студента данной группы, мы рискуем потерять информацию о том, кто из преподавателей кафедры, на которой обучается студент, являлся куратором этой группы.

Чтобы корректно изменить ФИО куратора группы, необходимо последовательно изменить все кортежи, описывающие студентов этой группы, т. е. в отношении СТУДЕНТЫ благодаря добавлению двух новых атрибутов по-прежнему существуют аномалии. Их можно устранить путем дальнейшей нормализации отношения. Дадим определение третьей нормальной формы.

Отношение R находится в третьей нормальной форме (3NF) в том и только в том случае, если находится в 2NF и каждый неключевой атрибут нетранзитивно зависит от первичного ключа, т. е. среди атрибутов отношения нет атрибутов, транзитивно зависящих от ключа (среди его неключевых атрибутов нет зависящих от другого неключевого атрибута).

Приведение отношения к 3NF (нормализация по 3NF) также заключается в декомпозиции этого отношения. Можно произвести декомпозицию отношения СТУДЕНТЫ в два отношения СПРАВОЧНИК СТУДЕНТОВ и ГРУППЫ

СПРАВОЧНИК СТУДЕНТОВ

№ зачетной книжки	ФИО студента	Дата рождения	Место рождения	№ Группы
1992412-11	Карасев А.А.	27.08.75	г. Чита	412-1
1992432-11	Данилов О.В.	27.08.75	г. Алматы	432-1
1992432-12	Раевский А.И.	20.05.75	г. Бишкек	432-1
1992432-22	Глазов О.А.	04.07.75	г. Киров	432-1

КУРАТОРЫ ГРУПП

№ Группы	ФИО куратора
412-1	Самойлов С.С.
432-1	Авдеев Р.М.

Рис. 5.9

Первичным ключом отношения КУРАТОРЫ ГРУПП является атрибут № ГРУППЫ. Для этого отношения характерна одна функциональная зависимость: № ГРУППЫ → ФИО КУРАТОРА. Результирующие отношения находятся в 3NF и свободны от отмеченных аномалий. Первичным ключом отношения СПРАВОЧНИК СТУДЕНТОВ является атрибут № ЗАЧЕТНОЙ КНИЖКИ. В отношении имеют место следующие функциональные зависимости:

№ зачетной книжки ⇒ ФИО студента

№ зачетной книжки ⇒ Дата рождения

№ зачетной книжки ⇒ Место рождения

№ зачетной книжки ⇒ № группы

При проектировании БД третья нормальная форма схем отношений достаточно в большинстве случаев, и приведением к третьей нормальной форме процесс проектирования реляционной базы данных обычно заканчивается. Однако иногда полезно продолжить процесс нормализации.

5.4 Итоги

Современные базы данных являются основой многочисленных информационных систем. Информация, накопленная в них, является чрезвычайно ценным материалом, и в настоящий момент широко распространяются методы обработки баз данных с точки зрения извлечения из них дополнительных знаний, методов, которые связаны с обобщением и различными дополнительными способами обработки данных.

Пробуйте и дерзайте, познакомившись с базами данных, вы еще не раз с ними столкнетесь в жизни. Желаем вам успехов и корректных запросов к базам данных. Вы ведь уже знаете: каков вопрос, таков и ответ. Любая база данных может стать вашим помощником или мучителем, это зависит от разработчиков, нам хочется, чтобы для вас они всегда играли только первую роль.



Контрольные вопросы по главе 5

- 1) Что такое реляционные базы данных?
- 2) Перечислите основные понятия реляционной модели.
- 3) Что включает в себя нормализация отношений?
- 4) Назовите основные этапы развития баз данных.
- 5) Где используются распределенные базы данных?

Глава 6

БЕЗОПАСНОСТЬ КОМПЬЮТЕРНЫХ СИСТЕМ

6.1 Понятие безопасности

Многие компании обладают ценной информацией, которую они тщательно охраняют. Эта информация может быть технической (например, архитектура новой микросхемы или программного обеспечения), коммерческой (исследования конкурентоспособности или маркетинговые планы), финансовой (планы биржевых операций), юридической (документы о потенциальном слиянии или разделе фирм) и т. д. Часто эта информация защищается при помощи охранника в униформе, стоящего у входа в здание и проверяющего у всех входящих в здание наличие определенного значка. Кроме того, многие офисы и картотечные шкафы могут запираются на ключ, чтобы гарантировать доступ к информации только авторизованных сотрудников.

По мере того как возрастают объемы информации, хранящейся в компьютерных системах, необходимость в защите информации становится все важнее. Некоторые аспекты, связанные с защитой информации, находящейся в компьютерных системах, аналогичны вопросам, связанным с обеспечением безопасности информации на бумажных носителях. Другие же являются уникальными для компьютерных систем.

Проблема безопасности многогранна. Тремя ее наиболее важными аспектами являются природа угроз, природа злоумышленников и случайная потеря данных.

6.2 Виды угроз

С позиции безопасности компьютерных систем можно выделить 3 основных вида угроз. Первый вид это **нарушение конфиденциальности данных**. Конфиденциальность данных заключается в том, что секретные данные должны оста-

ваться секретными. В частности, если владелец некоторых данных решил, что эти данные будут доступны только определенному кругу лиц, система должна гарантировать, что к этим данным не смогут получить доступ лица за пределами установленного круга. Второй вид — **нарушение целостности данных** (порча или подделка данных). Сохранение целостности данных подразумевает что пользователи, не имеющие на то прав, не могут редактировать или удалять данные. Третьим видом выделим **отказ в обслуживании**, когда в ходе каких-либо действий система выводится из строя, становясь недоступной. Например, если компьютер является сервером Интернета, он может быть затоплен мощным потоком запросов, при этом все его процессорное время уйдет на изучение входящих запросов, а не на ответы пользователям. Так, если обработка запроса чтения web-страницы занимает 100 мкс, то любой пользователь, способный послать 10 000 запросов в секунду, может ликвидировать сервер.

6.3 Злоумышленники

Большинство людей соблюдают закон, поэтому зачем беспокоиться о безопасности? Все дело в том, что, к сожалению, некоторые люди не столь добродетельны и желают доставить другим неприятности. Злоумышленники подразделяются на два вида. Пассивные злоумышленники просто пытаются прочитать файлы, которые им не разрешено читать. Активные злоумышленники пытаются незаконно изменить данные. Наиболее распространенными категориями злоумышленников являются:

- 1) Случайные любопытные пользователи, не применяющие специальных технических средств. У многих людей есть компьютеры, соединенные с общим файловым сервером. И если не установить специальной защиты, благодаря естественному любопытству многие люди станут читать чужую электронную почту и другие файлы.
- 2) Члены организации, занимающиеся шпионажем. Студенты, системные программисты, операторы и другой технический персонал часто считают взлом системы безопасности локальной компьютерной системы личным вызовом. Как правило, они имеют высокую квалификацию и готовы посвящать достижению поставленной перед собой цели значительное количество времени.
- 3) Те, кто совершают решительные попытки личного обогащения. Некоторые программисты, работающие в банках, предпринимали попытки украсть деньги у банка, в котором они работали. Используя схемы варьировались от изменения способов округления сумм в программах для сбора, таким образом, с миру по нитке, до шантажа («Заплатите мне, или я уничтожу всю банковскую информацию»).
- 4) Занимающиеся коммерческим и военным шпионажем. Шпионаж представляет собой серьезную и хорошо финансируемую попытку конкурента или другой страны украсть программы, коммерческие тайны, ценные идеи и технологии, схемы микросхем, бизнес-планы и т. д.

Помимо различных угроз со стороны злоумышленников, существует опасность потери данных в результате несчастного случая. К наиболее распространенным причинам случайной потери данных относятся:

- 1) Форс-мажор: пожары, наводнения, землетрясения, войны, восстания, крысы, изгрызшие ленты или гибкие диски.
- 2) Аппаратные и программные ошибки: сбой центрального процессора, нечитаемые диски или ленты, ошибки при передаче данных, ошибки в программах.
- 3) Человеческий фактор: неправильный ввод данных, неверно установленные диск или лента, запуск не той программы, потерянные диск или лента и т. д.

6.4 Основы криптографии

Задача криптографии заключается в том, чтобы взять сообщение или файл, называемый открытым текстом, и преобразовать его в зашифрованный текст таким образом, чтобы только посвященные могли преобразовать его обратно в открытый текст. Для всех остальных зашифрованный текст должен представлять собой просто непонятный набор битов.

6.4.1 Шифрование с закрытым (симметричным) ключом

Принцип такого шифрования заключается в том, что имеется некоторое правило, именуемое ключом, по которому исходный текст преобразуется в зашифрованный. Точно по этому же правилу зашифрованный текст преобразуют в исходный. В качестве простого примера можно привести следующее правило: каждой букве ставится в соответствии ее порядковый номер в алфавите. Следовательно, для дешифрации надо взять число и заменить его на букву, находящуюся на соответствующей позиции в алфавите.

В такой системе шифрования возникает затруднение при передаче ключа между пользователями. Необходимо следить за тем, чтобы ключ не попал к злоумышленнику. Так как, имея на руках закрытый ключ, злоумышленник без труда может читать зашифрованную информацию.

Даже если гарантировать отсутствие возможности перехвата злоумышленником закрытого ключа, такая система шифрования имеет существенный недостаток. Подобный шифр легко взламывается даже при довольно небольших порциях зашифрованного текста. Для подбора шифра может быть использовано преимущество статистических характеристик естественных языков. Например, в английском языке буква **e** встречается в тексте чаще всего. Следом за ней по частоте использования идут буквы **t**, **o**, **a**, **n**, **i** и т. д. Наиболее часто встречающимися комбинациями из двух символов, или биграмм, являются **th**, **in**, **er**, **re** и т. д. При помощи данной информации взлом такого шифра несложен. Для серьезного уровня безопасности следует использовать ключи длиной в 1024 бит.

Одним из преимуществ систем шифрования с закрытым ключом является высокая скорость работы алгоритмов шифрования и дешифрования.

6.4.2 Шифрование с открытым (несимметричным) ключом

Главное свойство этой системы заключается в том, что для шифрования и дешифрования используются различные ключи и что по заданному ключу шифрования определить соответствующий ключ дешифрации практически невозможно. Как правило, для связи между собой открытого и закрытого ключей используются достаточно сложные математические вычисления. Имея на руках зашифрованное сообщение и открытый ключ шифрования, произвести процедуру дешифрации практически невозможно. При таких условиях ключ шифрования может быть сделан открытым, и только ключ дешифрации будет храниться в секрете.

Шифрование с открытым ключом используется следующим образом. Все участники выбирают пару ключей (открытый ключ, закрытый ключ) и публикуют открытый ключ. Открытый ключ используется для шифрования, а закрытый — для дешифрации. Чтобы отправить пользователю секретное сообщение, корреспондент зашифровывает его открытым ключом получателя. Поскольку закрытый ключ есть только у получателя, только он один сможет расшифровать сообщение.

Недостаток систем шифрования с открытым ключом заключается в том, что они в тысячи раз медленнее, чем системы симметричного шифрования.

6.4.3 Цифровые подписи

Нередко возникает и обратная ситуация, когда необходимо чтобы документ могли прочитать все, но ни кто не мог внести в него изменения. Например: для декларирования товаров и услуг (таможенные декларации), использования в банковских системах, для обязательной отчетности перед государственными учреждениями и т. д.

Такая возможность реализована с помощью механизма цифровой подписи. С помощью цифровой подписи можно подписывать сообщения, посылаемые по электронной почте, и другие цифровые документы таким образом, чтобы отправитель не смог потом отрицать, что посылал их.

В общих чертах механизм цифровой подписи очень похож на шифрование с открытым ключом. Только в этом случае открытый ключ предназначен для дешифрации и известен всем, а закрытый ключ находится у автора документа и используется им для шифрования документа.

6.5 Аутентификация пользователей

В обычной жизни мы узнаем друг друга в лицо. Если знакомы. Если не знакомы — по паспорту или аналогичному документу с фотографией. Тем самым определяем, что дозволено данному человеку, можно ли ему читать или даже править документы, находящиеся на вашем рабочем столе или в ящиках вашего стола, или же, наоборот, при нем нежелательно даже разговаривать на какие-либо темы, касающиеся работы. То же самое происходит и в компьютерных системах, однако «опознать» человека, сидящего за компьютером, несколько сложнее — это требует достаточно специфичных методов. Большинство методов аутентификации пользователей основаны на распознавании:

- 1) чего-то, известного пользователю;
- 2) чего-то, имеющегося у пользователя;
- 3) чего-то, чем является пользователь.

6.5.1 С использованием пароля

В наиболее широко применяемой форме аутентификации от пользователя требуется ввести имя и пароль. Самый простой способ реализации паролей заключается в поддержании централизованного списка пар (имя регистрации, пароль). Вводимое имя отыскивается в списке, а введенный пользователем пароль сравнивается с хранящимся в списке. Если пароли совпадают, регистрация в системе разрешается, если нет — в регистрации пользователю отказывается. Как правило, при вводе пароля компьютер не должен отображать вводимые символы, чтобы находящиеся рядом посторонние люди не смогли узнать пароля.

Большинство взломщиков проникают в систему, просто перебирая множество комбинаций имени и пароля, пока не находят комбинацию, которая работает. Многие пользователи используют в качестве регистрационного имени свое собственное имя в той или иной форме. Конечно, угадать регистрационное имя — это еще не все. Также требуется подобрать пароль. Насколько это сложно? В 1979 году Моррисом и Томпсоном был написан труд по вопросу безопасности паролей на основе исследований систем UNIX. Авторы скомпилировали список вероятных паролей: имя и фамилия, названия улиц, городов, слова из словарей среднего размера (также слова, написанные задом наперед), автомобильные номера и короткие строки случайных символов. Затем они сравнили свой полученный таким образом список с системным файлом паролей, чтобы посмотреть, есть ли совпадения. Как выяснилось, более 86 % от общего количества паролей в файле оказались в их списке. Это означает, что взломщик без особого труда может получить список потенциальных регистрационных имен и список потенциальных паролей.

6.5.2 Совершенствование безопасности паролей

Добавление случайных чисел к файлу паролей защищает систему от взломщиков, пытающихся заранее составить большой список зашифрованных паролей. Однако данный метод бессилен помочь в том случае, когда пароль легко отгадать, например если пользователь использует пароль, совпадающий с регистрационным именем. Взломщик может просто попытаться отгадать пароли один за другим. Обучение пользователей в данной области может помочь. Рекомендуется использовать в качестве паролей случайные, легко произносимые бессмысленные слова, желательно с использованием прописных символов и специальных символов, добавленных внутрь.

Среди требований к паролю могут быть, например, следующие:

- 1) Пароль должен содержать как минимум семь символов.
- 2) Пароль должен содержать как строчные, так и прописные символы.
- 3) Пароль должен содержать как минимум одну цифру или специальный символ.
- 4) Пароль не должен представлять собой слово, содержащееся в словаре, имя собственное и т. д.

6.5.3 С использованием физического объекта

Другой метод аутентификации пользователей заключается в проверке некоторого физического объекта, который есть у пользователя, а не информации, которую он знает. Например, в течение столетий применялись металлические дверные ключи. Сегодня этим физическим объектом часто является пластиковая карта, вставляемая в специальное устройство чтения, подключенное к терминалу или компьютеру. Как правило, пользователь должен не только вставить карту, но также ввести пароль, чтобы предотвратить использование потерянной или украденной карты.

Существует две разновидности пластиковых карт, хранящих информацию: магнитные карты и карты с процессором. Магнитные карты содержат около 140 байт информации, записанной на магнитной ленте, приклеенной к пластику. Эта информация может быть считана терминалом и передана на центральный компьютер. Применять магнитные карты для идентификации рискованно, так как устройства чтения и записи этих карт дешевы и широко распространены.

Карты, содержащие в себе микросхемы, в свою очередь, подразделяются на две категории: карты, хранящие информацию, и смарт-карты (smart card — «умная» карта). Карты, хранящие информацию, содержат небольшое количество памяти (как правило, менее 1 Кбайт), использующей технологию EEPROM (Electrically Erasable Programmable Read-Only Memory — электрически стираемое программируемое ПЗУ). На такой карте нет центрального процессора, поэтому сохраняемое значение должно изменяться внешним центральным процессором (в считывающем устройстве). Смарт-карты обладают, как правило, 8-разрядным центральным процессором, работающим с тактовой частотой 4 МГц, 16 Кбайт ПЗУ, 4 Кбайт EEPROM, 512 байт временной оперативной памяти и каналом связи со скоростью 960 бит/с для обмена данными с устройством чтения.

Преимуществом смарт-карт перед кредитными и дебетными картами является то, что для использования смарт-карт не требуется соединения с банком в режиме on-line.

6.5.4 С использованием биометрических данных

Этот метод аутентификации основан на измерении физических характеристик пользователя, которые трудно подделать. Они называются биометрическими параметрами. Например, для идентификации пользователя может использоваться специальное устройство считывания отпечатков пальцев или тембра голоса.

Работа типичной биометрической системы состоит из двух этапов: внесение пользователя в список и идентификация. Во время первого этапа характеристики пользователя измеряются и оцифровываются. Затем извлекаются существенные особенности, которые сохраняются в записи, ассоциированной с пользователем. Второй этап процесса представляет собой идентификацию. Пользователь вводит регистрационное имя. Затем система снова производит замеры. Если новые значения совпадают с хранящимися в записи, регистрация разрешается, в противном случае в регистрации пользователю отказывается.

Измеряемые характеристики должны отличаться у различных пользователей в достаточно широких пределах, чтобы система могла безошибочно различать разных людей. Кроме того, эти характеристики не должны сильно изменяться со вре-

менем. Поскольку последующие измерения никогда точно не совпадут с первоначальными, разработчики такой системы должны решить, насколько точным должно быть сходство.

6.6 Атаки системы изнутри

Зарегистрировавшись на компьютере, взломщик может начать причинение ущерба. Если на компьютере установлена надежная система безопасности, возможно, взломщик сможет навредить только тому пользователю, чей пароль он взломал, но часто начальная регистрация может использоваться в качестве ступеньки для последующего взлома других учетных записей.

6.6.1 Троянские кони

Одним из давно известных вариантов атаки изнутри является троянский конь, представляющий собой невинную с виду программу, содержащую процедуру, выполняющую неожиданные и нежелательные функции. Этими функциями могут быть удаление или шифрование файлов пользователя, копирование их туда, где их впоследствии может получить взломщик, или даже отсылка их взломщику или во временное укромное место по электронной почте или с помощью протокола FTP.

Чтобы троянский конь заработал, нужно, чтобы программа, содержащая его, была запущена. Один способ состоит в бесплатном распространении такой программы через Интернет под видом новой игры, проигрывателя MP3, «специальной» программы для просмотра порнографии и т. д., лишь бы привлечь внимание и поощрить загрузку программы. При запуске программы вызывается процедура троянского коня, которая может выполнять любые действия в пределах полномочий запустившего ее пользователя (например, удалять файлы, устанавливать сетевые соединения и т. д.). Тактика применения троянского коня позволяет обойтись без взлома компьютера жертвы.

6.6.2 Фальшивые программы регистрации

В чем-то схожа с троянскими конями жульническая схема с фальшивой регистрацией. Эта схема работает следующим образом. Обычно при регистрации в системе пользователь видит экран регистрации. Когда пользователь садится за терминал и вводит свое регистрационное имя, система спрашивает у него пароль. Если пароль верен, пользователю разрешается вход в систему и оболочка запускается.

Теперь рассмотрим следующий сценарий. Злоумышленник пишет программу, изображающую экран регистрации. Она выглядит в точности как настоящее окно, предлагающее пользователю зарегистрироваться. Теперь злоумышленник отходит в сторонку и наблюдает за происходящим с безопасного расстояния. Когда пользователь садится за терминал и набирает имя, программа в ответ запрашивает пароль. Когда имя и пароль получены, они записываются в файл, после чего фальшивая программа регистрации посылает сигнал уничтожения собственной оболочки. В результате этого действия сеанс работы злоумышленника на этом терминале завершается и запускается настоящая процедура регистрации. Пользователь при

этом полагает, что он неверно ввел пароль и просто регистрируется еще раз. На этот раз все проходит успешно. Но злоумышленнику таким образом удастся получить имя и пароль. Зарегистрировавшись на нескольких терминалах и запустив на них свою обманывающую пользователей программу, он может за один день собрать много паролей.

6.6.3 Логические бомбы

Сегодня, когда мобильность наемных работников значительно увеличилась, появилась еще одна разновидность атаки системы изнутри, называемая логической бомбой. Логическая бомба представляет собой программу, написанную одним из сотрудников компании и тайно установленную в операционную систему. До тех пор пока программист каждый день входит в систему под своим именем и паролем, эта программа не предпринимает никаких действий. Однако если программиста внезапно увольняют и физически удаляют из помещения без предупреждения, то на следующий день (или на следующую неделю) логическая бомба, не получив своего ежедневного пароля «взрывается».

Взрыв логической бомбы может заключаться в форматировании жесткого диска, удалении файлов в случайном порядке, осуществлении сложно обнаруживаемых изменений в ключевых программах или шифровании важных файлов. В последнем случае компания оказывается перед сложным выбором: вызвать полицию (в результате чего много месяцев спустя злоумышленника, возможно, арестуют и признают виновным, но файлы уже не будут восстановлены) или сдаться шантажисту и снова нанять на работу этого программиста в качестве «консультанта» с астрономическим окладом для восстановления системы (и надеяться, что он при этом не заложит новые логические бомбы).

6.6.4 Потайные двери

Еще один способ создания дыры в системе безопасности изнутри называется потайной дверью. Для этого в систему системным программистом внедряется специальная программа, позволяющая обойти нормальную процедуру проверки. Например, программист может добавить к программе регистрации кусок программы, пропускающий в систему пользователя с определенным именем, независимо от того, что содержится в файле паролей.

6.7 Атаки системы извне

В последнее время, с распространением Интернета и локальных сетей, все большую угрозу для компьютеров представляют атаки снаружи. Компьютер, подключенный к сети, может быть атакован по этой сети с удаленного компьютера. Почти во всех случаях такая атака состоит из передачи по сети на атакуемую машину некоторой программы, при выполнении которой атакуемой машине наносится ущерб. По мере того как количество подключенных к нету компьютеров продолжает увеличиваться, опасность подобных атак также растет.

В последнее время появляются сообщения об атаке компьютеров каким-либо вирусом или червем. Вирусы и черви представляют главную проблему безопасности для отдельных пользователей и компаний.

Что такое вирус? В двух словах, вирус - это программа, которая может размножаться, присоединяя свой код к другой программе, что напоминает размножение биологических вирусов. Кроме того, вирус может выполнять и другие функции. Черви напоминают вирусов, но размножаются сами.

6.7.1 Как вирус причиняет ущерб

Поскольку вирус — это программа, он может делать то, что может программа. Например, он может выводить на экран сообщение или изображение, воспроизводить звуки или выполнять другие безвредные действия. К сожалению, он также может удалять, модифицировать, уничтожать или воровать файлы (передавая их кому-либо по электронной почте). Шантаж тоже возможен. Например, вирус шифрует все файлы на жестком диске жертвы, после чего выводит сообщение где предлагается перечислить денежные средства на определенный счет. Кроме того, вирус может сделать невозможным использование компьютера во время своей работы. Обычно для этого вирус поедает ресурсы компьютера, например процессорное время, или заполняет жесткий диск всяким мусором.

Что еще хуже, вирус может повредить аппаратное обеспечение компьютера. Многие современные компьютеры содержат подсистему ввода-вывода BIOS во флэш-ПЗУ, содержимое которого может программно изменяться (чтобы проще было обновлять BIOS). Вирус может записать в BIOS случайный мусор, после чего компьютер перестанет загружаться.

6.7.2 Как работает вирус

Автор вируса создает свое творение, после чего аккуратно вставляет его в программу на собственном компьютере. Затем инфицированная программа распространяется, возможно, с помощью опубликования ее в виде свободно распространяемой программы через Интернет. Эта программа может представлять собой занимательную новую игру, пиратскую версию коммерческого программного продукта или еще что-либо подобное, вызывающее интерес у публики. Затем пользователи начинают загружать программу на свои компьютеры.

После запуска программы вирус, как правило, начинает с того, что заражает другие программы на этой машине, после чего выполняет свою «полезную» нагрузку, то есть запускает ту часть программы, для которой и писался вирус. Во многих случаях эта программа может не запускаться, пока не наступит определенная дата или пока вирус гарантированно не распространится на большое число компьютеров.

6.7.3 Разновидности вирусов

Вирусы-компаньоны не заражают программу, а запускаются вместо какой-либо программы. Например, вариант атаки с использованием рабочего стола Windows, на котором расположены ярлыки программ. Вирус может подменить

путь, содержащийся в ярлыке, так, чтобы тот указывал не на программу, а на вирус. Когда пользователь щелкает дважды мышью на пиктограмме, запускается вирус. Закончив свое черное дело, вирус запускает оригинальную программу.

Вирусы, заражающие исполняемые файлы. Простейший вид таких вирусов просто записывает себя поверх исполняемой программы. Такие вирусы называются **перезаписывающими** вирусами. С точки зрения автора вируса, недостаток перезаписывающего вируса заключается в том, что его очень легко обнаружить. Инфицированная программа сможет распространить вирус, заразив еще несколько файлов, но она не выполнит то, что должна выполнять, и пользователь это мгновенно заметит. Соответственно, большинство вирусов прицепляются к программам, позволяя им нормально выполняться после того, как вирус выполнит свое черное дело. Такие вирусы называют **паразитическими** вирусами.

Резидентные вирусы — при запуске зараженной программы запускается вирус, который не передает управление настоящей программе и будучи загруженными в память, остается там навсегда.

Вирусы, поражающие загрузочный сектор. При включении большинства компьютеров BIOS считывает главную загрузочную запись с начала загрузочного диска в оперативную память и исполняет ее. Эта программа находит активный раздел диска, считывает его первый (загрузочный) сектор и исполняет его. Затем эта программа загружает либо операционную систему, либо загрузчик операционной системы. К сожалению, уже много лет назад кому-то пришла в голову идея создать вирус, перезаписывающий главную загрузочную запись или загрузочный сектор, последствия реализации которой оказались разрушительными. Такие вирусы очень широко распространены. Как правило, вирус, поражающий загрузочный сектор (или главную загрузочную запись), сначала копирует исходное содержимое загрузочного сектора в какое-либо безопасное место на диске, что позволяет ему загружать операционную систему после того, как он закончит свои дела.

Вирусы драйверов устройств инфицируют драйверы устройств. В системе Windows и в некоторых UNIX-системах драйверы устройств представляют собой просто исполняемые файлы на диске, загружаемые вместе с операционной системой. Если один из них может быть заражен паразитическим вирусом, этот вирус всегда будет официально загружаться при загрузке системы.

Макровирусы. Многие программы, такие как Word и Excel, позволяют пользователям писать макросы, чтобы группировать некоторые команды, которые позднее можно будет выполнить одним нажатием клавиши. При запуске на выполнение макроса, содержащего в себе макровирус, пользователь собственноручно выпускает вирус на волю. Поскольку макрос может содержать произвольную программу, он может выполнять любые действия, например заражать другие документы Word, удалять файлы и т. д.

Вирусы, заражающие исходные тексты программ. Паразитические вирусы и вирусы, заражающие загрузочные секторы, в высокой степени привязаны к определенной платформе. Документные вирусы в меньшей степени зависят от платформ. Самыми переносимыми вирусами являются вирусы, заражающие исходные тексты программ. В этом случае вирус или вызывающая его функция помещаются в исходном тексте программы до его компиляции.

6.7.4 Как распространяются вирусы

Классический вариант. Когда вирус создан, он помещается в какую-либо программу (как правило, чужую, хотя бывает, что автор вируса заражает им свою программу), после чего зараженная программа распространяется, например помещается на web-сайте бесплатных или оплачиваемых после скачивания программ. Эту программу кто-нибудь скачивает и запускает. Далее может быть несколько вариантов. Во-первых, вирус может заразить несколько файлов на жестком диске в надежде, что жертва решит поделиться этими файлами со своими друзьями. Он также может попытаться заразить загрузочный сектор жесткого диска. Как только загрузочный сектор инфицирован, вирус сможет запускаться в резидентном режиме при каждой последующей загрузке компьютера.

Вирус может проверять, подключена ли машина, на которой он работает, к локальной сети, вероятность чего очень высока для компьютеров университета или компании. Затем вирус может начать заражать незащищенные файлы на серверах этой локальной сети.

Часто компьютеры в локальной сети имеют полномочия регистрироваться по Интернету на удаленных машинах или даже выполнять удаленно команды без регистрации. В данном случае вирусы получают еще больше возможностей для распространения. Таким образом, одна ошибка системного администратора может привести к инфицированию компьютеров всей компании.

В последнее время все большее распространение получают вирусы, распространяемые вместе с документами (например, редактора Word). Эти документы рассылаются по электронной почте или публикуются в конференциях и на web-страницах Интернета, как правило, в виде файловых дополнений к письму. Даже люди, которым в голову не приходит запускать программу, присланную им незнакомым человеком, могут не понимать, что, открывая дополнение щелчком мыши, они могут впустить вирус в свою машину. Затем вирус может заглянуть в адресную книгу пользователя и разослать самого себя по всем адресам из этой книги.

6.8 Антивирусы

Вирусы пытаются спрятаться, а пользователи пытаются их обнаружить, играя, таким образом, друг с другом в кошки-мышки. Чтобы вируса не было видно в каталоге, вирусы, хранящие свои данные в отдельном файле (вирусы-компаньоны, вирусы исходных текстов и т. д.), используют возможности операционной системы для создания файлов скрытыми. Вирусы также могут прятаться в необычных и неожиданных местах, таких, как дефектные секторы и списки дефектных секторов, а также в реестре Windows (находящейся в памяти базе данных, в которой программы могут хранить различные текстовые строки). Флэш-ПЗУ и энерго-независимая память CMOS тоже могут использоваться, хотя механизм записи во флэш-ПЗУ достаточно сложен, а CMOS-память слишком мала. И конечно, основным местом, где прячутся вирусы, остаются исполняемые файлы и документы, хранящиеся на жестком диске.

6.8.1 Сканеры

После установки на компьютере пользователя антивирусная программа сканирует все исполняемые файлы на диске, сравнивая их содержимое с хранящимися в ее базе данных штаммами известных вирусов. У большинства компаний, занимающихся разработкой антивирусных программ, есть свои web-сайты, с которых клиенты данных компаний могут скачать описания недавно обнаруженных вирусов в свои базы данных. Если у пользователя 10 000 файлов, а в базе данных хранятся данные о 10 000 вирусах, то чтобы такая программа работала быстро, требуется очень умное программирование.

Так как незначительные мутации уже известных вирусов появляются постоянно, антивирусная программа должна уметь распознавать вирус, несмотря на изменение в нескольких байтах. Однако такой способ поиска не только медленнее точного поиска, но он может привести к ложным тревогам, то есть антивирусная программа будет выдавать предупреждение о незараженных файлах, которые содержат кусок кода, смутно напоминающего вирус.

Если антивирусная программа не нашла на диске вирусов на прошлой неделе, то это не означает, что их на нем нет сейчас, поэтому сканировать диск антивирусной программой следует регулярно.

Помимо проверки исполняемых файлов антивирусный сканер должен проверить главную загрузочную запись, загрузочные секторы, список дефектных блоков, ПЗУ, CMOS-память и т. д.

6.8.2 Проверка целостности

Принципиально другой метод обнаружения вирусов заключается в проверке целостности. Антивирусная программа, работающая подобным образом, сначала сканирует жесткий диск в поисках вирусов. Убедившись, что диск чист, она считает контрольную сумму для каждого исполняемого файла и сохранит список контрольных сумм для всех исполняемых файлов каталога в том же каталоге в файле контрольных сумм. При следующем запуске она пересчитывает все контрольные суммы и проверяет их соответствие данным, хранящимся в файле контрольных сумм. Зараженный файл будет тут же обнаружен по несовпадению контрольной суммы.

6.8.3 Проверка поведения

Другая стратегия, используемая антивирусным программным обеспечением, состоит в проверке поведения программ. При этом антивирусная программа резидентно находится в памяти во время работы компьютера и сама перехватывает все системные вызовы. Идея такого подхода состоит в том, что таким образом антивирусная программа может отслеживать всю активность системы и перехватывать все, что кажется ей подозрительным. Например, ни одна нормальная программа не должна пытаться перезаписать загрузочный сектор, поэтому такие попытки почти наверняка свидетельствуют о деятельности вируса. Изменения содержимого флэш-ПЗУ тоже являются крайне подозрительными.

Но есть множество случаев не столь очевидных. Например, перезапись исполняемого файла необычна, если только это делает не компилятор. Если антивирусная программа обнаруживает подобное действие, она может издать предупреждение в расчете на то, что пользователь знает, должен ли данный файл переписываться. Если редактор Word перезаписывает файл с расширением .doc новым документом, полным макросов, это не обязательно свидетельствует об активности вируса.

6.9 Предохранение от вирусов

Лучше предохраняться, чем потом сожалеть.

Постараться избежать заражения вирусом значительно проще, чем пытаться затем отыскать его на зараженном компьютере.

Что могут сделать пользователи, чтобы избежать заражения вирусом?

Во-первых, выбрать операционную систему, предоставляющую определенный уровень защиты, со строгим разграничением режимов работы ядра и пользователя, а также отдельными паролями регистрации для каждого пользователя и системного администратора. При таких условиях, даже если какой-либо пользователь случайно занесет вирус в систему, этот вирус не сможет заразить системные двоичные файлы.

Во-вторых, устанавливайте только архивированное программное обеспечение, приобретенное у надежного производителя. Загрузка программного обеспечения с web-сайтов весьма рискованна.

В-третьих, приобретите хорошее антивирусное программное обеспечение и используйте его так, как написано в инструкции. Обязательно получайте регулярные обновления с web-сайтов производителя.

В-четвертых, не щелкайте мышью на присоединенных к электронным письмам файлах и скажите, чтобы вам не присылали такие файлы. Электронная почта в виде простого ASCII-текста всегда безопасна, но вложенные файлы могут быть опасны.

В-пятых, архивируйте почаще ключевые файлы на внешних носителях, таких как гибкие диски, CD-R или на магнитной ленте. Храните несколько последовательных версий каждого файла на разных внешних дисках. Тогда, если вы обнаружите вирус, у вас появляется шанс восстановить файлы. Заархивированный вчера уже зараженный файл не поможет, а вот версия недельной давности может оказаться полезной.

Само собой, ни одно из этих предложений не будет восприниматься всерьез, пока действительно большой жареный вирусный петух нас всех не клюнет. Например, обнулив все банковские счета во всем мире. Хотя тогда что-либо предпринимать будет уже поздно.

6.10 Восстановление после вирусной атаки

При обнаружении вируса компьютер следует немедленно остановить, так как резидентный вирус может все еще работать. Компьютер следует перезагрузить

с CD-ROM или гибкого диска, на котором всегда должна быть установлена защита от записи. На этом диске должна содержаться полная операционная система, чтобы не использовать при загрузке жесткий диск с его загрузочным сектором, копией операционной системы и драйверами, которые могут быть инфицированы. Затем с оригинального CD-ROM следует запустить антивирусную программу, так как версия программы, хранящаяся на жестком диске, также может быть заражена.

Антивирусная программа может обнаружить некоторые вирусы и может даже устранить их, но нет никакой гарантии, что она найдет их все. Вероятно, самый надежный метод заключается в том, чтобы сохранить на внешнем носителе все файлы, которые не могут содержать вирусов (например, ASCII и JPEG-файлы).

Те файлы, которые могут содержать вирусы (например, документы редактора Word), должны быть преобразованы в другой формат, который не может содержать вирусы, скажем, в текст ASCII (или, по крайней мере, следует удалить все макросы).

Затем жесткий диск следует переформатировать программой форматирования, загруженной с защищенного от записи гибкого диска или CD-ROM, чтобы гарантировать, что сама программа форматирования не заражена вирусом. Особенно важно, чтобы главная загрузочная запись и загрузочные секторы были полностью стерты. Затем следует переустановить операционную систему с оригинального CD-ROM. Когда вы имеете дело с вирусом, не бойтесь прослыть параноиком.

6.11 Резюме

Компьютерные системы могут подвергаться различным атакам, от атак изнутри до вирусных атак. Многие атаки начинаются с того, что взломщик пытается просто угадать пароль. Для таких атак часто используются словари наиболее употребляемых паролей. Подобные атаки часто бывают удивительно успешными.

Существует много разновидностей атак системы, включая троянского коня, фальшивые программы регистрации, логические бомбы, потайные входы и атаки, использующие переполнение буфера. Кроме того, могут использоваться такие методы, как запрашивание страниц памяти и считывание информации, оставшейся в них, обращение к запрещенным системным вызовам и даже попытки обмана или подкупа сотрудников с целью выведать секретную информацию.

Все большую проблему последнее время представляют собой вирусы. Существует большое разнообразие форм вирусов, включая вирусы, резидентные в памяти, вирусы, заражающие загрузочный сектор диска, а также макровирусы. Использование антивирусной программы, ищущей зараженные файлы по определенным последовательностям байт, полезно, но серьезные вирусы могут зашифровать большую часть своего кода, а также модифицировать остальную часть при реплицировании, что очень сильно усложняет их обнаружение. Некоторое антивирусное программное обеспечение не ищет определенные куски вирусов, а пытается поймать их за подозрительными действиями. Лучшим средством против вирусов является предохранение от вирусов. Поэтому не загружайте и не запускайте программ, авторство которых неизвестно и доверие к которым под вопросом.

В последние годы все большую популярность приобретают мобильные программы, например активные web-страницы. К возможным средствам борьбы с по-

тенциальной опасностью таких программ относится помещение мобильной программы в «песочницу», интерпретация программы, а также запуск только программ, подписанных доверенными производителями.



.....
Контрольные вопросы по главе 6
.....

- 1) Что включает в себя понятие «безопасность»?
- 2) Что представляет собой криптография?
- 3) Какие угрозы несут атаки изнутри?
- 4) Как происходят атаки извне?
- 5) Чем опасны вирусы?

Глава 7

ЯЗЫКИ ПРОГРАММИРОВАНИЯ

«Язык формирует наш способ мышления и определяет то, о чем мы можем мыслить».

Б. Л. Ворф

7.1 Исторический обзор

Первые компьютеры, которые появились в 40-х годах, были ориентированы для военных целей, создавались в единственных экземплярах и выполняли узкоспециализированные задачи. Программы вводились в них посредством соединения гнезд на специальной коммутационной панели, которая управляла последовательностью выполняемых операций. Так, например, программа для компьютера ENIAC (выполнявшего расчет баллистических траекторий) не хранилась в оперативной памяти, а жестко задавалась при помощи шести тысяч перемычек на сорока коммутационных панелях, причем на каждое перепрограммирование ENIAC уходило не менее двух дней. Таким образом, все программирование происходило непосредственно на «железном уровне», а о языках программирования приходилось только мечтать (см. рис 7.1).

В то время элементная база таких компьютеров была очень скудной, памяти хватало буквально для хранения данных, не говоря уже о хранении кодов исполняемых инструкций. Впоследствии с развитием электронной техники стало возможным быстро загружать в память исполняемые программы. Тысячи проводов и коммутационные панели стали уходить в прошлое, их заменили массивы кодов инструкций, которые загружались с помощью перфолент, перфокарт и магнитных барабанов.

Программирование стало происходить в машинных кодах. Теперь, чтобы создавать программу для решения определенной задачи, требовалось знание кодов всех машинных операций, назначение и особенности применения каждой из них, а также помнить адреса конкретных ячеек памяти, хранящие те или иные операции

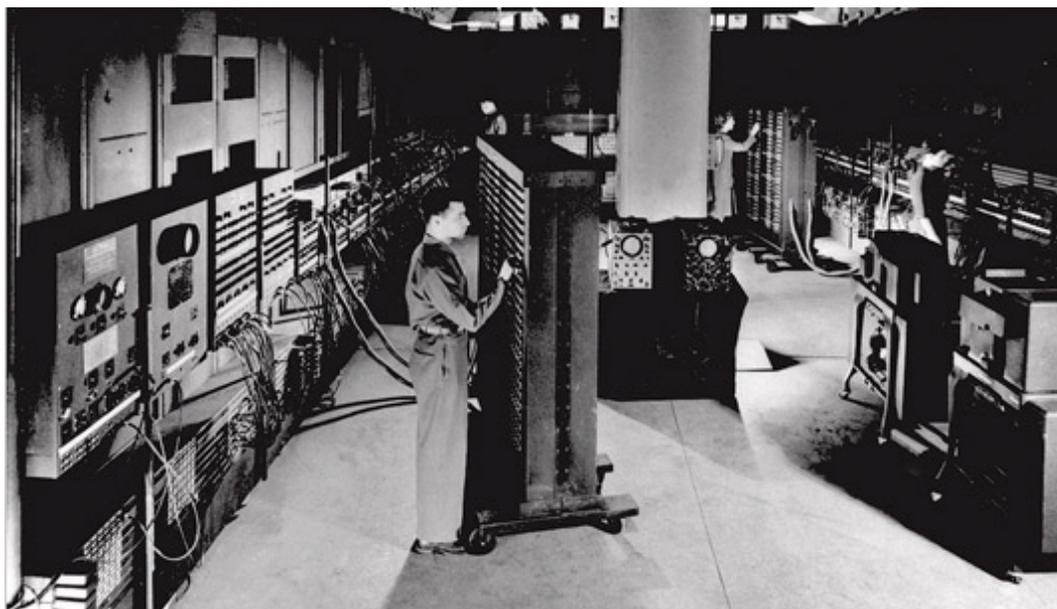


Рис. 7.1 – Создание программы на компьютере ENIAC

и данные. Создаваемые программы представлялись в двоичной форме при помощи нулей и единиц (рис. 7.2).

```
10010001 11100011 11110011 10110110 11101101 11101111
10001001 11011001 10101001 10100000 01100001 00010010
```

Рис. 7.2 – Программа в машинных кодах

Следующим этапом упрощения общения человека с компьютером, было создание языков программирования типа «ассемблер», в которых переменные величины стали изображаться символическими именами, а числовые коды операций были заменены на мнемонические (словесные) обозначения, которые легче запомнить. Например, команда поместить данные в ячейку памяти стала называться MOV, а сложить данные — ADD. Язык программирования приблизился к человеческому языку и удалился от языка машинных команд (рис. 7.3).

```
mov ax, 11011001 ; загружаем число 217 в регистр ax
mov bx, 00100001 ; загружаем число 33 в регистр bx
add ax, bx; ; складываем ax с bx, записываем сумму в ax
```

Рис. 7.3 – Сложение двух чисел на языке ассемблера

Чтобы компьютер мог работать на языке ассемблера, понадобился транслятор — программа, переводящая текст программы на ассемблере в эквивалентные машинные команды (см. рис. 7.4).

В 50-х годах XX века в связи с широким применением компьютеров в различных областях науки и техники возникла серьезная проблема: простые пользователи

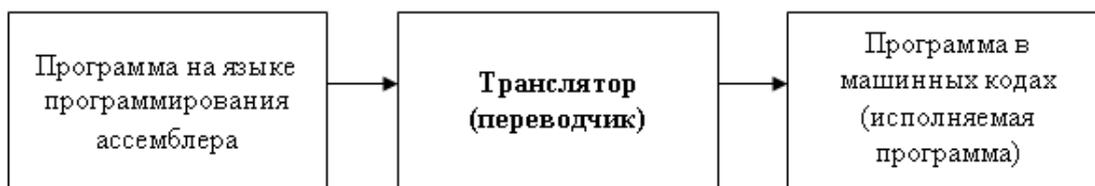


Рис. 7.4 – Назначение транслятора

не могли работать с компьютером из-за сложности языка ассемблера и машинных кодов, а профессиональные программисты были не в состоянии обслужить огромное количество пользователей. Решением данной проблемы явилось создание языков программирования, форма записи программ на которых стала ближе к традиционной математической форме и разговорному языку. Машинные языки, языки ассемблера стали называться языками низкого уровня, языки, позволяющие писать программы на родном для человека языке, — языками высокого уровня.



.....
Низкоуровневый язык программирования (язык программирования низкого уровня) — язык программирования, близкий к программированию непосредственно в машинных кодах.



.....
Высокоуровневый язык программирования (язык программирования высокого уровня) — язык программирования, содержащий смысловые конструкции, понятные для человека, которые в то же время очень длинные и сложные на языке низкого уровня или машинных кодах.

Первым языком высокого уровня стал FORTRAN (Formula Translation — Переводчик Формул), разработанный в 1957 году Джоном Бэкусом. Простота записи математических формул и эффективный исполняемый код, сравнимый с ассемблерным, позволили языку в короткие сроки завоевать популярность среди ученых, инженеров и научных работников. Следующий пример программы из 17 строк (рис. 7.5), написанный на первой версии языка FORTRAN, может быть понятным после непродолжительного «интуитивного» разбора. Если учесть, что данную программу требуется изменить, дополнить новыми возможностями, то станет очевидным, что затраченное время при работе над программой на языке FORTRAN не идет ни в какое сравнение с языками низкого уровня!

Идеи, заложенные в FORTRAN, нашли дальнейшее развитие в других языках программирования. Один за другим стали появляться новые языки.

В 1959 году появился язык COBOL (COmmon Business Oriented Language), предназначенный, в первую очередь, для разработки бизнес-приложений.

В конце 50-х годов был создан язык программирования Algol, который дал начало целому семейству Алгол-подобных языков (важнейший представитель — Pascal).

```
C      THE TPK ALGORITHM
C      FORTRAN I STYLE
      FUNE(T)=SQRTF(ABSF(T))+5.0*T**3
      DIMENSION A(11)
1     FORMAT(6F12.4)
      READ 1,A
      DO 10 J=1,11
      I=11-J
      Y=FUNF(A(I+1))
      IF(400.0-Y)4,8,8
4     PRINT 5,I
5     FORMAT(I10,10H TOO LARGE)
      GOTO 10
8     PRINT 9,I,Y
9     FORMAT(I10,F12.7)
10    CONTINUE
      STOP 52525
```

Рис. 7.5 – Программа на первой версии языка FORTRAN

В 1958 году появился язык Lisp — язык для обработки списков. Получил достаточно широкое распространение в системах искусственного интеллекта.

В 1963 году был создан язык программирования BASIC (Beginners' All-Purpose Symbolic Instruction Code — многоцелевой язык символических инструкций для начинающих). Язык задумывался в первую очередь как средство обучения и как первый изучаемый язык программирования.

В 1964 году корпорация IBM создала язык PL/1, который был призван заменить FORTRAN в большинстве приложений. Язык обладал исключительным богатством синтаксических конструкций. В нем впервые появилась обработка исключительных ситуаций и поддержка параллелизма.

В 1970 году Никлаусом Виртом создан язык программирования Pascal. Язык замечателен тем, что это первый широко распространенный язык для структурного программирования.

В 1972 году Керниганом и Ритчи был создан язык программирования C. Язык создавался для разработки операционной системы UNIX. C часто называют «переносимым ассемблером», так как он позволяет работать с данными практически так же эффективно, как на ассемблере.

В 1972 г. во Франции был создан язык Prolog для решения проблем в области «искусственного интеллекта». Prolog позволяет в формальном виде описывать различные утверждения, логику рассуждений и заставляет ЭВМ давать ответы на заданные вопросы.

В 1979–1980 годах появился язык Ada в результате проекта, предпринятого Министерством обороны США с целью разработать единый язык программирования для встраиваемых систем. Имелись в виду, прежде всего, бортовые системы управления военными объектами (кораблями, самолетами, танками, ракетами, снарядами и т. п.). Язык назван в честь Ады Лавлэйс.

В 1986 году Бьярн Страуструп создал первую версию языка C++, добавив в язык C объектно-ориентированные черты и исправив некоторые ошибки и неудачные решения языка. C++ продолжает совершенствоваться и в настоящее время, так, в 1998 году вышла новая (третья) версия стандарта, содержащая в себе некоторые довольно существенные изменения. Язык стал основой для разработки современных больших и сложных проектов.

В 1995 году в корпорации Sun Microsystems был создан язык Java на основе синтаксиса C и C++. Отличительной особенностью языка является компиляция в код некоей абстрактной машины, для которой затем пишется эмулятор (Java Virtual Machine) для реальных систем.

В 1999–2000 годах в корпорации Microsoft был создан язык C# (Си-шарп). Он в достаточной степени схож с Java (и задумывался как альтернатива последнему), но имеет и отличительные особенности.

Начиная с 90-х годов, в связи с развитием Интернет-технологий, широким распространением высокопроизводительных компьютеров и рядом других факторов получили распространение так называемые скриптовые языки. Эти языки первоначально ориентировались на использование в качестве внутренних управляющих языков во всякого рода сложных системах. Многие из них, однако же, вышли за пределы сферы своего изначального применения и используются ныне в совсем иных областях.

Язык Javascript был создан в компании Netscape Communications в качестве языка для описания сложного поведения веб-страниц. Интерпретируется браузером во время отображения веб-страницы.

Язык PHP — скриптовый язык программирования общего назначения, интенсивно применяющийся для разработки веб-приложений, был разработан в конце 90-х годов.

С течением времени одни языки развивались, приобретали новые черты и остались востребованы, другие утратили свою актуальность и сегодня представляют в лучшем случае чисто теоретический интерес. В значительной степени это связано с такими факторами, как:

- 1) **Возможности компьютеров.** Компьютеры эволюционировали от огромных, медленных и дорогих ламповых машин 50-х годов до современных суперкомпьютеров и микрокомпьютеров. В то же время между аппаратной частью компьютеров и языками программирования появилось промежуточное звено, представляющее собой программное обеспечение операционных систем.
- 2) **Области применения.** В 50-е годы компьютеры использовались лишь в военных целях, науке, деловом мире и промышленности, где высокая стоимость была обоснованной. В настоящее время их применение распространилось на область компьютерных игр, программ для персональных компьютеров, Интернета и вообще на приложения во всех областях человеческой деятельности.
- 3) **Методы программирования.** Структурное строение языка отражает изменяющееся с течением времени наше представление о том, ЧТО является хорошим методом написания большой и сложной программы, а также отражает изменяющуюся со временем среду, в которой осуществляется программирование.

- 4) **Теоретические исследования.** Исследование концептуальных основ разработки и реализации языка с помощью формальных математических методов углубляет понимание сильных и слабых сторон конкретного языка, что отражается на добавлении тех или иных свойств при создании новых языков.
- 5) **Стандартизация.** Необходимость в стандартных языках, которые могут быть легко реализованы в различных компьютерных системах (что позволяет переносить программы с одного компьютера на другой), сильно влияет на эволюцию принципов разработки языков программирования.

В следующей таблице кратко описаны факторы, оказавшие важное влияние на развитие языков программирования во второй половине XX столетия.

В таблице (7.1) кратко описаны факторы, оказавшие важное влияние на развитие языков программирования во второй половине XX столетия.

Таблица 7.1 – Развитие языков программирования

Годы	Факторы и новые технологии
1951–1955	Аппаратная часть: компьютеры на электронных лампах. Методы: языки ассемблера; основные концепции; подпрограммы; структуры данных. Языки: экспериментальное исследование компиляторов выражений
1956–1960	Аппаратная часть: запоминающие устройства на магнитных лентах; память на сердечниках, схемы на транзисторах. Методы: ранние технологии компилирования; оптимизация кода; методы динамического распределения памяти. Языки: FORTRAN, ALGOL, LISP
1961–1965	Аппаратная часть: запоминающие устройства на магнитных дисках. Методы: мультипрограммные операционные системы, синтаксические компиляторы. Языки: COBOL, BASIC
1966–1970	Аппаратная часть: увеличение размера и быстродействия при уменьшении стоимости; микропрограммирование; интегральные схемы. Методы: системы с разделением времени; оптимизирующие компиляторы; системы написания трансляторов. Языки: FORTRAN 66, COBOL 65, ALGOL 68, PL/1
1971–1975	Аппаратная часть: мини-компьютеры; полупроводниковая память. Методы: верификация программ, структурное программирование. Языки: Pascal, COBOL 74, C, Prolog
продолжение на следующей странице	

Таблица 7.1 – Развитие языков программирования. Продолжение

Годы	Факторы и новые технологии
1976–1980	Аппаратная часть: микрокомпьютеры; распределенные вычисления. Методы: абстракция данных; формальная семантика. Языки: Ada, FORTRAN 77
1981–1985	Аппаратная часть: персональные компьютеры, рабочие станции, видеоигры, локальные вычислительные сети. Методы: объектно-ориентированное программирование; интерактивные среды разработки; синтаксические редакторы. Языки: Turbo Pascal
1986–1990	Аппаратная часть: микрокомпьютеры, Интернет. Методы: клиент-серверные вычисления. Языки: FORTRAN 90, C++
1991–1995	Аппаратная часть: очень быстрые и недорогие рабочие станции; архитектура с массовым параллелизмом; звук, видео, факс. Методы: открытые системы, среды разработки. Языки: Ada 95, PHP
1996–2000	Аппаратная часть: компьютеры — дешевые приспособления; персональный электронный помощник; Всемирная паутина WWW; домашние кабельные сети; большой объем дисковой памяти. Методы: электронная коммерция. Языки: Java, Javascript

7.2 Классификация языков программирования

В настоящее время языки программирования в зависимости от применяемой вычислительной модели делятся на четыре основные группы:

Процедурные языки, которые представляют собой последовательность выполняемых операторов. Если рассматривать состояние ПК как состояние ячеек памяти, то процедурный язык — это последовательность операторов, изменяющих значение одной или нескольких ячеек. К процедурным языкам относятся FORTRAN, C, Ada, Pascal и некоторые другие. Процедурные языки иногда также называются императивными языками.

Апplikативные языки, в основу которых положен функциональный подход. Язык рассматривается с точки зрения нахождения функции, необходимой для перевода памяти ПК из одного состояния в другое. Программа представляет собой набор функций, применяемых к начальным данным, позволяющий получить требуемый результат. К аппlikативным языкам относится язык LISP.

Языки системы правил, называемые также **языками логического программирования**, основываются на определении набора правил, при выполнении которых возможно выполнение определенных действий. Правила могут задаваться в виде

утверждений и в виде таблиц решений. К языкам логического программирования относится язык Prolog.

Объектно-ориентированные языки, основанные на построении объектов как набора данных и операций над ними. Объектно-ориентированные языки объединяют и расширяют возможности, присущие процедурным и аппликативным языкам. К объектно-ориентированным языкам относятся C++, Java.

В настоящее время языки программирования применяются в самых различных областях человеческой деятельности.

Научные вычисления. Здесь по-прежнему доминирует язык FORTRAN. Появились новые версии языка FORTRAN 2003. В то же время с ним успешно конкурируют языки Java, C++. Определенный успех имеет MatLab, являющийся интерпретируемым языком, включающий широкий спектр функций и интерфейсы к другим языкам программирования.

Системное программирование. В этой области задействованы языки C и C++, обеспечивающие очень эффективное выполнение программ и позволяющие программистам получить полный доступ к операционной системе и аппаратной части. Также используется современный вариант языка Basic.

Искусственный интеллект. Наиболее подходящими языками являются языки Lisp и Prolog. Оба языка признаны лучшими для задач поиска оптимального решения.

Интернет-программирование. Здесь в основном популярны языки: PHP, JavaScript, Java, содержащие большой набор для разработки сетевых приложений.

7.3 Системы программирования

С развитием ов программирования совершенствовались и средства разработки программ — от режима командной строки до интегрированной среды проектирования. Такая среда предоставляет удобный графический интерфейс разработки (рис. 7.6) и большой спектр сервисов, включающих управление версиями хранимых данных, утилиты просмотра и управления информацией, библиотеки классов, мастера создания шаблонов приложений и т. п.

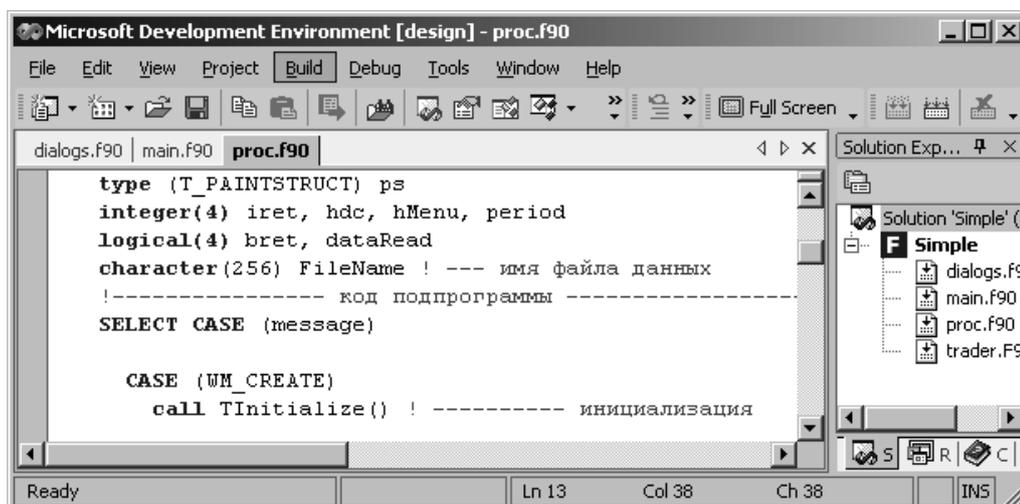


Рис. 7.6 – Среда программирования Microsoft Visual Studio



.....

Системой программирования называется комплекс инструментальных (языковых и программных) средств, которыми пользуется программист при создании новых компьютерных программ. Системы программирования называют также интегрированными средами разработки (англ. IDE, Integrated Development Environment).

.....

Системы программирования могут включать в себя:

- 1) Текстовый редактор, основная функция которого: возможности редактирования текста программы, сравнение программ, поиск и замена текста и т. п.
- 2) Транслятор — программа, преобразующая текст программы в последовательность исполняемых машинных команд. Во время трансляции осуществляется поиск синтаксических ошибок в программе.
- 3) Отладчик — программа, позволяющая контролировать правильность работы программы путем пошагового выполнения, контроля используемой памяти, назначение временных точек остановки, слежение за текущими значениями т. п.
- 4) Библиотеки программ — сборник подпрограмм, используемых для разработки программного обеспечения. Библиотеки существенно ускоряют и упрощают процесс создания новых программ, избавляя программиста заново создавать то, что уже создано десятилетиями назад. Например, часто при разработке компьютерных игр используются библиотеки OpenGL, DirectX.
- 5) Помощник, содержащий набор справочных средств по языку программирования, примеры программ, электронные книги.
- 6) Дополнительные возможности, включающие редакторов создания интерфейса пользователя (меню, диалоги), мастеров, выполняющих поэтапно некоторые стадии разработки программного обеспечения, утилиты, например, для анализа быстродействия создаваемой программы.

Существуют два типа трансляции: компиляция и интерпретация.



.....

Интерпретация — метод выполнения в ЭВМ программы, заданной на языке программирования, при котором каждая инструкция исходной программы переводится и сразу выполняется.

.....



.....

Компиляция — метод выполнения в ЭВМ программы, когда исходная программа переводится на машинный язык, а затем исполняется.

.....

Интерпретация используется в простых языках, где требуется несложная трансляция, или там, где компиляция слишком сложна или даже невозможна. Часто ис-

пользуют оба эти способа совместно: интерпретатор — для отладки и компилятор — для трансляции отлаженной программы.

Как правило, скомпилированные программы выполняются быстрее и не требуют для выполнения дополнительных программ, так как уже переведены на машинный язык. Вместе с тем при каждом изменении текста программы требуется ее перекомпиляция. Кроме того, скомпилированная программа может выполняться только на том же типе компьютеров и, как правило, под той же операционной системой, на которую был рассчитан компилятор. Чтобы создать исполняемый файл для машины другого типа, требуется новая компиляция.

Программа на интерпретируемом языке может быть зачастую запущена на разных типах машин и операционных систем без дополнительных усилий. Однако интерпретируемые программы выполняются заметно медленнее, чем компилируемые, кроме того, они не могут выполняться без дополнительной программы-интерпретатора.

7.4 Какой язык программирования лучше?

Некоторые причины успеха или неуспеха языка могут быть внешними по отношению к самому языку. Так, использование языков COBOL или Ada в Соединенных Штатах для разработки приложений в некоторых предметных областях было регламентировано указом правительства. Широкое распространение таких языков, как LISP и Pascal, объясняется как их использованием в качестве объектов теоретического изучения студентами, специализировавшимися в области разработки языков программирования, так и реальной практической значимостью этих языков.

Несмотря на большое влияние некоторых из перечисленных внешних причин, в конце концов, именно программисты иногда, может быть косвенно, решают, каким языкам жить, а каким нет. Существует множество причин, по которым программисты предпочитают тот или иной язык.

- 1) Ясность, простота и единообразие понятий языка. Язык должен стать помощником программиста задолго до того, как программа начнет записываться на языке программирования. Синтаксис языка влияет на удобство и простоту написания и тестирования программы, а в дальнейшем способствует ее пониманию и модификации. Центральным моментом здесь является удобочитаемость программы. Многие языки содержат такие синтаксические конструкции, которые сами подталкивают к неправильному восприятию программ, поскольку два почти одинаковых оператора на самом деле имеют кардинально различные значения. Хороший язык характеризуется тем, что конструкции, обозначающие различные понятия, должны и выглядеть совершенно по-разному, т. е. семантические отличия должны отражаться в синтаксисе.
- 2) Ортогональность. Термин «ортогональность» означает, что любые возможные комбинации различных языковых конструкций будут осмысленными. Когда конструкции языка ортогональны, язык легче выучить и на нем легче писать программы, поскольку в нем меньше исключений и специальных случаев, требующих запоминания.

- 3) Естественность для приложений. Язык должен предоставлять соответствующие решаемой задаче структуры данных, операции, структуры управления и естественный синтаксис. Язык, соответствующий определенному классу приложений, может сильно облегчить создание отдельных программ в этой области.
- 4) Поддержка абстракций. Язык должен позволять определять структуры данных, типы и операции и поддерживать их как самодостаточные абстракции. В этом случае программист сможет использовать их в других частях программы, зная только их абстрактные свойства и не вникая в их фактическую реализацию. Например, C++ был разработан именно по причине отсутствия этой возможности в более ранней версии C.
- 5) Удобство верификации программы. Существует множество технологий для проверки правильности выполнения программой своих функций. Правильность программы можно доказать с помощью проверки без выполнения (путем чтения текста и исправления ошибок), тестирования путем выполнения с тестовыми входными данными и проверкой выходных результатов и т. д. Основным фактором, влияющим на упрощение проверки программы, — простота семантики и синтаксических структур.
- 6) Среда программирования. Можно составить длинный список разнообразных определяющих факторов, которым должна удовлетворять среда программирования, но возглавляет его, несомненно, требование в ней надежной, эффективной и хорошо документированной реализации языка программирования. Специализированные текстовые редакторы и тестирующие пакеты, которые отражают специфику языка и работы с ним, могут сильно ускорить написание и тестирование программ.
- 7) Переносимость программ. Одним из важных критериев для многих программных проектов является переносимость разработанных программ с компьютера, на котором они были написаны, на другие компьютерные системы. Такие языки, как Ada, FORTRAN, C и Pascal, имеют стандартные определения, позволяющие реализовывать переносимые приложения.
- 8) Стоимость использования. Стоимость использования, конечно, является существенным компонентом оценки языка программирования и складывается из нескольких составляющих: стоимости выполнения программы, стоимости трансляции программы, стоимости создания, тестирования и использования программы, стоимости сопровождения программы. Многочисленные исследования показали, что самую большую часть стоимости программы, используемой в течение нескольких лет, составляет не стоимость начального создания, кодирования и тестирования программы, а стоимость полного жизненного цикла программы, куда входит стоимость как разработки, так и сопровождения программы. Поддержка включает в себя и исправление ошибок, выявленных уже после того, как программа отдана в эксплуатацию, и изменения, которые необходимо внести в программу в связи с обновлением аппаратной части или операционной системы, и усовершенствование и расширение возможностей программы для удовлетворения новых потребностей. Язык, который позволяет без особых проблем

вносить многочисленные изменения и исправления в программу и создавать различные расширения (причем разными программистами и в течение многих лет), окажется в конечном счете более выгодным, чем любой другой.



Контрольные вопросы по главе 7

- 1) Какие факторы оказали влияние на развитие языков программирования?
- 2) Что называют языком программирования высокого уровня?
- 3) Какие возможности предоставляет система программирования?
- 4) Какой из трех языков программирования C++, PHP или Basic следует использовать для написания программ?
- 5) Почему на сегодняшний день используется громадное число языков программирования (свыше тысячи)?

ЗАКЛЮЧЕНИЕ

Развитие информационных технологий стремительно меняют облик современного мира. Информация и знания становятся важным ресурсом и подлинной движущей силой социально-экономического, научного и технологического развития.

Совершенствуются аппаратные средства. Так на смену стационарным компьютерам приходят мобильные устройства, локальные сети вытесняются беспроводными. Накопление информации привели к развитию баз данных, на смену которым приходят хранилища данных.

Компьютер давно уже перестал быть просто устройством, помогающим в вычислениях или проводить свой досуг (игра, музыка, фильмы). Практически любая бытовая техника снабжена своим микропроцессором, будь то стиральная машина или микроволновая печь. А слово «программа» перестало быть сугубо профессиональным термином и знакомо даже домохозяйке.

Поэтому, закончив изучать данный курс, Вы не расстанетесь с информатикой ни на минуту. Вы продолжите изучать ее в других курсах, например, базы данных или программирование. Процесс информатизации не минует и Вашу жизнь. Давно уже существует и продолжает стремительно развиваться киберпространство. В виртуальном мире уже на настоящий момент проводится множество сделок, люди знакомятся друг с другом, делятся новостями.

ЛИТЕРАТУРА

- [1] Острейковский В. А. Информатика: учеб. для студентов техн. и экон. специальностей вузов / В. А. Острейковский. — 5-е изд, стер., М.: Высшая школа, 2009. — 511 с. (Гриф МО РФ).
- [2] Симонович С. В. Информатика для юристов и экономистов: учеб. для вузов / С. В. Симонович. — СПб.: Питер, 2008. — 688 с.
- [3] Симонович С. В. Информатика. Базовый курс: учеб. для техн. вузов / С. В. Симонович. — 2-е изд. — СПб.: Питер, 2009. — 640 с. (Гриф МО РФ).
- [4] Соболев Б. В. Информатика: учеб. / Б. В. Соболев [и др.]. — 4-е изд., перераб. и доп. — Ростов н/Д: Феникс, 2009. 446 с. (Серия Высшее образование).
- [5] Степанов А. Н. Информатика: учеб. для вузов / А. Н. Степанов. — 5-е изд., испр. и доп. — СПб.: Питер, 2008. — 768 с. (Гриф МО РФ).
- [6] Сенченко П. В. Организация баз данных: учеб. пособие / П. В. Сенченко — Томск: ТМЦ ДО, 2002 — 154 с.
- [7] Карпова Т. С. Базы данных: модели, разработка, реализация / Т. С. Карпова — СПб.: Питер, 2002 — 304 с.
- [8] Романова Ю. Д. Информатика и информационные технологии: учеб. пособие / Ю. Д. Романова [и др.]. — 3-е изд., перераб. и доп. — М.: Эксмо, 2008. — 592 с.
- [9] Леонтьев В. Новейшая энциклопедия персонального компьютера 2005 / В. Леонтьев — Москва: Олма-пресс образование, 2005. — 800 с.
- [10] Михайлов А. И. Исследования по информатике в СССР (1968–1977 гг.) / А. И. Михайлов, А. И. Черный, Р. С. Гиляревский // Научно-техническая информация. Сер. 2. — 1977. — №11–12. — С. 13.
- [11] Темников Ф. Е. Информатика // Известия ВУЗов. — 1963. — №11.
- [12] Михайлов А. И. Информатика / А. И. Михайлов, А. И. Черный, Р. С. Гиляревский // БСЭ. — М.: Советская энциклопедия, 1972. — Т. 10. — С. 348–349.

-
- [13] Пратт Т. Языки программирования. Разработка и реализация / Т.Пратт, М.Зелковец. — СПб.: Питер, 2002. — 688 с.
- [14] Таненбаум Э. Современные операционные системы / Э.Таненбаум. — СПб.: Питер, 2002. — 1040 с.

ГЛОССАРИЙ

3G — это мобильная связь 3-го поколения.

BIOS (Basic Input/Output System) см. базовая система ввода/вывода.

Dial-Up-технология — доступ в сеть Интернет по коммутируемой телефонной линии с помощью модема.

IP-адрес — идентификационный номер компьютера, подключенного к сети Интернет.

Software — см. программное обеспечение.

URL-адрес — адрес любого ресурса в Интернете с указанием того, с помощью какого протокола к нему следует обращаться.

Wi-Fi-интернет — современная беспроводная технология подключения компьютера к сети.

Адекватность информации — степень соответствия информации, полученной потребителем, тому, что автор вложил в ее содержание (то есть в данные).

Адресное пространство — список адресов данных исполняемой программы в памяти компьютера.

Актуальность информации — свойство, характеризующие степень ее соответствия текущему моменту времени.

Антивирус — программа для обнаружения компьютерных вирусов, а также нежелательных (считающихся вредоносными) программ вообще и восстановления зараженных (модифицированных) такими программами файлов, а также для профилактики — предотвращения заражения (модификации) файлов или операционной системы вредоносным кодом.

Арифметико-логическое устройство (АЛУ) — часть процессора, выполняющая команды.

Архиватор — служебная программа, позволяющая за счет применения специальных методов «упаковки» сжимать данные, т. е. создавать файлы меньшего размера, а также объединять несколько файлов в один архивный файл.

Архитектура компьютера — описание компьютера на некотором общем уровне, включающее описание пользовательских возможностей программирования, системы команд, системы адресации, организации памяти и т.д. Архитектура определя-

ет принципы действия, информационные связи и взаимное соединение основных логических узлов компьютера: процессора, оперативного ЗУ, внешних ЗУ и периферийных устройств. Общность архитектуры разных компьютеров обеспечивает их совместимость с точки зрения пользователя.

Ассемблер — язык программирования низкого уровня, в котором машинные команды заменены символьными именами

Атрибут — столбец таблицы.

Аудиоадаптер (Sound Blaster или звуковая плата) — специальная электронная плата, которая позволяет записывать звук, воспроизводить его и создавать программными средствами с помощью микрофона, наушников, динамиков, встроенного синтезатора и другого оборудования.

Аутентификация — проверка принадлежности субъекту доступа предъявленного им идентификатора; подтверждение подлинности.

База данных — совокупность взаимосвязанных данных, совместно хранимых в одном или нескольких компьютерных файлах.

Базовая система ввода/вывода — комплекс программ, который размещается в постоянном запоминающем устройстве материнской платы и отвечает за управление всеми ее компонентами.

Базовые протоколы — протоколы, отвечающие за физическую пересылку сообщений между компьютерами в сети Интернет.

Биоинформатика — направление информатики, в котором изучаются общие закономерности и особенности протекания информационных процессов в объектах биосферы (живых организмах и растениях).

Блез Паскаль (годы жизни 1623–1662 гг.) — сконструировал первый механический калькулятор — это был первый шаг человечества к электронному цифровому компьютеру сегодняшнего дня. Этот человек известен как один из наиболее выдающихся физиков и математиков своего времени — создавший теорию вероятностей, а также как один из самых больших мистических авторов в христианской литературе.

Видеоадаптер — электронная плата, которая обрабатывает видеоданные (текст и графику) и управляет работой дисплея.

Виртуальная память — метод управления памятью, когда часть адресов исполняемых программ хранится в оперативной памяти, а часть на диске и при необходимости они меняются местами.

Витая пара — это два перевитых изолированных медных провода.

Выделенный сервер — такой компьютер, который функционирует только как сервер и не используется в качестве клиента или рабочей станции.

Вычислительная техника — совокупность технических и математических средств, методов и приемов, используемых для облегчения и ускорения решения трудоемких задач, связанных с обработкой информации, в частности числовой, путем частичной или полной автоматизации вычислительного процесса; отрасль техники, занимающаяся разработкой, изготовлением и эксплуатацией вычислительных машин.

Глобальные сети — сети, охватывающие значительную территорию, часто целую страну или континент.

Готфрид Лейбниц (годы жизни 1646–1716 гг.) — создатель арифметической машины; сделал вклад в символическую логику, сформулировал принципиальные свойства логического сложения и умножения, отрицания, тождества; также видел преимущество двоичной системы счисления в приведении требуемых арифметических действий к самой простой форме.

Данные — представление информации в некотором формализованном виде, пригодном для передачи и обработки.

Дерево каталогов — иерархическая структура вложенных друг в друга каталогов.

Джон Маучли и *Джон Эккерт* — построили первый действующий электронный цифровой компьютер с названием ENIAC (Electronic Numerical Integrator And Computer). Он был создан на основе вакуумно-ламповой технологии, что обеспечило повышение быстродействия.

Домен — множество допустимых значений атрибута определенного типа.

Достоверность информации — свойство информации не иметь скрытых ошибок, т. е. это характеристика ее неискаженности.

Доступность информации — обобщенное свойство, характеризующие доступность данных и доступность информационных методов, необходимых для воспроизведения этих данных.

Драйвер — специальная системная программа для управления каждым устройством ввода/вывода, подключенным к компьютеру.

Жесткий диск (винчестер) — предназначен для постоянного хранения информации, используемой при работе с компьютером: программ операционной системы, часто используемых пакетов прикладных программ, текстовых документов, файлов базы данных и др.

Злоумышленники — лица, неумышленно, а чаще умышленно создающие угрозы информационной безопасности.

Интегральные схемы (чипы) — электронные схемы, содержащие на одной пластинке и транзисторы, и все необходимые соединения между ними.

Интерпретация — метод выполнения программы, при котором каждая инструкция исходной программы переводится в машинный язык и сразу выполняется.

Интерфейс — стандарт присоединения компонентов к системе. В качестве такового служат разъемы, наборы микросхем, генерирующих стандартные сигналы, стандартный программный код.

Информатика — основанная на использовании компьютерной техники дисциплина, изучающая структуру и общие свойства информации, а также закономерности и методы ее создания, хранения, поиска, преобразования, передачи и применения в различных сферах человеческой деятельности.

Информационная безопасность — качество объекта (данные, информационная система, информационный ресурс и т.д.), а так же деятельность, направленная на обеспечение защищенности объекта.

Информация — продукт взаимодействия данных и методов их обработки, рассмотренный в контексте этого взаимодействия.

Искусственный интеллект — наука и технология создания интеллектуальных машин, особенно интеллектуальных компьютерных программ. ИИ связан со сходной задачей использования компьютеров для понимания человеческого интеллекта, но не обязательно ограничивается биологически правдоподобными методами.

Каталог — файл, в котором хранятся данные о местоположении на диске других каталогов и файлов, виртуально вложенных в него.

Кибернетика — наука об общих закономерностях процессов управления и передачи информации в машинах, живых организмах и обществе.

Кластер — минимальная единица информации в рамках файловой системы.

Команда — элементарная инструкция, предписывающая компьютеру выполнить ту или иную операцию.

Коммуникационные узлы — модемы, повторители, коммутаторы, маршрутизаторы, шлюзы и др.

Компиляция — метод выполнения программы, когда исходная программа переводится на машинный язык, а затем исполняется.

Компьютер (англ. computer — «вычислитель»), ЭВМ (электронная вычислительная машина) — машина для проведения вычислений, а также приема, переработки, хранения и выдачи информации по заранее определенному алгоритму (компьютерной программе).

Контекст — логический (нематериальный) компонент, который влияет на содержание информации и обобщает условия и цели ее получения.

Контроллер — устройство, которое связывает периферийное оборудование или каналы связи с центральным процессором, освобождая процессор от непосредственного управления функционированием данного оборудования.

Контроллеры — устройства управления внешними устройствами.

Конфигурация ПК — полное описание набора и характеристик устройств, составляющих данных компьютер.

Корневой каталог — основной каталог, содержащий все остальные каталоги и файлы.

Кортеж — строка таблицы.

Кэш (англ. cache), или сверхоперативная память — ЗУ небольшого объема, которое используется при обмене данными между микропроцессором и оперативной памятью для компенсации разницы в скорости обработки информации процессором и несколько менее быстродействующей оперативной памятью.

Логическая бомба — программа, которая запускается при определенных временных или информационных условиях для осуществления вредоносных действий.

Локальные сети — сети, охватывающие ограниченную территорию и размещающиеся, как правило, в одном здании или на территории какой-либо организации.

Машинные коды — система команд конкретной ЭВМ.

Модель — объект или явление, аналогичные, то есть в достаточной степени повторяющие свойства моделируемого объекта или явления, существенные для целей

конкретного моделирования и опускающие несущественные свойства, в которых они могут отличаться от прототипа.

Модем — устройство, позволяющее компьютеру выходить на связь с другим компьютером посредством телефонных линий.

Нарушение конфиденциальности данных — действие (или бездействие) лиц повлекшее нарушение режима секретности данных.

Нарушение целостности данных — порча (умышленная или неумышленная) или подделка данных.

Непозиционная система счисления — система, в которой значение символа (цифры) не зависит от его положения в числе.

Оболочка системы — см. файловый менеджер.

Общественная информатика — изучение взаимоотношения процессов информатизации и человека.

Объективность информации — характеристика информации, выражающая степень ее соответствия реальной жизни.

Одноранговая сеть — сеть, в которой все компьютеры равноправны, каждый компьютер функционирует и как клиент, и как сервер.

Оперативная память (ОЗУ, англ. RAM, Random Access Memory — память с произвольным доступом) — это быстрое запоминающее устройство, непосредственно связанное с процессором и предназначенное для записи, считывания и хранения выполняемых программ и данных, обрабатываемых этими программами. Оперативная память используется только для временного хранения данных и программ, так как при выключении машины выключается все, что находилось в ОЗУ, пропадает. Доступ к элементам оперативной памяти прямой — это означает, что каждый байт памяти имеет свой индивидуальный адрес.

Операционная система — комплекс системных программ для организации взаимодействия пользователя с компьютером, управления другими программами и устройствами компьютера.

Отладчик — программа, позволяющая контролировать правильность работы программы.

Память компьютера построена из двоичных запоминающих элементов — битов, объединенных в группы по 8 битов, которые называются байтами. (Единицы измерения памяти совпадают с единицами измерения информации.) Все байты пронумерованы. Номер байта называется его адресом.

Папка — см. каталог.

Первичный ключ — набор атрибутов, значения которых однозначно определяют кортеж отношения.

Персональные компьютеры (ПК) — микрокомпьютеры универсального назначения, рассчитанные на одного пользователя и управляемые одним человеком.

Позиционная система счисления — система счисления, в которой значение символа зависит от его положения в ряду цифр, изображающих число.

Полное имя файла — текстовая строка, состоящая из пути к файлу и имени самого файла.

Полнота информации — относительная характеристика, определяющая количество информации, собранной об объекте или явлении.

Потайная дверь — программа или множество связанных программ, которые хакер устанавливает на компьютер жертву для получения доступа в систему в будущем.

Предметная область — множество всех предметов, свойства которых и отношения между которыми рассматриваются в научной теории.

Прикладная информатика — см. техническая информатика.

Прикладные протоколы — протоколы, отвечающие за функционирование специализированных служб.

Принтер — печатающее устройство. Осуществляет вывод из компьютера закодированной информации в виде печатных копий текста или графики.

Провайдер — специальная организация, предоставляющая услуги доступа в Интернет.

Программа — последовательность команд, которую выполняет компьютер в процессе обработки данных при решении какой-либо задачи.

Программирование — процесс подготовки задач для их решения с помощью компьютера.

Программное обеспечение — совокупность программ, необходимых для обработки различных данных на компьютере.

Протокол FTP — протокол, позволяющий получать и передавать файлы с одного компьютера на другой с использованием TCP-соединений.

Протокол http — протокол передачи гипертекстовых документов, по которому обеспечивается доставка документа с Web-сервера.

Процесс — программа в момент ее выполнения.

Путь к файлу — последовательность имен каталогов от корневого до того, в котором содержится файл.

Рабочая станция — компьютер, подключенный к сети и работающий под управлением локальной операционной системы.

Расширение файла — часть имени файла, обычно обозначающая его тип и находящаяся после последней точки.

Регистр представляет собой совокупность триггеров, связанных друг с другом определенным образом общей системой управления. Регистр выполняет функцию кратковременного хранения числа или команды. Над содержимым некоторых регистров специальные электронные схемы могут выполнять некоторые манипуляции. Например, «вырезать» отдельные части команды для последующего их использования или выполнять определенные арифметические операции над числами. Основным элементом регистра является электронная схема, называемая триггером, которая способна хранить одну двоичную цифру (разряд двоичного кода).

Сектор — минимальная единица информации со стороны аппаратного обеспечения.

Серверы сети — выполняют функции управления распределением сетевых ресурсов и предоставления различного рода сервисных услуг.

Сеть — система, позволяющая производить обмен информацией.

Система программирования — комплекс инструментальных средств для создания новых компьютерных программ.

Система счисления — совокупность приемов и правил записи чисел цифровыми знаками.

Сканер — устройство для ввода в компьютер графических изображений. Создает оцифрованное изображение документа и помещает его в память компьютера.

Служебные программы — программы, используемые при работе или техническом обслуживании компьютера для выполнения вспомогательных функций.

Социальная информатика — направление информатики, в котором изучаются общие закономерности информационного взаимодействия в обществе, включая проблемы социальной коммуникации, формирования информационных ресурсов и информационного потенциала общества, информатизации общества, особенностей информационного общества.

Стив Джобс и Стив Возняк — создатели персонального компьютера Apple.

Структура компьютера — это совокупность его функциональных элементов и связей между ними. Элементами могут быть самые различные устройства — от основных логических узлов компьютера до простейших схем. Структура компьютера графически представляется в виде структурных схем, с помощью которых можно дать описание компьютера на любом уровне детализации.

Схема отношения — конечное множество имен атрибутов, которым ставится в соответствие определенный тип данных.

Таблица — способ передачи содержания, заключающийся в организации структуры данных, в которой отдельные элементы помещены в ячейки, каждой из которых сопоставлена пара значений — номер строки и номер колонки. Таким образом, устанавливается смысловая связь между элементами, принадлежащими одному столбцу или одной строке.

Таблица процессов — связанный список структур выполняемых процессов.

Темников Федор Евгеньевич (1906–1993) — доктор технических наук, профессор, крупный ученый, один из основателей российской школы информатики.

Теоретическая информатика — построение и изучение моделей обработки, передачи, приема и использования информации.

Техническая информатика — направление информатики, в котором изучаются принципы и методы функционирования и построения технических средств информатики — вычислительной техники, средств телекоммуникаций, оргтехники, а также прикладные основы создания информационных технологий.

Топология сети — способ организации физических связей компьютеров и других сетевых компонентов.

Транзистор — полупроводниковый прибор.

Транслятор — программа, преобразующая текст программы в последовательность исполняемых машинных команд.

Троянский конь — вредоносная программа распространяемая под видом безвредного ПО.

Угроза безопасности информации — совокупность условий и факторов, создающих потенциальную или реальную опасность, связанную с утечкой информации или несанкционированными, непреднамеренными воздействиями на нее.

Устройство управления (УУ) — часть процессора, выполняющая функции управления устройствами.

Утилиты — см. служебные программы.

Файл — поименованная область данных на носителях информации.

Файловая система — часть операционной системы, работающая с файлами, обеспечивающая хранение данных на дисках и доступ к ним.

Файловый менеджер — служебная программа, предназначенная для разнообразной работы с файлами.

Факс-модем — модем, позволяющий также принимать и посылать факсимильные сообщения.

Харкевич Александр Александрович (1904–1965) — советский ученый, член-корреспондент АН УССР, профессор, один из основателей и первый директор Института проблем передачи информации (ИППИ) АН СССР.

Центральный процессор (CPU, от англ. Central Processing Unit) — основной рабочий компонент компьютера, который выполняет арифметические и логические операции, заданные программой, управляет вычислительным процессом и координирует работу всех устройств компьютера.

Чарльз Бэббидж (годы жизни 1791–1871 гг.) — один из величайших изобретателей XIX века. Он был математиком, инженером и больше всего конструктором компьютеров. В 1822 году он спроектировал разностную машину, рассматриваемую некоторыми как первое автоматическое вычислительное устройство. В 1834 году он начал конструирование своей аналитической машины, но фактически действующей машины построено не было, поэтому его притязания на славу остались в значительной степени только на тщательно разработанных чертежах. Бэббидж добился известности, будучи первым, кто постиг общую концепцию компьютера. Его аналитическая машина предназначалась для решения любых математических задач. Самое важное — машина также предусматривала наличие нескольких особенностей (условной передачи управления, подпрограмм и циклов), что могло бы сделать ее программируемой. Перфокарты, среда передачи данных, которые, в конечном счете, нашли свое место в компьютере, использовались для ввода.

Шифрование с закрытым ключом — способ шифрования, в котором для шифрования и дешифрования применяется один и тот же криптографический ключ.

Шифрование с открытым ключом — система шифрования, при которой открытый ключ передается по открытому (то есть незащищенному, доступному для наблюдения) каналу, и используется для шифрования сообщения.

Язык ассемблера — язык программирования низкого уровня, в котором машинные команды заменены символьными именами.

Язык программирования высокого уровня — язык программирования, содержащий смысловые конструкции понятные для человека, которые в тоже время очень длинные и сложные на языке низкого уровня или машинных кодах. Разрабатываются для быстроты и удобства использования программистом.

Язык программирования низкого уровня — язык программирования, в котором машинные коды запоминаются не как последовательности нулей и единиц, а в виде осмысленных символьных имен. Одно имя может соответствовать целой группе команд.

Предметный указатель

- 3G, 90, 91
- Ada, 135
- Algol, 134
- ARPAnet, 88
- ASCII, 28
- BASIC, 135
- BIOS, 49, 57
- C, 135
- C++, 136
- CMOS, 48, 58
- COBOL, 134
- CP 866, 29
- DNS, 93
- EDSAC, 36
- ENIAC, 36, 132
- FAR Manager, 67
- FORTRAN, 134
- FTP, 93, 94, 96
- GPRS, 90
- HTTP, 93, 94
- ICQ, 96
- IDE, 140
- Internet, 88
- IP-адрес, 93
- ISO-8859, 29
- Java, 136
- Javascript, 136
- Lisp, 135
- Microsoft Office, 75
- NSFnet, 88
- OLE, 66
- Pascal, 135
- PHP, 136
- PL/1, 135
- POP3, 96
- Prolog, 135
- Skype, 97
- SMTP, 93, 96
- Software, 55
- TCP/IP, 88
- Total Commander, 67
- TV-тюнеры, 50
- URL-адрес, 94
- Windows, 65
- WinRAR, 68
- WinZip, 68
- World Wide Web, 89
- WWW, 89, 94
- Адекватность, 13
- Адресное пространство, 62
- Актуальность, 15
- Антивирусы, 127
- Арифметико-логическое устройство, 42, 44
- Архиваторы, 67

- Аудиоадаптер, 50
Аутентификация пользователей, 120
- Биоинформатика, 10
Большая интегральная схема, 37
Буфер обмена, 66
- Видеоадаптер, 50, 51
Видеопамять, 49
Визуализация, 31
Винчестер, 49
Вирус, 125
Вирусы
 драйверов, 126
 заражающие код программ, 126
 компаньоны, 125
 макро, 126
 перезаписывающие, 126
 поражающие загрузочный сектор, 126
 резидентные, 126
Витая пара, 86
Выделенный канал, 91
Вычислительная техника, 11
- ГОСТ 19768-74, 29
Графика
 векторная, 31
 растровая, 30
 трехмерная, 31
Графический
 акселератор, 50
 редактор, 73
Графопостроитель, 53
- Данные, 16
Данные операции, 18
 архивация, 18
 группировка, 18
 защита, 18
 преобразование, 19
 сбор, 18
 сортировка, 18
 транспортировка, 18
 фильтрация, 18
 формализация, 18
Декомпозиция, 31
Дискретизация, 31
 частота, 32
- Домен, 93
Достоверность, 14
Доступность, 15
Драйвер, 63
- Жесткий диск, 49
- Запоминающие устройства, 45
 оперативные, 45
 перепрограммируемые, 46
 постоянные, 46
Злоумышленники, 118
- Интегрированная среда разработки, 140
Интерпретация, 140
Интерфейс, 42
Информатика, 9
 в природе, 11
 общественная, 11
 прикладная, 10
 социальная, 10
 теоретическая, 10
 техническая, 10
Информационные системы, 10
Информация, 11, 12
 кодирование, 19
Искусственный интеллект, 10
Источник, 77
- КОИ-8, 29
КОИ-8P, 29
КОИ-8У, 29
Каталог, 60
Квантование, 32
Кибернетика, 10
Классификация компьютеров, 38
Кластер, 62
Клиент, 18, 92
Коаксиальный кабель, 86
Кодирование, 19
 аналоговое, 19
 схема, 19
 таблично-символьное, 20
 таблично-цифровое, 20
 табличное, 19
 текстовой информации, 28
 цифровое, 19
Коммуникационный узел, 79

- Компакт-диски, 50
- Компиляция, 140
- Компьютер, 34, 41
 - архитектура, 44
 - архитектура классическая, 46
 - архитектура многопроцессорная, 46
 - архитектура с параллельными процессорами, 47
 - карманный, 41
 - конфигурация, 42
 - планшетный, 41
 - устройство, 41
- Контекст, 12
- Контроллер, 37, 46
- Логические бомбы, 124
- Методы обработки информации, 16
 - естественные, 16
 - технические, 16, 17
- Микрокомпьютер, 38, 39
- Микроконтроллер, 39
- Микропроцессор, 37
- Миникомпьютер, 39
- Мобильный интернет, 90
- Модем, 53
- Монитор, 51
 - ЭЛТ, 51
 - жидкокристаллический, 51
- Мэйнфрейм, 39
- Нарушение
 - конфиденциальности данных, 117
 - целостности данных, 118
- Настольный компьютер, 40
- Носители данных, 17
 - динамический диапазон, 18
 - разрешающая способность, 18
- Ноутбук, 40
 - настольный, 41
- Объективность, 14
- Операционная система, 59
 - функции, 59
- Операционные системы
 - встроенные, 64
 - для ПК, 64
 - реального времени, 64
 - серверные, 64
- Оптоволоконный кабель, 87
- Отказ в обслуживании, 118
- Память, 47
 - внешняя, 48
 - кэш, 48
 - оперативная, 48
 - перепрограммируемая, 48
 - постоянная, 48
 - специальная, 48
- Персональный компьютер, 42
- Пиксел, 30, 51
- Пикселизация, 31
- Плоттер, 53
- Поколение
 - второе, 37
 - первое, 36
 - третье, 37
 - четвертое, 37
- Полнота, 14
- Потайные двери, 124
- Приемник, 77
- Принтер, 52
 - лазерный, 52
 - матричный, 52
 - струйный, 52
- Провайдер, 92
- Программа, 55
- Программирование, 10
- Программное обеспечение, 55, 79
 - базовое, 56, 57
 - прикладное, 57, 70
 - системное, 56
- Программы служебные, 57, 66
- Протокол, 92
 - IP, 92
 - TCP, 92
 - базовый, 93
 - прикладной, 93
- Процесс, 62
- Процессор, 37, 42, 45
 - центральный, 47
- Рабочая группа, 82
- Рабочая станция, 40, 79
- Разряд, 21

- Регистр, 43
 - команд, 43
- Регистрация сигналов, 16
- Сверхбольшая интегральная схема, 37
- Свойства информации, 13
- Сенсорный экран, 52
- Сервер, 18, 64, 79, 83, 92
- Сети
 - глобальные, 80, 87
 - комбинированные, 83
 - компьютерные, 77
 - компьютерные классификация, 79
 - компьютерные назначение, 78
 - корпоративные, 81
 - локальные, 80
 - муниципальные, 80
 - на основе выделенного сервера, 82
 - общего пользования, 81
 - одноранговые, 81
 - приватные, 81
 - региональные, 80
 - топология, 83
 - топология звезда, 84
 - топология звезда-кольцо, 86
 - топология звезда-шина, 86
 - топология кольцо, 85
 - топология шина, 83
- Сигнал, 16
- Система программирования, 139
- Система счисления, 20
 - двоичная, 22
 - десятичная, 21
 - непозиционная, 20
 - позиционная, 21
 - позиционная основание, 21
 - шестнадцатеричная, 22
- Сканер, 53
- Сканеры, 128
- Сообщение, 78
- Спутниковый интернет, 91
- Среда передачи, 77
- Средства связи, 79
- Сумматор, 43
- Суперкомпьютер, 40
- Суперминикомпьютер, 39
- Счетчик команд, 43
- Таблица процессов, 63
- Текстовый
 - процессор, 71
 - редактор, 71
- Технология
 - ADSL, 90
 - Dial-Up, 89
 - DSL, 89
 - Wi-Fi, 92
- Тонер, 52
- Троянские кони, 123
- Угрозы, 117
- Узел, 92
- Управляющее устройство, 44
- Ускорители, 50
- Устройство управления, 43
- Утилиты, 66, 69
- Файл, 59
 - полное имя, 60
- Файловая система, 62
- Файловые менеджеры, 66
- Фальшивые программы, 123
- Фрейм-грабберы, 50
- Цифровые подписи, 120
- Частота развертки
 - кадровая, 51
 - строчная, 51
- Шифрование
 - с закрытым ключом, 119
 - с открытым ключом, 120
- Электронная
 - коммерция, 97
 - почта, 95
 - почта адрес, 95
- Языки программирования
 - высокого уровня, 134
 - низкого уровня, 134
 - объектно-ориентированные, 139
 - процедурные, 138

Учебное издание

Тимченко Сергей Викторович
Сметанин Сергей Викторович
Артемов Игорь Леонидович
Гураков Алексей Валерьевич
Абдалова Ольга Ивановна
Мещеряков Павел Сергеевич
Башкиров Павел Александрович
Шульц Денис Сергеевич

ИНФОРМАТИКА

Учебное пособие

Корректор Осипова Е. А.
Компьютерная верстка Хомич С. Л.

Подписано в печать 14.07.11. Формат 60x84/8.
Усл. печ. л. 18,6. Тираж 500 экз. Заказ

Издано в ООО «Эль Контент»
634029, г. Томск, ул. Кузнецова д. 11 оф. 17
Отпечатано в Томском государственном университете
систем управления и радиоэлектроники.
634050, г. Томск, пр. Ленина, 40
Тел. (3822) 533018.