**Содержание**

1 Теоретические основы ЭВМ

1.1 Информация и ее представление

1.2 Системы счисления

1.3 Логические операции

1.4 Единицы измерения информации

2 УСТРОЙСТВО СОВРЕМЕННЫХ ЭВМ

2.1 Схема фон Неймана

2.2 Основные устройства компьютера и их свойства

3 ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

3.1 Типы программного обеспечения

3.2 Файловая система

3.3 Основные операции с файлами. Буфер обмена

4 ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

4.1 Обработка текста. Текстовые процессоры

4.2 Компьютерная графика

4.3 Электронные таблицы

4.4 Реляционные базы данных

5 АЛГОРИТМЫ И ПРОГРАММЫ

5.1 Алгоритмы. Способы записи алгоритмов

5.2 Языки высокого уровня

5.3 Основные операторы и синтаксические конструкции

6 Компьютерные телекоммуникационные сети

6.1 Основные принципы организации современных компьютерных сетей

6.2 Служба Domain Name System (DNS)

6.3 Почтовая служба (E-mail)

6.4 Служба File Transfer Protocol (FTP)

6.5 Служба World Wide Web (WWW)

6.6 Ресурсы в сети

**1 Теоретические основы ЭВМ**

## 1.1 Информация и ее представление

*Информатика* — научное направление, занимающееся изучением законов, методов и способов накапливания, обработки и передачи информации с помощью ЭВМ (электронно-вычислительных машин, или компьютеров) и других технических средств.

*Информация* — сведения об окружающем мире, повышающие уровень осведомленности человека.

До тех пор пока информации было сравнительно немного, люди могли получать и обрабатывать ее без посредников. Увеличение объема информации привело к необходимости ускорения ее обработки. Для этого были разработаны механизмы, которые автоматизировали обработку информации. В настоящее время самым совершенным устройством переработки и хранения информации является компьютер.

Для машинной обработки информацию нужно записывать, обозначая буквы числами, т. е. кодировать ее. Поэтому необходимо знать способы записи числа.

## 1.2 Системы счисления

*Системой счисления* называют правила записи чисел с помощью некоторого набора знаков. В зависимости от способа использования этих знаков системы счисления делятся на *позиционные* и *непозиционные.*

*В непозиционных системах счисления* каждый знак обозначает всегда одно и то же число, и значения знаков в записи обычно суммируются. Поэтому для записи больших чисел приходится вводить все новые и новые знаки. Непозиционные системы неудобны для записи больших чисел и для выполнения арифметических действий.

Одна из непозиционных систем счисления используется до сих пор — это *римская система счисления.*

В римской системе счисления для небольших чисел используются такие знаки: I — один; V — пять; X — десять; L — пятьдесят; С — сто; D — пятьсот; М — тысяча.

*В позиционных системах счисления* один и тот же символ имеет разное количественное значение в зависимости от его позиции относительно других символов.

Поэтому в позиционных системах для записи любых чисел используется ограниченный набор знаков — цифр.

Наиболее распространенным способом записи чисел является *десятичная система счисления.* Каждое число записывается сочетанием десяти цифр, в котором вклад конкретной цифры зависит от ее позиции — разряда. Разряды отсчитываются справа налево. Первый разряд называется разрядом единиц, второй — десятков, третий — сотен и т. д.

Число в десятичной системе счисления можно представить с помощью операций сложения, умножения и возведения в степень. Например,

4321 = 4 • 103 + 3 • 102 + 2 • 101 + 1 • 10°.

Помимо десятичной системы счисления есть и другие позиционные системы: двоичная, троичная, четверичная, восьмеричная, шестнадцатеричная и т. д. Их названия соответствуют основаниям систем счисления.

*Основание системы счисления* — число цифр, допустимых в записи числа. Если число записано в позиционной системе счисления, отличной от десятичной, то основание указывается нижним индексом.

Например,

43218 = 4 • 83 + 3 • 82 + 2 • 81 + 1 • 8°.

Если основание системы счисления больше 10, то числа, которые больше 9, обозначают последовательно буквами латинского алфавита. Например,

AD2F16 = 10 • 163 + 13 • 162 + 2 • 161 + 15 • 16° = 4433510.

*В компьютерах используется двоичном, система счисления.*

Поскольку запись числа в двоичной системе получается достаточно длинной, в целях уменьшения ее длины часто используют восьмеричную или шестнадцатеричную системы счисления.

Для перевода числа из двоичной системы в десятичную достаточно записать его в виде суммы произведений и подсчитать результат. Например,

111001012 = 1 • 27 + 1 • 26 + 1 • 25 + 0 • 24 + 0 • 23 + 1 • 22 + 0 • 21 + 1 • 2° = 22910.

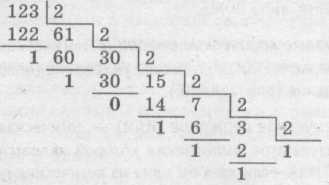
Аналогично осуществляется перевод из любой другой позиционной системы счисления в десятичную.

*Правило перевода чисел из позиционной системы с основанием* а10 *в десятичную систему:*



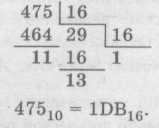
*Перевод чисел из десятичной системы в систему с произвольным основанием.* Метод перевода состоит в нахождении остатков от деления числа на степени основания той системы, в которую нужно перевести число. Последовательность этих остатков и есть запись числа в новой системе. Разряды отсчитываются справа налево. Делить надо до тех пор, пока не получен окончательный остаток.

Пример 1. *Перевести из десятичной в двоичную систему число* 123.



12310= 11110112.

Пример 2. *Перевести число* 475 *из десятичной системы в шестнадцатеричную.*



## 1.3 Логические операции

**В** основе всех действий с информацией лежат так называемые *логические операции.*

*Логические переменные* — переменные, которые могут принимать только два значения: ИСТИНА или ЛОЖЬ. Часто эти значения обозначают цифрами 1 и 0 (1 — ИСТИНА, 0 —ЛОЖЬ).

*Логическая операция —* действие, выполняемое над логическими переменными, его результат также либо ИСТИНА, либо ЛОЖЬ.

*Базовые логические операции:* логическое сложение (операция ИЛИ), логическое умножение (операция И) и отрицание (операция НЕ).

*Логическое сложение* (ИЛИ) — логическая операция, результатом выполнения которой является значение ИСТИНА, если хотя бы одна из логических переменных имеет значение ИСТИНА. Записывается с помощью знака «∨»: *A* ∨ *В.*

*Логическое умножение* (И) — логическая операция, результатом выполнения которой является ИСТИНА, если все логические переменные имеют значение ИСТИНА, во всех остальных случаях результат — ЛОЖЬ. Записывается с помощью знака «∧»: *А* ∧ *В.*

*Отрицание* (НЕ) — логическая операция, которая выполняется над одной логической переменной, ее результатом является значение ИСТИНА, если исходным значением было ЛОЖЬ, и ЛОЖЬ, если было ИСТИНА. Записывается двумя способами: ¬А или  *.*



Из логических переменных с помощью логических операций и скобок (для указания порядка действий) строятся *логические выражения.*

Например, *(А* ∧ *В)* ∨ ∨ (В ∧ С).



## 1.4 Единицы измерения информации

Поскольку в компьютерах используется запись информации в двоичной системе счисления, то количество информации измеряют, подсчитывая число двоичных разрядов (ячеек), необходимых для ее записи. Для удобства приняты следующие единицы измерения информации:

1 бит — одна ячейка, может хранить только значения 0 или 1;

1 байт = 8 бит;

1 килобайт = 1024 байта;

1 мегабайт = 1024 килобайта;

1 гигабайт = 1024 мегабайта.

*Обратите внимание на то, что единицы измерения информации основываются на степенях числа* 2. *Десятичные приставки (кило, мега и т. д.) дописываются только условно, так как* 210 = 1024 — *число, близкое к* 1000.

# 2 УСТРОЙСТВО СОВРЕМЕННЫХ ЭВМ

## 2.1 Схема фон Неймана

Принципиальная конструкция современных компьютеров опирается на схему фон Неймана. Эта схема определяет функции отдельных частей компьютера (рис. 1).



Рис. 1

Согласно схеме фон Неймана обработка информации выполняется процессором. Все действия, совершаемые процессором заданы программой — принцип программного управления. Данные и программы во время работы хранятся в оперативной памяти, для долгосрочного хранения те и другие переводятся из оперативной во внешнюю память. При этом человек вводит данные через устройства ввода (клавиатура, мышь, сканер, микрофон), а получает результат обработки через устройства вывода (монитор, принтер, акустические колонки).

Все программы и данные для работы процессора хранятся в памяти. Если их там нет, то компьютер работать не будет.

Объем устройств памяти определяется максимальным количеством информации, которое они могут хранить.

Оперативная память не может хранить данные при отсутствии электропитания. Для хранения данных без электропитания применяются разные виды внешней памяти. Самые распространенные сейчас устройства внешней памяти — дисковые, чаще всего это всевозможные магнитные диски.

Постоянная память содержит программы, с которых начинается работа ЭВМ. Без этих программ компьютер не сможет получить программы и данные из внешней памяти. Постоянная память не зависит от электропитания. Однако эта память медленнее, а объем ее невелик. Для изменения данных в ней требуется специальное устройство — программатор. В современных персональных компьютерах такое устройство есть, но используют его редко — только тогда, когда надо исправить ошибки в базовых программах.

Определяя назначение каждого устройства, схема не указывает способы изготовления, принципы работы устройств и методы связи между ними.

Принципы, которые определяют соединение устройств или их внутреннее устройство, называют архитектурой.

## 2.2 Основные устройства компьютера и их свойства

Почти все современные компьютеры сконструированы на основе шинной архитектуры. Согласно этой архитектуре различные устройства связываются между собой с помощью общего канала — шины. В некоторых случаях ее называют магистралью. К общему каналу подключаются различные устройства (процессор, оперативная память и т. д.). С шиной они часто соединены специальными устройствами-посредниками — контроллерами и адаптерами (рис. 2).



Рис.2

К одному контроллеру может присоединяться несколько устройств, работающих по известным ему стандартам. Контроллеры бывают самые разные — для подключения внешних устройств ввода-вывода, дисков и т. д.

Современные ЭВМ собираются из ультрабольших интегральных схем (УБИС). УБИС — это небольшие пластинки кремния, на которых вкраплениями других материалов созданы отдельные логические элементы. Такие элементы уложены очень плотно, что позволяет разместить на небольшой пластинке сложную схему.

Существует огромное количество микросхем для различных применений. Для решения конкретной задачи их часто выпускают целыми наборами — чипсетами (chipset).

Микросхемы собираются на платах — пластинах непроводящего материала, на которых закреплены проводники. К проводникам присоединяются микросхемы (припаиваются или вставляются в заранее припаянные разъемы).

Современные компьютеры конструируют, исходя из двух основных принципов: блочно-модульного и принципа открытой архитектуры.

Блочно-модульный принцип заключается в том, что отдельные по своим функциям устройства выполняются в виде отдельных модулей.

Принцип открытой архитектуры означает, что конструкторы ЭВМ предоставляют информацию о том, как разрабатывать устройства для нового компьютера.

Современные компьютеры собирают из отдельных частей, как конструктор, причем устройства стараются выполнять в виде отдельных модулей, а их контроллеры — в виде плат.

Некоторые основные устройства (сама шина, основные контроллеры и т. д.) собирают в одном модуле — материнской плате. Остальные устройства подключаются к материнской плате через специальные разъемы.

Если какое-то устройство выходит из строя, то весь компьютер, как правило, не ремонтируется, а заменяется только сломанный модуль. Если нужно подключить новое устройство, то можно разработать новый модуль-контроллер и встроить его в уже существующую ЭВМ.

Многие современные устройства и контроллеры — почти компьютеры. Они содержат свои процессоры, оперативную память, хранят и выполняют небольшие программы. Только это специализированные компьютеры, например, для вывода сложных изображений на экран.

# 3 ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

## 

## 3.1 Типы программного обеспечения

Для обработки информации компьютеру требуется последовательность команд преобразования информации — *программа.*

Программы необходимы для функционирования ЭВМ. Без них компьютер не работает и абсолютно бесполезен.

Каждый процессор имеет свой набор команд, т. е. множество операций, которые он «умеет» выполнять. Эти операции записываются в двоичном коде и для выполнения должны находиться в оперативной памяти.

Для первых ЭВМ все программы писались в двоичном коде. Причем для решения каждой задачи такая программа составлялась и вводилась в память отдельно. После выполнения ее стирали и вводили новую.

Сейчас, когда память достаточно дешева и доступна, применяют другой подход. Программы для компьютеров не стирают после выполнения, а накапливают.

Набор программ, который разработан для компьютера, называют его *программным обеспечением.*

Программное обеспечение (ПО) можно разделить на классы: системное, прикладное, а также класс сред для разработки программ.

*Системное программное обеспечение* — программы, которые обеспечивают работу других программ. Они позволяют хранить библиотеку программ, находить нужные и запускать их на выполнение, а также распределять ресурсы между ними во время работы.

*Прикладное программное обеспечение* — программы, которые предназначены для решения конкретных прикладных задач: редактирования текста или графики, выполнения каких-то расчетов и т. д.

*Среды для разработки программ* — специальные программы, позволяющие создавать новые системные и прикладные программы.

Самая важная часть системного программного обеспечения собрана в комплексе программ, который называется *операционной системой (ОС).*

***Операционная система выполняет следующие функции:***

* обеспечивает запуск программ;
* распределяет ресурсы компьютера между программами во время работы;
* предоставляет другим программам возможности работы с различными устройствами;
* предоставляет средства организации интерфейса пользователя.

***В состав современных операционных систем, как правило, входят несколько подсистем, основные из которых здесь перечислены:***

* подсистема управления процессами;
* файловая подсистема;
* драйверы — специальные программы, стандартизирующие работу с аппаратурой;
* функции для организации взаимодействия **программ** с пользователем;
* служба безопасности — разграничения прав доступа.

Самые распространенные в настоящее время операционные системы — системы Windows фирмы Microsoft.

Большинство компьютерных программ взаимодействуют с пользователем.

*Интерфейс* — способ взаимодействия пользователя с компьютером, т. е. правила, по которым отдаются команды и показываются результаты их выполнения.

В настоящее время чаще всего используется оконно-графический интерфейс, когда работа организуется с помощью окон, изображенных на экране.

**3.2 Файловая система**

Самая важная для пользователя часть операционной системы — работа с внешней памятью, т. е. с хранилищем программ и данных.

Самой крупной логической единицей внешней памяти в системе Windows является *том.* Как правило, его по традиции называют *диском.* На больших носителях может быть не один, а несколько томов — *логических дисков,* т. е. том не всегда является физическим устройством. В операционных системах Windows каждый том обозначается большой буквой латинского алфавита.

Внутри тома информация организована в *файловую систему.*

*Файловая система* — способ организации хранения информации на носителях внешней памяти. Обеспечением работы с ней в операционной системе занимаются специальные компоненты.

*Файл* — область внешней памяти, обозначенная именем. Правила именования и выделения областей зависят от конкретной операционной системы.

В системе Windows имя файла состоит из двух частей, разделенных точкой: само имя файла и его расширение. Например, в имени файла name.txt само имя — это name, а буквы txt обозначают, что файл является текстовым. Имена файлам рекомендуется давать, учитывая их содержимое, так чтобы по названию можно было понять, какую информацию содержит файл.

В операционных системах Windows в имени файла может быть до 250 символов. Это могут быть символы русского и латинского алфавитов, цифры и некоторые знаки препинания. В именах файлов нельзя использовать символы «/», «\», «\*», «,», «:», «?», «"», «<», «>», «⏐» они используются для записи команд.

Файлы различаются между собой не только именами, но и содержимым. В зависимости от типа содержимого файлам дают различные расширения. Некоторые распространенные расширения перечислены ниже.

|  |  |
| --- | --- |
| **Расширение** | **Содержимое** |
| ЕХЕ | Программа, выполняемая операционной системой |
| Расширение | Содержимое |
| СОМ | Программа в старом формате |
| SYS | Часть операционной системы |
| DLL | Библиотека функций для разных программ |
| ВАТ | Команды для операционной системы |
| DOC | Документ |
| ТХТ | Текстовый файл |
| BMP GIF JPG | Изображения |

Одно и тоже содержимое может быть записано в файлы с различным расширением, и файлы при этом будут отличаться *форматом файла.*

*Формат файла* — правила хранения информации в файле.

Вообще, многие расширения — это сокращенные названия форматов.

Жестких правил на названия расширений нет, и каждый разработчик может придумать свой формат файла и расширение для него. Тем не менее рекомендуется придерживаться общепринятых стандартов и не давать файлам своего собственного формата названия с известным расширением.

*Каталогом* называют специальный файл, в котором операционная система хранит информацию о других файлах (в частности, о других каталогах). На каждом устройстве есть так называемый корневой каталог — основной каталог диска.

Для большей понятности каталоги в операционной системе Windows называют *папками.*

Чтобы точно указать местонахождение файла используется *путь файла.*

*Путь к файлу* — указание точного местоположения файла. В нем слева направо последовательно указываются том (диск), на котором находится файл, и все папки, которые нужно раскрыть, чтобы добраться до файла. После тома ставится двоеточие, потом косая черта. Папки разделяют косой чертой. Например, C:\Windows\Mon документы.

*Полное имя файла* включает в себя путь к файлу и само имя файла.

Например, C:\Windows\Moи документы\ workl.doc — полное имя файла workl.doc, лежащего в папке «Мои документы» на диске С.

## 3.3 Основные операции с файлами. Буфер обмена

Для работы с файлами есть несколько стандартных операций, которые поддерживают все операционные системы: копирование, перемещение, удаление.

*Копирование* — переписывание информации в новое место, причем в старом месте информация сохраняется.

*Перемещение* — перенос информации в новое место, после чего в старом месте она автоматически стирается.

*Удаление* — стирание записи о файле из каталога.

В операционных системах Windows операции копирования, перемещения и удаления часто выполняются с помощью программы «Проводник». Для выполнения копирования и перемещения используется буфер обмена.

*Буфер обмена* — специальное средство, предоставляемое операционной системой программам для обмена информацией.

*Для работы с буфером обмена предусмотрены три команды:*

* *вырезать* — информация записывается в буфер обмена и удаляется со старого места;
* *копировать* — копия информации записывается в буфер обмена;

• *вставить* — информация из буфера обмена вставляется в новое место.

Таким образом, чтобы скопировать файл, нужно его найти, выделить, скопировать в буфер обмена, найти новое место и вставить его туда. Чтобы переместить его, нужно файл не копировать, а вырезать.

*Этим способом копируют и перемещают не только файлы, но и другую информацию.*

Для *удаления* файлов в программе «Проводник» есть специальная команда, которая так и называется. Найти ее можно в пункте меню «Файл» или на инструментальной панели в программе «Проводник».

Для *открытия* файлов на его значке нужно выполнить двойной щелчок левой клавишей мыши. Дальнейшие действия будут зависеть от типа файла.

Если этот файл — программа (расширение ЕХЕ или СОМ), операционная система попытается ее запустить. Если это документ, по расширению или содержанию которого операционная система может определить его тип, то она вызовет программу, которая и будет с этим файлом работать (например, запустит программу Microsoft Word).

Если же системе ничего о файле не известно, то она выдаст окно, в котором попросит выбрать программу для дальнейшей работы.

Специальное назначение имеют файлы-ярлыки (Ink — их расширение). В таком файле содержится ссылка на другой файл (каталог, программу, документ и т. д.). Двойной щелчок на нем откроет тот объект, на который ссылается ярлык.

Таким образом, операционные системы Windows упрощают работу пользователей, скрывая от них детали работы и показывая информацию в привычном для них виде.

# 4 ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

В настоящее время создано большое количество самых разных программ. Благодаря этому разнообразию программ можно, как правило, самому не писать программу, чтобы решить некоторую задачу, а воспользоваться уже готовым программным продуктом.

Для решения часто встречающихся задач были разработаны специальные информационные технологии, т. е. такие способы и приемы решения, которые позволяют решать большие классы задач.

Информационные технологии существовали и до компьютеров (например, каталогизация библиотек), но именно компьютеры позволили автоматизировать обработку данных и связывать эти технологии между собой в единое целое.

## 4.1 Обработка текста. Текстовые процессоры

Для подготовки всевозможных документов и других печатных материалов применяется класс программных средств, которые называются текстовыми редакторами и процессорами. Отличие текстового редактора от процессора заключается в том, что редактор позволяет только набрать текст, а текстовый процессор дает возможность изменить его начертание, вставить рисунок или таблицу и т. д.

Важным преимуществом этой технологии перед традиционными способами печати является возможность изменять текст без помарок на бумаге, сохранять его в виде файла и дорабатывать постепенно, а на печать отправлять только окончательно готовый текст. Текст легко тиражировать, поскольку если он хранится в электронном виде, то его можно распечатать в любом количестве экземпляров.

В текстовых редакторах и процессорах текст представляется в виде последовательности букв — символов. Некоторые символы имеют специальное назначение, например, служат концами строк, абзацев, страниц.

Подготовка документа при помощи ЭВМ проходит в три этапа: набор, редактирование, форматирование (верстка).

Современные текстовые процессоры обрабатывают текст абзацами. Место для ввода очередного символа показывает мерцающий прямоугольник или вертикальная черта — курсор.

При наборе текста рекомендуется соблюдать следующие правила:

* Текст должен набираться непрерывно, необходимо только указывать концы абзацев, а о распределении строк на листе текстовый процессор заботится сам.
* Слова в тексте отделяются друг от друга одним пробелом.
* Перед знаками препинания пробел не ставится. Тире выделяется пробелами с двух сторон.
* Парные знаки (кавычки, скобки и т. д.): открывающиеся примыкают к следующему слову, закрывающиеся — к предыдущему.

Текст может набираться в двух режимах — вставки и замены. В режиме вставки текст «раздвигается», и на место курсора вставляется новый символ. В режиме замены новый символ заменяет символ, стоящий справа от курсора. Удалить символы можно с помощью двух клавиш: нажатие клавиши «Deb удаляет один символ справа от курсора, a «Backspace» — слева.

Красную строку в абзацах, центрирование текста не следует выполнять, набирая пробелы, это может сильно усложнить работу процессора. Для выполнения подобного оформления текста существуют специальные команды форматирования.

После набора текст нужно отредактировать. При редактировании набранный текст прочитывается, в нем исправляются ошибки. Если нужно какую-то часть текста удалить, перенести в другое место документа или она повторяется несколько раз, то применяют операцию выделения фрагмента. В современных компьютерах фрагмент выделяют, как правило, с помощью мыши. Выделенный фрагмент можно скопировать, вырезать, вставить, удалить.

Форматирование является заключительным этапом при подготовке текста в текстовом процессоре. Форматирование — придание документу того вида, который должен быть на бумаге.

Текстовый процессор позволяет менять параметры шрифта, абзаца, страницы. Перечислим основные параметры оформления текста.

Параметры форматирования шрифта:

* стиль начертания букв — гарнитура;
* размер букв — кегль, он задается обычно в типографских пунктах (1 пункт — 1/72 дюйма).
* начертания шрифта — небольшие модификации внешнего вида: жирный, подчеркнутый.наклонный;
* цвета шрифта и фона, на котором написан текст.

Параметры форматирования абзацев:

* левый и правый отступы (от границы поля страницы) — места, откуда начинается и где кончается текст;
* красная строка — размер дополнительного отступа слева для первой строки абзаца;
* выравнивание абзаца — размещение текста в границах абзаца: ровный левый край (левое выравнивание), по центру, ровный правый край (правое), ровные оба края (по ширине);
* межстрочный интервал.

Параметры форматирования страницы:

* левое, правое, нижнее и верхнее поля — отступы от края страницы до границы текста;
* количество колонок текста на странице;
* колонтитулы — надписи сверху и снизу страниц;
* номера страниц.

При изменении параметров оформления действует правило: если ничего не выделено специально — изменения будут действовать либо на новый текст, либо на текущий абзац. Если часть текста выделена специально, то все изменения будут применены к этой части.

## 

## 4.2 Компьютерная графика

Основная задача информационных технологий при работе с графикой — позволить пользователям ЭВМ обрабатывать визуальную информацию.

Эту задачу выполняют программы рисования для художников, обработки фотографий, подготовки чертежей и видеофильмов и многие другие.

Основа всех технологий обработки изображений с помощью компьютера — представление графической информации в цифровом виде.

Для этого применяется способ растрового представления изображений. Его суть в том, что изображение представляется в виде растра — таблицы, в которой каждой точке ставится в соответствие цвет. Если таких точек много и размер их достаточно мал, то изображение получается качественным.

Количество точек на экране (по вертикали и горизонтали) называется разрешением экрана. Количество цветов, в которые можно окрасить точку, называют цветовым разрешением.

Эти два параметра — главные характеристики любого устройства, связанного с вводом или выводом изображений: мониторов, принтеров, сканеров, графических планшетов и т. д.

Для хранения и обработки информации в компьютере изображения представляют одним из двух способов: растровым и векторным. Изображения, представленные тем или иным способом, называются так же.

Растровые изображения. Изображение хранится как таблица (матрица) цветов точек — тем же способом, что отображается на экране.

Подобным образом кодируют фотографии, видеоизображения. Этот способ удобен для изображений, редактируемых целиком. Он универсален, однако не всегда удобен для исправления рисунка. Существенным недостатком является большая величина файлов, содержащих растровые изображения. Некоторые операции с растровыми изображениями приводят к ухудшению качества — потере цветов или точек.

Векторные изображения. Изображение составляются из готовых элементов, таких как прямые, окружности, сложные кривые и т. п. Каждый из них имеет набор параметров, которые определяют его положение на рабочем поле, цвет, толщину линий и т. д.

Векторным способом рисуются сложные шрифты, плакаты, трехмерные изображения, чертежи. Такие программы позволяют нарисовать с помощью математических операций очень многое, но далеко не всякое изображение можно представить в векторном виде. Фотографии, например, так представить нельзя.

Растровый и векторный способы дополняют друг друга и часто используются вместе. Как правило, программы, в основном предназначенные для работы с одним типом изображений, имеют средства и для обработки другого типа.

Чтобы отобразить или напечатать векторное изображение, его нужно преобразовать в матрицу точек. Эта операция называется растеризацией. Растеризация позволяет выводить векторные изображения с максимально возможным на этом устройстве качеством.

В сочетании с технологиями обработки текста, эти технологии позволяют с помощью компьютера готовить всевозможные печатные материалы — документы, журналы, книги и т. п. Так что этот комплекс получил название настольных издательских технологий.

## 

## 4.3 Электронные таблицы

Когда персональные компьютеры стали активно использоваться, выяснилось что для множества задач, связанных с подсчетами, нужно выполнять очень похожие операции, связанные с заполнением различных таблиц. Для решения таких задач были разработаны программы, которые называются табличными процессорами или электронными таблицами.

Табличный процессор предоставляет пользователю расчерченный по вертикали и горизонтали лист очень больших размеров — таблицу. Информация вносится в ячейки таблицы. Для записи ячеек каждая из них имеет две координаты: буквенное обозначение столбца строки и цифровое — строки.

Электронные таблицы позволяют вести экономические расчеты, анализировать данные экспериментов, иллюстрировать математические выкладки и т. д.

Поскольку таблицы предназначены для расчетов, то в ячейки можно записывать не только конкретные значения, но и формулы, по которым их можно вычислить. Формулы записывают по правилам, напоминающим правила записи программ. В формулах можно использовать математические операции, различные числа, строки и ссылки на значения других ячеек.

Если изменяются исходные значения (например, числа в ячейках, на которые ссылается формула), то содержимое ячейки пересчитывается автоматически.

В составе современных таблиц есть и средства деловой графики — на основе рассчитанных или введенных данных можно построить графики и диаграммы. Они также меняются при изменении данных.

Для оформления таблицы можно использовать разные шрифты, объединять несколько ячеек в одну, менять их высоту и ширину.

## 4.4 Реляционные базы данных

Для обработки больших объемов данных разработано много способов. Один из наиболее распространенных способов — технология реляционных баз данных.

Согласно технологии реляционных баз данных сами данные представляют в виде набора таблиц с фиксированным количеством столбцов, назначение и тип которых заданы при создании базы данных. Каждую колонку называют полем таблицы, а строку — записью. Записи в таблицах связываются между собой через так называемые ключевые поля.

Для работы с базами данных разрабатывают специальные комплексы программ — системы управления базами данных. Они позволяют вносить и получать информацию из базы с помощью запросов на специальном языке. Самый распространенный язык для этой цели — SQL (язык структурированных запросов, Structured Query Language).

Чтобы с базой данных смог работать обычный человек, разрабатываются приложения баз данных, которые обращаются к системе управления базами данных самостоятельно, а пользователю показывают результат в понятном для него виде.

# 5 АЛГОРИТМЫ И ПРОГРАММЫ

## 

## 5.1 Алгоритмы. Способы записи алгоритмов

*Алгоритм* — точное предписание исполнителю (человеку или автомату) выполнить последовательность действий, направленных на достижение поставленной цели.

Алгоритм всегда составляется для конкретного исполнителя, т. е. для человека или автомата, который может его исполнить.

Совокупность всех команд, которые исполнитель может выполнить, и всех состояний объектов, которые он в состоянии распознать, называется *системой команд исполнителя.*

*От обычных инструкций алгоритм отличается несколькими важными свойствами, допускающими его автоматическое исполнение.*

* *Дискретность.* Алгоритм состоит из отдельных команд, которые выполняются последовательно, начало и конец их выполнения строго фиксированы.
* *Понятность.* Команды алгоритма должны быть полностью понятны исполнителю.
* *Точность.* После выполнения каждой команды точно известно, завершено ли выполнение алгоритма или нужно перейти к следующей команде.
* *Результативность.* Алгоритм завершается либо достижением цели, либо обнаружением невозможности решения задачи.
* *Массовость.* Алгоритм единым образом применяется к любой корректной формулировке задачи, для решения которой он разработан.

Каждый алгоритм разрабатывается в строго определенных заранее условиях, включающих строгую формулировку задачи и систему команд исполнителя.

Для того чтобы алгоритм мог выполняться автоматом, его надо записать в той форме, в которой автомат его сможет «читать». Для ЭВМ такой формой является двоичный код.

Однако человеку записывать алгоритм в машинном коде крайне неудобно, так как на запись даже простого алгоритма уходит много времени, и появляется масса трудно обнаруживаемых ошибок. Поэтому для составления и подготовки программ используются другие формы записи алгоритма, в частности блок-схемы и языки программирования.

*Блок-схема* — последовательность, составленная из отдельных соединенных между собой в порядке выполнения блоков. С помощью блок-схем описывают структуру программы. Вид блоков определяется их назначением в программе. Форма блока определяет вид действий, а записи внутри — подробности (параметры).

Конкретные виды блоков будут упоминаться вместе с соответствующими частями программ.

*Язык программирования* — формализованный язык, предназначенный для описания алгоритмов решения задач на ЭВМ.

Языки программирования бывают: низкого, среднего и высокого уровня.

*Язык программирования низкого уровня* — язык программирования, структура команд которого определяется системой команд процессора и архитектурой ЭВМ. Часто эти языки называют *языками ассемблера.*

Писать программы на языках ассемблера немногим проще, чем в двоичном коде, поскольку по сути эти языки являются просто буквенными записями машинных команд.

На основе языков ассемблера были созданы *языки среднего уровня* или *языки макро-ассемблера.* В них система команд расширена более сложными конструкциями. Специальные программы-переводчики сами переводят эти конструкции в двоичный код.

## 5.2 Языки высокого уровня

*Языки высокого уровня* — языки программирования, средства которых допускают описание алгоритма в наглядном виде, т. е. не на основе команд процессора, а на основе слов естественного языка.

Программа на таком языке переводится на машинный с помощью *программы-транслятора,* которая переводит конструкции языка программирования на язык команд процессора. Языки высокого уровня не зависят от конкретного компьютера, а зависят от программы-транслятора.

При разработке новых процессоров для них вначале первым делом разрабатывают программы-ассемблеры, а потом переводят на язык ас-

*Программа на языке Basic* представляет собой последовательность команд-операторов. Программа начинает выполнение с первого от начала оператора и заканчивает работу либо на последнем операторе, либо на операторах остановки: END или STOP.

*Имена переменных* в языке Basic могут включать в себя от 1 до 40 символов и должны начинаться с буквы. Имена переменным рекомендуется давать так, чтобы из имени был понятен смысл переменной.

*Тип переменной* в языке Basic определяется с помощью последнего символа имени (суффикса). Используются следующие типы переменных:

* ! — дробное число с одинарной точностью;
* # — дробное число с двойной точностью;
* % — целое число;
* & — длинное целое число;
* $ — строка.

|  |  |
| --- | --- |
| **Имя** | **Тип** |
| Norma | Число с плавающей точкой, одинарная точность |
| Bint% | Целое число |
| S$ | Строка |
| А# | Число с плавающей точкой, двойная точность |

Правила записи выражений в языках программирования очень похожи на математические. Но при написании программы необходимо записывать выражения одной строкой — линеаризовать запись.

В зависимости от типа вычисляемого в выражении итогового результата, говорят о *типе выражения.*

*Числовые выражения* — выражения, результатом вычисления которых является число.

|  |  |  |
| --- | --- | --- |
| **Математические операции, доступные в языке Basic** | | |
| Знак операции | Операция | Пример |
| + | Сложение | А+В |
| - | Вычитание | А-В |
| \* | Умножение | А\*В |
| / | Деление | А/В |
| \ | Деление нацело | А\В |
| mod | Нахождение остатка от деления | A mod В |
| ^ | Возведение в степень | А^В |

Для обозначения порядка действий используют скобки.

В выражениях можно использовать встроенные математические функции:

* sin(x) — синус числа *х;*
* cos(x) — косинус числа *х;*
* tan(x) — тангенс числа *х;*
* atn (x) — арктангенс числа *х;*
* abs (x) — модуль числа *х;*
* sqr (x) — корень квадратный из числа *х.*

В тригонометрических функциях аргумент выражается в радианах.

Числовые выражения могут быть целыми и дробными. Трансляторы языка Basic учитывают это автоматически — при присваивании целочисленного выражения дробной переменной дробная часть считается равной 0, а при присваивании дробного выражения целой переменной отбрасываются знаки после запятой.

|  |  |
| --- | --- |
| **Примеры записи числовых выражений** | |
| **Математическая запись** | **Запись на языке Basic** |
|  |  |
|  |  |
|  |  |

*Строковые выражения* составляются из переменных и функций строкового типа, его результат — строка.

Для составления строковых выражений можно использовать:

* строки-константы, записанные в двойных кавычках;
* функции, возвращающие строки;
* операцию «склеивания» (конкатенацию) двух строк (записывается как сложение).

|  |  |
| --- | --- |
| **Строковые функции** | |
| Len(A$) | Длина строки А$. Внимание: длина строки А$ — число! |
| Left$(A$,N) | Первые N символов строки А$ |
| Right$(A$,N) | Последние N символов строки А$ |
| Mid$(A$,P,N) | N символов строки А$, начиная с символа номер Р |
| Str$(N) | Строка, содержащая запись числа N |
| Val(S$) | Число, записанное в строке S$ |

|  |  |
| --- | --- |
| **Примеры строковых выражений** | |
| "Привет!" | «Привет!» |
| 1еП$("Привет",3) | «При» |
| MID$("12345",3,2) | «34» |
| "Вася"+сhг$(32)+"Синицын" | «Вася Синицын» |

*Логические выражения* в языке Basic строятся из элементарных условий с помощью обычных логических операций, результат — значения ИСТИНА или ЛОЖЬ.

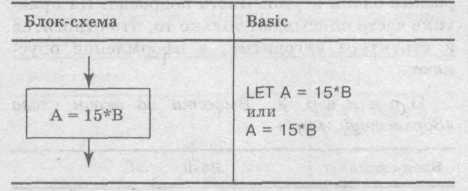
*Элементарными* называют условия, сравнивающие выражения между собой записывается) некоторое значение. Значение может быть предварительно вычислено.

|  |  |
| --- | --- |
| **Примеры условий** | |
| А>=0 | А — неотрицательно |
| (A>=10)AND(A<15) | А∈ [10,15] |
| (X<>0)AND(X<0.5) | X ≠ 0, X меньше 1/2 |
| (X<2)AND(X>5) | Противоречивое условие. Всегда имеет значение «ложь». |

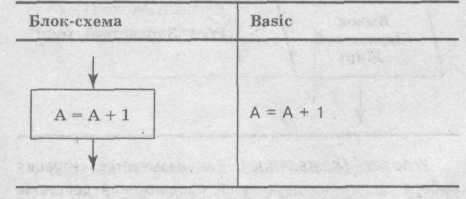
## 5.3 Основные операторы и синтаксические конструкции

*Присваивание.* В результате выполнения этого оператора переменной присваивается (иногда говорят — записывается) некоторое значение. Значение может быть предварительно вычислено.

Пример 1. *Записать в переменную А значение* 15, *умноженное на содержимое переменной В.*



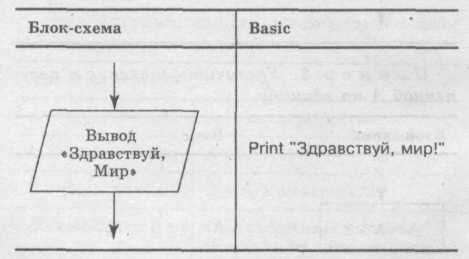
Пример 2. *Увеличить значение в переменной А на единицу.*



*Ввод и вывод.* Эти команды позволяют организовывать взаимодействие с пользователем в диалоговом режиме. Стандартные операторы предусматривают ввод пользователем данных с клавиатуры и вывод различных текстов на экран.

При вводе и выводе текст записывается в кавычках, переменные — именами без кавычек. По правилам вывод и ввод нужно разделять на разные блоки и записывать подробно. На практике часто описывают только то, что относится к структуре алгоритма, а оформление опускают.

Пример 3. *Вывести на экран слова «Здравствуй, мир!».*

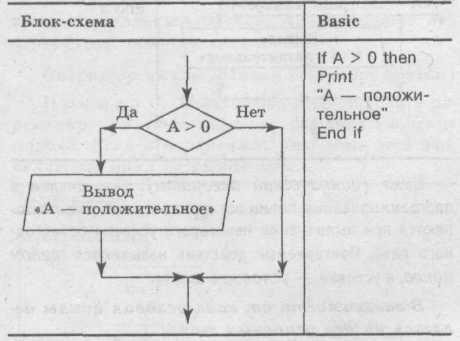


*Условие (ветвление). Так* называется ситуация выбора одного из двух путей продолжения действий, как правило, в случае выполнения некоторого условия.

В программах оператор условия имеет две формы — полную и краткую. В краткой форме выполняются только действия при выполнении условия, а в полной — и при невыполнении. Условие записывается логическим выражением.

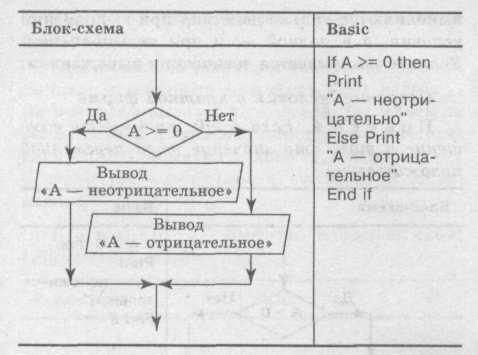
***Оператор условия в краткой форме***

Пример **4.** *Если А >* О, *то вывести сообщение о том, что значение этой переменной положительно.*



***Оператор условия в полной форме***

Пример 5. *Если А > 0, то вывести сообщение о том, что это значение неотрицательное; иначе* — *сообщение о том, что оно положительное.*



*Цикл (циклический алгоритм).* Под циклом в программировании понимают действия, которые повторяются при выполнении некоторого условия более одного раза. Повторяемые действия называются *телом цикла,* а условие —*условием цикла.*

*В зависимости от вида условия циклы делятся на два основных типа:*

* *Цикл «Пока»* (с предусловием). Цикл выполняется, пока условие истинно. Как правило, условие проверяется перед выполнением тела цикла.
* *Цикл «До»* (с постусловием). Цикл выполняется, пока условие ложно. Как правило, условие проверяется после тела цикла.

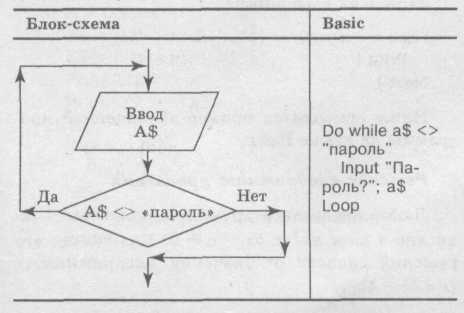
Важный частный случай цикла — определенный цикл.

*Определенным циклом* считается цикл, в котором условие наложено на количество повторений цикла, т. е. определенный цикл повторяется заданное количество раз. Переменная, которая отслеживает количество повторений называется *счетчиком цикла.*

В блок-схемах для циклов обычно не используют специальных блоков, а записывают эту структуру с помощью ветвления.

*Оператор цикла «Пока» (с предусловием)*

Пример 6. *Повторять ввод строки в переменную а$, пока там не появится значение пароля. Если в переменной уже есть это значение, то цикл не выполнится.*



Если дискриминант положителен *(D >* 0), то уравнение имеет два действительных корня:

*Оператор цикла «До» (с постусловием)*

Пример 7. *Повторять ввод строки в переменную а$, до появления в ней значения пароль. Цикл выполнится хотя бы один раз.*

Запись на языке Basic.

Do

Input "Пароль?";а$

Loop until a$ = "пароль".

*Оператор цикла с параметром (определенный цикл)*

П р и м е р 8. *Вывести на экран числа от 1 до 10.*

Запись на языке Basic.

For I = 1 to 10

Print I

Next I.

Ниже приводится пример законченной программы на языке Basic.

***Решение квадратного уравнения***

Любое квадратное уравнение может быть записано в виде *ах2* + b*х* + с = 0. Количество его решений зависит от значения дискриминанта *D* = b*2 -* 4ас.

Если дискриминант положителен (D>0), то уравнение имеет два действительных корня:

Х 1,2=



Если дискриминант равен нулю *(D* = 0), то решение одно: *х* =



Если дискриминант отрицателен *(D <* 0), то действительных корней это уравнение не имеет.

Rem Программа решения квадратных уравнений

Input "Коэффициент А"; А

Input "Коэффициент В"; В

Input "Коэффициент С"; С

D = В\*В - 4\*А\*С

If D > 0 then

Х1 = (-b-sqr(d))/(2\*a)

Х2 = (-b+sqr(d))/(2\*a)

Print "Х1 = "; Х1

Print "Х2 = "; Х2

Else

If D = 0 then

Print "X = "; -b/(2\*a)

Else

Print "Нет действительных корней."

End if

End if

End

# 6 Компьютерные телекоммуникационные сети

## 

## 6.1 Основные принципы организации современных компьютерных сетей

*Компьютерные сети* — комплексы аппаратуры и программного обеспечения, решающие задачу передачи данных от одного компьютера к другому.

*Узел сети* — один компьютер, соединенный с сетью.

*Канал связи* — устройства, обеспечивающие передачу сигнала между точками сети.

*Коммутация* — соединение точек информационным потоком, передача информации.

*Протокол* — правила передачи и приема информации.

*Локальная сеть —* сеть в рамках одного здания, меньше 20 км.

*Территориальная сеть* — сеть в рамках одной территории.

*Глобальная сеть* — сеть, охватывающая несколько крупных территорий.

Большинство сегодняшних компьютерных сетей основано на принципе *коммутации пакетов.* Согласно этому принципу каждое сообщение при передаче по сети делится на части — *пакеты.* Каждый пакет передается по сети отдельно как единое целое, а все сообщение собирается только в конечной точке.

*Принцип коммутации пакетов позволяет:*

* сократить до минимума количество линий связи;
* сократить потери на повторную пересылку в случае ошибок;
* легко расширять сеть.

Для связи одного узла с остальными достаточно связать его всего одной линией с другим узлом и передавать пакеты ему, а тот их будет передавать дальше. Такой подход дает возможность легко расширять сеть, используя столько каналов, сколько имеется в наличии. Если возникают помехи или обрывы связи, то нужно повторно передавать только утерянные и поврежденные пакеты, а не все сообщение.

Важное свойство сети — возможность связи с другими сетями, с помощью которых можно передавать данные на большие расстояния.

*Сети объединяют, руководствуясь следующими принципами:*

* при соединении сети не должны подвергаться внутренним переделкам;
* если пакет не дошел до адресата, то должна быть возможность это узнать и передать его дальше;
* передача информации между раздельными сетями происходит с помощью специальных устройств, называемых *шлюзами* и *маршрутизаторами,* эти устройства должны быть максимально просты и надежны.
* объединенная сеть не должна иметь единого центра управления.

Перечисленные принципы были использованы при проектировании сетей, которые позже стали основой сети *Internet.*

Основа пересылки информации в глобальной сети — *набор протоколов TCP/IP.* Для его создания задачи, решаемые сетью, разделили на уровни. Эти уровни определяют, какие задачи будет решать каждый протокол. Выполнив свою часть работы, протокол передает данные на уровень ниже. Верхним уровнем считается уровень пользователя, а нижним — уровень взаимодействия с аппаратурой передачи сигнала.

Самый последний уровень, *аппаратный,* не имеет протоколов из набора TCP/IP. Программы и аппаратура этого уровня разрабатываются производителями отдельно, и должны только соответствовать стандарту для работы протоколов в составе операционной системы.

Такая схема позволяет передавать данные по любому каналу, для которого разработан нижний уровень, и сеть может работать, используя любые каналы связи — спутниковые, специальные кабельные, радиосвязь, телефон...

На *сетевом уровне* действует один из основных протоколов стека TCP/IP, IP как раз и занимается определением конкретного маршрута передачи пакетов. Если есть несколько возможных путей передачи, то программное обеспечение автоматически выбирает наиболее выгодный из них с точки зрения длины, скорости и ' стоимости передачи.

Для ориентации в сети каждая машина имеет уникальный адрес. В соответствии с текущей версией протокола, это четырехбайтное число. Адреса записывают четырьмя однобайтовыми числами через точку. Например, 195.210.190.11.

Несмотря на то что таких адресов может быть очень много, уже сейчас ощущается их нехватка. Для того чтобы решить эту проблему, используют методы, позволяющие работать крупным сетям через компьютеры-посредники.

На более высоком *транспортном уровне* основным является протокол TCP. Он обеспечивает разделение сообщения на пакеты и контролирует правильность и порядок их отправления и доставки. Этот протокол обеспечивает надежную пересылку информации.

В дополнение к нему используется протокол UDP, который выполняет те же функции, но не контролирует доставку и не проверяет порядок пришедших пакетов. Это менее надежный, но зато более быстрый способ связи. Он применяется, например, для передачи звука или видеоизображения.

На *прикладном уровне,* самом высоком, действуют прикладные протоколы — правила обмена информацией между пользовательскими программами. Именно с ними и работают обычные пользователи. На этом уровне существует масса разных протоколов и при необходимости разрабатываются новые.

Обмен информацией между программами в сети Internet строится по так называемой *архитектуре «клиент-сервер».* Суть такого построения заключается в следующем. Все задачи в программе разделяются на две части. Первую — непосредственное общение с пользователем, получение от него команд и показ результатов — выполняет *клиентская часть.* Непосредственную обработку информации и выполнение команд обеспечивает *серверная часть.* При этом клиент не имеет данных и почти не занимается их обработкой, он только отсылает к серверу запросы и показывает ответы на них, а сервер только отвечает клиенту на его запросы.

В сети Internet клиент и сервер обычно находятся на разных машинах. Для каждого вида работы существуют свои клиенты и серверы. Общаются они по прикладному протоколу, который и определяет правила написания запросов и ответов. Такая система называется *сетевой службой (сервисом сети).*

Под словом *сервер* подразумевают одно из двух: либо *программу,* которая обслуживает запросы к информации определенного вида, либо *узел сети,* на котором такие программы работают.

Далее перечислены основные службы сети.

## 

## 6.2 Служба Domain Name System (DNS)

Адреса в сети Internet удобны для компьютеров, но неудобны для человека. Тем более что они никак не отражают ни местоположение, ни назначение узла. Для более удобной адресации была создана служба DNS — Domain Name System — *система именования доменов.*

В системе DNS все узлы группируются в большие логические объединения — *домены.* Такие домены выделяют по разным признакам — по стране, по назначению узла, по названию организации и т. д.

Самые крупные домены — домены первого уровня. Это сокращения от названий стран (например, ru — Россия, uk — Великобритания) или крупных классов организаций, например, коммерческих (com), учебных (edu), правительственных (gov). Внутри доменов первого уровня выделяют домены второго уровня, как правило, по названиям организаций, людей или проектов. В принципе система позволяет и дальше детализировать адрес, но часто эта возможность не нужна. Внутри доменов последнего уровня узлы сети получают персональные имена.

Адрес узла записывается от имени узла и дальше — к доменам первого уровня. В результате получаются записи вроде www.microsoft.com. В адресе DNS части разделяются точкой и записываются слева направо: узел www, из домена microsoft, который входит в домен com.

Таким способом система DNS позволяет запоминать и записывать не числовые адреса, а понятные названия.

Для работы этой системы на некоторых узлах работают специальные программы, преобразующие адреса DNS в числовые. Каждый компьютер (клиент) «знает», каким серверам направить запрос. Сервер либо сообщает числовой адрес, либо перенаправляет запрос другому серверу. Таким образом DNS функционирует как *распределенная база данных.*

## 

## 6.3 Почтовая служба (E-mail)

Электронная почта появилась одной из самых первых. До сих пор это одно из наиболее используемых, мощных и удобных средств сети. Эта служба позволяет обмениваться почтовыми сообщениями.

Основой для обмена такими сообщениями служат специальные почтовые сервера. На этих серверах каждый *абонент* имеет *почтовый ящик* — место, в котором хранится его почта.

Программа-клиент по команде пользователя обращается к этому почтовому ящику для выполнения операции с его письмами — просмотр, удаление, перемещение, отправку и пересылку.

Каждый ящик имеет хотя бы один *почтовый адрес,* состоящий из имени ящика (чаще всего это имя пользователя) и DNS-адреса сервера. Имя ящика и адрес разделяются символом @.

Пример такого почтового ящика: Flamer@hotmail.com

В принципе система позволяет одному ящику получать письма на несколько адресов, а одному адресу — на несколько почтовых ящиков.

## 

## 6.4 Служба File Transfer Protocol (FTP)

Относительно старая служба, предназначенная для обмена файлами. Она позволяет представить часть файловой системы (одну или несколько «ветвей») в общее пользование.

С помощью системы FTP поддерживается деятельность крупных файловых архивов. Еще эту систему используют для размещения своих файлов на других узлах.

## 

## 6.5 Служба World Wide Web (WWW)

В настоящее время эта служба сети — самая используемая. Именно с ней связывают широкое распространение сети Internet.

В этой службе информация представляется в виде *страниц,* содержащих текст, графические изображения и элементы управления. Страницы связываются между собой с помощью *гиперссылок* — специальных участков информации, через которые можно перейти на указанную страницу.

Содержание страниц пишется на специальном языке — HTML (HyperText Markup Language) — языке разметки гипертекста. Иллюстрации хранятся рядом с ними в графических файлах. Написанную на таком языке страницу можно просмотреть с помощью специальных программ просмотра — *браузеров.*

Страницы размещаются на узлах и предоставляются для просмотра с помощью WWW-cepверов. Написанные и не изменяемые на сервере страницы называют *статическими.* По некоторым правилам можно разрабатывать программы, которые будут формировать страницу на основании данных запроса, например, выбирать товары из общего списка по указанному критерию. Такие страницы называют *динамическими.*

С помощью такого способа представления можно представить большую часть всей информации, которая размещается в сети Internet. В виде статических страниц — информацию учебного, информационного, справочного характера. В виде динамических страниц — почтовые ящики, доступ к базам данных и т. д.

## 

## 6.6 Ресурсы в сети

Поскольку на одном узле функционирует, как правило, несколько служб, а в каждой службе подразумевается разбиение информации на части, то при работе в сети требуется способ ориентироваться не только среди узлов, но и внутри каждого узла и даже каждой службы. Такое средство нужно для отправки запросов и получения ответов, в частности — для записи адресов ссылок.

*В адрес ссылки должны входить:*

* Адрес узла, на котором должен находиться ресурс. Это может быть цифровой адрес или адрес DNS.
* Служба, к которой должен быть направлен запрос (или вид действий, с ней связанный, отправка почты по адресу, например).
* Положение ресурса в структуре службы, если она есть.
* Название ресурса.

Универсальным средством адресации в сети является URL (Uniform Resource Locator) — унифицированный указатель ресурса.

*Правила записи URL*

1. Сначала записывает действие (название службы), после которого ставится двоеточие.
2. Если потом записаны две косые черты, то предполагается адрес, записанный как место в файловой системе. Сначала пишется адрес машины, потом путь в каталогах, а потом имя файла ресурса.
3. Если двух косых черт нет, то дальше идет весь адрес сразу.
4. Дополнительные параметры — команды обрабатывающей этот адрес клиентской программе записываются после знака вопроса. Имя параметра отделяется от значения знаком =. Если параметров несколько, то они разделяются символом &.

*Ниже приведены несколько примеров.*

* http://www.microsoft.com/w2k/tech/faq.html —ресурс службы www (протокол HTTP — ее протокол), на узле www.microsoft.com, в каталогеw2k, в подкаталоге tech, файл faq.html.
* ftp://www.microsoft.com/pub/w2k/sp/w2ksp2en.exe — ресурс службы FTP, на узлеwww.microsoft.com, внутри каталога Pub, в каталоге w2k, в подкаталоге sp, файл w2ksp2.exe.
* mailto:web@hotmail.com?subject=Test URLlink — начать писать письмо (mailto) по адресуweb.hotmail.com, указав в качестве темы сообщения «Test URL link».

При вызове URL на машине пользователя начинает действовать программа, которая зарегистрирована для данного типа действий. Этой программе и передаются адрес узла и все остальные параметры.