

# Файловая система. Функции ОС по управлению файлами

---



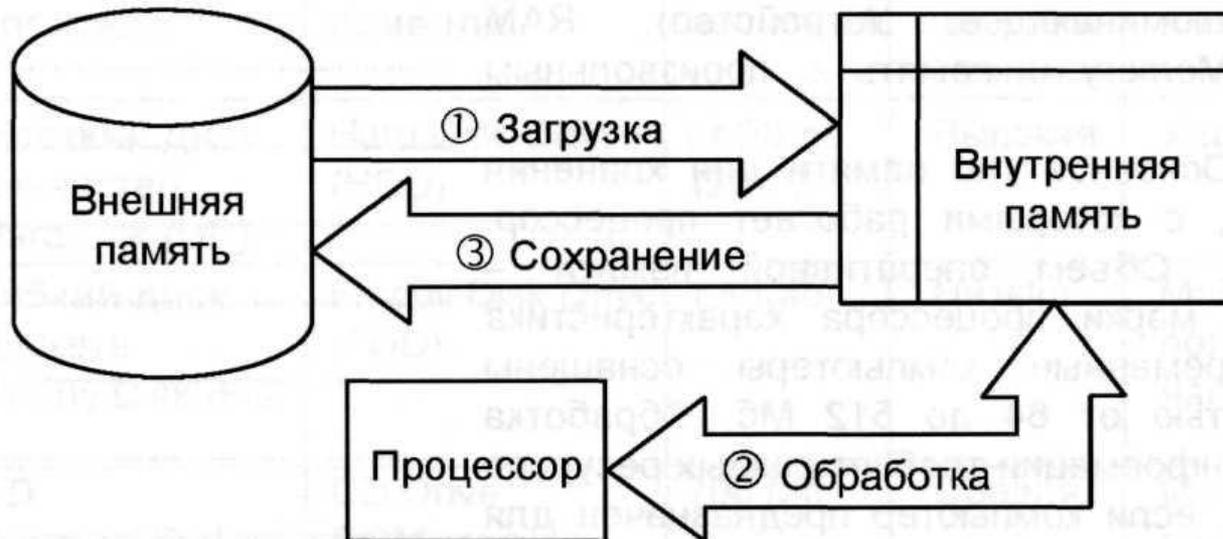
# План.

---

1. Файловая система. Функции файловой системы.
2. Понятие файла. Имя файла.
3. Типы файлов.
4. Обычные файлы.
5. Атрибуты файлов.
6. Организация файлов и доступ к ним
7. Операции над файлами.
8. Директории. Операции над директориями.
9. Древовидная структура файловой системы.
10. Защита файлов.
11. Заключение.

# Поддержка файловых систем

- Внешняя память вычислительной системы представляет собой **периферийные устройства**, на которых хранится большая часть пользовательской информации и системных данных.



# Поддержка файловых систем

- Для организации хранения информации на внешних носителях используется **файловая модель**.
- Принципы размещения файлов, каталогов и системной информации на реальном устройстве описываются **физической организацией файловой системы**



Общая модель файловой системы

# Поддержка файловых систем

---

- Для обеспечения доступа к данным используется специальный программный слой, обеспечивающий поддержку работы с конкретной файловой системой – **драйверы файловой системы**.

**Драйвер файловой системы** это программный компонент, интерпретирующий структуры файловой системы и предоставляющий логический иерархический вид использующим его приложениям. Он может являться частью операционной системы либо поставляться сторонним производителем как отдельный модуль

# Поддержка файловых систем

---

- Для обеспечения возможности работы с несколькими файловыми системами применяется подход, основанный применении **специального слоя**, с которым взаимодействуют приложения ОС.
  - Например, слой VFS (**virtual file system**) в некоторых версиях UNIX.
  - Целью VFS является обеспечение единообразного доступа клиентских приложений к различным типам файловых систем.



## Система управления данными (СУД)

---

- Система управления данными играет центральную роль в ОС.
- СУД должна управлять и хранить информацию и программы, принадлежащие как пользователям, так и самой системе.

# История СУД

---

- История систем управления данными во внешней памяти начинается с магнитных лент.
- Историческим шагом стал **переход к использованию централизованных систем управления файлами.**





# Система управления файлами обеспечивает (функции СУД)

---

- распределение внешней памяти,
- отображение имен файлов в адреса внешней памяти,
- обеспечение доступа к данным

# Файловая система - это

---

Файловая система - это часть операционной системы.  
Назначение:

1. Организация эффективной работы с данными, хранящимися во внешней памяти,
2. Обеспечение удобного интерфейса при работе с такими данными.

# В широком смысле ФС – это...

---

- совокупность всех файлов на диске;
- наборы структур данных, используемых для управления файлами, такие, например, как каталоги файлов, дескрипторы файлов, таблицы распределения свободного и занятого пространства на диске;
- комплекс системных программных средств, реализующих управление файлами, в частности: создание, уничтожение, чтение, запись, именованное, поиск и другие операции над файлами.



---

## **Основные функции файловой системы:**

- 1. Идентификация файлов** – связывание имени файла с выделенным ему пространством внешней памяти.



---

## **Основные функции файловой системы:**

### **2. Распределение внешней памяти между файлами**

- Для работы с конкретным файлом пользователю не требуется иметь информацию о местоположении этого файла на внешнем носителе информации.



---

## Основные функции файловой системы:

### 3. Обеспечение надежности и отказоустойчивости.

Стоимость информации может во много раз превышать стоимость компьютера.

- **Надежность** – работа системы без сбоев в определенных условиях в течение определенного времени.
- **Отказоустойчивость** — свойство технической системы сохранять свою работоспособность после отказа одного или нескольких составных компонентов.



---

## Основные функции файловой системы:

4. Обеспечение защиты от несанкционированного доступа и управление правами доступа.

**Несанкционированный доступ** — доступ к информации в нарушение должностных полномочий сотрудника, доступ к закрытой для публичного доступа информации со стороны лиц, не имеющих разрешения на доступ к этой информации.



---

## **Основные функции файловой системы:**

5. Обеспечение **совместного доступа к файлам**, так чтобы пользователю не приходилось прилагать специальных усилий по обеспечению синхронизации доступа.



---

## **Основные функции файловой системы:**

6. Обеспечение высокой производительности.



## **Основные функции файловой системы:**

---

7. Организация логической структуры файлов и доступа к файлам.

# Логическая организация файловой системы

---

- **Одной из основных задач ОС – предоставление удобного пользовательского интерфейса при работе с данными, хранящимися на носителях.**
- **Логическая модель в рамках ОС подменяет физическую модель размещения данных на носителях.**

## Основные функции файловой системы:

---

8. Управление каталогами файлов и манипуляций с файлами:
- создание файла (определение имени, выделение места во внешней памяти);
  - удаление файла (освобождение имени и занимаемого файлами пространства);
  - открытие файла (объявление ОС о намерении использовать файл с определенными функциями доступа и правами);
  - закрытие открытого файла (запрещение всякого доступа к файлу);
  - копирование и переименование файла.



## Файл – это

---

**Файл** – это именованная область внешней памяти, в которую можно записывать и из которой можно считывать данные.

С точки зрения пользователя, файл - единица внешней памяти, то есть данные, записанные на диск, должны быть в составе какого-нибудь файла.



# Имена файлов

---

- Файлы идентифицируются именами

# Типы файлов

---

- **Обычные файлы** – содержат пользовательскую информацию;
- **Каталоги** – системные файлы, поддерживающие структуру файловой системы;
- **Специальные файлы** – данных не хранят, обеспечивают механизм отображения физических внешних устройств в имена файлов файловой системы. Используются для унификации доступа к файлам и внешним устройствам;



# Обычные файлы

---

Обычные (или регулярные) файлы реально представляют собой набор блоков (возможно, пустой) на устройстве внешней памяти, на котором поддерживается файловая система.

Такие файлы могут содержать как **текстовую** информацию (обычно в формате ASCII), так и произвольную двоичную (**бинарную**) информацию.



# Обычные файлы делятся

---

- Текстовые
- Двоичные

# Текстовые файлы

---

- содержат символьные строки, которые можно распечатать, увидеть на экране или редактировать обычным текстовым редактором.
- Текстовые файлы состоят из строк символов, представленных в ASCII-коде. Это могут быть документы, исходные тексты программ и т.п. Текстовые файлы можно прочитать на экране и распечатать на принтере.

# Двоичные файлы

---

- Не используют ASCII-коды.
- Часто имеют сложную внутреннюю структуру, например, объектный код программы или архивный файл.
- Все операционные системы должны уметь распознавать хотя бы один тип файлов - их собственные исполняемые файлы.

- 
- Обычно прикладные программы, работающие с файлами, распознают тип файла по его имени в соответствии с общепринятыми соглашениями.
  - Например, файлы с расширениями .c, .pas, .txt - ASCII-файлы, файлы с расширениями .exe - выполнимые, файлы с расширениями .obj, .zip - бинарные и т. д.

# Специальные файлы

---

- это файлы, ассоциированные с устройствами ввода-вывода, которые позволяют пользователю выполнять операции ввода-вывода, используя обычные команды записи в файл или чтения из файла.
- Специальные файлы также, как и устройства ввода-вывода, делятся на блок-ориентированные и на байт-ориентированные.

# Атрибуты файлов

---

*Атрибут – информация, описывающая некоторые свойства файла*

- Список атрибутов в разных ОС может варьироваться. Обычно он содержит следующие элементы:
  - основную информацию (имя, тип файла),
  - адресную информацию (устройство, начальный адрес, размер),
  - информацию об управлении доступом (владелец, допустимые операции),
  - информацию об использовании (даты создания, последнего чтения, модификации и др.).
- Список атрибутов обычно хранится в структуре директорий или других структурах, обеспечивающих доступ к данным файла.

# Атрибуты файла

---

Атрибут – информация, описывающая некоторые свойства файла, например:

- Тип файла
- Владелец файла
- Создатель файла
- Пароль для доступа к файлу
- Информация о разрешенных операциях к файлу
- Время создания, последнего доступа и модификации файла
- Признак «только для чтения»
- Признак «скрытый файл»
- Признак «системный файл»
- Признак «двоичный/символьный файл»
- Признак «временный файл»
- Признак блокировки
- Длина записи в файле
- Др.

## Имена файлов

---

Правила именования файлов зависят от ОС.

Многие ОС поддерживают имена из **двух частей** (имя+расширение), например:

- `progr.c`
- `autoexec.bat`

Тип **расширения** файла позволяет ОС организовать работу с ним различных прикладных программ в соответствии с заранее оговоренными соглашениями.



## Имена файлов

---

### Имена файлов

Обычно ОС накладывают некоторые ограничения, как на используемые в имени символы, так и на длину имени файла.

В соответствии со стандартом POSIX, популярные ОС оперируют удобными для пользователя длинными именами (до 255 символов).

# Имена файлов

---

- Каждый файл имеет некоторое символьное имя. В иерархических системах выделяют три типа имен файлов:
  - **Простое** (имя файла в пределах одного каталога)
  - **Полное** (цепочка простых символьных имен всех каталогов, через которые проходит путь от корня до файла)
  - **Относительное** (имя включает имена каталогов, через которые проходит маршрут от текущего каталога к искомому файлу).
- В различных операционных системах есть свои ограничения на использование символов при присвоении имени, а также на длину относительного и полного имени файла.

# Полное имя файла

---

Для того чтобы однозначно определить файл по его имени (избежать коллизии имен), принято именовать файл так называемым **абсолютным или полным именем (pathname)**, состоящим из списка имен вложенных каталогов, по которому можно найти путь от корня к файлу плюс имя файла в каталоге, непосредственно содержащем данный файл.

# Пример

---

- Полное имя включает цепочку имен - путь к файлу, например **/usr/games/doom**. Такие имена уникальны.
- Компоненты пути разделяют различными символами: "/" (слэш) в Unix или обратными слэшем "\" в Windows (в Multics - ">"). Таким образом, использование древовидных каталогов минимизирует сложность назначения уникальных имен.

# Относительное имя файла

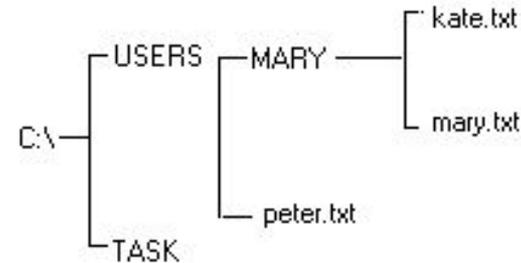
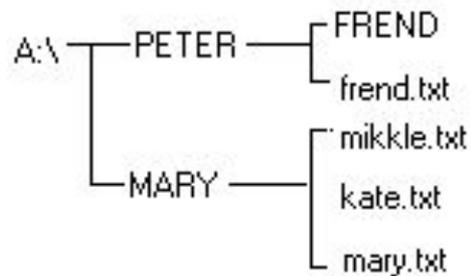
---

- Указывать полное имя не всегда удобно, поэтому применяют другой способ задания имени - **относительный** путь к файлу.
- Он использует концепцию рабочей или текущей директории, которая обычно входит в состав атрибутов процесса, работающего с данным файлом.
- Тогда на файлы в такой директории можно ссылаться только по имени, при этом поиск файла будет осуществляться в рабочем каталоге.

# Задания для выполнения

Заглавными буквами обозначены имена каталогов, строчными – имена файлов:

## I вариант      II вариант



Напишите для файла kate.txt

- ✓ Абсолютное имя файла
- ✓ Простое имя файла
- ✓ Относительное имя файла (относительно каталога MARY)

Напишите для файла mary.txt

- ✓ Абсолютное имя файла
- ✓ Простое имя файла
- ✓ Относительное имя файла (относительно каталога MARY)

# Логический и физический файл

---

- *Логический файл* - это организованная совокупность однородных элементов информации или однотипных записей для хранения информации в ИВС, которой присвоено имя.
- *Физический файл* - это область внешней памяти, хранящей информацию, и дескриптор файла, используемый для доступа к файлу и хранящий данные о физическом размещении и характеристиках файла.



# Каталоги

---

- Для решения проблем поиска и размещения файлов в СУФ используются иерархические, многоуровневые **каталоги** файлов.

# Каталог - это

---

- с одной стороны, группа файлов, объединенных пользователем исходя из некоторых соображений (например, файлы, содержащие программы игр, или файлы, составляющие один программный пакет),
- с другой стороны - это файл, содержащий системную информацию о группе файлов, его составляющих. В каталоге содержится список файлов, входящих в него, и устанавливается соответствие между файлами и их характеристиками (атрибутами).

# Каталоги

---

Все современные файловые системы поддерживают многоуровневое именование файлов за счет наличия во внешней памяти дополнительных файлов со специальной структурой - каталогов (или директорий).

Имя файла (каталога)	Тип файла (обычный или каталог)	
Anti	К	атрибуты
Games	К	атрибуты
Autoexec.bat	О	атрибуты
mouse.com	О	атрибуты

Каждый каталог содержит список каталогов и/или файлов, содержащихся в данном каталоге. Каталоги имеют один и тот же внутренний формат, где каждому файлу соответствует одна запись в файле директории

# Дерево каталогов

---

- Каталоги образуют
  - **дерево**, если файлу разрешено входить только в один каталог,
  - **сеть** - если файл может входить сразу в несколько каталогов.

## Директории

---

- Число директорий зависит от системы.
- В ранних ОС имелась только одна корневая директория, затем появились директории для пользователей (по одной директории на пользователя).
- В современных ОС используется произвольная структура дерева директорий.

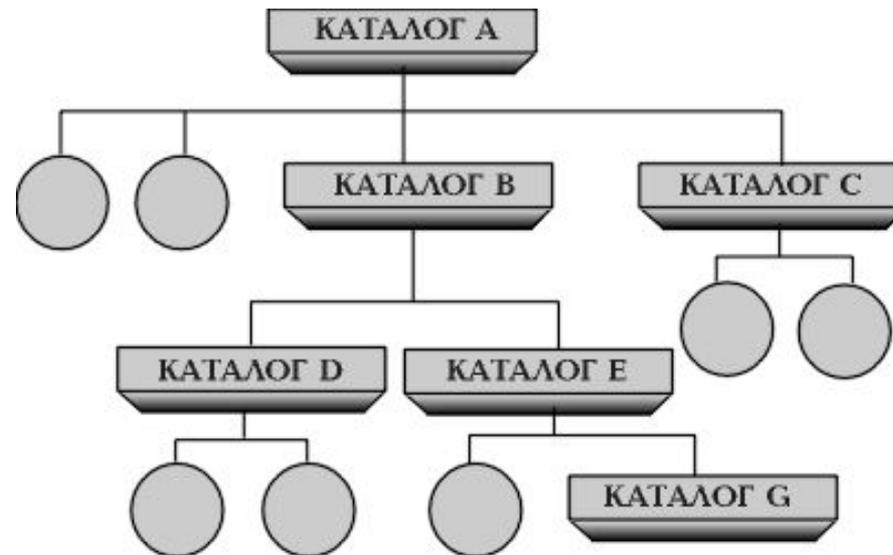
# Простой одноуровневый каталог

---



## Иерархический, многоуровневый каталог (древовидный или сетевой)

- это совокупность каталогов и дескрипторов файлов различной глубины

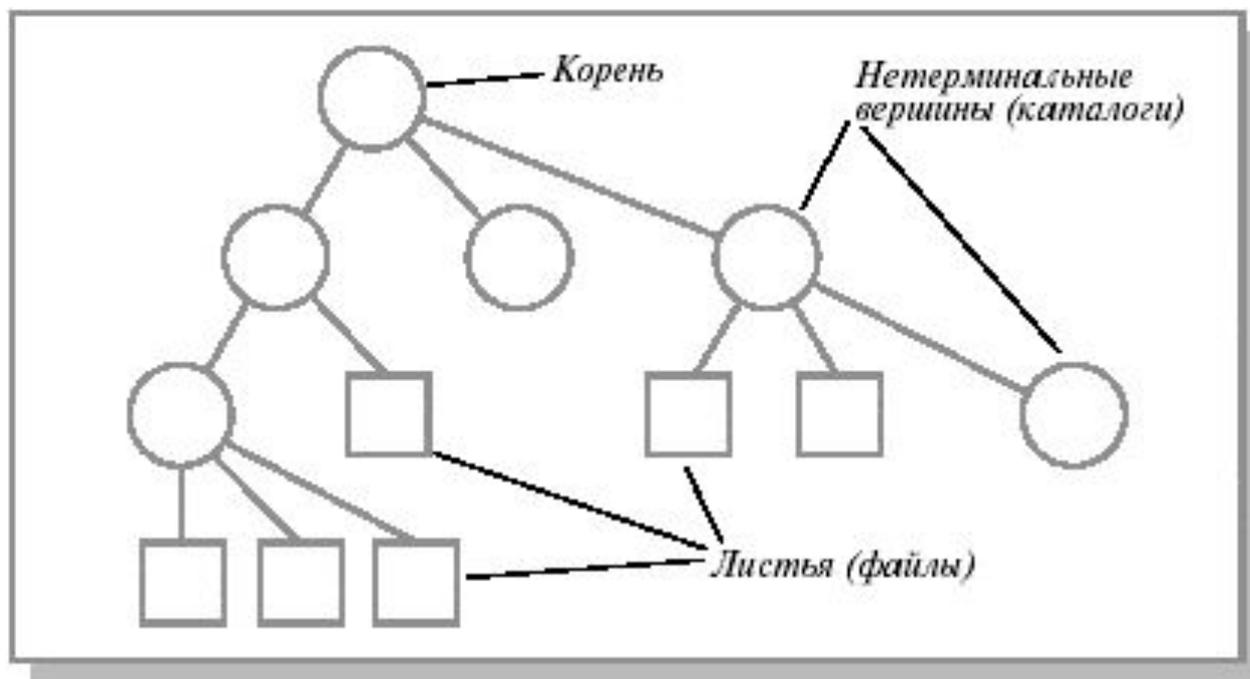


# Иерархическая структура ФС

---

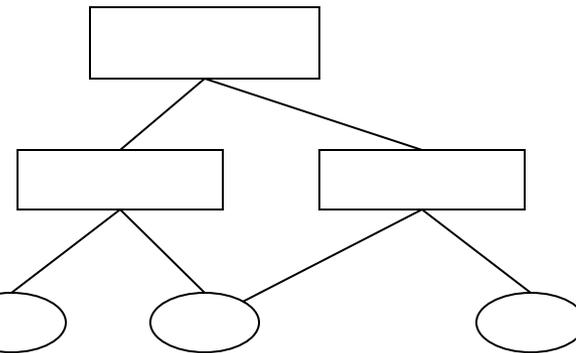
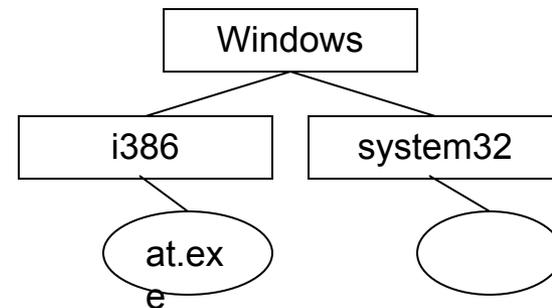
- Древовидные структуры являются графами, не имеющими циклов.
- Можно считать, что ребра графа направлены вниз, а корень - вершина, не имеющая входящих ребер.
- Внутри одного каталога имена листовых файлов уникальны.

# Древовидная структура файловой системы



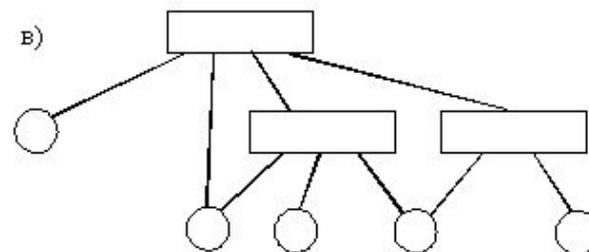
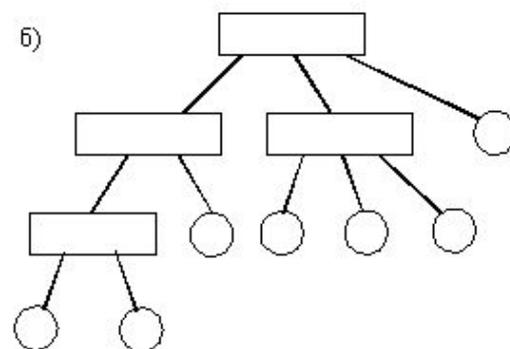
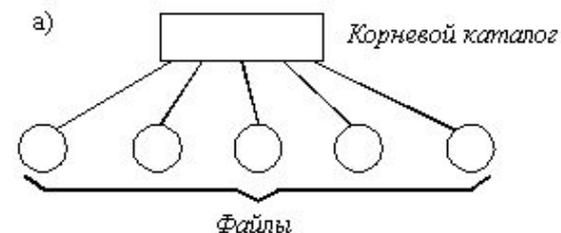
# Иерархическая структура файловой системы

- Пользователи обращаются к файлам по их символическим именам. Для удобства пользователя логическая структура хранения данных представляет иерархическую структуру.
- Граф, описывающий структуру файловой системы может представлять собой **дерево** или **сеть**.
- В Windows используется древовидная организация, в UNIX – сетевая.

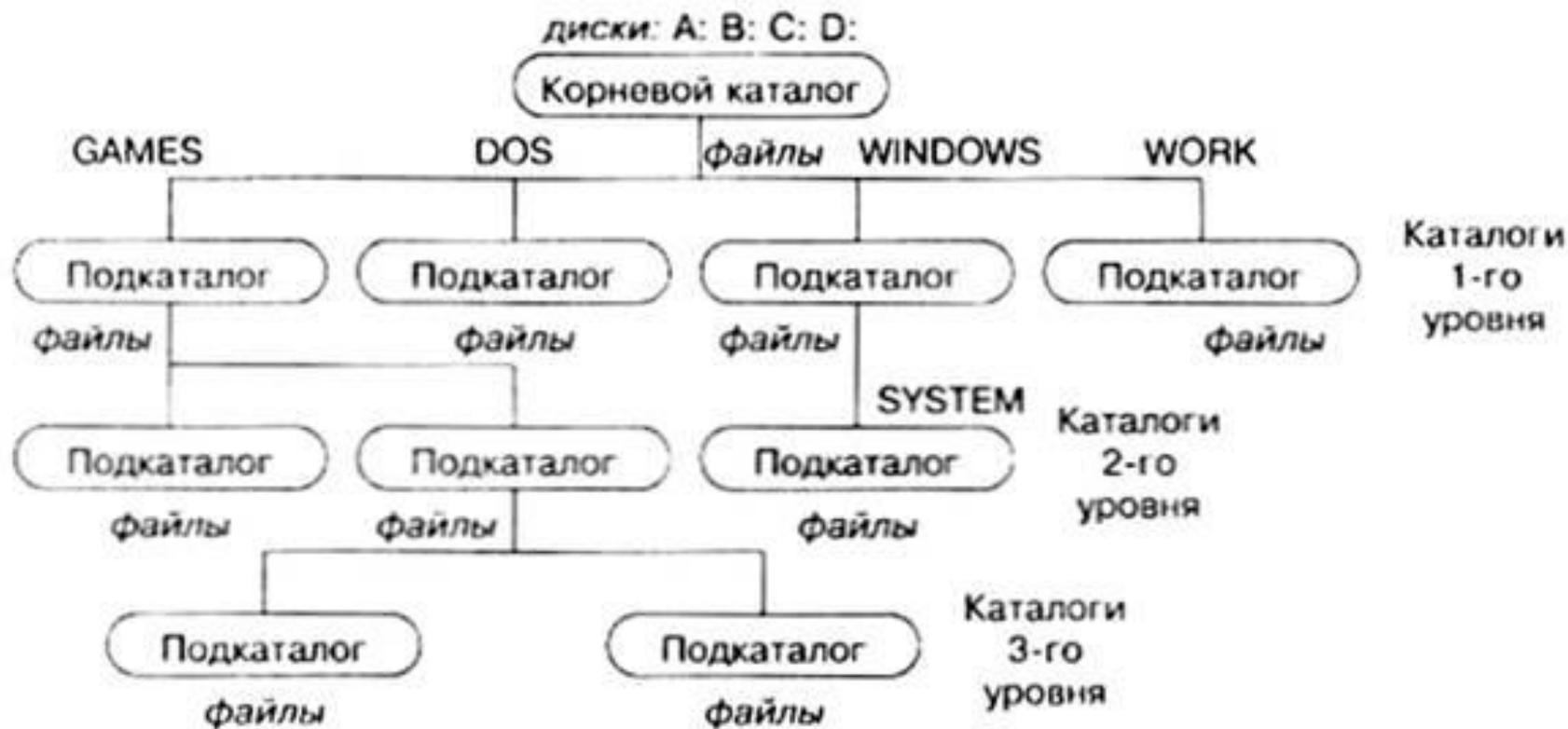


# Логическая организация файловой системы

- а - одноуровневая;
- б - иерархическая (дерево);
- в - иерархическая (сеть)



## Дерево каталогов



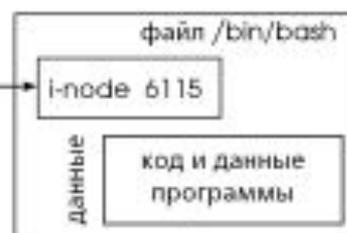
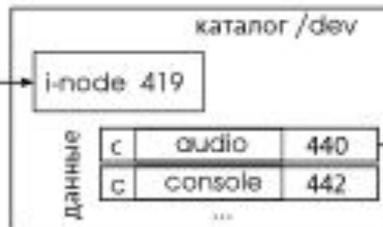
# Файловая система UNIX

суперблок:

Заголовок, версия	Число монтирований	Размер блока	Число свободных блоков и i-nodes	Первый i-node
-------------------	--------------------	--------------	----------------------------------	---------------



иерархия каталогов с общим корнем



файловая система состоит из i-nodes (индексных узлов)

виды файлов:

- файлы с данными
- каталоги
- файлы устройств
- символичные ссылки
- именованные каналы
- сокеты

## Пример жесткой ссылки

Имена файлов зависят от регистра, файлы `test.txt` и `TEST.txt` отличаются (т. е. могут находиться в одном каталоге и быть при этом разными файлами).



- Таким образом, каждый файл характеризуется одним i-node, но может быть связан с несколькими именами — в UNIX это называют *жесткими ссылками*.
- При такой схеме физическое удаление данных файла с носителя происходит только тогда, когда удаляется последняя из существующих жестких ссылок на этот файл.

# ДЗ

---

- **Виды файлов в ОС семейства Linux:**
  - Специальный файл устройства
  - Именованный канал
  - Символьная ссылка
  - Сокет

# Монтирование файловой системы

---

- В общем случае вычислительная система может иметь несколько устройств внешней памяти. Для обеспечения доступа к данным, хранящимся на разных носителях используются два подхода:
  - На каждом устройстве размещается **автономная файловая система**, со своим деревом каталогов (например, в MS Windows накопители нумеруются a:, c: и т.д.).
  - **Монтирование файловой системы** – операция объединения файловых систем в единую файловую систему (например, в операционных системах семейства UNIX).

# Монтирование файловых систем

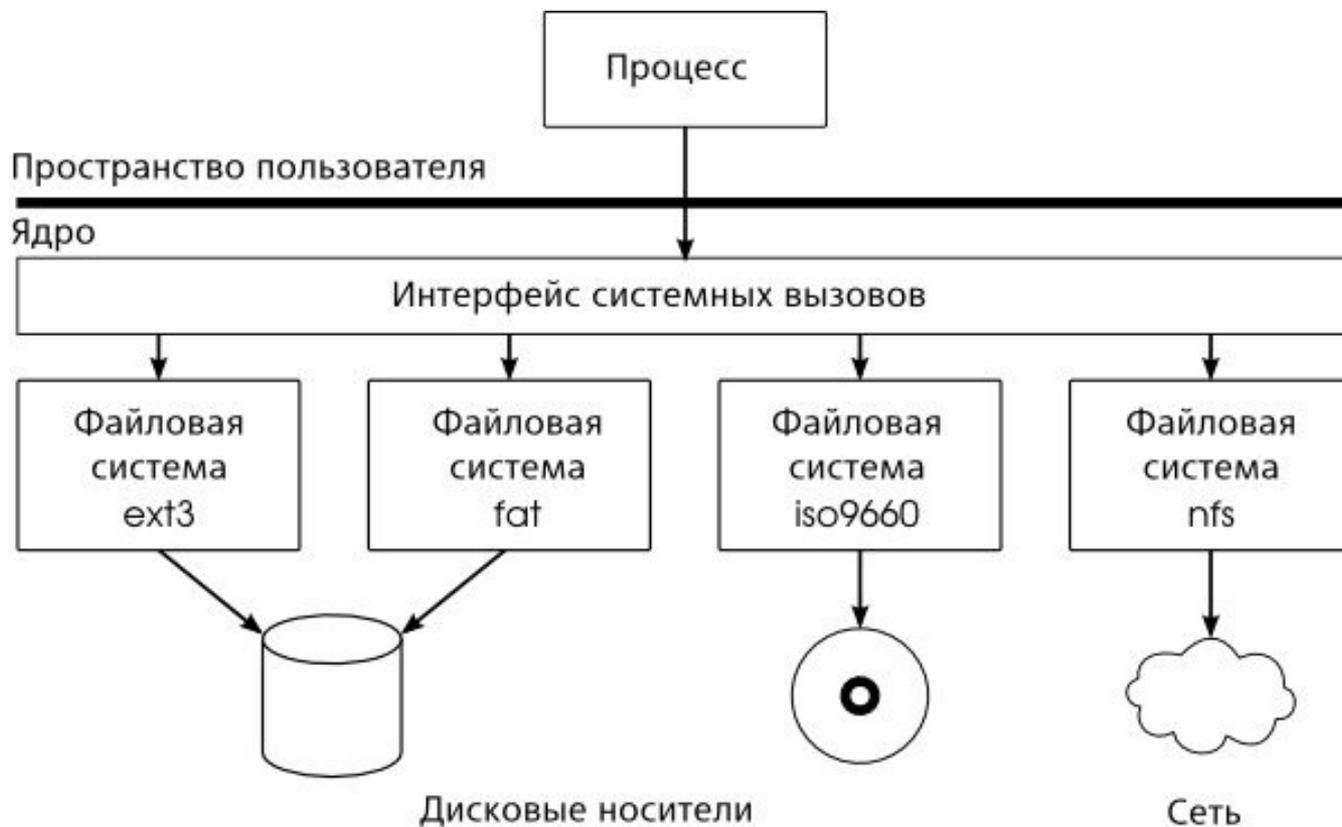
В операционной системе UNIX файловые системы присоединяются к единому дереву с общим корнем:



Имя файла в исходной файловой системе: `/dir/b.txt`

Имя файла после монтирования: `/media/floppy/dir/b.txt`

# Виртуальная файловая система



## Переключатель файловых систем или виртуальная файловая система (VFS)

---

- VFS предоставляет пользователю стандартный набор функций (интерфейс) для работы с файлами, вне зависимости от места их расположения и принадлежности к разным файловым системам.
- Пользовательская программа имеет стандартный набор системных вызовов (например, *открыть файл, прочитать данные, удалить файл*), которые операционная система транслирует соответствующей файловой системе.

**Защита файлов**      Информация в компьютерной системе должна быть защищена как от физического **разрушения** (reliability), так и от несанкционированного **доступа** (protection).

---

### **Контроль доступа к файлам**

- ✓ Наличие в системе многих пользователей предполагает организацию контролируемого доступа к файлам.
- ✓ Выполнение любой операции над файлом должно быть разрешено только в случае наличия у пользователя соответствующих привилегий.
- ✓ Обычно контролируются следующие операции:
  - ✓ чтение, запись и выполнение.
  - ✓ другие операции, например копирование файлов или их переименование, также могут контролироваться. Однако они чаще реализуются через перечисленные. Так, операцию копирования файлов можно представить как операцию чтения и последующую операцию записи.

## Списки прав доступа (access control list)

---

- ✓ Любой запрос на выполнение операции сверяется с ACL-таблицей.
- ✓ Чтобы разрешить всем пользователям читать файл, необходимо всех их внести в список.
- ✓ Классификации пользователей, например, в ОС Unix все пользователи разделены на три группы.
  - ✓ **Владелец (Owner).**
  - ✓ **Группа (Group).** Набор пользователей, разделяющих файл и нуждающихся в типовом способе доступа к нему.
  - ✓ **Остальные (Univers).**



# Операции над директориями

---

1. Создание директории.
2. Удаление директории.
3. Открытие директории для последующего чтения
4. Закрытие директории после ее чтения для освобождения места во внутренних системных таблицах.
5. Поиск.
6. Получение списка файлов в каталоге.
7. Переименование.



# Операции над файлами

---

- Создание файлов
- Переименование файла
- Перемещение и копирование файла
- Удаление файлов
- Поиск файлов
- Архивирование файлов



# Команды для работы в терминале Linux

---

- Система
- Каталоги
- Файлы

# Метасимволы имён файлов

---

- \* - звёздочка - заменяет любое количество символов, например `ls *.txt`
- ? - знак вопроса – заменяет любой конкретный единичный символ `ls mc6????`



## **Последовательное выполнение команд**

---

- через ; в одной строке – команде

# Условное выполнение команд

---

- **and** – если выполнение первой команды прошло успешно, то переход к другой команде  
Структура команды: **cd /user/bin&&ls** – эта команда указывает перейти в каталог user/bin и показать в нём файлы и каталоги.
- **or** – даже если выполнение первой команды не прошло успешно, то должен произойти переход к другой команде, если первая команда правильная, то вторая игнорируется.

# Использование символов перенаправления

---

- Символ  $>$  для перенаправления стандартного вывода.
- Символ  $<$  для перенаправления стандартного ввода.
- Символ  $>>$  для добавления стандартного вывода в конец существующего файла.
- Символ  $|$  (символ потока) для передачи стандартного вывода одной команды на стандартный вход другой.

# Работа с файлами

---

<b>file</b>	Показывает тип данных указанных файлов
<b>mv</b>	Изменение имени файла
<b>cat</b>	Показывает содержимое файла
<b>more</b>	Постранично показывает содержимое файла
<b>head</b>	Выводит первые десять строк указанного файлы
<b>cp</b>	Копирует файлы
<b>vi</b>	Вызывает текстовый редактор vi

# Работа с файлами

<b>whereis</b>	Поиск файлов
<b>find</b>	Поиск указанных файлов в каталогах по иерархии не выше указанного
<b>diff</b>	Сравнивает два файла и приводит доклад о различиях между этими файлами
<b>gunzip</b>	Сжимает и распаковывает файлы (расширение .gz)
<b>rm</b>	Удаляет указанный файл
<b>sort</b>	Построчно сортирует содержимое текстового файла
<b>tail</b>	Показывает последние 10 (по умолчанию) строк указанного файла (-50)
<b>wc</b>	подсчитывает количество строк, слов, байт, или символов в текстовом файле

## Создание файла (способы)

---

- **echo** "Текст записывается в файл test">test.txt
- **cat** *имя\_файла* >>*имя\_файла*
- **vi** **имя\_файла** – открытие редактора для редактирования файла
- **ed** *имя\_файла* – добавление в файл (*a*- добавить, *w*-сохранить, *q*-выйти).

# Изменение атрибутов файла

---

## □ Команда **chmod**

Опция команды chmod:	u	g	o	a	Доступ	+r (-r)	+x (-x)	+w (-w)
Уровень	Владелец	Группа	Прочие	Все	Описание	Чтение	Выполнение	Запись

- `chmod -r o-w /shared/docs`
- `chmod -r og-w,o-x private`

# Работа с каталогами

---

<b>cd</b>	Перейти в каталог
<b>cd ..</b>	Перейти в надкаталог
<b>mkdir</b>	Создаёт каталог с указанным именем
<b>rmdir</b>	Удаляет пустой указанный каталог
<b>pwd</b>	Показывает имя текущего каталога
<b>ls</b>	Отображает список содержимого каталога
<b>dir</b>	Показывает содержимое каталогов

# Система

<b>last</b>	Показывает дату и время последнего входа в систему для указанного пользователя
<b>printenv</b>	Выдаёт все текущие системные переменные
<b>df</b>	Показывает свободное место на диске
<b>du</b>	Предоставляет информацию о занятом дисковом пространстве
<b>arch</b>	Даёт описание архитектуры вашей машины
<b>who</b>	Показывает статистику по пользователям, зарегистрированным в системе в данный момент
<b>whoami</b>	Показывает идентификатор пользователя
<b>uname</b>	информация о типе ОС (если ввести команду <b>uname -a</b> , то будет показано тип ОС, хостнэйм, версия ОС, издание ОС, архитектуру компьютера, платформу).
<b>w</b>	машинное имя каждого пользователя, его терминал, время входа и последнюю введённую пользователем команду
<b>dd</b>	копирует файл, преобразует и форматирует в зависимости от ключей
<b>file</b>	определение типа файла

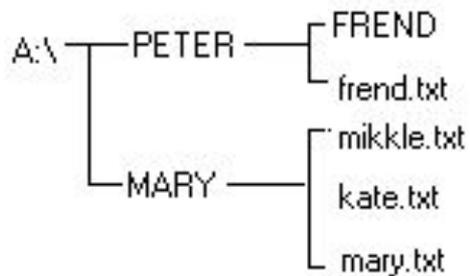
## Справка

---

<b>call</b>	Показывает календарь
<b>date</b>	Выдаёт текущую дату и время
<b>echo</b>	Отображает указанный текст
<b>clear</b>	Очищает экран
<b>whatis</b>	Показывает краткое описание указанной команды
<b>man</b>	Вызывает интерфейс справочной службы, например <code>man ls</code>
<b>ps</b>	Выводит информацию о текущих процессах

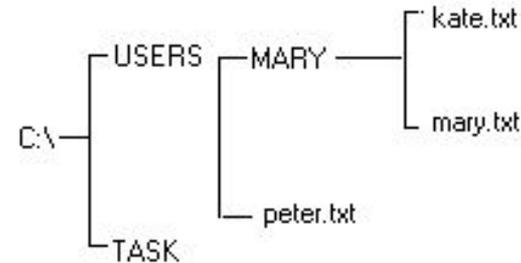
# Задания для выполнения

Напишите систему команд для создания файлов и каталогов, согласно файловой структуре. Заглавными буквами обозначены имена каталогов, строчными – имена файлов:



Напишите для файла kate.txt

- ✓ Абсолютное имя файла
- ✓ Простое имя файла
- ✓ Относительное имя файла (относительно каталога MARY)



Напишите для файла mary.txt

- ✓ Абсолютное имя файла
- ✓ Простое имя файла
- ✓ Относительное имя файла (относительно каталога MARY)

**I вариант    II вариант**



# Задания для выполнения

---

**Отобразить файловую структуру в виде дерева.**

Указаны пути от корневого каталога к некоторым файлам, хранящимся на магнитном диске. Заглавными буквами обозначены имена каталогов, строчными – имена файлов

**I вариант    II вариант**

**\COUNTRY\USA\INFO\culture.txt;  
\COUNTRY\USA\Washington.txt;  
\COUNTRY\RUSSIA\moskow.txt;  
\COUNTRY\RUSSIA\INFO\industry.txt  
\COUNTRY\RUSSIA\ INFO\culture.txt;**

**\BOX\LETTER\peter.txt;  
\BOX\LETTER\kate.txt;  
\LETTER\WORK\april.txt;  
\LETTER\WORK\mayl.txt;  
\LETTER\FREND\SCHOOL\mary.txt  
;  
\LETTER\FREND\sport.txt;**



## Организация файлов и доступ к ним

---

- Программист воспринимает файл в виде набора однородных записей.
- Запись - это наименьший элемент данных, который может быть обработан как единое целое прикладной программой при обмене с внешним устройством.



## ОС поддерживают несколько вариантов структуризации файлов

---

- **Последовательный файл, т.е.** есть файл является последовательностью записей.
- **Файл прямого доступа**

# Другие формы организации файлов

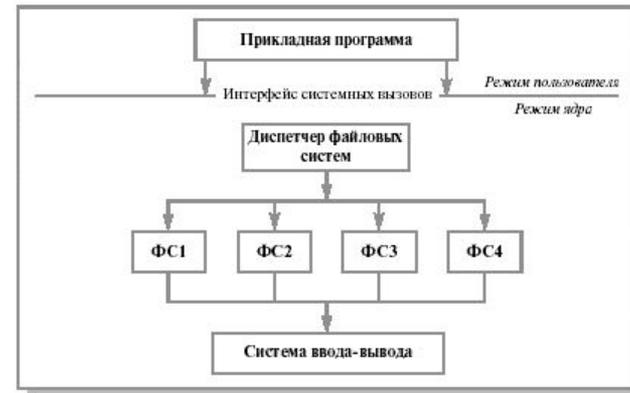
---

- хранение файла в виде **последовательности записей фиксированной длины**, каждая из которых имеет внутреннюю структуру
- **последовательность записей переменной длины** каждая из которых содержит ключевое поле в фиксированной позиции внутри записи. Такие файлы называются **индексированными**

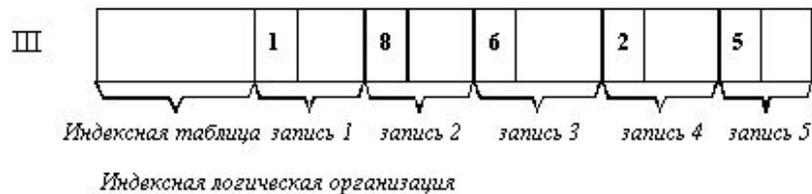
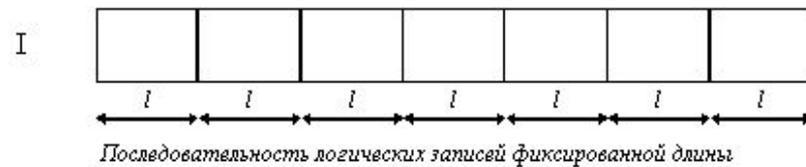


# Домашнее задание

## Архитектура современной ФС



# Способы логической организации файлов



Индекс	1	2	3	4	5	6
Адрес	21	201	315	661	670	715

*Индекс  $\equiv$  ключ*

# Заключение

---

- Файловая система представляет собой набор файлов, директорий и операций над ними.
- Имена, структуры файлов, способы доступа к ним и их атрибуты - важные аспекты организации файловой системы.
- Обычно файл представляет собой неструктурированную последовательность байтов.
- Главная задача файловой системы - связать символическое имя файла с данными на диске.
- Большинство современных ОС поддерживает иерархическую систему каталогов или директорий с возможным вложением директорий.
- Безопасность файловой системы, базирующаяся на ведении списков прав доступа, - одна из важнейших концепций ОС.



## Вопросы для повторения и самоконтроля:

---

1. Что такое файловая система?
2. Каковы функции ФС?
3. Какие ФС знаете вы?
4. Что такое файл, виды файлов. Приведите примеры. Функции над файлами.
5. Атрибуты файлов.
6. Типы файлов.
7. Каталог. Функции над каталогами.
8. Иерархическая структура файловой системы.

# Разделы диска

---

В современных ОС принято разбивать диски на **логические диски** (это низкоуровневая операция), иногда называемые **разделами** (partitions).

Каждый раздел представляет собой отдельный (виртуальный) диск.

Диск содержит иерархическую древовидную структуру, состоящую из набора файлов, каждый из которых является хранилищем данных пользователя, и каталогов или директорий (то есть файлов, которые содержат перечень других файлов, входящих в состав каталога), необходимых для хранения информации о файлах системы.