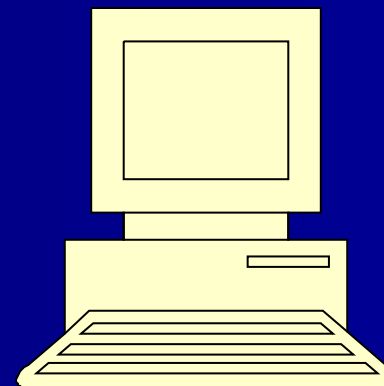




# ОПЕРАЦИОННЫЕ СИСТЕМЫ, СРЕДЫ И ОБОЛОЧКИ.



# ОБЗОР ОПЕРАЦИОННЫХ СИСТЕМ

- *Операционная система (ОС)* - это совокупность программных средств, обеспечивающих управление аппаратными ресурсами компьютера, поддержку выполнения программ, взаимодействие программ с аппаратной частью, другими программами и пользователем.
- ОС является базовым ПО («программным обеспечением»), без которого ЭВМ не может работать. Поэтому любой тип ПК (персональных компьютеров) комплектуется ОС. Обычно имеется несколько разновидностей ОС, которые могут быть установлены на ПК. Основная часть ОС - ядро загружается в оперативную память при включении компьютера и находится там постоянно в течение всего периода работы ЭВМ (т. е. резидентно).

- Прикладные программы могут работать только в среде какой-либо операционной системы. Для каждой разновидности ОС разрабатывается свой набор прикладных программ (приложений).
- Ситуация, когда программа, разработанная для одной операционной системы может выполняться в среде другой ОС непосредственно, встречается нечасто. Чаще программные продукты, ориентированные на какую-то конкретную ОС не могут функционировать в среде другой ОС (программная несовместимость).
- Основное назначение операционной системы - это связь между программными продуктами и непосредственно "железом" компьютера. Операционная система делает программы в определенной степени независимыми от конкретной модификации машины и установленного на ней оборудования. Она также позволяет "сказать" пользователю, что он хочет от компьютера.

# *ОСНОВНЫЕ ЗАДАЧИ ОС:*

1. поддержка работы программ; обеспечение их взаимодействия с аппаратной частью и друг с другом;
2. распределение ресурсов (процессорного времени, оперативной памяти, дискового пространства и т.п.); организация файловой системы (системы хранения данных на внешних носителях информации); учет использования ресурсов, управление видеосистемой;
3. обработка ошибочных ситуаций; защита информации;
4. поддержка возможности для пользователя управлять машиной с помощью специальных команд (обработка командного языка в процедурной среде) или воздействием на определённые объекты (кнопки и др. в объектно-ориентированной среде);
5. поддержка сети.

# *ИНТЕРФЕЙС ПОЛЬЗОВАТЕЛЯ*

Кроме управления ресурсами и поддержки работы программ ОС представляет пользователю возможность управлять компьютером в режиме диалога. Это происходит при помощи интерфейса пользователя.

Интерфейс пользователя(ИП) - составляющая программного продукта, обеспечивающая диалоговое взаимодействие между программой и пользователем.

Простейшая разновидность ИП - *интерфейс командной строки*. Он предполагает управление компьютером посредством ввода команд с клавиатуры.

Ярким примером служит командная строка в MS-DOS:

```
C:\USERS\DIPLOM\> copy head.htm C:\USERS\BAKALAVR
```

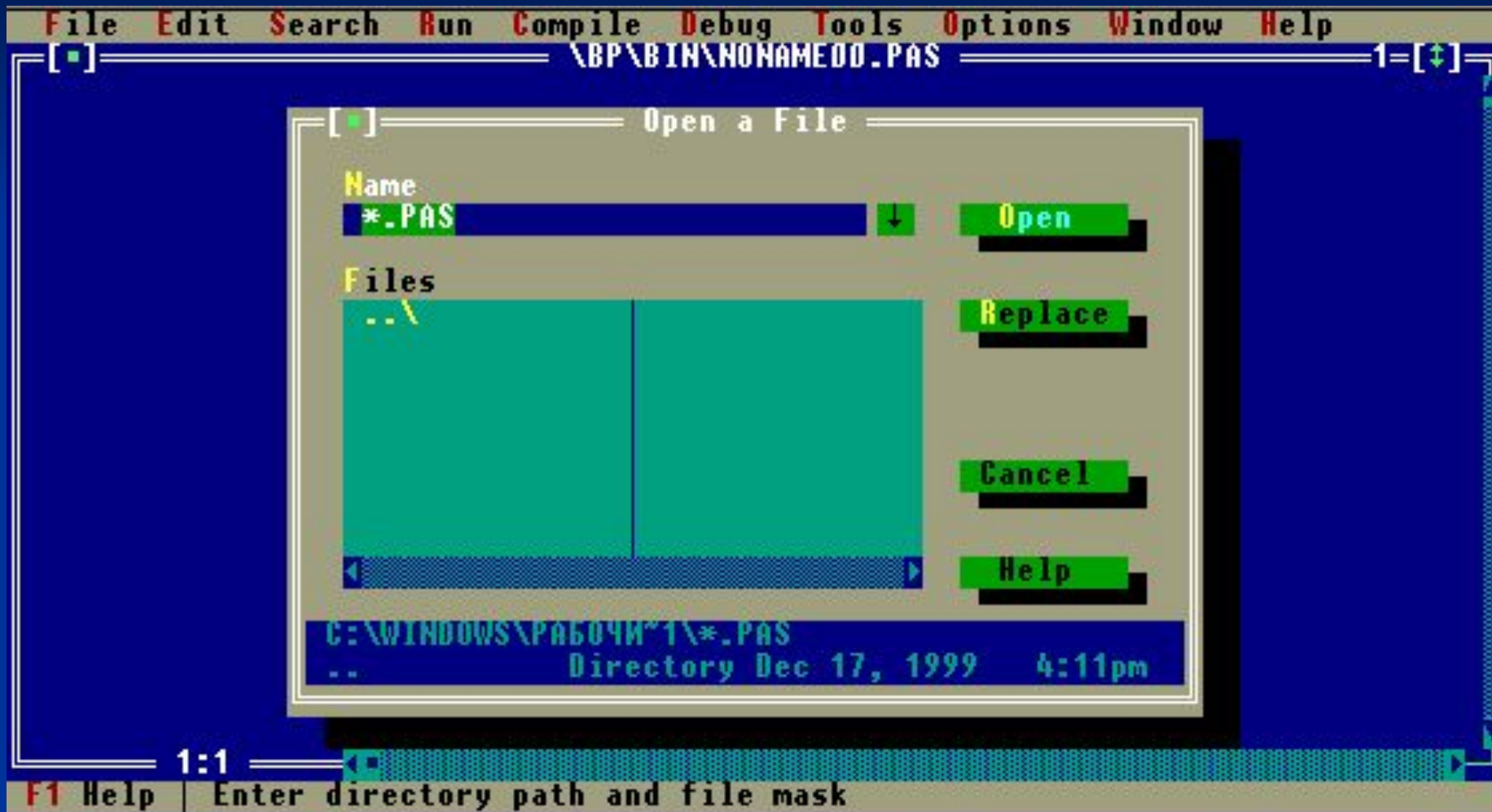
```
1 файл скопирован
```

Более удобный вид ИП - *текстовый оконный интерфейс*.

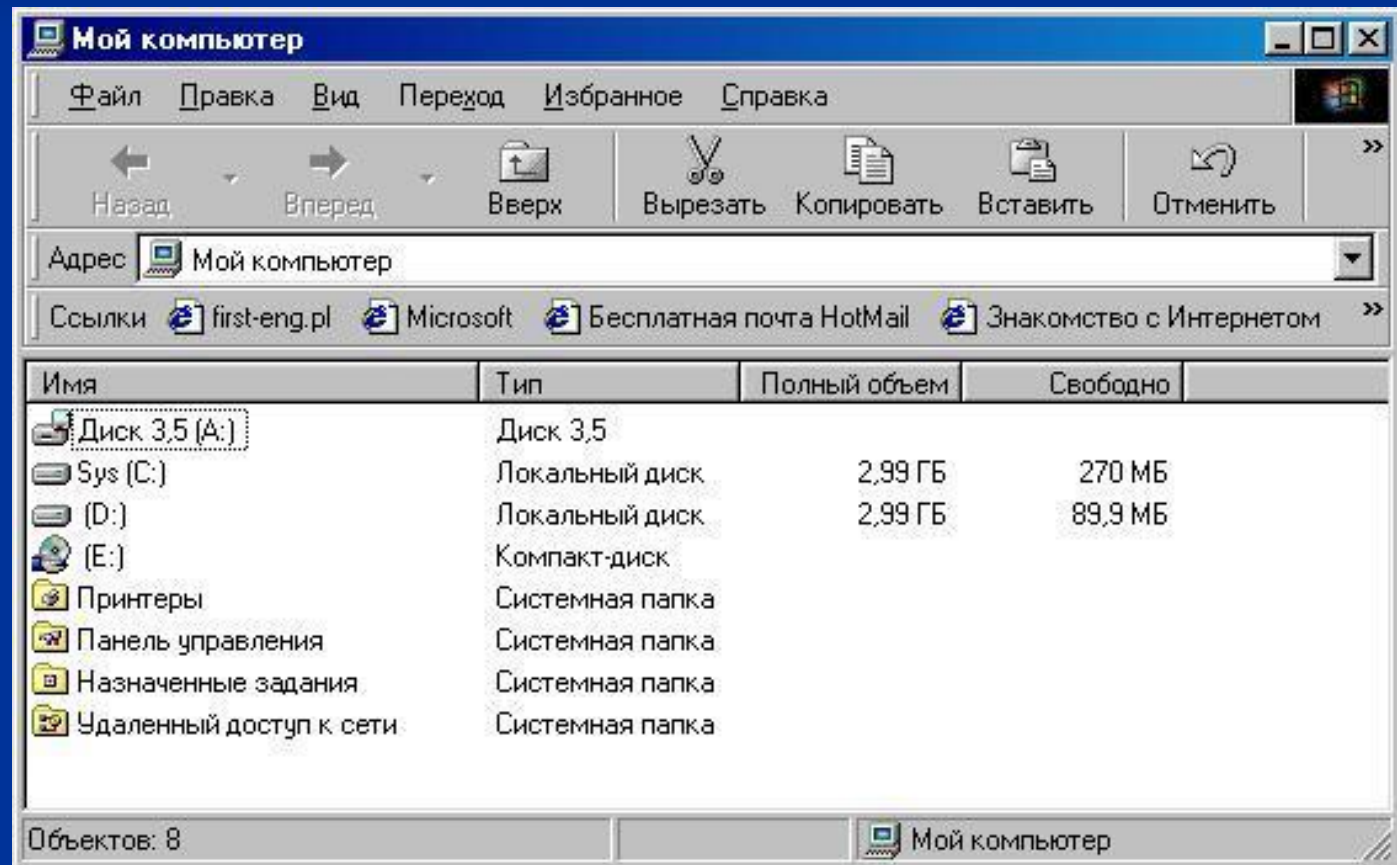
Он не требует набора команд на клавиатуре, а сводит управление к нажатию отдельных клавиш или кнопок мыши при выборе управляющих действий в меню и диалоговых окнах. Оконный интерфейс позволяет наглядно увидеть производимые действия и их результат.

Примером может служить инструментальная оболочка

Borland Pascal:



Наиболее современным является **графический оконный интерфейс**, соединяющий в себе развитые диалоговые средства оконного интерфейса (системы меню, диалоговые окна, панели инструментов, пиктограммы и др.) с большими изобразительными возможностями графического режима. Примером может служить окно папки «**Мой компьютер**» операционной системы **Windows** :





# ХАРАКТЕРИСТИКИ ОС

1. **разрядность** ( для ПК - 8-разрядные, 16-разрядные, 32-разрядные, 64-разрядные ОС);
2. **число программ, одновременно выполняемых под управлением ОС** (одно - и многозадачные ОС).

**Многозадачные ОС** поддерживают параллельное выполнение нескольких программ, работающих в рамках одной вычислительной системы, в один момент времени.

**Многозадачность** бывает **корпоративная** и **вытесняющая**.

При наличии **корпоративной многозадачности** приложения совместно используют процессор, периодически передавая его друг другу. Если какое-то приложение откажется освободить процессор, система ничего не сможет с этим поделать.

Если используется *вытесняющая многозадачность*, то операционная система полностью контролирует все приложения и распределяет между ними процессорное время, тем самым сильно понижая вероятность "зависания" системы при ошибках в работе программ.

*Однозадачные ОС* поддерживают режим выполнения только одной программы в отдельный момент времени;

3. ***многопоточность*** – это технология, позволяющая приложением должным образом осуществлять многозадачное выполнение своих процессов. *Процесс* – любая задача или деятельность, инициируемая программой. Одна программа может выполнять несколько процессов одновременно;
4. ***тип пользовательского интерфейса***: интерфейс командной строки, текстовый оконный интерфейс, графический оконный интерфейс пользователя (ИКС, ТИП – «текстовый интерфейс пользователя», ГИП);
5. ***требование к аппаратным ресурсам***;
6. ***производительность***;
7. ***надёжность*** (устойчивость в работе, защищённость данных от несанкционированного доступа );
8. ***обеспеченность прикладными программами***;
9. ***наличие сетевых возможностей*** ( сетевые, локальные ОС);

**Сетевые ОС** предназначены для управления ресурсами компьютеров, объединенных в сеть с целью совместного использования данных, и предоставляют мощные средства разграничения доступа к данным при обеспечении их целостности и сохранности, а также множество сервисных возможностей по использованию сетевых ресурсов;

**10.** *количество поддерживаемых процессоров:*  
*однопроцессорные, многопроцессорные;*  
**Многопроцессорные ОС**, в отличие от **однопроцессорных**, поддерживают использование нескольких процессоров для решения одной задачи;

**11.** **открытость** операционной системы, заключается в том, что компоненты ОС доступны в исходных кодах (т. е. допускают программное видоизменение программистом) для любого пользователя.

## **12. способ использования оперативной памяти;**

Различают два способа работы с памятью: **линейный адресный** - ОС работает со всей системной памятью, как с единым непрерывным пространством; **сегментарный** - ОС работает с небольшим объёмом доступной без специальных средств оперативной памяти.

# НАИБОЛЕЕ РАСПРОСТРАНЕННЫЕ ОС ДЛЯ ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРОВ

- **MSDOS**
- Семейство **WINDOWS**
- **OS/2**
- Семейство **UNIX**
- **Linux**

# ОС семейства DOS

Первый представитель этого семейства - *система MS-DOS* (Microsoft Disk Operating System -дисковая операционная система фирмы Microsoft) была выпущена в 1981 году в связи с появлением IBM PC.

Операционные системы семейства DOS являются однозадачными 16 разрядными и обладают следующими особенностями :

1. Интерфейс командной строки
2. Модульность структуры, упрощающая перенос системы на другие типы ЭВМ
3. Небольшой объём доступной без специальных средств оперативной памяти (640 Кбайт)
4. Низкие аппаратные требования, большой объём прикладных программ.

- Существенным недостатком операционных систем семейства DOS является отсутствие средств защиты от несанкционированного доступа к ресурсам ПК и ОС, а также низкая надёжность, отсутствие сетевых возможностей.
- Достоинства системы на сегодняшний день практически исчерпаны, кроме одного: возможность извлечения файлов с дисков при крахе более мощных ОС и невозможности восстановления системы ( с определенными ограничениями на файловую систему ).



# НАЧАЛЬНЫЕ СВЕДЕНИЯ ОБ MS-DOS

Сама *операционная система MS-DOS* (да и любая другая операционная систем тоже) состоит из нескольких частей:

*Загрузчик операционной системы* - это небольшая программа, хранящаяся в первом секторе любой системной дискеты (дискеты с записанной на ней операционной системой) или винчестера, которая загружает в память два системных файла **io.sys** и **msdos.sys**. Именно загрузчику операционной системы передает управление BIOS при начальном старте машины. Файлы **io.sys** и **msdos.sys** при работе постоянно находятся в памяти компьютера: **io.sys** осуществляет дополнение базовой системы ввода-вывода в зависимости от потребностей данной версии операционной системы, а **msdos.sys** реализует все стандартные функции данной версии. Помимо этого, **msdos.sys** загружает в память командный процессор.

*Командный процессор* (файл **command.com**) обслуживает работу системы с пользователем. Он сам выполняет часть команд операционной системы (эти команды называются внутренними), а при вызове внешних команд или выполнении других программ передает им управление, по окончании их работы снова берет управление на себя и выгружает отработавшую программу из памяти.

Внешние команды операционной системы представляют из себя отдельные программы, выполняющие какие-либо сервисные функции.

*Драйверы устройств* - это специальные резидентные программы, их основное назначение - расширение возможностей отдельных устройств компьютера (например, памяти), подключение дополнительного оборудования (скажем, мыши) и обеспечение нормальной работы нестандартных устройств.

# ПРИНЦИПЫ ОРГАНИЗАЦИИ ХРАНЕНИЯ ИНФОРМАЦИИ В КОМПЬЮТЕРЕ

ФАЙЛЫ ФАЙЛЫ, КАТАЛОГИ ФАЙЛЫ, КАТАЛОГИ, ДИСКИ  
(НАКОПИТЕЛИ) ФАЙЛЫ, КАТАЛОГИ, ДИСКИ (НАКОПИТЕЛИ),  
ПРИГЛАШЕНИЕ DOS ФАЙЛЫ, КАТАЛОГИ, ДИСКИ  
(НАКОПИТЕЛИ), ПРИГЛАШЕНИЕ DOS, ЗАПУСК ПРОГРАММ

## ФАЙЛЫ

*Файлы. Имя и расширение файла.*

Все данные в компьютере хранятся в файлах. **Файл** - именованная область памяти на физическом носителе (диске). Файл является учетной единицей информации в ОС. Способ представления информации внутри файла называется форматом файла.

Файлы можно разделить на четыре группы:

- **Исполняемые** — это файлы, которые представляют собой программы и которые можно запустить. Они могут иметь расширение **.COM** или **.EXE**;
- **Пакетные** — они содержат в себе наборы команд **MS-DOS** и тоже могут быть выполнены. Их расширение **.BAT**;
- **Текстовые** — файлы, которые содержат текстовую информацию и которые можно вывести на экран и прочитать. Они имеют расширение **.TXT**, **.DOC** или, если содержат исходный текст программ, **.C**, **.PAS**, **.BAS**, **.ASM** и др.
- Прочие — это **файлы конфигурирования системы, дополнительные файлы программ** и т.д.

Многие редакторы оставляют после своей работы страховочные копии файлов (варианты файлов до их изменения этими программами), эти файлы имеют расширение **.BAK**. Их рекомендуется время от времени просматривать и, если они не содержат полезных данных, удалять.

У каждого файла есть полное имя, которое состоит из имени файла и его типа (расширения), разделенных точкой, по которому компьютер и находит его. Имя файла может состоять из 1-8 букв, цифр, знаков "-" (минус) и "\_" (подчеркивание), расположенных в произвольном порядке, а расширение - до трех включительно. Друг от друга они отделяются символом "." (точка). Это правило можно назвать правилом "8.3". Оно *обязательно* только для операционной системы MS-DOS. Расширение является необязательным параметром и, в принципе, может отсутствовать. При отсутствии типа точку в задании имени файла ставить не обязательно. Тип файла обычно соответствует характеру хранимой информации.

Имя файла задается пользователем, тип также может задаваться пользователем, но иногда определяется программой, создавшей этот файл.

## Примеры полных имен файлов:

COMMAND.COM

AUTOEXEC.BAT

HELP.EXE

GAMES\_2.ARC

BLACK1.PAS

A.EXE

README.TXT

CONFIG.SYS

В названии файлов могут использоваться прописные и строчные латинские буквы (причем, DOS не делает разницы между большими и маленькими буквами, т.е. символ "G" и символ "g" в названии файла будут восприняты совершенно одинаково), символы \$ & # @ ' ! % ^ ( ) { } [ ] - \_ (в ДОС рекомендуется без необходимости не использовать эти специальные символы) и цифры.

Недопустимо использование символов ? \* . , : ; = + \ / | " . Следует заметить, что если Вы попытаетесь ввести более длинное имя, чем это допустимо, то ничего страшного не произойдет: лишние символы будут просто проигнорированы (в силу этого, например, названия файлов **readtext1.txt** и **readtext2.txt** будут восприниматься системой как один файл с названием **readtext.txt**, как, впрочем, и файлы **ReadText.Txt** и, скажем, **READTEXT.TXT**).

Здесь нужно сказать, что название файлов, т.е. имя и расширение файла вместе взятые, практически всегда называют просто именем файла, мы также будем пользоваться в дальнейшем этим соглашением.

## *Характеристики файлов.*

1. [полное] имя файла;
2. длина файла ( в байтах);
3. дата создания (после изменения) файла;
4. время создания (после изменения) файла;
5. специальные признаки - атрибуты:
  - *H* или ***hidden, скрытый файл*** — это файл, имя которого не выводится на экран при выдаче оглавления директории. Этот атрибут, как правило, имеют важные файлы, с которыми пользователь не должен ничего делать, в частности, этот атрибут имеют файлы операционной системы ***io.sys*** и ***msdos.sys***;



**S** или *system, системный файл* — это файл, имеющий важное значение для операционной системы. Трогать эти файлы пользователю не рекомендуется. Файлы операционной системы **io.sys** и **msdos.sys** имеют также и этот атрибут. Файлы с атрибутом *system*, кроме того, являются непереключаемыми, т.е. располагаются всегда на одном и том же месте диска;

**A** или *archive, неархивный файл* — этот атрибут означает, что файл не является архивной копией. Он установлен практически у всех файлов, доступных пользователю;

**R** или *read-only, только чтение* — файлы, запись в которые невозможна. У файлов **io.sys** и **msdos.sys** установлен и этот атрибут. Удалить с диска такие файлы можно только убрав этот атрибут (имеются в виду средства операционной системы, различные оболочки позволяют это делать напрямую).

# Шаблоны имён файлов.

*Шаблон имени файла* употребляется в командах для обозначения сразу нескольких файлов (там, где это имеет смысл и допустимо).

В шаблонах употребляются знаки:

"\*" ("звездочка") - любое число любых допустимых символов.

"?" (вопросительный знак) - один любой символ в имени и типе файла.

## Примеры:

**LI\*.\*** - все файлы с именами, начинающимися с LI и любыми типами (например, LIST.TXT, LINUX.DOC, LIO1, LIPP22.123).

**T\*.L\*** - все файлы, имена которых начинаются с T, а типы с L (например, TOP.LIT, TTT.LLL, T1234.L23, но не TTT.2L3).

\*.\* - все файлы текущего каталога.

\*. - все файлы без типов.

????.TXT - все файлы типа TXT с именами не более чем из четырех букв.

A?? - все файлы, имена которых начинаются с A и содержат не более двух букв, и с типами, состоящими не более чем из одной буквы.

## ЗАРЕЗЕРВИРОВАННЫЕ ИМЕНА

В операционной системе существует несколько зарезервированных имен, которые пользователь не может использовать для своих целей. Эти имена присвоены определенным устройствам, и когда происходит обращение к такому файлу, то на самом деле происходит работа с данным устройством. Эти имена и устройства следующие:

- **AUX** — асинхронный интерфейс (вспомогательный асинхронный выход);
- **COM1** — первый последовательный порт;
- **COM2** — второй последовательный порт;
- **COM3** — третий последовательный порт;
- **COM4** — четвертый последовательный порт;
- **LPT1** — первый параллельный порт;
- **LPT2** — второй параллельный порт;
- **LPT3** — третий параллельный порт;
- **PRN** — принтер;
- **CON** — консоль (при выводе — экран, при вводе — клавиатура);
- **CLOCK\$** — драйвер часов;
- **NUL** — пустое устройство (все операции с этим устройством игнорируются).

Для всех зарезервированных имен справедливо одно правило: при наличии расширения это расширение игнорируется, т.е. имена **PRN.PRN**, **PRN.N1** и **PRN** будут значить одно и то же устройство PRN, или принтер. Если же Вы все-таки попытаетесь работать с такими файлами, то сообщения об ошибках, получаемые в этих случаях, будут далеко не всегда говорить об истинной причине возникновения аварийной ситуации (например, при работе с файлом **NUL.TXT** вполне можно получить сообщение "файл пуст"). Но все эти ограничения не имеют никакого отношения к собственно расширениям, например, файл **erson.prn** имеет полное право на существование.

# КАТАЛОГИ ФАЙЛОВ

При небольших объемах накопителей количество файлов, которые могут быть на них записаны, относительно не велико, если же мы попытаемся записать все нужные нам файлы, скажем, на винчестер, то при попытке найти среди них файл с определенным именем у нас возникнут весьма серьезные трудности, т.к. отыскать среди нескольких сотен файлов один не так-то просто. Для решения этой проблемы были придуманы каталоги. Следует отметить то, что без каталогов нельзя было бы записать два файла с одним и тем же именем на накопитель. Каталоги можно сравнить с бумажными папками, в которой могут находиться документы - файлы и (или) другие папки - каталоги.

***Каталог (директория)*** - это группа файлов на одном носителе, объединенных по какому-либо критерию.

Этот критерий определяется пользователем, обычно это файлы, относящиеся к какой-либо системе, то есть логически взаимосвязанные. Каталоги можно рассматривать как разделы внешней памяти, с каждым из которых можно работать независимо.

## Классификация каталогов.

На каждом накопителе всегда имеется один *основной каталог*, называемая также *корневым* (или *root-dir*). В нём хранятся различные каталоги и файлы, про которые говорят, что они находятся в корневом каталоге. Каждый каталог (в этом случае он будет называться *надкаталогом*), в свою очередь, может содержать другие каталоги (они будут называться *подкаталогами этого каталога*) и файлы и т.д. В результате на диске получается древовидная структура (напоминающая по внешнему виду при ее графическом изображении дерево). Пример такой структуры можно увидеть на рисунке:

Корневой каталог C:\

- Autoexec.bat
- Command.com
- Config.sys
- Config.dos
- Io.sys
- MsDOS.sys

Arc

Acrobat3

Reader

ActivX

Browser

Fonts

Help

Optional

Plug\_ins

Pfm



# М А Р Ш Р У Т - префикс имени файла.

Для точной идентификации файла необходимо, кроме имени, указать его точное местоположение в каталоговой системе, так как одноименные и различные по содержанию файлы могут находиться в нескольких каталогах. **Местоположение файла - цепочка подчиненных каталогов - называется маршрутом.**

При задании маршрута он отделяется от имени файла разделителем "\".

Файл полностью задается следующими элементами:

- *именем накопителя* (может отсутствовать);
- *маршрутом* - цепочкой подчиненных каталогов (может отсутствовать);
- *именем файла*, который должен содержаться в последнем упомянутом каталоге.

Если имя накопителя опущено, рассматривается текущий накопитель. Если маршрут не указан, то местоположение файла вычисляется от текущего каталога. Если маршрут начинается со знака "\", то местоположение вычисляется от корневого каталога текущего накопителя.

Пример:

`\DOS\ALFA.EXE` - файл ALFA.EXE находится в каталоге DOS на текущем накопителе.

## ДИСКИ (НАКОПИТЕЛИ НА ДИСКАХ)

Все файлы в компьютерах хранятся на *дисках*, это могут быть *дискеты* или *жесткий диск (винчестер)* или *лазерных дисках (CD)*. Если продолжать аналогию, то диски можно сравнить с ящиками стола. Как в каждом ящике может храниться некоторое количество документов, так и на каждый диск можно записать некоторое количество файлов.

# ИМЕНА ДИСКОВ.

Для того чтобы отличать диски друг от друга, каждому из них присваивается свое имя. Так флоппи дисководы имеют имена **A:** и **B:**, а если он один, то только **A:**, винчестер всегда имеет имя **C:**, а если на машине стоят два жестких диска или один винчестер разбит на два раздела, то их имена будут соответственно **C:** и **D:**, в последнем случае их (диски) называют логическими дисками. Следует отметить, что загрузка операционной системы может производиться с дисковода с именем **A:**, или винчестера с именем **C:**.

## ПРИГЛАШЕНИЕ DOS.

Приглашение к вводу команд выдается операционной системой, если она ожидает действий пользователя.

Стандартное приглашение DOS указывает имя рабочего накопителя. Иногда в приглашение вставляется маршрут текущего каталога.

Примеры приглашений:

- ✓ **A:\ >** - указывает на рабочий накопитель A:
- ✓ **C:\WORK >** - приглашение, указывает на рабочий накопитель C: и на текущий каталог первого уровня **WORK**

## ЗАПУСК ПРОГРАММ.

Для запуска программы необходимо перейти на рабочий накопитель, на котором находится нужная программа. Эта операция осуществляется следующей командой: в ответ на приглашение вводится имя нужного накопителя.

Пример:

**C:\>a:** - переход на накопитель A: с текущего накопителя C:

Если программа находится не в корневом каталоге, то она в этом случае недоступна. Для запуска такой программы необходимо перейти с помощью команд ДОС в соответствующий каталог

Запускать можно только программы, имеющие определенные типы: COM, EXE и BAT.

Файлы типов EXE и COM являются *программами в машинных кодах*, файлы типа BAT - *командные* (то есть состоят из команд DOS).

Файлы типов EXE и COM различаются лишь способом загрузки с внешних носителей и размещением в памяти. Для пользователя эта разница не ощутима.

Для запуска программы достаточно в качестве команды указать имя соответствующего файла без типа.

Некоторые программы требуют задания параметров. Параметрами могут быть числовые или текстовые константы или имена файлов.

**Примеры:**

**C:\>tetris** - запуск игровой программы TETRIS.COM.

**C:\>turbo first.pas** - запуск системы TURBO.EXE с параметром - текстовым файлом FIRST.PAS.

**Операционная система MSDOS – устарела. Последняя ее версия - MSDOS 6.22 и больше не обновлялась. Тем не менее, она имеет право на существование и применение и в сегодняшних условиях. Типичная ситуация: одна из современных ОС претерпела крах (в результате действия вирусных программ, неверных и фатальных действий пользователя или по каким-то другим причинам) и восстановление ее невозможно. В то же время на жестком диске располагаются важные для пользователя файлы, потеря которых крайне не желательна (тексты, документы, фото, мультимедийные файлы и т.п.).**

**Именно в этом случае приходит на помощь «старая, добрая» MSDOS. Она может быть загружена с 3-х дюймовой дискеты (в силу ее малого объема) и средствами этой ОС нужные файлы могут быть скопированы на другой носитель и, следовательно, «спасены». Эта операция применима даже для другой файловой системы (например, NTFS), конечно с некоторой модификацией самой ОС, например, NTFSDOS.PRO – одной из разновидностей MSDOS. Для этого необходимо знать некоторые основные команды системы MSDOS (всего их свыше 150, они рассматриваются в любом справочнике по MSDOS).**

# ОСНОВНЫЕ КОМАНДЫ ОПЕРАЦИОННОЙ СИСТЕМЫ MSDOS.

1. Вывод каталога на экран – *dir* (от *directory* – директория, каталог)

```
A:\>dir
Том в устройстве A не имеет метки.
Серийный номер тома: 2A87-6CE1

Содержимое папки A:\

08.06.2000   17:00           58 870  EGA2.CPI
08.06.2000   17:00           58 753  EGA3.CPI
08.06.2000   17:00           58 870  EGA.CPI
08.06.2000   17:00          21 607  KEYB.COM
08.06.2000   17:00          34 566  KEYBOARD.SYS
08.06.2000   17:00          31 942  KEYBRD2.SYS
08.06.2000   17:00          31 633  KEYBRD3.SYS
08.06.2000   17:00          13 014  KEYBRD4.SYS
08.06.2000   17:00          29 239  MODE.COM
08.06.2000   17:00          93 040  COMMAND.COM
08.06.2000   17:00          17 175  DISPLAY.SYS
25.03.2008   21:41           96     AUTOEXEC.BAT
25.03.2008   21:41           33     CONFIG.SYS
09.09.2008   22:16           <DIR>      katalog
                13 файлов           448 838 байт
                1 папок           887 808 байт свободно
```

```
A:\>
```



После набора этой команды на экран выводится содержимое текущего каталога (на рисунке – каталог дискеты A: В каждой строке выводится дата, время, размер(в байтах) и имя файла. Каталоги имеют метку <DIR> - на рисунке **katalog**

2. Переход в другой каталог – **cd** (Change *Dir* – смена каталога)

```
A:\>cd katalog  
A:\katalog>
```

После выполнения этой команды происходит переход в папку **katalog**. Команда **CD..** – переход на каталог старшего уровня (вплоть до корневого).

### 3. Создание нового каталога в текущем *md* (*Make Dir* – «сделать» каталог)

```
A:\>md katalog2
```

После выполнения этой команды будет создан каталог с именем *katalog2*. Проверить можно командой *dir*:

```
A:\>dir
Том в устройстве A не имеет метки.
Серийный номер тома: 2A87-6CE1

Содержимое папки A:\

08.06.2000  17:00           58  870  EGA2.CPI
08.06.2000  17:00           58  753  EGA3.CPI
08.06.2000  17:00           58  870  EGA.CPI
08.06.2000  17:00           21  607  KEYB.COM
08.06.2000  17:00           34  566  KEYBOARD.SYS
08.06.2000  17:00           31  942  KEYBRD2.SYS
08.06.2000  17:00           31  633  KEYBRD3.SYS
08.06.2000  17:00           13  014  KEYBRD4.SYS
08.06.2000  17:00           29  239  MODE.COM
08.06.2000  17:00           93  040  COMMAND.COM
08.06.2000  17:00           17  175  DISPLAY.SYS
25.03.2008  21:41             96  AUTOEXEC.BAT
25.03.2008  21:41             33  CONFIG.SYS
09.09.2008  22:16             <DIR>   katalog
09.09.2008  22:36             <DIR>   katalog2
                13 файлов           448 838 байт
                2 папок           886 784 байт свободно
```

#### 4. Копирование файлов и каталогов *сору*

Формат команды: *сору* имя файла(каталога) источника  
имя файла(каталога) приемника.

```
C:\>сору c:\TP\DOC\crt.int a:  
Скопировано файлов: 1.
```

По этой команде будет скопирован файл **crt.int** из подкаталога DOC каталога TP логического диска C: в корневой каталог дискеты A: под тем же именем.

## 5. Вывод содержимого файла на экран *type*

```
A:\>type crt.int
```

```
{*****}  
{  
{ Turbo Pascal Version 7.0 }  
{ CRT Interface Unit }  
{  
{ Copyright (C) 1988,92 Borland International }  
{  
{*****}
```

```
unit Crt;
```

```
{ $D-, I-, S- }
```

```
interface
```

```
const
```

```
{ CRT modes }
```

```
BW40      = 0;      { 40x25 B/W on Color Adapter }  
CO40      = 1;      { 40x25 Color on Color Adapter }  
BW80      = 2;      { 80x25 B/W on Color Adapter }
```

По этой команде на экран выводится содержимое файла **crt.int**. Если содержимое файла имеет текстовый формат, то текст выводится на экран. В противном случае на экране будет выведен беспорядочный набор символов:

```
A:\>type command.com
MZp@||  -◆  @  ▲  @
        6363ENU@  |@NSCO
```

Все перечисленные команды являются внутренними, т.е. воспринимаются командным процессором операционной системы непосредственно. Большинство остальных команд MSDOS являются внешними, т.е. существуют в виде отдельных файлов в каталоге операционной системы DOS.

# ОС семейства Windows.

## **WINDOWS: развитие.**

На сегодняшний момент операционная система Windows фирмы Microsoft во всех ее проявлениях бесспорно считается самой распространенной операционной системой на ПК: в мире более 150 млн. IBM PC-совместимых компьютеров, и система Windows установлена на 100 млн. из них. Очевидно, что ознакомление с ПК необходимо начинать с ознакомления с Windows, ведь без нее работа на ПК немыслима для большинства пользователей. Первоначально, Windows, разрабатывалась не как операционная система, какой мы привыкли видеть современные ее версии, а как графическая оболочка MS-DOS. Надо отметить, что концепция графического интерфейса была разработана отнюдь не Microsoft.

Уже за несколько лет до внедрения Windows существовали компьютеры Apple Macintosh, с графической операционной системой MacOS, интерфейс которой был более дружелюбным и понятным рядовому пользователю, в отличие от командной строки MS-DOS. Строго говоря, Windows, не единственная попытка избавить пользователя от командной строки на IBM-совместимых компьютерах. Очень известной, в свое время была псевдографическая (на самом деле работавшая в текстовом режиме), оболочка Norton Commander корпорации Symantec.

# WINDOWS: Начало

В те времена, когда работа над Windows только начиналась, считалось, что будущее принадлежит интегрированным средам.

Позиция фирмы Microsoft была иной. После посещения исследовательского центра Xerox PARC президент фирмы Билл Гейтс решил заняться созданием графической среды, которая послужила бы стандартной платформой для разработчиков прикладных программ.

Итак, Microsoft поставила себе задачу по созданию платформы для разработчиков. Предполагалось обеспечить разработчиков встроенными функциями для реализации пользовательского интерфейса и его компонентов, которые могли бы управляться с помощью клавиатуры или мыши.



Когда работа над средой Windows только начиналась, в распоряжении Microsoft был компьютер с процессором 8088 и максимальным объемом памяти 640 Кбайт. Предполагалось, что пользователи этой среды имеют порядка 256 Кб памяти, а жесткий диск им не доступен из-за высокой цены. Наилучшим графическим адаптером в те времена был CGA с разрешением 320 на 200 при работе с четырьмя цветами. Адаптер Hercules обеспечивавший разрешение 720 на 348 в монохромном режиме был тогда новинкой. Работа графической среды в таких условиях была безумно медленной. В 1983 г стали просачиваться первые сведения о разработке Windows.

И, наконец,

# Windows 1.0

Представлен на выставке  
Comdex, проходившей в ноябре  
1985 года в Лас-Вегасе.

# Два пути развития.

Windows – повсеместно распространенная, по существу, стандартная многозадачная операционная система для современных IBM – совместимых компьютеров.

Существуют два главных её семейства: условно говоря, для домашнего использования (Windows версии 3.1, 3.11, 95, 98 и Millennium Edition) и для профессиональных применений ( Windows NT версий 3.5, 4.0, 2000, XP и VISTA). Системы первого типа ставят, конечно же, не только на домашних компьютерах, но и в большинстве учреждений, фирм, институтов. NT же используют тогда, когда хотят создать локальную или глобальную сеть предприятия или учебного заведения (впрочем, тут у Windows NT есть серьёзные конкуренты ) или когда необходимо обеспечить повышенную устойчивость системы, что, вообще говоря, желательно для всех, но для профессионалов просто жизненно необходимо.

Фирма Microsoft собирается постепенно перевести всех домашних пользователей на новую технологию (NT в переводе – New Technology), для чего, и созданы были системы Windows 2000 , а затем – XP и VISTA. Разработка «старой технологии» прекращена – за Millennium ничего уже не будет.

# *Преимущества и недостатки Windows.*

## *Преимущества.*

### *- Удобство и поддержка устройств.*

Основное отличие программ для DOS и для Windows состоит в том, что DOS-программа может работать с аппаратными средствами компьютера (монитором, клавиатурой, принтером и т. д.) непосредственно минуя DOS (и, как правило, поступает именно таким образом), в то время как Windows-программа должна обращаться к внешним устройствам только через посредство Windows. Поэтому после установки в Windows драйвера («специализированная программа»), обеспечивающего поддержку данного устройства (то есть настраивающего Windows на особенности данного устройства) все Windows-программы могут работать с этим устройством. Это ликвидирует весьма болезненную для DOS проблему обеспечения совместимости программ с конкретными устройствами.

## *-Единый пользовательский интерфейс.*

Windows представляет программистам все необходимые средства для создания пользовательского интерфейса, поэтому программисты пользуются ими, а не изобретают аналогичные собственные средства. Вследствие этого пользовательский интерфейс Windows-программ в значительной степени унифицирован, и пользователям не требуется изучать для каждой программы новые принципы организации взаимодействия с этой программой.

## *- Поддержка масштабируемых шрифтов.*

В таких приложениях, как редактирование документов, издательское и рекламное дело, создание таблиц или презентаций и т.д., необходимо использование большого количества шрифтов – текстовых, заголовочных, декоративных, пиктографических и других, причем символы этих шрифтов могут потребоваться в самых различных размерах. Поэтому в Windows 3.1 была встроена поддержка масштабируемых шрифтов формата True Type.

## *- Поддержка мультимедиа.*

- Одним из усовершенствований Windows явилась поддержка мультимедиа. При подключении соответствующих устройств Windows может воспринимать звуки от микрофона, компакт-диска или MIDI – синтезатора, изображения от цифровой видеокамеры или с компакт-диска, выводить звуки и движущиеся изображения.
- Это открывает большие возможности для обучающих, игровых и других программ, позволяя делать общение с компьютером более лёгким и приятным даже для непрофессионалов.



## *- Многозадачность.*

- Windows обеспечивает возможность одновременного выполнения нескольких программ и переключения с одной программы на другую.

## *- Средства обмена данными.*

Windows предлагает следующие способы:

- *Буфер обмена данными*
- *Динамический обмен данными*
- *Механизм связи и внедрения объектов*

## *- Совместимость с DOS-программами.*

- Работа в среде Windows не вынуждает отказываться от использования DOS программ. Более того, для запуска DOS программ, как правило, нет необходимости выходить из Windows. Однако следует заметить, что DOS программы под управлением Windows выполняются медленнее.

## *Недостатки.*

- преимущества Windows достигаются за счет значительного увеличения нагрузки на аппаратные средства компьютера.
- Программы с приблизительно одинаковыми возможностями для DOS и для Windows отличаются по требованиям к компьютерным ресурсам в несколько раз.
- Windows в принудительном порядке заставляет программистов использовать средства программного интерфейса Windows (API) - это более 600 функций. Кроме того, программист должен для этого свободно владеть весьма сложными концептуальными понятиями объектно-ориентированного программирования, оперировать с объектами, сообщениями, разделяемыми ресурсами и т.д., поэтому обучение программированию под Windows весьма непростое дело.

# Операционная система OS/2

В базовой комплектации *OS/2* по функциональным возможностям нисколько не уступает *Windows NT*. Благодаря «ударному» труду разработчиков система оказалась менее требовательна к ресурсам компьютера и намного проще в настройке.

*OS/2 Warp* — не многопользовательская система в широком смысле слова, однако она позволяет полностью контролировать права пользователей и гибко управлять ими при использовании таких сетевых приложений, как telnet, ftp, или при построении одноранговой сети.

## Основные характеристики:

## - *Модульная 32-разрядная архитектура*

*OS/2 Warp 4.0* является полностью 32-разрядной системой. В последней версии системы абсолютно все компоненты и драйверы устройств выполнены 32-разрядными. Система также способна выполнять старые 16-разрядные Windows - и Win32s – приложения благодаря поставляемой в ее составе версии Windows 3.1 оригинальной разработки IBM. Приложения MS-DOS выполняются на собственных виртуальных DOS-машинах. По маркетинговым соображениям поддержка приложений *Windows 95* и *NT* не реализована.

## *- Линейное адресное пространство памяти*

*OS/2* работает не с сегментной моделью памяти, а со всей системной памятью как с единым непрерывным пространством. Сегментная архитектура памяти несет в себе значительные ограничения на размер приложений и используемых блоков памяти.

Все приложения выполняются в собственном адресном пространстве, что обеспечивает высокую стабильность и нечувствительность к ошибкам в программах. При ошибке выполнения приложения система прекращает его работу и записывает специальный log-файл.

## - *Многопроцессорная поддержка*

Операционная система *OS/2* не является многоплатформенной, однако поддерживает SMP-спецификацию многопроцессорной обработки.

## - *Вытесняющая многозадачность*

Как и *Windows NT*, операционная система *OS/2 Warp* использует вытесняющую многозадачность. При этом она полностью контролирует все приложения и распределяет между ними процессорное время, тем самым сильно снижая вероятность «подвешивания» системы ошибочной программой. Механизм распределения системных ресурсов между приложениями в *OS/2* более гибок по сравнению с *Windows NT*. Опытный пользователь может изменить настройки, используемые по умолчанию, для достижения максимальной производительности какого-либо приложения или группы задач.

## - *Многопоточность*

*OS/2 Warp* поддерживает многопоточность. Все основные компоненты системы созданы с использованием этого механизма. Кроме того, все сетевые драйверы и приложения являются многопоточными, что повышает производительность работы с сетью.

## - *Файл config.sys*

В отличие от *Windows NT*, в которой все настройки и параметры работы хранятся в специальном системном реестре, в системе *OS/2* часть настроек (драйверы устройств и файловой системы, переменные окружения, параметры работы планировщика задач и т. д.) определяется в текстовом файле *config.sys*. Такой подход представляется несколько старомодным, но гарантирует простоту и ясность процесса настройки и оптимизации. Все остальные системные параметры хранит двоичный файл *os2.ini*. Таким образом, централизованной базы данных с информацией о конфигурации системы не существует.



Файл *os2.ini* выполняет только часть функций системного реестра, отвечая за настройку графического интерфейса и некоторых других параметров, определяемых пользователем.

### - *Улучшенная сетевая поддержка*

Система OS/2 реализует поддержку одноранговых и доменных сетей. Доменные сети строятся на основе компьютеров с системой **OS/2 Warp Server**, регистрирующих пользователей в сети и хранящих информацию об их учетных карточках. OS/2 Warp может работать со всеми основными сетевыми протоколами, включая **TCP/IP**, **IPX/SPX** и **NetBEUI**

### - *Файловая система OS/2 Warp*

В качестве основной файловой системы в OS/2 используется так называемая **High Performance File System (HPFS)**. Однако для обеспечения совместимости со старыми приложениями оставлена поддержка **FAT**.

# ОПЕРАЦИОННАЯ СИСТЕМА UNIX

Операционная система *UNIX* - это набор программ, который управляет компьютером, осуществляет связь между вами и компьютером и обеспечивает вас инструментальными средствами, чтобы помочь вам выполнить вашу работу.

Разработанная, чтобы обеспечить легкость, эффективность и гибкость программного обеспечения, система *UNIX* имеет несколько полезных функций:

- основная цель системы - это *выполнять широкий спектр заданий и программ;*
- *интерактивное окружение*, которое позволяет вам связываться напрямую с компьютером и получать немедленно ответы на ваши запросы и сообщения;

- *многопользовательское окружение*, которое позволяет вам разделять ресурсы компьютера с другими пользователями без уменьшения производительности. Этот метод называется *разделением времени*. Система *UNIX* взаимодействует с пользователями поочередно, но так быстро, что кажется, что взаимодействует со всеми пользователями одновременно;
- *многозадачное окружение*, которое позволяет вам выполнять более одного задания в одно и тоже время.

**Система *UNIX* имеет 4 основных компонента:**

**1) *ядро* -**

*это программа, которая образует ядро операционной системы; она координирует внутренние функции компьютера ( такие как размещение системных ресурсов). Ядро работает невидимо для вас;*

## 2) *shell* -

это программа, которая осуществляет связь между вами и ядром, интерпретируя и выполняя ваши команды. Так как она читает ваш ввод и посылает вам сообщения, то описывается как интерактивная;

## 3) *commands* -

это имена программ, которые компьютер должен выполнить. Пакеты программ называются инструментальными средствами. Система UNIX обеспечивает инструментальными средствами для таких заданий как создание и изменение текста, написание программ, развитие инструментария программного обеспечения, обмен информацией с другими посредством компьютера;

## 4) *file system* -

файловая система - это набор всех файлов, возможных для вашего компьютера. Она помогает вам легко сохранять и отыскивать информацию.

# Ядро

Ядро контролирует доступ к компьютеру, управляет памятью компьютера, обслуживает файловую систему и распределяет ресурсы компьютера среди пользователей. На рис. 1 приведено функциональное представление ядра.

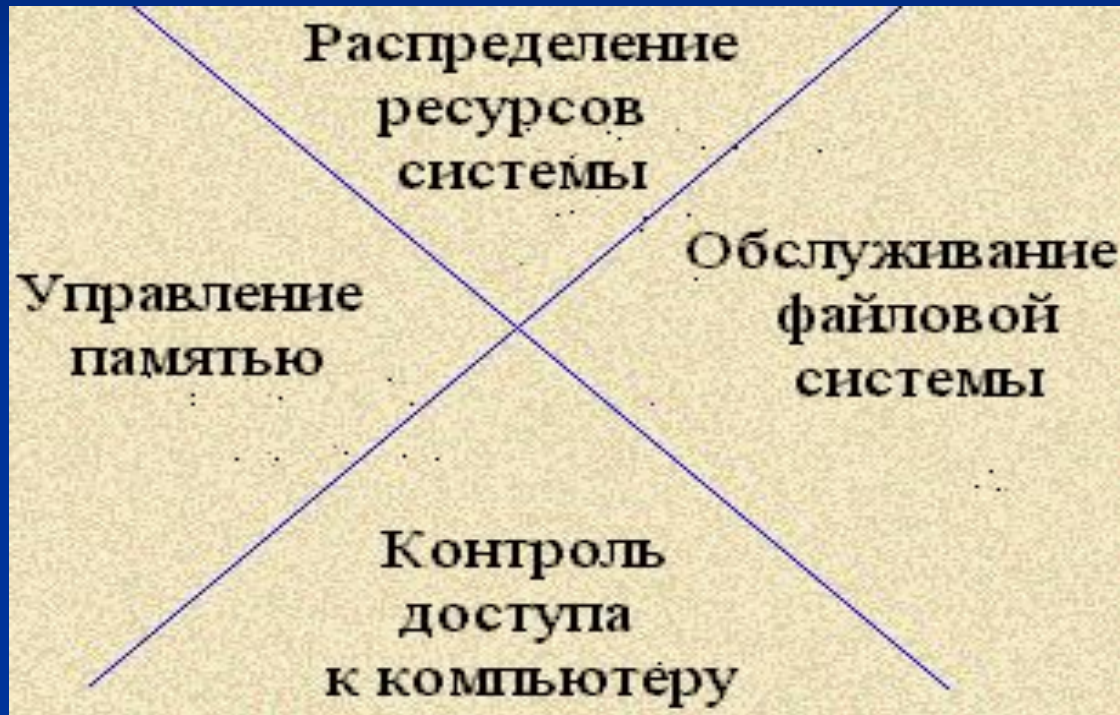


Рис.1

# Shell

*shell* - это программа, которая позволяет вам связываться с операционной системой. *shell* считывает команды, которые вы вводите, и интерпретирует их как запросы на выполнение других программ, на доступ к файлу или обеспечение вывода. *shell* также является мощным языком программирования, не похожим на язык программирования Си, который обеспечивает условное выполнение и управление потоками данных.

*Программа* - это набор инструкций для компьютера. Программы, которые могут быть выполнены компьютером без предварительной трансляции, называются исполняемыми программами или командами. Как обычному пользователю системы *UNIX* вам доступно множество стандартных программ и инструментальных средств. Если вы используете систему *UNIX* для написания программ и развития программного обеспечения, то вы также можете задействовать системные вызовы, подпрограммы и другие инструментальные средства. И, конечно, любая написанная вами программа будет в вашем распоряжении.

Внешний круг системы *UNIX* образуют *программы* и *инструментальные средства системы*, разделенные на категории функционально. Эти функции включают:

- ***программное окружение*** -

несколько программ системы *UNIX*, устанавливающих дружественное программное окружение, обеспечивающее интерфейсы между системой и языками программирования и использование обслуживающих программ;

- ***обработка текстов*** -

система обеспечивает программы, такие как строковый и экранный редакторы, для создания и изменения текстов, орфографическую программу проверки для обнаружения ошибок орфографии, и необязательный форматор текста для создания высококачественных копий, которые подходят для публикаций;

- *организация информации* -

система предоставляет много программ, которые позволяют вам создавать, организовывать и удалять файлы и каталоги;

- *обслуживающие программы* -

*инструментальные средства*, создающие графику и выполняющие вычисления;

- *электронная связь* -

несколько программ (например, **mail**) предоставляют вам возможность передавать информацию другим пользователям и в другие системы *UNIX*.



# Как выполнять команды

Чтобы ваш запрос был понятен системе *UNIX* вы должны ввести каждую команду в корректном формате или синтаксисе командной строки. Этот синтаксис определяет порядок, в котором вы вводите компоненты командной строки. И вы должны расположить все составные части командной строки в требуемом синтаксисом порядке, иначе *shell* не сможет интерпретировать ваш запрос.

Пример синтаксиса командной строки:

**command option(s) argument(s) <CR>**

Для каждой командной строки системы *UNIX* вы должны ввести как минимум два компонента: имя команды и клавишу <RETURN>. (Обозначение <CR> используется в документации как инструкция для нажатия клавиши RETURN).

Командная строка может также содержать ключи и аргументы. В указанном примере синтаксиса командной строки:

***command*** - это имя программы, которую вы хотите выполнить;

***option*** - ключи, которые указывают как запустить команду;

***argument*** - указывает на данные, которые эта команда обрабатывает, обычно это имя каталога или файла.

## Файловая система

***Файловая система*** является краеугольным камнем операционной системы UNIX. Она обеспечивает логический метод организации, восстановления и управления информацией. Файловая система имеет иерархическую структуру

Файл, который является основной единицей системы *UNIX*, может быть: *обыкновенным файлом, справочником, специальным файлом* или *символическим каналом связи*.

- *Обыкновенные файлы* являются набором символов.

*Обыкновенные файлы* используются для хранения любой информации. Они могут содержать тексты для писем или отчетов, коды программ, которые вы написали, либо команды для запуска ваших программ. Однажды создав обыкновенный файл вы можете добавить нужный материал в него, удалить материал из него, либо удалить файл целиком.

- *Справочники* являются супер-файлами, которые могут содержать файлы или другие справочники. Обычно файлы, содержащиеся в них, устанавливают отношения каким-либо способом. Например, справочник, названный **sales** может хранить файлы, содержащие цифры ежемесячных продаж, названные **jan, feb, mar**, и т.д.

Вы можете создать каталоги, добавить или удалить файлы из них или удалить каталоги.

Все справочники, которые вы создаете, будут размещены в вашем собственном справочнике. Этот справочник назначается вам системой во время входа в систему. Никто кроме привилегированных пользователей не может читать или записывать файлы в этот справочник без вашего разрешения и вы определяете структуру этого справочника.

Система *UNIX* также содержит несколько справочников для собственного использования. Структура этих справочников аналогична во всех системах *UNIX*. Этот справочник, включающий в себя несколько системных справочников, размещается непосредственно под справочником **root**. Справочник **root** (обозначенный /) является исходным в файловой структуре *UNIX*. Все справочники и файлы иерархически располагаются ниже.

- **Специальные файлы** соответствуют физическим устройствам, таким как терминал, дисковое устройство, магнитная лента или канал связи. Система читает и записывает из/в специальные файлы также как и в обыкновенные файлы. Однако запросы системы на чтение и запись не приводят в действие нормальный механизм доступа к файлу. Вместо этого они активизируют драйвер устройства, связанный с файлом, приводя, возможно, в действие головки диска или магнитной ленты.

## Расположение системы

Некоторые операционные системы требуют от вас определить тип файла и использовать его определенным способом. От этого будет зависеть то, как будут файлы сохраняться, т.к. файлы могут быть последовательными, двоичными или произвольной выборки.

Для системы *UNIX* все файлы одинаковы. Это делает файловую структуру *UNIX* легкой в использовании. Например, вам нет необходимости указывать требования к памяти для ваших файлов, т.к. система автоматически это сделает для вас. Или если вам или написанной вами программе необходим доступ к определенному устройству (например, принтеру) вы указываете устройство также как любой из ваших файлов. В системе *UNIX* существует только один интерфейс для всего вашего ввода и вывода для вас; это упрощает ваше взаимодействие с системой.

Справочник *root* содержит несколько важных системных справочников:

- **/stand -**

содержит загрузочные программы и файлы данных, используемые в процессе загрузки;

- **/sbin** -

содержит основные исполняемые программы, используемые в процессе загрузки и при восстановлении системы;

- **/dev** -

содержит специальные файлы, которые представляют периферийные устройства (например, консоль, построочно-печатное устройство, терминалы пользователя и диски);

- **/etc** -

содержит машинно-зависимые файлы конфигурации и базы данных организации системы;

- **/home** -

корневой справочник поддеревя для справочника пользователя;

- **/tmp** -

содержит временные файлы;

- **/var** -

корневой справочник поддерева для изменяющихся файлов (например, файлы log);

- **/usr** -

содержит другие справочники, включая lib и bin.



# ОПЕРАЦИОННАЯ СИСТЕМА LINUX

*Linux* - это современная POSIX-совместимая и Unix-подобная операционная система для персональных компьютеров и рабочих станций.

Это многопользовательская сетевая операционная система с сетевой оконной графической системой *XWindow System*. *ОС Linux* поддерживает стандарты открытых систем и протоколы сети *Internet* и совместима с системами Unix, DOS, MS Windows. Все компоненты системы, включая исходные тексты, распространяются с лицензией на свободное копирование и установку для неограниченного числа пользователей.

Разработка *ОС Linux* выполнена **Линусом Торвалдсом** (Linus Torvalds) из университета Хельсинки и не поддающейся подсчету обширной командой из тысяч пользователей сети *Internet*, сотрудников исследовательских центров, фондов, университетов и т.д.

# Возможности, которые предоставляет *ОС Linux*

- дает возможность бесплатно и легально иметь современную ОС для использования как на работе, так и дома;
- обладает высоким быстродействием;
- работает надежно, устойчиво, совершенно без зависаний;
- не подвержена вирусам;
- позволяет использовать полностью возможности современных ПК, снимая ограничения, присущие DOS и MS Windows по использованию памяти машины и ресурсов процессора(ов);
- эффективно управляет многозадачностью и приоритетами, фоновые задачи (длительный расчет, передача электронной почты по модему, форматирование дискеты и т.д. и т.п.) не мешают интерактивной работе;

- позволяет легко интегрировать компьютер в локальные и глобальные сети, в т.ч. в Internet; работает с сетями на базе **Novell** и **MSWindows**;
- позволяет выполнять представленные в формате загрузки прикладные программы других ОС - различных версий **Unix**, **DOS** и **MSWindows**;
- обеспечивает использование огромного числа разнообразных программных пакетов, накопленных в мире **Unix** и свободно распространяемых вместе с исходными текстами;
- предоставляет богатый набор инструментальных средств для разработки прикладных программ любой степени сложности, включая системы класса клиент-сервер, объектно-ориентированные, с многооконным текстовым и/или графическим интерфейсом, пригодных для работы как в **Linux**, так и в других ОС;
- дает пользователю и особенно разработчику замечательную учебную базу в виде богатой документации и исходных текстов всех компонент, включая ядро самой ОС;

- дает всем желающим попробовать свои силы в разработке, организовать общение и совместную работу через Internet с любыми из разработчиков ОС Linux и сделать свой вклад, став соавтором системы.

## Кому и зачем может понадобиться ОС *Linux*

- *Linux* это полноценная 32-х разрядная (64-х разрядная на платформе DEC AXP) операционная система, которая использует компьютер на полную мощность. *Linux* превращает персональный компьютер IBM PC в настоящую рабочую станцию. По цене персоналки, которая значительно ниже стоимости рабочей станции.

- ❑ Выигрыш в цене очень большой, так как помимо экономии на оборудовании, программное обеспечение в *Linux* поставляется со свободной лицензией, разрешающей бесплатное неограниченное копирование системы. Ядро, редакторы, трансляторы, СУБД, сеть, графические интерфейсы, игры и масса другого программного обеспечения объемом в тысячи мегабайт - бесплатно и на законной основе.
- ❑ Выигрыш только на программном обеспечении может составить от тысяч до десятков тысяч долларов. Для многих пользователей в России свободная лицензия - это единственная возможность легально снабдить себя полноценным набором программного обеспечения.
- ❑ Для пиратского рынка не платить за программы - обычное дело. Однако в случае с *Linux* мало того, что за это никто преследовать не будет, еще и полную документацию дадут! Более того, с исходными текстами всех программ! Пиратам такое и не снилось.

- Большой интерес *Linux* представляет для потребителей и разработчиков прикладных систем. Представьте себе фирму, состоящую из нескольких филиалов, территориально разнесенных по разным районам города и даже по разным городам и странам. На головном предприятии работает сервер базы данных, клиенты - рабочие места в филиалах - через сеть взаимодействуют с сервером. Такая система быстро, дешево и удобно делается в *Linux*. Вспомните об устойчивости *Linux*.
- Допустим, вы разработчик коммерческих программ для коммерческих ОС, например, **SCO Unix**, **Solaris** или **DOS**. Вы используете многоплатформенный компилятор, мощную многооконную систему отладки, эмуляторы и системы совместимости. Все это есть в *Linux*. Можете поставить его на свою домашнюю персоналку и, если у вас есть модем, ходить на работу только за зарплатой. Кстати, и здесь устойчивость не мешает. Представьте, что вы открыли пару десятков окон, много что в них наработали, а система зависла. В *Linux* такого не бывает.

- А если вы научный работник и пишете статьи в международные журналы? К вашим услугам в *Linux* издательская система **TeX**, в формате которой принимаются статьи в эти журналы. Статью можете отправить по электронной почте прямо со своей Linux-машины. Пока вы писали статью, машина делала длительный сложный расчет и вам это совершенно не мешало. И массивы в вашей программе вовсе не ограничены пределом в 640 КВ. Хотите массив в 50 мегабайт? Пожалуйста, даже если у вас только 8 МВ RAM, так как в Linux работает виртуальная память. Как вы думаете, приятно ли столкнуться с зависанием ОС, когда до окончания трехчасового расчета осталось три минуты? В Linux это вам не грозит.
- Может, вам нужно синтезировать рекламную картинку профессионального качества? Пока система рендеринга будет это делать, можете поиграть в DOOM. Или отлаживать описание другой картинке. Рендеринг сложной картинке идет часами. Устойчивость ОС и тут не помешает.

# Характерные особенности *Linux* как ОС.

- **многозадачность** : много программ выполняются одновременно;
- **многопользовательский режим** : много пользователей одновременно работают на одной и той же машине;
- **защищенный режим процессора** (386 protected mode);
- **защита памяти процесса** : сбой программы не может вызвать зависания системы;
- **экономная загрузка** : Linux считывает с диска только те части программы, которые действительно используются для выполнения;
- разделение страниц по записи между экземплярами выполняемой программы. Это значит, что процессы-экземпляры программы могут использовать при выполнении одну и ту же память. Когда такой процесс пытается произвести запись в память, то 4-х килобайтная страница, в которую идет запись, копируется на свободное место. Это свойство увеличивает быстродействие и экономит память;



- **виртуальная память со страничной организацией** (т.е. на диск из памяти вытесняется не весь неактивный процесс, а только требуемая страница); виртуальная память в самостоятельных разделах диска и/или файлах файловой системы; объем виртуальной памяти до 2 Гбайт; изменение размера виртуальной памяти во время выполнения программ;
- **общая память программ и дискового кэша**: вся свободная память используется для буферизации обмена с диском;
- динамические загружаемые разделяемые библиотеки;
- **дамп программы для пост-мортем анализа**: позволяет анализировать отладчиком не только выполняющуюся, но и завершившуюся аварийно программу;
- **сертификация по стандарту POSIX.1**, совместимость со стандартами **System V** и **BSD** на уровне исходных текстов;
- через iBCS2-согласованный эмулятор совместимость с **SCO**, **SVR3**, **SVR4** по загружаемым программам;

- **наличие исходного текста всех программ**, включая тексты ядра, драйверов, средств разработки и приложений. Эти тексты свободно распространяются. В настоящее время некоторыми фирмами для Linux поставляется ряд коммерческих программ без исходных текстов, но все, что было свободным так и остается свободным;
- **управление заданиями** в стандарте POSIX;
- **эмуляция сопроцессора в ядре**, поэтому приложение может не заботиться об эмуляции сопроцессора. Конечно, если сопроцессор в наличии, то он и используется;
- **поддержка национальных алфавитов и соглашений**, в т.ч. для русского языка; возможность добавлять новые;
- **множественные виртуальные консоли**: на одном дисплее несколько одновременных независимых сеансов работы, переключаемых с клавиатуры;
- **поддержка ряда распространенных файловых систем (MINIX, Xenix, файловые системы System V)**; наличие собственной передовой файловой системы объемом до 4 Терабайт и с именами файлов до 255 знаков;