Experiment No : 3

Aim : File system hierarchy in a common Linux distribution, file and device permissions, study of system configuration files in /etc, familiarizing log files for system events, user activity, network events.

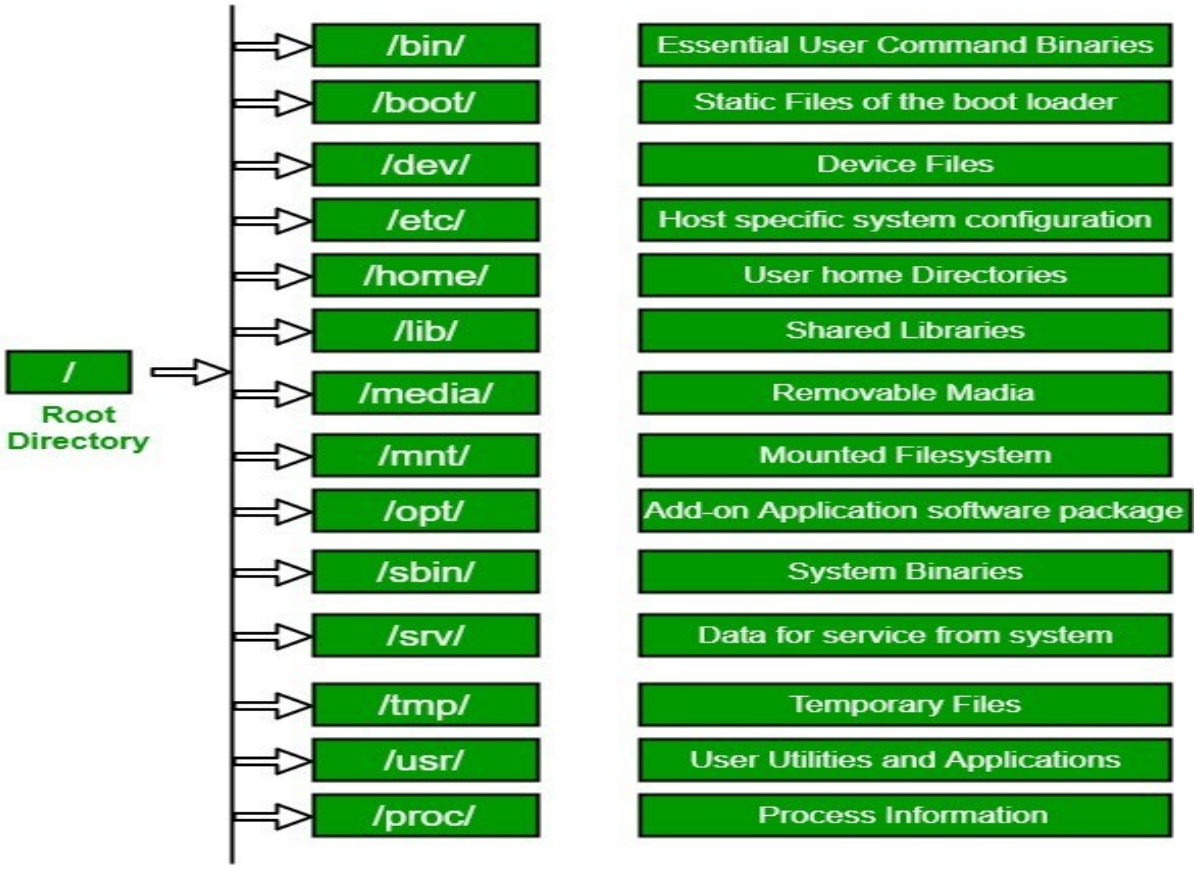
Linux Filesystem Hierarchy Standard (FHS)

Filesystem hierarchy standard describes directory structure and its content in Unix and Unix like operating system. It explains where files and directories should be located and what it should contain.

◆ In the FHS, all files and directories appear under the root directory /, even if they are stored on different physical or virtual devices.

◆ Some of these directories only exist on a particular system if certain subsystems, such as the X Window System, are installed.

◆ Most of these directories exist in all UNIX operating systems and are generally used in much the same way; however, the descriptions here are those used specifically for the FHS, and are not considered authoritative for platforms other than Linux.



Directory type	Types of files stored
Binary directories	Contains binary or compiled source code files, eg, /bin, /sbin, etc.
Configuration directories	Contains configuration files of the system, eg, /etc, /boot.
Data directories	Stores data files, eg, /home, /root, etc.
Memory directories	Stores device files which doesn't take up actual hard disk space, eg, /dev, /proc, /sys.
Usr (Unix System Resources)	Contains sharable, read only data, eg, /usr/bin, /usr/lib, etc.
var (variable directory)	Contains larger size data, eg, /var/log, /var/cache, etc.
Non standerd directories	Directories which do not come under standard FHS, eg, lost+found, /run, etc.

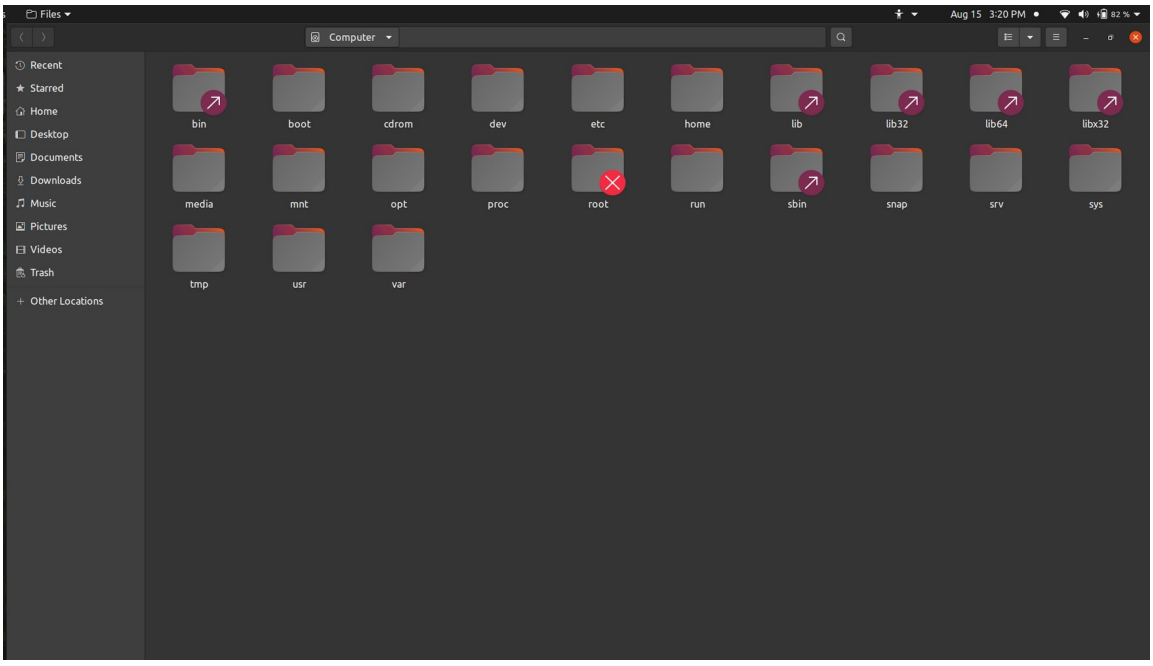
1. Root Directory

Everything on your Linux system is located under the / directory, known as the root directory. You can think of the / directory as being similar to the C:\ directory on Windows – but this isn't strictly true, as Linux doesn't have drive letters. While another partition would be located at D:\ on Windows, this other partition would appear in another folder under / on Linux.

◆ Every single file and directory starts from the root directory

◆ The only root user has the right to write under this directory

◆ /root is the root user's home directory, which is not the same as /

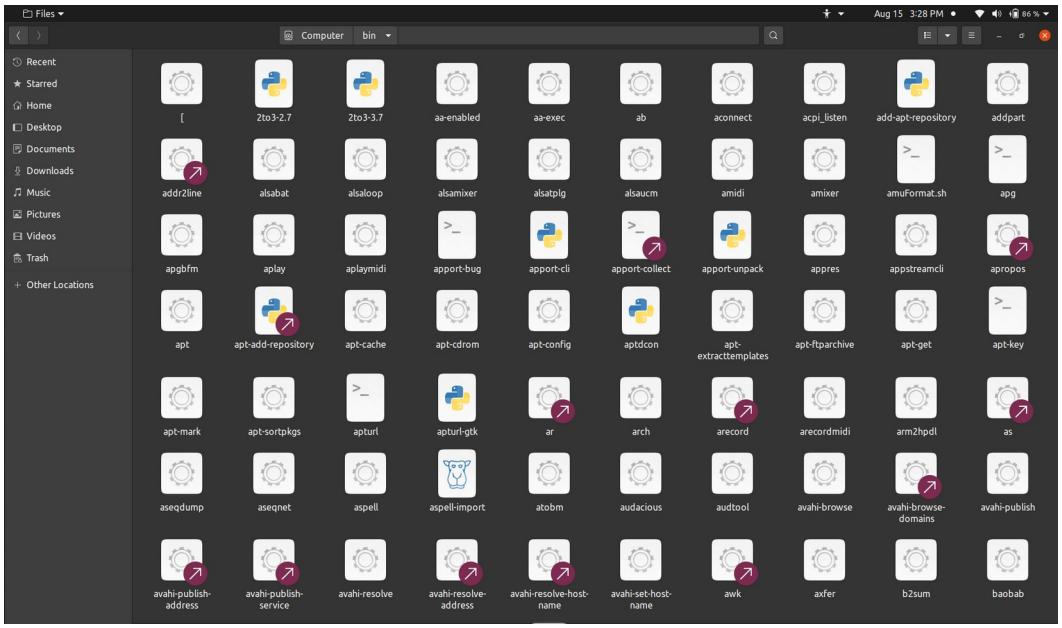


2. /bin : Essential command binaries that need to be available in single-user mode; for all users, e.g., cat, ls, cp.

◆Contains binary executables

◆Common linux commands you need to use in single-user modes are located under this directory.

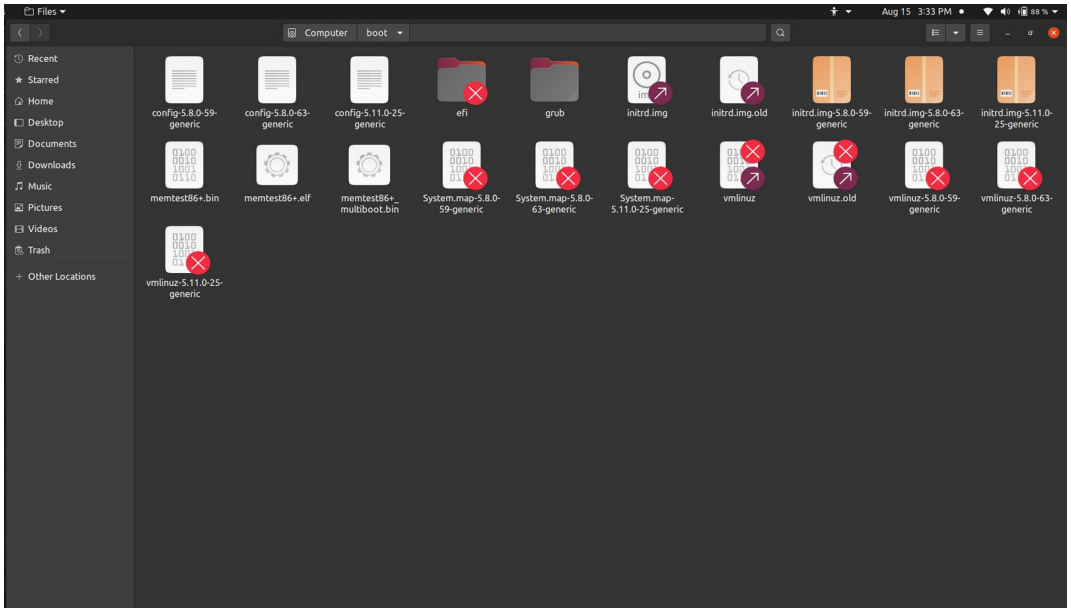
◆Commands used by all the users of the system are located here e.g. ps, ls, ping, grep, cp



3. /boot : Boot loader files, e.g., kernels, initrd.

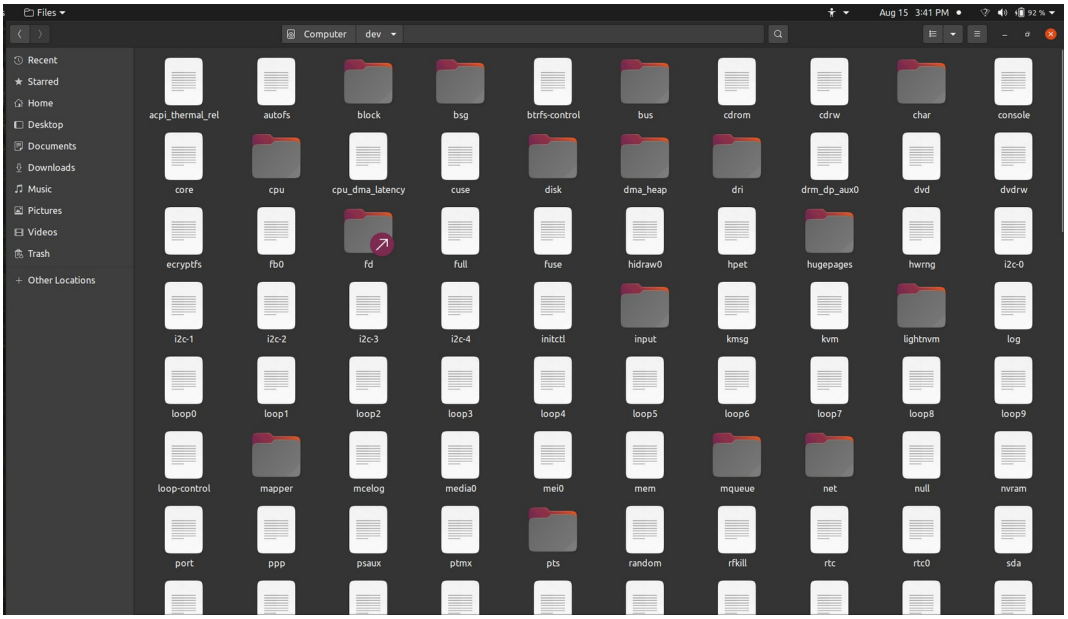
•Kernel initrd, vmlinux, grub files are located under /boot

•Example: initrd.img-2.6.32-24-generic, vmlinuz-2.6.32-24-generic



4. /dev : Device files

Linux exposes devices as files, and the /dev directory contains a number of special files that represent devices. These are not actual files as we know them, but they appear as files – for example, /dev/sda represents the first SATA drive in the system. If you wanted to partition it, you could start a partition editor and tell it to edit /dev/sda.

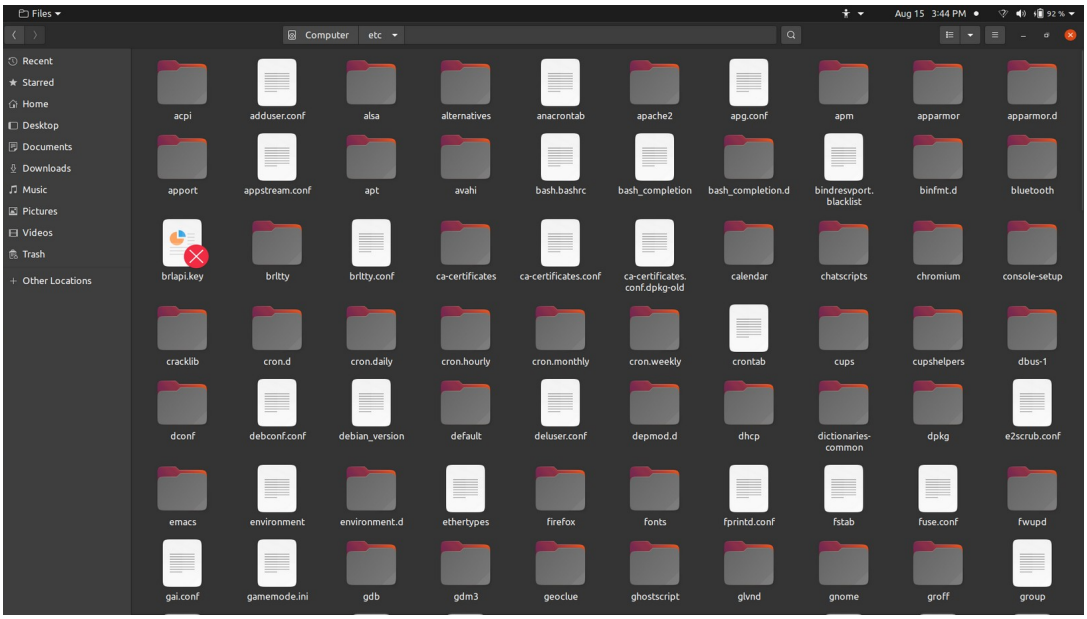


5. /etc :Host-specific system-wide configuration files.

◆ Contains configuration files required by all programs.

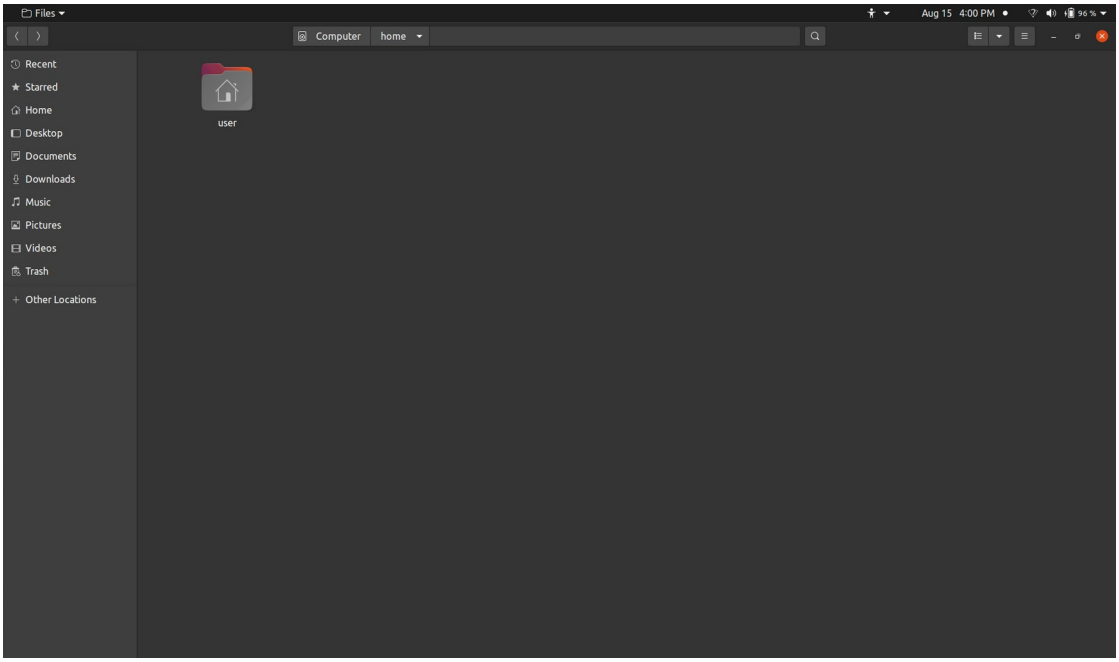
◆ This also contains startup and shutdown shell scripts used to start/stop individual programs.

◆ Example: /etc/resolv.conf, /etc/logrotate.conf.



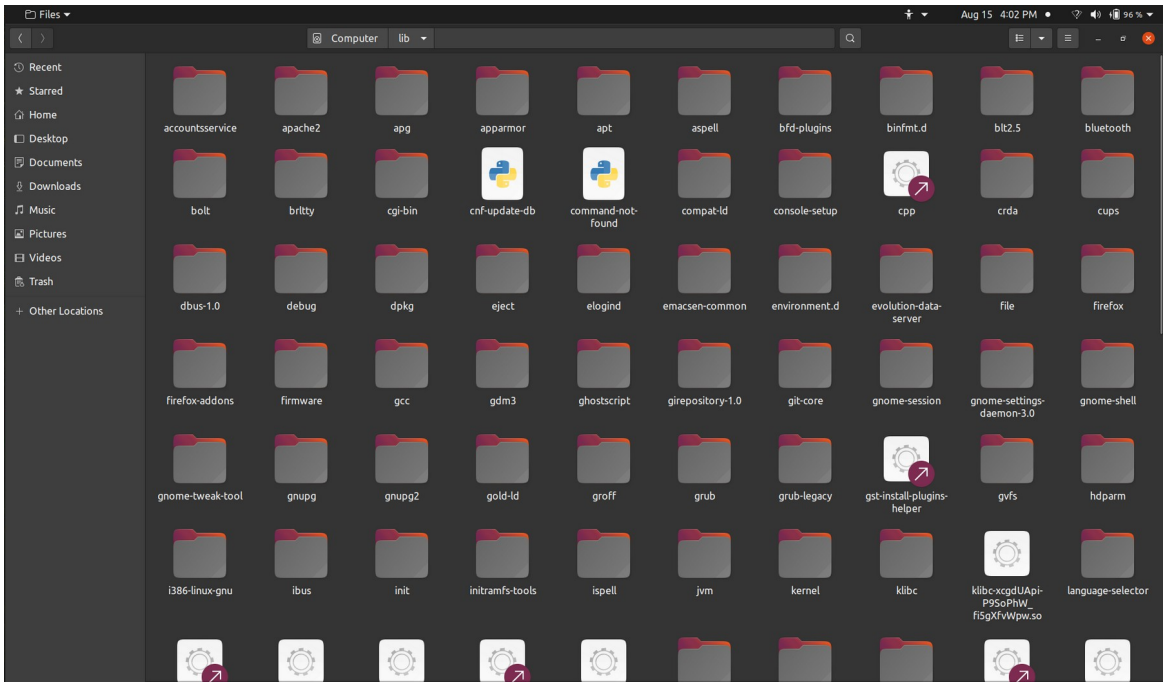
6. /home : Users' home directories, containing saved files, personal settings, etc.

- Home directories for all users to store their personal files.
- example: /home/kishlay, /home/kv.



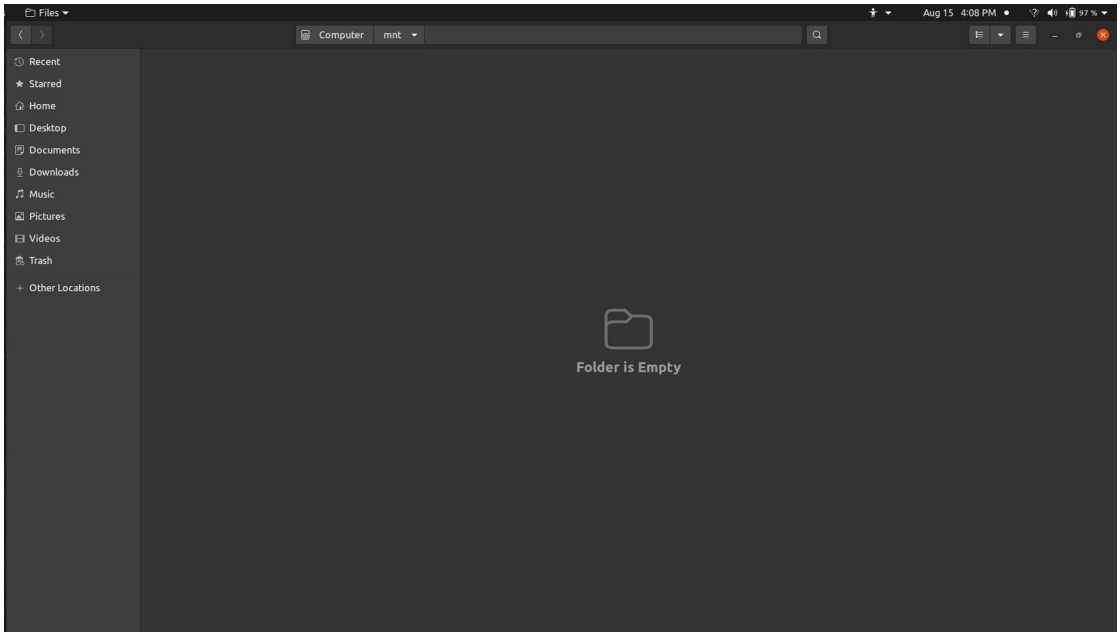
7. /lib : Libraries essential for the binaries in /bin/ and /sbin/.

- Library filenames are either ld* or lib*.so.*
- Example: ld-2.11.1.so, libncurses.so.5.7



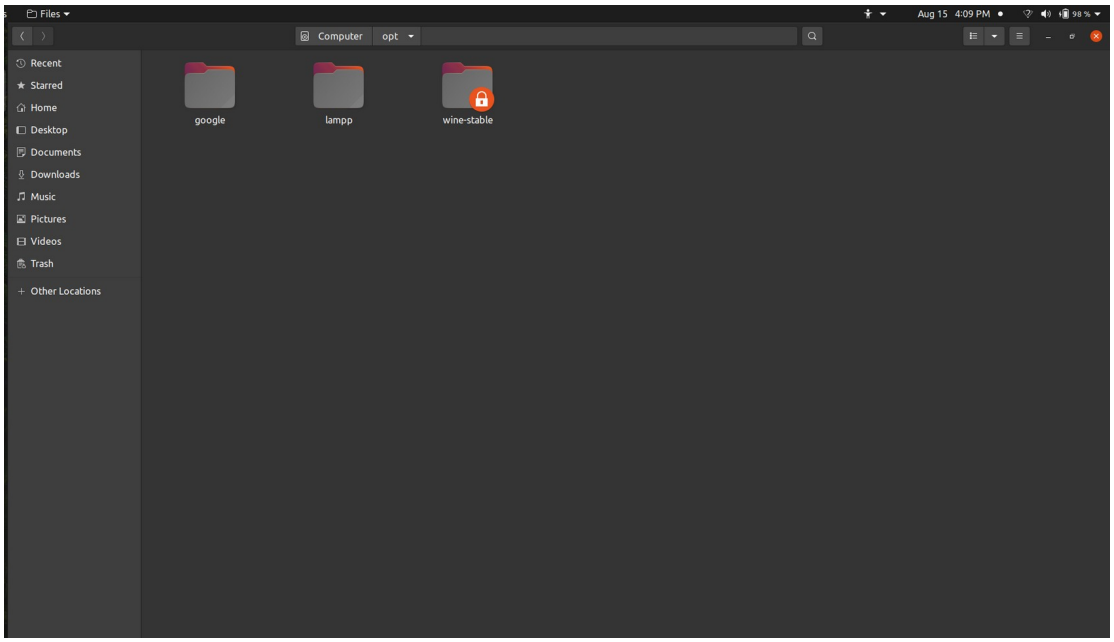
8. /mnt : Temporary Mount Points

Historically speaking, the /mnt directory is where system administrators mounted temporary file systems while using them. For example, if you're mounting a Windows partition to perform some file recovery operations, you might mount it at /mnt/windows. However, you can mount other file systems anywhere on the system.



9. /opt : Optional Packages

The /opt directory contains subdirectories for optional software packages. It's commonly used by proprietary software that doesn't obey the standard file system hierarchy – for example, a proprietary program might dump its files in /opt/application when you install it.

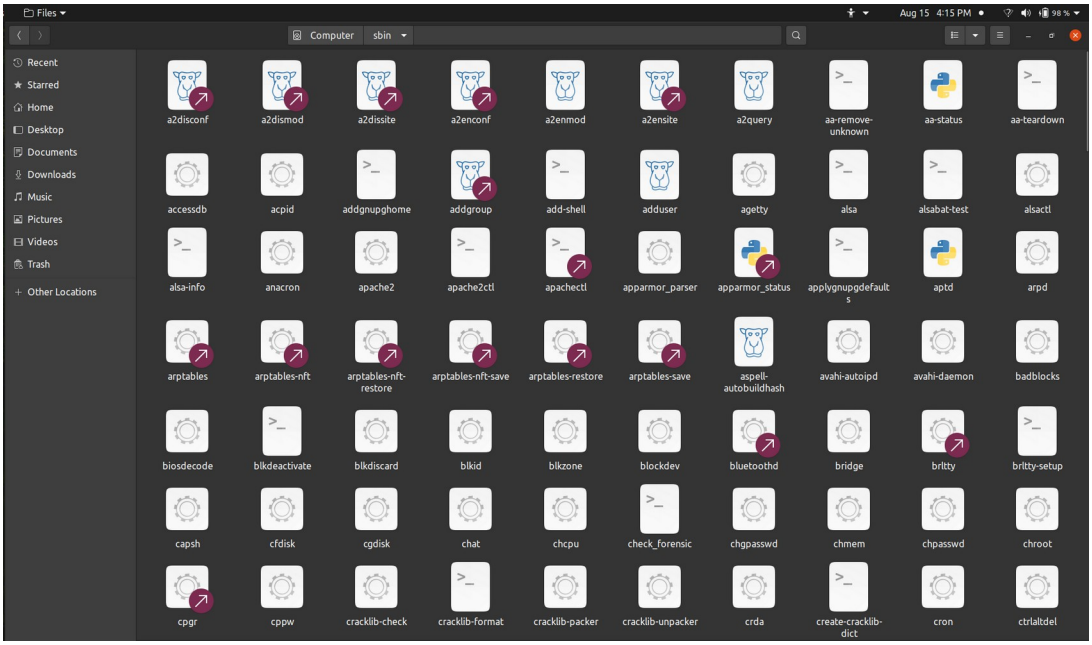


10. /sbin : Essential system binaries, e.g., fsck, init, route.

◆Just like /bin, /sbin also contains binary executables.

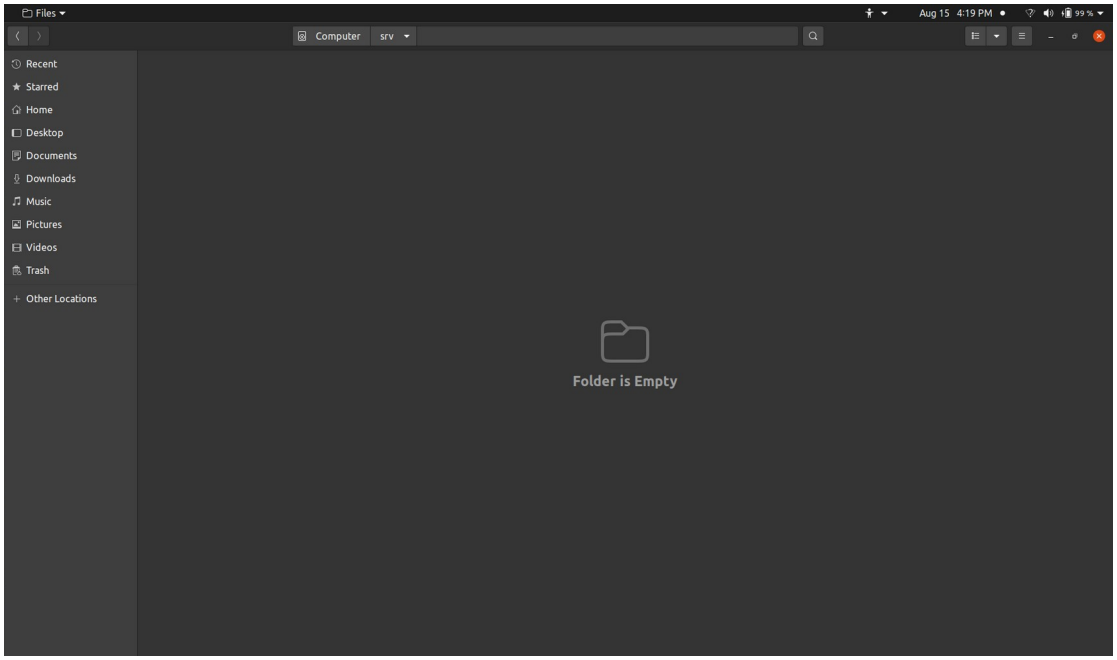
◆The linux commands located under this directory are used typically by system administrator, for system maintenance purpose.

◆Example: iptables, reboot, fdisk, ifconfig, swapon



11. /srv : Service Data

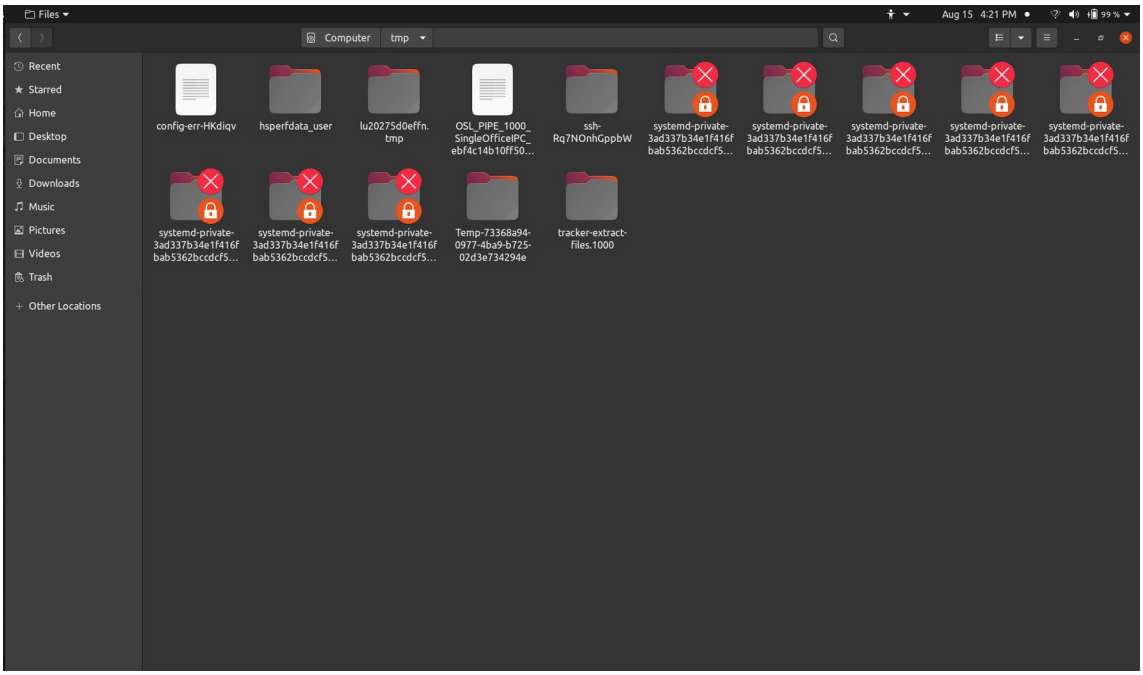
The /srv directory contains “data for services provided by the system.” If you were using the Apache HTTP server to serve a website, you’d likely store your website’s files in a directory inside the /srv directory.



12. /tmp : Temporary files. Often not preserved between system reboots, and may be severely size restricted.

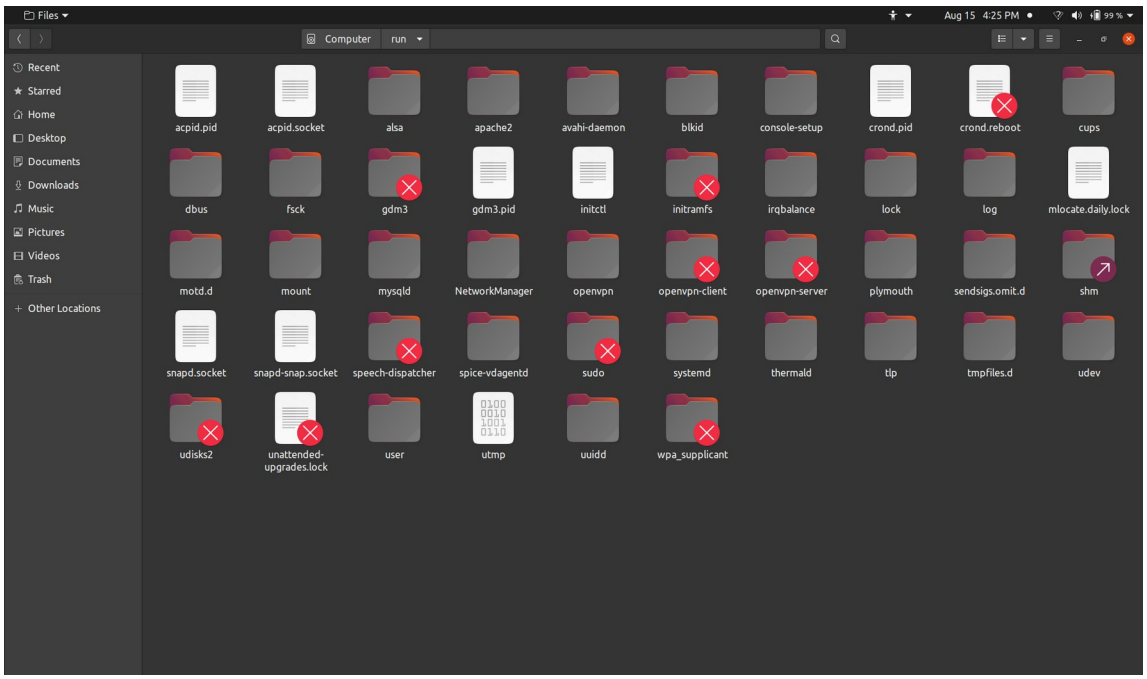
◆ Directory that contains temporary files created by system and users.

◆ Files under this directory are deleted when system is rebooted.



13./run : Application State Files

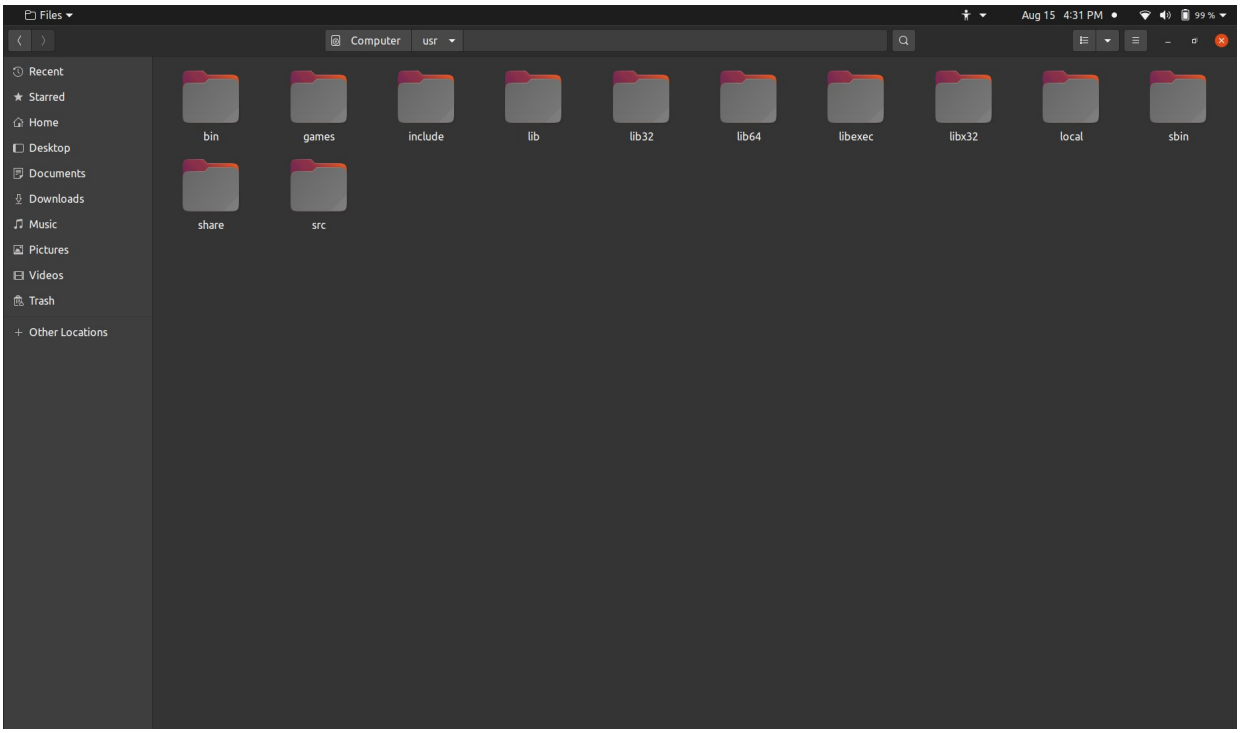
The /run directory is fairly new, and gives applications a standard place to store transient files they require like sockets and process IDs. These files can't be stored in /tmp because files in /tmp may be deleted.



14./usr – User Binaries & Read-Only Data

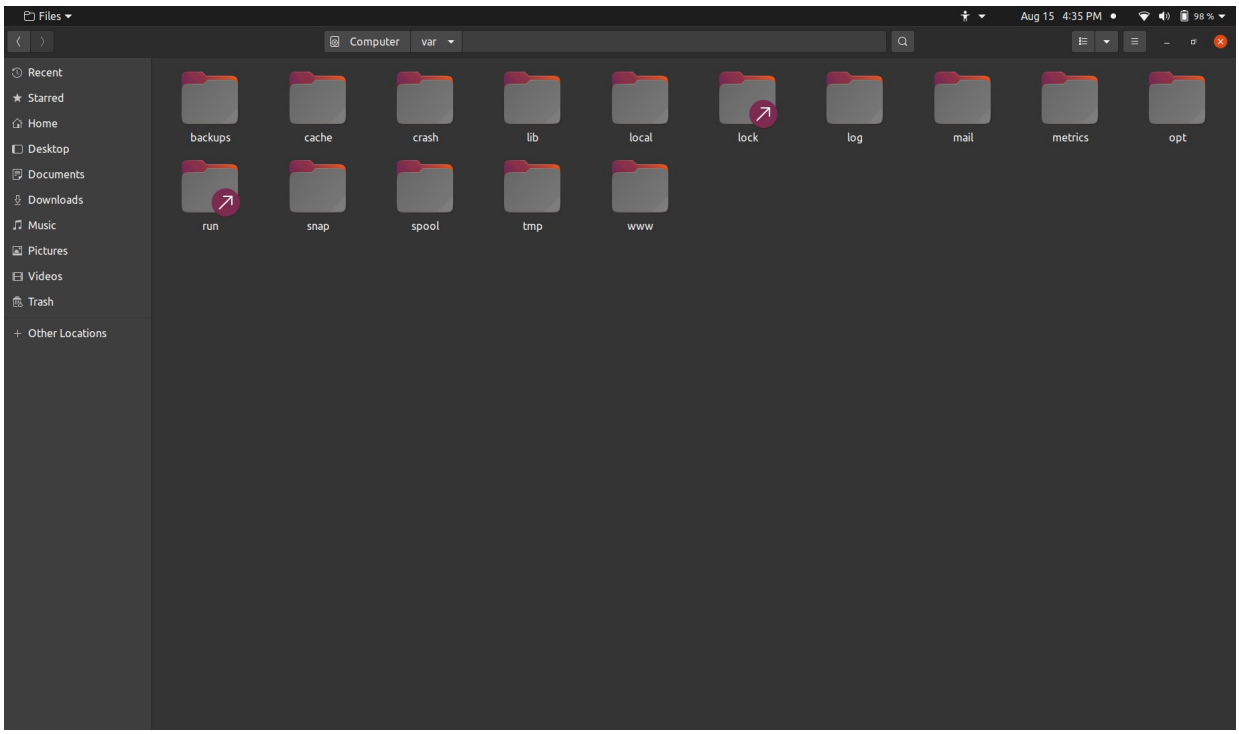
The /usr directory contains applications and files used by users, as opposed to applications and files used by the system. For example, non-essential applications are located inside the /usr/bin directory instead of the /bin directory and non-essential system administration binaries are located in the /usr/sbin directory instead of the /sbin directory. Libraries for each are located inside the /usr/lib directory. The /usr directory also contains other directories – for example, architecture-independent files like graphics are located in /usr/share.

The /usr/local directory is where locally compiled applications install to by default – this prevents them from mucking up the rest of the system.



15. /var – Variable Data Files

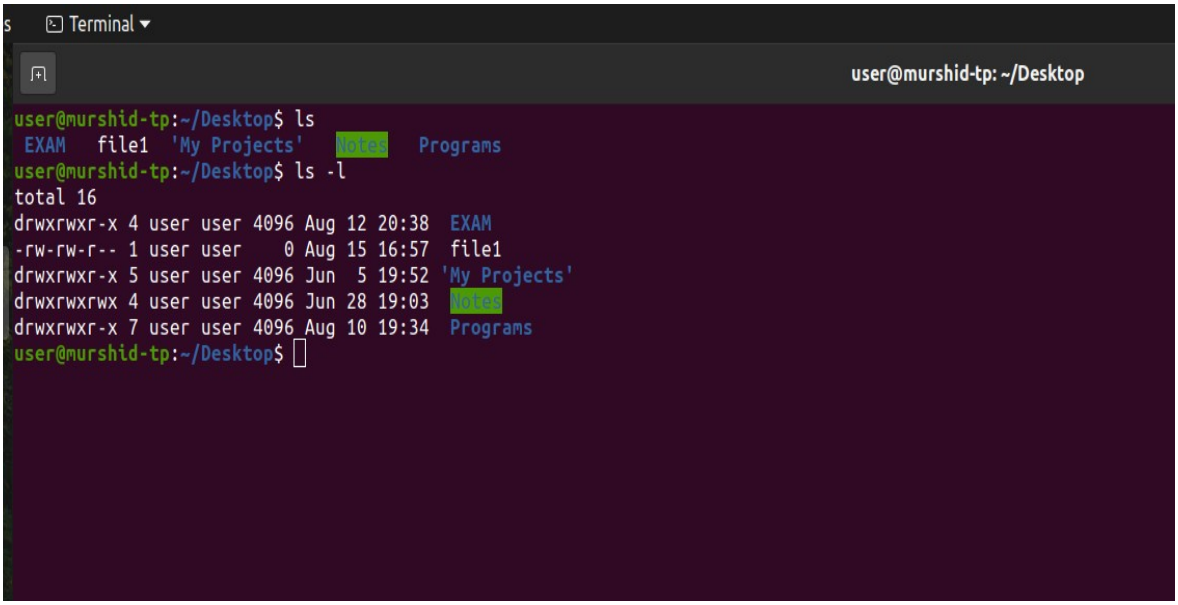
The /var directory is the writable counterpart to the /usr directory, which must be read-only in normal operation. Log files and everything else that would normally be written to /usr during normal operation are written to the /var directory. For example, you'll find log files in /var/log.



File permissions

In Linux, file permissions, attributes, and ownership control the access level that the system processes and users have to files. This ensures that only authorized users and processes can access specific files and directories.

When you execute an “ls” command, you are not given any information about the security of the files, because by default “ls” only lists the names of files. You can get more information by using an “option” with the “ls” command. All options start with a ‘-‘. For example, to execute “ls” with the “long listing” option, you would type ls -l



The basic Linux permissions model works by associating each system file with an owner and a group and assigning permission access rights for three different classes of users:

- **The file owner.**
- **The group members.**
- **Others.**

All the three owners (user owner, group, others) in the Linux system have three types of permissions defined. Nine characters denotes the three types of permissions.

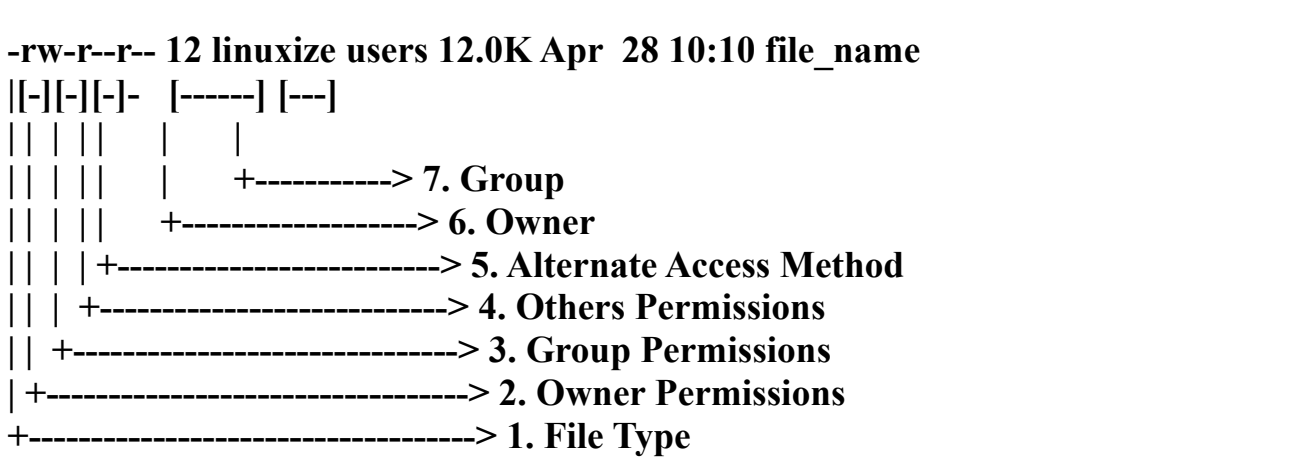
1. **Read (r) :** The read permission allows you to open and read the content of a file. But you can't do any editing or modification in the file.

2. **Write (w) :** The write permission allows you to edit, remove or rename a file. For instance, if a file is present in a directory, and write permission is set on the file but not on the directory, then you can edit the content of the file but can't remove, or rename it.

3. **Execute (x):** In Unix type system, you can't run or execute a program unless execute permission is set.But in Windows, there is no such permission available.

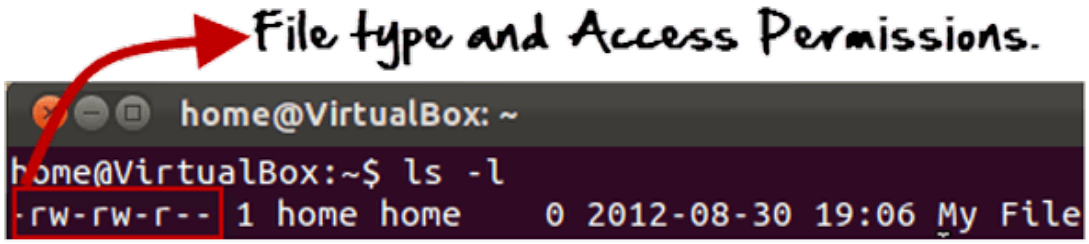
To view the file permissions, use the **ls** command:

ls -l file_name



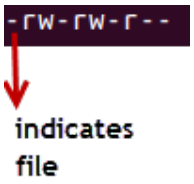
Let's see file permissions in Linux with examples:

ls – l on terminal gives



Here, we have highlighted **'-rw-rw-r--'**and this weird looking code is the one that tells us about the Unix permissions given to the owner, user group and the world.

Here, the first '**-**' implies that we have selected a file.p>



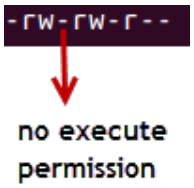
Else, if it were a directory, **d** would have been shown.

The characters are pretty easy to remember.

r= read permission
w = write permission
x = execute permission
- = no permission

Let us look at it this way.

The first part of the code is **'rw-'**. This suggests that the owner 'Home' can:



- Read the file
- Write or edit the file
- He cannot execute the file since the execute bit is set to '-'.

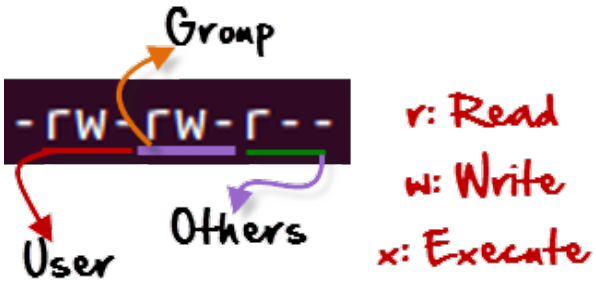
By design, many Linux distributions like Fedora, CentOS, Ubuntu, etc. will add users to a group of the same group name as the user name. Thus, a user 'tom' is added to a group named 'tom'.

The second part is**'rw-'.**It for the user group 'Home' and group-members can:

- •Read the file
- •Write or edit the file

The third part is for the world which means any user. It says **'r--'.** This means the user can only:

- •Read the file



Changing file/directory permissions with 'chmod' command

Say you do not want your colleague to see your personal images. This can be achieved by changing file permissions.

We can use the '**chmod'** command which stands for 'change mode'. Using the command, we can set permissions (read, write, execute) on a file/directory for the owner, group and the world. **Syntax:**

chmod permissions filename
There are 2 ways to use the command -

1. **Absolute mode**
2. **Symbolic mode**

1.Absolute(Numeric) Mode

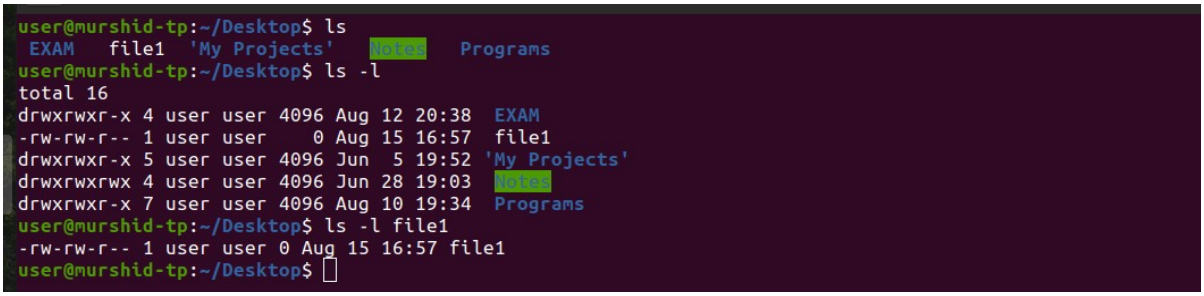
In this mode, file **permissions are not represented as characters but a three-digit octal number**.

The table below gives numbers for all for permissions types.

Number	Permission Type	Symbol
0	No Permission	---
1	Execute	--x
2	Write	-w-
3	Execute + Write	-wx
4	Read	r--
5	Read + Execute	r-x
6	Read +Write	rw-
7	Read + Write +Execute	rwx

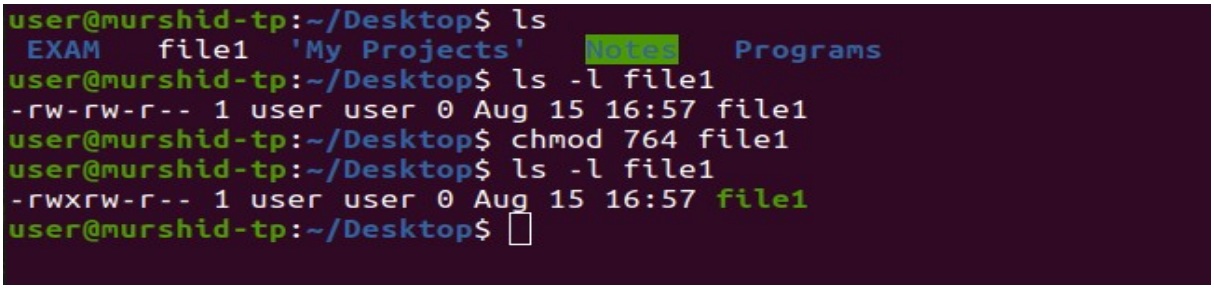
Example :

Checking current file permissions of file1 **: 664(- rw-rw-r--)**



After change the file permission to **764(rwx-rw-r--)** by using the command:

chmod 764 file1



'764' absolute code says the following:

- Owner can read, write and execute
- Usergroup can read and write
- World can only read

2.Symbolic Mode

In the Absolute mode, you change permissions for all 3 owners. In the symbolic mode, you can modify permissions of a specific owner. It makes use of mathematical symbols to modify the Unix file permissions.

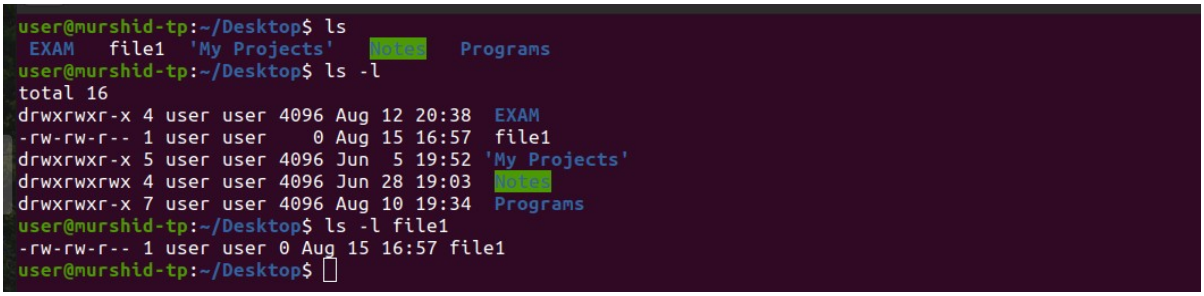
Operator	Description
+	Adds a permission to a file or directory
-	Removes the permission
=	Sets the permission and overrides the permissions set earlier.

The various owners are represented as -

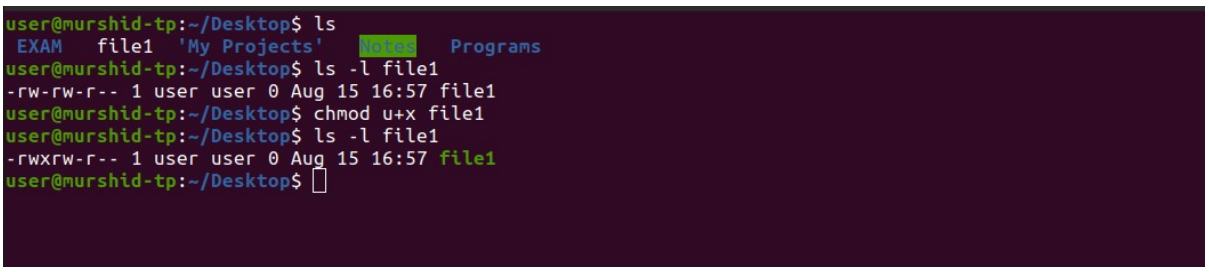
User Denotations	
u	user/owner
g	group
o	other
a	all

Example :

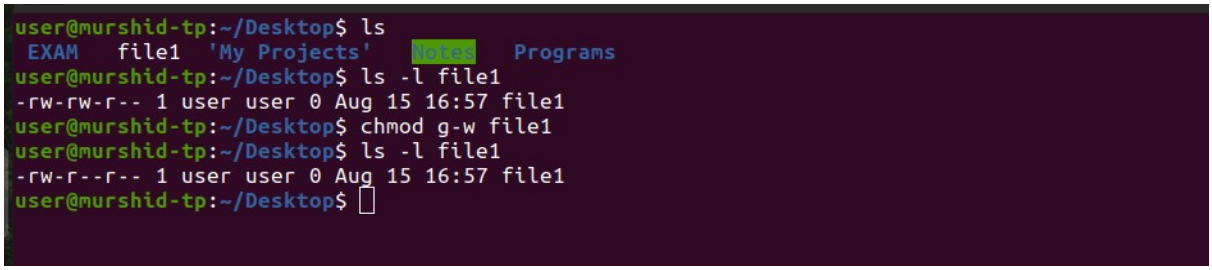
Checking current file permissions of file1 : - rw-rw-r--



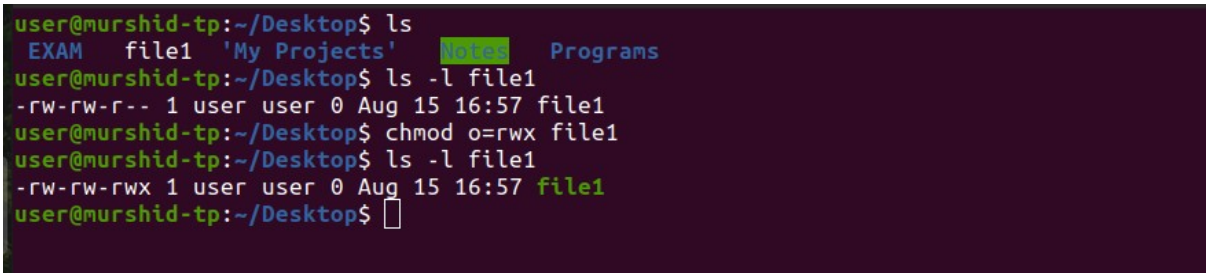
Adding execute permission to owner :



Removing wright permission to group :



Setting all permission to others:



Changing Ownership and Group

For changing the ownership of a file/directory, you can use the following command:

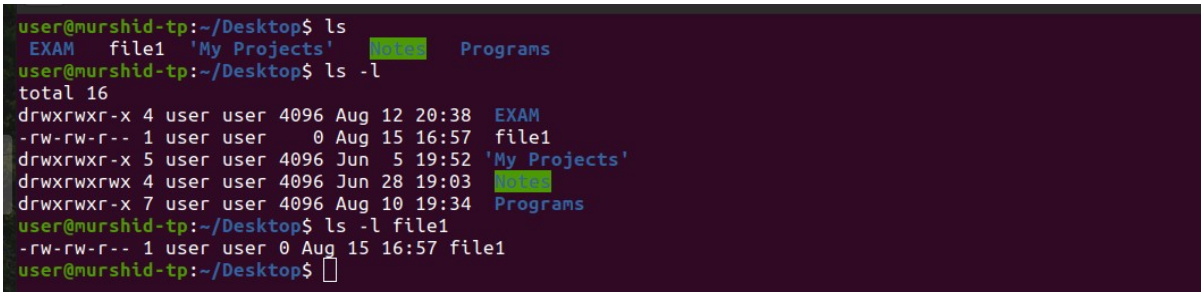
```
chown user filename
```

In case you want to change the user as well as group for a file or directory use the command

```
chown user:group filename
```

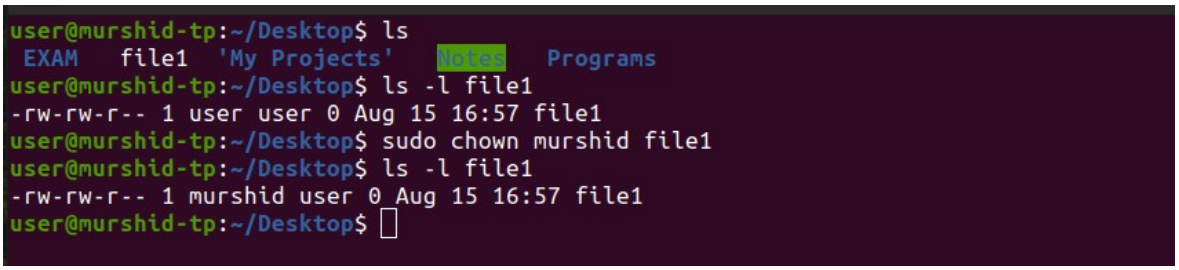
Let's see this in action

Checking the current ownership:



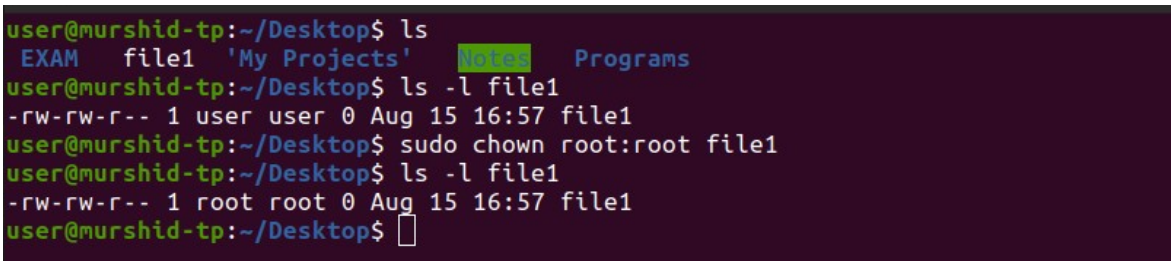
Change the file owner to musrhid,by using the command :

sudo chown murshid file1



Changing user and group to root :

sudo chown root:root file1



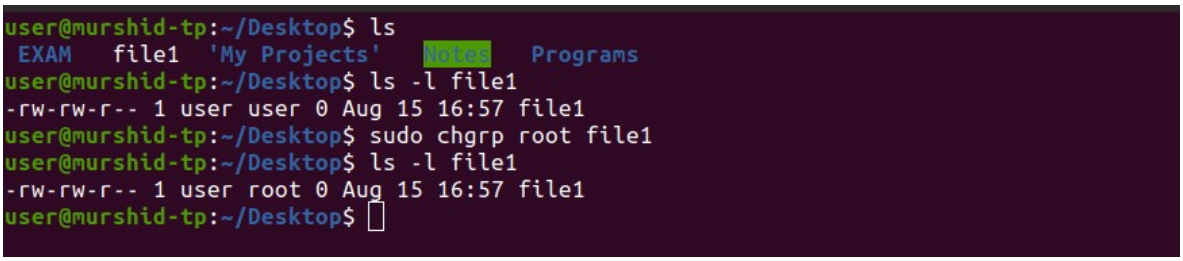
In case you want to change group-owner only, use the command

```
chgrp group_name filename
```

'**chgrp**' stands for change group.

Change the group owner to root :

sudo chgrp root file1



The most important configuration files

As we mentioned before, most configuration files are stored in the /etc directory. Content can be viewed using the **cat** command, which sends text files to the standard output (usually your monitor). The syntax is straight forward:

cat file1 file2 ... fileN

In this section we try to give an overview of the most common configuration files. This is certainly not a complete list. Adding extra packages may also add extra configuration files in /etc. When reading the configuration files, you will find that they are usually quite well commented and self-explanatory. Some files also have man pages which contain extra documentation, such as **man *group***.

Table 3-3. Most common configuration files

File	Information/service
aliases	Mail aliases file for use with the Sendmail and Postfix mail server. Running a mail server on each and every system has long been common use in the UNIX world, and almost every Linux distribution still comes with a Sendmail package. In this file local user names are matched with real names as they occur in E-mail addresses, or with other local addresses.
apache	Config files for the Apache web server.
bashrc	The system-wide configuration file for the Bourne Again SHell. Defines functions and aliases for all users. Other shells may have their own system-wide config files, like cshrc.
crontab and the cron.* directories	Configuration of tasks that need to be executed periodically - backups, updates of the system databases, cleaning of the system, rotating logs etc.
default	Default options for certain commands, such as **useradd**.
filesystems	Known file systems: ext3, vfat, iso9660 etc.
fstab	Lists partitions and their *mount points*.
ftp*	Configuration of the ftp-server: who can connect, what parts of the system are accessible etc.
group	Configuration file for user groups. Use the shadow utilities **groupadd**, **groupmod** and **groupdel** to edit this file. Edit manually only if you really know what you are doing.
hosts	A list of machines that can be contacted using the network, but without the need for a domain name service. This has nothing to do with the system's network configuration, which is done in /etc/sysconfig.
inittab	Information for booting: mode, number of text consoles etc.
issue	Information about the distribution (release version and/or kernel

File	Information/service
	info).
ld.so.conf	Locations of library files.
lilo.conf, silo.conf, aboot.conf etc.	Boot information for the LInux LOader, the system for booting that is now gradually being replaced with GRUB.
logrotate.*	Rotation of the logs, a system preventing the collection of huge amounts of log files.
mail	Directory containing instructions for the behavior of the mail server.
modules.conf	Configuration of modules that enable special features (drivers).
motd	Message Of The Day: Shown to everyone who connects to the system (in text mode), may be used by the system admin to announce system services/maintenance etc.
mtab	Currently mounted file systems. It is advised to never edit this file.
nsswitch.conf	Order in which to contact the name resolvers when a process demands resolving of a host name.
pam.d	Configuration of authentication modules.
passwd	Lists local users. Use the shadow utilities **useradd**, **usermod** and **userdel** to edit this file. Edit manually only when you really know what you are doing.
printcap	Outdated but still frequently used printer configuration file. Don't edit this manually unless you really know what you are doing.
profile	System wide configuration of the shell environment: variables, default properties of new files, limitation of resources etc.
rc*	Directories defining active services for each run level.
resolv.conf	Order in which to contact DNS servers (Domain Name Servers only).
sendmail.cf	Main config file for the Sendmail server.
services	Connections accepted by this machine (open ports).
sndconfig or sound	Configuration of the sound card and sound events.
ssh	Directory containing the config files for secure shell client and server.
sysconfig	Directory containing the system configuration files: mouse, keyboard, network, desktop, system clock, power management etc. (specific to RedHat)
X11	Settings for the graphical server, X. RedHat uses XFree, which is reflected in the name of the main configuration file, XFree86Config. Also contains the general directions for the window managers available on the system, for example **gdm**, **fvwm**, **twm**, etc.
xinetd.* or inetd.conf	Configuration files for Internet services that are run from the system's (extended) Internet services daemon (servers that don't run an independent daemon).

Linux log files

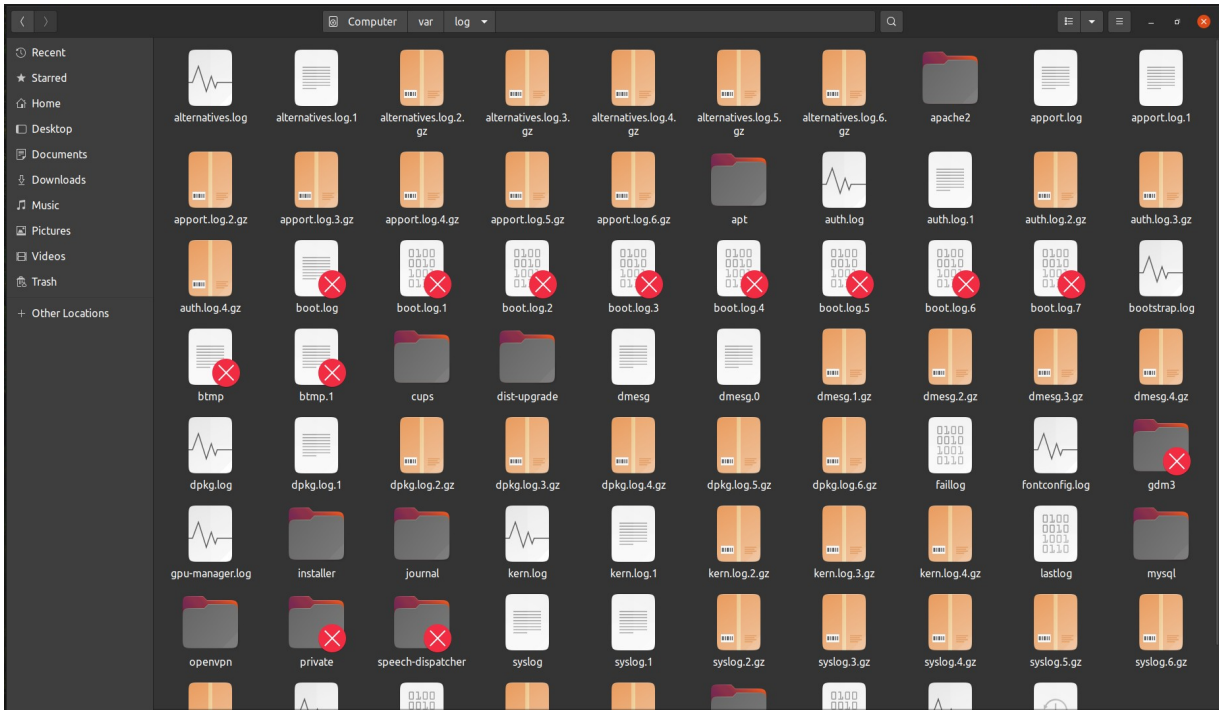
Log files are the records that Linux stores for administrators to keep track and monitor important events about the server, kernel, services, and applications running on it. In this post, we'll go over the top Linux log files server administrators should monitor.

Log files are a set of records that Linux maintains for the administrators to keep track of important events. They contain messages about the server, including the kernel, services and applications running on it.

Linux provides a centralized repository of log files that can be located under the **/var/log** directory.

The log files generated in a Linux environment can typically be classified into four different categories:

- Application Logs
- Event Logs
- Service Logs
- System Logs



Common Linux log files names and usage

- /var/log/messages : General message and system related stuff
- /var/log/auth.log : Authenication logs
- /var/log/kern.log : Kernel logs
- /var/log/cron.log : Crond logs (cron job)
- /var/log/maillog : Mail server logs
- /var/log/qmail/ : Qmail log directory (more files inside this directory)
- /var/log/httpd/ : Apache access and error logs directory
- /var/log/lighttpd/ : Lighttpd access and error logs directory
- /var/log/nginx/ : Nginx access and error logs directory
- /var/log/apt/ : Apt/apt-get command history and logs directory
- /var/log/boot.log : System boot log
- /var/log/mysqld.log : MySQL database server log file
- /var/log/secure or /var/log/auth.log : Authentication log
- /var/log/utmp or /var/log/wtmp : Login records file
- /var/log/yum.log or /var/log/dnf.log: Yum/Dnf command log file.