

# **КОНСТРУИРОВАНИЕ ПРОГРАММ И ЯЗЫКИ ПРОГРАМИРОВАНИЯ**

**Лекция № 22.0 – Файловая система**

**Преподаватель: Поденок Леонид Петрович, 505а-5**

**+375 17 293 8039 (505а-5)**

**+375 17 320 7402 (ОИПИ НАНБ)**

**prep@lsi.bas-net.by**

**ftp://student:2ok\*uK2@Rwox@lsi.bas-net.by**

**Кафедра ЭВМ, 2021**

## Оглавление

Файловая система и файловый ввод-вывод.....	3
Общие понятия.....	3
Файловая система ОС *NIX.....	5
Ограничения на имя файла.....	6

# Файловая система и файловый ввод-вывод

## Общие понятия

**Файл (Именованный файл)** — некий набор хранимых на внешнем носителе данных, имеющий имя (набор данных — data set).

Это не строгое определение файла — оно лишь отражает те свойства, которые мы обычно используем на прикладном уровне.

**Термин «файловая система» имеет 2 значения:**

- 1) подсистема ОС, отвечающая за хранение информации на внешних запоминающих устройствах (ЗУ) в виде именованных файлов;
- 2) обозначение структур данных, создаваемых на внешнем ЗУ с целью организации хранения на этом устройстве данных в виде именованных файлов.

**Имя файла** — определяется алфавитом и длиной.

**Каталог (Directory)** — особый тип файла, хранящий имена файлов, некоторые из которых, возможно, сами являются каталогами.

При создании на диске файловой системы создается один каталог — **корневой каталог**.

При необходимости в корневом каталоге можно создавать другие каталоги, а их имена записывать в корневой — **каталоги первого уровня вложенности**.

В них, в свою очередь, тоже можно создавать каталоги.

При использовании каталогов говорят о **кратком** или **локальном** имени файла (строка символов, уникальная в пределах каталога) и о **полном имени файла** или **полном пути к файлу** (строка символов, включающая имена всех каталогов, начиная с корневого и заканчивая локальным).

Имена каталогов разделяются символом «/».

*(URI – Uniform Resource Identifier)*

В каждом каталоге существует файл со специальным именем, обозначающим каталог более высокого уровня (родительский). Его имя «..».

Еще одно специальное имя – файл, описывающий сам каталог. Его имя «.».

При работе с файлами один из каталогов тем или иным способом объявляется **текущим (current directory)**.

К файлам из текущего каталога можно обращаться по короткому имени.

Для обращения к файлам вне текущего каталога можно использовать **полный (абсолютный)** или **относительный** путь.

Относительный путь начинается с текущего каталога.

Операционные системы различают полные и относительные пути по наличию символа-разделителя «/» перед первым именем в пути.

/home/user/music/ACDC/Back in Black/Back in Black.flac  
music/ACDC/Back in Black/Back in Black.flac

# Файловая система ОС \*NIX

Единое дерево каталогов.

**В имя файла ни в каком виде не входит имя устройства, на котором файл находится.**

Если в системе присутствует несколько устройств прямого доступа (dasd), один из них объявляется **корневым (root)**, и все остальные «**монтируются (mount)**» в тот или иной каталог, называемый точкой **монтирования (mount point)**.

Для указания полного пути к файлу на примонтированном устройстве спереди к имени файла добавляется полный путь к точке монтирования. Например, если у нас есть дискета, смонтированная в `/media/myfloppy`, а на ней есть каталог `work`, а в нем файл `foo.c`, то полный путь к этому файлу будет `/media/myfloppy/work/foo.c`.

**В ОС \*NIX каталоги хранят только имя файла и некоторый номер, позволяющий идентифицировать соответствующий файл.**

Вся остальная информация о файле, как то размер, расположение на диске, даты создания, модификации и последнего обращения, данные о владельце и о правах доступа к нему связываются не с именем файла, а с этим самым номером.

**ИНДЕКСНЫЙ ДЕСКРИПТОР — хранимая на внешнем ЗУ (диске) структура данных, содержащая всю информацию о файле, исключая его имя.**

Допускается, чтобы несколько имен файлов, в том числе и в разных каталогах, ссылались на один и тот же номер индексного дескриптора.

Индексный дескриптор содержит:

- номер (уникальный в рамках файловой системы данного диска);
- тип файла;
- права доступа к файлу;
- количество связей (ссылок на файл в каталогах) файла;
- идентификатор пользователя и группы-владельца;
- размер файла в байтах;
- время последнего доступа к файлу;
- время последнего изменения файла;
- время последнего изменения индексного дескриптора файла;
- указатели на блоки данных файла (обычно 10);
- указатели на косвенные блоки (обычно 3).

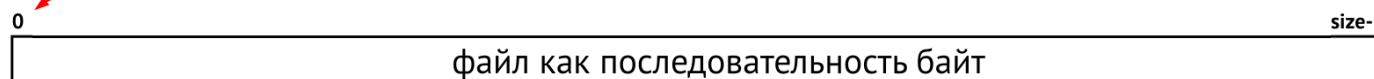
Имя файла inode #

.	2056
..	3306
foo.conf	3124
bar.cc	2345
<b>Debug</b>	6347
baz.asm	0239
<b>Release</b>	7371

Имя файла inode #

.	1725
..	3715
text.new	7247
foo.diff	8120
bar.conf	3124
project.i	4387
<b>Debug</b>	9721

inode #	type	size	c.date	a.date	m.date	...	block->
---------	------	------	--------	--------	--------	-----	---------



## Ограничения на имя файла

В имени файла допустимы любые символы ASCII за исключением разделителя каталогов «/» и нулевого символа '\0'.

Ограничения на длину имени существуют — обычно 255.

Современные ФС допускают использование UNICODE.

Стандарт POSIX (**P**ortable **O**perating **S**ystem **I**nterface) требует реализации поддержки двух типов связей имен с индексными дескрипторами

- **жестких;**
- **символических.**

**Жесткой ссылкой (hard link)** считается элемент каталога, указывающий непосредственно на некоторый индексный дескриптор.

Жесткие связи очень эффективны, но у них существуют определенные ограничения, так как они могут создаваться только в пределах одной физической файловой системы.

Когда создается такая связь, связываемый файл должен уже существовать.

Кроме того, каталоги не могут связываться жесткой связью.

**Фактически, файл может иметь несколько совершенно равноправных имен.**

Файл создается (**open( )**, **create( )**) всегда с одним именем. Дополнительные имена можно назначить файлу используя системный вызов **link( )**.

Этот вызов проецируется на команду оболочки **ln**.

В индексном дескрипторе содержится счетчик жестких ссылок на дескриптор.

Создание жестких ссылок на каталоги не допускается (на уровне ядра) из-за вероятности возникновения рекурсии и заикливания.

Создание жестких ссылок на файлы, расположенные на других устройствах, не допускается (уникальность индексных дескрипторов только в пределах устройства).

Файл удаляется функцией **unlink( )**, которая удаляет одну из жестких ссылок. Когда будет удалена последняя, файл считается удаленным и ресурсы, которые он занимает на носителе освобождаются.



## Типы файлов. Символические ссылки

**Каталоги** — это файлы специального типа. Они хранят имена и номера индексных дескрипторов. Все, чем отличается каталог от обычного файла на низком уровне — это значение признака типа в индексном дескрипторе.

В остальном хранение каталогов на диске не отличается от хранения обычных файлов.

Кроме обычных файлов и каталогов операционные системы поддерживают и другие специальные типы файлов. FAT (MS DOS, Windows), например, поддерживает тип «метка тома» (volume label).

UNIX поддерживает достаточно большое количество разновидностей файлов специального типа:

- файлы байт-ориентированных устройств;
- файлы блок-ориентированных устройств;
- имена сокетов;
- именованные каналы (FIFO);
- символические ссылки.

**Файловая система обеспечивает перенаправление запросов, адресованных периферийным устройствам, к соответствующим модулям подсистемы ввода-вывода.**

**Символическая ссылка (Symbolic link)** — файл специального типа, содержащий путь к другому файлу.

Указание на то, что данный элемент каталога является символической ссылкой, находится в индексном дескрипторе. *Обычные* команды доступа к файлу вместо получения данных из физического файла, берут их из файла, имя которого приведено в ссылке.

Этот путь может указывать на что угодно — это может быть каталог, он может даже находиться в другой физической файловой системе, более того, указанного файла может и вовсе не быть.

В описании пути можно использовать любое сочетание символических и жестких ссылок.

Операция открытия символической ссылки на чтение или запись приводит к открытию на чтение или запись того файла, на который она ссылается, а не ее самой.

**Символическая ссылка в отличие от жесткой имеет свой собственный номер индексного дескриптора и имеет свой тип.**

Создание и/или удаление символической ссылки никогда не затрагивают ни имени файла, на который она ссылается, ни его индексного дескриптора.

Файл может быть удален, а ссылка остается («висящая»).

Файл может не существовать в момент создания символической ссылки.

## Права доступа к файлам

Файловая система контролирует права доступа к файлам, выполняет операции создания и удаления файлов, а также выполняет запись/чтение данных файла.

Каждый файл в ОС UNIX содержит набор прав доступа, по которому определяется, как пользователь взаимодействует с данным файлом. Этот набор хранится в индексном дескрипторе данного файла в виде 16-разрядного целого значения, из которого обычно используется 12 битов. Каждый бит используется как переключатель, разрешая (значение 1) или запрещая (значение 0) тот или иной доступ.

Три первых бита устанавливают различные виды поведения при выполнении.

Оставшиеся девять делятся на три группы по три, определяя права доступа для владельца, группы и остальных пользователей.

Каждая группа задает права на чтение, запись и выполнение.