Contents

[**Link:** 2](#_Toc477298569)

[**Significance of links:** 2](#_Toc477298570)

[**Hard Link:** 2](#_Toc477298571)

[**When to use Hard Links:** 3](#_Toc477298572)

[**Soft Link:** 4](#_Toc477298573)

[**When to use Soft Links:** 5](#_Toc477298574)

[**Notes:** 5](#_Toc477298575)

# **Link:**

* Link is simply pointer to file/directory that points to the location of that file/directory.
* There are two types of links in Linux:

1. Hard Link
2. Soft Link/Symlink/Symbolic Link

* Say, there is a file named file1 and we are going to create hard link of the file1 named file2 and to create soft link of the file1 named file3.

## **Significance of links:**

File names are stored in directories. Actual data is stored somewhere else on storage place. Same file may have multiple names and may have located at different places (even in different directory and even in different file system). Linux user are allowed to create link which points to a file having multiple names and Linux allows user to access data through file’s inode number.

* When it is need of accessing same file from more than one locations, Links can be created to required file rather saving file on each location and wasting storage space.

## **Hard Link:**

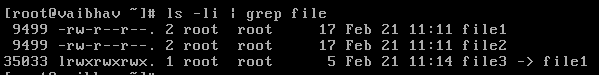
* To create hard link of file1,

Command: ln file1 file2

This will create hard link of file1. It means file2 is a hard link of file1.



* Hard-linked file (file2)have following properties compared to original file (file1):



* Same inode number
* same permission set
* same file size
* same content
* same time of creation
* different file name
* Once hard link is created, it is impossible to know which one is original file and which hard-linked file is.

File1--------------

\_\_\_\_\_\_\_

Content

Inode

\_\_\_\_\_\_\_

Vaibhav Jariwala

File2………………

9499

* Even if, original file is deleted, hard-linked file is still accessible. It means, we are still able to access data of file through its hard link as both files points to same inode.
* Source file must be present to create hard link. Hard link cannot be created on non-existent file.
* Hard link works at only file level. It doesn’t work at directory level. It means that we cannot create hard link for directory.
* Hard link can be created only within file system. Hard Link cannot span across file systems because hard links use same inode number as of original file and inode number is unique within the file system.

### **When to use Hard Links:**

* Storage is less free:

Hard links takes very negligible amount of space (4KB depending upon file system) and do not consume new inodes.

* Performance:

Hard links are better while accessing a file than soft links because hard links access file directly through inode.

* Moving File Location:

Hard link can still work even if original file is moved to other location whereas soft link won’t work.

* Redundancy:

Even if source/original file is removed, data of the file is still accessible until all hard links are removed.

## **Soft Link:**

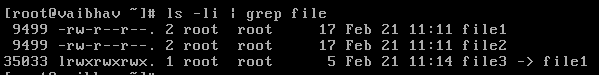
* To create soft link of file1,

Command: ln -s file1 file30

This will create soft link of file1. It means file3 is a soft link of file1.



* Soft-linked file (file3)have following properties compared to original file (file1):



* different inode number
* different permission set
* different file size
* same content
* different time of creation
* different file name

Fille1………………..

9499

Vaibhav Jariwala

Content

Inode

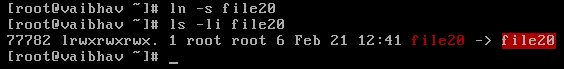
Inode

File3…………………

File1

35033

* If original file is deleted, soft-linked file becomes inaccessible. It means, we are not able to access data of file through its soft link because both files have different inode number.
* As you see in diagram, soft-linked file have different inode number than of original file and new inode number points to file 1. In case, original file (file1) is deleted, new inode number will not be able to point original file (file1).
* Soft link can be created on non-existent file as well. For an example, we create soft link named file,



* Soft link works at both file level and directory level. It means that we can also create soft link for file and directory as well.
* Soft Link can span across File systems. It means that soft link is accessible from different file system as well.
* Soft links are similar to shortcuts in Windows. Like one wants to create shortcut (soft link) of one application like gedit,

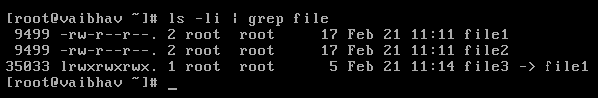
ln -s /usr/bin/gedit ~/Desktop/gedit

### **When to use Soft Links:**

* If there is a need of spanning links across file systems.
* If there is a need of spanning links across directories.

## **Notes:**

1. Once links are created, to differentiate between hard and soft links,



* Soft link is preceded by “***l***” in 1st column of ls command.
* There is a pointer mark (**->**) associated with soft link.
* Original file (file1) and hard-linked file (file2) are of same size (17) but soft-linked file (file3) is of different size (5). This is because, in soft-linked file, name of original file (file having data) gets stored and hence file size is stored. So Soft Link needs more memory than hard link.
* Hard link and soft link acts as normal file in file system. So we can make soft link to a hard link and vice a versa.
* In stat command, Link count indicates total count including all created hard links of the file and source file.
* To list out all hard-linked files and original file if inode number is known,

Command: find -inum <inode number>