

Ю.Э. Корчагин С.А. Слинчук

ОСНОВЫ ИНФОРМАТИКИ

Часть 1

Учебное пособие



Воронеж 2007

ГОУВПО «Воронежский государственный
технический университет»

Ю.Э. Корчагин С.А. Слинчук

ОСНОВЫ ИНФОРМАТИКИ

Часть 1

Утверждено Редакционно-издательским советом
университета в качестве учебного пособия

Воронеж 2007

УДК 621.3

Корчагин Ю.Э. Основы информатики: учеб. пособие / Ю.Э. Корчагин, С.А. Слинчук. Воронеж: ГОУВПО «Воронежский государственный технический университет, 2007. Ч.1. 119 с.

В учебном пособии приведены теоретические сведения по курсу «Информатика». В первой части пособия большое внимание уделяется описанию теоретических основ, а также истории развития дисциплины «Информатика». Изложены принципы построения современных электронно-вычислительных машин, их классификация. Рассмотрены виды программного обеспечения.

Издание соответствует требованиям Государственного образовательного стандарта высшего профессионального образования по направлению 210300 «Радиотехника», специальности 210302 «Радиотехника», дисциплине «Информатика».

Учебное пособие предназначено для студентов специальности 210302 "Радиотехника" очной формы обучения (в том числе по сокращенной программе), изучающих дисциплину «Информатика».

Табл. 20. Ил.21. Библиогр.: 12 назв.

Научный редактор профессор Г.В. Макаров

Рецензенты: кафедра информатики и вычислительной техники Международного института компьютерных технологий (зав. кафедрой д-р. техн. наук, проф. В.П. Ирхин); канд. физ.-мат. наук Е.В. Литвинов

© Корчагин Ю.Э., Слинчук С.А., 2007

© Оформление. ГОУВПО "Воронежский государственный технический университет", 2007

ВВЕДЕНИЕ

Дисциплина «Информатика» преподается в настоящее время как в школе, так и на любом факультете вуза. Однако вопрос об определении самой информатики как науки, вопрос о предмете ее исследования остаются дискуссионными до сих пор.

Англоязычный аналог названия «информатика» — computer science — трактуется как наука о компьютерах. Довольно широко распространено мнение [1,2], что за понятием ***информатика*** закреплены области знаний, *«связанные с разработкой, созданием, использованием и материально-техническим обслуживанием систем обработки информации»*. К таким системам относятся компьютеры и их программное обеспечение. Следовательно, **информатика** — это основанная на использовании компьютерной техники дисциплина, изучающая структуру и общие свойства информации, а также закономерности и методы её создания, хранения, поиска, преобразования, передачи и применения в различных сферах человеческой деятельности [1].

Существуют также альтернативные мнения, трактующие термин «информатика» значительно шире. Так, например, в работе [3] дано более общее понятие **информатики** как науки «об описании, осмыслении, определении (толковании), представлении, обобщении (формализации) и использовании **знаний** с применением средств вычислительной техники и программ ее работы для получения нового знания в различных сферах деятельности человека». Ключевым словом в приведенном определении является понятие «знания», которое само по себе шире, чем понятие «информация». В таком понимании информатика тесно соприкасается с другими науками, такими как теория познания, лингвистика и др. По мнению автора работы [3] профессора А.А. Красилова, «информатика зародилась на интуитивном уровне с незапамятных времен. Инфор-

матика проявилась при фундаментальных исследованиях Лейбница, когда впервые была высказана мысль о работе с символами как с числами. Информатика вышла из кабинетов ученых с появлением вычислительных машин широкого применения. Информатика в наши дни превратилась в науку, которую осваивают все науки...»

Приведенные определения характеризуют информатику как фундаментальную науку. Бурное развитие информатика получила именно в последние 50 лет. Это связано со стремительным ростом в последние десятилетия возможностей компьютерной техники, развитием информационных сетей, созданием новых информационных технологий. Все это приводит к значительным изменениям во всех сферах общества: в производстве, науке, образовании, медицине и т.д. Поэтому роль информатики в развитии современного общества чрезвычайно велика. И эта роль не ограничивается только ролью фундаментальной науки. Информатику в некотором роде можно считать отраслью народного хозяйства и прикладной дисциплиной.

Действительно, персональный компьютер (ПК) стал в современном обществе инструментом, необходимым в каждой профессии. Не исключением является и профессия радиоинженера. Моделирование работы проектируемых устройств, оптимизация их параметров, подготовка конструкторской документации — это лишь меньшая часть тех задач, которую может решить радиоинженер с помощью ПК. Уверенное владение этим инструментом дает выпускнику серьезные преимущества на рынке труда.

Современному радиоинженеру в своей работе также неизбежно придется встретиться с понятиями информация, алгоритм обработки информации, алгоритм принятия решения, сигнальный процессор, микроконтроллер и т.д. Все больше радиотехнических электронных устройств реализуют с помощью цифровой и микропроцессорной техники, например, на основе сигнальных процессоров. Для того чтобы проектировать подобные устройства, инженер обязательно должен вла-

деть знаниями и навыками, полученными при изучении курса информатики.

Преподавание курса информатики студентам разных специальностей имеет свою специфику. С учетом специфики преподавания информатики студентам радиотехнических специальностей, тематику курса можно условно разделить на несколько частей. В **первой** части, более теоретической, изучаются начальные понятия теории информации. Вводится само понятие «информация», исследуются ее свойства. Во **второй** части курса изучаются основы вычислительной техники, вопросы построения ЭВМ, методы представления, накопления, передачи и обработки информации с помощью ЭВМ. Эту часть принято называть также основами информационных технологий. В **третьей** части курса изучаются вопросы алгоритмизации и программирования.

1. ИНФОРМАЦИЯ И ЗНАНИЯ

1.1. Понятие информации

Слово «информация» с середины XX века стало очень популярным во всех областях человеческой деятельности. Оно берет начало от латинского «*informatio*» — разъяснение, изложение, осведомленность [1,2]. Однако смысл этого слова понятен лишь интуитивно. **Информация** — это совокупность сведений, данных, существующих в форме речи, в виде текста, таблиц, рисунков, чертежей, схем, условных обозначений, либо в каком-либо другом известном или неизвестном человеку виде.

Вполне понятно, что такое определение является слишком расплывчатым и неоднозначным. Возможно, в будущем оно станет более четким, а может и приобретет иную трактовку. На сегодняшний день в различных отраслях человеческой деятельности понятие «информация» наполняется различными смысловыми оттенками [1,2].

В **обыденном** понимании информацией называют все то, что человек способен сформулировать, понять или почувствовать. Это, например, сообщение о чем-либо, полученное из различных источников, осведомление о положении дел и т.п.

В **философском** понимании информацию получает субъект об объекте в процессе познания.

В **технике** под информацией понимают сообщения, передаваемые в форме сигналов между техническими устройствами.

Отдельного внимания заслуживает определение информации в **кибернетике**. Сама кибернетика определяется как «наука о законах и закономерностях управления в живых и технических системах, а также в обществе, на основе обратной информационной связи» [3,4]. Под информацией здесь понимают «ту часть знаний, которая используется для ориентиро-

вания, активного действия, управления, т.е. в целях сохранения, совершенствования, развития системы» (Н. Винер).

Один из основоположников теории информации американский учёный Клод Шеннон сформулировал **вероятностный** подход к определению информации. Он рассматривает информацию как «раскрытую» неопределенность наших знаний о чем-то. Каждому объекту присуща какая-либо неопределенность. Например, он может находиться в одном из нескольких фиксированных состояний. Упавшая монета находится в одном из двух состояний, игральный кубик — в одном из шести и т.д. В процессе испытания (бросание монеты, кубика) объект приобретает одно из возможных состояний. Следовательно, неопределенность состояния объекта при этом уменьшается. Тем самым испытатель получает какую-то информацию. Степень уменьшения неопределенности при испытании может быть разной. Она зависит от количества возможных состояний, а также от их вероятностей.

Таким образом, емкое и многогранное понятие «информация» пока не поддается четкому определению. Наиболее плодотворным для науки оказалось последнее вероятностное понимание. Именно оно дало возможность ввести количественную меру информации.

1.2. Понятие знания

Человечество в течение всего своего развития накапливало знания. Сначала люди запоминали факты и явления окружающего их мира. Эти накапливаемые в памяти сведения и представляли собой знания. Потребность в обмене накопленными знаниями люди реализовывали через языковой обмен и наскальные зарубки и рисунки.

Большим скачком в развитии человечества явилось появление письменности. Оно повлияло прежде всего на способы передачи и фиксирования знаний. Зафиксированные на твер-

дом носителе знания могли быть распространены в пространстве и времени. Появились первые архивы знаний.

Следующим важным этапом стало изобретение книгопечатания. В результате тиражирования книг знания могли передаваться и распространяться. Но возможности обработки знаний, получения новых знаний на основе анализа существующих по-прежнему зависели от возможностей человека.

Поначалу объем накопленных человечеством знаний изменялся достаточно медленно. Затем процесс получения новых знаний получил заметное ускорение. Считается [5], что общая сумма человеческих знаний до XIX века удваивалась каждые 50 лет, начиная с 1950 г. — каждые 10 лет, с 1970 г. — каждые 5 лет, а с 1990 г. — ежегодно. Появилось противоречие между возрастающим объемом знаний и ограниченными возможностями их переработки. Поэтому развитие средств вычислительной техники и автоматической связи являлось и является актуальной задачей начиная с середины прошлого столетия.

Согласно [4], «знание — это зафиксированная информация в памяти человека, общества или технического устройства, ... формализованное определение знания гласит: знание — это представление фактов, утверждений о фактах или правило получения утверждений о фактах из имеющихся фактов».

Таким образом, понятия информации и знания тесно связаны. Знания — это отсортированная, проверенная на опыте, осмысленная, связанная, систематизированная информация.

1.3. Информационные процессы

Каждый из нас слышал, что информацию (знания) можно собирать, хранить, передавать, обрабатывать и использовать. Наглядный пример информационной системы — зрение. Глаз собирает информацию. В нервной ткани глаза информация сложным образом преобразуется и передается в зрительные отделы головного мозга. Здесь она подвергается дальнейшей

обработке, и результат обработки немедленно используется: к нашим мышцам поступают сигналы. Аналогичным примером может служить система терморегуляции человека.

Процесс конспектирования лекции также состоит из нескольких этапов.

- **Кодирование.** Лектор выражает хранящиеся в памяти знания с помощью слов.
- **Передача.** С помощью звуковых колебаний и световых волн он передает сообщение.
- **Накопление.** Студенты записывают в тетрадь.
- **Обработка.** Расставляются необходимые знаки препинания.

В приведенных примерах присутствуют следующие информационные процессы.

Кодирование — это представление информации (знаний) в виде какой-либо последовательности сигналов. Любая работа с информацией требует ее кодирования. Одну и ту же информацию можно кодировать по-разному. Кодирование сигнала СОС может быть таким — SOS, или таким — · —. Один из наиболее простых и надежных способов — двоичное кодирование. При этом используются всего два сигнала. Пример — азбука Морзе.

Передача информации осуществляется по каналам связи с помощью каких-либо носителей. Например, человеческая речь распространяется в воздухе с помощью звуковых волн, информация от телецентра распространяется тоже в воздухе (хотя может распространяться и в вакууме) с помощью электромагнитных волн и д.т. В любом реальном канале связи обязательно присутствует шум — мешающее воздействие.

Для **накопления** информации (знаний) используют долговременные носители: скалы (наскальные рисунки — хоть и примитивное, но накопление информации), книги (и бумага вообще), виниловые пластинки для накопления звуковой информации, магнитная лента, магнитные диски (гибкие и жёсткие), CD и DVD диски.

Обработка — это внесение изменений в имеющуюся информацию (выполнение арифметических действий, исправление ошибок в сочинении, оформление результатов лабораторного опыта и т.д.)

1.4. Количество информации

Широкое многообразие характерно не только для определения самого понятия “информация”, но и для определения количества информации. Условно все подходы к определению количества информации можно разделить на пять видов: комбинаторный, вероятностный, алгоритмический, семантический, прагматический [6].

Согласно [6], «первые три вида дают количественное определение сложности описываемого объекта или явления. Четвертый — описывает содержательность и новизну передаваемого сообщения для получателя (пользователя) сообщения. Наконец, пятый вид обращает внимание на полезность полученного сообщения для пользователя».

1. Комбинаторный подход

Пусть имеется некоторая физическая система, которая может находиться в одном из n равновероятных состояний. В результате какого-либо испытания или измерения наблюдатель узнает, в каком именно состоянии находится эта система. Тем самым снимается неопределенность состояния системы, а наблюдатель получает информацию. Обозначим через I количество полученной информации. Согласно комбинаторному подходу, предложенному Р. Хартли, количество информации, полученной при испытании, определяется логарифмом общего числа возможных состояний системы n :

$$I = \log(n) . \quad (1)$$

Такое определение является довольно удобным и обладает рядом замечательных свойств, а именно:

- физическая система с одним единственным состоянием не содержит информации, то есть количество информации $I=0$;
- если информация поступает от двух независимых источников информации с n_1 и n_2 числом возможных состояний, то $I(n_1 n_2) = I(n_1) + I(n_2)$;
- чем большее число состояний имеет физическая система, тем большее количество информации она содержит, то есть при $n_1 > n_2$, $I(n_1) > I(n_2)$.

Недостатком комбинаторного подхода является предположение о равновероятности состояний системы. Действительно, на количество информации, получаемой из наблюдения, влияет степень ее неожиданности для получателя. Неожиданность можно охарактеризовать вероятностью состояний системы. Чем меньше эта вероятность, тем сообщение о реализации состояния более неожиданно и более информативно. Сообщение с высокой вероятностью содержит меньше информации, чем сообщение с низкой вероятностью. Следовательно, при определении количества информации необходимо учитывать не только количество разнообразных сообщений, которые можно получить от источника, но и вероятность их получения.

2. Вероятностный подход

Предположим теперь, что состояния системы неравновероятны, а также известны вероятности состояний p_1, p_2, \dots, p_n ,

причем $\sum_{i=1}^n p_i = 1$. Согласно вероятностному подходу, развитому К. Шенноном, вводится понятие энтропии

$$H = H(p_1, p_2, \dots, p_n),$$

которая количественно характеризует неопределенность состояния системы. Количество информации, получаемое при снятии неопределенности, считается равным энтропии H . Шеннон предложил для энтропии следующее выражение:

$$H = -\sum_{i=1}^n p_i \log(p_i). \quad (2)$$

Величина, определенная по формуле (2), обладает следующими свойствами.

Свойство 1. Если одна из вероятностей p_i равна 1, а остальные равны нулю, то энтропия физической системы равна нулю.

Свойство 2. Энтропия максимальна, когда все состояния источника информации равновероятны. Для доказательства рассмотрим функцию

$$H = -\sum_{i=1}^n p_i \log p_i + \lambda \left(\sum_{i=1}^n p_i - 1 \right). \quad (3)$$

Второе слагаемое здесь равно нулю в силу свойств распределения вероятностей. Найдем локальный экстремум энтропии (3) по переменным p_i , $i = \overline{1, n}$. Для этого приравняем нулю частные производные функции (3) по переменным p_i , получим систему уравнений

$$\begin{cases} \log p_1 = \lambda - \log e, \\ \log p_2 = \lambda - \log e, \\ \dots \\ \log p_n = \lambda - \log e. \end{cases}$$

В силу равенства правых частей должны быть равны все левые части, что свидетельствует о равновероятных состояниях физической системы, то есть $p_1 = p_2 = \dots = p_n = 1/n$. Тогда

$$H = -\sum_{i=1}^n \frac{1}{n} \log\left(\frac{1}{n}\right) = \log n.$$

Полученное выражение совпадает с формулой Хартли для количества информации. Следовательно, формула Хартли является частным случаем формулы Шеннона.

Свойство 3. Всякое изменение вероятностей p_1, p_2, \dots, p_n в сторону их выравнивания увеличивает энтропию (без доказательства).

Предложенная Шенноном формула для энтропии позволила ввести единицы измерения количества информации. За единицу информации принимается один бит. Это количество информации, получаемое в результате реализации одного из двух равновероятных событий, например при бросании монеты. Термин "бит" произошел от выражения binary digit, что означает "двоичная цифра", принимающая значение 0 или 1. Следовательно,

$$H = -\sum_{i=1}^2 \frac{1}{2} \log\left(\frac{1}{2}\right) = -\log\left(\frac{1}{2}\right) = 1.$$

Для справедливости этого выражения основание логарифма должно быть равно 2.

Более крупная единица информации — **байт** — равна 8 бит. Проверка присутствия или отсутствия на лекции 24 студентов дает лектору три байта информации (при условии равновероятности и независимости появления и не появления на лекции каждого из них). Еще более крупная единица информации — 1 Кбайт (Кб) — равна 1024 байтам. Далее — 1 Мбайт (Мб) равен 1024 Кбайтам, 1 Гбайт (Гб) равен 1024 Мбайтам, а 1 Тбайт (Тб) равен 1024 Гбайтам.

3. Алгоритмический подход

В рамках алгоритмического подхода [6,7] можно ответить на вопрос «Сколько нужно информации, чтобы воссоздать (описать) объект?». А.Н. Колмогоров сформулировал такую задачу для последовательностей нулей и единиц. Известно, что для объекта, представляющего собой последовательность нулей и единиц, можно ввести меру сложности, используя которую А.Н. Колмогоров предложил алгоритмический подход к определению количества информации.

Этот подход основан на теории алгоритмов и предполагает наличие априорной вероятностной меры на множестве сигналов.

Пусть имеется некоторый алфавит символов X . Последовательность символов этого алфавита называют словом W в алфавите X . Пусть также имеется двоичный алфавит $\{0,1\}$, содержащий лишь два символа. Чтобы слово W представить в алфавите $\{0,1\}$, его необходимо подвергнуть преобразованию или описать в алфавите $\{0,1\}$. Для этого вводят понятие способа описания f . Обозначим через g описанное в алфавите $\{0,1\}$ слово W . Тогда $f(g) = W$. При фиксированном способе описания f длину кратчайшего описания считают мерой сложности слова. Среди алгоритмических способов описания существует оптимальный, который дает более короткие описания, чем остальные способы. Сложность слова при оптимальном способе описания называется колмогоровской сложностью $R(W)$ и определяет количество информации в слове W .

Сведения о семантическом и прагматическом подходах к определению информации можно найти в [6].

2. ИСТОРИЯ РАЗВИТИЯ ЭВМ. ПОКОЛЕНИЯ ЭВМ

2.1. История развития вычислительной техники

Первые ЭВМ появились в середине XX века, хотя с древних времен человек пытался сделать устройства, помогающие ему в выполнении рутинной работы, какой является, например, арифметический счет. Считается, что к созданию первой ЭВМ привело слияние двух направлений развития человеческой мысли. Первоначально эти направления были взаимонезависимы. В дальнейшем их объединение привело к формированию принципов построения ЭВМ, а затем, при наличии надлежащей технической базы, ЭВМ стали реальностью.

Первое направление — стремление к автоматизации арифметического счета. В далекой древности в Средиземноморье существовало «устройство» для счета — абак, который напоминает современные счеты. Необходимо отметить, что арифметические действия выполнялись человеком, абак помогал только запомнить промежуточные и окончательный результаты [5].

Самое первое устройство, автоматизирующее счет, было разработано немецким профессором **Вильгельмом Шиккардом** в 1623 г. [5,8]. Оно было реализовано в единственном экземпляре на базе механических часов и предназначалось для выполнения операции сложения. До наших дней дошли чертежи машины Шиккарда.

В 1642 году **Блез Паскаль** изобрел механическую машину для сложения чисел. В 1673 году машина Паскаля была доработана **Г. Лейбницем**. Арифметическая машина Лейбница могла складывать и перемножать, делить числа и извлекать квадратный корень. С 1820 года начался серийный выпуск механических арифмометров [8].

Вторым направлением развития человеческой мысли, приведшим к построению ЭВМ, был принцип программной

управляемости. Устройства, работающие согласно заранее заданной программе, также существовали с глубокой древности. Ярким примером такого устройства является шарманка. «Программа», несущая информацию о последовательности извлечения звуков, наносилась на вращающийся диск в виде шипов. В эпоху первых буржуазных революций принцип программной управляемости механических устройств был применен к ткацким станкам.

Предвестниками идеи построения ЭВМ были английский математик и экономист **Чарлз Бэббидж** (Charles Babbage) (1791–1871) и его ученица **Ада Лавлейс** (Ada Lovelace). Они впервые сформулировали идею создания программируемой счетной машины и в 1834 г. создали проект «аналитической машины».

В проекте аналитической машины Бэббиджа были упомянуты все основные узлы современной ЭВМ, а также принцип управления с помощью запоминаемой программы.

Ч. Бэббидж выделял в своей машине следующие составные части [5,8]: «склад» для хранения чисел, «мельница» для производства арифметических действий, устройство, управляющее последовательностью выполнения операций, устройства ввода и вывода данных». В современных ЭВМ имеются аналоги упомянутых частей: память, арифметико-логическое устройство, устройство управления.

Однако в силу отсутствия надлежащей технической базы проект аналитической машины остался нереализованным.

Тем не менее работа Ч. Бэббиджа была очень важна, поскольку впервые затрагивались вопросы программирования. Например, был разработан принцип «условной передачи управления» [5].

Для аналитической машины были разработаны также первые программы. Ада Лавлейс в качестве примера составила программу вычисления чисел Бернулли для аналитической машины [5].

Следующим шагом в направлении создания ЭВМ было использование вычислительных машин с перфокартами. Счетно-перфорационная машина **Германа Холлерита** (Herman Hollerith) создана в 1890 г. для обработки данных по переписи населения в США. Машина в качестве носителя информации использовала перфокарты.

Вычислительные машины с перфокартами имели огромный успех. Герман Холлерит стал основателем фирмы Computing Tabulation Company, которую вскоре переименовали в известную всем фирму International Business Machine Corporation (IBM).

Очередным этапом становления ЭВМ стало построение электромеханических вычислительных машин, построенных с использованием реле. Реле как элемент, реализующий два взаимоисключающих состояния, позволяло эффективно использовать двоичную систему счисления. Первая автоматическая электромеханическая вычислительная машина была названа Zuse1 в честь своего создателя немецкого инженера Конрада Цузе (Konrad Zuse).

В 1944 г. была построена еще одна электромеханическая вычислительная машина Mark1. Изобретателем ее был Говард Айкен (Aiken Howard).

Эпоха полностью *электронных* вычислительных машин началась с 40 годов нашего века созданием **Эккерт и Маучли** (John W. Mauchly and J. Presper Eckert, Jr.) в 1945 году в США машины ENIAC. Она содержала 18000 ламп, весила 30 тонн, занимала 100 кв. м площади, потребляла 150 кВт энергии, имела водяное охлаждение, могла делать 5 тыс. операций/сек.

Программирование ENIAC осуществлялось соединением проводами гнезд на коммутационной панели, то есть программа хранилась вне памяти ЭВМ. Память же использовалась для хранения обрабатываемых и промежуточных данных.

Располагать исполняемую программу в памяти ЭВМ совместно с данными впервые предложил **Джон фон Нейман**

(1903—1957). Этот принцип, называемый сейчас одним из «принципов фон Неймана», используется всеми ЭВМ вплоть до настоящего времени.

Согласно [5], «в докладе фон Неймана, посвященном описанию ЭВМ, выделено пять базовых элементов компьютера:

- арифметико-логическое устройство (АЛУ);
- устройство управления (УУ);
- запоминающее устройство (ЗУ);
- система ввода информации;
- система вывода информации.

Описанную структуру ЭВМ принято называть архитектурой фон Неймана».

2.2. Поколения ЭВМ

Процесс совершенствования ЭВМ шел одновременно с развитием электроники. Появление новых электронных компонентов оказывало существенное влияние на элементарную базу ЭВМ, технические характеристики (быстродействие, объем памяти, система команд) и сложность архитектуры. Перечисленные критерии позволяют разделить ЭВМ на поколения. В настоящее время насчитывается четыре поколения ЭВМ, характерные черты которых представлены в табл. 2.1.

Таблица 2.1

Поколения ЭВМ

Поколения	Элементарная база	Быстродействие, оп/сек.	Память	Программное обеспечение
Первое	электронные лампы	до 2000	на электронно-лучевых трубках и ферритовых сердечниках	однопрограммный режим

Продолжение табл. 2.1

Второе	транзисторы	до 10^4 – 10^5	до 150 слов по 50 двоичных разрядов	Появились языки программирования (Фортран, Алгол, Кобол).
Третье	микросхемы	до 10^6 – 10^7	строилась на ИС, объем до 10^5 – 10^6 байт	широкий выбор языков программирования, появились операционные системы, многопрограммный и терминальный режимы
Четвертое	большие и сверхбольшие интегральные схемы	до 10^9	до 10^{10} байт	разнообразие программного обеспечения

Нашими современниками являются ЭВМ четвертого поколения, в рамках которого сформировались два направления — многопроцессорные и персональные ЭВМ. В период развития ЭВМ четвертого поколения появились развитые компьютерные сети, разрабатывалось и разрабатывается разнообразное программное обеспечение.

Современная вычислительная техника развивается очень динамично. По-видимому, не за горами появления ЭВМ пятого поколения. Однако критерии смены поколения в настоящее время являются достаточно расплывчатыми. Некоторые авторы предполагают, что дальнейшее повышение производительности ЭВМ приведет к принципиально иному стилю общения человека и ЭВМ, и это станет критерием смены поколения. Согласно другим источникам, для смены поколения необхо-

дима смена элементарной базы, которая связана с технологией производства современных микросхем.

2.3. История развития отечественной вычислительной техники

Независимо от западных коллег в СССР в предвоенные годы также велась активная работа по созданию ЭВМ. Однако, в связи с Великой отечественной войной российские разработки ЭВМ были прерваны. Работа возобновилась лишь после войны, когда первая ЭВМ ENIAC уже существовала в США. Первая же отечественная ЭВМ МЭСМ (Малая электронная счетная машина), аналогичная ENIAC, была создана в 1951 году под руководством Сергея Алексеевича Лебедева в институте электротехники Академии наук Украины.

Вопросы дальнейшего развития вычислительной техники в СССР хорошо изложены в [5]. По материалам книги [5] отечественные ЭВМ и их краткая характеристика представлены в табл. 2.2.

Таблица 2.2

Отечественные ЭВМ

Название ЭВМ	Поколение	Характеристика
МЭСМ, БЭСМ, БЭСМ-2	1	Создана в институте электротехники Академии наук Украины под руководством С.А. Лебедева
М-1, М-2, М-3	1	Созданы под руководством И.С. Брука
Сетунь	1	Создана в 1953 г. под руководством Н.П. Брусенцова, работала в троичной системе счисления

Продолжение табл. 2.2

Днепр	2	Разработана под руководством академика В.М. Глушкова в конце 50-х годов в Институте кибернетики АН Украины. Первая ЭВМ второго поколения
БЭСМ-6	2	Создана в 1966 году в коллективе С.А. Лебедева. Для своего времени была одной из самых производительных в мире
М-40, М-50	2	Использовалась для расчетов систем противоракетной обороны
Урал-11	2	Применение: инженерно-технические и планово-экономические задачи
Минск-2	2	Применение: инженерные, научные и конструкторские задачи
Минск-22	2	Применение: научно-технические и экономические задачи
М-20	2	Применение: сложные математические задачи
Наири	2	Применение: различные инженерные задачи

Продолжение табл. 2.2

<p>Наири-3, Днепр-2, ЕС-10х0 МИР-2, Наири-2,</p>	<p>3</p>	<p>Первая ЭВМ 3-го поколения «Наири-3» создана в 1970 году</p>
<p>ЕС-10х5, ЕС-10х6, СМ-1420, СМ-1600, СМ-1700,</p>	<p>4</p>	<p>Отечественные ЭВМ четвертого поколения</p>
<p>Электроника МС 0501, Электроника- 85, Искра-226, ЕС-1840, ЕС-1841, ЕС-1842, Нейрон И9.66</p>	<p>4</p>	<p>Персональные ЭВМ четвертого поколения</p>

3. СИСТЕМЫ СЧИСЛЕНИЯ

3.1. Классификация систем счисления

Под системой счисления (СС) понимают способ изображения чисел с помощью символов совместно с правилами выполнения действий над этими числами.

Все системы счисления делятся на позиционные и непозиционные.

В **непозиционных** системах каждая цифра имеет свой вес и ее значение не зависит от положения в числе — от позиции, например, римская система. Скажем, число 76 в этой системе выглядит так:

LXXVI, где $L=50$, $X=10$, $V=5$, $I=1$.

Как видно, цифрами здесь служат латинские символы.

В **позиционных** системах значения цифр зависят от их положения (позиции) в числе.

Так, например, человек привык пользоваться десятичной позиционной системой — числа записываются с помощью 10 цифр. Самая правая цифра целого числа обозначает единицы, левее — десятки, ещё левее — сотни и т.д.

В любой позиционной системе число может быть представлено в виде многочлена. Покажем, как представляют в виде многочлена десятичное целое число:

$4567 = 4000 + 500 + 60 + 7 = 4 \cdot 10^3 + 5 \cdot 10^2 + 6 \cdot 10^1 + 7 \cdot 10^0$,
а теперь с дробью:

$$\begin{aligned} 34,567 &= 30 + 4 + 0,5 + 0,06 + 0,007 = \\ &= 3 \cdot 10^1 + 4 \cdot 10^0 + 5 \cdot 10^{-1} + 6 \cdot 10^{-2} + 7 \cdot 10^{-3}. \end{aligned}$$

Обобщим это представление на случай использования произвольного набора цифр:

$$\begin{aligned}
& a_n a_{n-1} \dots a_1 a_0, a_{-1} a_{-2} \dots a_{-m} = \\
& = a_n p^n + a_{n-1} p^{n-1} + \dots + a_1 p + a_0 + a_{-1} p^{-1} + a_{-2} p^{-2} + \dots + a_{-m} p^{-m} = \\
& = \sum_{j=-m}^n a_j p^j.
\end{aligned}$$

здесь обозначено p — общее количество цифр, используемое в системе счисления, которое называют базой системы. Ни в одной системе нет цифр, равной основанию системы.

База системы счисления — это множество цифр, используемых для записи числа.

Все позиционные системы счисления являются равноправными, однако по историческим причинам человек повседневно использует лишь десятичную систему с основанием, равным 10 и базой 0,1,2,3,4,5,6,7,8,9.

Известно [5], что, чем больше основание СС, тем компактнее запись числа. Например, число, записанное в двоичной системе, более чем в 3 длиннее, чем то же число, записанное в десятичной системе.

Человеку в повседневной жизни привычной пользоваться десятичной системой счисления. Однако, все современные ЭВМ работают с данными, представленными в двоичной системе. Это связано с тем, что технически значительно легче реализовать элементы с двумя различными состояниями, нежели с десятью.

При низкоуровневом программировании (в машинных кодах или на языке Ассемблер) часто используются шестнадцатеричная и восьмеричная СС. Выбор этих систем связан с тем, что перевод чисел между двоичной и шестнадцатеричной и восьмеричной системами осуществляется достаточно просто. Необходимость работы с шестнадцатеричной системой имеется также и в языках высокого уровня. Некоторые библиотечные подпрограммы языков программирования Паскаль, Си и др. требуют задания параметров в шестнадцатеричной системе счисления. В шестнадцатеричном виде представляются также

данные, записанные на жестком диске различными тестовыми и диагностическими программами. Поэтому чтобы найти неисправность или восстановить потерянные данные необходимо уметь работать с различными системами счисления.

3.2. Двоичная система счисления

Представление информации с помощью двоичного кодирования наиболее оптимально для ЭВМ, так как данные в ЭВМ передаются по проводам с помощью двух сигналов "есть напряжение" и "нет напряжения". Поскольку все данные в ЭВМ кодируются числами, то для передачи их по проводам необходимо применять двоичную систему.

Двоичная система имеет основание $p=2$ и базу 0 и 1. То есть, для изображения числа используются только два знака (табл. 3.1).

Таблица 3.1

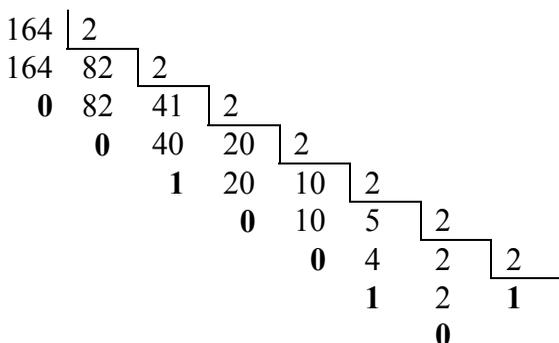
Таблица соответствия двоичной СС десятичной СС

10-я	2-я	10-я	2-я	10-я	2-я	10-я	2-я
1	1	6	110	11	1011	16	10000
2	10	7	111	12	1100	17	10001
3	11	8	1000	13	1101	18	10010
4	100	9	1001	14	1110	19	10011
5	101	10	1010	15	1111	20	10100

Для перевода десятичного числа в двоичную систему отдельно переводят дробную и целую части.

Чтобы перевести **целое число** из 10-ой в 2-ую систему нужно выполнять последовательное деление числа на 2 до тех пор, пока результат не станет меньше 2. Последний результат и остатки от деления, взятые в обратном порядке, дают двоичное число.

Например:



В результате $164_{10}=10100100_2$.

Для перевода **правильной дроби** из 10-системы счисления в 2-ю СС нужно умножить исходную дробь и дробные части получающихся произведений на основание 2, представленное в старой 10-системе. Целые части получающихся произведений дают последовательность цифр, которая является представлением дроби в 2-ой системе счисления.

Например:

$$\begin{aligned}
 0,378 \cdot 2 &= 0,756; \\
 0,756 \cdot 2 &= 1,512; \\
 0,512 \cdot 2 &= 1,024; \\
 0,024 \cdot 2 &= 0,048; \\
 0,048 \cdot 2 &= 0,096; \\
 0,096 \cdot 2 &= 0,192; \\
 0,192 \cdot 2 &= 0,384; \\
 0,384 \cdot 2 &= 0,768; \\
 0,768 \cdot 2 &= 1,536; \\
 0,536 \cdot 2 &= 1,072.
 \end{aligned}$$

В результате $0,378_{10} \approx 0,0110000011_2$. Здесь ограничились 10 знаками после запятой двоичного числа. Для более точного перевода необходимо продолжать умножение дальше и получать больше знаков после запятой. Если в процессе умножения результат равен 1, то дробь переводится в двоичную систему без приближения.

Для перевода числа из двоичной в десятичную систему необходимо разложить число в ряд по основанию системы счисления и посчитать результат.

Например,

$$\begin{aligned} 10100100,101_2 &= 1 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + \\ &\quad + 0 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} = \\ &= 2^2 + 2^5 + 1 \cdot 2^7 + 1 \cdot 2^{-1} + 1 \cdot 2^{-3} = 4 + 32 + 128 = 164,625_{10} \end{aligned}$$

3.3. Системы счисления, родственные двоичной

На ранних этапах развития вычислительной техники программы писали в машинных кодах, то есть без использования языков программирования. Для обозначения кодов операций машина оперирует с довольно длинными двоичными числами. Программисту трудно было работать с таким количеством знаков. Поэтому стали использовать системы счисления, которые с одной стороны относительно малозначны, а с другой — обеспечивают легкий перевод чисел в двоичную систему и обратно. Таковыми системами являются системы, родственные двоичной.

Система называется **родственной** двоичной, если ее основание является степенью числа 2. К таким системам относятся четверичная, восьмеричная и шестнадцатеричная. Мы рассмотрим восьмеричную и шестнадцатеричную системы.

3.3.1. Восьмеричная система

Основание $p=8$. База — цифры от 0 до 7.

Посчитаем в восьмеричной системе и сравним ее с десятичной СС (табл. 3.2).

Таблица 3.2

Таблица соответствия цифр восьмеричной системы
десятичным числам

10-я	8-я	10-я	8-я	10-я	8-я	10-я	8-я
0	0	5	5	10	12	15	17
1	1	6	6	11	13	16	20
2	2	7	7	12	14	17	21
3	3	8	10	13	15	18	22
4	4	9	11	14	16	19	23

Поскольку двоичная и восьмеричная системы являются родственными, каждая цифра восьмеричной системы может быть переведена в двоичную систему независимо от остальных цифр. Для этого нужно составить таблицу соответствия цифр восьмеричной системы двоичным числам (табл. 3.3), только двоичные числа должны быть представлены в виде триад, то есть совокупности из трех цифр.

Таблица 3.3

Таблица соответствия цифр восьмеричной
системы двоичным числам

2-я	8-я	2-я	8-я
000	0	100	4
001	1	101	5
010	2	110	6
011	3	111	7

Для перевода восьмеричного числа в двоичную систему нужно каждую цифру представить ее двоичным эквивалентом согласно таблице 3.3.

Пример: $567,23_8 = 101\ 110\ 111, 010\ 011_2$.

Для перевода двоичного числа в восьмеричную систему необходимо разделить число по триадам от запятой вправо и влево и каждую триаду представить восьмеричной цифрой со-

гласно таблице. При необходимости слева до запятой и справа после запятой можно дописывать незначащие нули.

Пример: $1110100,111101_2 = 001\ 110\ 100,111\ 101_2 = 164,758$.

Для перевода целого десятичного числа в восьмеричную необходимо выполнить последовательное деление на 8 до тех пор, пока результат не станет меньше 8. Последний результат и остатки, взятые в обратном порядке дадут восьмеричное число.

Пример: $986_{10} = 1732_8$.

Для перевода правильной дроби из 10-системы счисления в 8-ю СС нужно умножить исходную дробь и дробные части получающихся произведений на основание 8. Целые части получающихся произведений дают последовательность цифр, которая является представлением дроби в 8-ой системе счисления.

Для перевода восьмеричного числа в десятичную систему необходимо разложить его по степеням основания системы 8 и выполнить арифметические действия.

Пример: $425,7_8 = 4 \cdot 8^2 + 2 \cdot 8^1 + 5 \cdot 8^0 + 7 \cdot 8^{-1} = 277,875_{10}$.

3.3.2. Шестнадцатеричная система

Основание $p=16$. База — цифры от 0 до 9 и буквы A, B, C, D, E, F.

В табл. 3.4 приведено соответствие цифр шестнадцатеричной системы десятичным числам.

Таблица 3.4

Таблица соответствия цифр шестнадцатеричной системы десятичным числам

10-я	16-я	10-я	16-я	10-я	16-я	10-я	16-я
0	0	9	9	18	12	27	1B
1	1	10	A	19	13	28	1C
2	2	11	B	20	14	29	1D
3	3	12	C	21	15	30	1E

Продолжение табл. 3.4

4	4	13	D	22	16	31	1F
5	5	14	E	23	17	32	20
6	6	15	F	24	18	33	21
7	7	16	10	25	19	34	22
8	8	17	11	26	1A	35	23

Каждая цифра шестнадцатеричной системы может быть переведена в двоичную систему независимо от остальных цифр. Для этого нужно составить таблицу соответствия цифр шестнадцатеричной системы двоичным числам только двоичные числа должны быть представлены в виде тетрад, то есть совокупности из четырёх цифр (табл. 3.5).

Таблица 3.5

Таблица соответствия цифр шестнадцатеричной системы двоичным числам

2-а	8-я	2-я	8-я
0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

Для перевода шестнадцатеричного числа в двоичную систему нужно каждую цифру представить ее двоичным эквивалентом согласно таблице.

Пример: $56, A8_{16} = 101\ 0110, 1010\ 1000_2$.

Для перевода двоичного числа в шестнадцатеричную систему необходимо разделить число по тетрадам от запятой

вправо и влево и каждую тетраду представить шестнадцатеричной цифрой согласно таблице. При необходимости слева до запятой и справа после запятой можно дописывать недостающие нули.

Пример: 111 0100 1110 0111, $1101_2=74E7_{16}$.

Для перевода **целого** десятичного числа в шестнадцатеричную систему необходимо выполнить последовательное деление на 16 до тех пор, пока результат не станет меньше 16. Последний результат и остатки, взятые в обратном порядке дадут шестнадцатеричное число.

Пример: $986_{10}=3DA_{16}$.

Для перевода правильной дроби из 10-системы счисления в 16-ю СС нужно умножить исходную дробь и дробные части получающихся произведений на основание 16. Целые части получающихся произведений дают последовательность цифр, которая является представлением дроби в 16-ой системе счисления.

Для перевода шестнадцатеричного числа в десятичную систему необходимо разложить его по степеням основания системы 16 и выполнить сложение.

Пример:

$$4B5,2_{16} = 4 \cdot 16^2 + B \cdot 16^1 + 5 \cdot 16^0 = 4 \cdot 16^2 + 11 \cdot 16^1 + \\ + 5 \cdot 16^0 + 2 \cdot 16^{-1} = 1205,125_{10}.$$

Арифметические действия с восьмеричными и шестнадцатеричными числами выполняются аналогично десятичной системе, но с учетом цифр, используемых в системе.

3.4. Правила перевода чисел между системами счисления

Правила перевода чисел между системами счисления, сформулированные для рассмотренных систем счисления легко обобщить на системы счисления с произвольным основанием. Пусть число A , записанное в системе счисления с основанием p , необходимо перевести в систему с основанием q . Це-

лая часть числа переводится последовательным делением целой части числа A на q до тех пор, пока результат не станет меньше q . Последний результат и остатки, взятые в обратном порядке, дают число в системе с основанием q . Дробная часть числа переводится последовательным умножением дробной части числа A и дробных частей полученных произведений на q . Целые части произведения дают дробную часть числа A в системе с основанием q . Заметим также, что арифметические действия должны выполняться в системе счисления с основанием p .

3.5. Арифметические действия в позиционных системах счисления

Правила выполнения арифметических операций сложения, вычитания, умножения и деления в десятичной системе применимы и ко всем другим позиционным системам счисления. Необходимо только пользоваться особыми для каждой системы таблицами сложения и умножения.

1. Сложение

Таблицы сложения составить несложно с помощью обычного порядкового счета по строкам или столбцам таблицы. В табл. 3.6 – 3.8 представлены таблицы сложения двоичной, восьмеричной и шестнадцатеричной систем.

Таблица 3.6

Таблица сложения в двоичной системе

+	0	1
0	0	1
1	1	10

Таблица 3.7

Таблица сложения в восьмеричной системе

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
2	2	3	4	5	6	7	10	11
3	3	4	5	6	7	10	11	12
4	4	5	6	7	10	11	12	13
5	5	6	7	10	11	12	13	14
6	6	7	10	11	12	13	14	15
7	7	10	11	12	13	14	15	16

Таблица 3.8

Таблица сложения в шестнадцатеричной системе

+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10
2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11
3	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12
4	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13
5	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14
6	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15
7	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16
8	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17
9	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18
A	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19
B	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A
C	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B
D	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
F	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

При сложении в столбик числа выстраиваются друг под другом так, чтобы выровнять положение запятой. При необходимости слева до запятой и справа после запятой можно дописать необходимое количество незначащих нулей. Далее цифры по разрядно суммируются с переносом избытка в левые разряды. В качестве примера сложим числа 42,9375 и 10,8125 в разных системах счисления.

Десятичная.

$$\begin{array}{r}
 + \quad 42,9375 \\
 \quad 10,8125 \\
 \hline
 53,7500
 \end{array}$$

$5+5=10$
 $7+2+1=10$
 $3+1+1=5$
 $9+8=17=10+7$
 $2+0+1=3$
 $4+1=5$

Двоичная.

$$\begin{array}{r}
 + \quad 101010,1111 \\
 \quad 1010,1101 \\
 \hline
 110101,1100
 \end{array}$$

$1+1=2_{10}=10_2$
 $1+1=2_{10}=10_2$
 $1+1+1=3_{10}=11_2$
 $1+1+1=3_{10}=11_2$
 $1+0=1$
 $1+1=2_{10}=10_2$
 $0+0+1=1$
 $1+1=2_{10}=10_2$
 $1+0=1$
 $1+0=1$

Восьмеричная.

$$\begin{array}{r}
 + \quad 5 \ 2 \ , \ 7 \ 4 \\
 \quad 1 \ 2 \ , \ 6 \ 4 \\
 \hline
 \quad 6 \ 5 \ , \ 6 \ 0
 \end{array}$$

$4+4=8_{10}=10_8$
 $7+6+1=14_{10}=16_8=10_8+6$
 $2+2+1=5$
 $5+1=6$

Шестнадцатеричная.

$$\begin{array}{r}
 + \quad 2 \ A \ , \ F \\
 \quad A \ , \ D \\
 \hline
 \quad 3 \ 5 \ , \ C
 \end{array}$$

$F+D=1C=10+C$
 $A+A+1=15=10+5$
 $2+1=3$

2. Умножение

Для выполнения умножения чисел в различных позиционных системах счисления потребуются также следующие таблицы умножения.

Таблица 3.9

Таблица умножения в двоичной системе

x	0	1
0	0	0
1	0	10

Таблица 3.10

Таблица умножения в восьмеричной системе

x	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	10	12	14	16
3	0	3	6	11	14	17	22	25
4	0	4	10	14	20	24	30	34
5	0	5	12	17	24	31	36	43
6	0	6	14	22	30	36	44	52
7	0	7	16	25	34	43	52	61

Таблица 3.11

Таблица умножения в шестнадцатеричной системе

+	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	2	4	6	8	A	C	E	10	12	14	16	18	1A	1C	1D
3	3	5	9	C	F	12	15	18	1B	1E	21	24	27	2A	2D
4	4	8	C	10	14	18	1C	20	24	28	2C	30	34	38	3C
5	5	A	F	14	19	1E	23	28	2D	32	37	3C	41	46	4B
6	6	C	12	18	1E	24	2A	30	36	3C	42	48	4E	54	5A
7	7	E	15	1C	23	2A	31	38	3F	46	4D	54	5B	62	69
8	8	10	18	20	28	30	38	40	48	50	58	60	68	70	78
9	9	12	1B	24	2D	36	3F	48	51	5A	63	6C	75	7E	87
A	A	14	1E	28	32	3C	46	50	5A	64	6E	78	82	8C	96
B	B	16	21	2C	37	42	4D	58	63	6E	79	84	8F	9A	A5
C	C	18	24	30	3C	48	54	60	6C	78	84	90	9C	A8	B4
D	D	1A	27	34	41	4E	5B	68	75	82	8F	9C	A9	B6	C3
E	E	1C	2A	38	46	54	62	70	7E	8C	9A	A8	B6	C4	D2
F	F	1D	2D	3C	4B	5A	69	78	87	96	A5	B4	C3	D2	E1

Умножение многозначных чисел в столбик, знакомое всем применительно к десятичной системе, применимо и в осталь-

Шестнадцатеричная.

$$\begin{array}{r} \times \quad 2 \quad \text{A}, \quad 8 \\ \hline \quad \quad \text{A}, \quad 4 \\ \hline \quad \quad \text{A} \quad \text{A} \quad 0 \\ 1 \quad \text{A} \quad 9 \quad 0 \\ \hline 1 \quad \text{B} \quad 3, \quad \text{A} \quad 0 \end{array}$$

Операции вычитания и деления также выполняются аналогично десятичной системе.

4. ПРЕДСТАВЛЕНИЕ ЧИСЕЛ В ЭВМ

4.1. Прямой, обратный и дополнительный коды

Данные, с которыми работает программа, хранятся в памяти ЭВМ и обрабатывается процессором в двоичном виде. Формат записи данных в памяти называется внутренним представлением информации в ЭВМ. Для внутреннего представления информации часто применяются различные коды. Использование кодов позволяет упростить техническую реализацию узлов ЭВМ, в частности процессора. Так, например, использование обратного и дополнительного кодов в представлении целых чисел позволяет свести операцию вычитания к операции сложения, а умножение — к сложению и побитовому сдвигу. Облегчается также определение знака результата операции.

Прямой код используется для представления целого двоичного числа. Самый старший разряд числа называют знаковым и используют для кодирования знака. При этом знак «+» кодируется нулем в старшем разряде числа, а знак «-» — единицей. Обозначим $K_{mod}(x)$ — двоичный код модуля целого числа x без использования знакового разряда. Тогда для прямого кода можно записать

$$K_{np}(x) = \begin{cases} 0'K_{mod}(x), & x \geq 0, \\ 1'K_{mod}(x), & x < 0. \end{cases}$$

Здесь апострофом условно (для удобства определения знака) отделены знаковые разряды. Например, числа +6 и -6, представленные в прямом восьмиразрядном коде, выглядят так: +6 = 0'0000110; -6 = 1'0000110. а $K_{mod}(\pm 6) = 0000110$.

Обратный код используется для представления отрицательных чисел путём постановки в знаковый разряд единицы и замены во всех других разрядах числа единиц нулями, а нулей

единицами (инверсия), то есть обратный код $K_{обр}(x)$ связан с прямым следующей формулой

$$K_{обр}(x) = \begin{cases} 0' K_{mod}(x), & x \geq 0, \\ 1' \overline{K_{mod}(x)}, & x < 0, \end{cases}$$

где горизонтальной чертой сверху обозначена инверсия.

Пусть, например, необходимо получить обратный код для числа $x = -14$. Переводим число 14 в двоичную систему, получаем $K_{mod}(-14) = 0001110$. Считая, что числа представлены 8 разрядами, записываем прямой код

$$K_{пр}(-14) = 1' K_{mod}(x) = 1'0001110.$$

Поскольку число отрицательное, инвертируем все разряды кроме знакового для получения обратного кода

$$K_{обр}(-14) = 1' \overline{K_{mod}(x)} = 1'1110001.$$

Дополнительный код формируется согласно следующей формуле

$$K_{дон}(x) = \begin{cases} 0' K_{mod}(x), & x \geq 0, \\ 1' \overline{K_{mod}(x)} + 1, & x < 0. \end{cases}$$

Таким образом, дополнительный код положительного числа совпадает с прямым кодом. Для отрицательного числа дополнительный код равен обратному коду, к которому алгебраически прибавлена единица.

Найдем для примера дополнительный код для числа $x = -13$. Получаем прямой код $K_{пр}(-13) = 1'0001101$, инвертируем все, кроме знакового разряда, получаем обратный код $K_{обр}(-13) = 1'1110010$. Прибавляем к обратному коду 1, получаем $K_{обр}(-13) = 1'1110011$ — дополнительный код.

4.2. Представление в ЭВМ целых чисел

Для хранения целого числа в памяти ЭВМ отводится одна или несколько ячеек памяти размером в байт. Хранимое в них число можно интерпретировать как знаковое целое или как беззнаковое. Знаковое целое число может быть как положительным, так и отрицательным и для кодирования знака используется старший разряд числа. Часто по смыслу программируемой задачи целое число должно быть только положительным. Примером тому могут служить целые числа, подсчитывающие количество повторений каких-либо действий. Тогда знаковый разряд можно не использовать. Такие целые числа называют беззнаковыми.

Если целому числу без знака выделен в памяти компьютера один байт, то оно может принимать значения от 00000000_2 до $11111111_2 = 255_{10}$. В двубайтовом формате диапазон значений значительно шире: от $00000000\ 00000000_2$ до $11111111\ 11111111_2 = 65535_{10}$. Четырехбайтовый формат позволяет использовать еще большие числа.

Примеры:

а) число $49_{10} = 110001_2$ в однобайтовом формате:

Номера разрядов	7	6	5	4	3	2	1	0
Биты числа	0	0	1	1	0	0	0	1

б) это же число в двубайтовом формате:

Номера разрядов	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Биты числа	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1

Целым числам со знаком обычно выделяется в памяти компьютера один, два или четыре байта. Учитывая, что старший разряд кодирует знак числа, а остальные разряды хранят его модуль, запишем диапазоны значений знаковых целых чисел:

1 байт $-2^7 \dots 2^7 - 1$ $-128 \dots 127$,
 2 байт $-2^{15} \dots 2^{15} - 1$ $-32768 \dots 32767$,

4 байт $-2^{31} \dots 2^{31}-1$ $-2147483648 \dots 2147483647$.

Положительные числа хранятся в памяти в прямом коде. Обычно отрицательные десятичные числа при вводе в машину автоматически преобразуются в обратный или дополнительный двоичный код и в таком виде хранятся, перемещаются и участвуют в операциях. При выводе таких чисел происходит обратное преобразование в отрицательные десятичные числа.

4.3. Представление в ЭВМ вещественных чисел

Вещественные числа, применительно к ЭВМ несколько отличаются от математического понятия вещественных чисел. Это связано с тем, что для хранения вещественного числа в ЭВМ отводится ограниченное число разрядов. Следовательно, множество вещественных чисел в ЭВМ является в отличие от своего математического прототипа дискретным и конечным.

Существуют два способа представления вещественных чисел в ЭВМ — с фиксированной и с плавающей запятой. Как правило, ЭВМ общего назначения используют представление с плавающей запятой. ЭВМ более узкой направленности, например, предназначенные для обработки сигналов и построенные на сигнальном процессоре, могут использовать представление чисел с фиксированной запятой. Однако в любом случае представление вещественного числа в ЭВМ является приближенным и имеет свои ограничения. Они связаны с хранением и обработкой слишком маленьких и слишком больших чисел.

1. Представление чисел с фиксированной запятой

Поскольку хранение целой части числа всегда можно организовать отдельно, считают, что все хранимые в ЭВМ вещественные числа меньше единицы. Значит, запятая всегда находится перед старшим разрядом. Диапазон изменения и точность представления вещественного числа в ЭВМ зависит от количества разрядов n , отведенных для его хранения.

Диапазон изменения вещественных чисел характеризуется максимальным числом x_{\max} , которое можно записать в разрядную сетку. При использовании n разрядов оно равно по модулю

$$x_{\max} = 0, \underbrace{1111 \dots 1111}_{n \text{ разрядов}} = \sum_{k=1}^n \frac{1}{2^k}.$$

Сумму в последнем выражении можно рассматривать как сумму n членов геометрической прогрессии с первым членом $1/2$ и знаменателем $1/2$, а значит

$$x_{\max} = 1 - 2^{-n}.$$

Точность представления характеризуется самым маленьким числом x_{\min} , которое можно записать в разрядную сетку. При использовании n разрядов оно равно по модулю

$$x_{\min} = 0, \underbrace{0000 \dots 0001}_{n \text{ разрядов}} = 2^{-n}$$

Все вещественные числа, меньшие x_{\min} воспринимаются ЭВМ как нуль, а числа большие x_{\max} являются для ЭВМ бесконечно большими. Следовательно, величина x_{\min} характеризует так называемый машинный нуль, а величина x_{\max} — машинную бесконечность.

2. Представление чисел с плавающей запятой

Для представления вещественных чисел, которые могут быть как очень маленькими, так и очень большими, используется экспоненциальная форма записи чисел. Например, десятичное число 1.25 в этой форме можно представить так:

$$1.25 * 10^0 = 0.125 * 10^1 = 0.0125 * 10^2 = 12.5 * 10^{-1} = 125.0 * 10^{-2}.$$

Любое число x в системе счисления с основанием q можно записать в виде

$$x = m \cdot q^p.$$

Вещественное число m называют мантиссой, а целое число p — порядком. На экспоненциальной записи числа основано представление чисел с плавающей запятой.

Если "плавающая" запятая расположена в мантиссе перед первой значащей цифрой, то число называется нормализованным, а мантисса принимает свое максимальное значение, меньшее единицы. Для нормализованных чисел обеспечивается запись максимального количества значащих цифр числа, при фиксированном количестве разрядов мантиссы.

Рассмотрим диапазон изменения и точность представления вещественного числа. Пусть для хранения мантиссы выделено n разрядов, для хранения порядка — k разрядов, для хранения знака числа — 1 разряд. Тогда максимально возможное число, хранимое в памяти равно по модулю

$$x_{\max} = 0, \underbrace{1111 \dots 1111}_{n \text{ разрядов}} \cdot 2^{+\overbrace{1111111111}^{k \text{ разрядов}}} = (1 - 2^{-n}) \cdot 2^{(2^k - 1)}.$$

Модуль минимально возможного числа равен

$$x_{\min} = 0, \underbrace{1000 \dots 0000}_{n \text{ разрядов}} \cdot 2^{-\overbrace{1111111111}^{k \text{ разрядов}}} = \frac{1}{2} \cdot 2^{-(2^k - 1)} = 2^{(-2^k)}.$$

Распределение мантиссы и порядка для представления вещественного числа с плавающей запятой в компьютерах различных типов происходит по-разному. Поэтому рассмотрим сам принцип хранения вещественного числа. Выделенная для этого память распределяется, как показано на рис. 4.1.

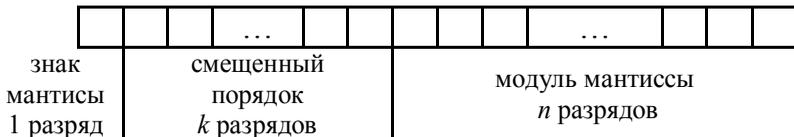


Рис. 4.1. Распределение памяти для хранения вещественных чисел

Здесь для хранения порядка выделено k разрядов. Поскольку порядок является целым числом со знаком, для его хранения в k разрядах можно применить способы представления в ЭВМ целых чисел, то есть использовать знаковый разряд, обратный или дополнительный код и т.д. Однако значительно удобнее и проще выполнять арифметические действия над порядками как беззнаковыми числами. Поэтому порядок хранится в памяти в смещенной форме. К значению порядка прибавляют самое меньшее отрицательное число, укладывающееся в выделенное количество разрядов k , что обеспечивает неотрицательное значение смещенного порядка.

Мантисса нормализованного вещественного числа хранится в отведенных ей n разрядах, от количества которых зависит точность представления мантиссы. Поскольку для представления вещественного числа используется нормализованная форма, мантисса всегда начинается с единицы. Следовательно, для хранения первого знака после запятой нет необходимости выделять отдельный бит.

Существуют стандартные форматы представления вещественных чисел, представленные в таблице.

Форматы представления вещественных чисел

Стандарт	Общее число бит, включая знаковый	Количество бит, отводимое порядку	Количество бит, отводимое мантиссе
одинарный	32	8	23
двойной	64	11	52
расширенный	80	15	64

5. АРХИТЕКТУРА ЭВМ. ПОТОКИ ИНФОРМАЦИИ В ЭВМ

5.1. Основные блоки ЭВМ. Принципы фон Неймана

В 1946 один из создателей первых ЭВМ Джон фон Нейман (1903 – 1957) изложил используемые до сих пор принципы построения ЭВМ, суть которых заключается в следующих трех утверждениях.

1) Машина должна работать не в десятичной системе, а в двоичной. Это означает, что программа и данные должны быть записаны в кодах двоичной системы, где каждое число или символ представляется определенной комбинацией нулей и единиц.

2) Программа, которая управляет последовательностью выполнения операций, должна храниться в памяти машины. Там же должны храниться исходные данные и промежуточные результаты.

3) Чтобы достаточно быстро можно было считать, память компьютера следует организовать по иерархическому принципу, то есть она должна состоять, по крайней мере, из двух частей: быстрой, но небольшой емкости (оперативной) и большой (и потому более медленной) внешней.

Согласно этим принципам можно изобразить структурную схему ЭВМ в виде рис. 5.1.

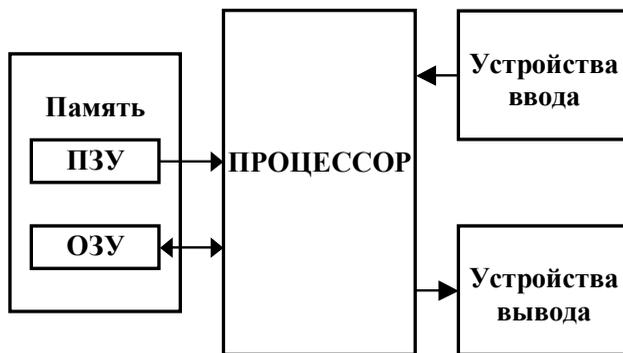


Рис. 5.1. Структурная схема ЭВМ

5.2. Процессор

Процессор (центральный процессор) — устройство, обеспечивающее обработку данных по заданной программе. Он также организует обмен данными и командами между устройствами ЭВМ. Процессор умеет выполнять некоторый заданный набор операций по обработке данных и управления процессом этой обработки. Набор операций определяется конструкцией процессора.

Процессор может выполнять обработку данных только при наличии заранее составленной программы. Программы и данные для ее выполнения хранятся в памяти. Программы состоят из команд, которые содержат коды операций, информацию об операндах и о том, куда поместить результат.

Автоматическая работа процессора по заданной программе обеспечивается двумя основными устройствами, входящими в его состав: арифметико-логическим устройством (АЛУ) и устройством управления (УУ). При этом используются регистры (ячейки памяти внутри процессора, предназначенные для кратковременного хранения данных в процессе их обработки и быстрого доступа к ним).

Устройство управления формирует и подает во все блоки машины в нужные моменты времени определенные сигналы управления (управляющие импульсы), обусловленные спецификой выполняемой операции и результатами предыдущих операций; формирует адреса ячеек памяти, используемых выполняемой операцией, и передает эти адреса в соответствующие блоки ЭВМ, получает от генератора тактовых импульсов опорную последовательность импульсов.

Арифметико-логическое устройство предназначено для выполнения всех арифметических и логических операций над числовой и символьной информацией. АЛУ выполняет арифметические операции только над двоичной информацией с запятой, фиксированной после последнего разряда, т.е. только над целыми двоичными числами.

Выполнение операций над двоичными числами с плавающей запятой и над двоично-кодированными десятичными числами осуществляется или с привлечением математического сопроцессора, или по специально составленным программам.

Микропроцессорная память (МПП) — память небольшой емкости, но чрезвычайно высокого быстродействия (время обращения к МПП, т.е. время, необходимое на поиск, запись или считывание информации из этой памяти, измеряется наносекундами).

Она предназначена для кратковременного хранения, записи и выдачи информации, непосредственно в ближайшие такты работы машины участвующей в вычислениях; МПП используется для обеспечения высокого быстродействия машины, ибо основная память не всегда обеспечивает скорость записи, поиска и считывания информации, необходимую для эффективной работы быстродействующего процессора.

Генератор тактовых импульсов генерирует последовательность электрических импульсов, частота которых определяет тактовую частоту машины.

Промежуток времени между соседними импульсами определяет время одного такта работы машины или просто такт работы машины.

Частота генератора тактовых импульсов является одной из основных характеристик персонального компьютера и во многом определяет скорость его работы, ибо каждая операция в машине выполняется за определенное количество тактов.

Основными характеристиками процессоров являются *разрядность* и *быстродействие*. **Разрядность** — это число одновременно обрабатываемых бит. **Быстродействие** — число выполняемых команд в секунду. Быстродействие связано с тактовой частотой, на которой работает процессор. Чем выше тактовая частота, тем выше и быстродействие.

Современные процессоры комплектуются сопроцессором, который может реализовывать разнообразные функции,

например, выполнять арифметические действия с вещественными числами.

Подключение процессора к схеме предусматривает наличие шины данных (ШД), шины адреса (ША), шины управления (ШУ). Разрядность ША определяет адресное пространство процессора и количество напрямую адресуемой памяти. Разрядность шины данных характеризует количество бит, одновременно передаваемых между устройствами ЭВМ. Не следует путать разрядность шины данных с разрядностью процессора в целом.

Функционирование процессора представляет собой циклический процесс последовательной обработки информации, разработанный фон Нейманом, состоящий из следующих основных этапов.

1. Процессор выставляет число, хранящееся в регистре счётчика команд, на шину адреса, и отдаёт памяти команду чтения.
2. Выставленное число является для памяти адресом; память, получив адрес и команду чтения, выставляет содержимое, хранящееся по этому адресу, на шину данных, и сообщает о готовности
3. Процессор получает число с шины данных, интерпретирует его как команду (машинную инструкцию) из своей системы команд и исполняет её.
4. Если последняя команда не является командой перехода, процессор увеличивает на единицу (в предположении, что длина каждой команды равна единице) число, хранящееся в счётчике команд; в результате там образуется адрес следующей команды
5. Снова выполняется п. 1

Этот цикл выполняется бесконечно и называется процессом (отсюда и произошло название устройства). Большинство команд процессора требуют также дополнительных пунктов процесса. Например, арифметические команды могут требовать считывания из памяти своих аргументов.

5.3. История развития процессоров

Более правильным названием процессора является словосочетание «центральное процессорное устройство» (ЦПУ, CPU). ЦПУ вычислительных машин первых поколений были поostrены на дискретных электронных компонентах и представляли собой довольно сложные устройства. Современные ЦПУ как правило выполнены в виде единственной микросхемы, и поэтому называются микропроцессорами или просто процессорами. С середины 80-х они практически вытеснили прочие виды процессоров, вследствие чего само слово «процессор» стало всё чаще и чаще восприниматься как синоним слова «микропроцессор» и заменило термин ЦПУ. Вместе с тем ЦПУ некоторых суперкомпьютеров даже сегодня представляют собой сложные комплексы больших и сверхбольших интегральных схем [9,10]. В данном пункте речь пойдет о микропроцессорах, которые использовались и используются в современных персональных компьютерах (ПК).

Вначале совершенствования микропроцессоров главной задачей разработчиков было расширение их функций. Процессор должен был выполнять как можно больше разнообразных инструкций. Поэтому система команд процессора включала в себя много сложных и разных по назначению инструкций. Компьютеры с подобной архитектурой процессора получили название CISC — Complex Instruction Set Computers (компьютеры со сложным набором команд). Однако по мере развития архитектуры CISC стало ясно, что вычисления с расширенным набором сложных команд приводят к ряду трудностей. Крайне трудоемким был процесс разработки и изготовления более новых процессоров. В разработке программного обеспечения многие команды процессора не были задействованы совсем. Зачастую сложная команда процессора в связи с затратой времени на ее расшифровку работала медленнее, чем аналогичная ей цепочка простых команд.

Отдельной ветвью в развитии процессоров стало упрощение их архитектуры. Была сведена к минимуму система команд, а

также количество способов адресации, значительно увеличено число регистров общего назначения. Это позволило программам сохранять большую часть данных хранить именно в регистрах, а не в оперативной памяти. Существенное упрощение архитектуры позволяло также повысить тактовую частоту. Для обозначения компьютеров с подобной архитектурой процессора используют аббревиатуру **RISC** (Reduced Instruction Set Computers).

Вместе с появлением RISC процессоров было введено понятие конвейера. Конвейерная архитектура (pipelining) была введена в центральный процессор с целью повышения быстродействия. Обычно для выполнения каждой команды требуется осуществить некоторое количество однотипных операций, например: выборка команды из ОЗУ, дешифрация команды, адресация операнда в ОЗУ, выборка операнда из ОЗУ, выполнение команды, запись результата в ОЗУ. Каждую из этих операций сопоставляют одной ступени конвейера. После освобождения любой ступени конвейера она сразу приступает к работе над следующей командой. Это позволяет значительно повысить быстродействие процессора.

На основе идеологии RISC были спроектированы несколько типов архитектур процессоров, например, MIPS, Sparc, ARM PowerPC, DEC Alpha и др..

MIPS (Microprocessor without Interlocked Pipeline Stages — «микропроцессор без блокировок в конвейере») — семейство RISC-микропроцессоров, разрабатываемое компанией MIPS Technologies. Основная идея этой архитектуры заключается в том, что сильно упростив внутреннее устройство процессора и используя очень длинный конвейер, можно получить процессор, не умеющий выполнять сравнительно сложные инструкции, зато работающий на очень высоких тактовых частотах. В 1985 был выпущен первый 32-разрядный MIPS-процессор R2000; в 1988 году — значительно более быстрый R3000, который использовался, например, в рабочих станциях Silicon Graphics и приставках Sony PlayStation.

В 1991 году появился первый 64-разрядный MIPS R4000. Начиная с R4600 и R4700 MIPS-процессоры стали основой для

маршрутизаторов Cisco. В 1995 году вышел R10000, который работал на частоте 200 МГц и был одним из самых лучших процессоров того времени. Одной из последних моделей MIPS процессора был R14000 (2001 г.) с тактовой частотой всего 600 МГц. На этом активная разработка процессоров этой серии была приостановлена. На сегодняшний день архитектура MIPS используется в основном во встраиваемых системах: в маршрутизаторах, КПК, сотовых телефонах и т.д.

SPARC (Scalable Processor ARChitecture) — масштабируемая процессорная архитектура разработана в 1985 году компанией Sun Microsystems. Основная идея этой архитектуры заключается в том, чтобы свести к минимуму работу со стеком при выполнении подпрограмм. Для этого используется набор из 128 регистров; но в каждый момент времени рабочими из них могут быть только 24 подряд идущих регистра. Они образуют окно с общим пространстве регистров. Имеются еще восемь отдельных глобальных регистров. При вызове подпрограммы окно смещается, а при окончании работы подпрограммы окно возвращается на прежнее место. Тем самым снимается необходимость запоминать состояние процессора в стеке при переходе к выполнению подпрограммы. SPARC-архитектура развивалась до середины 90-х годов, затем не выдержала конкуренции.

Особенностью архитектуры **ARM** (Advanced RISC Machines) стала возможность использовать каждую команду процессора совместно с кодом условия. Команда выполнялась только при если в процессоре был выставлен соответствующий флаг.

Появились первые процессоры ARM архитектуры в 1985 году. Они были 32-разрядными, отличались низким энергопотреблением и производительностью, превосходящей производительность 286-го процессора.

На сегодняшний день технология ARM является доминирующей микропроцессорной архитектурой 32-х разрядных RISC-процессоров для портативных цифровых устройств (мобильные телефоны, органайзеры и т.д.).

PowerPC — микропроцессорная RISC-архитектура, созданная в 1991 альянсом компаний Apple-IBM-Motorola. Это классический RISC-процессор. Существуют 32- и 64-разрядные версии PowerPC. Примерно до 2001 года развитие линии процессоров PowerPC шло синхронно с процессорами Intel и AMD. На их базе производились компьютеры фирмы Apple. Однако постепенно наметилось отставание PowerPC от своих конкурентов. В настоящее время архитектура PowerPC положена в основу многоядерных процессоров для игровых приставок

DEC Alpha, также известный как **Alpha AXP**, — 64-разрядный RISC микропроцессор, разработанный компанией Digital Equipment Corp. (DEC). Особенностью процессоров Alpha является не столько сама архитектура, сколько её продуманная реализация. Процессоры Alpha обозначались как серия DECchip 21x64, где «21» обозначало 21-й век, а «64» указывало на то, что процессор являлся 64-разрядным, средняя цифра указывала поколение архитектуры Alpha.

Первый процессор Alpha 21064 появился в 1992 и работал на частоте до 192 МГц. Процессор Alpha 21164 был выпущен в 1995 году и работал на частотах до 333 МГц. В июле 1996 частота была доведена до 500 МГц, а в марте 1998 — до 666 МГц. Кроме того, в 1998 году был выпущен 21264, первоначальная тактовая частота которого составляла 450 МГц; со временем она возросла до 1,25 ГГц. В 2003 году был выпущен 21364 Marvel, работавший на частоте от 1 до 1,15 ГГц. Далее последовала покупка компании DEC компанией Compaq, и через некоторое время было заявлено о прекращении выпуска процессоров Alpha и использовании вместо них Intel Itanium. Через некоторое время компания HP, новый владелец Compaq, приняла решение пока продолжить развитие серии процессоров Alpha.

Модельный ряд процессоров серии Alpha представлен в табл. 5.1.

Таблица 5.1

Модельный ряд процессоров Alpha

Модель	Год выпуска	Тактовая частота, МГц
21064	1992	100—200
21064A	1994	200—300
21066	1993	100—166
21066A	1994	166—233
21164	1995	266—500
21164A	1996	400—767
21164PC	1997	400—667
21264	1998	450—600
21264A	1999	667—750
21264B	2001	800—833
21264C	2001	1000—1250
21364	2003	800—1300

Несколько подробнее остановимся на истории развития процессоров фирм **Intel** и **AMD**, поскольку именно они были и остаются самыми распространенными в России. Следует отметить, что процессоры Intel и AMD первоначально относились к процессорам с CISC архитектурой. Постепенная их эволюция привела к совмещению RISC и CISC архитектур в одном процессоре. С одной стороны процессор имеет расширенный набор команд, то есть он является CISC-процессором. С другой стороны, сложная CISC команда, будучи представленной в виде последовательности более простых, отдается на выполнение RISC-ядру процессора.

Первый 4-х разрядный процессор фирмы Intel 4004 был выпущен в 1971 году. Он имел тактовую частоту 108 кГц и предназначался для калькуляторов Busicom. В 1972 году появился 8-ми разрядный Intel 8008 с тактовой частотой 200 кГц, а в 1974 году — Intel 8080 с тактовой частотой 2 МГц.

На базе следующей модели Intel 8086 (1978 г.) начали выпускать компьютеры IBM PC. Он обладал 16-разрядной шиной данных и 20-разрядной шиной адреса. Использовалась тактовая частота от 4,77 до 10 МГц. Облегченная 8-разрядная модификация этого процессора называлась Intel 8088 (1979 г.).

В 1982 году были выпущены сразу две 16-разрядные модели Intel 80186 и 80286. Первый из них был довольно быстро признан неудавшимся. Процессор 80286 мог поддерживать многозадачный режим. При работе в многозадачном режиме процессор поочередно переключается от одной задачи к другой, но в каждый момент времени выполняется только одна программа. Тактовая частота этой модели — от 6 до 12 МГц, шина данных 16 разрядов, шина адреса — 24 разряда. Для процессора 80286 выпускался математический сопроцессор 80287.

Следующая модель уже 32-разрядного процессора Intel 80386 (1985 г.) отличается от всех предыдущих использованием конвейера. Процессор 80386 может работать в двух режимах: в режиме реальной адресации и режиме защищенной виртуальной адресации. Были впервые применены 32-разрядные шины данных и адреса.

С 1989 по 1994 годы выпускались процессоры Intel 80486 с различными суффиксами DX, DX2, DX4, SX, SL, SX2. Отличительной особенностью этих процессоров было наличие КЭШ-памяти первого уровня (памяти внутри процессора) размером 8 Кб (DX4 — 16 Кб). Модели с суффиксами DX* комплектовались математическим сопроцессором (FPU). Модели с суффиксами SX* считались облегченной версией моделей DX* и обладали ограниченными возможностями. Процессор 80486SL является расширенной версией 80486DX. Тактовая частота процессоров 80486 в зависимости от модели принимала значение между 20 и 100 МГц.

Следующим шагом эволюции процессоров Intel стало появление модельного ряда Pentium (1993 – 1997). Это первый процессор с двухконвейерной структурой. КЭШ памяти первого уровня размером 16 Кб была разделена на 2 часть — 8 Кб для данных,

8 Кб — для инструкций. Характеристики разных моделей Pentium представлены в табл. 5.2.

Таблица 5.2

Характеристики процессоров Pentium

Модель Pentium	Тактовая частота, МГц	Разрядность		
		ШД	ША	Общая
P5	60-66	64	32	32
P54C	75-200	64	32	32
Pro	150-200	64	32	32
MMX P55C	166-233	64	32	32

Отличительной особенностью Pentium Pro было наличие КЭШ памяти второго уровня, собранной в одном корпусе с процессором. Для всех более ранних моделей КЭШ память второго уровня располагалась на материнской плате.

Аббревиатура MMX (Multi Media eXtention) означает наличие у процессора дополнительных инструкций для вычислений с плавающей точкой. Такое техническое решение позволило существенно увеличить производительность компьютера в мультимедиа-приложениях.

Дальнейшее развитие процессоров фирмы Intel от Pentium II до Pentium III происходило плавным эволюционным путем без радикальных нововведений. Совершенствовался технологический процесс производства процессоров, уменьшались размеры элементов внутри микросхемы. В этой связи увеличивалась тактовая частота, варьировался размер КЭШ-памяти первого и второго уровней, изменялся тип корпуса. Но основные характеристики процессоров оставались неизменными: разрядность шины данных 64, шины адреса 36, общая разрядность 32. В табл. 5.3 представлены характеристики процессоров Pentium II – Pentium III.

Таблица 5.3

Характеристики процессоров Pentium II и Pentium III

Модель	Тактовая частота, МГц	КЭШ 1 (L1), Кб	КЭШ 2 (L2), Кб	Тип корпуса, год вып.
Pentium II (Klamath)	233-300	32	512 (вне процессора)	Slot1, 1997
Pentium II (Deschutes)	266-450	32	512 (вне процессора)	Slot1, 1998
Pentium II OverDrive	333	32	512	Socket8, 1998
Pentium II (Tonga)	233-300	16 для данных, 16 для команд	512	MMC (для ноутбук-ов), 1998
Celeron (Covington)	266-300	32 (16+16)	—	Slot1, 1998
Pentium II Xeon	400-450	32 (16+16)	512-2048	Slot2, 1998
Celeron (Mendocino)	300-433	32 (16+16)	128	Slot1, 1998
Celeron (Mendocino)	300-533	32 (16+16)	128	Socket 370 1999
Pentium II PE (Dixon)	266-500	32 (16+16)	256	MMC, 1999
Pentium III (Katmai)	450-600	32 (16+16)	512	Slot1, 1999
Pentium III Xeon (Tanner)	500-900	32 (16+16)	512-2048	Slot2, 1999
Pentium III (Coppermine)	533-1000	32 (16+16)	256	Socket 370 1999
Celeron (Coppermine)	566-1100	32 (16+16)	128	Socket 370 1999

Продолжение табл. 5.3

Pentium III-S (Tualatin)	1130-1400	32 (16+16)	512	Socket 370 2001
Celeron (Tualatin)	1000-1300	32 (16+16)	256	Socket 370 2001

Первый процессор Pentium 4 был выпущен в 2000 году. Это был принципиально новый процессор с гиперконвейеризацией. Процессоры, основанные на данной технологии, позволяют добиться увеличения частоты примерно на 40 процентов по сравнению с предыдущими моделями. В табл. 5.4 показан модельный ряд процессоров серии Pentium 4.

Таблица 5.4

Модельный ряд Pentium 4

Модель	Тактовая частота, ГГц	КЭШ 1 (L1), Кб	КЭШ 2 (L2), Кб	Тип корпуса, год вып.
Pentium 4 (Willamette)	1,3-2	8 для данных, 12 для команд	256	Socket 423, 478 2000
Pentium 4 (Northwood)	1,6-3,4	8 для данных, 12 для команд	512	Socket 478 2002
Pentium 4 (Prescott)	2,8-3,8	16 для данных, 12 для команд	1024	Socket 478, 775 2004
Celeron D	2.13-3.33	16 для данных, 12 для команд	256	Socket 775 2005

Первые модели процессора Pentium 4 были 32 разрядными, хотя и пользовались 64 разрядными шинами данных и адреса. Дальнейшее развитие процессорной техники привело к необходимости перехода к 64-разрядным моделям процессоров. Поэтому в Pentium 4 были введены расширения EMТ64, позволяющие исполнять в режиме эмуляции инструкции, характерные для 64-разрядных процессоров.

Главным конкурентом процессорам Intel в последние годы по праву считаются процессоры формы AMD. В годы своего становления фирма AMD выпускала процессоры по лицензии Intel. Со временем собственные разработки AMD перестали уступать, а затем и превосходили своих конкурентов. Модельный ряд процессоров AMD приведен в табл. 5.5.

Таблица 5.5

Модельный ряд процессоров AMD

Модель	Год выпуска	Тактовая частота, МГц	Разрядность		
			ШД	ША	общая
Am 286	1982	12-16	16	24	16
Am 386 DX	1983	16-32	32	32	32
Am 386 SX	1985	16-32	16	24	16
Am 486 DX, DX2, DX4	1991-1994	25-120	32	32	32
Am 586	1995	133	32	32	32
K5 (5k86)	1996	90-133	64	32	32
K6	1997	166-233	64	32	32
K6 (Little Foot)	1997	233-300	64	32	32
K6-2	1998	266-500	64	32	32
K6-III	1999	350-500	64	32	32
Athlon	1999	500-1400	64	32	32
Duron	2000	600-1300	64	32	32
Athlon XP	2001	650-2250	64	32	32
Opteron	2003	1400-2600	64	64	32/64
Sempron	2005	1600-2200	64	64	32/64
Athlon 64	2005	1800-2800	128	64	64

Процессоры AMD K5 построены по архитектуре x86-to-RISC86, принципиально отличной от архитектуры примененной в процессорах Intel Pentium. Начиная с модели Athlon XP при маркировке процессоров фирмы AMD используется не реальная тактовая частота, а индекс производительности, т. е. показывается

какому Pentium 4 соответствует данный процессор. Например Athlon XP 2000+ работает на частоте 1666 МГц, Athlon 64 3200+ работает на частоте 2000 МГц, но соответствует производительности Pentium 4 с частотой 3800 МГц.

Совершенствование микропроцессоров тесно связано с развитием технологии их производства, которое вплоть до настоящего времени шло по пути уменьшения размеров и уплотнения элементов на кристалле микропроцессора. На современном этапе достигнуты рекордные показатели: размер одного транзистора достиг порядка 65 нм. Дальнейшее уменьшение размеров элементов является затруднительным, так как размер транзистора становится сравнимым с размером молекул, из которого он состоит. Поэтому являются актуальными иные пути совершенствования микропроцессорной техники. Одним из таких путей является применение многоядерных процессоров.

Многоядерные процессоры представляют собой несколько процессоров, размещенных на одном кристалле и использующих общую кэш-память. Современными представителями двуядерных процессоров для настольных ПК являются AMD Athlon 64 X2, Pentium D, Pentium Extrim Edition, Intel Core 2 Duo, Intel Xeon Dual Core, четырехядерных — Intel Xeon Quad Core, Intel Core 2 Quad.

5.4. Память

Память предназначена для хранения команд и обрабатываемых данных. Память современных ЭВМ строится по иерархическому принципу, согласно которому существуют несколько запоминающих устройств, разных по объему и быстродействию. Наиболее общим является деление памяти на внутреннюю и внешнюю. Внутренняя память имеет малый объем, но высокое быстродействие, в то время как внешняя память большого объема, но медленная. Однако **внутренняя** память также имеет свою иерархию. Память, ячейки которой «пронумерованы», называют адресуемой памятью. Доступ к такой памяти осуществляется с использованием шины адреса. Помимо адресуемой существует так-

же неадресуемая, недоступная для программиста память. Охарактеризует основные запоминающие устройства внутренней памяти персонального компьютера.

Регистровая память является частью процессора и представляет собой набор регистров общего назначения (РОН). Ее объем и разрядность зависит от типа процессора. Каждый из РОН имеет либо свое уникальное имя, либо номер (адрес). В процессорах Intel и AMD различается также назначение каждого регистра. Регистровая память считается адресуемой и доступна для программиста. Именно данные, хранящиеся в регистрах являются операндами для большинства процессорных команд. Регистровая память является самой быстродействующей, но имеет очень маленький объем.

Кэш-память берет свое название от английского *cache* (тайник). Она относится к неадресуемой памяти и предназначена для кратковременного хранения «промежуточных» данных, которые еще могут потребоваться. Кэш-памятью обладают многие устройства ЭВМ и используют ее по своему усмотрению. Например, процессор имеет кэш-память нескольких уровней. Кэш первого уровня (L1) размещается на одном кристалле с процессором, является самым быстрым. Кэш второго уровня (L2) сначала располагался на материнской плате в виде отдельного устройства из дискретных микросхем. Затем его интегрировали в процессор, что позволило увеличить его быстродействие.

Кэш-памятью можно назвать также видеопамять, расположенную на видеокарте, так как лишь малая часть этой памяти имеет в режиме VGA свои адреса в общем адресном пространстве. Кэш-памятью обладают многие современные жесткие диски.

Основная идея использования кэш-памяти разных уровней состоит в том, чтобы при стыковке устройств с разной скоростью работы минимизировать с ее помощью время простоя наиболее быстрого из них.

Оперативное запоминающее устройство (ОЗУ, RAM) является основной энергозависимой адресуемой памятью ЭВМ. Именно здесь располагаются выполняемые программы совместно

с обрабатываемыми данными. Содержимое ОЗУ стирается при выключении питания ЭВМ.

По принципу действия различают статические и динамические ОЗУ. Статическая оперативная память (Static RAM — SRAM) использует для запоминания триггеры. В динамическое оперативной памяти (Dynamic RAM — DRAM) запоминающими элементами являются конденсаторы, плотно упакованные в микросхему. Поскольку конденсаторы достаточно быстро теряют заряд, для его поддержания требуется постоянная регенерация заряда. В современных ПК используются несколько типов динамических ОЗУ:

- SDRAM (Synchronous DRAM) — память, синхронизированная с системной шиной;
- RDRAM (Rambus DRAM, используется в модулях RIMM) — память SDRAM с удвоенной шиной передачи данных и увеличенной частотой;
- DDR (Double Data Rate) — усовершенствование SDRAM, заключающееся в том, что за один такт работы памяти данные передаются дважды: вначале цикла и в конце;
- DDR2 — данные считываются из микросхемы памяти в буфер также как в модулях DDR, но далее из буфера к контроллеру памяти передаются вдвое быстрее.

Внешний вид модуля SDRAM показан на рис. 5.2.

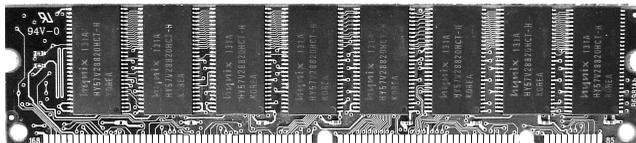


Рис. 5.2. Модуль памяти SDRAM 256 Мб.

Постоянное запоминающее устройство (ПЗУ, ROM BIOS) занимает часть адресного пространства ЭВМ. Оно служит для хранения информации, которая не изменяется в ходе работы ЭВМ. Как правило ПЗУ компьютера отождествляют с BIOS (Basic

Input/Output System, базовая система ввода-вывода), которая является встроенным программным обеспечением и содержит подпрограммы, необходимые для управления клавиатурой, видеокартой, дисками, портами и другими устройствами ЭВМ. В ПЗУ располагаются также программы начального старта и тестирования работоспособности.

При включении ПК запускается находящаяся в BIOS программа начального старта BOOT-ROUTINE. Она в свою очередь вызывает программу самотестирования POST (Power-On Self Test). Далее производится поиск доступного загрузочного устройства, с которого считывается начальный загрузчик операционной системы. Подпрограммы, расположенные в BIOS остаются доступными в ходе работы ЭВМ.

Конструктивно BIOS выполнен в виде одной или нескольких микросхем ПЗУ, в которых могут быть использованы различные типы памяти:

ROM — данные записываются в ПЗУ в процессе ее изготовления;

PROM (Programmable Read-Only Memory) — данные могут быть записаны однократно с помощью программатора;

EPROM (Erasable Programmable Read Only Memory) - стираемое программируемое ПЗУ) — данные можно перезаписать с помощью программатора ограниченное число раз после стирания ультрафиолетовыми лучами;

EEPROM — стирание и запись производятся электрическим сигналами повышенного напряжения;

Flash BIOS — данные могут перезаписываться электрическими сигналами рабочего напряжения при помощи специальной программы. Такой подход облегчает модернизацию BIOS, но открывают простор вирусам.

Энергозависимая память **CMOS** (Complementary Metal Oxide Semiconductor, КМОП) служит для запоминания аппаратной конфигурации компьютера. Для питания этого вида памяти на

системной плате устанавливают отдельный малогабаритный аккумулятор.

5.5. Магистрально-модульный принцип построения

Большинство современных ЭВМ строятся согласно магистрально-модульному принципу. Структурная схема ЭВМ, построенной согласно магистрально-модульному принципу, изображена на рис. 5.3.

Процессор связан с остальными блоками ЭВМ посредством системной магистрали. Поскольку для минимального функционирования ЭВМ достаточно наличия процессора и памяти, процессор и память называют центральными устройствами ЭВМ, а все остальные устройства — внешними.



Рис. 5.3. Структурная схема ЭВМ

Системная магистраль (системная шина, внутримашинный системный интерфейс) — это основная интерфейсная система компьютера, обеспечивающая сопряжение и связь всех его устройств между собой.

Системная шина включает в себя:

- шину данных (ШД), содержащую провода и схемы сопряжения для параллельной передачи всех разрядов числового кода операнда;

- шину адреса (ША), включающую провода и схемы сопряжения для параллельной передачи всех разрядов кода адреса ячейки основной памяти или порта ввода-вывода внешнего устройства;
- шину управления (ШУ), содержащую провода и схемы сопряжения для передачи управляющих сигналов, импульсов во все блоки машины;
- шину питания, имеющую провода и схемы сопряжения для подключения блоков ПК к системе энергопитания.

Системная шина обеспечивает три направления передачи информации:

- между микропроцессором и основной памятью;
- между микропроцессором и портами ввода-вывода внешних устройств;
- между основной памятью и портами ввода-вывода внешних устройств (в режиме прямого доступа к памяти).

Основной функцией системной шины является передача информации между процессором и остальными устройствами ЭВМ.

Внешние устройства подключаются, в свою очередь, к системной шине через контроллеры (или адаптеры).

Контроллер — специализированное устройство согласования центральных и внешних устройств. Большинство контроллеров известно под другими названиями. Например, контроллер монитора называют видеокартой, для контроллера звуковых устройств (колонки, микрофон и т.д.) используют название звуковая карта. Контроллеры многих устройств к настоящему времени стандартизированы и поэтому собираются вместе с системной магистралью на материнской плате. Такими, например, являются контроллеры клавиатуры, мыши, жестких дисков, дисководов и др.

Конструктивно системная магистраль состоит как минимум из двух микросхем. Первая из них, называемая «северный мост», обеспечивает взаимодействие процессора с наиболее

быстрыми устройствами, такими как память, видеокарта. Вторая — «южный мост» организует связь процессора с более медленными устройствами, такими как контроллеры жестких дисков, звуковых устройств и т.д. В совокупности северный и южный мосты называют Chipset или набор микросхем системной логики.

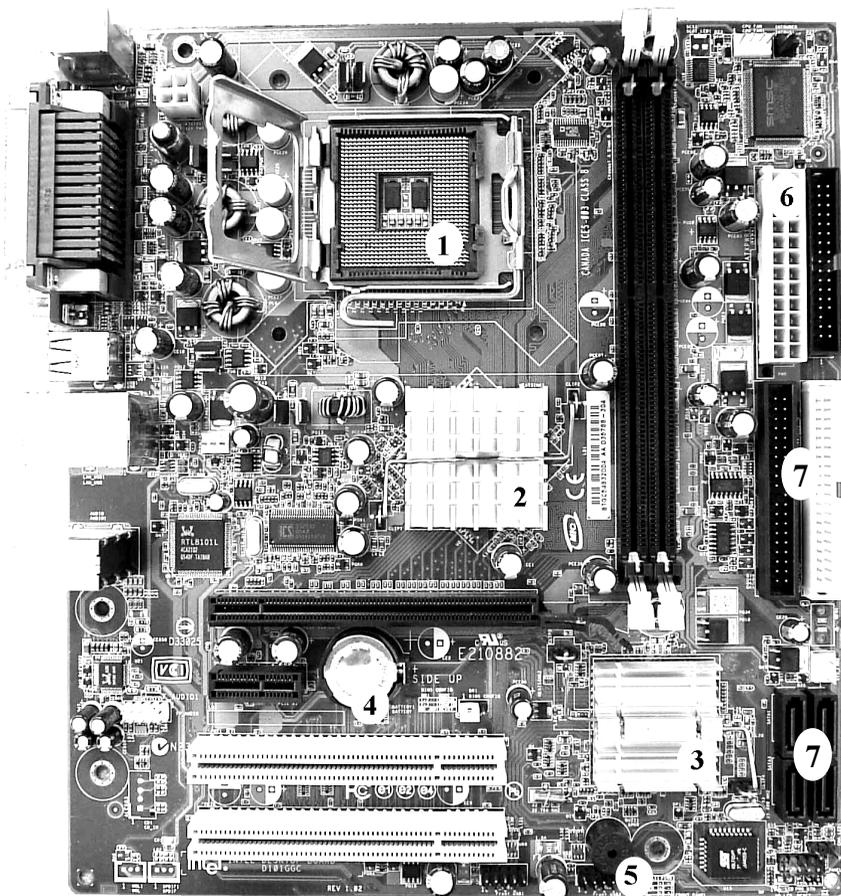


Рис. 5.4. Материнская плата

На рис. 5.4 показана материнская плата, на которой обозначено:

- 1 – разъем для установки процессора;
- 2 – микросхема северного моста под радиатором;
- 3 – микросхема южного моста под радиатором;
- 4 – аккумулятор для поддержания памяти CMOS;
- 5 – системный динамик;
- 6 – разъем питания;
- 7 – разъемы для подключения НЖМД.

5.6. Характеристики периферийных устройств

Внешние или периферийные устройства (ВУ) — это важная составная часть любого вычислительного комплекса, так как ВУ ПК обеспечивают взаимодействие машины с окружающей средой: пользователями, объектами управления и другими ЭВМ. ВУ весьма разнообразны и могут быть классифицированы по ряду признаков.

Так, по назначению можно выделить следующие виды ВУ:

- внешние запоминающие устройства (ВЗУ);
- диалоговые средства пользователя;
- устройства ввода информации;
- устройства вывода информации;
- средства связи и телекоммуникации.

Внешние запоминающие устройства (ВЗУ) предназначены для долговременного хранения программ и данных. Устройства выполняют в виде накопителей, носителями информации в которых могут служить диски и ленты, flash-микросхемы. Накопители могут быть со сменными носителями и со стационарными носителями информации.

По способу доступа ВЗУ делятся на устройства прямого доступа и устройства последовательного доступа. Накопители на дисках — устройства прямого доступа, так как позволяют

обратиться непосредственно к любому месту дискового пространства. Накопители на магнитных лентах — устройства последовательного доступа, так как доступ к нужной информации требует предварительного просмотра всей предыдущей.

Основные характеристики ВЗУ— информационная емкость и время доступа.

Накопители на гибких магнитных дисках (НГМД, FDD).

Носителями информации являются гибкие полимерные диски с нанесенным магнитным покрытием. Диск помещают в плотный конверт (корпус), на внутреннюю поверхность которого нанесено очищающее покрытие.

В настоящее время используются диски диаметром 3,5 дюйма (89 мм). Информация записывается с двух сторон по концентрическим дорожкам, которые радиальным разбиением разделены на секторы. Емкость сектора фиксирована 512 байт. Количество секторов — 9 или 15 на 5,25, 9 или 18 на 3,5. Количество дорожек бывает 40 (DD) и 80 (QD, HD). Тогда

$$\text{Емкость} = \text{количество секторов} * \text{количество дорожек} * \\ * \text{количество сторон} * 512.$$

Двусторонние дискеты маркируют DS 2S или просто 2. 40-дорожечные — DD, 80-дорожечные — QD или HD.

Накопители на жестких магнитных дисках (НЖМД, HDD).

Носителями информации являются несъемные диски, изготовленные из алюминиевого сплава с магниточувствительным покрытием. Диски вращаются с большой скоростью 3600 – 10000 об/мин. Каждая дисковая поверхность имеет свою магнитную головку.

Достоинства: большая емкость, малое время доступа к информации. Недостатки: чувствительность к механическим воздействиям (вибрации, толчки).

Накопители на CD дисках.

Закодированная информация наносится на оптический компакт-диск лазерным лучом, который создает на его поверхности микроскопические впадины, разделенные плоским

участком. Использование в качестве одного сигнала плоского участка, а другого — выжженной впадины дают возможность хранения на диске двоично закодированной информации.

Считывание производится менее мощным лучом. Отраженный от поверхности диска луч направляется на фотодетектор. Наличие напряжения на фотодетекторе соответствует плоскому участку диска, а отсутствие напряжения — впадине.

Емкость диска зависит от плотности размещения дорожек, которая в свою очередь помимо тонкостей технической реализации устройств чтения-записи зависит от длины волны используемого света. При длине волны 800 нм (красный цвет) емкость диска составляет 700Мб.

Накопители на магнитной ленте — стримеры.

Здесь реализуется последовательный способ доступа. Наиболее часто используются четвертьдюймовые картриджи для стримера, 8мм и 4мм картриджи.

Диалоговые средства пользователя включают в свой состав видеомониторы (дисплей), реже пультовые пишущие машинки (принтеры с клавиатурой) и устройства речевого ввода-вывода информации.

Видеомонитор (дисплей) — устройство для отображения вводимой и выводимой из ПК текстовой и графической информации на цветном или монохромном экране.

Устройства речевого ввода-вывода относятся к быстро развивающимся средствам мультимедиа. Устройства речевого ввода — это различные микрофонные акустические системы, "звуковые мыши", например, со сложным программным обеспечением, позволяющим распознавать произносимые человеком буквы и слова, идентифицировать их и закодировать.

Устройства речевого вывода — это различные синтезаторы звука, выполняющие преобразование цифровых кодов в буквы и слова, воспроизводимые через громкоговорители (динамики) или звуковые колонки, подсоединенные к компьютеру.

К устройствам ввода информации относятся:

- клавиатура — устройство для ручного ввода числовой, текстовой и управляющей информации в ПК;

- графические планшеты (диджитайзеры) — для ручного ввода графической информации, изображений путем перемещения по планшету специального указателя (пера); при перемещении пера автоматически выполняются считывание координат его местоположения и ввод этих координат в ПК;

- сканеры — устройства для автоматического считывания с бумажных носителей и ввода в ПК текстов, графиков, рисунков, чертежей, фотографий;

В устройстве кодирования сканера в текстовом режиме считанные символы после сравнения с эталонными контурами специальными программами преобразуются в коды, а в графическом режиме считанные графики и чертежи преобразуются в последовательности двумерных координат;

- манипуляторы (устройства указания): джойстик — рычаг, мышь, трекбол - шар в оправе, световое перо и др. — для ввода графической информации на экран дисплея путем управления движением курсора по экрану с последующим кодированием координат курсора и вводом их в ПК;

- сенсорные экраны - для ввода отдельных элементов изображения, программ или команд с полиэкрана дисплея в ПК.

К устройствам вывода информации относятся:

- принтеры — печатающие устройства для регистрации информации на бумажный носитель; современные принтеры: матричные, лазерные, струйные (черно-белые и цветные);

- графопостроители (плоттеры) — для вывода графической информации (графиков, чертежей, рисунков) из ПК на бумажный или пленочный носитель; плоттеры бывают векторные с вычерчиванием изображения с помощью пера и растровые: термографические, электростатические, струйные, лазерные и режущие. По конструкции плоттеры подразделяются на планшетные и барабанные. Основные характеристики всех

плоттеров примерно одинаковы, в частности скорость вычерчивания составляет 100-1000 мм/с. Для лучших моделей характерны цветное изображение и передача полутонов, наиболее высокую разрешающую способность и четкость изображения обеспечивают лазерные плоттеры.

Устройства связи и телекоммуникации используются для связи с приборами и другими средствами автоматизации (согласователи интерфейсов, адаптеры, цифро-аналоговые и аналого-цифровые преобразователи и т.п.), а также для подключения ПК к каналам связи, к другим ЭВМ и вычислительным сетям (сетевые интерфейсные платы, "стыки", мультиплексоры передачи данных, модемы).

Многие из названных выше устройств относятся к условно выделенной группе — средствам мультимедиа.

Средства мультимедиа — это комплекс аппаратных и программных средств, позволяющих человеку общаться с компьютером, используя самые разные, естественные для себя среды: звук, видео, графику, тексты, анимацию и др.

К средствам мультимедиа относятся: устройства речевого ввода и вывода информации; сканеры, в силу того, что они позволяют автоматически вводить в компьютер печатные тексты и рисунки; высококачественные видео- и звуковые платы, платы видеозахвата, снимающие изображение с видеомонитора или видеокамеры и вводящие его в ПК; высококачественные акустические и видеовоспроизводящие системы с усилителями, звуковыми колонками, большими видеоэкранами; внешние запоминающие устройства большой емкости на оптических дисках, используемые для записи звуковой и видеоинформации.

6. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ

6.1. Классификация программного обеспечения ЭВМ

Все существующие программы можно разделить на три вида. Для наглядности такого деления изобразим классификацию программного обеспечения (ПО) на рис. 6.1.

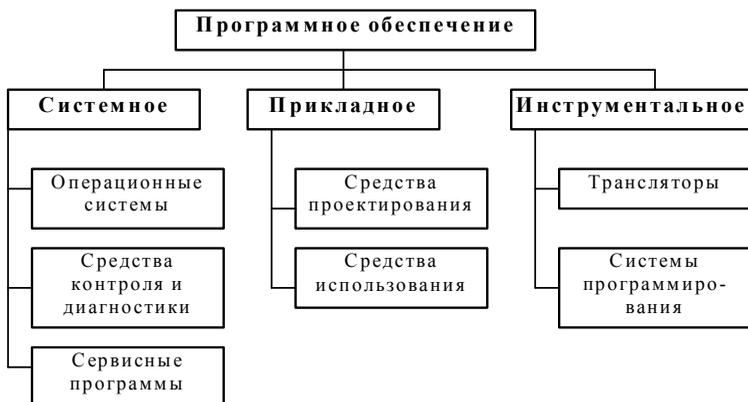


Рис. 6.1. Классификация программного обеспечения ЭВМ

Системное ПО служит для обеспечения работоспособности ЭВМ и общего управления ее ресурсами. К системному ПО относятся:

- **Операционные системы (ОС)** — совокупность программ, предназначенных для управления ресурсами ЭВМ, организации диалога пользователя с ЭВМ, выполнения программ пользователя.

- **Средства контроля и диагностики** — тестовые программы, служат для контроля работоспособности аппаратной части ЭВМ. Эти программы выявляют неисправности и место их возникновения.

- **Сервисные программы** (утилиты) расширяют возможности ОС (программы оболочки, архиваторы, антивирусные

программы, программы восстановления стертых файлов и т.д.).

Инструментальное ПО — совокупность программ для разработки других программ. Они являются основой работы программистов. К ним относятся:

- **Системы программирования** — набор программных средств, которые обеспечивают потребности при разработке программ (редакторы текстов программ, интегрированные среды). Примерами таких программ могут служить Turbo Pascal, Borland Delphi, Microsoft Visual Studio, Kdevelop, Anjuta.

- **Трансляторы** — программы перевода программ в машинный код. Программа–транслятор входит в состав системы программирования. Однако существует и независимые отдельные программы–трансляторы.

Прикладное ПО — программы для решения конкретных задач пользователя. Эти программы обращены к человеку, который не составляет программ, а лишь использует их для решения своих задач. Прикладные программы ориентированы на создание максимального комфорта для пользователя. Он может знать о компьютере минимальный набор сведений. Таки образом, эти программы доступны широкому внедрению среди людей, не знакомых с принципами работы компьютера.

Прикладное программное обеспечение подразделяется на средства проектирования и средства использования, определяет разнообразие информационных технологий и состоит из отдельных прикладных программ или пакетов, называемых приложениями (рис. 6.2.).



Рис. 6.2. Классификация средств проектирования и использования

К средствам проектирования можно отнести: системы управления базами данных (СУБД), экспертные системы, системы автоматизации проектирования (САПР), типовые пакеты прикладных программ (ППП), системы электронного документооборота (СЭД), информационные хранилища, геоинформационные системы (ГИС), системы искусственного интеллекта.

СУБД предназначены для проектирования баз данных. Если база данных размещается на нескольких компьютерах, то используются технологии распределенной обработки данных.

Экспертные системы и информационные хранилища предназначены для облегчения принятия решения.

Типовые пакеты прикладных программ — комплекс взаимосвязанных программ для решения задач определенного класса конкретной предметной области.

Системы электронного документооборота позволяют реализовывать безбумажные технологии на предприятии.

Геоинформационные системы дают возможность моделировать сложные процессы и системы по управлению природными богатствами, экологией, информационной политикой правительств.

Системы искусственного интеллекта позволяют моделировать деловые процессы, производственные и социальные технологии.

Средства использования зависят от типа обрабатываемой информации. Для обработки текстовых документов служат текстовые процессоры.

Текстовые процессоры обладают следующим стандартным набором возможностей: автоматическое форматирование документов, вставка рисованных объектов и графики, составление оглавлений и указателей, проверка орфографии, шрифтовое оформление, подготовка шаблонов документов. Развитием данного направления программных продуктов являются издательские системы.

Данный класс программ включает программы, обеспечивающие информационную технологию компьютерной издательской деятельности; форматирование и редактирование текстов; автоматическую разбивку текста на страницы; создание заголовков; компьютерную верстку печатной страницы; монтирование графики; подготовку иллюстраций и т.п.

Графическая информация обрабатывается посредством графических процессоров.

Удобным средством обработки табличных документов являются табличные процессоры. Табличный процессор — удобная среда для вычислений силами конечного пользовате-

ля; средства деловой графики, специализированная обработка (встроенные функции, работа с базами данных, статистическая обработка данных и др.).

Обмен информацией между удаленными пользователями осуществляется посредством электронной почты. Для анализа информации применяются пакеты прикладных программ (ППП), реализующие многие экономико-математические, статистические методы, методы прогнозирования и выявления тенденций. Созданы и широко используются следующие ППП: ППП автоматизированного бухгалтерского учета; ППП финансовой деятельности; ППП управления персоналом (кадровый учет); ППП управления материальными запасами; ППП управления производством; банковские информационные системы и т.п.

Так как большинство приложений применяются на одном рабочем месте, их объединяют в интегрированные пакеты прикладных программ для более удобной обработки смешанной информации.

Для обработки текстовой информации, структурно представленной в виде сети, применяется гипертекстовая технология. Она же послужила инструментом для создания технологии мультимедиа, где одновременно обрабатываются звуковая, видеоинформация, тексты, движущаяся и неподвижная графика.

6.2. Операционные системы, понятие и назначение

Операционные системы (ОС) — совокупность программ, предназначенных для управления ресурсами ЭВМ, выполнения программ пользователя и организации диалога пользователя и ЭВМ. ОС управляет работой других программ и выделяет им ресурсы — память, время, дисковое пространство и т.д. Она обеспечивает работу машины как единого целого, связь между устройствами. В то же время ОС выступает посредником между человеком и ЭВМ и скрывает от пользовате-

ля много трудных элементарных операций, создает удобный интерфейс.

6.2.1. Классификация операционных систем

Принято классифицировать операционные системы по ряду признаков. По количеству одновременно обрабатываемой информации существуют 16-, 32-, 64-разрядные ОС. По количеству одновременно работающих пользователей ОС подразделяют на однопользовательские и многопользовательские. По количеству одновременно выполняемых программ операционные системы делятся на однозадачные и многозадачные.

В зависимости от работающей ОС на ЭВМ можно одновременно выполнять одну или сразу несколько программ. Например, набирая текст, пользователь может одновременно запустить воспроизведение музыки, ожидать соединения по сети и т.д.

В однозадачной ОС в каждый момент времени может выполняться только одна программа, не считая резидентных программ, которые постоянно присутствуют в памяти и включаются в работу по мере необходимости.

В многозадачной одновременно активными могут быть более одной программы. Многозадачность бывает кооперативной и вытесняющей. При вытесняющей многозадачности система отводит каждой задаче строго определенной время — режим разделения времени.

При кооперативной все программы получают столько времени, сколько они попросят у ОС и делят его между собой, обмениваясь сообщениями.

6.2.2. Обзор существующих операционных систем

Первые ОС появились во времена ЭВМ третьего поколения, когда ощущалась острая нехватка программного обеспечения. Программы в то время в основном разрабатывались свободными объединениями программистов-энтузиастов и бесплатно раздавались всем нуждающимся вместе и исходным кодом на одном из языков программирования. Положение изменилось с появлением вначале 80-х годов персональных компьютеров. Разработка программного обеспечения стала экономически выгодной сферой деятельности. Появились коммерческие ОС, распространяемые в виде готовый исполняемых программ.

Рассмотрим несколько основных направления развития ОС. Наибольшее распространение среди российских пользователей получили ОС, разработанные компанией Microsoft (MS). Первая из них MS DOS 1.0 (дискетная операционная система) появилась в 1981 году и относилась к однопользовательским однозадачным, 16-разрядным ОС. Работа с ОС осуществлялась с помощью командной строки в символьном режиме. ОС обслуживала 64 Кб памяти, могла использовать дискеты 5,25'' емкостью 160 Кб. В 1983 году вышла в свет MS DOS 2.0. Она работала с дискетами 360 Кб и жесткими дисками 10 Мб. Появилась древовидная система каталогов диска, а также возможность загружать драйверы.

Первые ОС обладали рядом принципиальных недостатков: примитивный пользовательский интерфейс (в виде командной строки и символьного режима), однозадачность, ограниченность объема адресуемой памяти. От версии к версии эти ОС совершенствовались, но указанные недостатки оставались. С 1984 года началась работа над графической ОС, которая обладала бы возможностью полного использования памяти, была бы многозадачной. Вместе с тем существовала необходимость разработки 32-разрядной ОС, поскольку к тому времени персональные компьютеры стали оснащаться 32-

разрядными процессорами. Была разработана графическая оболочка к MS DOS, названная Windows. Появился графический интерфейс, но ограничение прямой адресации всего 640 Кб памяти оставалось. Вся остальная память считалась расширенной, и работа с ней осуществлялась с помощью драйверов расширенной памяти. В 1990 году фирма MS выпустила оболочку Windows 3.x. Для неё характерно использование всего адресного пространства и виртуальной памяти, кооперативная многозадачность. Однако, это была не полноценная графическая ОС, а только графическая программная оболочка для MS-DOS.

В 1993 году появилась Windows NT — 32-разрядная, многозадачная ОС с сетевыми возможностями и графическим интерфейсом. Потомком этой системы является ОС Windows 2000, которые также является 32-разрядной многозадачной. Как Windows NT, так и Windows 2000 существуют в виде нескольких разновидностей, среди которых есть однопользовательские и многопользовательские системы. К примеру, Windows 2000 Server является многопользовательской, а Windows 2000 Professional — однопользовательской ОС.

Параллельно развитию технологии Windows NT разрабатывалась альтернативная технология Win32. На ее основе выпущена в 1995 году 32-разрядная, однопользовательская ОС Windows 95. Она обладает вытесняющей многозадачностью для 32-разрядных приложений и кооперативной — для 16-разрядных. Требуемые ресурсы: 386 процессор и 4 Мб памяти, 40 Мб свободного места на жестком диске. Ее потомками являются Windows 98 и Windows Millennium Edition (ME).

Объединение операционных систем линии Windows 95,98, ME с семейством ОС, основанных на технологии NT привело к созданию 32-разрядной однопользовательской ОС Windows XP в трех редакциях: Home, Professional, Media Center, а также 32-разрядной многопользовательской ОС Windows 2003 Server.

Широкое распространение 64-разрядных процессоров привело к необходимости создания 64-разрядных операционных систем. Этим обусловлена разработка Windows XP 64 bit Edition. Наконец, последняя на сегодняшний день разработка фирмы Microsoft ОС Windows Vista тоже существует в нескольких редакциях: для домашних ПК, для бизнеса, для серверов, для 32- и для 64-разрядных ПК.

Второе направление развития операционных систем связано с разработками программного обеспечения фирмы IBM, которая выпустила ОС PC DOS, аналогичную MS DOS. К 1992 году фирмой IBM была разработана операционная система OS/2 — 32-разрядная, многозадачная, однопользовательская ОС, аналогичная по характеристикам Windows 95. Поддерживались многопоточные программы, совместимость с продуктами фирмы MS. Системные требования: 386 процессор, 8Мб памяти, 50Мб свободного места на жестком диске. Но в дальнейшем разработчики этой линии ОС отказались от поддержки программного обеспечения фирмы MS, что по-видимому явилось причиной завершения дальнейших разработок.

Отдельно от перечисленных развивалась еще одна ветвь — ОС UNIX. В своем первоначальном виде UNIX была создана программистами компании Bell Laboratories (подразделение компании AT&T) Кеном Томпсоном и Денисом Ричи в 1969 году. Будучи написанной на языке C, система была пригодна (после перекомпиляции) для работы на различных компьютерных платформах. Первоначально создатели UNIX не ставили коммерческих целей при разработке системы, тем более что компания AT&T специализировалась на разработке технических решений в области телекоммуникаций и не вела коммерческой деятельности в компьютерной области. Исходные тексты UNIX безвозмездно передавались всем нуждающимся и подвергались дальнейшей доработке. Так возникли многочисленные «клоны» UNIX. Большое влияние на развитие системы оказали сотрудники Калифорнийского Университета в г. Беркли. Поначалу они распространяли в исходных кодах дополне-

ния к ОС UNIX. Затем этот проект вырос в самостоятельную операционную систему. В дальнейшем компания AT&T образовала дочернюю компанию UNIX System Laboratories с целью развития коммерческого направления UNIX-систем. На сегодняшний день существует довольно много ОС потомков UNIX как коммерческих, так и свободных. Охарактеризуем некоторые из них.

ОС Solaris, разработанная компанией Sun Microsystems, является свободной системой. Отличительной ее особенностью является оптимизации работы на многопроцессорных системах. ОС HP-UX компании Hewlett-Packard и ОС AIX компании IBM являются наиболее близкими «клонами» исходных классических систем UNIX. На исходных лицензионных текстах UNIX базируется ОС SCO UNIX компании SCO.

Операционные системы семейства BSD (Berkeley Software Distribution) разработаны в Калифорнийском Университете г. Беркли. В ходе своей эволюции ОС BSD фактически были написаны заново с использованием идеологии UNIX. Проект BSD имеет коммерческую (BSDI) и некоммерческую ветви. К некоммерческой ветви относятся широко известные на сегодняшний день системы FreeBSD, OpenBSD и NetBSD.

К семейству UNIX следует также отнести ОС Linux, которая пользуется в наши дни довольно большой популярностью. Ее появление датируется 1991 годом, однако имеется некоторая предыстория появления Linux. В 1984 году был основан Фонд Свободного программного обеспечения (Free Software Foundation, FSF). Основателем FSF был Ричард Столлман, американский ученый, глубоко убежденный, что любое программное обеспечение должно быть свободным для распространения, модификации и изучения. Вся совокупность программного обеспечения, разрабатываемая Фондом, получила название проект GNU (что означает «не принадлежащий UNIX»), поскольку даже само название UNIX уже к тому времени стало зарегистрированной товарной маркой). Программы проекта GNU распространяются на условиях лицензии GPL

(General Public License). Суть этой лицензии на русском языке передать значительно проще, чем на английском. Программа, распространяемая на основе GPL лицензии, является не только бесплатной, но и свободной. Это означает, что программу можно свободно копировать распространять, модифицировать ее исходный код. Модифицированную программу можно не только распространять свободно, но и продавать за любую сумму, сохранив при этом открытым ее исходный код. Соккрытие исходного кода, по мнению автора лицензии, останавливает ее свободное развитие и совершенствование. В рамках проекта GNU было разработано большое количество программ. Для того чтобы из проекта GNU родилась полноценная операционная система не хватало лишь ядра самой этой системы.

Ядро было написано в 1991 году молодым финским программистом Линусом Торвальдсом (университет г. Хельсинки). По словам Торвальдса, он просто развлекался со своим компьютером, занимался любимым делом. На основе ядра Линуса и программного обеспечения проекта GNU и родилась операционная система Linux. Распространяемая на условиях GPL, она имела и имеет условия для своего совершенствования усилиями программистов всего мира. В этой связи ОС Linux имеет широкий спектр версий. Современные версии относятся к многопользовательским, многозадачным, разработанным для разных компьютерных платформ ОС.

7. ОРГАНИЗАЦИЯ ДИСКОВОГО ПРОСТРАНСТВА

7.1 Геометрия диска

Накопители на жестких магнитных дисках (НЖМД, HDD или просто жесткие диски) являются основными внешними запоминающими устройствами персонального компьютера. Почти всегда файлы операционной системы хранятся на жестком диске, загружаются в память при включении ПК, считываются по мере необходимости в ходе работы. На жестком диске также располагаются так называемые файлы подкачки, которые представляют собой «продолжение» оперативной памяти и задействуются согласно нуждам ОС.



Рис. 7.1. НЖМД

Жесткий диск представляет собой одну или несколько пластин с магнитным покрытием, вращающихся на одной оси (рис. 7.1). Каждая поверхность диска имеет свою магнитную головку для чтения и записи информации. Головка снабжена кронштей-

ном, перемещающим ее вдоль радиуса диска. Таким образом, при вращении диска информация записывается по concentрическим дорожкам. Несколько дорожек равного радиуса на разных поверхностях пластин образуют цилиндр. Каждая дорожка радиальным сечением разбита на секторы. Объем каждого сектора равен 512 байт. Для описания конструкции диска необходимо указать число цилиндров C , число дорожек в цилиндре H и число секторов на дорожке S . Совокупность этих трех чисел определяет геометрию диска. Нумерация секторов на дорожке начинается с 1, а нумерация дорожек и цилиндров —

с 0. Чтобы считать или записать информацию на диск, необходимо указать с каким сектором какой дорожки какого цилиндра мы собираемся работать.

Однако использование CHS-геометрии диска быстро устарело. Действительно, радиальное разбиение дорожек на секторы приводило к тому, что плотность записи росла при перемещении к центру диска. По мере развития технологии производства жестких дисков плотность записи данных на диск существенно увеличилась. Разработчиками было принято решение отказаться от радиального разбиения и число секторов на дорожке стало зависеть от ее длины. Геометрию CHS была заменена на линейную адресацию блоков (Linear Block Addressing, LBA), согласно которой все секторы диска пронумерованы начиная с 0. Задачу позиционирования головки над сектором с указанным номером возложили на аппаратуру жесткого диска и его контроллера. Нулевой сектор диска (первый сектор нулевой дорожки нулевого цилиндра) может выполнять особую роль — содержать главную загрузочную запись диска (Master Boot Record, MBR), начальный загрузчик и таблицу разделов диска, о которых речь пойдет ниже.

7.2. Разделы на диске

Дисковое пространство накопителя принято делить на разделы. Это позволяет во-первых установить на разные разделы разные ОС, во-вторых преодолеть возможные ограничения максимального размера диска, присущее ранним версиям ОС. Информация об используемом разбиении диска хранится в нулевом секторе диска и называется таблицей разделов.

Таблица разделов занимает 64 байта и состоит из 4 записей по 16 байт в каждой для 4 первичных разделов. Каждая запись содержит следующую информацию.

1. Признак активности раздела (1 байт). Активным называется раздел, с которого происходит загрузка ОС семейства Microsoft. Для признака активности предусмотрено два

значения: 0 — раздел неактивный, 80₁₆ — раздел активный. Активным может быть только один первичный раздел.

2. CHS первого сектора раздела (3 байта).
3. Тип раздела (1 байт). Для каждой ОС характерны свои типы разделов на диске. Все существующие типы разделов имеют свой однобайтовый код.
4. CHS последнего сектора раздела (3 байта).
5. Номер начального сектора (4 байта) для дисков с LBA.
6. Количество секторов в разделе (4 байта) для дисков с LBA.

Таким образом, на диске может быть создано от 1 до 4 первичных разделов. Для создания более 4 разделов необходимо один и только один из первичных разделов назначить расширенным (тип раздела 5₁₆, F₁₆, 85₁₆). Расширенный раздел служит контейнером для хранения так называемых логических разделов. Первый сектор расширенного раздела содержит свою локальную таблицу разделов. Ее структура совпадает со структурой глобальной таблицы разделов, но используются только первые две записи. Первая из них задает характеристики логического раздела, вторая — характеристики следующего вложенного расширенного раздела. Получается цепочка вложенных друг в друга расширенных разделов. Последний из них содержит только один логический раздел. Пример разбиения диска изображен на рис. 7.2.

MBR	Первичный раздел Диск C: Активный	Первичный раздел Диск D:	Расширенный раздел			
			Табл. разд.	Логический раздел Диск E:	Расш. разд.	
					Табл. разд.	Логический раздел Диск F:

Рис. 7.2. Пример разбиения диска на разделы

Использование дисковых разделов отличается в разных ОС. Так, например, MS-DOS и Windows назначают каждому

разделу однобуквенное имя С, D, E, и т.д (буквы А и В зарезервированы для дисководов). ОС семейства UNIX имеют единую файловую систему, к папкам которой монтируются разделы.

Разбиение дискового пространства осуществляется программой fdisk для ОС семейства Microsoft и программами fdisk, cfdisk или sfdisk для ОС семейства UNIX. Необходимо учесть, что работа с разделами диска может повлечь за собой **уничтожение всех имеющихся на диске данных**. Альтернативным способом управления разделами диска является использование программ сторонних производителей, таких как Power Quest Partition Magic или Acronis Partition Expert. Указанные программы позволяют управлять разделами диска без потери данных.

7.3. Понятие файловой системы

Вся информация на дисках хранится в виде файлов. Файл — это совокупность логически связанных данных. Это может быть текст книги, программа, коды графического изображения и т.д. Поскольку файлов на диске может быть довольно много, для удобства их можно группировать в каталоги (в терминологии Windows — папки). Каждый каталог может содержать файлы и вложенные в него каталоги. Таким образом, формируется иерархическая или древовидная система файлов и каталогов, с которой работает пользователь. Однако логика размещения файлов в секторах диска может быть разной, и определяется она типом файловой системы (ФС). Предназначение файловой системы как логического понятия заключается в том, чтобы сформулировать правила сопоставления имени файла с номерами секторов на диске, в которых этой файл размещен. Выполнить такое сопоставление можно различными способами. Помимо хранения файлов вместе с их атрибутами, датой создания, информацией о владельцах и правах доступа и т.д., ФС должна обеспечивать целостность данных, контроли-

ровать идентичность процессов записи и чтения. Наибольшую опасность для целостности файловой системы представляет внезапное отключение питания во время операции записи на диск.

Перед использованием дискового раздела на нем необходимо создать файловую систему. Процесс создания ФС называется форматированием.

Рассмотрим наиболее распространенные на сегодняшний день файловые системы.

Файловая система FAT была разработана в конце 70-х годов и применялась в ОС MSDOS. После форматирования с использованием ФС FAT раздел диска будет иметь структуру, показанную на рис. 7.3.

Загруз. сектор	FAT	Копия FAT	Корневой каталог	Область файлов
-------------------	-----	--------------	---------------------	----------------

Рис. 7.3. Структура раздела с файловой системой FAT

Первый сектор раздела считается загрузочным, его назначение будет описано ниже. ФС FAT не способна работать с каждым сектором диска. Поэтому смежные секторы объединяются в кластеры. Кластер представляет собой минимальную единицу хранения информации, а также определяет минимальный размер файла. Начиная со второго сектора располагается таблица размещения файлов (File Allocation Table, FAT) и ее копия. С ее помощью происходит сопоставление имени файла и номеров кластеров, в которых размещается файл. Каждому кластеру соответствует запись в FAT, в которой указано состояние кластера: свободен, занят, помечен как сбойный. Изначально для нумерации кластеров в FAT таблице использовались 12 разрядов, а размер кластера был фиксирован и равен 4Кб. Затем разрядность была увеличена до 16 и введен переменный размер кластера, зависящий от размера дискового раздела, но не превышающий 32Кб. Это позволяло адресовать 65535 кла-

стеров на одном разделе. Максимальный размер раздела, поддерживаемый ФС FAT 65535*32Кб=2Гб.

Конструкция имен файлов и каталогов также определяется файловой системой. Имя файла или каталога в FAT состоит из двух частей: собственно имени и расширения, отделяемых друг от друга точкой. Имя должно начинаться с латинской буквы или цифры и может содержать любые символы из кодовой таблицы ASCII кроме пробела, точки и символов "\[|:;|=,^*?". Длина имени не превышает 8 символов. Расширение начинается с точки, за которой следуют от 0 до 3 символов, то есть расширение может отсутствовать. Регистр символов в именах файлов и каталогов не различается.

В ходе работы происходит запись одних файлов, удаление других. При этом кластеры, соответствующие удаляемым файлам помечаются как свободные. В результате свободные кластеры оказываются разбросанными по диску. Запись нового файла на диск начинается с поиска самого первого свободного кластера в таблице размещения файлов. Часть записываемого файла записывается в свободный кластер. Затем происходит поиск следующего свободного кластера и т.д. Если кластеры, содержащие данные файла, расположены не подряд, то файл оказывается фрагментированным. Наличие сильно фрагментированных файлов заметно снижает эффективность работы файловой системы, поэтому время от времени требуется выполнять дефрагментацию диска с помощью специальных программ дефрагментации.

Файловая система VFAT (Virtual FAT) является модификацией ФС FAT, включающая поддержку длинных имен файлов и каталогов. Имя может иметь длину от 1 до 255 любых символов (кроме "\[|:;|=,^*?"), включая пробел, национальные символы и точку. Расширением считается часть имени после последней точки. Модификации ФС коснулись лишь структуры таблицы FAT. Для каждого файла и каталога в VFAT хранится два имени: длинное и короткое в формате 8.3

для совместимости со старыми программами. ФС VFAT используется в ОС Windows 95 в качестве основной.

Файловая система FAT32 является усовершенствованием ФС VFAT с целью поддержки разделов размером больше 2Гб. Структура раздела с ФС FAT32 соответствует рис. 7.3, за исключением следующих усовершенствований. Загрузочная область состоит не из одного, а из двух секторов, второй из которых является копией первого. Для идентификации кластеров используются 32 бита, что позволяет использовать разделы размером до 8Тб. Расширены атрибуты файлов, позволяющие хранить время и дату создания, модификации и последнего доступа к файлу или каталогу. ФС FAT32 выбрана в качестве основной в ОС Windows 98 и Windows Millennium Edition.

Перечисленные файловые системы спроектированы для однопользовательских ОС. Поэтому в них не предусмотрено хранение информации о владельце файлов и о разрешениях доступа. Не предусмотрено также никаких механизмов, защищающих данные от внезапного отключения электроэнергии.

Файловая система NTFS (New Tehnology File System) считается основной ФС в ОС Windows NT, 2000, XP и обладает значительно большими возможностями по сравнению с ранее рассмотренными системами. Размер NTFS раздела теоретически не ограничен. Дисковое пространство разделено на кластеры, размер которых выбирается от 512 байт (1 сектор) до 64Кб. Стандартным считается размер кластера равный 4Кб.

Структура NTFS раздела показана на рис. 7.4.

Область MFT	Область данных	Копия части MFT	Область данных
----------------	-------------------	-----------------------	----------------

Рис. 7.4. Структура раздела с файловой системой NTFS

Служебная информация о размещении файлов хранится в главном файле MFT (Master File Table - общая таблица фай-

лов). Он размещается в MFT зоне и содержит записи фиксированного размера. Каждая запись файла MFT соответствует одному файлу, включая также служебные файлы, расположенные в области MFT. Первые 16 файлов этой области называются метафайлами. Сам MFT является самым первым метафайлом. Метафайлы всегда располагаются вначале области MFT. Копия первых трех метафайлов хранится ровно посередине диска. Сама область MFT первоначально занимает 12% диска объема раздела и поддерживается пустой для предотвращения фрагментации файла MFT. При заполнении диска размер области MFT уменьшается вдвое, освобождая место для файлов. После освобождения места MFT область снова расширяется.

NTFS относится к так называемым журналируемым ФС. Это означает, что перед выполнением «потенциально опасных» действий, информация о них заносится в журнал, расположенный в области MFT. После успешного выполнения действие в журнале помечается как законченное. Если в ходе выполнения произошел сбой (к примеру отключение питания), то анализ журнала на наличие незаконченных действий позволяет восстановить логическую целостность ФС. Журналирование в NTFS не гарантирует сохранности данных, но обеспечивает логическую целостность самой файловой системы.

Файловая система ext2 является долгое время являлась стандартом для ОС Linux. Ее строение является типичным для файловых систем UNIX и достаточно сложным. Рассмотрим лишь основные принципы функционирования ФС ext2. Все дисковое пространство делится на блоки размером 1, 2 или 4Кб. Блоки объединяются в группы блоков. Информация об атрибутах и о физическом расположении каждого файла находится в информационных структурах, называемых информационными узлами (information node, inode). Каждая группа блоков содержит свою таблицу inode. Основным элементов файловой системы является суперблок, который располагается вначале первой группы блоков. Он содержит информацию об

общем числе блоков и информационных узлов файловой системы, о размере блоков и inode, о количестве блоков и inode в группе и т.д. Суперблок похож по степени своей важности на таблицу размещения файлов ФС FAT. Первые версии ФС дублировали суперблок вначале каждой группы блоков, что приводило к существенным потерям дискового пространства. Затем количество копий суперблока было ограничено. Однако в отличие от таблицы FAT ни суперблок, ни информационные узлы не содержат имен файлов. Соответствие между именами файлов и inode устанавливается с помощью каталогов. Каталог представляет собой особого вида файл, также описываемый с помощью inode. Каталог содержит массив записей о своем содержимом.

С точки зрения пользователя ФС ext2 обладает следующими особенностями:

- различается регистр символов в именах файлов и каталогов;
- отсутствует жесткое понятие расширения;
- разрешается наличие нескольких равноправных имен файлов.

Файловая система ext3 создана на основе ФС ext2 и отличается наличием журналирования. Конвертирование ext2 в ext3 заключается только в создании файла журнала. ФС ext3 осуществляет журналирование не только метаданных и логической структуры ФС, но и журналирование данных. Поэтому ext3 является одной из самых надёжных файловых систем для Linux.

Файловая система ReiserFS — журналируемая файловая система, поддерживаемая многими UNIX и Linux системами. Она отличается высокой скоростью работы с каталогами, содержащими большое количество маленьких файлов.

Процесс создания файловой системы обычно в MS-DOS и Windows системах называется форматированием. Форматирование осуществляется с помощью программы format, вхо-

дящей в комплект ОС. Для форматирования раздела с именем X необходимо выполнить команду

```
format X:
```

Для указания необходимой файловой системы команда `format` обладает ключом `/FS`. Так, например, для форматирования в системе NTFS необходимо выполнить команду

```
format X: /FS:NTFS
```

Использование ключа `/?` Приводит к выводу подробной информации о функционировании команды `format`.

Создание файловой системы `ext2` в ОС Linux производится командой

```
mke2fs ИмяРаздела
```

Опция `-j` той же команды

```
mke2fs -j ИмяРаздела
```

позволяет создать файловую систему `ext3`.

7.4. MBR и процесс загрузки операционной системы

При включении компьютера запускается находящаяся в BIOS программа начального старта `BOOT-ROUTINE`. BIOS с одной стороны относится к аппаратному обеспечению ЭВМ, с другой стороны считается составной частью всех операционных систем. После самотестирования BIOS производит поиск доступного загрузочного устройства. Загрузочными устройствами могут служить приводы `CD` и `DVD`, дисководы, жесткие диски, `Flash`-накопители и т.д. Признаком загрузочного устройства является наличие в первом секторе диска признака того, что он является сектором `MBR`, а также программы, именуемой начальным загрузчиком. Таким образом, в `MBR` играет важную роль в процессе загрузки ОС. Рассмотрим подробнее его структуру.

`MBR` является первым сектором диска, имеет размер 512 байт и может содержать:

- первичный загрузчик размером до 446 байт;
- таблица разделов диска размером 64байта;

- признак того, что сектор есть MBR — 2 байта равные AA55₁₆.

Все ОС фирмы Microsoft используют один и тот же первичный загрузчик. Он записывается в MBR при установке системы или с помощью команды

fdisk /mbr

Назначение первичного загрузчика состоит в том, чтобы проанализировать таблицу разделов и найти раздел, обладающий признаком активности (код 80₁₆ в первой байте записи о разделе). Далее происходит считывание первого сектора активного раздела, где располагается вторичный загрузчик. Вторичный загрузчик считывает в память файлы ОС. Для MS-DOS и Windows 95,98 такими файлами являются io.sys и msdos.sys. Оба файла размещаются на диске всегда первыми. В случае Windows NT, 2000, XP вместо io.sys загружается файл ntldr. Он является мощной программой загрузки следующего уровня (третичный загрузчик). Программа ntldr анализирует файл boot.ini и выводит на экран, если это необходимо, меню загрузки.

Загрузка других ОС отличается от описанного процесса загрузки незначительно, но каждая ОС использует свои программы-загрузчики. Например в ОС Linux наиболее распространены 2 загрузчика: LILO и GRUB.

Загрузчик LILO (LInux LOader) тоже состоит из двух частей: первичного загрузчика в MBR и вторичного загрузчика, который располагается в каталоге /boot раздела Linux. Программа GRUB является более мощным загрузчиком и предоставляет пользователю интерфейс в виде командной строки.

8. КОМАНДНАЯ СТРОКА

Несмотря на то, что современные ОС предоставляет пользователю наглядный графический пользовательский интерфейс (GUI, Graphic User Interface), каждая ОС дает возможность работы с командной строкой. Работа с графическим интерфейсом не представляет трудности. Умение же работать с командной строкой требует определенных знаний и способствует более глубокому пониманию принципов работы ОС.

8.1. Командная строка операционной системы Windows

ОС Windows в отличие от других систем полноценно функционирует лишь при загрузке графического пользовательского интерфейса. Командная строка доступна в виде отдельного приложения. Но вместе с тем существуют задачи администрирования системы, которые можно решить только с помощью командной строки. Особенно это относится к «недокументированным возможностям» системы.

Работа пользователя с командной строкой заключается во вводе команд в текстовом режиме с помощью командной строки. В начале командной строки располагается приглашение ОС, которое, как правило, содержит имя текущего диска и каталога. Например:

```
C:\ТЕХТ>
```

Текущим называется диск или каталог, с которым работает пользователь в данный момент.

В общем виде любая команда выглядит так:

```
C:\ТЕХТ>ИмяКоманды Параметры /ключ1 /ключ2
```

Обязательная часть команды — имя команды, которую надлежит выполнить системе. Некоторые команды нуждаются в дополнительной информации, которую они получают из параметров. Параметры отделяются от имени программы и друг от друга одним или несколькими пробелами. Помимо параметров команды могут использовать ключи, которые модифици-

руют действие команд. Каждый ключ указывается после символа / (прямой слеш). Набирать команду можно и большими, и маленькими буквами. Ввод команды заканчивается нажатием Enter. Подробное рассмотрение всех команд выходит за рамки данной книги. Рассмотрим лишь основные команды.

8.1.1. Команды для работы с каталогами

- 1) Для *смены текущего диска* необходимо набрать
ИмяДиска : Enter

Например: b :

- 2) Для *смены текущего каталога* служит команда
CD [Диск :] Путь

Квадратными скобками помечены необязательные части команды. Путь — это указание, через какие каталоги начиная с корневого или текущего необходимо пройти, чтобы попасть в требуемый каталог. Если путь начинается с символа \, то каталоги отсчитываются от корневого. В противном случае — от текущего. Названия каталогов отделяются друг от друга символом \ (обратный слеш).

**CD ** — переход в корневой каталог,

CD . . — переход в надкаталог.

Например: **cd d:\games\poker.**

- 3) Просмотр *содержимого каталога* осуществляется командой

DIR [Диск : Путь]

Команда сообщает имя, расширение, дату создания или обновления файлов и подкаталогов. Полезно помнить некоторые важные ключи команды DIR, которые модифицируют ее действие. Ключ /r вынуждает команду осуществлять постраничный вывод. Такая необходимость появляется при выводе слишком длинного списка файлов. Ключ /w приводит к выводу списка

файлов в кратком формате. Ключ `/s` означает «с подкаталогами» и одинаково используется многими командами. Каждая команда обладает также ключом `/?`, который означает «вызов помощи». Например, команды

```
dir c:\Program /p
dir c:\Program /w
dir c:\Program /p /w /s
```

выводят на экран информацию о содержимом каталога `Program`, расположенного на диске `C`. Первая команда выводит содержимое поэкранно, вторая — в кратком формате. Третья команда выводит поэкранно в кратком формате информацию о содержимом каталога `Program` и всех его подкаталогов, если они существуют.

Команда `dir` без параметров выводит содержание текущего каталога.

4) *Создание каталога* осуществляется командой

```
MD [Диск:]Путь
```

Здесь `Путь` — путь к новому каталогу.

5) *Удаление каталога* осуществляется командой

```
RD [Диск:]Путь
```

Если каталог не пуст, то удаления не происходит. Необходимо либо сначала очистить каталог — удалить из него все файлы и подкаталоги, либо воспользоваться ключом `/s` для удаления с подкаталогами.

6) Команда **PATN**.

С самого возникновения MS-DOS команды делились на внутренние и внешние. Внутренние команды выполняет командный процессор. Командным процессором называют программу, которая организует общение с пользователем в командной строке. В более ранних версиях ОС командный процессор назывался `command.com`, в современных — `cmd.exe`. Внешние команды операционной системы реализуются про-

граммами, которые хранятся на диске в виде отдельных файлов. То есть любая программа является для командной строки внешней командой. Для того, чтобы командный процессор знал, где искать внешние команды, существует переменная окружения PATH. Она представляет собой строку, где через точку с запятой перечислены названия каталогов с внешними командами. Изменить переменную PATH позволяет одноименная команда

PATH СписокКаталогов через точку с запятой
Например: **PATH c:\windows;c:\Program;d:\Games.**
PATH; — устанавливает только текущий каталог для поиска,
PATH без параметров выводит информацию и текущих установках.

Если программа находится в одном из каталогов, указанных в команде PATH, то для запуска программы достаточно набрать в командной строке имя программы и нажать Enter, в противном случае для запуска программы необходимо указывать полный путь к файлу программы.

8.1.2. Команды работы с файлами

Кратко перечислим основные команды для работы с файлами:

1) **type ИмяФайла** — вывод содержимого файла на экран.

2) **Copy Источник Приемник** — копирование файлов.
Например:

Copy proba.txt b:\text\text.doc.

Файл proba.txt копируется из текущего каталога в каталог text на диске b и ему присваивается новое имя text.doc. если имя менять не требуется, то новое имя писать не нужно:

Copy proba.txt b:\text

Допускается объединение (конкатенация) файлов при копировании

Copy proba.txt+proba1.txt b:\text\text.doc

3) Команду копирования можно применять для создания файлов копированием с клавиатуры:

Copy con File.txt

Тогда набираемый с клавиатуры текст копируется в создаваемый файл. Именем **Con** в Windows называется клавиатура.

4) Аналогичным образом можно печатать файлы. Для этого нужно знать, что стандартное имя принтера **Prn**. Скопируем файл на принтер — он будет распечатан:

Copy proba.txt prn.

Допускается работа с группой файлов. Для задания группы файлов используются шаблоны, которые строятся с использованием символов * и ?. Символ * означает произвольную последовательность произвольных символов. Например, *.* — файл с любым именем, любым расширением, а *.* — файл, который начинается с буквы а с любым расширением и т. д. Символ ? означает один произвольный символ. Например, File?.txt — это файл, в имени которого вместо ? допускается любой символ.

5) Перенос файлов между каталогами осуществляется командой **move Источник Приемник**. Принцип использования этой команды совпадает с командой **Copy**.

6) Для переименования файлов служит команда

Ren СтароеИмя НовоеИмя.

7) Команда **Del ИмяФайла** производит удаление файла. Например,

Del c:\tp\turbo.exe
Del *.*

8.1.2. Команды общего назначения

1) **Cls** — очистка экрана.

2) **Date** — установка даты. При выполнении этой команды на экран выводится текущая дата и выдается просьба ввести новую дату. Можно набрать новую дату и нажать Enter, а можно не меняя даты просто нажать Enter, осведомившись о текущей дате.

3) **Time** — установка времени. Работа команда аналогична команде Date.

4) **Sys ИмяДиска:** — создание системного диска. На указанный диск переносятся системные файлы и загрузчик.

8.2. Командная строка Linux

В ОС Linux командная строка занимает значительно более важное место. Это связано с тем, что все клоны UNIX не связаны с графическим интерфейсом пользователя неразрывно. GUI является не более чем программой, создающей пользователь удобную рабочую обстановку. Но вместе с тем, сама ОС способна функционировать и без наличия GUI. Тогда общение с пользователем может быть организовано с помощью командной строки [11].

Во времена создания ОС UNIX господствовала технология терминального доступа к ЭВМ. Пользователи взаимодействовали с ОС посредством множества терминалов. Терминалом принято называть совокупность монитора и клавиатуры. В современных условиях терминалу принадлежит также мышь. Первоначально терминалы подключались по последовательным интерфейсам ЭВМ. Один из терминалов считался главным, предназначался для системного администратора и назывался консолью. В наше время совокупность устройств для общения с пользователем, подключенных непосредственно к ПК, также называют консолью. Осталась возможность подключать удаленные терминалы по последовательным линиям

связи. Более того, современные системы UNIX и Linux позволяют работать с несколькими виртуальными терминалами на одной консоли.

Для переключения между виртуальными терминалами используется комбинация клавиш Ctrl+Alt+F?, где F? — одна из функциональных клавиш. Каждый из терминалов выдает приглашение на ввод имени пользователя и пароля, после чего появляется приглашения командной строки. Для графического рабочего стола обычно используется терминал с номером 7.

При работе с графическим интерфейсом пользователя командная строка остается доступна в виде отдельного приложения «Терминал».

8.2.1. Команды для работы с каталогами

Перед перечислением основных команд следует сделать несколько замечаний. Linux различает регистр при вводе команд в командной строке. Для доступа к разделам диска Linux не используются однобуквенные обозначения. Все разделы объединяются в единое дерево файлов и каталогов, вершиной которого является корневой каталог (/). Для разделения имен каталогов при записи пути к файлу используется прямой слеш. Каждый раздел диска монтируется к своему каталогу, а один из разделов (корневой раздел) монтируется к корневому каталогу.

1) Для *смены текущего каталога* служит команда

cd Путь

Если путь начинается с символа /, то каталоги отсчитываются от корневого. В противном случае — от текущего. Названия каталогов отделяются друг от друга символом / (прямой слеш).

cd / — переход в корневой каталог,

cd .. — переход в надкаталог.

Например: **cd /home/user**

2) Просмотр *содержимого каталога* осуществляется командой

ls Путь

Команда **ls** без параметров выводит содержание текущего каталога.

3) *Создание каталога* осуществляется командой

mkdir Путь

Здесь Путь — путь к новому каталогу.

4) *Удаление каталога* осуществляется командой

rmdir Путь

Если каталог не пуст, то удаления не происходит. Необходимо либо сначала очистить каталог — удалить из него все файлы и подкаталоги.

8.2.2. Команды работы с файлами

Кратко перечислим основные команды для работы с файлами:

8) **cat ИмяФайла** — вывод содержимого файла на экран. Команду **cat** можно применять для создания файлов копированием с клавиатуры:

cat > File.txt

Тогда набираемый с клавиатуры текст перенаправляется в создаваемый файл. Символ **>** используется в командной строке для перенаправления вывода.

9) **cp Источник Приемник** — копирование файлов. Например:

cp proba.txt /text/text.doc.

Допускается работа с группой файлов. Для задания группы файлов используются шаблоны, которые строятся с использованием символов ***** и **?** по тем же правилам, что и в ОС Windows.

10) Перенос файлов между каталогами осуществляется командой **mv** **Источник** **Приемник**. Принцип использования этой команды совпадает с командой **cp**.

11) Для переименования файлов служит команда **rename** **СтароеИмя** **НовоеИмя**.

12) Команда **rm** **ИмяФайла** производит удаление файла. Например,

```
rm /home/user/turbo.exe  
rm *
```

Следует также заметить, что каждая команда Linux обладает множеством ключей, модифицирующих ее действие, описание которых выходит за рамки данной книги.

9. ПРОГРАММЫ – ОБОЛОЧКИ

Программы-оболочки предназначены для создания удобной рабочей обстановки и в основном направлены на работу с файловой системой.

Программы-оболочки ведут свою родословную от самой известной во времена MS-DOS программы Norton Commander. Эта программа фактически стала стандартом среди пользователей ПК и заменила не вполне удобный интерфейс командной строки интуитивно понятным интерфейсом. Горячие сочетания клавиш, используемые в Norton Commander, стали стандартными почти для всех современных программ-оболочек. Аналогичной программой для UNIX-систем является Midnight Commander. Его внешний вид, принципы работы, сочетания клавиш совпадают с Norton Commander. Внешний вид современного Norton Commander показан на рис. 9.1, а Midnight Commander — на рис. 9.2. Запустить указанные программы можно (помимо использования ярлыка на рабочем столе) с помощью команд `nc` и `mc` командной строки.

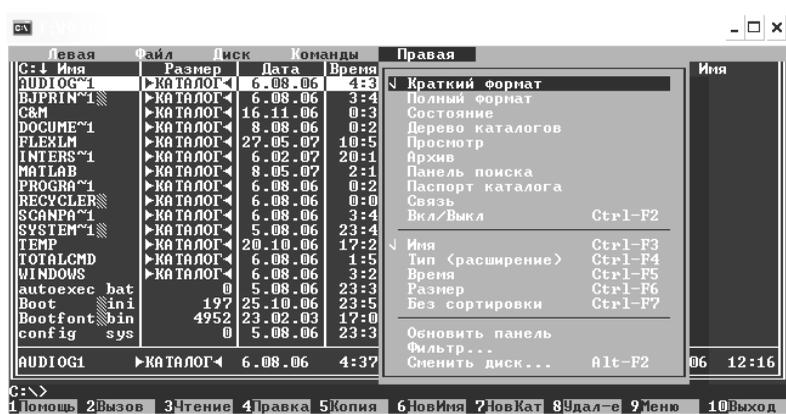


Рис. 9.1. Внешний вид Norton Commander

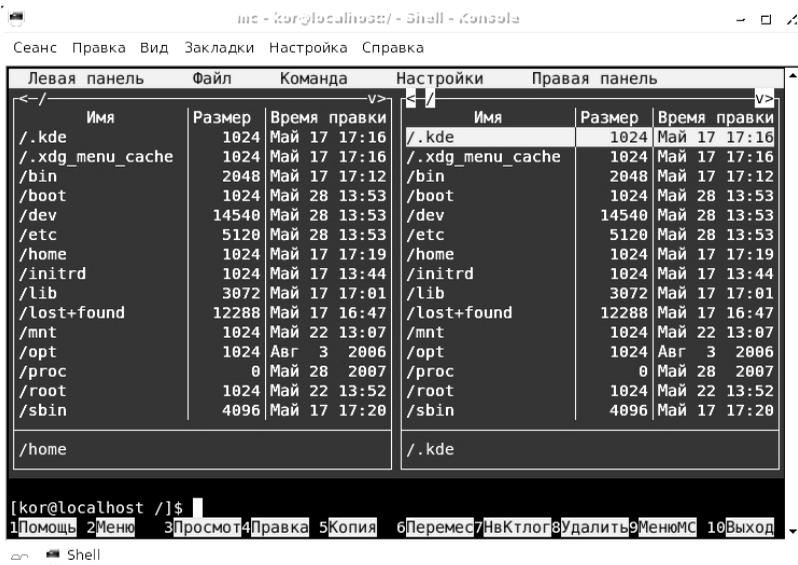


Рис. 9.2. Внешний вид Midnight Commander

Перечислим и кратко охарактеризуем наиболее распространенные современные программы-оболочки.

Total Commander является одной из самых удобных коммерческих оболочек. Ее оконный внешний вид (см. рис.9.3) является привычным для пользователей Windows. Вместе с тем программа обладает богатыми возможностями. В последней версии Total Commander предусмотрена даже запись на CD и DVD диски. Аналогом Total Commander для ОС Linux является программа Gnome Commander (лицензия GPL), изображенная на рис. 9.4.

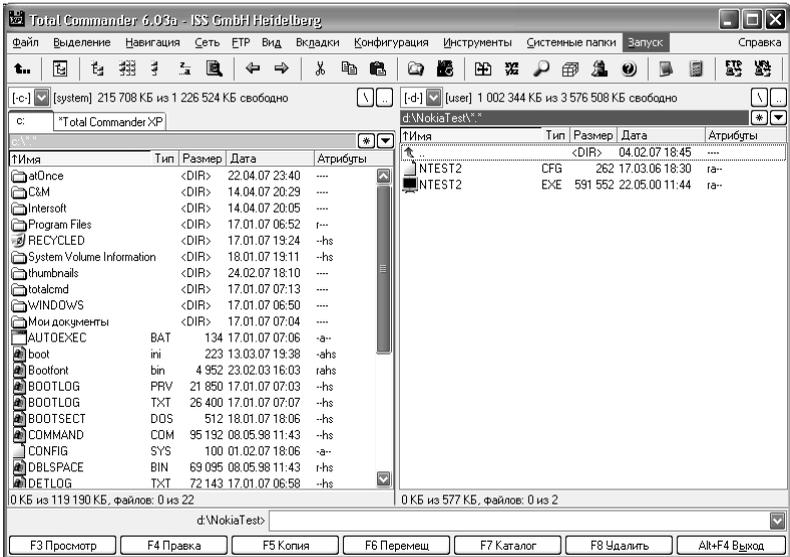


Рис. 9.3. Внешний вид Total Commander

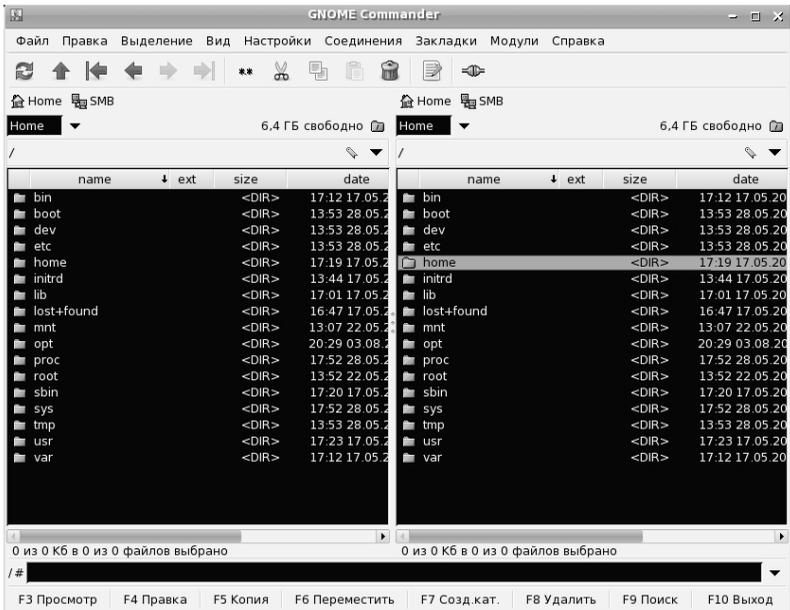


Рис. 9.4. Внешний вид Gnome Commander

Другая не менее удобная программа-оболочка Far manager является бесплатной для жителей бывшего СССР. Внешним видом (см. рис.9.5) она напоминает Norton Commander, но обладает очень гибкими возможностями настройки, с помощью которых ее внешний вид можно изменить до неузнаваемости.

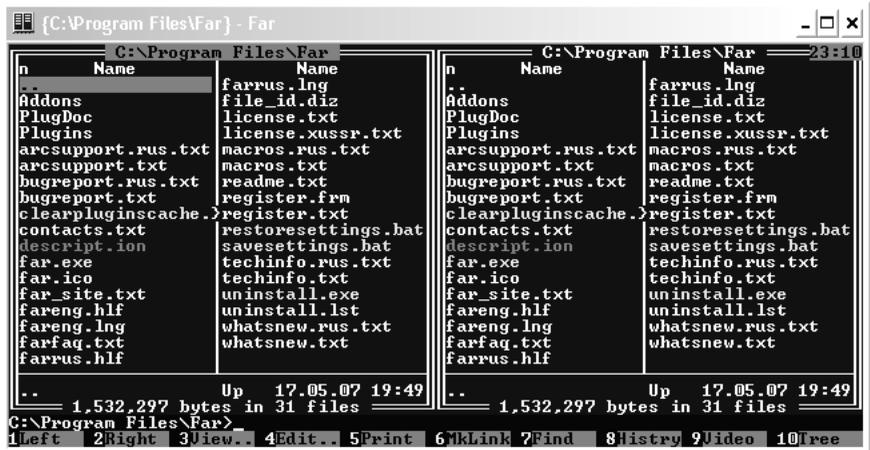


Рис. 9.5. Внешний вид Far manager

Заслуживают внимания еще несколько программ:
 Unreal Commander (рис.9.6), для длительного использования которого необходима регистрация на сайте разработчика;
 File Navigator (рис. 9.7) — простая, удобная, свободно распространяемая оболочка;
 Tux Commander (рис. 9.8) — весьма удобная оболочка для Linux.

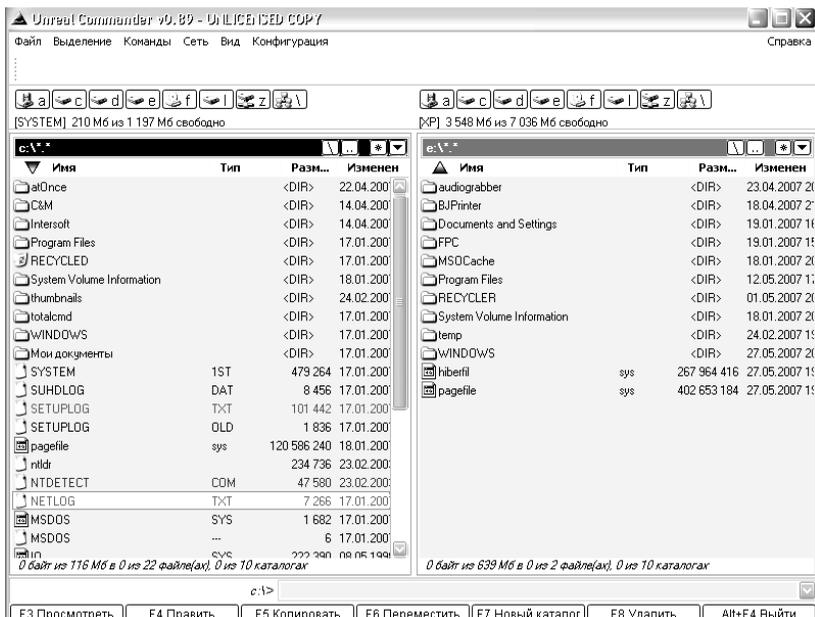


Рис. 9.6. Внешний вид Unreal Commander

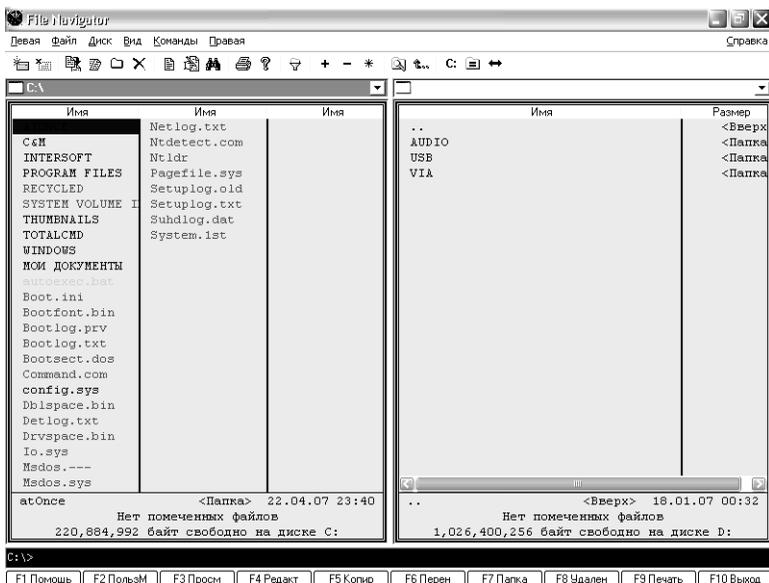


Рис. 9.7. Внешний вид File Navigator

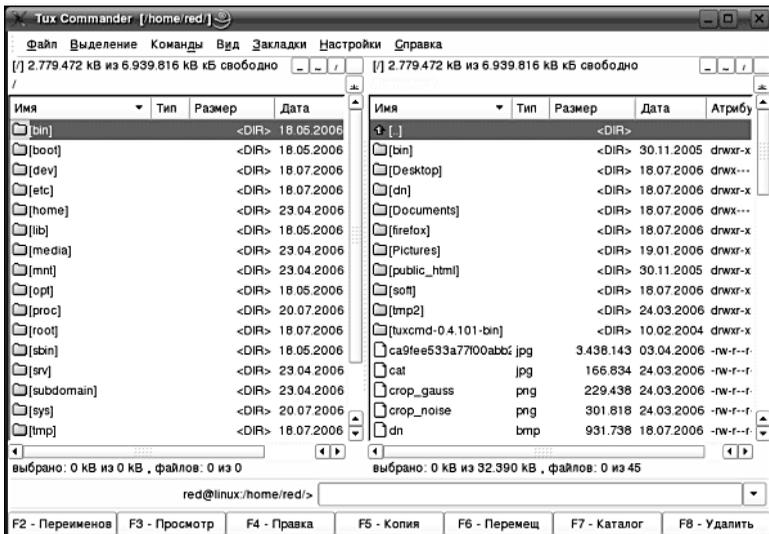


Рис. 9.8. Внешний вид Tux Commander

9.1. Основы работы с программами-оболочками

По историческим причинам интерфейс программ-оболочек различается лишь в деталях.

В верхней строке окна расположено *Главное меню*, которое для удобства может быть скрыто, но его можно активировать нажатием F9. Главное меню включает в себе все возможности программы, отсортированные по более или менее однотипным командам. Покинуть меню можно, нажав клавишу Esc.

Основную часть окна занимают *Панели* – две равноправные таблицы, внутри которых может отображаться различная информация. Одна из панелей является активной (на ней находится курсор, и заголовок панели подсвечен), другая панель – пассивная. Смена активной панели достигается нажатием клавиши Tab или щелчком мыши в любой части панели. При этом курсор перемещается с одной панели на другую. Внутри панели как правило отображается список файлов и каталогов какого-либо текущего каталога. Для управления внешним видом панелей служат пункты меню «Левая», «Правая» и «Вид».

Под панелями располагается *Командная строка*, обладающая высшим приоритетом среди всех элементов экрана программы. При нажатии алфавитно-цифровых клавиш они появляются именно в командной строке. Далее при нажатии Enter действия программы зависят от того, пуста ли командная строка. Если она не пуста, то выполняется набранная команда. Иначе действия зависят от положения курсора на панелях или в меню.

Экран выполнения команд из командной строки отличается в разных оболочках. Некоторые раскрывает в отдельное окно для выполнения команд. Некоторые выдают результат выполнения команд в своем окне под панелями. Временно прикрыть панели можно комбинацией клавиш Ctrl+O. Выполняемые команды запоминаются программой оболочкой, и их можно повторить. Комбинация клавиш Ctrl+E позволяет перелистать команды по одной в командной строке в порядке,

обратном их выполнении. Для выбора одной из выполненных команд можно также воспользоваться журналом команд, к появлению которого приводит нажатие Alt+F8.

В самой нижней части окна расположено **Меню функциональных клавиш**. Оно имеет вид подсказок к функциональным клавишам.

Просмотр содержимого файла. Для просмотра содержимого файла служит клавиша F3. На экране раскрывается окно с содержимым того файла, на котором стоял курсор. Выход из режима просмотра — Esc.

Редактирование файла встроенным редактором. Для редактирования содержимого файла служит клавиша F4. На экране раскрывается окно с содержимым того файла, на котором стоял курсор. Кроме того, содержимое файла можно изменять. При нажатии Shift+F4 сначала появляется диалоговое окно, в котором нужно указать имя того файла, подлежащего редактированию. Если такого файла не существует, то появляется пустое окно для написания содержимого файла. Таким образом, Shift+F4 может служить для создания новых текстовых файлов. Выход из режима редактирования — Esc.

Копирование файла или каталога. Для копирования файлов и каталогов служит клавиша F5. Технология копирования заключается в следующем.

1) Раскрыть в одной панели тот каталог, в который копируем.

2) В другой панели поставить курсор на тот файл или каталог, который копируем.

3) Нажать F5, появится диалоговое окно с параметрами копирования (куда копируем, с каким именем копируем и т.д.).

4) Нажать Enter для подтверждения.

При использовании вместо F5 комбинации Shift+F5 предлагается ввести имя копируемого файла.

Переименование и перенос файла или каталога. Для переноса файлов и каталогов из одного места дерева в другое

служит клавиша F6. Технология переноса заключается в следующем.

- 1) Раскрыть в одной панели тот каталог, в который переносим.
- 2) В другой панели поставить курсор на тот файл или каталог, который переносим.
- 3) Нажать F6.
- 4) Нажать Enter.

Таким образом, перенос файлов и каталогов происходит так же, как и копирование.

Для переименования файла необходимо в диалоговом окне переименование/перенос написать новое имя файла вместо пути, который предлагает Windows Commander для переноса. Необходимо подчеркнуть, что переименование и перенос это практически одно и то же, так как полным именем файла считается его имя с указанием пути к нему от корневого каталога диска.

При использовании комбинации Shift+F6 предлагается переименовать файл или каталог.

Создание нового каталога. Для создания каталога необходимо перейти в каталог, внутри которого будет создаваться новый каталог, и нажать F7. Появится диалоговое окно для ввода имени. После нажатия Enter каталог будет создан.

Удаление файлов и каталогов. Удаление происходит с помощью клавиши F8, после нажатия, которой нужно необходимое количество раз подтвердить удаление.

Работа с группами файлов и каталогов. Зачастую возникает необходимость выполнения операций со многими файлами и каталогами. Тогда их необходимо предварительно выделить. Выделения и снятие выделения для отдельного файла или каталога происходит с помощью клавиши Insert или правым щелчком мыши. Выбранный файл отмечается другим цветом. В низу панели появляются сведения о количестве выбранных файлов и объеме занимаемой ими дисковой памяти.

Имеется возможность выбрать группу файлов по маске, для этого необходимо нажать на клавишу СЕРЫЙ ПЛЮС и задать маску (образец) для выбора. В маске можно использовать символы * и ?. Их смысл тот же, что и командах командной строки. Чтобы отменить выбор группы файлов по маске, нажмите клавишу СЕРЫЙ МИНУС и задайте маску файлов, выбор которых хотите отменить.

Атрибуты файла и способы их изменения.

Каждый файл на диске имеет атрибуты. Работа с атрибутами отличается в разных ОС. Например, в Windows главными считаются 4 взаимонезависимых атрибута: **только чтение, архивный, скрытый, системный**.

Для изменения атрибутов файла предусмотрен пункт меню «Файл», «Изменить атрибуты». В появившемся диалоговом окне (рис. 9.9) необходимо отметить те атрибуты, который необходимо установить или снять. Здесь же можно изменить дату создания файла.

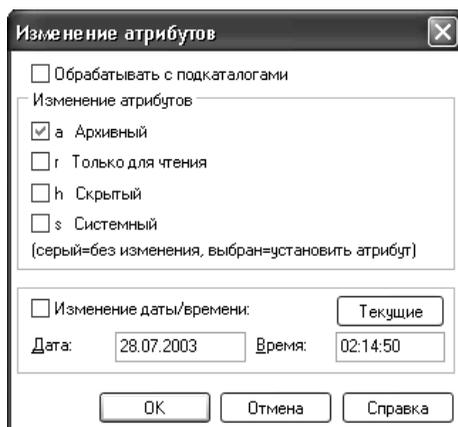


Рис. 9.9. Диалоговое окно изменения атрибутов

Поиск файлов. Для поиска файлов используется пункт меню «Поиск файлов» или комбинация клавиш Alt+F7. В диалоговом окне «Поиска файла» (рис.9.10) следует ввести имя

файла или шаблон для поиска. После поиска формируется список искомых файлов, по которому можно перемещать курсор. Нажатие Enter на одном из найденных файлов приводит к открытию нужной папки с выбранным файлом в одной из панелей.

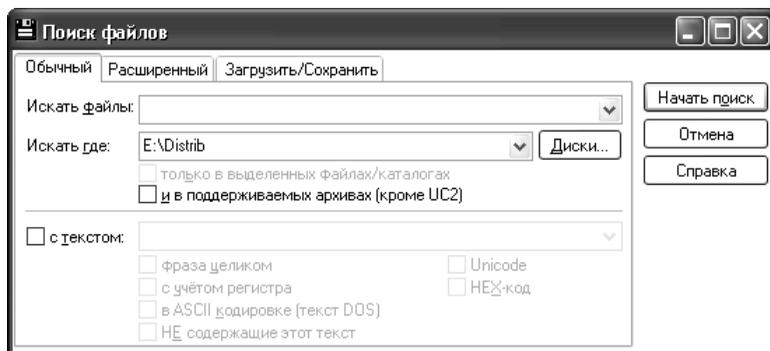


Рис. 9.10. Диалоговом окне «Поиск файла»

Поскольку программы-оболочки имеют схожие элементы управления, основанные на функциональных и горячих клавишах, удобно обобщить назначение этих клавиш в виде таблицы.

Таблица назначения функциональных клавиш

	—	Shift	Alt	Ctrl
F1	вызов помощи		смена диска в левой панели	краткий формат
F2		сравнить каталоги	смена диска в правой панели	подробный формат
F3	просмотр содержимого файла	просмотр содержимого файла	выход из программы	сортировать по имени
F4	редактирование	редактирование с запросом имени	редактирование	сортировать по расширению
F5	Копирование	копирование с запросом имени	упаковать в архив	сортировать по времени модификации
F6	перенос файлов и папок	переименование	распаковать	сортировать по размеру
F7	создание папки	создание папки	поиск файла	не сортировать
F8	удаление	удаление	журнал команд	показать в виде дерева папок
F9	главное меню	аналог правого щелчка мыши	распаковать	распаковать
F10	выход	аналог правого щелчка мыши	показать дерево папок	показать все файлы
F11	—	—	—	показать только программы
F12	—	—	—	фильтр

Горячие клавиши:

- Ctrl+O — вкл/выкл панели;
- Ctrl+U — поменять панели местами;
- Ctrl+E — предыдущая команда в командную строку;
- Ctrl+P — вкл/выкл неактивную панель;
- Ctrl+L — вкл/выкл информационную панель;
- Ctrl+Q — вкл/выкл быстрый просмотр;
- Ctrl+N — показывать/ не показывать длинные имена;
- Ctrl+R — перечитать диск;
- Ctrl+A — атрибуты файла;
- Ctrl+B — вкл/выкл строку подсказок.

ЗАКЛЮЧЕНИЕ

В учебном пособии изложены вопросы, изучаемые в курсе «Информатика» студентами радиотехнических специальностей вузов.

Обсуждаются различные подходы к определению понятий «информация» и «количество информации», а также истоки возникновения информатики как науки и дисциплины.

Показано наличие тесной связи информатики с вычислительной техникой. Приводятся сведения из истории развития электронных вычислительных машин.

Важное место в пособии отведено математическим основам построения ЭВМ и техническим особенностям построения современных персональных компьютеров. Рассмотрены системы счисления, способы представления данных в ЭВМ, архитектура современных персональных компьютеров. Приведены сравнительные характеристики современных процессоров, описаны принципы построения памяти ЭВМ.

Отдельная глава пособия раскрывает вопросы организации дискового пространства, способы деления дискового пространства на разделы. Здесь же обсуждаются особенности использования разделов в разных операционных системах.

Рассмотрена классификация программного обеспечения ЭВМ, и операционных систем. Приведены справочные сведения по работе с командной строкой операционных систем Windows и Linux. Изложены основные принципы работы с программами-оболочками.

Материал учебного пособия может быть использован при изучении специальных курсов, где для выполнения конкретных заданий требуются базовые знания ЭВМ.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Унижаев Н.В. Введение в информатику / Н.В. Унижаев // <http://u-n-v.narod.ru/informatika.htm>.
2. Шауцукова Л.З. Информатика 10-11 / Л.З. Шауцукова. М.: Просвещение, 2000.
3. Красилов А.А. Информатика в семи томах. / А.А. Красилов. М. 2000. Т. 1: Основы информатики (Введение в информатику).
4. Винер Н. Кибернетика или управление и связь в живом и машине / Н. Винер. М.: Советское радио, 1968. 340с.
5. Алексеев А.П. Информатика 2002 / А.П. Алексеев. М.: Солон-Р, 2002.
6. Кузнецов Н.А. Информационное взаимодействие в технических и живых системах / Н.А. Кузнецов // Информационные процессы. №1. 2001. Т. 1. С. 1–9.
7. Колмогоров А.Н. Три подхода к определению понятия «количество информации» / А.Н. Колмогоров // Проблемы передачи информации. 1965. Вып. 1. № 1. С. 3–11.
8. Информатика. Базовый курс / С. В. Симонович и др. Спб.: Питер, 2002. 640с.
9. Корнеев В. Современные микропроцессоры. / В. Корнеев, А. Киселев. 3-е изд. Спб.: БХВ-Петербург, 2003. 448 с.
10. Микропроцессоры: учеб. пособие: в 5 кн. / под ред. В.А. Шахнова. М.: Высш. шк., 1998.
11. Костромин В.А. Самоучитель Linux для пользователя / В.А. Костромин. Спб.: БХВ-Петербург, 2003. 672с.
12. Новейшая энциклопедия персонального компьютера 2002, М.: ОЛМА-ПРЕСС, 2002.

ОГЛАВЛЕНИЕ

Введение	3
1. Информация и знания	6
1.1. Понятие информации	6
1.2. Понятие знания	7
1.3. Информационные процессы	8
1.4. Количество информации	10
2. История развития ЭВМ. Поколения ЭВМ	15
2.1. История развития вычислительной техники	15
2.2. Поколения ЭВМ	18
2.3. История развития отечественной вычислительной техники	20
3. Системы счисления	23
3.1. Классификация систем счисления	23
3.2. Двоичная система счисления	25
3.3. Системы счисления родственные двоичной	27
3.3.1. Восьмеричная система	27
3.3.2. Шестнадцатеричная система	29
3.4. Правила перевода чисел между системами счисления	31
3.5. Арифметические действия в позиционных системах счисления	32
4. Представление чисел в ЭВМ	39
4.1. Прямой, обратный и дополнительный коды	39
4.2. Представление в ЭВМ целых чисел	41
4.3. Представление в ЭВМ вещественных чисел	42
5. Архитектура ЭВМ. Потoki информации в ЭВМ	46
5.1. Основные блоки ЭВМ. Принципы фон Неймана	46
5.2. Процессор	47
5.3. История развития процессоров	50
5.4. Память	60
5.5. Магистрально-модульный принцип построения	64
5.6. Характеристики периферийных устройств	67
6. Программное обеспечение ЭВМ	72

6.1. Классификация программного обеспечения ЭВМ	72
6.2. Операционные системы, понятие и назначение	76
6.2.1. Классификация операционных систем	77
6.2.2. Обзор существующих операционных систем	78
7. Организация дискового пространства	83
7.1. Геометрия диска	83
7.2. Разделы на диске	84
7.3. Понятие файловой системы	86
7.4. MBR и процесс загрузки операционной системы	92
8. Командная строка	94
8.1. Командная строка операционной системы Windows	94
8.1.1. Команды для работы с каталогами	95
8.1.2. Команды работы с файлами	97
8.1.3. Команды общего назначения	99
8.2. Командная строка Linux	99
8.2.1. Команды для работы с каталогами	100
8.2.2. Команды работы с файлами	101
9. Программы – оболочки	103
9.1. Основы работы с программами-оболочками	109
Заключение	116
Библиографический список	117

Учебное издание

Корчагин Юрий Эдуардович
Слинчук Светлана Александровна

ОСНОВЫ ИНФОРМАТИКИ
Часть 1

В авторской редакции
Выпускающий редактор И.В. Медведева

Подписано в печать 20.06.2007
Формат 60x84/16. Бумага для множительных аппаратов.
Усл. печ. л. 7,4. Уч.-изд. л. 6,4. Тираж 100 экз.
Заказ № _____

ГОУВПО «Воронежский государственный технический
университет»
394026 Воронеж, Московский просп., 14