

**Символьная (мягкая) ссылка:**

- Указывает на целевой файл или каталог. Фактически является небольшим файлом, содержащим путь до целевого файла.
- Не содержит внутри себя содержимого самого файла. Содержит путь к целевому файлу.
- Имеет собственные права доступа, которые не распространяются на целевой файл.
- Удаление / переименование / перемещение целевого файла не обновляет автоматически ссылку.
- Ссылка начинает указывать на несуществующий файл, становится неработающей.
- Изменение прав доступа у целевого файла не обновляет права доступа у ссылки.
- Может быть создана для директории.
- Ссылка и целевой файл имеют разные файловые индексы (inode) в файловой системе.
- Может указывать на несуществующий файл.
- Символическая ссылка может использовать относительный путь до целевого файла.

**Жесткая ссылка:**

- Является своего рода еще одним именем на файл. *данные физически записываются на диск*
- Не может указывать на директорию.
- Нельзя создавать жесткие ссылки между файлами разных файловых систем.
- Не может указывать на несуществующий файл.
- Жесткая ссылка и файл, для которого она создавалась, имеют одинаковые индексы (inode) в файловой системе.

**Ln:**

- d — разрешить создавать жесткие ссылки для директорий суперпользователю;
- sf — перезаписать целевую ссылку; `ln -sf test_file.txt link_file.txt`
- i — спрашивать нужно ли удалять существующие ссылки;
- P — создать жесткую ссылку; *по умолчанию*
- r — создать символическую ссылку с относительным путем к файлу;
- s — создать символическую ссылку.

`rm link_file.txt / unlink - удаление`

номер inode является уникальным для каждого файла или каталога в файловой системе и используется для идентификации файла или каталога на уровне ядра операционной системы. это структура данных, которая хранит информацию о файле или каталоге, такую как его владелец, права доступа, дата и время создания и изменения, размер и расположение на жестком диске

**Абсолютный путь.** Полный и точный путь к файлу или каталогу из корневого каталога системы.

**Относительный путь.** Расположение файла или каталога относительно текущего рабочего каталога.

**LS:**

- a — отображение всего содержимого директории, включая скрытые файлы (их имена начинаются с точки).
- t *sort by time* — u *когда файлы использовались посл. раз*
- b — r *reverse*
- c — вывод файлов с сортировкой по времени внесения последнего изменения.
- S *size*
- d — вывод сведений о директории без вывода ее содержимого.
- i — отобразить inode, в котором находится файл.
- l — вывести длинный список с подробной информацией. *дата, владелец, права, время изменения*

```

1. user@bash: ls -l chick.png
2. -rw-r--r-- x 1 harry users 2.7K 4 янв 07:32 chick.png
3. user@bash:
4. user@bash: chmod g + x chick.png
5. user@bash: ls -l chick.png
6. -rw-r--r-- x 1 harry users 2.7K 4 янв 07:32 chick.png
7. user@bash:
8. user@bash: chmod uw chick.png
9. user@bash: ls -l chick.png
10. -r-xr-x - x 1 harry users 2.7K 4 янв 07:32 chick.png
11. user@bash:

```

*group + x* *user -w*

*чел-во ссылок*

**chmod [разрешение] [путь] — изменение прав доступа**

*-r read*  
*-w write*  
*-x execute*

**pwd** - текущее местоположение  
**cat** - чтение файла + конкатенация  
**mkdir** - создать директорию  
**echo** - вывод в консоль  
**touch** - создать пустой файл  
**more(!)** - постраничный вывод файла(выход по нажатию q)  
**open()**  
**ln(!)** - создание ссылок  
**man** - помощь  
**wc(-l)** - word count  
**head(!)** - начало файла  
**tail(!)** - конец файла (как прочитать со 2 по 4 строку?)  
**sort(-k)** - отсортировать  
**grep(!)**  
**chmod** - изменить права  
**rm, rmdir** - (почитать про различия)  
6 - copy (почитать в чем разница dir и dir/)  
**mv - rename, move**  
**cd(..name\_directory)** - сменить каталог(change directory)  
**ls(-l -a -b -t -d -i -r** будет на защите!)- список в текущей директории

- **rm** removes complete directories, including [subdirectories](#) and [files](#).
- The **rmdir** command removes empty directories.

## ср

**-f** удаляет существующие файлы, в которые происходит копирование.  
**-i** Спрашивает, нужно ли перезаписывать существующие файлы, в которые происходит копирование.  
**-r** Сохраняет исходные параметры файла, такие как владелец, группа, права доступа, время последней модификации и время последнего доступа к файлу.  
**-R** Копировать каталоги рекурсивно;

## touch

**-c** меняет ctime у выбранного файла, и если файлы не существует, то не создает его.  
**-a** меняет atime, и, соответственно,  
**-m** меняет mtime.  
**-t** задать выбранное время

## sort:

**-o** - output to file  
**-r** - in reverse order  
**-n** - sort numerically  
**-k (number)** - sort by 'number'th column  
**-c** check if the file is already sorted

## wc:

**-l** - number of lines  
**-w** - number of words  
**-c** - number of bytes  
**-m** - number of characters

## Cat:

**-n** show line numbers  
**-b** - like -n, but omit blank lines  
**-s** do not show errors about non-existing files

## mv

**-f** не спрашивать разрешения.  
**-i** спрашивать разрешение пользователя.  
**-u** Перемещать только если исходный файл новее, чем файл назначения или если файл-назначения отсутствует.  
**-v** Выводить информацию о каждом файле,

Регулярные выражения — инструмент для определения общих шаблонов поиска.

`grep '^T' poem.txt`

(^) найти строки, начинающиеся с конкретного символа

Grep:

-i - ignore case

-n - show the number of string found

-v - show everything that DOESN'T match the expression typed





