

# LN Command - Create Links

<https://linuxize.com/post/how-to-create-symbolic-links-in-linux-using-the-ln-command/>

## Links Types

There are two types of links in Linux/UNIX systems:

- **Hard links.** You can think a hard link as an additional name for an existing file. Hard links are associating two or more file names with the same `inode`. You can create one or more hard links for a single file. Hard links cannot be created for directories and files on a different filesystem or partition.
- **Soft links.** A soft link is something like a shortcut in Windows. It is an indirect pointer to a file or directory. Unlike a hard link, a symbolic link can point to a file or a directory on a different filesystem or partition.

## How to Use the `ln` Command

`ln` is a command-line utility for creating links between files. By default, the `ln` command creates hard links. To create a symbolic link, use the `-s` (`--symbolic`) option.

The `ln` command syntax for creating symbolic links is as follows:

```
ln -s [OPTIONS] FILE LINK
```

### Copy

- If both the `FILE` and `LINK` are given, `ln` will create a link from the file specified as the first argument (`FILE`) to the file specified as the second argument (`LINK`).
- If only one file is given as an argument or the second argument is a dot (`.`), `ln` will create a link to that file in the **current working directory**. The name of the symlink will be the same as the name of the file it points to.

By default, on success, `ln` doesn't produce any output and returns zero.

# Creating Symlink To a File

To create a symbolic link to a given file, open your terminal and type:

```
In -s source_file symbolic_link
```

Copy

Replace `source_file` with the name of the existing file for which you want to create the symbolic link and `symbolic_link` with the name of the symbolic link.

The `symbolic_link` parameter is optional. If you do not specify the symbolic link, the `ln` command will create a new link in your current directory:

In the following example, we are creating a symbolic link named `my_link.txt` to a file named `my_file.txt`:

```
In -s my_file.txt my_link.txt
```

Copy

To verify that the symlink was successfully created, use the `ls` command:

```
ls -l my_link.txt
```

Copy

The output will look something like this:

```
lrwxrwxrwx 1 linuxize users 4 Nov 2 23:03 my_link.txt -> my_file.txt
```

Copy

The `l` character is a file type flag that represents a symbolic link. The `->` symbol shows the file the symlink points to.

# Creating Symlinks To a Directory

The command for creating a symbolic link to a directory is the same as when creating a symbolic link to a file. Specify the directory name as the first parameter and the symlink as the second parameter.

For example, if you want to create a symbolic link from the `/mnt/my_drive/movies` directory to the `~/my_movies` directory you would run:

```
In -s /mnt/my_drive/movies ~/my_movies
```

Copy

## Overwriting Symlinks

If you try to create a [symbolic link that already exists](#), the `ln` command will print an error message.

```
In -s my_file.txt my_link.txt
```

Copy

```
In: failed to create symbolic link 'my_link.txt': File exists
```

Copy

To overwrite the destination path of the symlink, use the `-f` (`--force`) option.

```
In -sf my_file.txt my_link.txt
```

Copy

## Removing Symlinks

To [delete/remove symbolic links](#) use either the `unlink` or `rm` command.

The syntax of the `unlink` is very simple:

```
unlink symlink_to_remove
```

Copy

Removing a symbolic link using the `rm` command is the same as when removing a file:

```
rm symlink_to_remove
```

Copy

No matter which command you use, when removing a symbolic link not append the `/` trailing slash at the end of its name.

If you delete or move the source file to a different location, the symbolic file will be left dangling (broken) and should be removed.

---

Revision #1

Created 23 December 2023 15:12:55 by ColtM

Updated 7 August 2024 23:24:39 by ColtM