Last updated Saturday, Nov 16, 2019

Linux and Unix ln command tutorial with examples

Tutorial on using ln, a UNIX and Linux command to make links between files. Examples of creating a hard link, creating a symbolic link, and a broken symbolic link.

Estimated reading time: 3 minutes

Table of contents


```
LN(1)                  User Commands                  LN(1)

NAME
       ln - make links between files
```

What is the ln command in UNIX?

The `ln` command is a command line utility for making links between files. It supports creating a hard and symbolic links to data on disk.

What is the difference between a hard and symbolic link?

To understand the difference between a hard and symbolic link it is important to first understand the relationship between a file and data on disk. When a file is created the filename connects a file system with bytes that have been written to disk. When a file is removed the data on disk remains but the file system has no way of accessing it.

A hard link is a direct link to the data on disk. This means data can be accessed directly via an original filename or a hard link. Both the original file and the hard link are direct links to the data on disk. The use of a hard link allows multiple filenames to be associated with the same data on disk.

A symbolic link (also sometimes known as a soft link) does not link directly to the data on disk but to another link to the data on disk. On most operating systems folders may only be linked using a symlink. Symbolic links can link across file systems to link a folder on an external hard drive.

How to create a hard link

To create a hard link using the `ln` command pass the full path of the target file and the link name. This has the effect of creating a new file that links to the same data on disk as the target file. In the following example `target.txt` is linked to via `link.txt` .

```
ls
cat target.txt
target file
ln target.txt link.txt
ls
target.txt link.txt
cat link.txt
target file
```

Editing `link.txt` has the effect of changing the underlying data on disk. The files `link.txt` and `target.txt` are therefore equivalent.

```
echo "link edit" >> link.txt
cat target.txt
target file
link edit
```

How to create a hard link in the current directory

To create a hard link in the current directory to a target file pass the path of the file or folder. This will create a hard link to the target file.

```
tree
.
└── foo
    └── bar.txt

1 directory, 1 file
ln foo/bar.txt
tree
.
├── bar.txt
└── foo
    └── bar.txt

1 directory, 2 files
```

How to create a symbolic link

To create a symbolic link pass the `-s` option to the `ln` command followed by the target file and the name of link. In the following example a file is symlinked into the `bin` folder.

```
ln -s ~/code/notes/notes ~/bin/notes
ls -l ~/bin/ | grep notes
lrwxrwxrwx 1 george users   29 Oct  7 10:07 notes -> /home/george/cod
```

In the following example a mounted external drive is symlinked into a home directory. This allows for convenient browsing of the external drive within the home directory.

```
ln -s /mnt/external-drive ~/mydrive
```

If the hard drive is unmounted the symlink in the home directory will still be present. This scenario is a broken symlink.

```
cd ~/mydrive
cd: no such file or directory mydrive
```