

# Symlink/Hardlink

Разберём, как понять, какая система инициализации работает на сервере под управлением Linux.

Давайте посмотрим на наш процесс с PID 1 — он называется просто `init`.

Для начала воспользуемся утилитой `ps` с ключом `-p`.

На машине с `upstart` мы увидим, что процесс называется `upstart`. Это сразу даёт нам информацию, где хранятся конфигурационные файлы и как мы можем управлять запущенными сервисами.

Далее посмотрим на файл, который запускает `init`, внимательнее. В нашем дистрибутиве он расположен по пути `/sbin/init`.

Воспользуемся командой `ls -la`, которая показывает нам расширенную информацию о файлах и директориях:

```
ls -la /sbin/init
```

Файл `/sbin/init` — это ссылка на другой файл, который называется `systemd`. Это даёт нам информацию о том, что система работает под управлением `systemd`.

## Symlink (Симлинк)

**Симлинк** (или символическая ссылка) — это такой специальный вид файла. Вместо данных он хранит путь до другого файла, который открывается при обращении к симлинку.

«Указывать» симлинк может на другой файл, каталог или даже на несуществующий файл. Такой симлинк называют битым.

Симлинки позволяют поддерживать обратную совместимость при переходе. Для пользователя ничего не меняется, процесс с PID 1 всё ещё называется `init`, однако теперь его работу выполняет исполняемый файл `systemd`.

Они позволяют не копировать файлы. Вместо того чтобы копировать эти файлы в директории, к которым обращается программа, мы просто создаём символические ссылки.

Давайте вернёмся к конфигурации `nginx`. Мы видим здесь две директории: `sites-enabled` и `sites-available`. В этих директориях хранятся конфигурационные файлы для `nginx`. Мы можем хранить конфигурацию в `sites-available`, а в `sites-enabled` хранить только симлинк. Тогда мы можем гибко отключать или подключать сайты без необходимости удалять файлы или хранить их в стороне.

Чтобы создать симлинк, используется команда:

```
ln -s file1 symlink1
```

Мы создаём новый объект файловой системы с именем `symlink1`, который указывает на уже существующий файл `file1`.

Давайте теперь посмотрим на созданный симлинк при помощи команды `ls`:

```
ls -la symlink1
```

Обратите внимание на тип (или атрибут), указанный буквой `l` перед правами доступа к файлу: раньше мы видели здесь либо прочерк, либо букву `d` (что означало обычный файл либо директорию соответственно). В данном случае он указывает на то, что этот файл является символической ссылкой на другой файл. То же самое отображается и в его имени — `symlink1 -> file1`.

Если мы удалим символическую ссылку (при помощи команды `unlink` либо уже знакомой нам `rm`), исходный файл не удалится:

```
rm symlink1
ls
ln -s file1 symlink1
```

Если же мы удалим исходный файл, символическая ссылка тоже продолжит существовать. Однако вести она будет в никуда.

```
cat symlink1
touch file1
```

При попытке доступа к ней мы получим ошибку.

Для работы с симлинками существует полезная команда `readlink`. Она показывает нам, куда указывает симлинк.

Кроме того, для поиска символических ссылок при помощи `find` используется ключ:

```
find -type l
```

У утилиты `test` есть возможность проверять, является ли файл символической ссылкой:

```
test -l symlink1
```

Кроме символической ссылки существует понятие жёсткой ссылки или `hardlink`. Для создания жёсткой ссылки также используется команда `ln`:

```
ln file1 hardlink1
ls -la
```

В отличие от символической ссылки, жёсткая ссылка отображается командой `ls` просто как ещё один файл.

## Hardlink (Хардлинк)

Разберём отличия симлинка и хардлинка.

Inode хранит метаинформацию, служебную информацию о файле или каталоге (время модификации, время создания, права доступа). Это такой специальный объект файловой системы, просмотреть его содержимое мы можем при помощи команды `stat file`.

Каждый inode имеет свой номер, который присваивается ему в момент создания.

Чтобы увидеть номера inode наших файлов, мы можем использовать команду `ls -lih`

Давайте теперь внимательно посмотрим на симлинк и исходный файл. Номера их inode различаются. Это значит, что симлинк — это отдельный объект файловой системы, то есть это просто ещё один файл.

А вот в случае хардлинка его номер inode точно такой же, как и у исходного файла — то есть это один и тот же файл.

Чтобы определить, является ли файл хардлинком, нам требуется узнать номер inode при помощи команды `stat` или `ls -la`.

Далее использовать команду `find . -inum NUM`.

Или можно использовать команду `find . -samefile /path/to/file`.

Hardlink не может указывать на файл в другой файловой системе (так как inode может принадлежать только одной ФС), а symlink — может.

При редактировании файла-ссылки в случае с хардлинком изменятся оба файла, так как это один и тот же объект, а в случае с симлинком можно изменять его имя, атрибуты, направить его на другой файл, и при этом оригинальный файл не будет затронут.