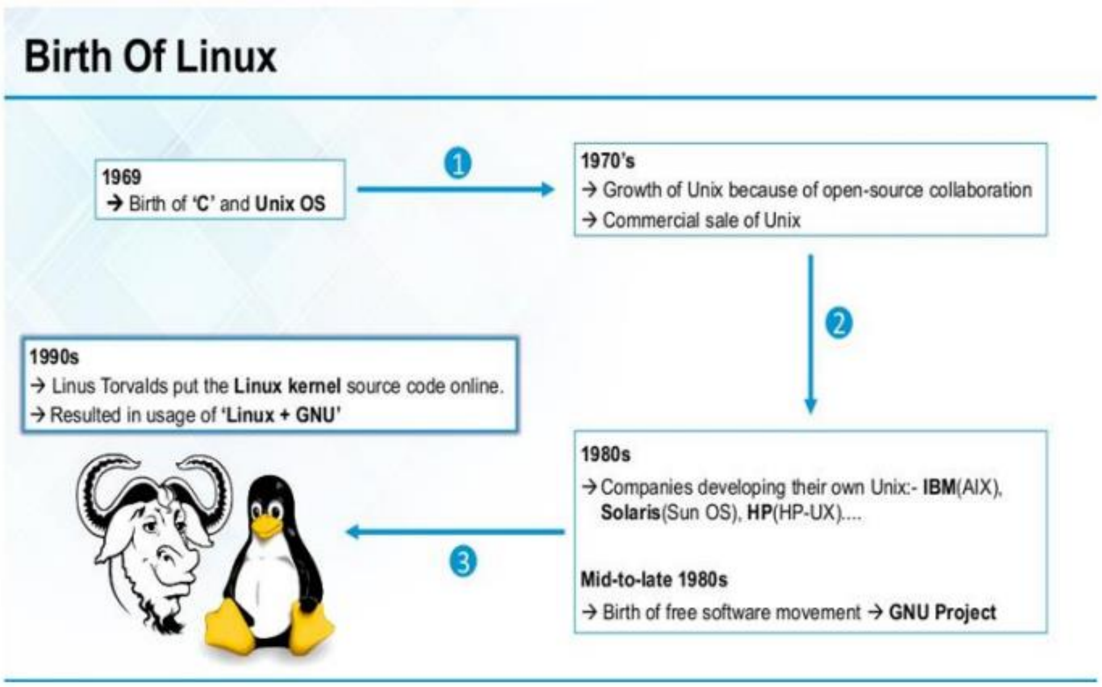
LINUX



Birth Of Linux

1969
→ Birth of 'C' and Unix OS

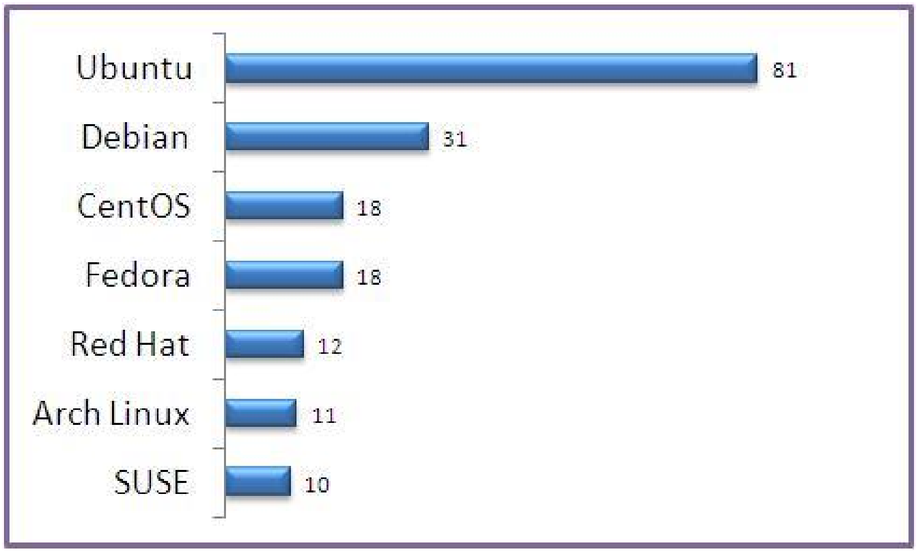
1970's
→ Growth of Unix because of open-source collaboration
→ Commercial sale of Unix

1980s
→ Companies developing their own Unix:- **IBM**(AIX), **Solaris**(Sun OS), **HP**(HP-UX)....

Mid-to-late 1980s
→ Birth of free software movement → **GNU Project**

1990s
→ Linus Torvalds put the **Linux kernel** source code online.
→ Resulted in usage of **'Linux + GNU'**

Top Linux Distribution



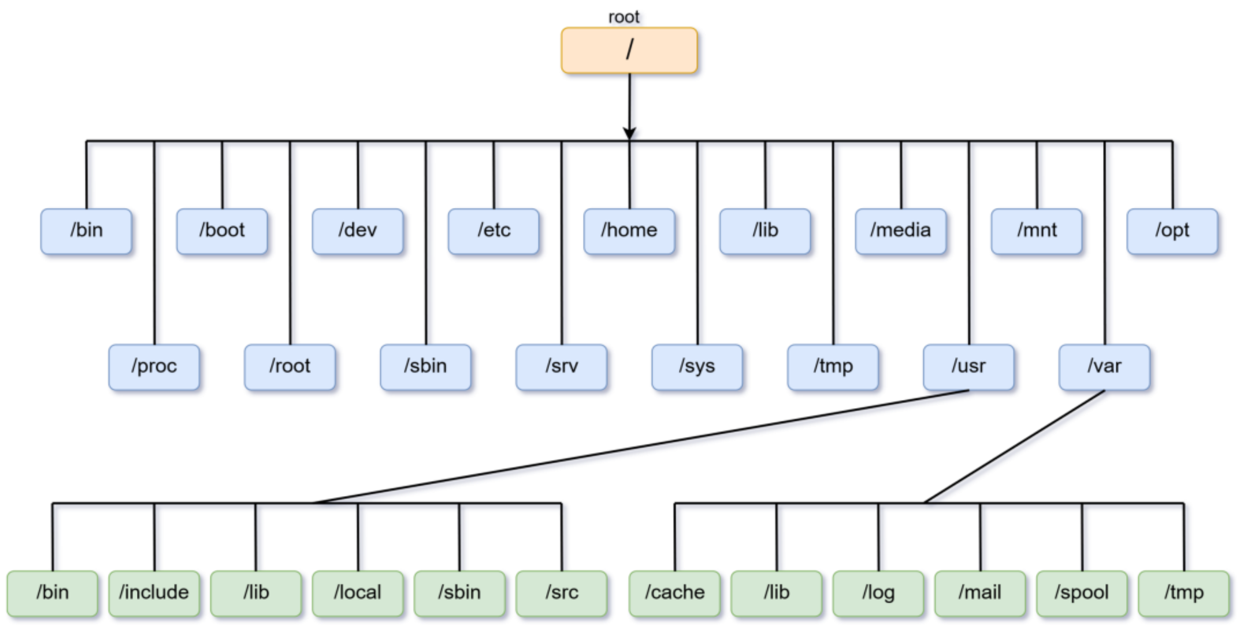
Linux Boot Process

Every time you power on your Linux PC, it goes through a series of stages before finally displaying a login screen that prompts for your username or password. There are 4 distinct stages that every Linux distribution goes through in a typical boot-up process.

The booting process takes the following 4 steps that we will discuss in greater detail:

- BIOS Integrity check (POST)
- Loading of the Boot loader (GRUB2)
- Kernel initialization
- Starting systemd, the parent of all processes

Linux File System Hierarchy



- /bin: All the executable binary programs (file) required during booting, repairing, files required to run into single-user-mode, and other important, basic commands viz., cat, du, df, tar, rpm, wc, history, etc.

- /boot: Holds important files during boot-up process, including Linux Kernel.
- /dev: Contains device files for all the hardware devices on the machine e.g., cdrom, cpu, etc
- /etc: Contains Application's configuration files, startup, shutdown, start, stop script for every individual program.
- /home: Home directory of the users. Every time a new user is created, a directory in the name of user is created within home directory which contains other directories like Desktop, Downloads, Documents, etc.
- /lib: The Lib directory contains kernel modules and shared library images required to boot the system and run commands in root file system.
- /lost+found: This Directory is installed during installation of Linux, useful for recovering files which may be broken due to unexpected shut-down.
- /media: Temporary mount directory is created for removable devices viz., media/cdrom.
- /mnt: Temporary mount directory for mounting file system.
- /opt: Optional is abbreviated as opt. Contains third party application software. Viz., Java, etc.
- /proc: A virtual and pseudo file-system which contains information about running process with a particular Process-id aka pid.
- /root: This is the home directory of root user and should never be confused with '/'
- /run: This directory is the only clean solution for early-runtime-dir problem.
- /sbin: Contains binary executable programs, required by System Administrator, for Maintenance. Viz., iptables, fdisk, ifconfig, swapon, reboot, etc.
- /srv: Service is abbreviated as 'srv'. This directory contains server specific and service related files.
- /sys: Modern Linux distributions include a /sys directory as a virtual filesystem, which stores and allows modification of the devices connected to the system.
- /tmp: System's Temporary Directory, Accessible by users and root. Stores temporary files for user and system, till next boot.
- /usr: Contains executable binaries, documentation, source code, libraries for second level program.
- /var: Stands for variable. The contents of this file is expected to grow. This directory contains log, lock, spool, mail and temp files.

Linux Commands

1. ls command: This command will list the directories and files
    a. ls -l : This command will shows file or directory, size, modified date and time, file or folder name and owner of the file, and its permission.
    b. ls -a: List all files including hidden files starting with '.'.
    c. ls -F: Using the **-F** option with the **ls** command will add the '/' character at the end of each directory.
    d. ls -r: The following command with the **ls -r** option display files and directories in reverse order.
    e. ls -ltr: A combination of **-ltr** will show the latest modification file or directory date as last.
    f. ls -l /tmp: This command will display all the files and directories in /tmp folder
2. cd command: Change directories
    a. cd /usr/local: Change from current directory to /usr/local
    b. cd /usr/local/lib: Chang from current directory to /usr/local/lib
    c. cd -: Switch back to previous directory where you working earlier.
    d. cd ..: Change Current directory to parent directory.
    e. cd ../../: Move two directory up from where you are now.
    f. cd ~: Move to users home directory from anywhere.
3. pwd command: Present working directory command
4. touch command: Creates/append files
    a. touch file1: Create a empty file
    b. touch file2 file3 file4: Creates multiple empty files
5. copy and move commands:
    a. cp /tmp/file1 /etc/file1 : This command will copy file1 from /tmp directory to /etc directory
    b. mv file1 file2: This command will rename file1 to file2
6. cat command: The **cat** (short for "**concatenate**") command is one of the most frequently used commands in Linux/Unix-like operating systems. **cat** command allows us to create single or multiple files, view content of a file, concatenate files and redirect output in terminal or files.
    a. cat /etc/passwd: The below example will show the contents of **/etc/passwd** file.
    b. cat test1 test2: it will display the contents of the **test1** and **test2** file in the terminal.
    c. cat > file5: It will create a file called **file5** file
    d. cat song.txt | more

e. cat song.txt | less: If a file having a large number of content that won't fit in the output terminal and the screen scrolls up very fast, we can use parameters more and less with the **cat** command as shown below.

7. df command: This command is used to check disk space in Linux
   a. df : The "**df**" command displays the information of device name, total blocks, total disk space, used disk space, available disk space, and mount points on a file system.
   b. df -a: The same as above, but it also displays information of dummy file systems along with all the file system disk usage and their memory utilization.
8. du command: The Linux "**du**" (**Disk Usage**) is a standard **Unix/Linux** command, used to check the information of disk usage of files and directories on a machine.
9. Find command: The Linux **find command** is one of the most important and frequently used command command-line utility in Unix-like operating systems. The find command is used to search and locate the list of files and directories based on conditions you specify for files that match the arguments.
   a. find . -name techmint.txt: Find all the files whose name is **tecmint.txt** in a current working directory.
   b. find /home -name techmint.txt: Find all the files under **/home** directory with the name **tecmint.txt**.
   c. find /home -iname techmint.txt: Find all the files whose name is **tecmint.txt** and contains both capital and small letters in **/home** directory.
   d. find / -type d -name Techmint: Find all directories whose name is **Tecmint** in **/** directory.
10. Grep command: Have you ever been confronted with the task of looking for a particular string or pattern in a file, yet have no idea where to start looking? Well then, here is **grep** to the rescue!

11. free command: This command is used to check memory usage in Linux
12. echo command: This command is used to print the message
    e.g: echo This is a test line for checking echo command
13. useradd command: In Linux, a '**useradd**' command is a low-level utility that is used for adding/creating user accounts in **Linux** and other **Unix-like** operating

systems. The '**adduser**' is much similar to the **useradd** command because it is just a symbolic link to it.

a. Useradd Raju: It will create a user called Raju
b. Passwd Raju: Using this command we can set the password for Raju user
c. Cat /etc/passwd: It will display all the users along with their details

   e.g: useradd Anusha
   passwd Anusha
   cat /etc/passwd | grep Anusha
d. Useradd -u 1002 navin: Create user with specific userID
e. Useradd -u 1005 -g devops khan: Creates user with specific user ID and group ID

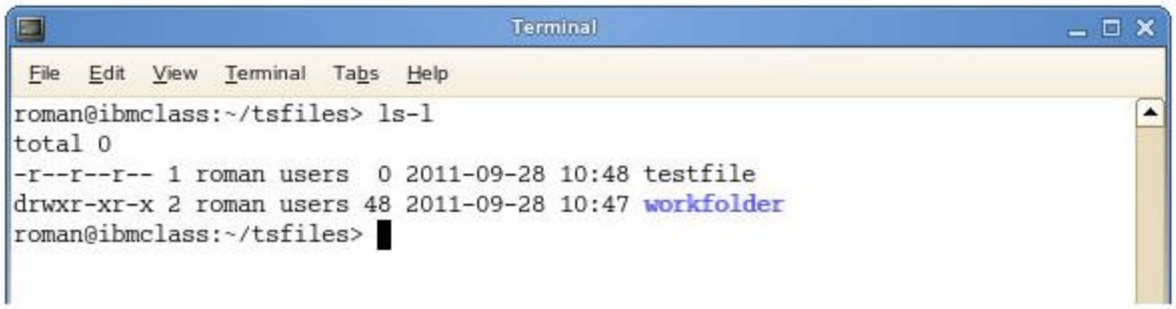
   e.g: groupadd admins
   groupadd webadmins
   groupadd developers
   useradd -u 1006 -g admins,webadmins,developers neha
   id neha

   Create user with expiry date:
   e.g: useradd -e 2022-09-27 Anil

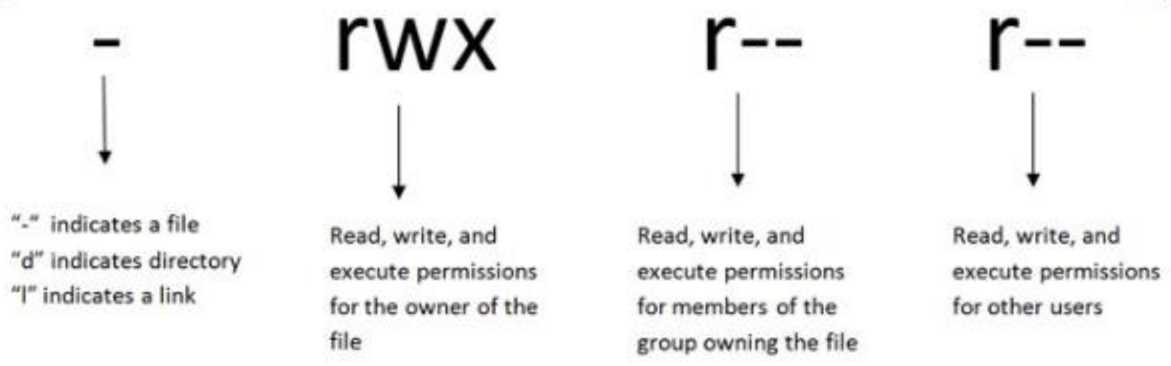
   Create user with password expiry
   e.g: useradd -e 2022-09-27 -f 45 manav

13. Chmod commands: change directory permissions in Linux
    a. chmod +rwx filename
    b. chmod -rwx directoryname
    c. chmod +x filename
    d. chmod -wx filename

    Note that "r" is for read, "w" is for write, and "x" is for execute.



```
roman@ibmclass:~/tsfiles> ls-l
total 0
-r--r--r-- 1 roman users  0 2011-09-28 10:48 testfile
drwxr-xr-x 2 roman users 48 2011-09-28 10:47 workfolder
roman@ibmclass:~/tsfiles>
```



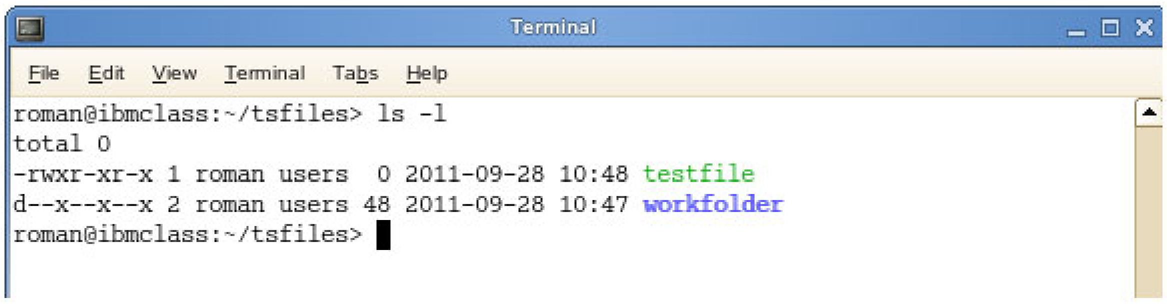
-
rwx
r--
r--
"-" indicates a file
"d" indicates directory
"l" indicates a link
Read, write, and execute permissions for the owner of the file
Read, write, and execute permissions for members of the group owning the file
Read, write, and execute permissions for other users

How to Change Groups of Files and Directories in Linux

By issuing these commands, you can change groups of files and directories in Linux.

- chgrp groupname filename
- chgrp groupname foldername

Note that the group must exit before you can assign groups to files and directories.

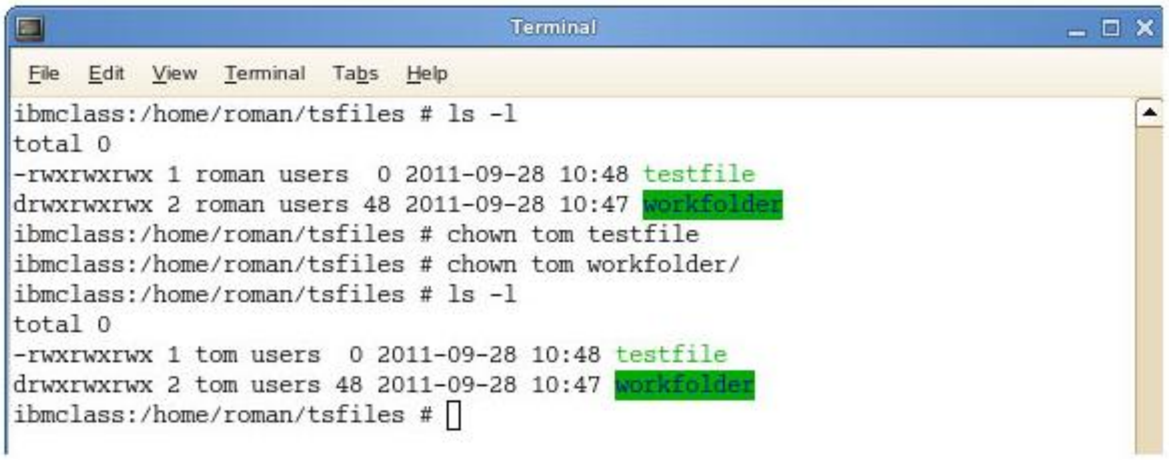


```
roman@ibmclass:~/tsfiles> ls -l
total 0
-rwxr-xr-x 1 roman users  0 2011-09-28 10:48 testfile
d--x--x--x 2 roman users 48 2011-09-28 10:47 workfolder
roman@ibmclass:~/tsfiles>
```

How to Change Ownership in Linux

Another helpful command is changing ownerships of files and directories in Linux:

- chown name filename
- chown name foldername



```
ibmclass:/home/roman/tsfiles # ls -l
total 0
-rwxrwxrwx 1 roman users  0 2011-09-28 10:48 testfile
drwxrwxrwx 2 roman users 48 2011-09-28 10:47 workfolder
ibmclass:/home/roman/tsfiles # chown tom testfile
ibmclass:/home/roman/tsfiles # chown tom workfolder/
ibmclass:/home/roman/tsfiles # ls -l
total 0
-rwxrwxrwx 1 tom users  0 2011-09-28 10:48 testfile
drwxrwxrwx 2 tom users 48 2011-09-28 10:47 workfolder
ibmclass:/home/roman/tsfiles # 
```

How to Change Permissions in Numeric Code in Linux

You may need to know how to change permissions in numeric code in Linux, so to do this you use numbers instead of “r”, “w”, or “x”.

- 0 = No Permission
- 1 = Execute
- 2 = Write
- 4 = Read

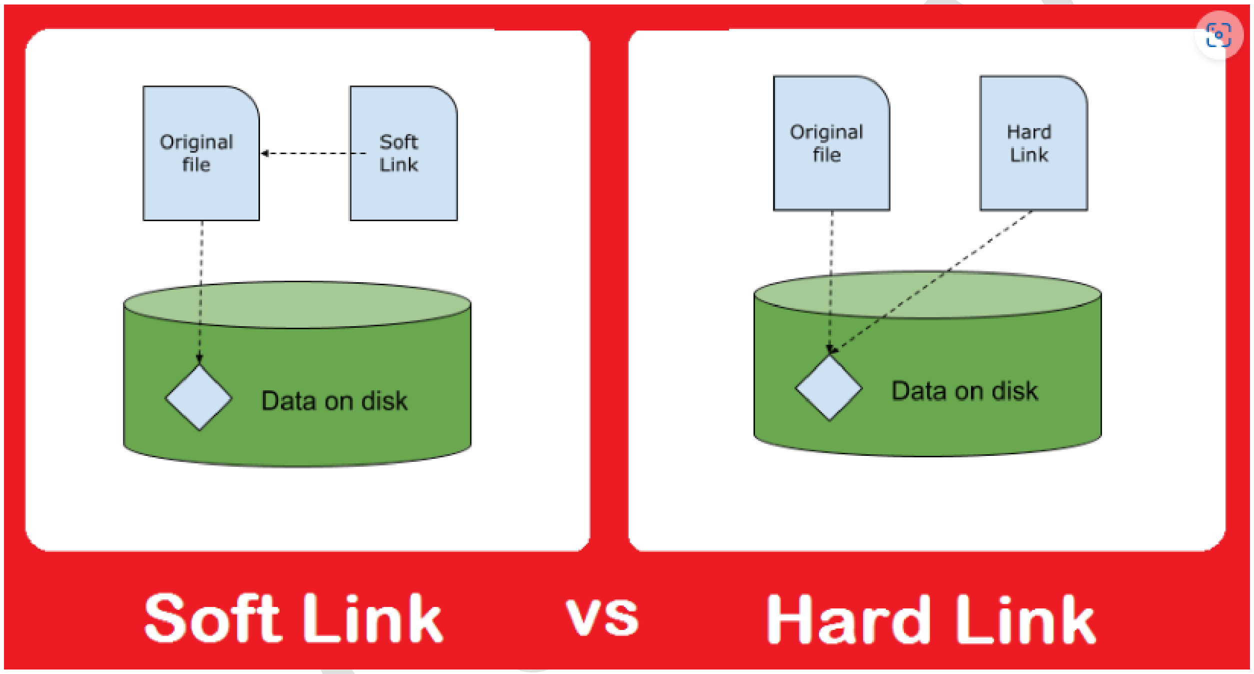
Basically, you add up the numbers depending on the level of permission you want to give.

For example:

- chmod 777 foldername will give read, write, and execute permissions for everyone.

- chmod 700 foldername will give read, write, and execute permissions for the user only.
- chmod 327 foldername will give write and execute (3) permission for the user, w (2) for the group, and read, write, and execute for the users.

What is soft link and hard link in Linux



Soft Link:
We have a file shortcut feature in windows that is used to create a shortcut for files. A soft link is similar to a file shortcut. Each soft-linked file has its own Inode value, which refers to the original file. Any changes to the information in one file are mirrored in the other. You can connect soft links across different file systems, but if the source file is removed or transferred, the soft-linked file will not function properly. This link is called a hanging link. Removing a soft link has no effect, but If you remove the original file, it may cause the link will stop working.

Hard Link: A hard link in Linux is equivalent to a file saved on a hard drive – and it really refers to or links to a location on a hard drive. A hard link is essentially a mirror image of the original file. The difference between a hard link and a soft link is that removing the source file has no effect on a hard link but makes a soft link unworkable. So the most significant benefit of making a hard link is that you can still access the contents of the file even if you unintentionally erase it.

Creating Hard Link:

$ln file1.txt hardlink.txt : Create hardlink between original file file1.txt and hardlink.txt
$ls -l : You can see both the files with l in starting mentioning is linked file
$ vi hardlink.txt : Open hardlink file and add some text in it
$ cat file1.txt: You can see the same content in file1 as we added in hardlink.txt
$ rm file1.txt: Remove the file1.txt and you will see still hardlink.txt still have the data in it

Creating Soft Link:

$ ln -s file1.txt softlink.txt: Create softlink between file1.txt and softlink.txt files
$ cat softlink.txt: you can see the content of file1.txt is present in softlink.txt
$ rm file1.txt: After you remove the file1.txt, softlink.txt will not work.