

Работа с файлами в Linux



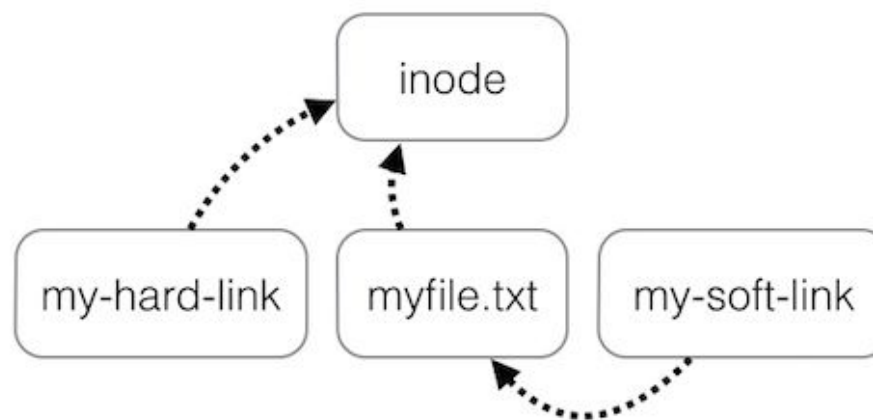
Абстракция ссылок

Номер inode также называют порядковым номером файла

Ссылки — это дополнительные записи каталога, позволяющие обращаться к файлам или каталогам по нескольким именам

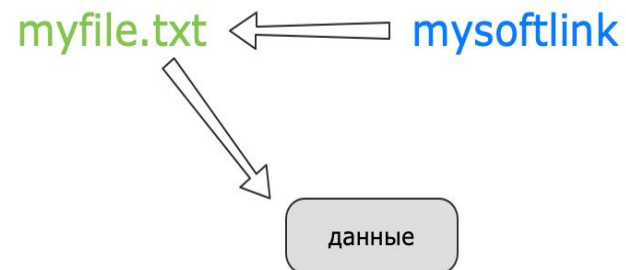
Жесткая ссылка — это запись каталога, указывающая на дескриптор inode

Мягкая (или символическая) ссылка — это запись каталога, указывающая на имя объекта с другим inode



Отличия жестких и мягких ссылок

Жесткая	Мягкая
можно создать только для файлов	На любой тип данных
можно использовать только в пределах одной файловой системы	В пределах разных файловых систем
Указывает на inode	Указывает на имя другого файла или каталога
Файл удаляется когда все жесткие ссылки удалены	Файл и ссылка независимы друг от друга — ссылка становится битой, если файл удален



Редактирование файлов

В мире Linux-UNIX используется множество
текстовых редакторов

Остановимся подробнее на vim и nano



```

:::
iLE88Dj. :jD88888Dj:
.LGitE888D.f8GjjjL8888E;
iE :8888Et. .G8888.
;i E888, ,8888,
D888, :8888:
D888, :8888:
D888, :8888:
D888, :8888:
888W, :8888:
W88W, :8888:
W88W: :8888:
DGGD: :8888:
:8888:
:W888:
:8888:
E888i
tw88D
```

Vim

Может работать в трех режимах:

- основной (визуальный) режим;
- командный режим;
- режим просмотра

После запуска vim главное — знать, как из него выйти

Команда - Описание

:q! - Выход без сохранения

:w - Сохранить изменения

:wq - Сохранить и выйти

:q - Выйти, если нет изменений

i - Перейти в режим вставки символов в позицию курсора

u - Отменить последнее действие

Переключение в режим команд осуществляется клавишей <Esc>





Как выйти из Vim?

Nano

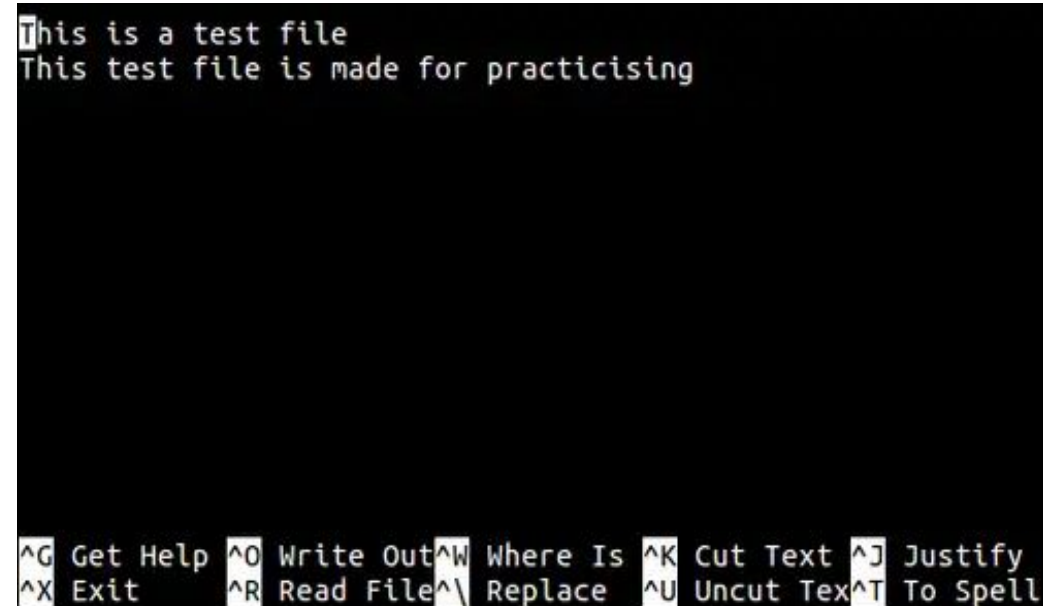
Редактор nano не требует выполнения специальных команд для начала редактирования - открыв документ, вы можете сразу начинать печатать

`nano мемо.txt` - Открывает файл мемо.txt для редактирования

`nano -b мемо.txt` - При сохранении осуществляет резервное копирование предыдущего варианта файла

`nano -m мемо.txt` - Включает управление курсором с помощью мыши (если это поддерживается)

`nano +83 мемо.txt` - Начинает редактирование со строки 83



```
This is a test file
This test file is made for practising

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify
^X Exit      ^R Read File ^\ Replace  ^U Uncut Tex ^T To Spell
```

Права доступа

Существуют три права доступа: чтение (r), запись (w), выполнение (x)

Чтобы просмотреть текущие права доступа, введите команду:

`ls -l <имя файла/каталога>`

Например, `ls -l video.txt`

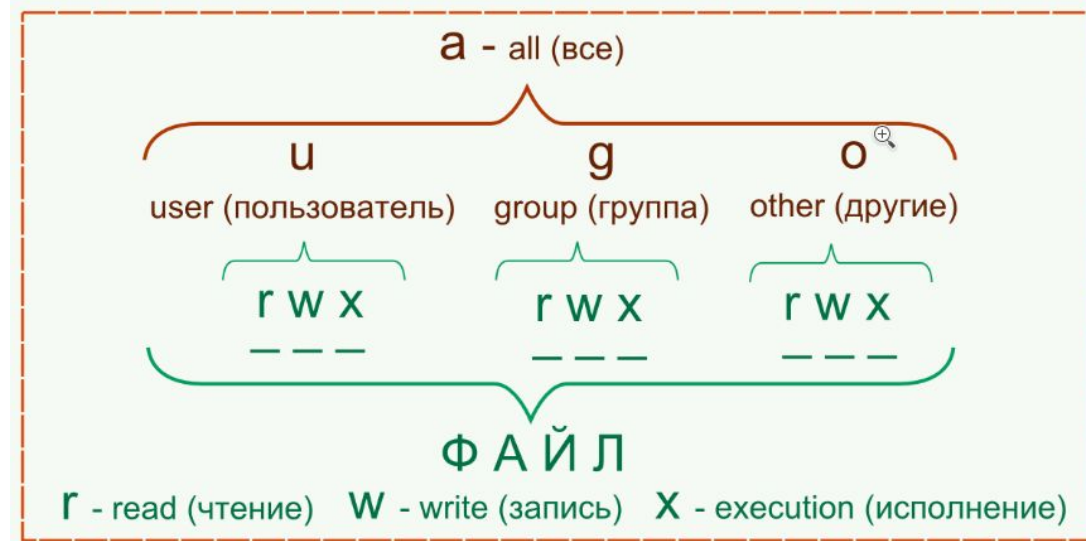
В ответ программа выведет следующую строку:

```
-r--r----- 1den group 300 Apr 1111:11 video.txt
```

Первый символ — это признак каталога.

- Последующие три символа (r--) права доступа владельца файла
- Следующие три символа (r--) права группы владельца
- Последние три символа (---) задают права доступа для прочих

Права доступа к файлам (rwx)



Команды chmod и chown

Права доступа задаются командой chmod

Два способа указания прав доступа: символьный и абсолютный

Пример использования chmod 640 <имя_файла>

Иногда символьный метод оказывается проще:

chmod +x script

chmod -x script

Сделать пользователя владельцем файла:

chown пользователь файл

drwxrwxrwx

d = Directory
r = Read
w = Write
x = Execute

chmod 777

The diagram shows the command 'chmod 777' with three arrows pointing down to 'rwx | rwx | rwx'. Below this, the labels 'Owner | Group | Others' are aligned with the respective permission sets.

rwx | rwx | rwx
Owner | Group | Others

7	rwx	111
6	rw-	110
5	r-x	101
4	r--	100
3	-wx	011
2	-w-	010
1	--x	001
0	---	000

Что означает права доступа 644? 755? 777?

Подстановочные символы

*	Соответствует любым символам
?	Соответствует любому единичному символу
[символы]	Соответствует любому символу, который является членом набора символов
[!символы]	Соответствует любому символу, который не является членом набора символов
[[:класс:]]	Соответствует любому символу, который является членом указанного класса

Класс символов	Значение
[alnum:]	Соответствует любому алфавитно-цифровому символу
[alpha:]	Соответствует любому алфавитному символу
[digit:]	Соответствует любому цифровому символу
[lower:]	Соответствует любой цифре
[upper:]	Соответствует любой букве в нижнем регистре

С чем совпадает Шаблон:

1)*

2)g*

3)b*.txt

4)Data???

5)[abc]*

6)BACKUP.[0-9][0-9][0-9]

7)[[:upper:]]*

8)[![:digit:]]*

9)*[[:lower:]]123]

Команда find

Утилита find реализует поиск файлов методом перебора

find [папка] [параметры] критерий шаблон

Параметры:

-name <имя> - поиска по имени

-perm <режим> - файлы с определенными привилегиями

-size n — поиск по размеру

-group <имя> - заданной группе по имени

-gid <GID> - идентификатор группы равен

-user имя — поиск по имени владельца

-uid UID — по id владельца

-maxdepth <уровень> - глубина поиска

```
sergiy@sergiy-VirtualBox: ~$ find . -name "*.png"
./.cache/mozilla/firefox/n4nit571.default-release/thumbnails/93d303912.png
./.cache/thumbnails/fail/gnome-thumbnail-factory/748da8646b3c9a
./.cache/thumbnails/large/80669861f9aa5aea3fe60cb442f453cc.png
./.cache/thumbnails/large/79c838271e63e41a058caee97086a946.png
./.cache/thumbnails/large/e6d54044b9453935bfde801d7bb4f0f0.png
./.cache/thumbnails/large/8ee7f8d364cd8e60e32d5e39142a2b3f.png
./Изображения/Снимок экрана от 2021-08-12 22-34-59.png
./Изображения/Снимок экрана от 2021-08-02 22-43-37.png
./Изображения/Снимок экрана от 2021-10-12 09-42-38.png
./Изображения/Снимок экрана от 2021-10-12 09-44-48.png
sergiy@sergiy-VirtualBox: ~$
```

Регулярные выражения

Многие программы имеют поддержку регулярных выражений

Символьные классы

- Диапазон (пример `a[2-4]z`)
- Любой одиночный символ — точка `.`
- Начало и конец строки - «карет» `(^)` начало, доллара `($)` - конец
- Повторение символа - `(*)`
- Экранирование - для специальных символов ставим `(\)` пример `www\.sybex\.com`
- Несколько возможных подстрок. Вертикальная черта `(|)`
- Круглые скобки. Обычные круглые скобки обрамляют подвыражения

Команда grep

Команда grep ищет файлы, содержащие заданную строку их имена и найденный фрагмент

grep [опции] шаблон [имя файла...]

Параметры:

--count или -c строк номера

-file <файл> - шаблон из файла

--ignore-case или -i

-recursive или -R или -r — искать во всех подпапках

-extended-regexp -E — расширенные регулярные выражения

Примеры grep eth[01] /etc/*

grep -E "(games|mail).*nologin" /etc/passwd

Команда cut

Извлекает текст из полей файловых записей

Пример `cut -f 6 -d : /etc/passwd`

Основные параметры

`-characters -c` Выбирает только заданные позиции символа (-ов)

`-delimiter -d` Использует заданный разделитель вместо стандартной табуляции

`-fields -f`

Выбирает только заданные поля

`-only-delimited -s` Строки без разделителя не выводятся

Команда awk

Является интерпретатором одноименного языка программирования awk, который впоследствии повлиял на другой язык - Perl. С помощью языка программирования awk можно выполнять следующие действия:

- Объявлять переменные для хранения данных.
- Использовать арифметические и строковые операторы для работы с данными.
- Использовать структурные элементы и управляющие конструкции языка, такие, как оператор if-then и циклы, что позволяет реализовать сложные алгоритмы обработки данных.
- Создавать форматированные отчёты
- Манипулирование данными в текстовых файлах

Команда awk

По умолчанию awk назначает следующие переменные каждому полю данных, обнаруженному им в записи:

\$0 — представляет всю строку текста (запись).

\$1 — первое поле.

\$2 — второе поле.

\$n — n-ное поле.

Поля выделяются из текста с использованием символа-разделителя

Примеры awk '{print \$1}' myfile

\$ awk -F: '{print \$1}' /etc/passwd

<https://habr.com/ru/company/ruvds/blog/327754/>

Команда sed

Sed -- это неинтерактивный строчный редактор.

\$ sed `опции -е` команды файл

Параметры:

- n, --quiet - не выводить содержимое буфера шаблона в конце каждой итерации;
- e - команды, которые надо выполнить для редактирования;
- f - прочитать команды редактирования из файла;
- i - сделать резервную копию файла перед редактированием;
- l - указать свою длину строки;
- r - включить поддержку расширенного синтаксиса регулярных выражений;
- s - если передано несколько файлов, рассматривать их как отдельные потоки, а не как один длинный.

Команда sed

СИНТАКСИС РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ

* - любой символ, любое количество;

\+ - как звездочка, только один символ или больше;

\? - нет или один символ;

\{i\} - любой символ в количестве i;

\{i,j\} - любой символ в количестве от i до j;

\{i,\} - любой символ в количестве от i и больше.

Команда sed

q - завершает работу сценария;

d - удаляет буфер шаблона и запускает следующую итерацию цикла;

p - вывести содержимое буфера шаблона;

n - вывести содержимое буфера шаблона и прочитать в него следующую строку;

s/что_заменять/на_что_заменять/опции - замена символов, поддерживаются регулярные выражения;

y/символы/символы - позволяет заменить символы из первой части на соответствующие символы из второй части;

w - записать содержимое буфера шаблона в файл;

D - если буфер шаблона не содержит новую строку, удалить его содержимое и начать новую итерацию цикла, иначе удалить содержимое буфера до символа перевода строки и начать новую итерацию цикла с тем, что останется;

g - заменить содержимое буфера шаблона, содержимым дополнительного буфера;

G - добавить новую строку к содержимому буфера шаблона, затем добавить туда же содержимое дополнительного буфера.

Команда sed

```
sed -n '5,10p' /etc/group
```

```
sed '1,20d' /etc/group
```

```
sed 's/root/losst/g' /etc/group
```

```
sed '1,10 s/0/1000/g' /etc/group
```

<https://habr.com/ru/company/ruvds/blog/327530/>

Архивирование

Архивирование - это объединение нескольких небольших файлов в один

Архивирование в linux это не одно и то же что и сжатие файлов linux

Tar - это стандартная утилита, с помощью которой выполняется архивирование файлов Linux

tar опции файл_для_записи /папка_файлами_для_архива

A - добавить файл к архиву **c** - создать архив в linux **d** - сравнить файлы архива и распакованные файлы в файловой системе

j - сжать архив с помощью Bzip **z** - сжать архив с помощью Gzip **r** - добавить файлы в конец архива

t - показать содержимое архива **u** - обновить архив относительно файловой системы **x** - извлечь файлы из архива

v - показать подробную информацию о процессе работы **f** - файл для записи архива **-C** - распаковать в указанную папку

Архивирование tar

Примеры

`tar -cvf archive.tar.gz /path/to/files` - создать

`tar -xvf archive.tar.gz` — распаковать

`tar -zcvf archive.tar.gz /path/to/files` — со сжатием gzip

`tar -zxvf archive.tar.gz`

`tar -xvf archive.tar file.txt` — извлечь один файл

`tar -xvf archive.tar --wildcards '*.php'` — извлечь файлы по шаблону

`tar -xvf archive.tar -C /path/to/dir` — распаковать в папку

Архивирование GZIP

Это стандартная утилита сжатия в Unix/Linux

gzip опции файл

gunzip опции файл

Параметры:

-c - выводить архив в стандартный вывод -d — распаковать -f - принудительно
распаковывать или сжимать -l - показать информацию об архиве -r - рекурсивно
перебирать каталоги -0 - минимальный уровень сжатия -9 - максимальный уровень
сжатия

Но чтобы сжать папку в Linux вам придется сначала заархивировать ее с помощью tar, а уже потом сжать файл архива с помощью gzip

Архивирование BZIP

bzip2 - еще одна альтернативная утилита сжатия для Linux. Она более эффективная чем gzip, но работает медленнее

Параметры аналогичные gzip

bzip2 опции file

Bunzip2 опции file

Архивирование RAR

Формат RAR был разработан российским программистом, Евгением Рошалом

Изначально формат был разработан для DOS

Распаковка rar в linux выполняется с помощью утилиты unrar. Упаковка архива командой rar

e - распаковать архив;

l - вывести список файлов внутри архива;

p - вывести распакованный файл в стандартный вывод;

t - проверить архив;

v - вывести подробную информацию про архив;

x - извлечь файлы, сохраняя полный путь внутри архива.

Архивирование RAR

Распаковать в текущую папку достаточно передать утилите имя архива и команду e:

```
unrar e file.rar
```

Вы можете вывести содержимое архива, ничего не распаковывая с помощью команды l:

```
unrar l file.rar
```

Чтобы сохранять полный путь, который прописан в архиве нужно использовать команду x:

```
unrar x file.rar
```

Также вы можете проверить архив на целостность с помощью команды t:

```
unrar t file.rar
```

