Работа с файловой системой и получение помощи.

Базовые понятия

> *"On a UNIX system, everything is a file; if something is not a file, it is a process."*

Файл (regular file) — массив байт, содержащий данные: текстовые файлы, исполняемые файлы, программы и пр.

Также есть специальные файлы, которые для простоты представлены в виде файлов, но на самом деле ими не являются: например, сокеты и именованные каналы.

Директория - файл, содержащий имена других файлов (пары вида имя — номер inode). В Linux, как и в UNIX, нет разницы между файлом и директорией.

Устройство файловой системы

Файлы образуют дерево:

- корнем является директория `/`, ее также называют **корневой директорией (root)**
- листьями дерева являются файлы

```
python_public
├── README.md
└── src
    ├── Notebook.ipynb
    └── src_titanic.csv
```

inode

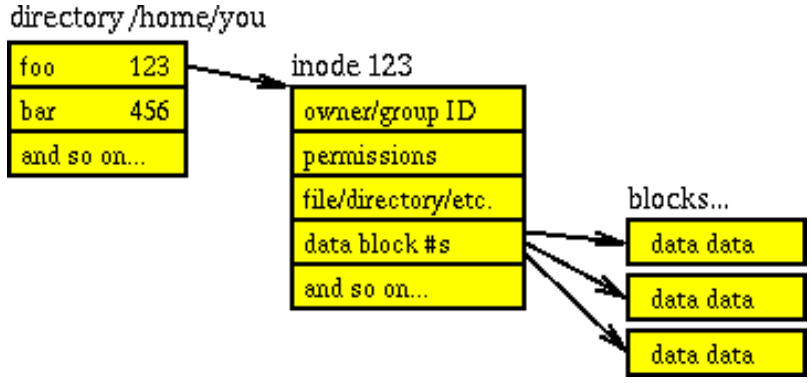
inode — структура данных, используемая файловой системой для хранения всей информации о файле:

- владелец и группа владельца файла,
- тип файла (обычный, каталог, ...),
- дата и время создания, последнее чтение и изменение,
- размер файла,
- информация о физическом расположении данных на диске,
- количество ссылок на файл.

inode

В **inode** содержится вся информация о файле, кроме имени и директории.

Эта информация содержится в специальных файлах типа директория.



Слайд про ls

Этот слайд специально оставлен пустым

Права доступа

`rm -rf /` — **R**e**M**ove **r**ecursively (входя в поддиректории), **f**orcibly (насильно, игнорируя предостережения) `/` (корневую директорию).

Удаляет все.

Если запустить от обычного пользователя (не администратора), то удалятся только файлы самого пользователя.

Это обеспечивается **механизмом прав доступа**.

Интерпретация прав

`ls -l` демонстрирует для каждого файла и для каждой директории колонки такого вида:

```
-r-x------
-rw-------
-rwxr-xr-x
lrwxrwxrwx
drwxr-xr-x
drwxr-xr-x
-rwx------
drwxrwxrwt
```

Интерпретация прав

Первый символ обозначает тип файла:

- `-` — обычные файлы
- `d` — директории.

Следующие три символа определяют права владельца;

Затем описываются права группы-владельца и, наконец, права всех остальных.

Если вместо буквы стоит прочерк, это означает, что право не предоставлено.

Интерпретация прав

- `r` для файлов обозначает право на чтение,
- `w` — право на изменение содержимого,
- `x` — право на исполнение.

Если у текущего пользователя есть право на исполнение файла `x`, который находится в текущей директории, то он может вызвать `./x`, и программа, заданная в этом файле, будет запущена.

Дополнительный материал

Подробное описание полей, которые выводит `ls -l`, можно увидеть в

```
info '(coreutils)What information is listed'
```

Подробное описание прав доступа можно увидеть в

```
info '(coreutils)Mode Structure'
```

Изменение прав доступа

Для изменения прав доступа к файлу используется команда `chmod`, для смены владельца — `chown`.

```
$ chmod +x x.bash # добавить во все триады право на исполнение
$ chmod 700 x.sh # установить права в rwx------
$ chmod 644 x.sh # установить права в rw-r--r--
```

Ссылки

Жесткая ссылка (англ. hard link) — один из путей файла.

Нет разницы между именем при создании и любой другой жесткой ссылкой.

Нельзя создать жесткую ссылку на директорию.

Символьная (англ. symbolic link) — это файл UNIX, содержащий путь к оригинальному файлу, на который ссылается.

Ссылки

Новые ссылки можно создать с помощью утилиты `ln`

```
$ ls -i
661565 target

# создать жесткую ссылку
$ ln target dest

$ ls -i
661565 target   661565 dest

# создать символьную ссылку
$ ln -s target symb_dest
$ readlink -f symb_dest
/test/target
```

Ограничения на имена файлов

В имени файла может быть что угодно, кроме нулевого байта и символа `/`.

В силу этого честно обрабатывать произвольные возможные имена файлов сложно: например, если нужно передать между программами список файлов, то записи в этом списке обычно разделяют нулевым байтом.

Многие утилиты поддерживают специальные флаги (`-0`, `-z`) специально для этого.

Пути

Абсолютный путь — путь к файлу относительно корневой директории, *начинается с /*

Относительный путь — путь к файлу относительно текущей директории, *начинается с любого символа кроме /*

```
$ cd ../a/b # относительный путь
$ cd /home/example/foo # абсолютный
```

Получение помощи

Для получения помощи можно попробовать `--help`, `-h`, `-?`

```
$ ls --help
$ ip addr help
$ git add -h # выведет справку
$ git add --help # откроет man-страницу
```

man

```
man ls
```

man-страница обычно содержит:

- формата вызова,
- причины различных кодов ошибок,
- подробное описание ключей
- иногда примеры.

man. Формат вызова

- квадратные скобки — аргумент не обязательный,
- многоточие — допустимо несколько аргументов такого вида подряд.

```
scp [-346BCpqrTv] [-c cipher] [-F ssh_config]
    [-i identity_file] [-J destination]
    [-l limit] [-o ssh_option] [-P port]
    [-S program] source ... target
```

man. Разделы

```
$ man man # здесь можно найти про разные разделы документации

$ whatis printf
printf (1)           - format and print data
printf (1p)          - write formatted output
printf (3)           - formatted output conversion
printf (3p)          - print formatted output

$ man 1p printf # показать man-страницу из раздела 1p
```

man. apropos

Если неизвестно не только что делает команда, но и как она называется, с этим поможет `apropos`:

```
$ apropos "binary search"
bsearch (3)    - binary search of a sorted array
bsearch (3p)   - binary search a sorted table
git-bisect (1) - Use binary search to find the commit that
introduced a bug
tdelete (3)    - manage a binary search tree
tdelete (3p)   - manage a binary search tree
tdestroy (3)   - manage a binary search tree
tfind (3)      - manage a binary search tree
```

Кавычки нужны, чтобы `binary search` считалось цельной ключевой фразой

man. apropos

Иногда интересует только один конкретный раздел.

Например, какие есть **команды** для сортировки?

```
$ apropos -s 1 sort
bunzip2 (1)          - a block-sorting file compressor, v1.0.8
bzip2 (1)            - a block-sorting file compressor, v1.0.8
comm (1)             - compare two sorted files line by line
ctwill-refsort (1)   - (unknown subject)
sort (1)             - sort lines of text files
sort (1p)            - sort, merge, or sequence check text
files
tsort (1)            - perform topological sort
tsort (1p)           - topological sort
xindy (1)            - create sorted and tagged index from raw
index
```

Формат программ

Операционная система определяет, что делать с исполняемым файлом, по его первым двум байтам.

Пара байтов `#!` называется **shebang** и обозначает, что далее будет представлена команда, аргументом к которой подается данный файл.

Формат программ

Пусть, например, в файле `x.sh`, право на исполнение которого есть, содержится

```
#!/usr/bin/env sh

echo Hi!
```

Тогда при его запуске операционная система вызовет команду

```
/usr/bin/env sh x.sh
```

Переменные

sh поддерживает переменные

```
foo=26
bar="String"
printf "%s" "$foo is a big dragon, who loves to $bar"
```

Отсутствие пробелов вокруг = важно.

```
$ foo =bar # запустить foo с аргументом =bar
$ foo= bar # присвоить foo null string и выполнить bar.
```

Домашнее задание

С помощью man (и info!) самостоятельно изучите доступную документацию к утилитам:

`cd`, `pwd`, `cp`, `mkdir`, `mv`, `rm`, `ls`, `find`