

# Логическая организация файловой системы

Выполнила студентка  
4 курса ФМФ  
Седина Евгения

- Для того чтобы предоставить пользователю удобный интерфейс при работе с данными, хранящимися на дисках, ОС подменяет физическую структуру хранящихся данных некоторой удобной для пользователя логической моделью.
- Структура данных материализуется в виде дерева каталогов, выводимого на экран утилитами Norton Commander или Windows Explorer.

# **Основные цели использования файла:**

- 1. Долговременное и надежное хранение информации.**
- 2. Совместное использование информации.**
- 3. Эти цели реализуются в ОС файловой системой.**

## Основные функции ФС нацелены на решение следующих задач:

- именованние файлов;
- программный интерфейс для приложений;
- отображения логической модели файловой системы на физическую организацию хранилища данных;
- устойчивость файловой системы к сбоям питания, ошибкам аппаратных и программных средств.
- совместного доступа к файлу из нескольких процессов
- защита файлов одного пользователя от несанкционированного доступа другого пользователя


# Типы файлов

```
graph TD; A[Типы файлов] --> B[Обычные файлы]; A --> C[Каталоги]; A --> D[Специальные файлы];
```

Обычные  
файлы


Каталоги

Специальные  
файлы



**Обычные файлы**, или просто файлы, содержат информацию произвольного характера, которую заносит в них пользователь или которая образуется в результате работы системных и пользовательских программ.

Содержание обычного файла определяется приложением, которое с ним работает.



**Каталоги** — это особый тип файлов, которые содержат системную справочную информацию о наборе файлов, сгруппированных пользователями по какому-либо неформальному признаку.

Во многих операционных системах в каталог могут входить файлы любых типов, в том числе другие каталоги, за счет чего образуется древовидная структура, удобная для поиска.

Каталоги устанавливают соответствие между именами файлов и их характеристиками, используемыми файловой системой для управления файлами.

**Специальные файлы** — это фиктивные файлы, ассоциированные с устройствами ввода-вывода, которые используются для унификации механизма доступа к файлам и внешним устройствам.

Специальные файлы позволяют пользователю выполнять операции ввода-вывода посредством обычных команд записи в файл или чтения из файла.

Эти команды обрабатываются сначала программами ФС, а затем на некотором этапе выполнения запроса преобразуются ОС в команды управления каким либо устройством.



## Структура файловой системы

Большинство ФС имеет иерархическую структуру, в которой уровни создаются за счет того, что каталог более низкого уровня может входить в каталог более высокого уровня.

Каталоги образуют дерево, если файлу разрешено входить только в один каталог ( MS-DOS и Windows) и сеть — если файл может входить сразу в несколько каталогов (в UNIX ).

Каталог самого верхнего уровня называется корневым каталогом, или корнем (root).

Частным случаем иерархической структуры является одноуровневая организация, когда все файлы входят в один каталог.

## Имена файлов

Все типы файлов имеют символьные имена. В иерархически организованных файловых системах обычно используются три типа имен - файлов:

- простые,
- составные
- относительные.

**Простое, или короткое, символьное** имя идентифицирует файл в пределах одного каталога. Простые имена присваивают файлам пользователи и программисты, при этом они должны учитывать ограничения ОС как на номенклатуру символов, так и на длину имени.

*( в ФС s5, поддерживаемой многими версиями ОС UNIX, простое символьное имя не могло содержать более 14 символов).*

**Полное** имя представляет собой цепочку простых символьных имен всех каталогов, через которые проходит путь от корня до данного файла.

Полное имя является составным, в нем простые имена отделены друг

от друга принятым в ОС разделителем.

*Например, два файла имеют простое имя main.exe, но их составные имена /depart/main.exe и /user/anna/main.exe различаются.*

**Атрибуты файлов** — это информация, описывающая его свойства:

- тип файла (обычный файл, каталог, специальный файл и т. п.);
- владелец файла;
- создатель файла;
- пароль для доступа к файлу;
- информация о разрешенных операциях доступа к файлу;
- времена создания, последнего доступа и последнего изменения;
- текущий размер файла;
- максимальный размер файла;
- признак «только для чтения»;
- признак «скрытый файл»;
- признак «системный файл»;
- признак «архивный файл»;

Данные, содержащиеся в файле, имеют некую логическую структуру.

**Неструктурированная модель файла** позволяет легко организовать разделение файла между несколькими приложениями: разные приложения могут по-своему структурировать и интерпретировать данные, содержащиеся в файле (ОС UNIX, MS-DOS, Windows NT/2000, NetWare. ).

**В структурированной модели** поддержание структуры файла поручается файловой системе. ФС видит файл как упорядоченную последовательность логических записей.

ФС предоставляет приложению доступ к записи, а вся дальнейшая обработка данных, содержащихся в этой записи, выполняется приложением.

СУБД поддерживают как сложную структуру данных, так и взаимосвязи между ними.

**Логическая запись** является наименьшим элементом данных, которым может оперировать программист при организации обмена с внешним устройством.

Файловая система может использовать *два способа доступа к логическим записям:*

**последовательный доступ** - читать или записывать логические записи последовательно

**прямой доступ** - позиционировать файл на запись с указанным номером.

Файлы, доступ к записям которых осуществляется последовательно, по номерам позиций, называются **неиндексированными**, или **последовательными**.

**Индексированные файлы** допускают более быстрый прямой доступ к отдельной логической записи, которая имеет одно или более ключевых (индексных) полей и могут адресоваться путем указания значений этих полей.