

File system in Linux:

1. ls

The ls command is used in Linux/Unix systems to list directory contents. It's a fundamental command and has various options to display information in different formats. Here are some of the common options used with the ls command:

Basic Usage:

ls: Simply lists the names of files and directories in the current working directory.

Options:

Long Format:

- **-l:** Displays detailed information about files and directories in long format, showing permissions, owner, group, size, modification time, and name.
- **-a:** Shows all files, including hidden files (those starting with a dot .).
- **-h:** Prints file sizes in a human-readable format (e.g., KB, MB, GB).

Sorting:

- **-t:** Sorts by modification time, with the newest files first.
- **-r:** Reverses the sorting order.
- **-S:** Sorts by file size.
- **-X:** Sorts by extension.

Directories:

-d: Lists directories themselves, not their contents.

-R: Lists subdirectories recursively.

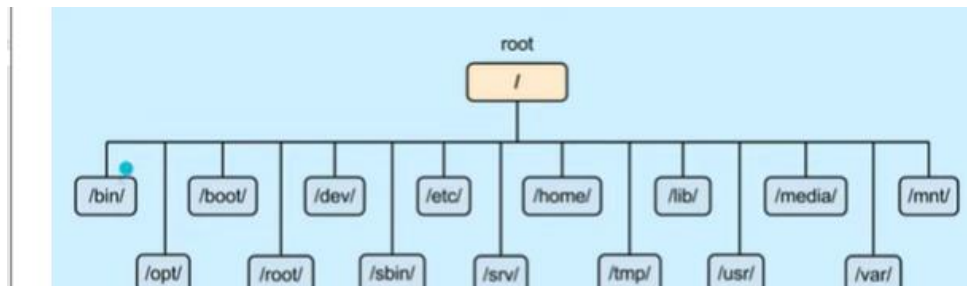
```
total 2744404
drwxr-xr-x 14 root root      4096 Aug  8 04:22 usr
drwxr-xr-x  2 root root      4096 Aug  8 04:22 srv
drwxr-xr-x  2 root root      4096 Aug  8 04:22 mnt
drwxr-xr-x  2 root root      4096 Aug  8 04:22 media
drwxr-xr-x 14 root root      4096 Aug  8 04:28 var
drwxr-xr-x 13 root root      4096 Aug  8 04:29 snap
drwx----- 2 root root    16384 Oct 30 12:47 lost+found
-rw----- 1 root root 2810183680 Oct 30 12:47 swapfile
lrwxrwxrwx 1 root root      10 Oct 30 12:47 libx32 -> usr/libx32
lrwxrwxrwx 1 root root       9 Oct 30 12:47 lib64 -> usr/lib64
lrwxrwxrwx 1 root root       9 Oct 30 12:47 lib32 -> usr/lib32
lrwxrwxrwx 1 root root       7 Oct 30 12:47 lib -> usr/lib
lrwxrwxrwx 1 root root       7 Oct 30 12:47 bin -> usr/bin
lrwxrwxrwx 1 root root       8 Oct 30 12:47/sbin -> usr/sbin
drwxrwxr-x 2 root root      4096 Oct 30 12:54 cdrom
drwxr-xr-x 3 root root      4096 Oct 30 12:57 home
drwxr-xr-x 3 root root      4096 Oct 30 13:19 opt
drwx----- 5 root root      4096 Oct 30 13:49 root
drwxr-xr-x 3 root root      4096 Oct 30 14:00 timeshift
drwxr-xr-x 131 root root    12288 Oct 31 13:39 etc
drwxr-xr-x  4 root root      4096 Oct 31 13:40 boot
dr-xr-xr-x 13 root root       0 Oct 31 14:25 sys
dr-xr-xr-x 260 root root       0 Oct 31 14:25 proc
drwxr-xr-x 19 root root     4200 Oct 31 14:25 dev
drwxr-xr-x 34 root root      920 Oct 31 14:25 run
drwxrwxrwt 22 root root      4096 Oct 31 14:54 tmp
root@Ubuntu:/# ls -ltr
usr media lost+found lib64 bin home timeshift etc proc tmp
srv var swapfile lib32/sbin opt .. boot dev
mnt snap libx32 lib cdrom root . sys run
root@Ubuntu:/#
```

- In Linux even when we use ls commands this is also a file here

```
root@Ubuntu: /  
root@Ubuntu:/# whereis ls  
ls: /usr/bin/ls /usr/share/man/man1/ls.1.gz  
root@Ubuntu:/#
```

Types of File system in Linux:

- Ext3
- Ext4
- XFS
- FAT



File System Structure:

- Root Directory (/): The top-level directory in the Linux file system.
- /bin: Contains essential system binaries (commands) accessible to all users.
- /etc: Configuration files for the system and installed applications.
- /home: Home directories for regular users.
- /var: Variable files like logs, temporary files, spool, mail, etc.
- /usr: User-related programs and files.
- /tmp: Temporary files.

```

sr0
root@Ubuntu:/# cd /dev
root@Ubuntu:/dev# ls
autofs          loop14      sr0         tty36       tty50       urandom
block           loop2       stderr      tty37       tty51       userfaultfd
bkg            loop3       stdin       tty38       tty510      userio
btrfs-control  loop4       stdout      tty39       tty511      vboxguest
bus            loop5       tty         tty4        tty512      vboxuser
cdrom          loop6       tty0        tty40       tty513      vcs
char           loop7       tty1        tty41       tty514      vcs1
console        loop8       tty10       tty42       tty515      vcs2
core           loop9       tty11       tty43       tty516      vcs3
cpu            loop-control tty12       tty44       tty517      vcs4
cpu_dma_latency mapper      tty13       tty45       tty518      vcs5
cuse           mcelog     tty14       tty46       tty519      vcs6
disk           men         tty15       tty47       tty52       vcsa
dma_heap       mqueue     tty16       tty48       tty520      vcsa1
dri            net         tty17       tty49       tty521      vcsa2
ecryptfs       null        tty18       tty5        tty522      vcsa3
fbs            nvram      tty19       tty50       tty523      vcsa4
fd             port        tty2        tty51       tty524      vcsa5
full           ppp         tty20       tty52       tty525      vcsa6
fuse           psaux      tty21       tty53       tty526      vcsu
hidraw0        ptmx       tty22       tty54       tty527      vcsu1
hpet           pts         tty23       tty55       tty528      vcsu2
hugepages      random      tty24       tty56       tty529      vcsu3
hwng           rfkill      tty25       tty57       tty53       vcsu4
i2c-0          rtc         tty26       tty58       tty530      vcsu5
initctl        rtc0        tty27       tty59       tty531      vcsu6
input          sda         tty28       tty6        tty54       vfio
kmsg           sda1        tty29       tty60       tty55       vga_arbiter

root@Ubuntu:/etc# cd /etc
root@Ubuntu:/etc# ls
acpi             hostname      printcap
adduser.conf     hosts         profile
alsa             hosts.allow  profile.d
alternatives     hosts.deny   protocols
anacrontab       hp            pulse
apg.conf         ifplugd      python3
apm              init          python3.10
apparmor         init.d        rc0.d
apparmor.d       initramfs-tools rc1.d
appport          inputrc       rc2.d
appstream.conf   inserv.conf.d rc3.d
apt              ipmi-usb      rc4.d
avahi            iproute2      rc5.d
bash.bashrc      issue         rc6.d
bash_completion issue.net      rc5.d
bash_completion.d kernel         resolv.conf
bindresvport.blacklist kernel-img.conf rmt
binfmt.d         kernelloops.conf rpc
bluetooth        ldap          rsyslog.conf
brlapi.key       ld.so.cache  rsyslog.d
brltty           ld.so.conf   rygel.conf
brltty.conf      ld.so.conf.d sane.d
ca-certificates  legal        security
ca-certificates.conf libao.conf   selinux
ca-certificates.conf.dpkg-old libaudit.conf sensors3.conf
chatscripts      libblockdev  sensors.d
console-setup    libl3        services
cracklib          libpaper.d   sgml
cron.d            libreoffice  shadow
cron.daily        locale.alias shadow-

```

```

root@Ubuntu:/etc# cd ..
root@Ubuntu:/# cd /home
root@Ubuntu:/home# ls
simran
root@Ubuntu:/home# cd /simran
bash: cd: /simran: No such file or directory
root@Ubuntu:/home#

```

- Now check Root Folder

```

root@Ubuntu:~# cd ..
root@Ubuntu:/# cd root/
root@Ubuntu:/# ls -ltr
total 12
-r-xr-xr-x 1 root root 7820 Oct 30 13:17 vboxpostinstall.sh
drwx----- 5 root root 4096 Oct 30 13:22 snap
root@Ubuntu:~#

```

Commands:

\$: represents Normal user

#: represents root user

- To check the host name: hostname
- To check the user logged in right now: whoami
- To check IP : ip addr
- To know the present working directory: pwd

```

simran@Ubuntu: ~
simran@Ubuntu:~$ hostname
Ubuntu
simran@Ubuntu:~$ whoami
simran
simran@Ubuntu:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:3a:48:c2 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
        valid_lft 83196sec preferred_lft 83196sec
    inet6 fe80::b248:5ac0:fb9c:1231/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
simran@Ubuntu:~$ pwd
/home/simran

```

- To list the file and folders/directories: ls
- To create a folder/directory: mkdir folder1
- Switch inside the directory which you have created : cd folder1
- To see what's inside the folder: ls

```

simran@Ubuntu:~$ ls
Desktop  Downloads  Pictures  snap      Videos
Documents Music      Public    Templates
simran@Ubuntu:~$ mkdir folder1
simran@Ubuntu:~$ ls
Desktop  Downloads  Music      Public  Templates
Documents folder1    Pictures  snap    Videos
simran@Ubuntu:~$ cd folder1
simran@Ubuntu:~/folder1$ pwd
/home/simran/folder1
simran@Ubuntu:~/folder1$

```

- Now let's create another folder inside this directory.

Mkdir folder2

```

simran@Ubuntu:~/folder1$ mkdir folder2
simran@Ubuntu:~/folder1$ ls
folder2
simran@Ubuntu:~/folder1$ pwd
/home/simran/folder1
simran@Ubuntu:~/folder1$

```

- Now lets get outside of this folder: cd ..
- If directly wants to go in folder2: cd folder1/folder2
- Want to come outside directly to home location: cd

```

simran@Ubuntu:~/folder1$ cd ..
simran@Ubuntu:~$ pwd
/home/simran
simran@Ubuntu:~$ cd folder1/folder2
simran@Ubuntu:~/folder1/folder2$ pwd
/home/simran/folder1/folder2
simran@Ubuntu:~/folder1/folder2$ cd
simran@Ubuntu:~$

```

- To find a folder if multiple folders are there: find folder1/ -name folder2
- Now if there are directories with different names present in folder1

```

/home/simran/folder1/folder2
simran@Ubuntu:~/folder1/folder2$ cd
simran@Ubuntu:~$ find folder1/ -name folder2
folder1/folder2
simran@Ubuntu:~$ cd folder1
simran@Ubuntu:~/folder1$ ls
folder2
simran@Ubuntu:~/folder1$ mkdir media
simran@Ubuntu:~/folder1$ mkdir folder3
simran@Ubuntu:~/folder1$ mkdir folder4
simran@Ubuntu:~/folder1$ ls
folder2 folder3 folder4 media
simran@Ubuntu:~/folder1$ ls folder*
folder2:

folder3:

folder4:
simran@Ubuntu:~/folder1$

```

- Creating a file: touch file1
- Now wants to find this file then,

```

simran@Ubuntu:~/folder1$ cd folder2
simran@Ubuntu:~/folder1/folder2$ touch file1
simran@Ubuntu:~/folder1/folder2$ cd ..
simran@Ubuntu:~/folder1$ cd..
cd..: command not found
simran@Ubuntu:~/folder1$ cd ..
simran@Ubuntu:~$ find folder1/ -name file1
folder1/folder2/file1
simran@Ubuntu:~$

```

- How to delete a folder/directory: rmdir folder2
- Delete a directory with all of its files: rm -r media

```

simran@Ubuntu:~$ cd folder1
simran@Ubuntu:~/folder1$ ls
folder2 folder3 folder4 media
simran@Ubuntu:~/folder1$ rmdir folder2
rmdir: failed to remove 'folder2': Directory not empty
simran@Ubuntu:~/folder1$ rm -r folder2
simran@Ubuntu:~/folder1$ ls
folder3 folder4 media
simran@Ubuntu:~/folder1$ rmdir media
simran@Ubuntu:~/folder1$ ls
folder3 folder4
simran@Ubuntu:~/folder1$

```

- To list file in order of modification dates: ls -lt
- To see reverse order of modification date: ls -ltr
- To edit the file use VIM editor
- Command: vi file2
 - Insert mode: press 'i'
 - Exit Mode: esc+ :wq
- See whats inside the file file2: cat file2

```

simran@Ubuntu:~/folder1$ vi file2
simran@Ubuntu:~/folder1$ cat file2
Hello there
simran@Ubuntu:~/folder1$

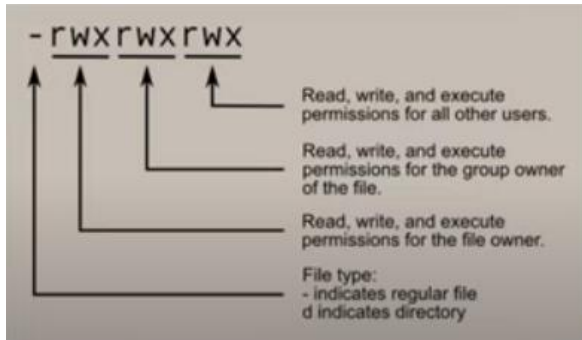
```

- To see full details: ls -ltr

```

simran@Ubuntu:~/folder1$ ls -ltr
total 12
drwxrwxr-x 2 simran simran 4096 Oct 31 15:30 folder3
drwxrwxr-x 2 simran simran 4096 Oct 31 15:30 folder4
-rw-rw-r-- 1 simran simran  12 Oct 31 15:41 file2
simran@Ubuntu:~/folder1$

```



Practice commands

Switch to root user and where as an Other user we created the file there create the another file but now as an root user

```
root@Ubuntu:/home/simran/folder18# ls
audio folder1 folder2 folder3 media
root@Ubuntu:/home/simran/folder18# cd folder1
root@Ubuntu:/home/simran/folder18/folder1# ls
file18
root@Ubuntu:/home/simran/folder18/folder1# touch file2
root@Ubuntu:/home/simran/folder18/folder1# ls -l
total 0
-rw-rw-r-- 1 simran simran 0 Oct 31 17:51 file18
-rw-r--r-- 1 root   root   0 Nov  1 14:20 file2
root@Ubuntu:/home/simran/folder18/folder1# cd
root@Ubuntu:~# exit
exit
simran@Ubuntu:~$
```

Now exit from root user and as an other user try to delete that file2 created by Root user. It will show permission denied.

So file created by root user, we can't do any change in that file

```
simran@Ubuntu:~/folder18/folder1$ ls -l
total 0
-rw-rw-r-- 1 simran simran 0 Oct 31 17:51 file18
-rw-r--r-- 1 root   root   0 Nov  1 14:20 file2
simran@Ubuntu:~/folder18/folder1$ rm file2
rm: remove write-protected regular empty file 'file2'?
simran@Ubuntu:~/folder18/folder1$ vi file2
simran@Ubuntu:~/folder18/folder1$
```

To see the file content we can use cat command or less command

Cat file18 or less file18

```
simran@Ubuntu:~/folder18/folder1$ cat file18
hello There!
simran@Ubuntu:~/folder18/folder1$ less file18
simran@Ubuntu:~/folder18/folder1$
simran@Ubuntu:~/folder18/folder1$ less file18
simran@Ubuntu:~/folder18/folder1$
simran@Ubuntu:~/folder18/folder1$
```

Use of word count commands: wc file18

1 is for 1 line is present

2 is for number of words

13 is for number of total characters

```
simran@Ubuntu:~/folder18/folder1$ wc file18
 1  2 13 file18
simran@Ubuntu:~/folder18/folder1$
```

To compare the file we use diff command

Diff file18 file181

```
1  2 13 file18
simran@Ubuntu:~/folder18/folder1$ touch file181
simran@Ubuntu:~/folder18/folder1$ vi file181
simran@Ubuntu:~/folder18/folder1$ diff file18 file181
1c1
< hello There!
---
> Hello There What a beautiful day it is!
simran@Ubuntu:~/folder18/folder1$
```

Now how to compress and decompress the file(like zip or unzip)

Tar cvf files.tar file18 file181

Ls

```
simran@Ubuntu:~/folder18/folder1$ tar cvf files.tar file18 file181
file18
file181
simran@Ubuntu:~/folder18/folder1$ ls
file18  file181  file2  files.tar
simran@Ubuntu:~/folder18/folder1$
```

The command tar cvf specifically means:

- c: Create a new archive. This flag tells tar to create an archive file.
- v: Verbose mode. This flag provides detailed information about the files being processed, displaying the names of the files as they are added to the archive.
- f: File. This flag specifies the filename of the archive. The f flag should be followed by the name of the archive file.

Now compress this file

Gzip files.tar

```

simran@Ubuntu:~/folder18/folder1$ ls
file18 file181 file2 files.tar
simran@Ubuntu:~/folder18/folder1$ gzip files.tar
simran@Ubuntu:~/folder18/folder1$ ls
file18 file181 file2 files.tar.gz
simran@Ubuntu:~/folder18/folder1$ ls -ltr
total 12
-rw-r--r-- 1 root root 0 Nov 1 14:20 file2
-rw-rw-r-- 1 simran simran 13 Nov 1 14:27 file18
-rw-rw-r-- 1 simran simran 40 Nov 1 14:32 file181
-rw-rw-r-- 1 simran simran 198 Nov 1 14:34 files.tar.gz
simran@Ubuntu:~/folder18/folder1$

```

Now if we want to move this zip file from folder1 to folder5 which is present inside folder1

Cp files.tar.gz folder5/ (destination_folder)

```

simran@Ubuntu:~/folder18/folder1$ ls
file18 file181 file2 files.tar.gz folder18
simran@Ubuntu:~/folder18/folder1$ mkdir folder5
simran@Ubuntu:~/folder18/folder1$ cp files.tar.gz folder5/
simran@Ubuntu:~/folder18/folder1$ cd folder5
simran@Ubuntu:~/folder18/folder1/folder5$ ls
files.tar.gz
simran@Ubuntu:~/folder18/folder1/folder5$

```

Unzip this file now

Gunzip files.tar.gz

Untar the file

Tar xvf files.tar

```

files.tar.gz
simran@Ubuntu:~/folder18/folder1/folder5$ gunzip files.tar.gz
simran@Ubuntu:~/folder18/folder1/folder5$ ls
files.tar
simran@Ubuntu:~/folder18/folder1/folder5$ tar xvf files.tar
file18
file181
simran@Ubuntu:~/folder18/folder1/folder5$ ls -ltr
total 20
-rw-rw-r-- 1 simran simran 13 Nov 1 14:27 file18
-rw-rw-r-- 1 simran simran 40 Nov 1 14:32 file181
-rw-rw-r-- 1 simran simran 10240 Nov 1 14:40 files.tar
simran@Ubuntu:~/folder18/folder1/folder5$

```

To rename the files

mv file18 file17 (move the file18 to file17 and delete the file18 now)

```

-rw-rw-r-- 1 simran simran 10240 Nov 1 14:40 files.tar
simran@Ubuntu:~/folder18/folder1/folder5$ mv file18 file17
simran@Ubuntu:~/folder18/folder1/folder5$ ls
file17 file181 files.tar
simran@Ubuntu:~/folder18/folder1/folder5$

```

To add the content of 2 files into third file

Cat file17 file181 > filea

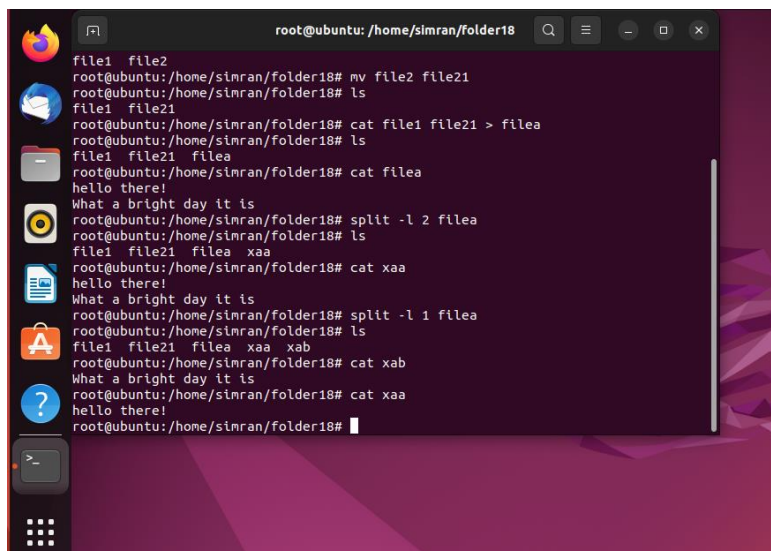

```

simran@Ubuntu:~/folder18/folder1/folder3$ cat file17
hello There!
simran@Ubuntu:~/folder18/folder1/folder3$ cat file181
Hello There What a beautiful day it is!
simran@Ubuntu:~/folder18/folder1/folder3$ cat file17 cat181 > filea
cat: cat181: No such file or directory
simran@Ubuntu:~/folder18/folder1/folder3$ cat file17 file181 > filea
simran@Ubuntu:~/folder18/folder1/folder3$ ls
file17  file181  filea  files.tar
simran@Ubuntu:~/folder18/folder1/folder3$ cat file a
cat: file: No such file or directory
cat: a: No such file or directory
simran@Ubuntu:~/folder18/folder1/folder3$ cat filea
hello There!
Hello There What a beautiful day it is!
simran@Ubuntu:~/folder18/folder1/folder3$

```

To split the file into two parts

Split -l 1 filea (1 representd number of lines want to split)



```

root@ubuntu: /home/simran/folder18
file1  file2
root@ubuntu:/home/simran/folder18# mv file2 file21
root@ubuntu:/home/simran/folder18# ls
file1  file21
root@ubuntu:/home/simran/folder18# cat file1 file21 > filea
root@ubuntu:/home/simran/folder18# ls
file1  file21  filea
root@ubuntu:/home/simran/folder18# cat filea
hello there!
What a bright day it is
root@ubuntu:/home/simran/folder18# split -l 2 filea
root@ubuntu:/home/simran/folder18# ls
file1  file21  filea  xaa
root@ubuntu:/home/simran/folder18# cat xaa
hello there!
What a bright day it is
root@ubuntu:/home/simran/folder18# split -l 1 filea
root@ubuntu:/home/simran/folder18# ls
file1  file21  filea  xaa  xab
root@ubuntu:/home/simran/folder18# cat xab
What a bright day it is
root@ubuntu:/home/simran/folder18# cat xaa
hello there!
root@ubuntu:/home/simran/folder18#

```

Usage of RM command:

The `rm` command in Linux is used to remove files and directories. It's a powerful command, so be very careful when using it as deleted files are usually not recoverable. Here are several ways to use the `rm` command:

1. Remove a Single File

To delete a single file:

...

rm filename

...

For example, to delete a file named `example.txt`:

```
'''  
rm example.txt  
'''
```

2. Remove Multiple Files

You can remove multiple files at once:

```
'''  
rm file1 file2 file3  
'''
```

For example, to delete files named `file1.txt`, `file2.txt`, and `file3.txt`:

```
'''  
rm file1.txt file2.txt file3.txt  
'''
```

3. Remove Files Using Wildcards

Wildcards can be used to match multiple files based on a pattern:

```
'''  
rm *.txt  
'''
```

This command will remove all files with a `.txt` extension in the current directory.

4. Remove a Directory and Its Contents

To remove a directory and all its contents (including files and subdirectories):

'''

rm -r directory_name

'''

For example, to delete a directory named `folder` and all its contents:

'''

rm -r folder

'''

5. Prompt Before Removal

To be prompted before each file is removed, use the `-i` flag:

'''

rm -i filename

'''

For example:

'''

rm -i example.txt

'''

This will prompt for confirmation before deleting `example.txt`.

6. Force Removal (No Confirmation)

To force deletion without confirmation, use the `-f` flag:

'''

rm -f filename

'''

For example:

```
'''
```

```
rm -f file.txt
```

```
'''
```

This will delete `file.txt` without confirmation.

7. Verbose Output

To show a more detailed output while deleting files, use the `-v` flag:

```
'''
```

```
rm -v filename
```

```
'''
```

For example:

```
'''
```

```
rm -v example.txt
```

```
'''
```

This will display the process of removing `example.txt`.

#CAT Command:

The cat command (short for concatenate) is used to display the contents of one or multiple files, creating, or combining files.

Usage:

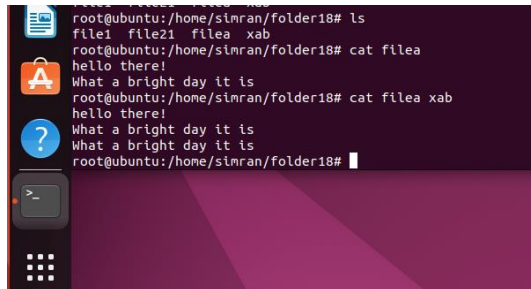
```
cat filea
```

This command displays the contents of the file named filename in the terminal.

bash

cat filea xab

This command displays the contents of both file1 and file2 in the order listed.

A terminal window screenshot with a dark purple background. The prompt is 'root@ubuntu:/hone/simran/folder18#'. The user enters 'ls', showing 'file1 file21 filea xab'. Then they enter 'cat filea', showing 'hello there!'. Finally, they enter 'cat filea xab', showing 'hello there!' followed by 'What a bright day it is' on a new line. The prompt is now 'root@ubuntu:/hone/simran/folder18#'.

```
root@ubuntu:/hone/simran/folder18# ls
file1 file21 filea xab
root@ubuntu:/hone/simran/folder18# cat filea
hello there!
What a bright day it is
root@ubuntu:/hone/simran/folder18# cat filea xab
hello there!
What a bright day it is
What a bright day it is
root@ubuntu:/hone/simran/folder18#
```

2. less Command

The less command is a **terminal pager** that allows you to view and navigate through the contents of a file.

Usage:

less filename

This command opens filename in a paginated view, allowing you to scroll up and down through the content.

3. more Command

Similar to less, the more command is a **terminal pager** used to display the contents of a file one screen at a time.

Usage:

more filename

This command displays the content of the file filename one page at a time. Press the spacebar to view the next page and press q to exit.

4. head Command

The head command is used to display the beginning lines of a file.

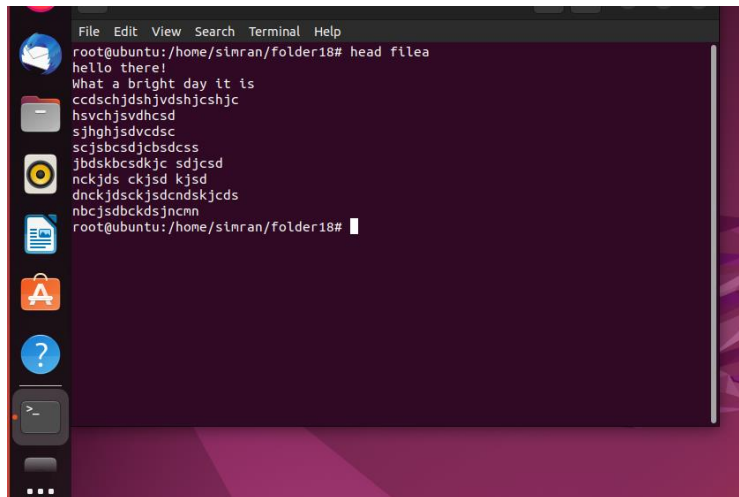
Usage:

head filename

This command shows the first ten lines of the file filename by default. You can also specify the number of lines to display using the -n option. For example:

head -n 20 filename

This will display the first 20 lines of the file.

A terminal window on a Linux desktop. The terminal title bar shows 'File Edit View Search Terminal Help'. The prompt is 'root@ubuntu:/home/simran/folder18#'. The command 'head filea' has been entered. The output displays the first 20 lines of the file 'filea'. The first line is 'hello there!'. The subsequent lines contain random alphanumeric strings. The prompt returns to 'root@ubuntu:/home/simran/folder18#'.

```
File Edit View Search Terminal Help
root@ubuntu:/home/simran/folder18# head filea
hello there!
What a bright day it is
ccdschjdshjvdshjcs hjc
hsvchjvdhcsd
sjhghjdvcdsc
scjsbcsdjcbdcss
jbdsbcsdkjc sdjcsd
nckjds ckjds kjds
dnckjdsckjdsdcndskjcds
nbcjsdbckdsjncm
root@ubuntu:/home/simran/folder18#
```

5. tail Command

The tail command is used to display the ending lines of a file.

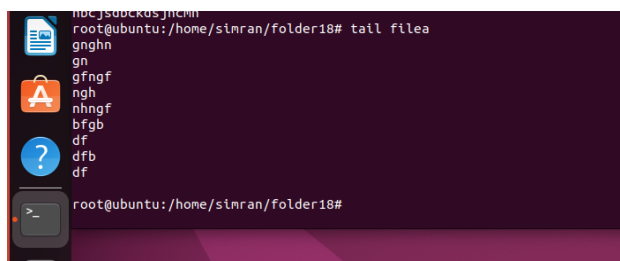
Usage:

tail filename

This command displays the last ten lines of the file filename by default. Similar to head, you can specify the number of lines to display using the -n option. For example:

tail -n 15 filename

This will display the last 15 lines of the file.

A terminal window on a Linux desktop. The terminal title bar shows 'File Edit View Search Terminal Help'. The prompt is 'root@ubuntu:/home/simran/folder18#'. The command 'tail filea' has been entered. The output displays the last 10 lines of the file 'filea'. The lines contain random alphanumeric strings. The prompt returns to 'root@ubuntu:/home/simran/folder18#'.

```
File Edit View Search Terminal Help
root@ubuntu:/home/simran/folder18# tail filea
nbcjsdbckdsjncm
gnghn
gn
gfngf
ngh
nhngf
bfgb
df
dfb
df
root@ubuntu:/home/simran/folder18#
```

Remember, these commands are used to view or display the content of files in the terminal. They don't modify the files themselves. They are useful for reading or quickly viewing the content of files directly within the terminal.

Permissions in FileSystem:

Ubuntu (and other Linux-based systems), the `chmod` command is used to change the permissions of files and directories. It allows users to modify the read, write, and execute permissions for the owner, group, and others. Here's a detailed breakdown of how to use `chmod`:

Basic Permissions

Each file and directory in Linux has permissions represented by three categories: owner, group, and others. These permissions can be set as **read (r), write (w), and execute (x)**. These permissions are represented numerically as well: **read (4), write (2), and execute (1)**.

chmod Syntax

The `chmod` command follows this syntax:

chmod who=permissions file

who: Represents who the permission change will apply to and can be:

u (user/owner)

g (group)

o (others)

a (all/ugo - user, group, others)

permissions: Represents the permissions to be set, using combinations of:

r (read) 4

w (write) 2

x (execute) 1

file: Specifies the file or directory whose permissions are to be changed.

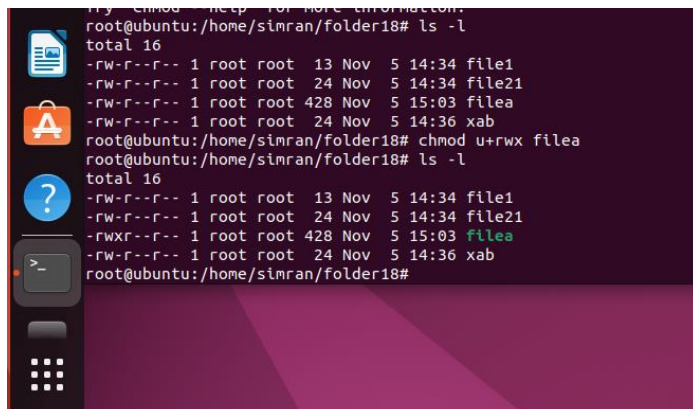
Usage Examples:

Changing File Permissions

Symbolic Mode:

To give the owner of the file read, write and execute permissions:

Chmod u+rw filea

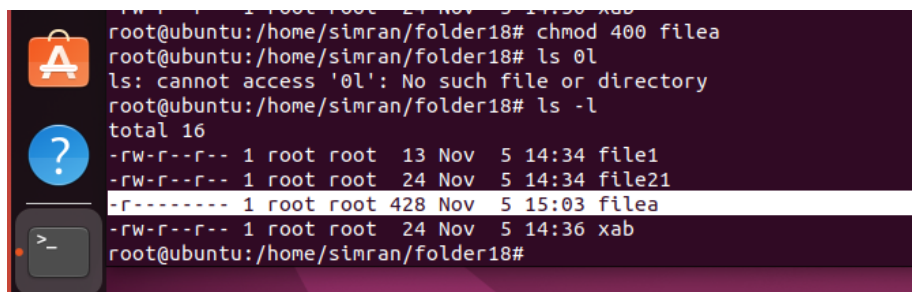


```
root@ubuntu:/home/simran/folder18# ls -l
total 16
-rw-r--r-- 1 root root 13 Nov 5 14:34 file1
-rw-r--r-- 1 root root 24 Nov 5 14:34 file21
-rw-r--r-- 1 root root 428 Nov 5 15:03 filea
-rw-r--r-- 1 root root 24 Nov 5 14:36 xab
root@ubuntu:/home/simran/folder18# chmod u+rw filea
root@ubuntu:/home/simran/folder18# ls -l
total 16
-rw-r--r-- 1 root root 13 Nov 5 14:34 file1
-rw-r--r-- 1 root root 24 Nov 5 14:34 file21
-rwxr--r-- 1 root root 428 Nov 5 15:03 filea
-rw-r--r-- 1 root root 24 Nov 5 14:36 xab
root@ubuntu:/home/simran/folder18#
```

Numeric Mode:

To give the owner only read permission:

Chmod 400 filea



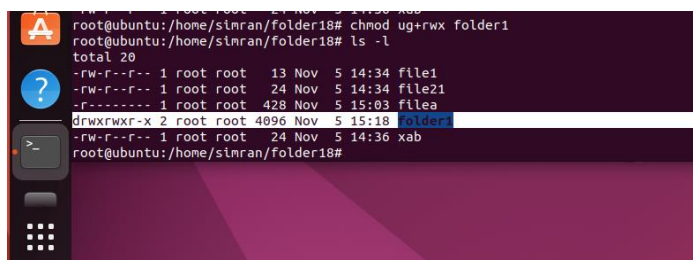
```
root@ubuntu:/home/simran/folder18# chmod 400 filea
root@ubuntu:/home/simran/folder18# ls -l
ls: cannot access '0l': No such file or directory
root@ubuntu:/home/simran/folder18# ls -l
total 16
-rw-r--r-- 1 root root 13 Nov 5 14:34 file1
-rw-r--r-- 1 root root 24 Nov 5 14:34 file21
-r----- 1 root root 428 Nov 5 15:03 filea
-rw-r--r-- 1 root root 24 Nov 5 14:36 xab
root@ubuntu:/home/simran/folder18#
```

Changing Directory Permissions

Symbolic Mode:

To give the owner and group read, write, and execute permissions:

Chmod ug+rw folder1

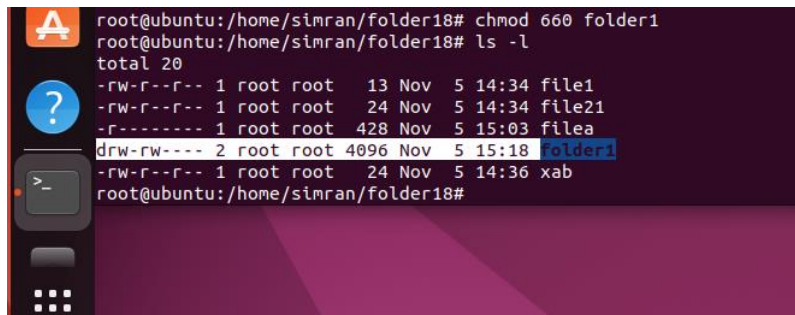


```
root@ubuntu:/home/simran/folder18# chmod ug+rw folder1
root@ubuntu:/home/simran/folder18# ls -l
total 20
-rw-r--r-- 1 root root 13 Nov 5 14:34 file1
-rw-r--r-- 1 root root 24 Nov 5 14:34 file21
-r----- 1 root root 428 Nov 5 15:03 filea
drwxrwxr-x 2 root root 4096 Nov 5 15:18 folder1
-rw-r--r-- 1 root root 24 Nov 5 14:36 xab
root@ubuntu:/home/simran/folder18#
```


Numeric Mode:

To give the owner and group read and write permissions:

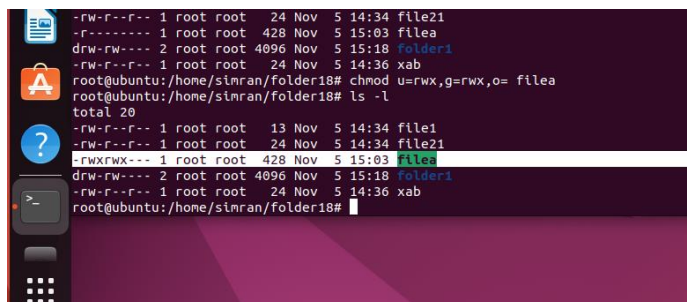
Chmod 660 folder1



```
root@ubuntu:/home/simran/folder18# chmod 660 folder1
root@ubuntu:/home/simran/folder18# ls -l
total 20
-rw-r--r-- 1 root root 13 Nov 5 14:34 file1
-rw-r--r-- 1 root root 24 Nov 5 14:34 file21
-r----- 1 root root 428 Nov 5 15:03 filea
drw-rw---- 2 root root 4096 Nov 5 15:18 folder1
-rw-r--r-- 1 root root 24 Nov 5 14:36 xab
root@ubuntu:/home/simran/folder18#
```

Permission Combination Examples

Symbolic Representation:



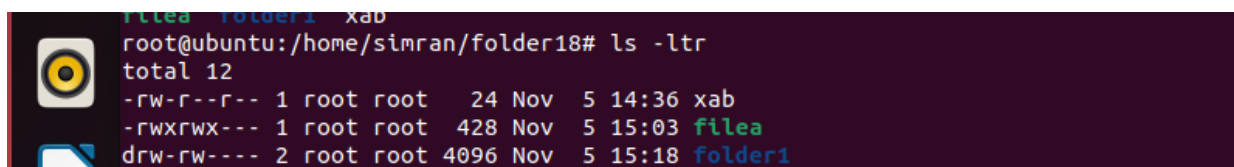
```
root@ubuntu:/home/simran/folder18# chmod u=rwx,g=rwx,o= filea
root@ubuntu:/home/simran/folder18# ls -l
total 20
-rw-r--r-- 1 root root 13 Nov 5 14:34 file1
-rw-r--r-- 1 root root 24 Nov 5 14:34 file21
-rwxrwx--- 1 root root 428 Nov 5 15:03 filea
drw-rw---- 2 root root 4096 Nov 5 15:18 folder1
-rw-r--r-- 1 root root 24 Nov 5 14:36 xab
root@ubuntu:/home/simran/folder18#
```

The **chmod** command often requires administrative permissions (use sudo or be logged in as the owner or with relevant permissions)

CHOWN Command: (Change ownership)

Chown owner:group filename


Chown simran:simran filea



```
root@ubuntu:/home/simran/folder18# ls -ltr
total 12
-rw-r--r-- 1 root root 24 Nov 5 14:36 xab
-rwxrwx--- 1 root root 428 Nov 5 15:03 filea
drw-rw---- 2 root root 4096 Nov 5 15:18 folder1
```

```
root@ubuntu:/home/simran/folder18# chown simran:simran filea
root@ubuntu:/home/simran/folder18# ls -ltr
total 12
-rwxrwx--- 1 simran simran 428 Nov 5 15:03 filea
drw-rw---- 2 root root 4096 Nov 5 15:18 folder1
root@ubuntu:/home/simran/folder18#
```

Linux File Types:

File Symbol	File Type
-	Regular file
d	Directory
l	Link
c	Device File
s	Socket
p 	FIFO or Named Pipe
b	Block device

In Linux, files are categorized into several types based on their nature and purpose. Here are some common file types:

1. Regular Files (Regular Text Files):

- These are the most common type of files.
- They contain data, such as text files, binary files, and program executables.

...

\$ ls -l

```
-rw-r--r-- 1 user user 12345 Nov 16 10:00 example.txt
```

```
...
```

2. Directories:

- Directories contain lists of file names and their corresponding inode numbers.
- They are analogous to folders in other operating systems.

```
...
```

```
$ ls -l
```

```
drwxr-xr-x 2 user user 4096 Nov 16 09:59 my_directory/
```

```
...
```

3. Symbolic Links (Soft Links):

- Symbolic links are shortcuts or pointers to another file or directory.
- They are similar to Windows shortcuts.

```
...
```

```
$ ls -l
```

```
lrwxrwxrwx 1 user user 8 Nov 16 10:02 link_to_file -> myfile
```

```
...
```

4. Block Special Files:

- These files provide buffered input/output for device drivers.
- They are often used for accessing devices like hard drives.

```
...
```

```
$ ls -l
```

```
brw-rw---- 1 root disk 8, 1 Nov 16 10:04 sda1
```

```
...
```

```
root@ubuntu:/home/simran# find / -type b
find: '/run/user/1000/doc': Permission denied
find: '/run/user/1000/gvfs': Permission denied
/run/systemd/inaccessible/blk
/dev/loop15
/dev/loop14
/dev/loop13
/dev/loop12
/dev/loop11
/dev/loop10
/dev/loop9
/dev/loop8
/dev/sda3
/dev/sda2
/dev/sda1
/dev/sda
/dev/sr0
/dev/loop7
/dev/loop6
/dev/loop5
/dev/loop4
/dev/loop3
/dev/loop2
/dev/loop1
/dev/loop0
root@ubuntu:/home/simran# ^C
root@ubuntu:/home/simran# ls -l /dev/sda1
brw-rw---- 1 root disk 8, 1 Nov 16 14:42 /dev/sda1
root@ubuntu:/home/simran#
```

5. Character Special Files:

- Similar to block special files, but provide unbuffered access.
- Often used for devices like terminals.
- Files that read/write data character by character

...

\$ ls -l

crw-rw-rw- 1 root tty 5, 1 Nov 16 10:06 tty1

...

```
/dev/vcsa
/dev/vcsu
/dev/vcs
/dev/tty0
/dev/console
/dev/tty
/dev/kmsg
/dev/urandom
/dev/random
/dev/full
/dev/zero
/dev/port
/dev/null
/dev/mem
/dev/vga_arbiter
/dev/rfkill
root@ubuntu:/home/simran# ^C
root@ubuntu:/home/simran# ls -l /dev/null
crw-rw-rw- 1 root root 1, 3 Nov 16 14:42 /dev/null
root@ubuntu:/home/simran#
```

6. FIFO (Named Pipes):

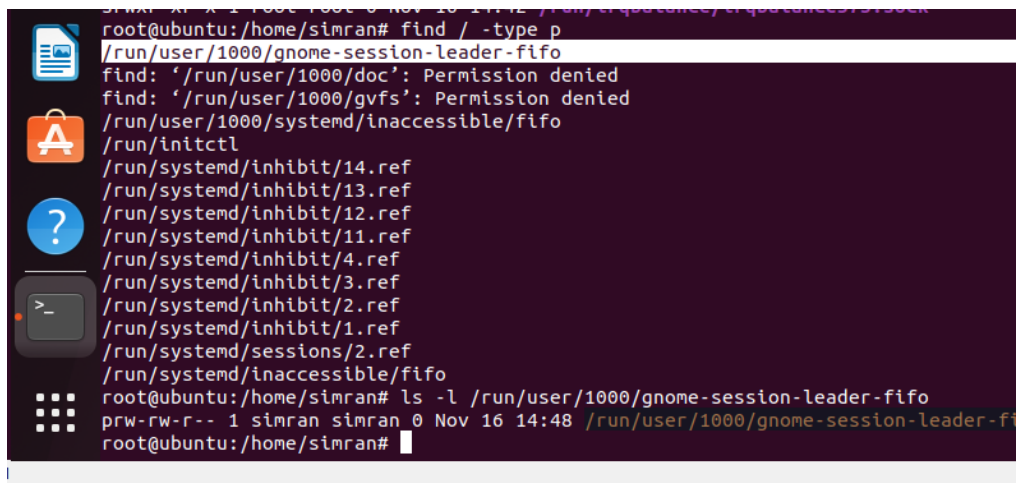
- FIFOs are special files used for communication between processes.
- They are named pipes.
- data getting transferred between processes in FIFO manner

...

```
$ ls -l
```

```
prw-r--r-- 1 user user 0 Nov 16 10:08 my_pipe
```

...



A terminal window screenshot showing the creation and listing of a named pipe. The user is root@ubuntu:/home/simran. The command `find / -type p` is executed, listing various system files including `/run/user/1000/gnome-session-leader-fifo`. The user then runs `ls -l /run/user/1000/gnome-session-leader-fifo`, showing the permissions `prw-rw-r-- 1 simran simran 0 Nov 16 14:48 /run/user/1000/gnome-session-leader-fifo`.

7. Sockets:

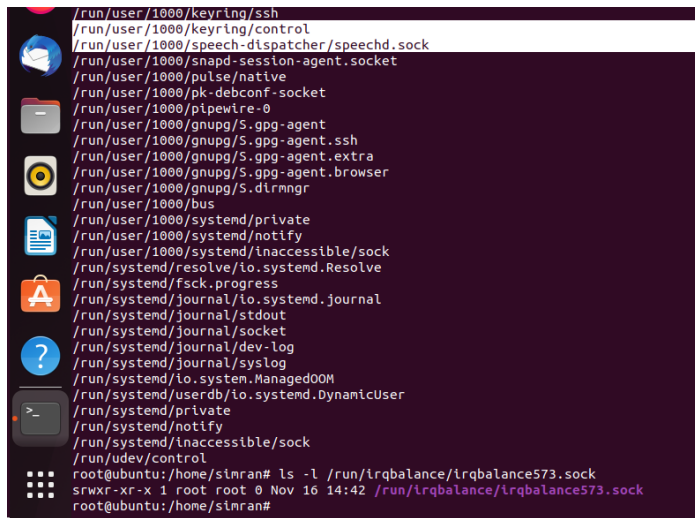
- Sockets are special files used for inter-process communication.
- They enable communication between processes on the same or different machines.

...

```
$ ls -l
```

```
srwxr-xr-x 1 user user 0 Nov 16 10:10 my_socket
```

...

A terminal window with a dark background and light text. On the left side, there is a vertical sidebar with several application icons: a blue circle with a white person icon, a folder icon, a yellow circle with a black dot, a blue document icon, an orange shopping bag icon, a blue question mark icon, and a terminal icon. The main area of the terminal displays a list of files and directories in the /run directory, including /run/user/1000/keyring/ssh, /run/user/1000/keyring/control, /run/user/1000/speech-dispatcher/speechd.sock, /run/user/1000/snapd-session-agent.socket, /run/user/1000/pulse/native, /run/user/1000/pk-debconf.socket, /run/user/1000/pipewire-0, /run/user/1000/gnupg/S.gpg-agent, /run/user/1000/gnupg/S.gpg-agent.ssh, /run/user/1000/gnupg/S.gpg-agent.extra, /run/user/1000/gnupg/S.gpg-agent.browser, /run/user/1000/gnupg/S.dirmngr, /run/user/1000/bus, /run/user/1000/systemd/private, /run/user/1000/systemd/notify, /run/user/1000/systemd/inaccessible.sock, /run/systemd/resolve/lo.systemd.Resolve, /run/systemd/fsck.progress, /run/systemd/journal/lo.systemd.journal, /run/systemd/journal/stdout, /run/systemd/journal/socket, /run/systemd/journal/dev-log, /run/systemd/journal/syslog, /run/systemd/lo.system.ManagedOOM, /run/systemd/userdb/lo.systemd.DynamicUser, /run/systemd/private, /run/systemd/notify, /run/systemd/inaccessible.sock, /run/udev/control, /run/lrqlbalance/lrqlbalance573.sock, and /run/lrqlbalance/lrqlbalance573.sock. The prompt root@ubuntu:/home/stnran# is visible at the bottom.

8. Device Files:

- Device files represent physical and virtual devices on the system.
- They are found in the `/dev`` directory.

...

```
$ ls -l /dev/sda
```

```
brw-rw---- 1 root disk 8, 0 Nov 16 10:12 /dev/sda
```

...

These are the main file types in Linux, and they serve various purposes in the system. The output of the ``ls -l`` command provides information about file permissions, ownership, size, and modification time.

#Link Files in Linux:

In Linux, a "link" typically refers to two types of links: symbolic links (soft links) and hard links.

1. Symbolic Links (Soft Links):

- A symbolic link is a separate file that acts as a pointer or reference to another file or directory.
- It is similar to a Windows shortcut.
- Symbolic links can span different file systems and can link to files or directories that may not exist.

Creating a symbolic link:

```
```bash
```

```
ln -s target_file link_name
```

```
```
```

Example:

```
```bash
```

```
ln -s /path/to/target/file /path/to/link
```

```
```
```

Example Output:

```
```bash
```

```
$ ls -l
```

```
lrwxrwxrwx 1 user user 8 Nov 16 10:02 link_to_file -> myfile
```

```
```
```

2. Hard Links:

- A hard link is a reference to the same inode (data and disk block location) as the original file.
- Changes to the data in one file are reflected in all hard links.
- Hard links cannot span different file systems, and they must be on the same file system as the original file.

Creating a hard link:

```
```bash
```

```
ln target_file link_name
```

```
```
```

Example:

```
```bash
```

```
ln /path/to/target/file /path/to/link
```

```

Example Output:

```bash

\$ ls -l

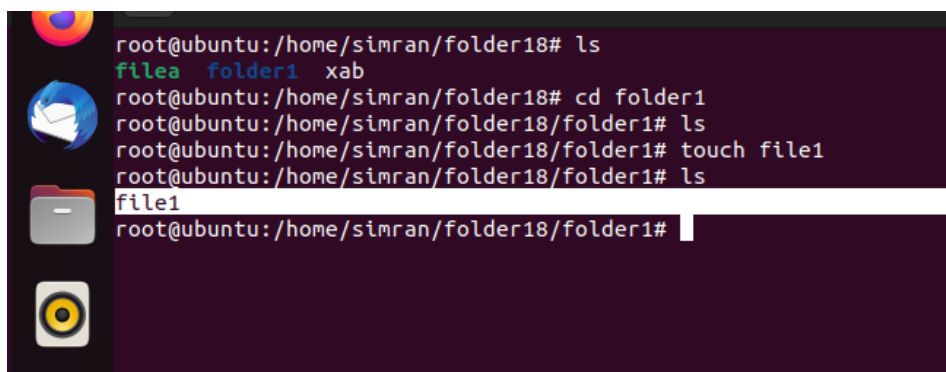
-rw-r--r-- 2 user user 12345 Nov 16 10:00 myfile

```

It's important to note that when the last hard link to a file is deleted, the data is not immediately deleted from the disk. It is only freed when there are no more references to the inode, which happens when all links to the file are removed.

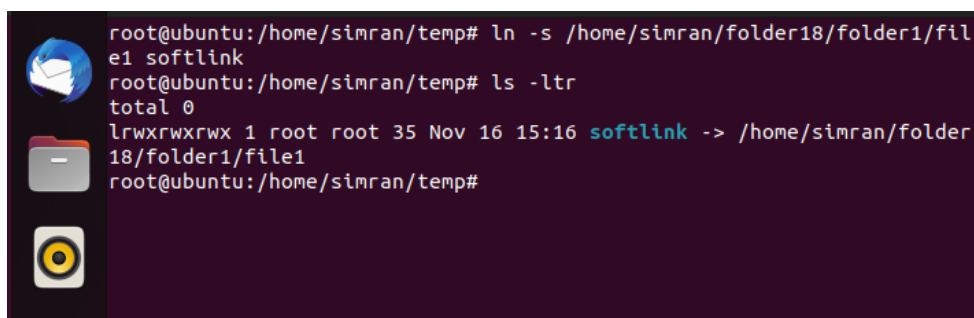
Practical:

Let's say we have a file1 inside a folder1

A terminal window with a dark purple background and Ubuntu icons on the left. The commands and output are as follows:

```
root@ubuntu:/home/simran/folder18# ls
filea folder1 xab
root@ubuntu:/home/simran/folder18# cd folder1
root@ubuntu:/home/simran/folder18/folder1# ls
root@ubuntu:/home/simran/folder18/folder1# touch file1
root@ubuntu:/home/simran/folder18/folder1# ls
file1
root@ubuntu:/home/simran/folder18/folder1#
```

- Now we want to create a soft link (shortcut) for this file1
- Inside a temp folder we will create a soft link

A terminal window with a dark purple background and Ubuntu icons on the left. The commands and output are as follows:

```
root@ubuntu:/home/simran/temp# ln -s /home/simran/folder18/folder1/file1 softlink
root@ubuntu:/home/simran/temp# ls -ltr
total 0
lrwxrwxrwx 1 root root 35 Nov 16 15:16 softlink -> /home/simran/folder18/folder1/file1
root@ubuntu:/home/simran/temp#
```

Now ill go to the folder1 where our file1 exists originally and make some changes in the file at that location.

You will see the change you are making by staying in the original location those changes are getting reflected in the softlink as well


```
root@ubuntu:/home/simran/temp# ln -s /home/simran/folder18/folder1/file1 softlink
root@ubuntu:/home/simran/temp# ls -ltr
total 0
lrwxrwxrwx 1 root root 35 Nov 16 15:16 softlink -> /home/simran/folder18/folder1/file1
root@ubuntu:/home/simran/temp# cat softlink
HelloThere1
root@ubuntu:/home/simran/temp#
```

```
root@ubuntu:/home/simran/folder18/folder1# ls
file1
root@ubuntu:/home/simran/folder18/folder1# touch file1
root@ubuntu:/home/simran/folder18/folder1# echo "HelloThere1" > file1
root@ubuntu:/home/simran/folder18/folder1# ls
file1
root@ubuntu:/home/simran/folder18/folder1# cat file1
HelloThere1
root@ubuntu:/home/simran/folder18/folder1#
```

Also the changes you make in shortcuts are visible in the file1 present in original location.

```
root@ubuntu:/home/simran/temp# ls
softlink
root@ubuntu:/home/simran/temp# echo "today" >> softlink
root@ubuntu:/home/simran/temp# cat softlink
HelloThere1
today
root@ubuntu:/home/simran/temp#
```

```
root@ubuntu:/home/simran/folder18/folder1# ls
file1
root@ubuntu:/home/simran/folder18/folder1# cat file1
HelloThere1
root@ubuntu:/home/simran/folder18/folder1# cat file1
HelloThere1
today
root@ubuntu:/home/simran/folder18/folder1#
```

If you delete the original file the softlink will also be deleted

HardLink:

First create a test File inside Folder1 TestFile

```
root@ubuntu:/home/simran/folder18/folder1# ls
file1 TestFile
root@ubuntu:/home/simran/folder18/folder1# cat Testfile
cat: Testfile: No such file or directory
root@ubuntu:/home/simran/folder18/folder1# cat TestFile
testing
root@ubuntu:/home/simran/folder18/folder1#
```

Now will create a hardlink

```
root@ubuntu:/home/simran/temp# ls
softlink
root@ubuntu:/home/simran/temp# ln /home/simran/folder18/folder1/TestFile HardLink
root@ubuntu:/home/simran/temp# ls
HardLink softlink
root@ubuntu:/home/simran/temp# ls -ltr
total 4
lrwxrwxrwx 1 root root 35 Nov 16 15:16 softlink -> /home/simran/folder18/folder1/file1
-rw-r--r-- 2 root root 8 Nov 16 15:26 HardLink
root@ubuntu:/home/simran/temp#
```

```
root@ubuntu:/home/simran/folder18/folder1# ls
file1 TestFile
root@ubuntu:/home/simran/folder18/folder1# cat TestFile
cat: Testfile: No such file or directory
root@ubuntu:/home/simran/folder18/folder1# cat TestFile
testing
root@ubuntu:/home/simran/folder18/folder1#
```

- Hardlink will appear to you as a normal file only
- Adding some content in the Test File present in original location
- You will see whatever you are updating in Test file present in the original location those same changes are getting reflected in the Hardlink file as well this works same as your soft file.

```
root@ubuntu:/home/simran/temp# ls
softlink
root@ubuntu:/home/simran/temp# ln /home/simran/folder18/folder1/TestFile HardLink
root@ubuntu:/home/simran/temp# ls
HardLink softlink
root@ubuntu:/home/simran/temp# ls -ltr
total 4
lrwxrwxrwx 1 root root 35 Nov 16 15:16 softlink -> /home/simran/folder18/folder1/File1
-rw-r--r-- 2 root root 8 Nov 16 15:26 HardLink
root@ubuntu:/home/simran/temp# cat HardLink
testing
root@ubuntu:/home/simran/temp# cat HardLink
testing
Testing HardLink
root@ubuntu:/home/simran/temp#
```

```
root@ubuntu:/home/simran/folder18/folder1# ls
File1 TestFile
root@ubuntu:/home/simran/folder18/folder1# cat TestFile
cat: TestFile: No such file or directory
root@ubuntu:/home/simran/folder18/folder1# cat TestFile
testing
root@ubuntu:/home/simran/folder18/folder1# echo "Testing HardLink" >> TestFile
root@ubuntu:/home/simran/folder18/folder1#
```

- Now let's try to delete the original file

```
root@ubuntu:/home/simran/temp# cat HardLink
testing
root@ubuntu:/home/simran/temp# cat HardLink
testing
Testing HardLink
root@ubuntu:/home/simran/temp# ls
HardLink softlink
root@ubuntu:/home/simran/temp# cat HardLink
testing
Testing HardLink
root@ubuntu:/home/simran/temp#
```

```
root@ubuntu:/home/simran/folder18/folder1# ls
File1 TestFile
root@ubuntu:/home/simran/folder18/folder1# cat TestFile
testing
Testing HardLink
root@ubuntu:/home/simran/folder18/folder1# rm TestFile
root@ubuntu:/home/simran/folder18/folder1# ls
File1
root@ubuntu:/home/simran/folder18/folder1#
```

- Content is still there in the Hardlink File even when we have deleted the original File