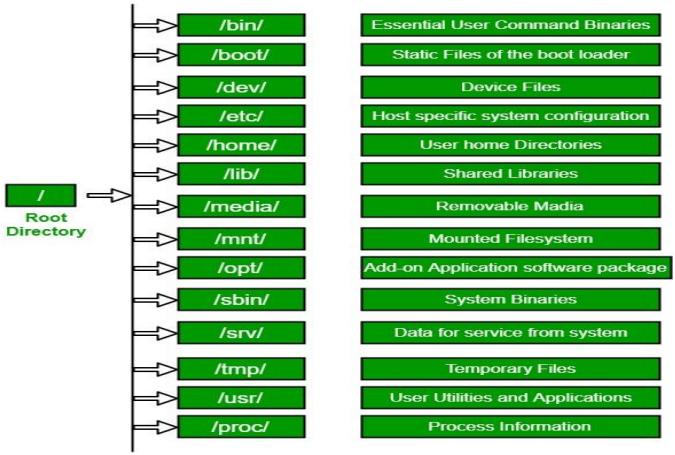
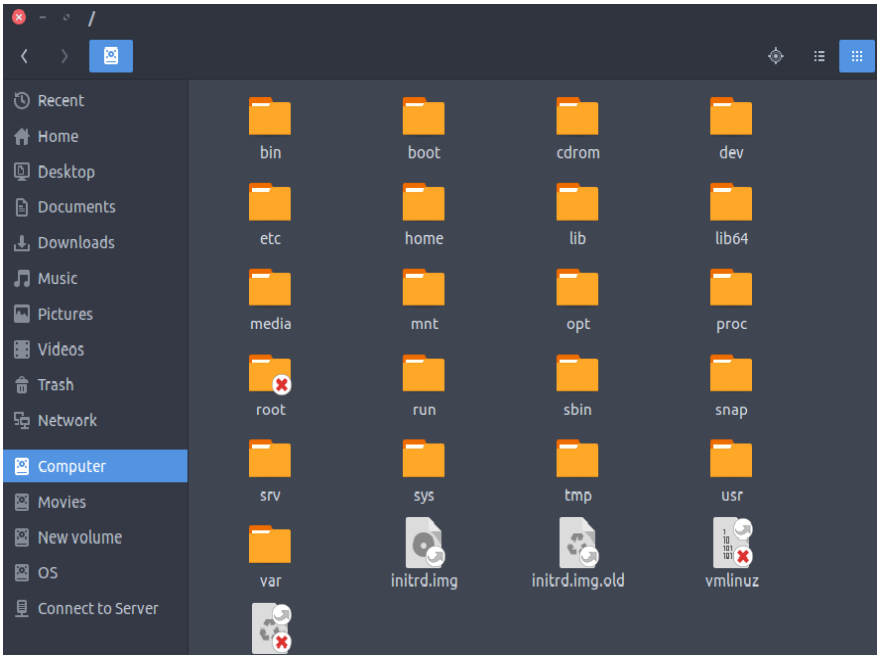
File System Structure, Navigation Commands (cd, ls, and pwd)

The Linux File Hierarchy Structure or the Filesystem Hierarchy Standard (FHS) defines the directory structure and directory contents in Unix-like operating systems. It is maintained by the Linux Foundation.

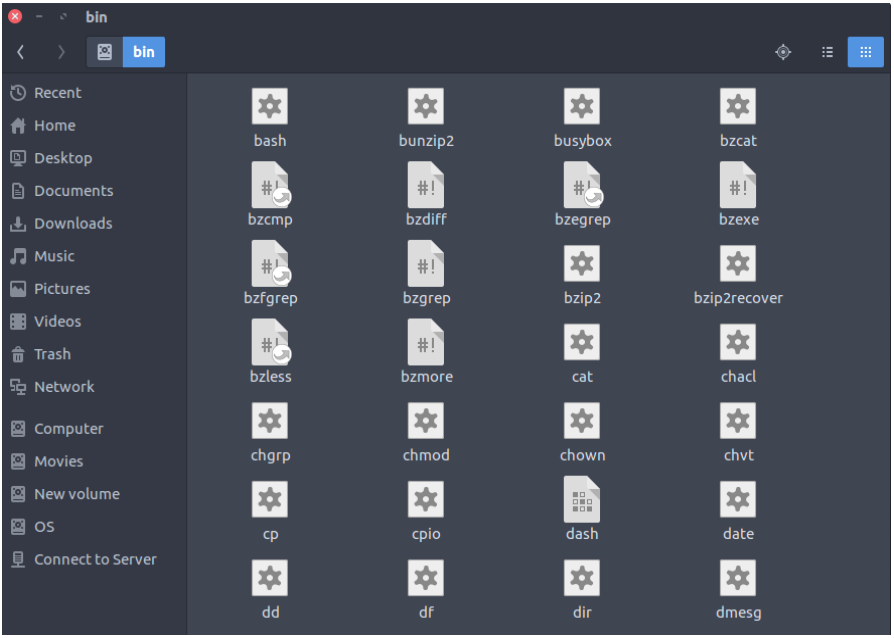
- In the FHS, all files and directories appear under the root directory /, even if they are stored on different physical or virtual devices.
- Some of these directories only exist on a particular system if certain subsystems, such as the X Window System, are installed.
- Most of these directories exist 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.



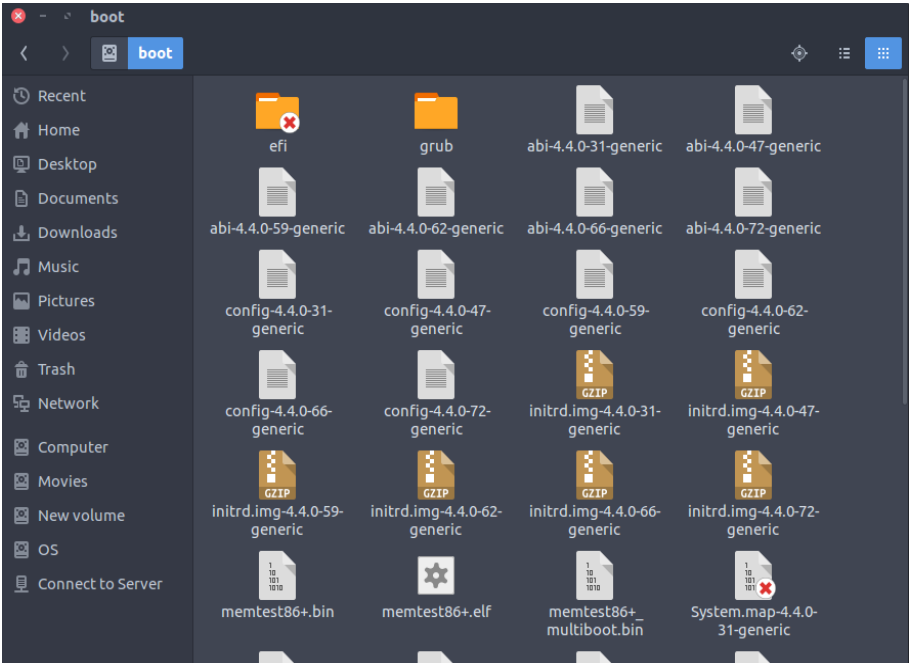
1. **/ (Root):** Primary hierarchy root and root directory of the entire file system hierarchy.
    - Every single file and directory starts from the root directory
    - The only root user has the right to write under this directory
    - /root is the root user's home directory, which is not the same as /



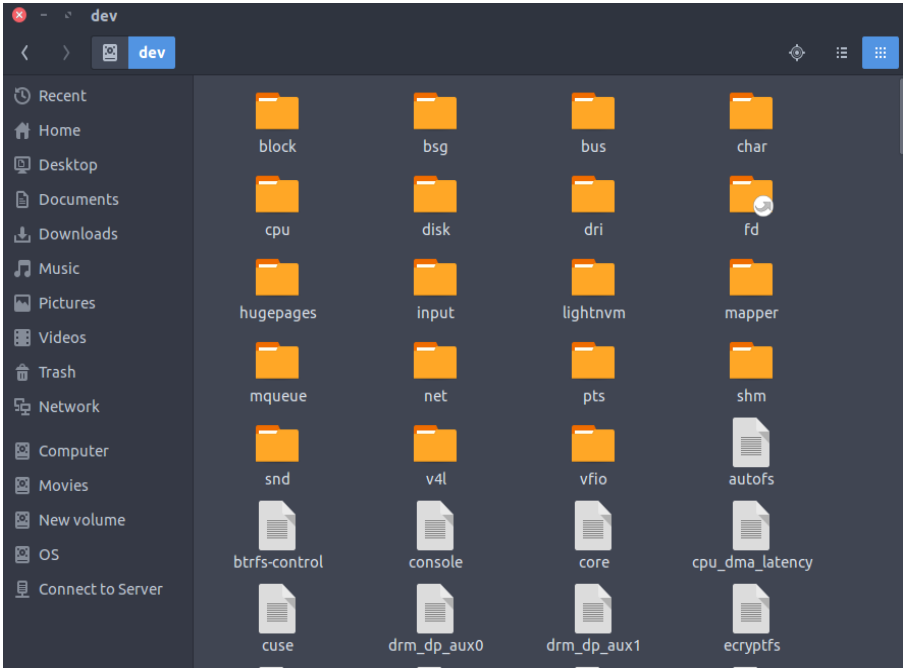
2. **/bin :** Essential command binaries that need to be available in single-user mode; for all users, e.g., cat, ls, cp.
   - Contains binary executables
   - Common linux commands you need to use in single-user modes are located under this directory.
   - Commands used by all the users of the system are located here e.g. ps, ls, ping, grep, cp



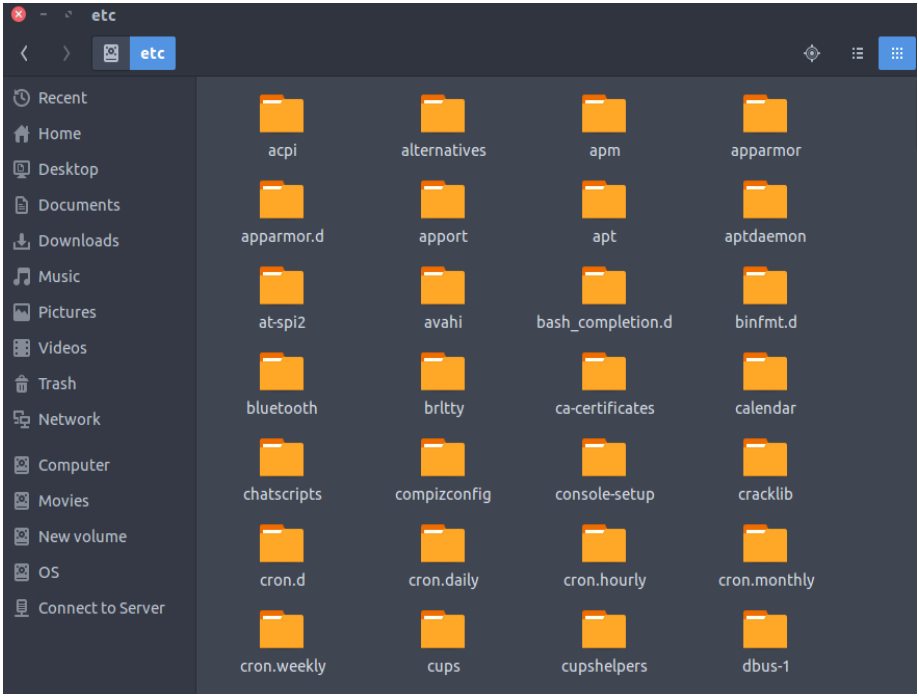
3. **/boot :** Boot loader files, e.g., kernels, initrd.
   - Kernel initrd, vmlinux, grub files are located under /boot
   - Example: initrd.img-2.6.32-24-generic, vmlinuz-2.6.32-24-generic



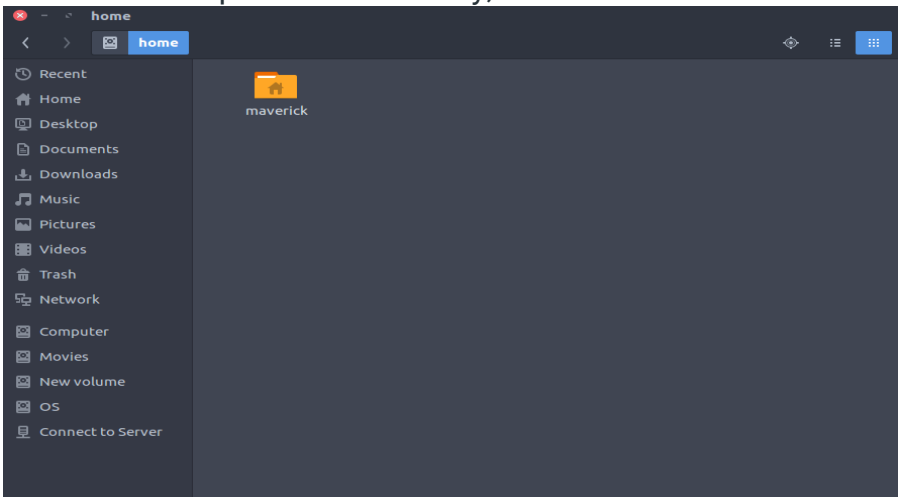
4. **/dev :** Essential device files, e.g., /dev/null.
   - These include terminal devices, usb, or any device attached to the system.
   - Example: /dev/tty1, /dev/usbmon0

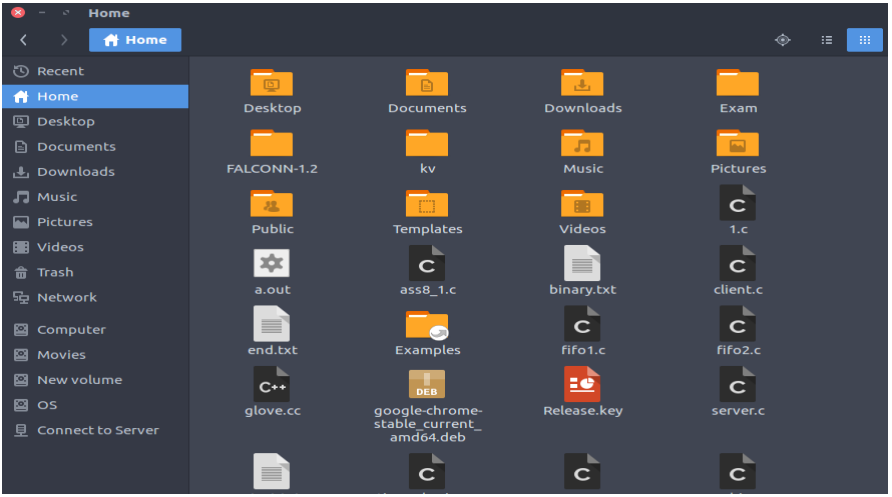


5. **/etc :** Host-specific system-wide configuration files.
   - Contains configuration files required by all programs.
   - This also contains startup and shutdown shell scripts used to start/stop individual programs.
   - Example: /etc/resolv.conf, /etc/logrotate.conf.

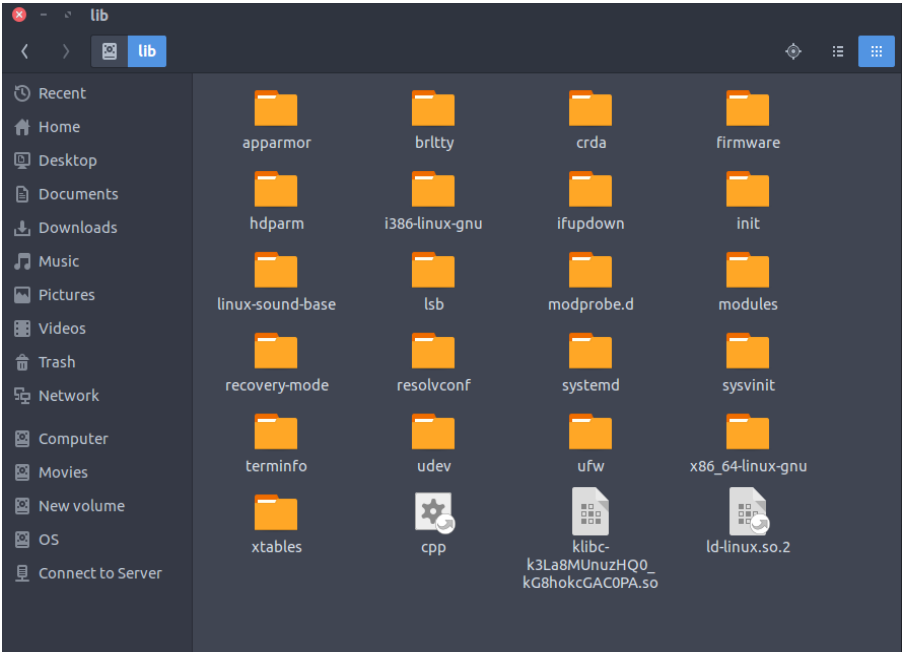


6. **/home :** Users' home directories, containing saved files, personal settings, etc.
   - Home directories for all users to store their personal files.
   - example: /home/kishlay, /home/kv

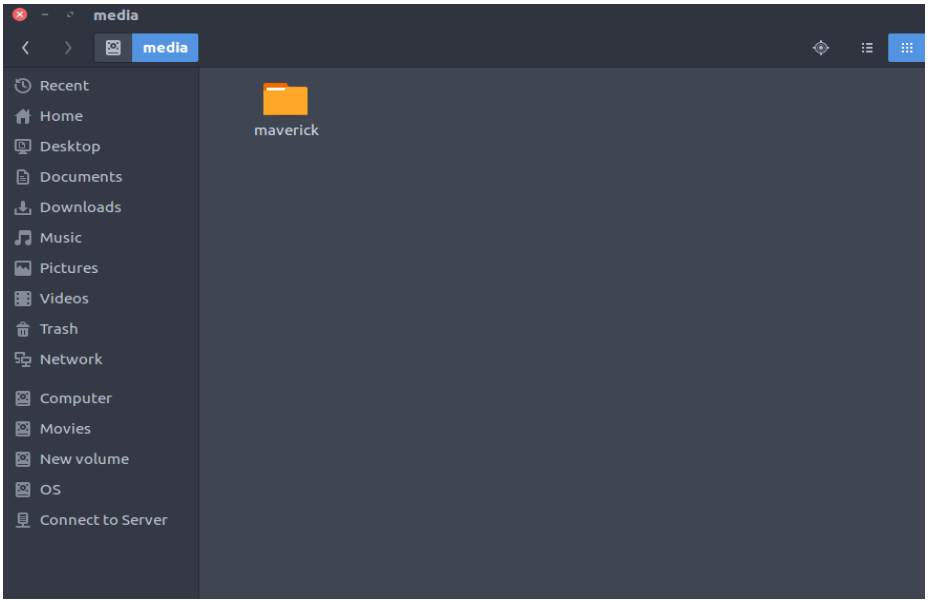




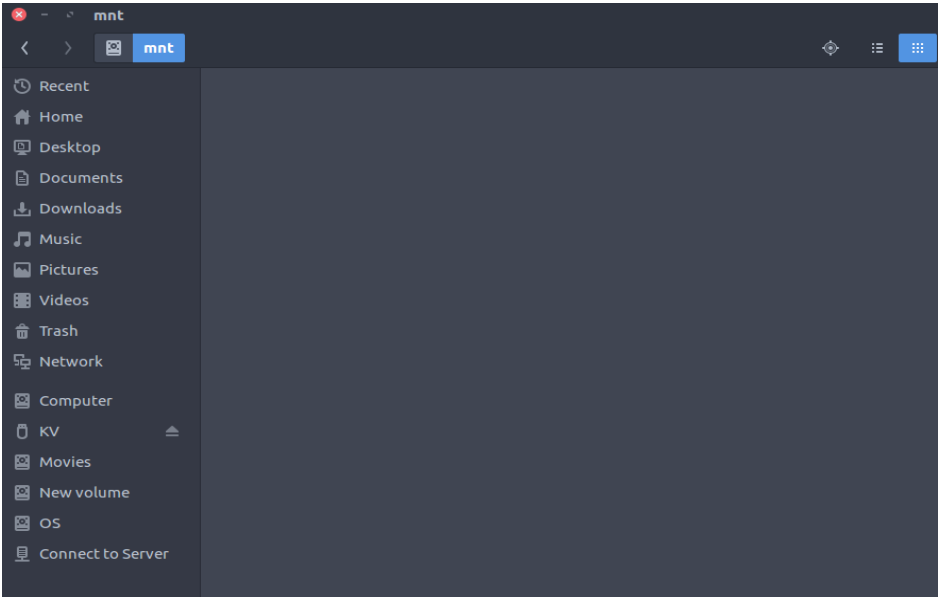
7. **/lib :** Libraries essential for the binaries in /bin/ and /sbin/.
   - Library filenames are either ld* or lib*.so.*
   - Example: ld-2.11.1.so, libncurses.so.5.7



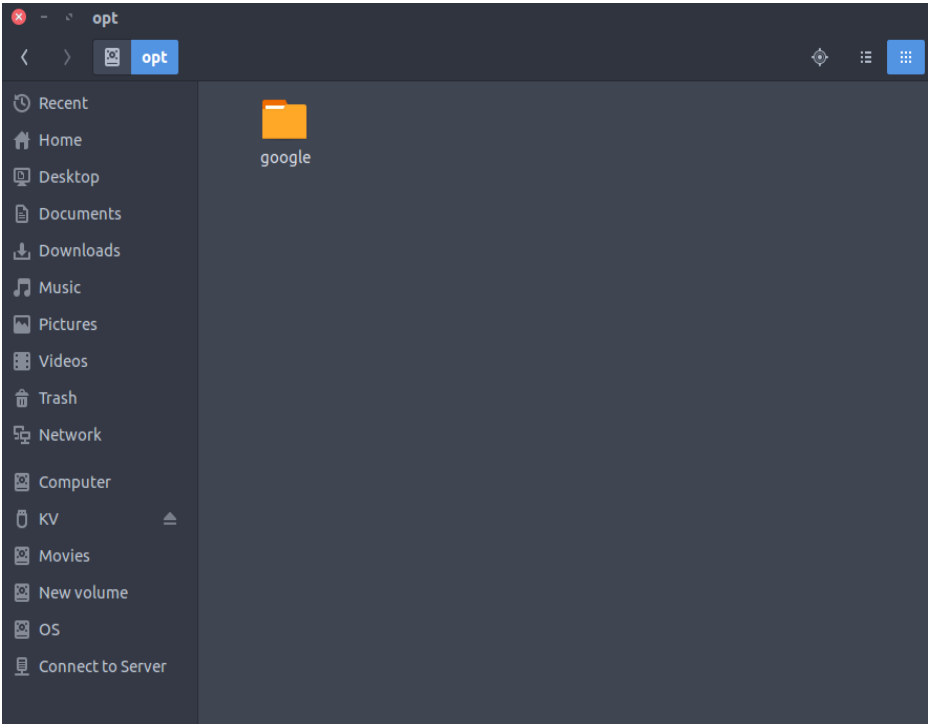
8. **/media :** Mount points for removable media such as CD-ROMs (appeared in FHS-2.3).
    - Temporary mount directory for removable devices.
    - Examples, /media/cdrom for CD-ROM; /media/floppy for floppy drives; /media/cdrecorder for CD writer



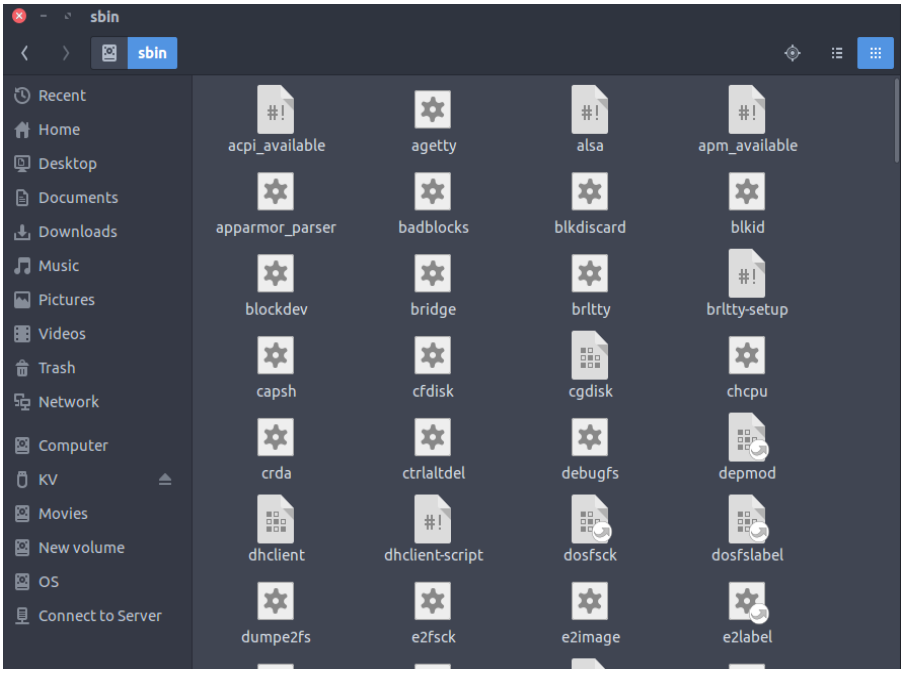
9. **/mnt :** Temporarily mounted filesystems.
   - Temporary mount directory where sysadmins can mount filesystems.



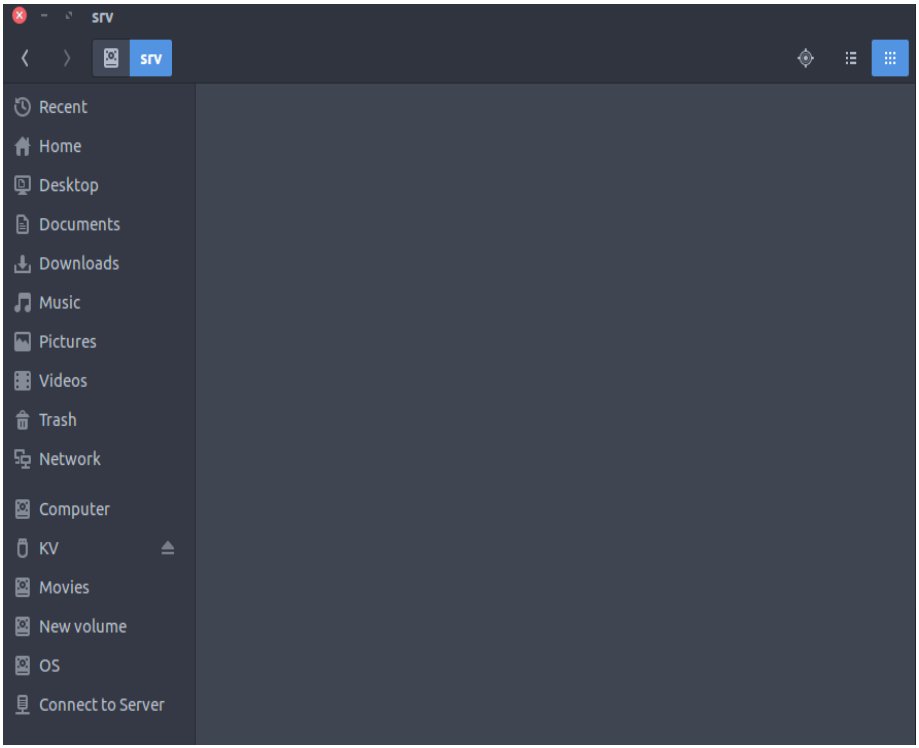
10. **/opt :** Optional application software packages.
    - Contains add-on applications from individual vendors.
    - Add-on applications should be installed under either /opt/ or /opt/ sub-directory.



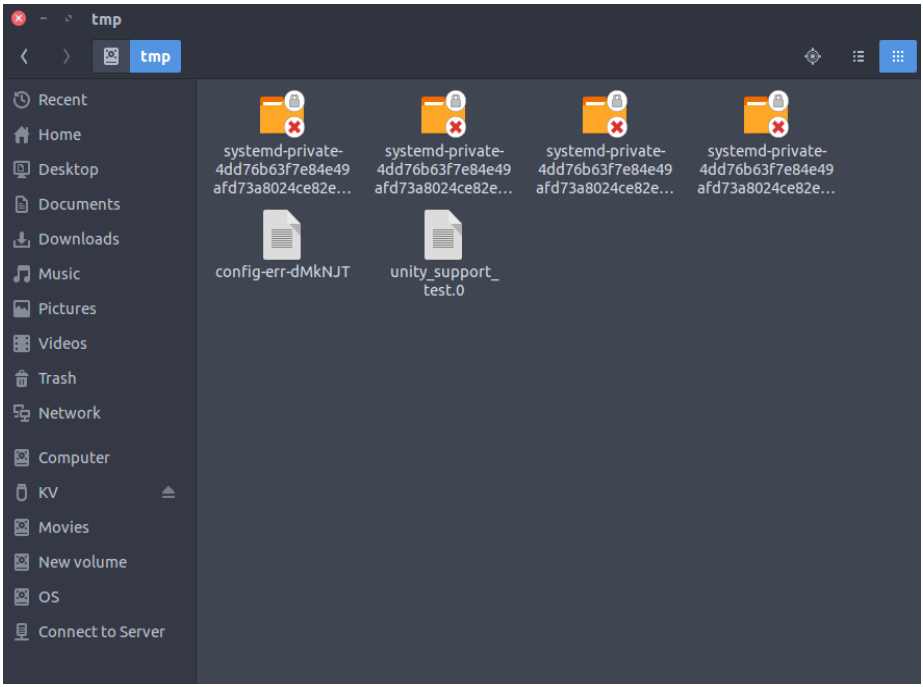
11. **/sbin :** Essential system binaries, e.g., fsck, init, route.
    - Just like /bin, /sbin also contains binary executables.
    - The Linux commands located under this directory are used typically by the system administrator, for system maintenance purpose.
    - Example: iptables, reboot, fdisk, ifconfig, swapon



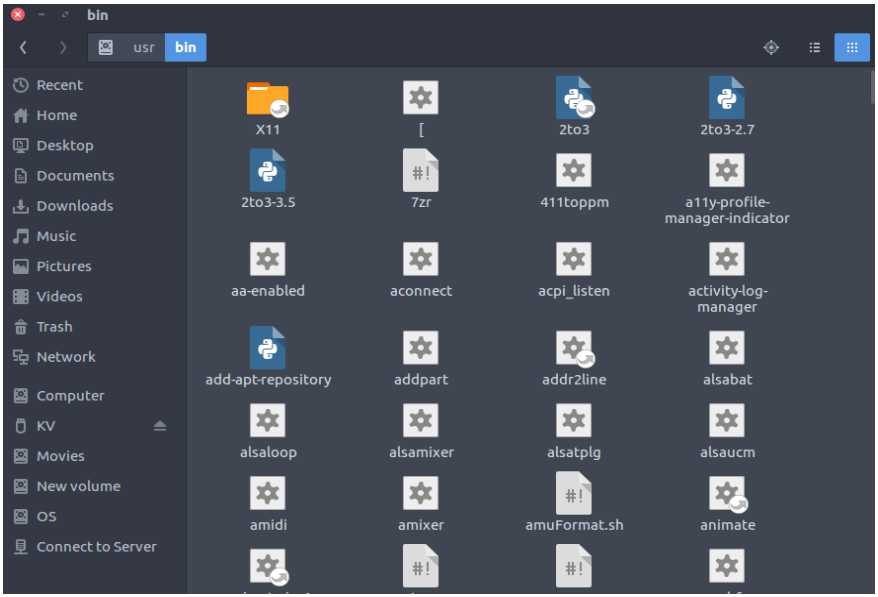
12. **/srv :** Site-specific data served by this system, such as data and scripts for web servers, data offered by FTP servers, and repositories for version control systems.
    - srv stands for service.
    - Contains server specific services related data.
    - Example, /srv/cvs contains CVS related data.

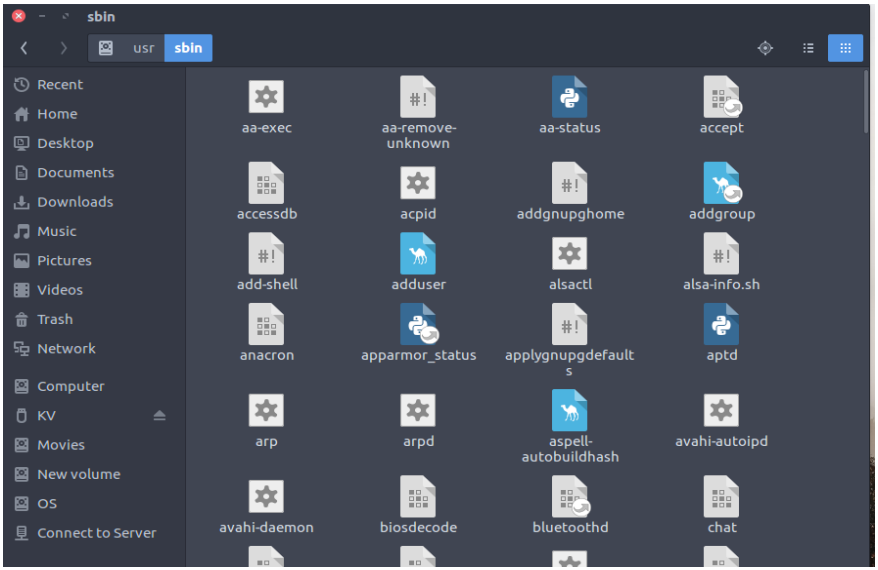


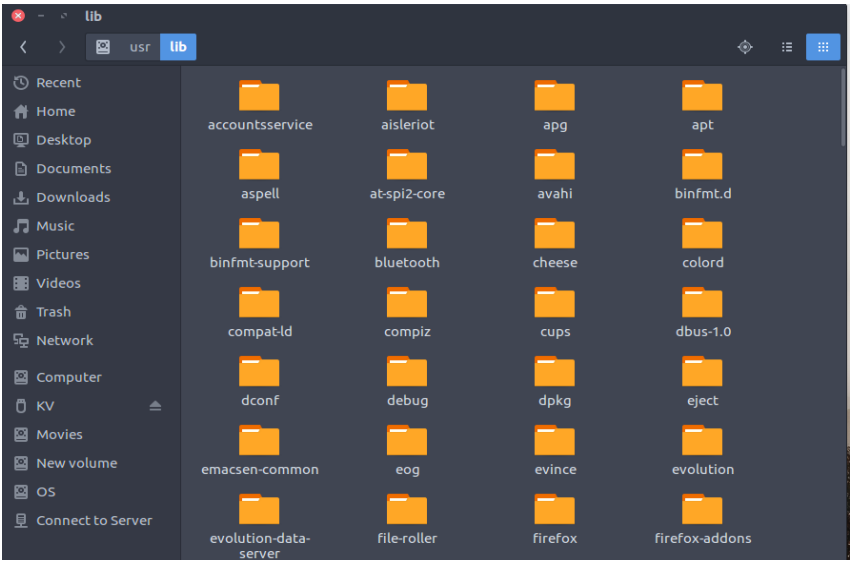
13. **/tmp :** Temporary files. Often not preserved between system reboots, and may be severely size restricted.
    - Directory that contains temporary files created by system and users.
    - Files under this directory are deleted when system is rebooted.

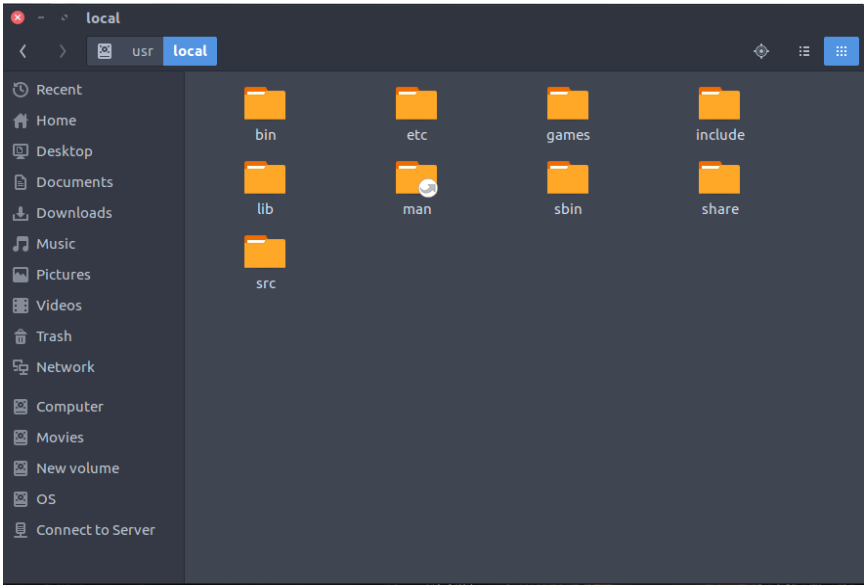


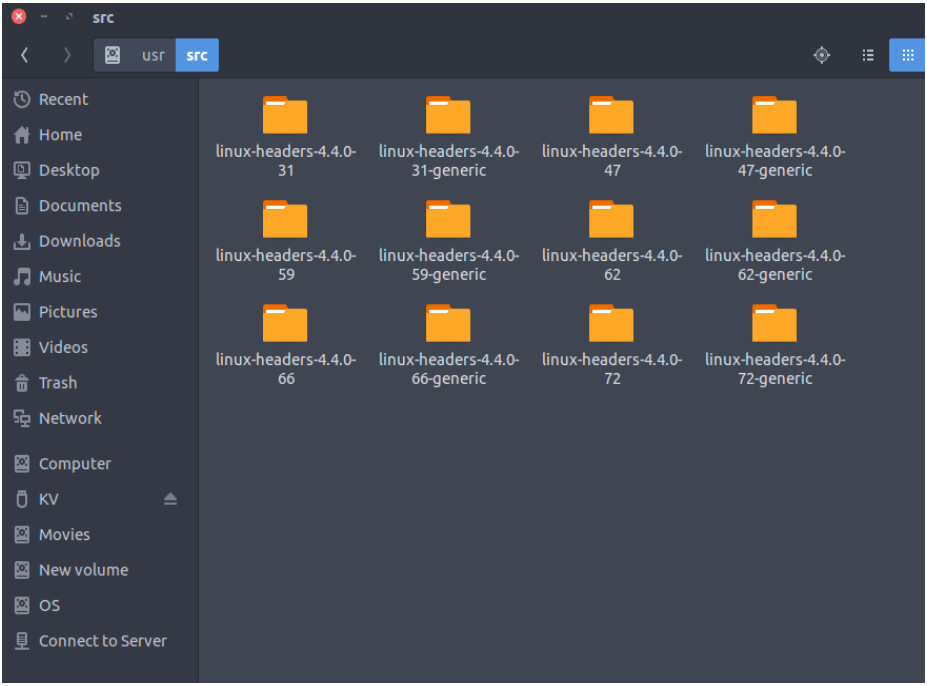
14. **/usr :** Secondary hierarchy for read-only user data; contains the majority of (multi-)user utilities and applications.
    - Contains binaries, libraries, documentation, and source-code for second level programs.
    - /usr/bin contains binary files for user programs. If you can't find a user binary under /bin, look under /usr/bin. For example: at, awk, cc, less, scp
    - /usr/sbin contains binary files for system administrators. If you can't find a system binary under /sbin, look under /usr/sbin. For example: atd, cron, sshd, useradd, userdel
    - /usr/lib contains libraries for /usr/bin and /usr/sbin
    - /usr/local contains users programs that you install from source. For example, when you install apache from source, it goes under /usr/local/apache2
    - /usr/src holds the Linux kernel sources, header-files and documentation.


bin
usr
bin
Recent
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
Trash
Network
Computer
KV
Movies
New volume
OS
Connect to Server
X11
[
2to3
2to3-2.7
2to3-3.5
7zr
411toppm
a11y-profile-manager-indicator
aa-enabled
aconnect
acpi_listen
activity-log-manager
add-apt-repository
addpart
addr2line
alsabat
alsaloop
alsamixer
alsatplg
alsaucm
amidi
amixer
amuFormat.sh
animate


sbin
usr
sbin
Recent
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
Trash
Network
Computer
KV
Movies
New volume
OS
Connect to Server
aa-exec
aa-remove-unknown
aa-status
accept
accessdb
acpid
addgnupghome
addgroup
add-shell
adduser
alsactl
alsa-info.sh
anacron
apparmor_status
applygnupgdefaults
aptd
arp
arpd
aspell-autobuildhash
avahi-autoipd
avahi-daemon
biosdecode
bluetoothd
chat

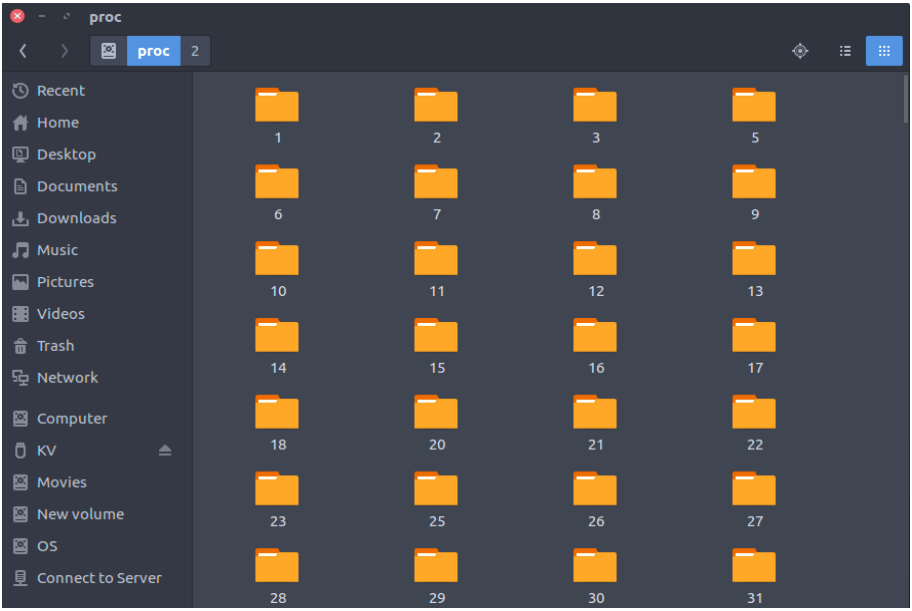

lib
usr
lib
Recent
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
Trash
Network
Computer
KV
Movies
New volume
OS
Connect to Server
accountsservice
aisleriot
apg
apt
aspell
at-spi2-core
avahi
binfmt.d
binfmt-support
bluetooth
cheese
colord
compat-ld
compiz
cups
dbus-1.0
dconf
debug
dpkg
eject
emacsen-common
eog
evince
evolution
evolution-data-server
file-roller
firefox
firefox-addons

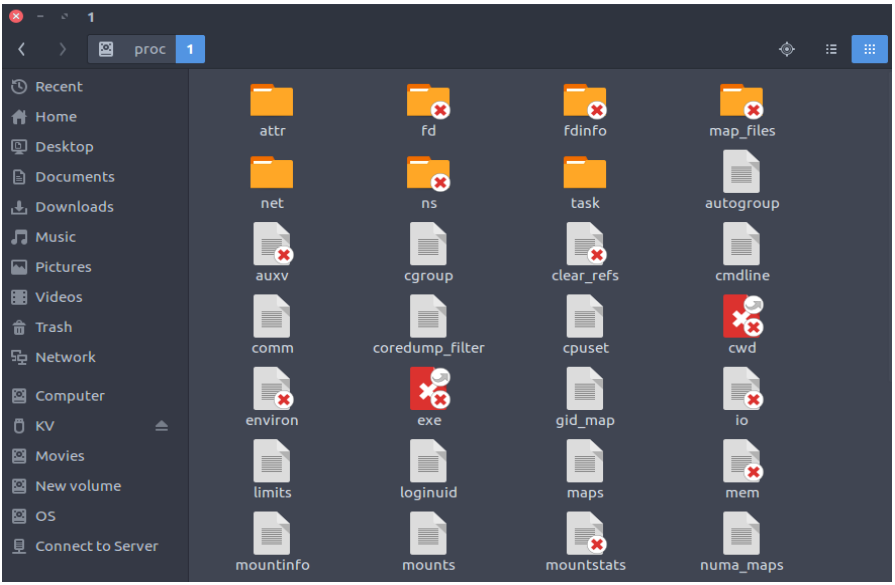

local
usr
local
Recent
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
Trash
Network
Computer
Movies
New volume
OS
Connect to Server
bin
etc
games
include
lib
man
sbin
share
src


src
usr
src
Recent
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
Trash
Network
Computer
KV
Movies
New volume
OS
Connect to Server
linux-headers-4.4.0-31
linux-headers-4.4.0-31-generic
linux-headers-4.4.0-47
linux-headers-4.4.0-47-generic
linux-headers-4.4.0-59
linux-headers-4.4.0-59-generic
linux-headers-4.4.0-62
linux-headers-4.4.0-62-generic
linux-headers-4.4.0-66
linux-headers-4.4.0-66-generic
linux-headers-4.4.0-72
linux-headers-4.4.0-72-generic

15. /proc : Virtual filesystem providing process and kernel information as files. In Linux, corresponds to a procfs mount. Generally automatically generated and populated by the system, on the fly.

- Contains information about system process.
- This is a pseudo filesystem contains information about running process. For example: /proc/{pid} directory contains information about the process with that particular pid.
- This is a virtual filesystem with text information about system resources. For example: /proc/uptime





Modern Linux distributions include a /run directory as a temporary filesystem (tmpfs) which stores volatile runtime data, following the FHS version 3.0. According to the FHS version 2.3, such data were stored in /var/run but this was a problem in some cases because this directory is not always available at early boot. As a result, these programs have had to resort to trickery, such as using /dev/.udev, /dev/.mdadm, /dev/.systems or /dev/.mount directories, even though the device directory isn't intended for such data.Among other advantages, this makes the system easier to use normally with the root filesystem mounted read-only. For example, below are the changes Debian made in its 2013 Wheezy release:

- /dev/.* ? /run/*
- /dev/shm ? /run/shm
- /dev/shm/* ? /run/*

- /etc/* (writable files) ? /run/*
- /lib/init/rw ? /run
- /var/lock ? /run/lock
- /var/run ? /run

Navigation Commands (cd, ls, and pwd)

In this lesson, we will go through Linux commands for managing directories and performing actions such as listing directories, printing the current working directory, creating and deleting directories, copying directories, transferring directories from one place to another, and renaming directories.

- **pwd.**

The pwd(print working directory) command is used to print out the current directory we are in.

The syntax is as follows,

```
pwd
```

Output.

```
/home/user
```

The output states that we are currently in the user directory which is the working directory.

- **ls.**

The ls(list directory) command is used to list files and subdirectories within a directory.

The basic syntax is as follows,

```
ls
```

Commonly used options include,

- ls -a, to print out all files including hidden files which are preceded with a period .
- ls -l, to list files in a long list. Files are listed together with their permissions, ownership, time stamps.
- ls -l --block-size=SIZE, to list files and directories while displaying their sizes.
  for example to list files and show sizes in megabytes we write,

```
ls -l --block-size=M
```

We could also use G for gigabytes, K for kilobytes, T for terabytes, P for petabytes and so on.

- ls -d */, to list only subdirectories, this command will exclude files in its listing.
- ls -g, to exclude owner information in the listing.
- ls -lG, to exclude group information in the listing.
- ls --color=COLOR, to add color, or remove color in the listing.

```
ls --color=never
```

To de-colorize the listing.

```
ls --color=auto
```

To color the listing.

- ls -F, lists files and subdirectories with distinction, it shows differences between files, directories, and executables.
- ls -R, to list subdirectories recursively.
- ls -Rl, list subdirectories recursively and in a long list.

- **cd.**

The cd(change directory) command is used to change the current working directory. This command enables users to navigate through the system's directories.

The syntax is as follows,

```
cd [DIRNAME]
```

- To change from the current directory to another directory we write,

```
cd /home/user/Desktop
```

We could also write,

```
cd ~/Desktop
```

since the ~ character represents the home directory.

- To change directory using an absolute path, we write the entire path starting from root / directory.
  An example

```
cd /etc/network/
```

/etc/network/ is an absolute path.

- To change a directory using a relative path, we write,

```
cd Desktop
```

Desktop is a relative path since it is relative to the current directory.

- To change to the previous directory we use - character,

```
cd -
```

If we were in Documents directory and change to Downloads, executing the command will take us back to Documents, if we execute it again, we will go back to Downloads.

- To change from the current directory to the parent directory of the working directory we write,

```
cd .
```

- To change to another user's directory, we write,

```
cd ~username
```

We can also change into subdirectories as follows,

```
cd /home/user/Desktop/files/
```

- We can also change directories as follows,

```
user:~$ cd Documents/Dir1/Dir2/Dir3
user:~/Documents/Dir1/Dir2/Dir3$ cd ..
user:~/Documents/Dir1/Dir2$ cd ..
user:~/Documents/Dir1$ cd ..
user:~/Documents$
```

- **mkdir.**

The mkdir(make directory) command is used to create a new directory.
The syntax is as follows,

```
mkdir [DIRNAME]
```

- To create a single directory we write,

```
mkdir newDir
```

- To create multiple directories we write,

```
mkdir newDir1 newDir2 newDir2
```

- To print output of the command we use the -v option,

```
$ mkdir newDir
mkdir: created directory 'newDir'
```

- To create new subdirectories inside a newly created directory or an existing one we use the -p option. If the directory exists, no error will be printed, otherwise, new subdirectories will be created inside the directory.

```
mkdir -p newDir/dir1 newDir/dir2
```

- Create a directory while specifying permissions we use the -m, --mode=MODE option,
  An example
  To create anew directory with full access(read, write, execute) we write,

```
mkdir -m 777 newDir
```

- **rmdir.**

The rmdir(remove directory) command is used to delete an empty directory.
The syntax is as follows,

```
rmdir [DIRNAME]
```

To remove a non-empty directory, that is a directory with files and subdirectories we use the rm -r dirname command.

- **mv.**

The mv(move) command is used to move files and directories in Linux, it can also be used to rename files and directories.
The syntax is as follows,

```
mv [OLD] [NEW]
```

- To change the name of NewDir to ChangedDir we write,

```
mv newDir ChangedDir
```

- To move a directory/file we can write,

```
mv newDir ~/Documents/Dirs
```

This command will remove newDir from the current directory and move it to ~/Documents/Dirs.

- **cp.**

The cp(copy) command is used to copy files or directories.

```
cp -r newDir ~/Documents/Dirs
```

The command will copy newDir directory into ~/Documents/Dirs. Note that newDir will not be removed from where it is. It is only copied.
We use the -r to recursively copy files and subdirectories.
To copy files we write,

```
cp file.txt ~/DocumentsDirs
```