



Tutorial Link <https://course.testpad.chitkara.edu.in/tutorials/Hard and Soft Links in Linux/6305a1815611a5348833e4eb>

TUTORIAL

Hard and Soft Links in Linux

Topics

1.1 Introduction to Links in Linux

Introduction to Links in Linux

A link in Linux is a pointer to a file. Like pointers in any programming languages, links in Linux are pointers pointing to a file or a directory. Creating links is a kind of shortcut to access a file. Links allow more than one file name to refer to the same file, elsewhere.

There are two types of links :

1 Soft Link or Symbolic links

2 Hard Links

These links behave differently when the source of the link (what is being linked to) is moved or removed. Symbolic links are not updated (they merely contain a string which is the pathname of its target); hard links always refers to the source, even if moved or removed.

For example, if we have a file a.txt. If we create a hard link to the file and then delete the file, we can still access the file using hard link. But, if we create a soft link of the file and then delete the file, we cannot access the file through soft link and soft link becomes dangling. Basically, hard link increases the reference count of a location while soft links work as a shortcut (like in Windows).

1. Hard Links

These links behave differently when the source of the link (what is being linked to) is moved or removed. Symbolic links are not updated (they merely contain a string which is the pathname of its target); hard links always refers to the source, even if moved or removed.

For example, if we have a file a.txt. If we create a hard link to the file and then delete the file, we can still access the file using hard link. But, if we create a soft link of the file and then delete the file, we cannot access the file through soft link and soft link becomes dangling. Basically, hard link increases the reference count of a location while soft links work as a shortcut (like in Windows).

- Removing any link just reduces the link count, but does not affect other links.
- We cannot create a hard link for a directory to avoid recursive loops.
- If the original file is removed then the link will still show the content of the file.
- The command to create a hard link is:

```
$ ln [original filename] [link name]
```

2. Soft Links

A soft link is similar to the file shortcut feature used in Windows Operating systems. Each soft linked file contains a separate Inode value that points to the original file. As similar to hard links, any changes to the data in either file is reflected in the other. Soft links can be linked across different file systems, although if the original file is deleted or moved, the soft linked file will not work correctly (called hanging link).

- `ls -l` command shows all links with first column value `l`? and the link points to the original file.
- Soft Link contains the path for the original file and not the contents.
- Removing soft link does not affect anything but removing the original file, the link becomes “dangling” link which points to a non-existent file.
- A soft link can link to a directory.
- Link across file systems: If you want to link files across the file systems, you can only use symlinks/soft links.
- The command to create a Soft link is:

```
$ ln -s [original filename] [link name]
```

Summary

- A link in UNIX is a pointer to a file.
- There are two types of links :Soft Link or Symbolic links and Hard Links

- Each hard linked file is assigned the same Inode value as the original, therefore they refer the same physical file location.
- Each soft linked file contains a separate Inode value that points to the original file.



CodeQuotient

Tutorial by codequotient.com | All rights

reserved, CodeQuotient 2025