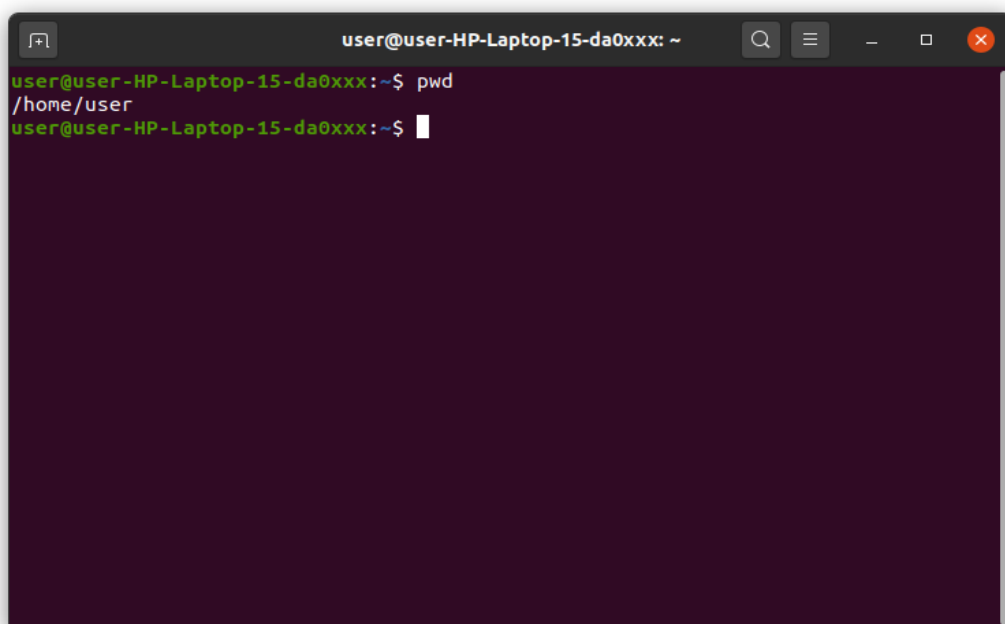


BASIC LINUX COMMANDS

1. pwd

Use the pwd command to find out the path of the current working directory (folder) you're in. The command will return an absolute (full) path, which is basically a path of all the directories that start with a forward slash (/). An example of an absolute path is /home/username.

A terminal window titled 'user@user-HP-Laptop-15-da0xxx: ~' with standard window controls. The terminal shows the command 'pwd' being entered and executed, resulting in the output '/home/user'.

```
user@user-HP-Laptop-15-da0xxx: ~  
user@user-HP-Laptop-15-da0xxx:~$ pwd  
/home/user  
user@user-HP-Laptop-15-da0xxx:~$
```

2. cd

To navigate through the Linux files and directories, use the cd . It requires either the full path or the name of the directory, depending on the current working directory that you're in.

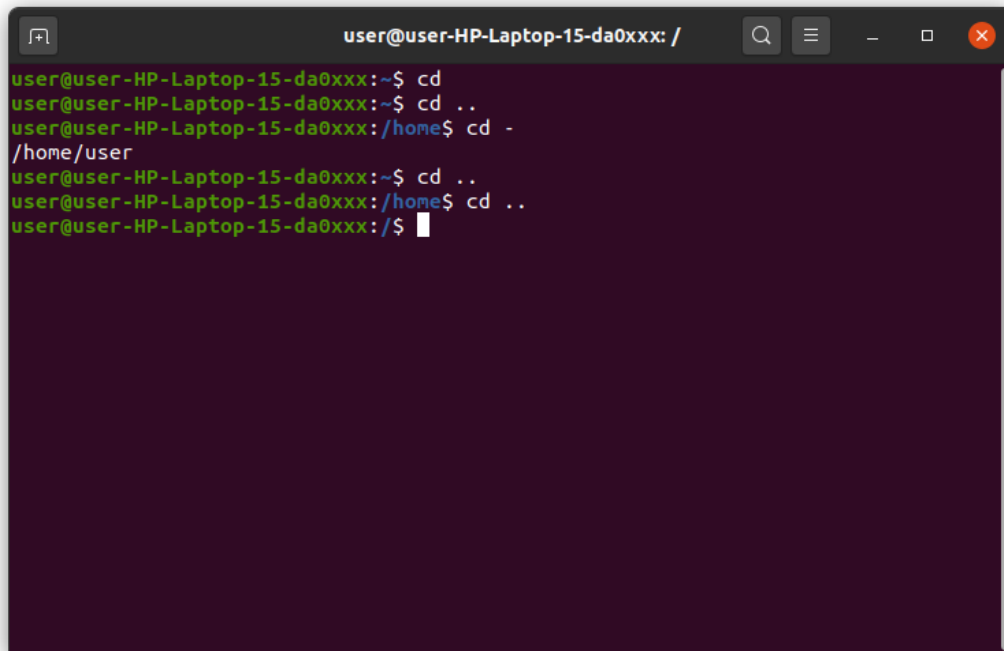
Let's say you're in /home/username/Documents and you want to go to Photos, a subdirectory of Documents. To do so, simply type the following command: cd Photos.

Another scenario is if you want to switch to a completely new directory, for example, /home/username/Movies. In this case, you have to type cd followed by the directory's absolute path: cd /home/username/Movies.

There are some shortcuts to help you navigate quickly:

- `cd ..` (with two dots) to move one directory up
- `cd` to go straight to the home folder
- `cd-` (with a hyphen) to move to your previous directory

On a side note, Linux's shell is case sensitive. So, you have to type the name's directory exactly as it is.

A terminal window titled 'user@user-HP-Laptop-15-da0xxx: /' with standard window controls. The terminal shows a sequence of commands and their outputs: 1. 'cd' leads to a new prompt. 2. 'cd ..' leads to a new prompt. 3. 'cd -' leads to the output '/home/user' and a new prompt. 4. 'cd ..' leads to a new prompt. 5. 'cd ..' leads to a new prompt. 6. The final prompt is shown with a cursor. The terminal has a dark purple background and green text.

```
user@user-HP-Laptop-15-da0xxx:~$ cd
user@user-HP-Laptop-15-da0xxx:~$ cd ..
user@user-HP-Laptop-15-da0xxx:/home$ cd -
/home/user
user@user-HP-Laptop-15-da0xxx:~$ cd ..
user@user-HP-Laptop-15-da0xxx:/home$ cd ..
user@user-HP-Laptop-15-da0xxx:/$
```

3. ls

The `ls` command is used to view the contents of a directory. By default, this command will display the contents of your current working directory.

If you want to see the content of other directories, type `ls` and then the directory's path. For example, enter `ls /home/username/Documents` to view the content of Documents.

There are variations you can use with the `ls` command:

- `ls -R` will list all the files in the sub-directories as well
- `ls -a` will show the hidden files
- `ls -al` will list the files and directories with detailed information like the permissions, size, owner, etc.
- `ls -t` lists files sorted in the order of “last modified”
- `-r` option will reverse the natural sorting order. Usually used in combination with other switches such as `ls -tr`. This will reverse the time-wise listing.

```
user@user-HP-Laptop-15-da0xxx: ~  
user@user-HP-Laptop-15-da0xxx:~$ ls  
bitstring.c  Desktop  GIT  Pictures  Templates  
cn  Documents  Music  prims.c  Videos  
cpgms  Downloads  '#newfile.txt#'  Public  
user@user-HP-Laptop-15-da0xxx:~$
```

```
user@user-HP-Laptop-15-da0xxx:~$ ls -R  
.:  
bitstring.c  Desktop  GIT  Pictures  Templates  
cn  Documents  Music  prims.c  Videos  
cpgms  Downloads  '#newfile.txt#'  Public  
./cn:  
a.1.png 'd$.2.png' dw.1.png I.2.png l.1.png O.3.png  
A.1.png 'd^1.png' dw.2.png I.3.png l.2.png x.1.png  
a.2.png D.1.png emacs.png i.png o.1.png X.1.png  
A.2.png 'd^2.png' 'emacs version.png' j.1.png O.1.png x.2.png  
a.3.png D.2.png h.1.png j.2.png o.2.png X.2.png  
A.3.png dd.1.png h.2.png k.1.png O.2.png  
'd$.1.png' dd.2.png I.1.png k.2.png o.3.png  
./cpgms:  
a.c dfs.c kruskal.c singledeletion.out  
a.out dfs.out kruskal.out singlell.c  
bbits.c dijkstra.c linkedlists  
bbits.out disjoint.c llfinal.c sma.c  
bfs.c disjoint.out llfinal.c smallarray.c  
bfs.out display.c mergesort.c smallarray.out  
binomial.c display.out mergesort.out smallesst.c  
binomial.out doublyll.c mul.c smallesst.out  
bits.c doublyll.out mul.out sma.out  
bitsfinal.c ds.c oebig.c ssh.....  
bitsfinal.out ds.out oebig.out stackarray.c  
bits.out ds.out prims.c stackarray.out  
bitstring duply.c prims.out stackll.c  
bitstring.c evenoddsun.c queuearray.c stackll.out  
bitstringorg.c evenoddsun.out queuearray.out stack.out  
bitstringorg.out evenodsum.c queueell.c sum.c  
bitstring.out evenodsum.out queueell.out sum.out  
'bridge_course' great.c rbtrees.c topological.c  
bst.c great.out rbtrees.c topological.out  
bst.out lnfront.c rectcircle.c tree.c  
btrees.c lnfront.out rectcircle.out ttt.c  
btrees.out lnfront.c reversearr.c ttt.out  
circular.c lnfront.out reversearr.out 'Untitled Document 1.c'
```

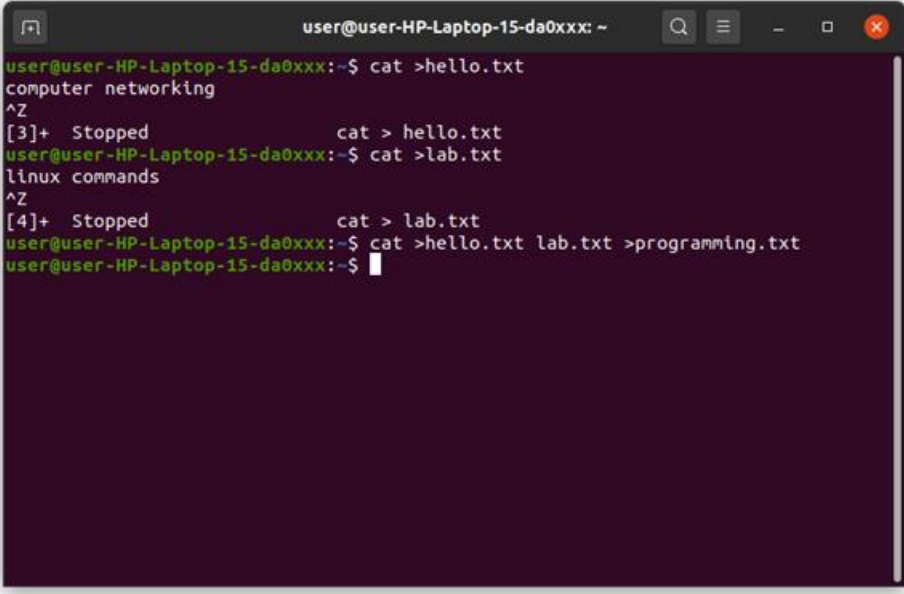
```
user@user-HP-Laptop-15-da0xxx:~$ ls -al  
total 160  
drwxr-xr-x 20 user user 4096 Jun 11 02:08 .  
drwxr-xr-x 3 root root 4096 Dec 1 2020 ..  
-rw-r--r-- 1 user user 10016 Jun 11 02:20 .bash_history  
-rw-r--r-- 1 user user 220 Dec 1 2020 .bash_logout  
-rw-r--r-- 1 user user 3771 Dec 1 2020 .bashrc  
-rw-rw-r-- 1 user user 458 Jan 3 00:59 bitstring.c  
drwxr-xr-x 16 user user 4096 Jun 10 03:39 .cache  
drwxrwxr-x 2 user user 4096 Jun 11 02:18 cn  
drwx----- 16 user user 4096 Jun 10 03:39 .config  
drwxrwxr-x 5 user user 4096 Mar 21 14:32 cpgms  
drwxr-xr-x 2 user user 4096 Dec 2 2020 Desktop  
drwxr-xr-x 2 user user 4096 Dec 2 2020 Documents  
drwxr-xr-x 2 user user 4096 Jun 11 17:27 Downloads  
drwx----- 3 user user 4096 Jun 10 03:47 .emacs.d  
drwxrwxr-x 3 user user 4096 Dec 28 01:32 GIT  
-rw-rw-r-- 1 user user 58 Dec 25 21:00 .gitconfig  
drwx----- 3 user user 4096 Jun 11 02:19 .gnupg  
drwxr-xr-x 3 user user 4096 Dec 2 2020 .local  
drwx----- 5 user user 4096 Dec 5 2020 .mozilla  
drwxr-xr-x 2 user user 4096 Dec 2 2020 .music  
-rw-rw-r-- 1 user user 29 Jun 10 03:58 '#newfile.txt#'  
drwxr-xr-x 4 user user 4096 Jun 11 17:28 Pictures  
-rw-rw-r-- 1 user user 1625 Mar 21 14:18 prims.c  
-rw-r--r-- 1 user user 807 Dec 1 2020 .profile  
drwxr-xr-x 2 user user 4096 Dec 2 2020 Public  
drwx----- 2 user user 4096 Dec 26 21:35 .ssh  
-rw-r--r-- 1 user user 0 Dec 2 2020 .sudo_as_admin_successful  
-rw----- 1 user user 12288 Jun 11 02:08 .swm  
-rw----- 1 user user 12288 Jun 10 04:34 .swp  
drwxr-xr-x 2 user user 4096 Dec 2 2020 Templates  
drwxr-xr-x 2 user user 4096 Dec 2 2020 Videos  
-rw----- 1 user user 782 Jun 11 02:08 .viminfo  
user@user-HP-Laptop-15-da0xxx:~$
```

4. cat :



cat (short for concatenate) is one of the most frequently used commands in Linux. It is used to list the contents of a file on the standard output stdout . To run this command, type cat followed by the file's name and its extension. For instance: cat file.txt.



Here are other ways to use the cat command:

- cat > filename creates a new file
- cat filename1 filename2>filename3 joins two files (1 and 2) and stores the output of them in a new file (3)
- to convert a file to upper or lower case use, cat filename | tr a-z A-Z >output.txt



A terminal window titled 'user@user-HP-Laptop-15-da0xxx: ~' with standard window controls. The terminal shows a sequence of commands and their outputs: 1. 'cat >hello.txt' followed by 'computer networking' and '^Z'. 2. '[3]+ Stopped cat > hello.txt'. 3. 'user@user-HP-Laptop-15-da0xxx:~\$ cat >lab.txt' followed by 'linux commands' and '^Z'. 4. '[4]+ Stopped cat > lab.txt'. 5. 'user@user-HP-Laptop-15-da0xxx:~\$ cat >hello.txt lab.txt >programming.txt'. 6. The prompt 'user@user-HP-Laptop-15-da0xxx:~\$' is shown again with a cursor.

```
user@user-HP-Laptop-15-da0xxx: ~
user@user-HP-Laptop-15-da0xxx:~$ cat >hello.txt
computer networking
^Z
[3]+  Stopped                  cat > hello.txt
user@user-HP-Laptop-15-da0xxx:~$ cat >lab.txt
linux commands
^Z
[4]+  Stopped                  cat > lab.txt
user@user-HP-Laptop-15-da0xxx:~$ cat >hello.txt lab.txt >programming.txt
user@user-HP-Laptop-15-da0xxx:~$
```

Open ▾  hello.txt Save   

File1  hello.txt 

1 computer networking

Plain Text ▾ Tab Width: 8 ▾ Ln 1, Col 1  

Open ▾  lab.txt Save   

1 linux commands

Plain Text ▾ Tab Width: 8 ▾ Ln 1, Col 1  

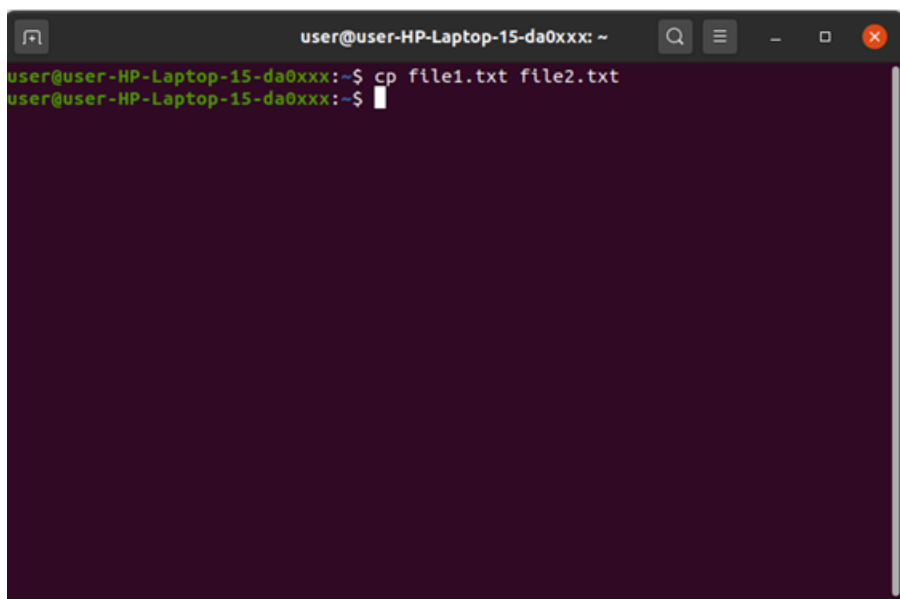
Open ▾  programming.txt Save   

1 computer networking
2 linux commands

5. cp

Use the cp command to copy files from the current directory to a different directory. For instance, the command `cp scenery.jpg /home/username/Pictures` would create a copy of scenery.jpg (from your current directory) into the Pictures directory.

- `cp -i` will ask for user's consent in case of a potential file overwrite.
- `cp -p` will preserve source files' mode, ownership and timestamp.
- `cp -r` will copy directories recursively.
- `cp -u` copies files only if the destination file is not existing or the source file is newer than the destination file.



```
user@user-HP-Laptop-15-da0xxx: ~  
user@user-HP-Laptop-15-da0xxx:~$ cp file1.txt file2.txt  
user@user-HP-Laptop-15-da0xxx:~$
```



```
Open file1.txt Save  
1 computer networks  
Plain Text Tab Width: 8 Ln 1, Col 1 INS
```

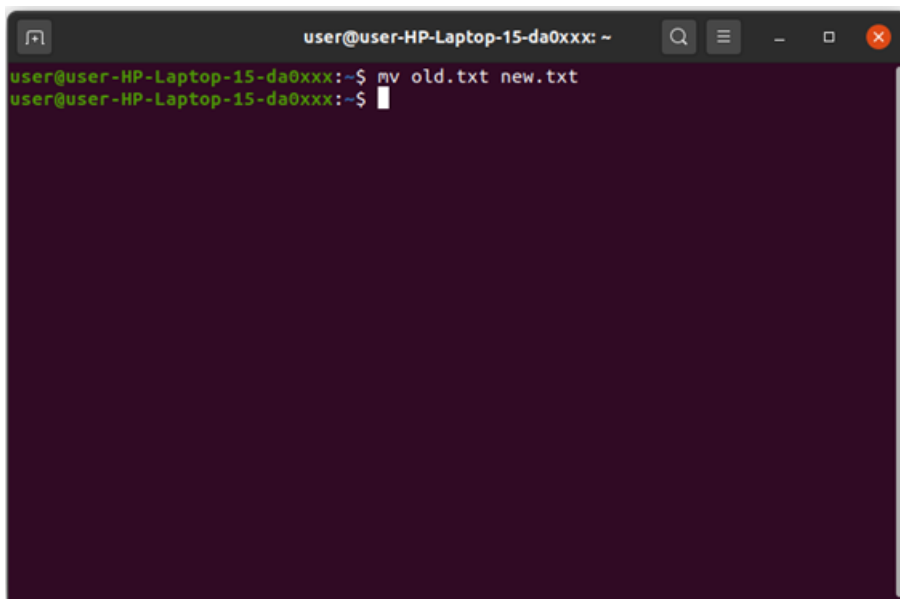


6 mv

The primary use of the mv command is to move files, although it can also be used to rename files.

The arguments in mv are similar to the cp command. You need to type mv, the file's name, and the destination's directory. For example: mv file.txt /home/username/Documents.

To rename files, the Linux is mv oldname.ext newname.ext

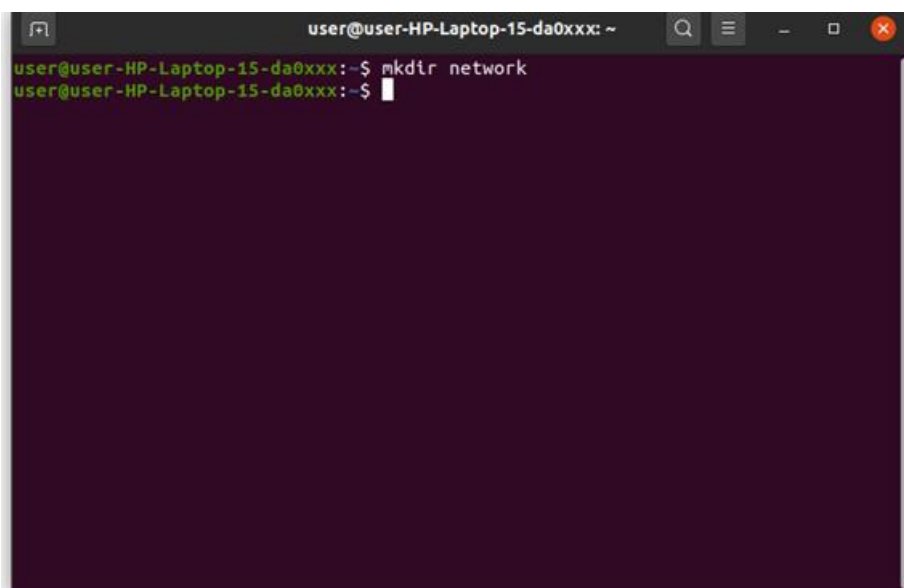




7. mkdir

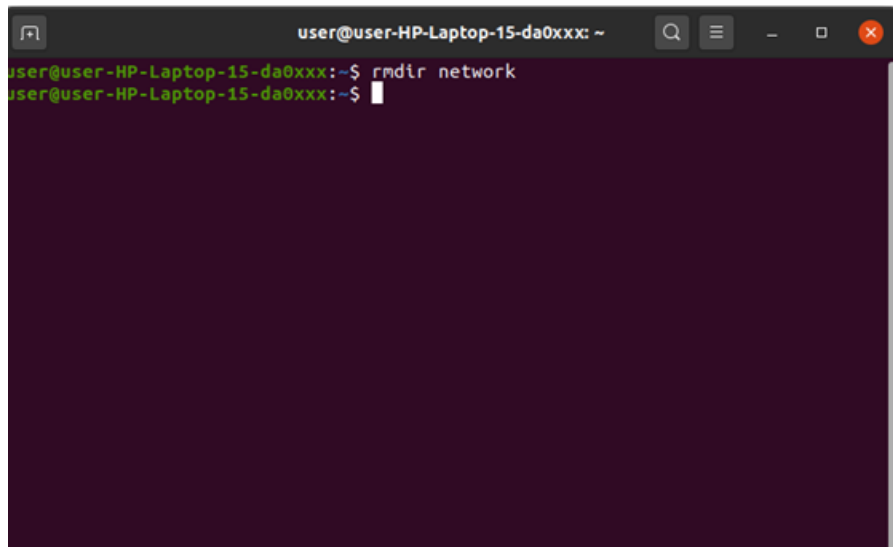
Use mkdir command to make a new directory — if you type mkdir Music it will create a directory called Music. here are extra mkdir commands as well:

- To generate a new directory inside another directory, use this Linux basic command mkdir Music/Newfile
- use the p (parents) option to create a directory in between two existing directories. For example, mkdir -p Music/2020/Newfile will create the new “2020” file.



8. rmdir

If you need to delete a directory, use the `rmdir` command. However, `rmdir` only allows you to delete empty directories.

A terminal window with a dark purple background. The title bar at the top reads "user@user-HP-Laptop-15-da0xxx: ~". The terminal shows two lines of text: the first line is "user@user-HP-Laptop-15-da0xxx:~\$ rmdir network" and the second line is "user@user-HP-Laptop-15-da0xxx:~\$ " followed by a white cursor. The window has standard Linux window controls (minimize, maximize, close) on the right side.

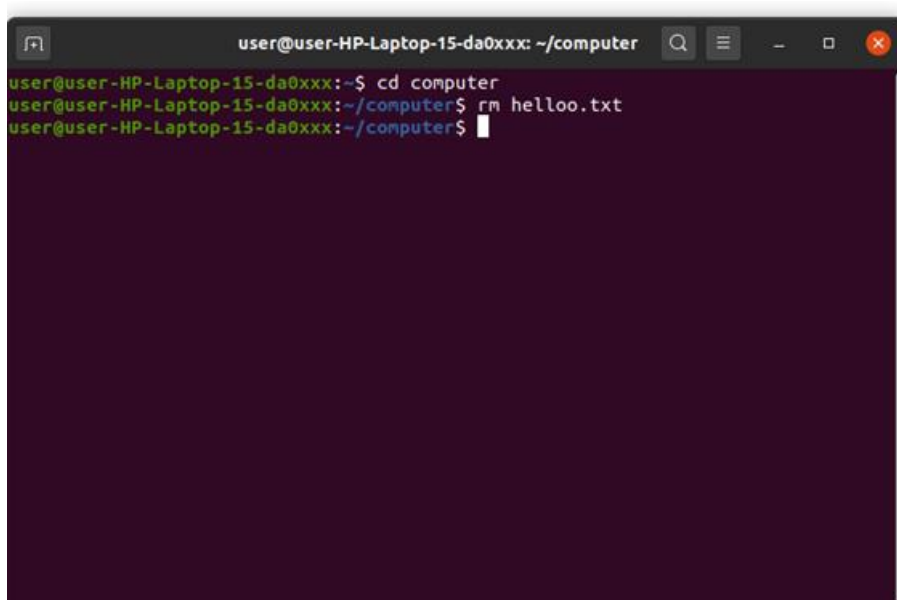
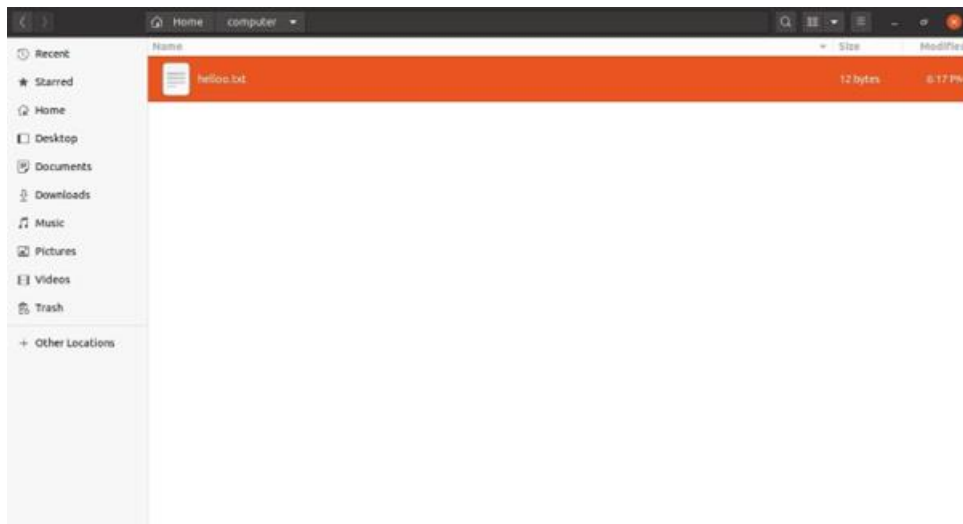
```
user@user-HP-Laptop-15-da0xxx:~$ rmdir network
user@user-HP-Laptop-15-da0xxx:~$
```

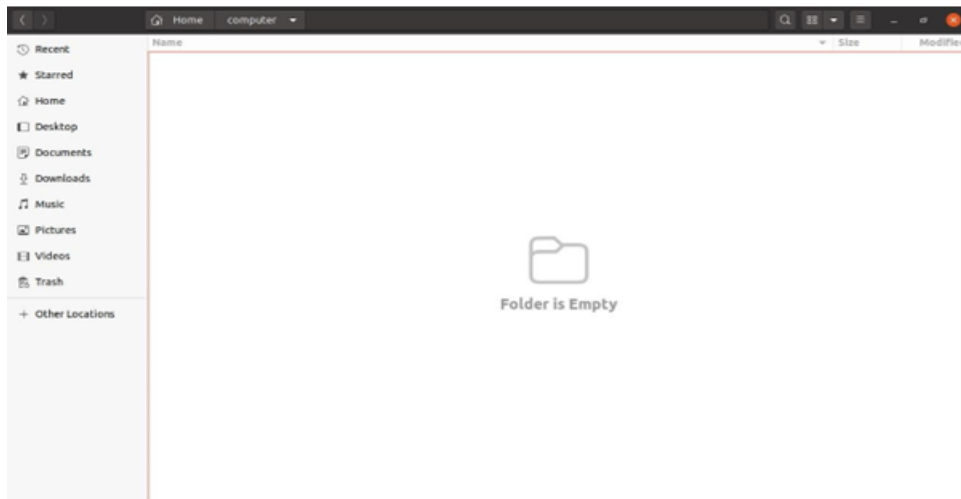
9. rm

The `rm` command is used to delete directories and the contents within them. If you only want to delete the directory — as an alternative to `rmdir` use `rm -r`.

Note: Be very careful with this command and double-check which directory you are in.

This will delete everything and there is no undo.

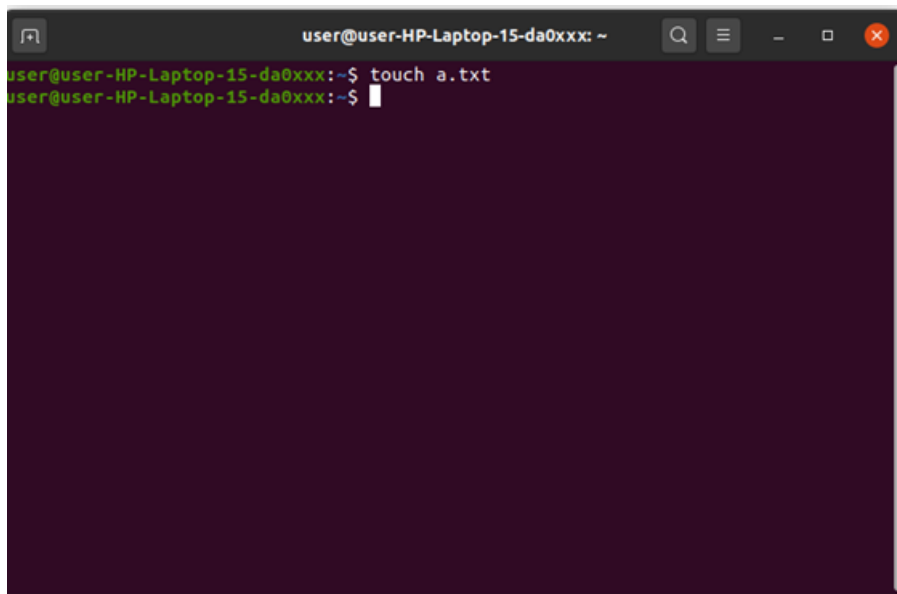




10. touch

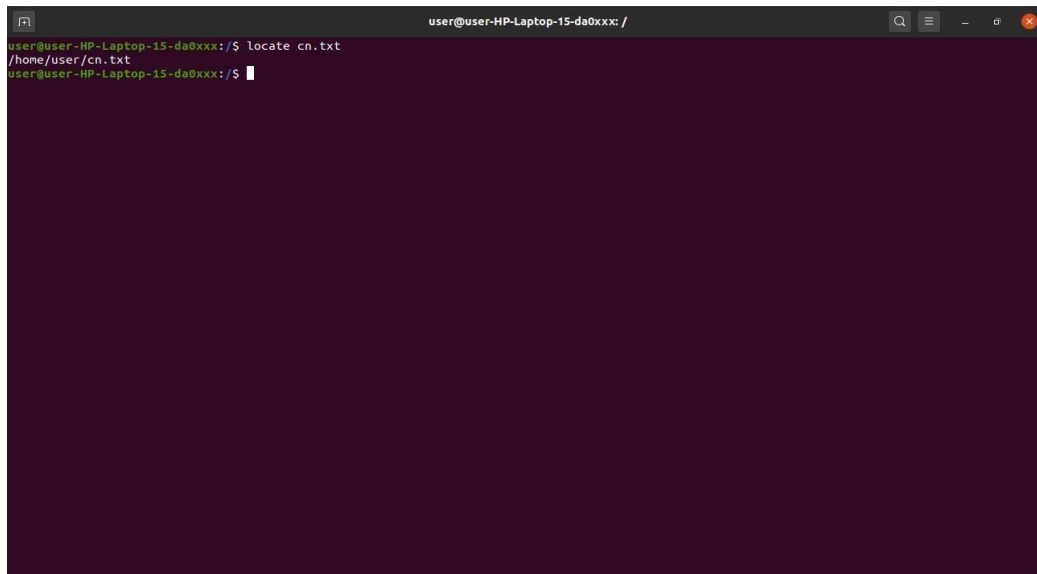
The touch command allows you to create a blank new file through the Linux command line.

As an example, enter `touch /home/username/Documents/Web.html` to create an HTML file entitled Web under the Documents directory.



11. locate

You can use this command to locate a file, just like the search command in Windows. What's more, using the `-i` argument along with this command will make it caseinsensitive, so you can search for a file even if you don't remember its exact name. To search for a file that contains two or more words, use an asterisk (*). For example, `locate -i school*note` command will search for any file that contains the word "school" and "note", whether it is uppercase or lowercase.

A terminal window with a dark purple background. The title bar at the top reads "user@user-HP-Laptop-15-da0xxx: /". The terminal shows the following commands and output:

```
user@user-HP-Laptop-15-da0xxx:/$ locate cn.txt
/home/user/cn.txt
user@user-HP-Laptop-15-da0xxx:/$
```

12. find

Similar to the locate command, using find also searches for files and directories. The difference is, you use the find command to locate files within a given directory.

As an example, `find /home/ -name notes.txt` command will search for a file called notes.txt within the home directory and its subdirectories.

Other variations when using the find are:

- To find files in the current directory use, `find . -name notes.txt`
- To look for directories use, `/ -type d -name notes.txt`

```
user@user-HP-Laptop-15-da0xxx: /home
user@user-HP-Laptop-15-da0xxx:~$ cd ..
user@user-HP-Laptop-15-da0xxx:/home$ find /home/ -name cn.txt
/home/user/cn.txt
user@user-HP-Laptop-15-da0xxx:/home$
```

13. grep

Another basic Linux command that is undoubtedly helpful for everyday use is grep. It lets you search through all the text in a given file.

To illustrate, grep blue notepad.txt will search for the word blue in the notepad file. Lines that contain the searched word will be displayed fully. You should refer to some grep tutorial

Useful for command line use as well. Usually output of a previous command is piped into the grep command. For example `ls -l | grep "kernel"`

```
user@user-HP-Laptop-15-da0xxx: ~
user@user-HP-Laptop-15-da0xxx:~$ cat file1.txt
computer networks
user@user-HP-Laptop-15-da0xxx:~$ grep networks file1.txt
computer networks
user@user-HP-Laptop-15-da0xxx:~$
```

14. sudo

Short for “SuperUser Do”, this command enables you to perform tasks that require administrative or root permissions. You must have sufficient permissions to use this command.

```
user@user-HP-Laptop-15-da0xxx:~$ sudo -h
sudo - execute a command as another user

usage: sudo -h | -K | -k | -V
usage: sudo -v [-Aknst] [-g group] [-h host] [-p prompt] [-u user]
usage: sudo -l [-Aknst] [-g group] [-h host] [-p prompt] [-u user] [-v user]
[command]
usage: sudo [-AbCHKknpst] [-r role] [-t type] [-C num] [-g group] [-h host] [-p
prompt] [-i timeout] [-u user] [VM=value] [-ll-s] [-command]
usage: sudo -e [-Aknst] [-r role] [-t type] [-C num] [-g group] [-h host] [-p
prompt] [-i timeout] [-u user] file ...

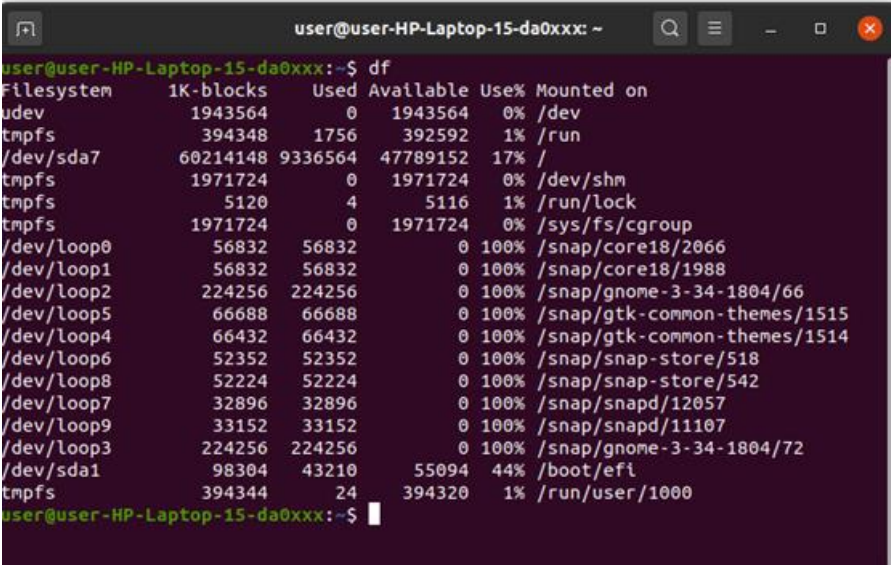
Options:
-A, --askpass          use a helper program for password prompting
-b, --background      run command in the background
-B, --bell            ring bell when prompting
-C, --close-from-num  close all file descriptors >= num
-E, --preserve-env     preserve user environment when running command
    --preserve-env-list
                        preserve specific environment variables
-e, --edit            edit files instead of running a command
-g, --group=group     run command as the specified group name or ID
-H, --set-home        set HOME variable to target user's home dir
-h, --help            display help message and exit
-H, --host=host       run command on host (if supported by plugin)
-l, --login           run login shell as the target user; a command
                        may also be specified
-K, --remove-timestamp
                        remove timestamp file completely
-k, --reset-timestamp
                        invalidate timestamp file
-l, --list            list user's privileges or check a specific
                        command; use twice for longer format
-n, --non-interactive
                        non-interactive mode, no prompts are used
-P, --preserve-groups
                        preserve group vector instead of setting to
                        target's
-p, --prompt=prompt   use the specified password prompt
-r, --role=role       create SELinux security context with specified
                        role
-S, --stdin           read password from standard input
-s, --shell           run shell as the target user; a command may
                        also be specified
```

```
user@user-HP-Laptop-15-da0xxx:~$ sudo -h
Options:
-A, --askpass          use a helper program for password prompting
-b, --background      run command in the background
-B, --bell            ring bell when prompting
-C, --close-from-num  close all file descriptors >= num
-E, --preserve-env     preserve user environment when running command
    --preserve-env-list
                        preserve specific environment variables
-e, --edit            edit files instead of running a command
-g, --group=group     run command as the specified group name or ID
-H, --set-home        set HOME variable to target user's home dir
-h, --help            display help message and exit
-H, --host=host       run command on host (if supported by plugin)
-l, --login           run login shell as the target user; a command
                        may also be specified
-K, --remove-timestamp
                        remove timestamp file completely
-k, --reset-timestamp
                        invalidate timestamp file
-l, --list            list user's privileges or check a specific
                        command; use twice for longer format
-n, --non-interactive
                        non-interactive mode, no prompts are used
-P, --preserve-groups
                        preserve group vector instead of setting to
                        target's
-p, --prompt=prompt   use the specified password prompt
-r, --role=role       create SELinux security context with specified
                        role
-S, --stdin           read password from standard input
-s, --shell           run shell as the target user; a command may
                        also be specified
-t, --type=type       create SELinux security context with specified
                        type
-T, --command-timeout=timeout
                        terminate command after the specified time limit
-U, --other-user=user
                        in list mode, display privileges for user
-u, --user=user       run command (or edit file) as specified user
                        name or ID
-V, --version         display version information and exit
-v, --validate        update user's timestamp without running a
                        command
--                  stop processing command line arguments

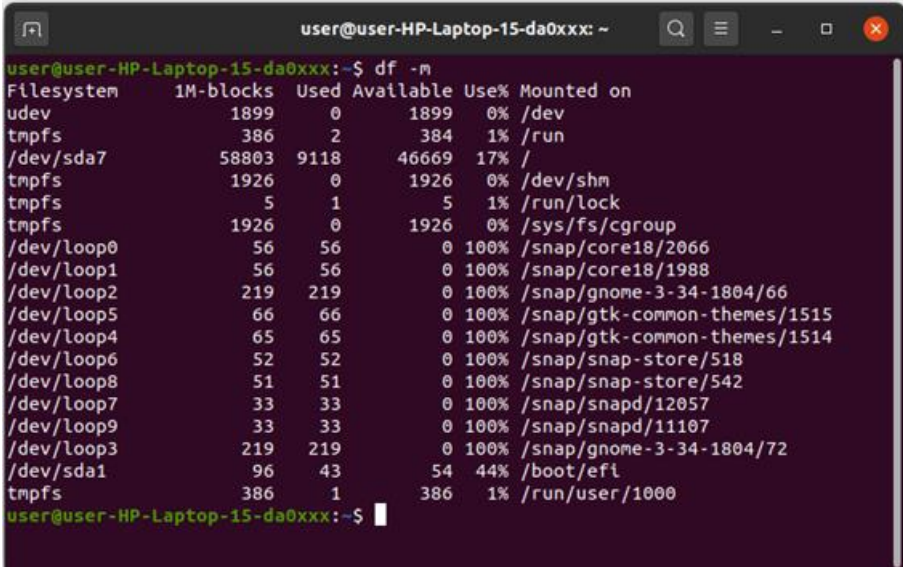
user@user-HP-Laptop-15-da0xxx:~$
```

15. df

Use df command to get a report on the system's disk space usage, shown in percentage and KBs. If you want to see the report in megabytes, type df -m.



```
user@user-HP-Laptop-15-da0xxx: ~  
user@user-HP-Laptop-15-da0xxx:~$ df  
Filesystem      1K-blocks    Used Available Use% Mounted on  
udev            1943564      0    1943564   0% /dev  
tmpfs           394348      1756    392592   1% /run  
/dev/sda7       60214148  9336564  47789152  17% /  
tmpfs           1971724      0    1971724   0% /dev/shm  
tmpfs           5120        4      5116    1% /run/lock  
tmpfs           1971724      0    1971724   0% /sys/fs/cgroup  
/dev/loop0      56832      56832      0 100% /snap/core18/2066  
/dev/loop1      56832      56832      0 100% /snap/core18/1988  
/dev/loop2      224256     224256      0 100% /snap/gnome-3-34-1804/66  
/dev/loop5      66688      66688      0 100% /snap/gtk-common-themes/1515  
/dev/loop4      66432      66432      0 100% /snap/gtk-common-themes/1514  
/dev/loop6      52352      52352      0 100% /snap/snap-store/518  
/dev/loop8      52224      52224      0 100% /snap/snap-store/542  
/dev/loop7      32896      32896      0 100% /snap/snapd/12057  
/dev/loop9      33152      33152      0 100% /snap/snapd/11107  
/dev/loop3      224256     224256      0 100% /snap/gnome-3-34-1804/72  
/dev/sda1       98304      43210    55094   44% /boot/efi  
tmpfs           394344      24    394320   1% /run/user/1000  
user@user-HP-Laptop-15-da0xxx:~$
```



```
user@user-HP-Laptop-15-da0xxx: ~  
user@user-HP-Laptop-15-da0xxx:~$ df -m  
Filesystem      1M-blocks    Used Available Use% Mounted on  
udev            1899      0    1899   0% /dev  
tmpfs           386      2     384   1% /run  
/dev/sda7       58803    9118   46669  17% /  
tmpfs           1926      0    1926   0% /dev/shm  
tmpfs           5        1      5    1% /run/lock  
tmpfs           1926      0    1926   0% /sys/fs/cgroup  
/dev/loop0      56      56      0 100% /snap/core18/2066  
/dev/loop1      56      56      0 100% /snap/core18/1988  
/dev/loop2      219     219      0 100% /snap/gnome-3-34-1804/66  
/dev/loop5      66      66      0 100% /snap/gtk-common-themes/1515  
/dev/loop4      65      65      0 100% /snap/gtk-common-themes/1514  
/dev/loop6      52      52      0 100% /snap/snap-store/518  
/dev/loop8      51      51      0 100% /snap/snap-store/542  
/dev/loop7      33      33      0 100% /snap/snapd/12057  
/dev/loop9      33      33      0 100% /snap/snapd/11107  
/dev/loop3      219     219      0 100% /snap/gnome-3-34-1804/72  
/dev/sda1       96      43     54   44% /boot/efi  
tmpfs           386      1     386   1% /run/user/1000  
user@user-HP-Laptop-15-da0xxx:~$
```

16. du

If you want to check how much space a file or a directory takes, the du (Disk Usage) command is the answer. However, the disk usage summary will show disk block numbers instead of the usual size format. If you want to see it in bytes, kilobytes, and megabytes, add the -h argument to the command line.

```
user@user-HP-Laptop-15-dabxxx:~$ du
4      ./Videos
1768   ./Pictures/commands
3736   ./Pictures/outputs
232    ./Pictures/Downloads
13360  ./Pictures
16     ./ssh
4      ./enacs.d/auto-save-llist
8      ./enacs.d
4      ./Documents
968    ./cn
4      ./Public
408    ./local/share/tracker/data
412    ./local/share/tracker
72     ./local/share/xorg
4      ./local/share/webkitgtk/deviceidhashsalts/1
8      ./local/share/webkitgtk/deviceidhashsalts
4      ./local/share/webkitgtk/localstorage
4      ./local/share/webkitgtk/databases/indexeddb/v1
8      ./local/share/webkitgtk/databases/indexeddb
12     ./local/share/webkitgtk/databases
40     ./local/share/webkitgtk
4      ./local/share/ibus-table
4      ./local/share/sounds
12     ./local/share/gnome-shell
4      ./local/share/evolution/nautilus/trash
8      ./local/share/evolution/nautilus
4      ./local/share/evolution/addressbook/system/photos
92     ./local/share/evolution/addressbook/system
4      ./local/share/evolution/addressbook/trash
100    ./local/share/evolution/addressbook
8      ./local/share/evolution/calendar/system
4      ./local/share/evolution/calendar/trash
16     ./local/share/evolution/calendar
4      ./local/share/evolution/nemos/trash
8      ./local/share/evolution/nemos
4      ./local/share/evolution/tasks/system
4      ./local/share/evolution/tasks/trash
16     ./local/share/evolution/tasks
```

```
user@user-HP-Laptop-15-dabxxx:~$ du -h
4.0K   ./Videos
2.9M   ./Pictures/commands
3.7M   ./Pictures/outputs
232K   ./Pictures/Downloads
13M    ./Pictures
16K    ./ssh
4.0K   ./enacs.d/auto-save-llist
8.0K   ./enacs.d
4.0K   ./Documents
968K   ./cn
4.0K   ./Public
412K   ./local/share/tracker/data
416K   ./local/share/tracker
72K    ./local/share/xorg
4.0K   ./local/share/webkitgtk/deviceidhashsalts/1
8.0K   ./local/share/webkitgtk/deviceidhashsalts
4.0K   ./local/share/webkitgtk/localstorage
4.0K   ./local/share/webkitgtk/databases/indexeddb/v1
8.0K   ./local/share/webkitgtk/databases/indexeddb
12K    ./local/share/webkitgtk/databases
40K    ./local/share/webkitgtk
4.0K   ./local/share/ibus-table
4.0K   ./local/share/sounds
12K    ./local/share/gnome-shell
4.0K   ./local/share/evolution/nautilus/trash
8.0K   ./local/share/evolution/nautilus
4.0K   ./local/share/evolution/addressbook/system/photos
92K    ./local/share/evolution/addressbook/system
4.0K   ./local/share/evolution/addressbook/trash
100K   ./local/share/evolution/addressbook
8.0K   ./local/share/evolution/calendar/system
4.0K   ./local/share/evolution/calendar/trash
16K    ./local/share/evolution/calendar
4.0K   ./local/share/evolution/nemos/trash
8.0K   ./local/share/evolution/nemos
4.0K   ./local/share/evolution/tasks/system
4.0K   ./local/share/evolution/tasks/trash
16K    ./local/share/evolution/tasks
```

17. head

The head command is used to view the first lines of any text file. By default, it will show the first ten lines, but you can change this number to your liking. For example, if you only want to show the first five lines, type `head -n 5 filename.ext`.


```
user@user-HP-Laptop-15-da0xxx: ~  
user@user-HP-Laptop-15-da0xxx:~$ cat file1.txt  
computer networks  
user@user-HP-Laptop-15-da0xxx:~$ head -n 1 file1.txt  
computer networks  
user@user-HP-Laptop-15-da0xxx:~$
```

18. tail

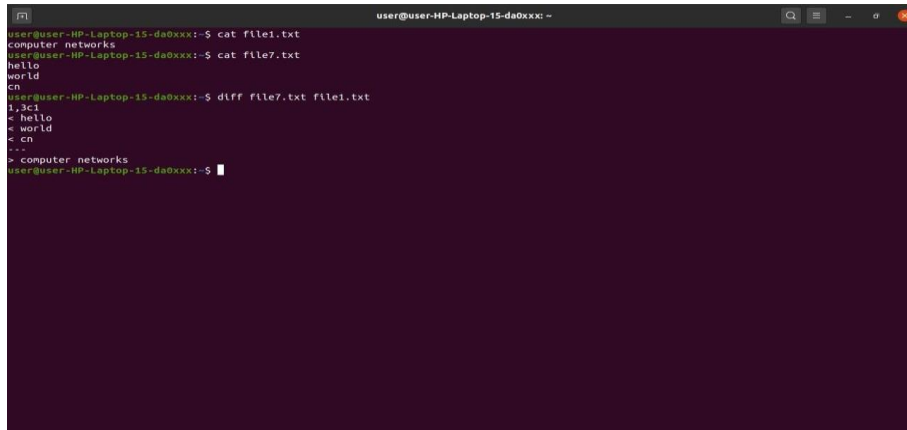
This one has a similar function to the head command, but instead of showing the first lines, the tail command will display the last ten lines of a text file. For example, tail -n filename.ext.

```
user@user-HP-Laptop-15-da0xxx:~$ cat file7.txt  
hello  
world  
cn  
user@user-HP-Laptop-15-da0xxx:~$ tail -n 2 file7.txt  
world  
cn  
user@user-HP-Laptop-15-da0xxx:~$
```

19. diff

Short for difference, the diff command compares the contents of two files line by line. After analyzing the files, it will output the lines that do not match. Programmers often use this command when they need to make program

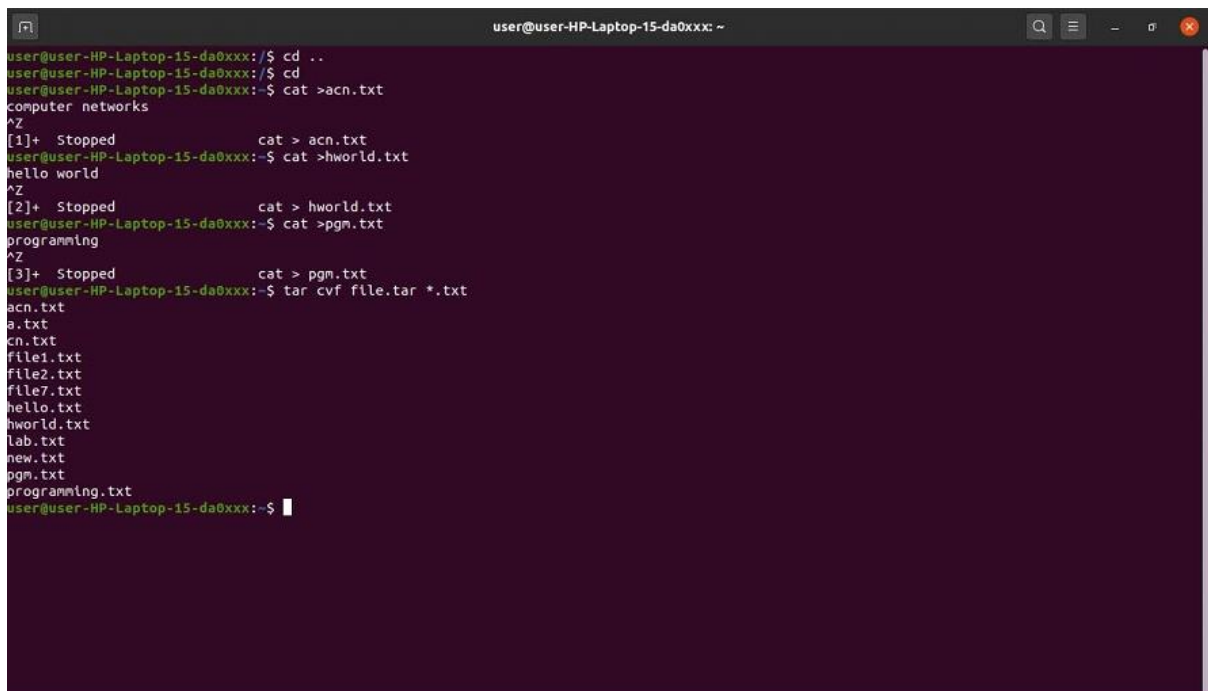
alterations instead of rewriting the entire source code. The simplest form of this command is `diff file1.txt file2.txt`

A terminal window titled 'user@user-HP-Laptop-15-da0xxx: ~' showing the following commands and output:

```
user@user-HP-Laptop-15-da0xxx:~$ cat file1.txt
computer networks
user@user-HP-Laptop-15-da0xxx:~$ cat file7.txt
hello
world
cn
user@user-HP-Laptop-15-da0xxx:~$ diff file7.txt file1.txt
1,3c1
< hello
< world
< cn
---
> computer networks
user@user-HP-Laptop-15-da0xxx:~$
```

20. tar

The tar command is the most used command to archive multiple files into a tarball — a common Linux file format that is similar to zip format, with compression being optional. This command is quite complex with a long list of functions such as adding new files into an existing archive, listing the content of an archive, extracting the content from an archive, and many more. Read some tutorial on net.

A terminal window titled 'user@user-HP-Laptop-15-da0xxx: ~' showing the following commands and output:

```
user@user-HP-Laptop-15-da0xxx:/$ cd ..
user@user-HP-Laptop-15-da0xxx:/$ cd
user@user-HP-Laptop-15-da0xxx:~$ cat >acn.txt
computer networks
^Z
[1]+  Stopped                  cat > acn.txt
user@user-HP-Laptop-15-da0xxx:~$ cat >hworld.txt
hello world
^Z
[2]+  Stopped                  cat > hworld.txt
user@user-HP-Laptop-15-da0xxx:~$ cat >pgn.txt
programming
^Z
[3]+  Stopped                  cat > pgn.txt
user@user-HP-Laptop-15-da0xxx:~$ tar cvf file.tar *.txt
acn.txt
a.txt
cn.txt
file1.txt
file2.txt
file7.txt
hello.txt
hworld.txt
lab.txt
new.txt
pgn.txt
programming.txt
user@user-HP-Laptop-15-da0xxx:~$
```

```
user@user-HP-Laptop-15-da0xxx: ~  
computer networks  
^Z  
[1]+  Stopped                  cat > acn.txt  
user@user-HP-Laptop-15-da0xxx:~$ cat >hworld.txt  
hello world  
^Z  
[2]+  Stopped                  cat > hworld.txt  
user@user-HP-Laptop-15-da0xxx:~$ cat >pgm.txt  
programming  
^Z  
[3]+  Stopped                  cat > pgm.txt  
user@user-HP-Laptop-15-da0xxx:~$ tar cvf file.tar *.txt  
acn.txt  
a.txt  
cn.txt  
file1.txt  
file2.txt  
file7.txt  
hello.txt  
hworld.txt  
lab.txt  
new.txt  
pgm.txt  
programming.txt  
user@user-HP-Laptop-15-da0xxx:~$ tar xvf file.tar  
acn.txt  
a.txt  
cn.txt  
file1.txt  
file2.txt  
file7.txt  
hello.txt  
hworld.txt  
lab.txt  
new.txt  
pgm.txt  
programming.txt  
user@user-HP-Laptop-15-da0xxx:~$
```

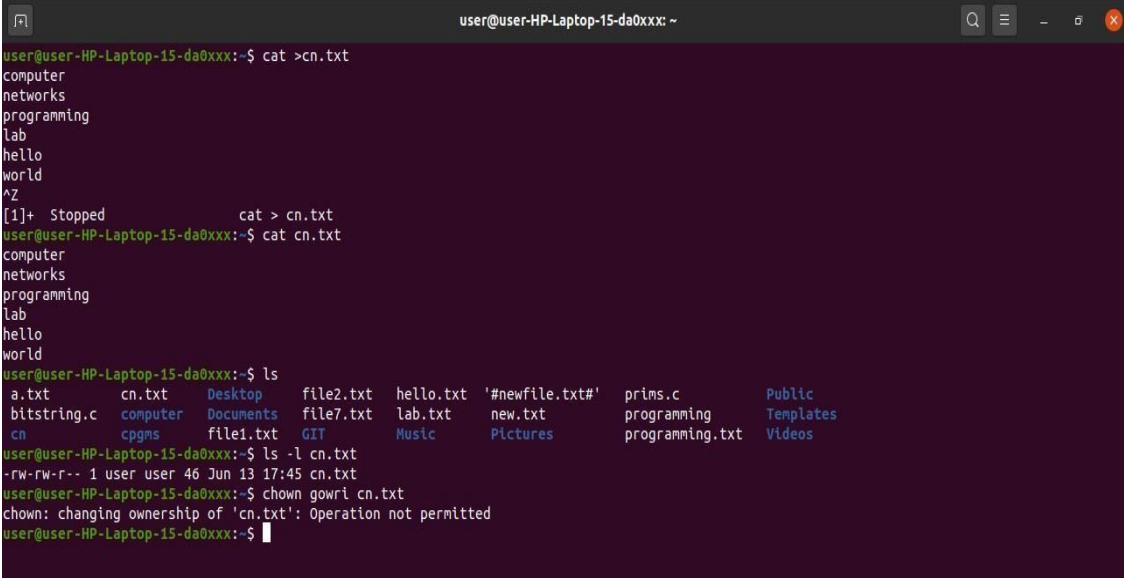
21. chmod :

chmod is another Linux command, used to change the read, write, and execute permissions of files and directories. Read about permissions and how to manipulate them .

```
user@user-HP-Laptop-15-da0xxx: ~  
user@user-HP-Laptop-15-da0xxx:~$ ls -l  
total 80  
-rw-rw-r-- 1 user user    0 Jun 12 01:13  a.txt  
-rw-rw-r-- 1 user user  458 Jan  3 00:59  bitstring.c  
drwxrwxr-x 2 user user 4096 Jun 11 02:18   cn  
drwxrwxr-x 2 user user 4096 Jun 11 18:19  computer  
drwxrwxr-x 5 user user 4096 Mar 21 14:32  cpqms  
drwxr-xr-x 2 user user 4096 Dec  2 2020  Desktop  
drwxr-xr-x 2 user user 4096 Dec  2 2020  Documents  
-rw-rw-r-- 1 user user   18 Jun 11 17:59  file1.txt  
-rw-rw-r-- 1 user user   18 Jun 11 18:04  file2.txt  
-rw-rw-r-- 1 user user   15 Jun 12 01:50  file7.txt  
drwxrwxr-x 3 user user 4096 Dec 28 01:32  GIT  
-rw-rw-r-- 1 user user    0 Jun 11 17:54  hello.txt  
-rw-rw-r-- 1 user user   15 Jun 11 17:54  lab.txt  
drwxr-xr-x 2 user user 4096 Dec  2 2020  Music  
-rw-rw-r-- 1 user user   29 Jun 10 03:58  '#newfile.txt#'  
-rw-rw-r-- 1 user user   12 Jun 11 18:09  new.txt  
drwxr-xr-x 5 user user 4096 Jun 12 01:35  Pictures  
-rw-rw-r-- 1 user user 1625 Mar 21 14:18  prims.c  
-rw-rw-r-- 1 user user    0 Jun 11 17:49  programming  
-rw-rw-r-- 1 user user   15 Jun 11 17:54  programming.txt  
drwxr-xr-x 2 user user 4096 Dec  2 2020  Public  
drwxr-xr-x 2 user user 4096 Dec  2 2020  Templates  
drwxr-xr-x 2 user user 4096 Dec  2 2020  Videos  
user@user-HP-Laptop-15-da0xxx:~$ ls -l notes.txt  
ls: cannot access 'notes.txt': No such file or directory  
user@user-HP-Laptop-15-da0xxx:~$ ls -l file1.txt  
-rw-rw-r-- 1 user user 18 Jun 11 17:59 file1.txt  
user@user-HP-Laptop-15-da0xxx:~$ chmod u=rw,og=r file1.txt  
user@user-HP-Laptop-15-da0xxx:~$ ls -l file1.txt  
-rw-r--r-- 1 user user 18 Jun 11 17:59 file1.txt  
user@user-HP-Laptop-15-da0xxx:~$
```

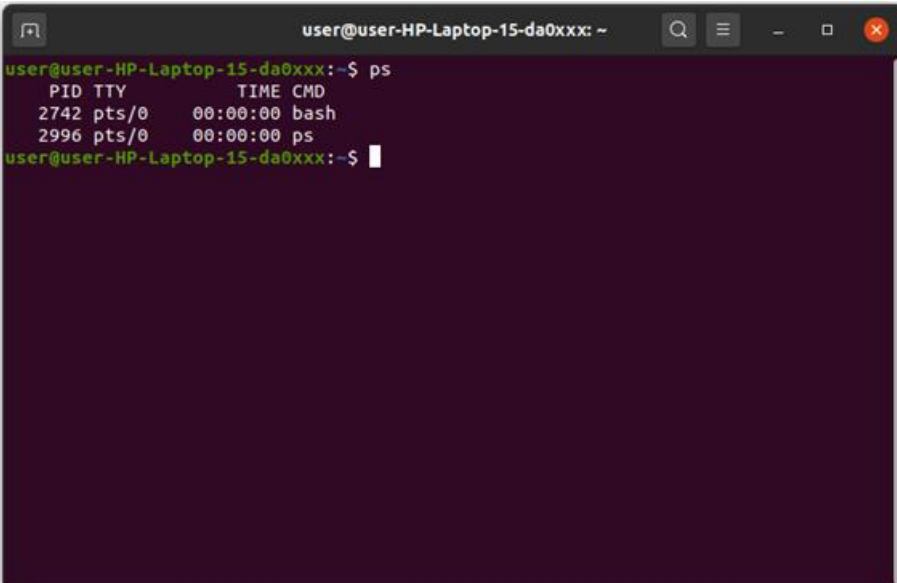
22. chown

In Linux, all files are owned by a specific user. The chown command enables you to change or transfer the ownership of a file to the specified username. For instance, chown linuxuser2 file.ext will make linuxuser2 as the owner of the file.ext.

A terminal window titled 'user@user-HP-Laptop-15-da0xxx: ~' showing a series of commands and their outputs. The user creates a file 'cn.txt' using 'cat > cn.txt' and then displays its contents with 'cat cn.txt'. The file contains the words 'computer', 'networks', 'programming', 'lab', 'hello', and 'world' on separate lines. The user then lists files with 'ls', showing a directory listing with various files and folders. Finally, the user attempts to change the ownership of 'cn.txt' to 'gowri' using 'chown gowri cn.txt', which results in an error message: 'chown: changing ownership of 'cn.txt': Operation not permitted'.

23. ps

Ps command will display all current processes along with their process ids (PID) . Read manuals for various options

A terminal window titled 'user@user-HP-Laptop-15-da0xxx: ~' showing the output of the 'ps' command. The output is a table with four columns: PID, TTY, TIME, and CMD. It lists two processes: 'bash' with PID 2742 and 'ps' with PID 2996. The user's prompt is followed by a cursor.

```
user@user-HP-Laptop-15-da0xxx:~$ ps -ux
user      PID  %CPU  %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
user      1449  0.1   0.2   19272  10628 ?        Ss   17:27   0:00 /lib/systemd/
user      1451  0.0   0.0   108904  3556 ?        S    17:27   0:00 (sd-pam)
user      1464  0.0   0.4   227604  19652 ?        S-sl 17:27   0:00 /usr/bin/puls
user      1466  0.0   0.0   520892  24400 ?        Ssl  17:27   0:00 /usr/libexec/
user      1468  0.1   0.1   8740   5816 ?        Ss   17:27   0:00 /usr/bin/dbus
user      1472  0.0   0.2   248804  7968 ?        Sl   17:27   0:00 /usr/bin/gnom
user      1477  0.0   0.1   240316  7796 ?        Ssl  17:27   0:00 /usr/libexec/
user      1482  0.0   0.2   302056  8552 ?        Sl   17:27   0:00 /usr/libexec/
user      1502  0.0   0.2   320640  11804 ?        Ssl  17:27   0:00 /usr/libexec/
user      1507  0.0   0.1   244100  6192 ?        Ssl  17:27   0:00 /usr/libexec/
user      1511  0.0   0.9   550416  36356 ?        Sl   17:27   0:00 /usr/libexec/
user      1518  0.0   0.2   327272  11556 ?        Sl   17:27   0:00 /usr/libexec/
user      1525  0.0   0.1   325352  7572 ?        Ssl  17:27   0:00 /usr/libexec/
user      1531  0.0   0.1   246596  6684 ?        Ssl  17:27   0:00 /usr/libexec/
user      1532  0.0   0.1   172628  6596 tty2    Ssl+ 17:27   0:00 /usr/lib/brda
user      1536  0.0   0.1   244320  6180 ?        Ssl  17:27   0:00 /usr/libexec/
user      1543  4.4   2.2   865900  88972 tty2    Sl+   17:27   0:18 /usr/lib/naoq
user      1570  0.0   0.3   199404  15604 tty2    Sl+   17:27   0:00 /usr/libexec/
user      1644  0.0   0.0   6032   456 ?        Ss   17:27   0:00 /usr/bin/ssh-
user      1661  0.2   0.2   323520  9596 ?        Ssl  17:27   0:00 /usr/bin/ibus
user      1668  0.0   0.1   248808  9060 ?        Sl   17:27   0:00 /usr/libexec/
user      1669  0.1   1.8   728184  73516 ?        Sl   17:27   0:00 /usr/libexec/
user      1672  0.4   1.5   686308  61652 ?        Sl   17:27   0:01 /usr/libexec/
user      1678  0.0   0.7   208996  29996 ?        Sl   17:27   0:00 /usr/libexec/
user      1681  0.0   0.2   248648  9016 ?        Sl   17:27   0:00 /usr/libexec/
user      1686  0.0   0.2   309820  9488 ?        Ssl  17:27   0:00 /usr/libexec/
user      1695  0.0   0.1   7216   4288 ?        S    17:27   0:00 /usr/bin/dbus
user      1707  0.0   0.1   162812  7640 ?        Sl   17:27   0:00 /usr/libexec/
user      1716  0.0   0.2   1012088  10572 ?        Ssl  17:27   0:00 /usr/libexec/
user      1725  0.0   0.1   98672   4328 ?        Ssl  17:27   0:00 /usr/libexec/
user      1727  0.0   0.1   466048  6480 ?        Ssl  17:27   0:00 /usr/libexec/
user      1733  0.0   0.4   508944  17520 ?        Ssl  17:27   0:00 /usr/libexec/
user      1738  0.0   0.1   244220  6932 ?        Ssl  17:27   0:00 /usr/libexec/
user      1748  0.0   0.2   175172  9124 ?        Sl   17:27   0:00 /usr/libexec/
user      1752  0.1   0.8   506680  32204 ?        Ssl  17:27   0:00 /usr/libexec/
user      1762  0.0   0.1   156208  5732 ?        Sl   17:27   0:00 /usr/libexec/
```

24. Kill

If you have an unresponsive program, you can terminate it manually by using the kill command. It will send a certain signal to the misbehaving app and instructs the app to terminate itself.

There is a total of sixty-four signals that you can use, but people usually only use two signals:

- *SIGTERM (15) — requests a program to stop running and gives it some time to save all of its progress. If you don't specify the signal when entering the kill command, this signal will be used.

- *SIGKILL (9) — forces programs to stop immediately. Unsaved progress will be lost. Besides knowing the signals, you also need to know the process identification number (PID) of the program you want to kill. If you don't know the PID, simply run the command ps ux.

After knowing what signal you want to use and the PID of the program, enter the following syntax: kill [signal option] PID. You can issue kill -9 PID

25. ping

Use the ping command to check your connectivity status to a server. For example, by simply entering ping google.com, the command will check whether you're able to connect to Google and also measure the response time.


```
user@user-HP-Laptop-15-da0xxx: ~  
user@user-HP-Laptop-15-da0xxx:~$ ping google.com  
PING google.com (64.6800:4009:80f::200e): 56 data bytes  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=1 ttl=117 time=305 ms  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=2 ttl=117 time=122 ms  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=3 ttl=117 time=145 ms  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=4 ttl=117 time=167 ms  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=5 ttl=117 time=194 ms  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=6 ttl=117 time=167 ms  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=7 ttl=117 time=233 ms  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=8 ttl=117 time=256 ms  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=9 ttl=117 time=278 ms  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=10 ttl=117 time=301 ms  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=11 ttl=117 time=324 ms  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=12 ttl=117 time=143 ms  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=13 ttl=117 time=163 ms  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=14 ttl=117 time=185 ms  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=15 ttl=117 time=207 ms  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=16 ttl=117 time=231 ms  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=17 ttl=117 time=251 ms  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=18 ttl=117 time=274 ms  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=19 ttl=117 time=296 ms  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=20 ttl=117 time=114 ms  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=21 ttl=117 time=137 ms  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=22 ttl=117 time=159 ms  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=23 ttl=117 time=182 ms  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=24 ttl=117 time=204 ms  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=25 ttl=117 time=226 ms  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=26 ttl=117 time=248 ms  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=27 ttl=117 time=270 ms  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=28 ttl=117 time=294 ms  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=29 ttl=117 time=110 ms  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=30 ttl=117 time=133 ms  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=31 ttl=117 time=156 ms  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=32 ttl=117 time=178 ms  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=33 ttl=117 time=200 ms  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=34 ttl=117 time=223 ms  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=35 ttl=117 time=246 ms  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=36 ttl=117 time=269 ms  
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=37 ttl=117 time=291 ms
```

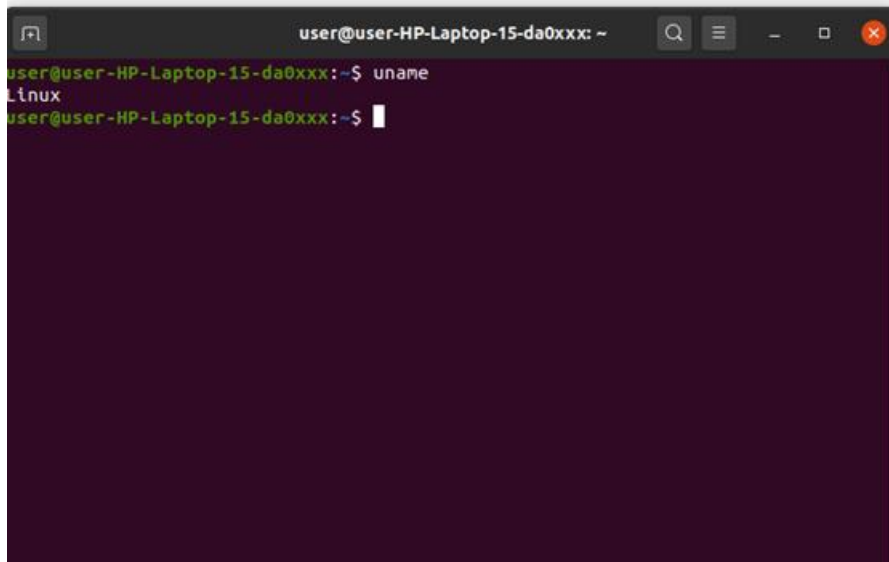
26. wget

The Linux command line is super useful — you can even download files from the internet with the help of the `wget` command. To do so, simply type `wget` followed by the download link.

```
user@user-HP-Laptop-15-da0xxx: ~  
user@user-HP-Laptop-15-da0xxx:~$ wget https://www.oracle.com/in/index.html  
--2021-06-13 17:53:10-- https://www.oracle.com/in/index.html  
Resolving www.oracle.com (www.oracle.com)... 2405:200:1630:4b8::a15, 2405:200:1630:482::a15, 23.9.76.25  
Connecting to www.oracle.com (www.oracle.com)|2405:200:1630:4b8::a15|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: unspecified [text/html]  
Saving to: 'index.html'  
  
index.html      [ <=> ] 38.10K  186KB/s  in 0.2s  
  
2021-06-13 17:53:12 (186 KB/s) - 'index.html' saved [39011]  
  
user@user-HP-Laptop-15-da0xxx:~$
```

27. uname

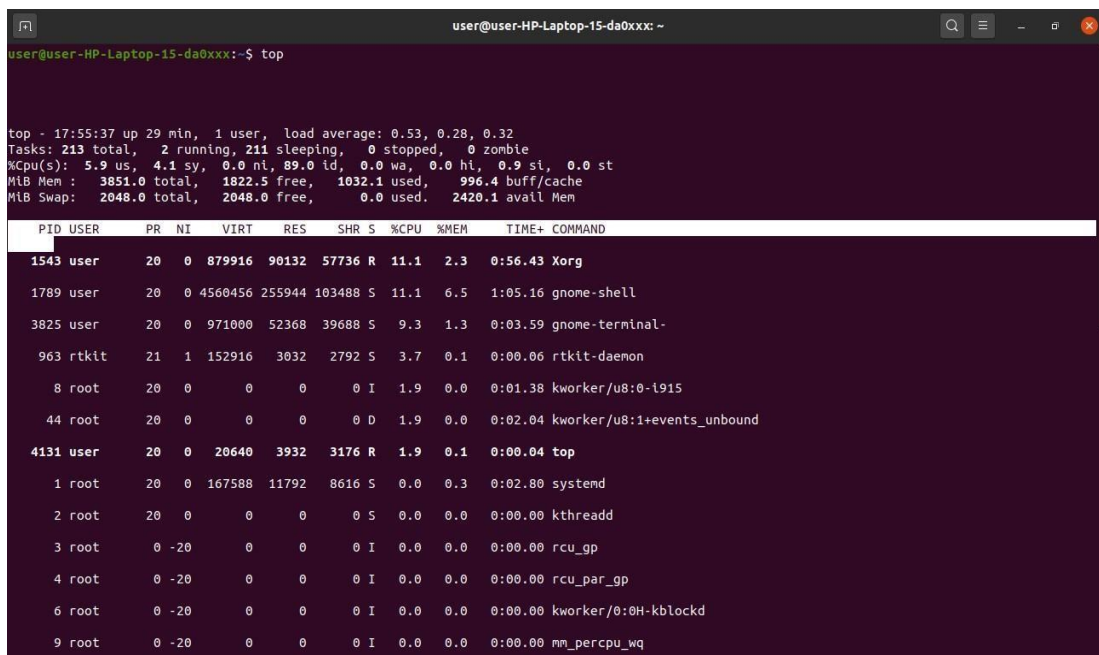
The `uname` command, short for Unix Name, will print detailed information about your Linux system like the machine name, operating system, kernel, and so on.



```
user@user-HP-Laptop-15-da0xxx: ~  
user@user-HP-Laptop-15-da0xxx:~$ uname  
Linux  
user@user-HP-Laptop-15-da0xxx:~$
```

28. top

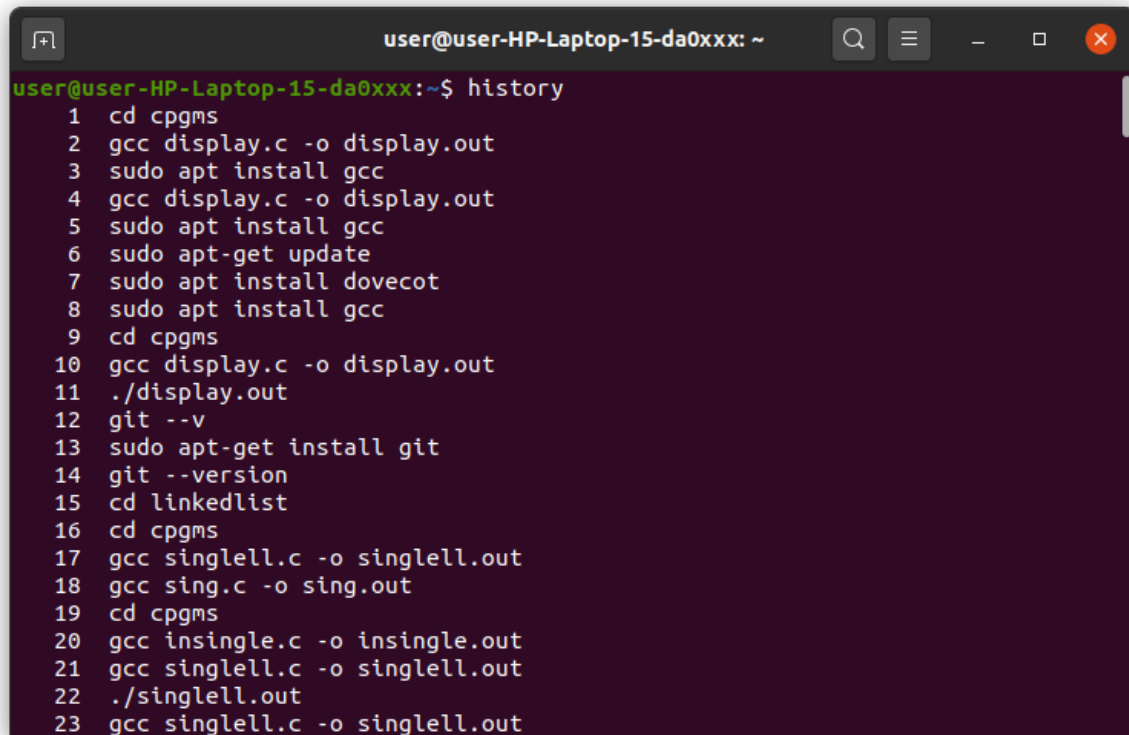
As a terminal equivalent to Task Manager in Windows, the `top` command will display a list of running processes and how much CPU each process uses. It's very useful to monitor system resource usage, especially knowing which process needs to be terminated because it consumes too many resources.



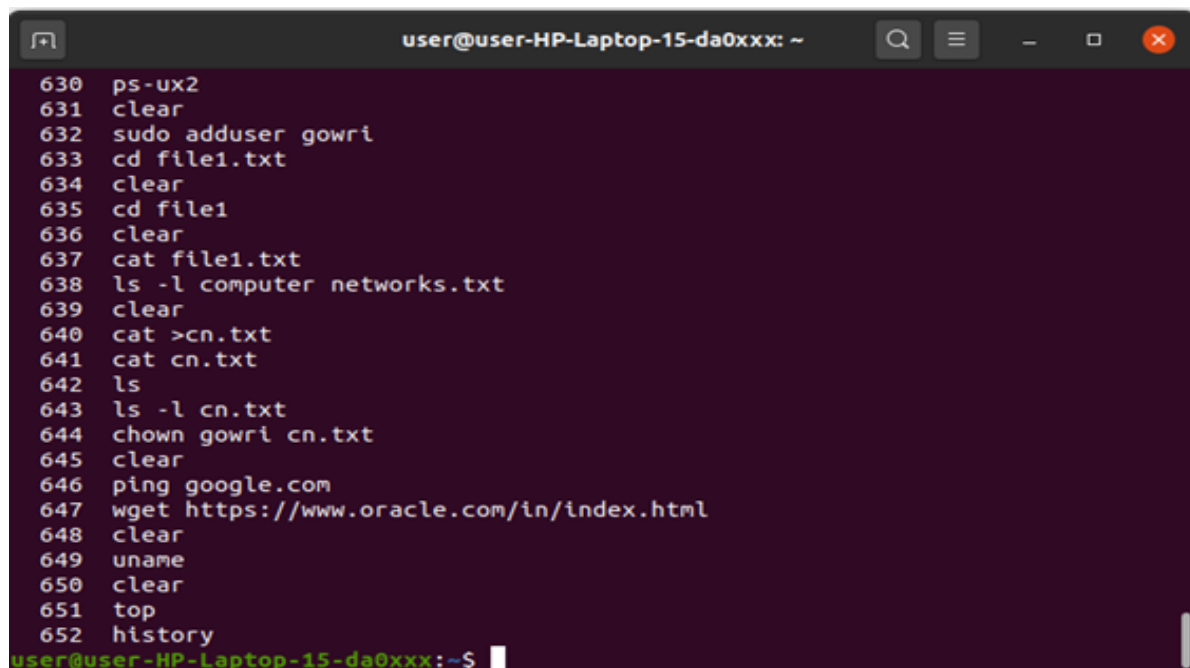
```
user@user-HP-Laptop-15-da0xxx:~$ top  
  
top - 17:55:37 up 29 min, 1 user, load average: 0.53, 0.28, 0.32  
Tasks: 213 total, 2 running, 211 sleeping, 0 stopped, 0 zombie  
%Cpu(s): 5.9 us, 4.1 sy, 0.0 ni, 89.0 id, 0.0 wa, 0.0 hi, 0.9 si, 0.0 st  
MiB Mem : 3851.0 total, 1822.5 free, 1032.1 used, 996.4 buff/cache  
MiB Swap: 2048.0 total, 2048.0 free, 0.0 used, 2420.1 avail Mem  
  
  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND  
1543 user      20   0  879916  90132  57736 R   11.1   2.3   0:56.43 Xorg  
1789 user      20   0 4560456 255944 103488 S   11.1   6.5   1:05.16 gnome-shell  
3825 user      20   0  971000  52368  39688 S    9.3   1.3   0:03.59 gnome-terminal  
 963 rtkit     21   1 152916   3032   2792 S    3.7   0.1   0:00.06 rtkit-daemon  
    8 root       20   0     0     0     0  I    1.9   0.0   0:01.38 kworker/u8:0-l915  
   44 root       20   0     0     0     0  D    1.9   0.0   0:02.04 kworker/u8:1+events_unbound  
4131 user      20   0   20640   3932   3176 R    1.9   0.1   0:00.04 top  
    1 root       20   0 167588  11792   8616 S    0.0   0.3   0:02.80 systemd  
    2 root       20   0     0     0     0  S    0.0   0.0   0:00.00 kthreadd  
    3 root        0 -20     0     0     0  I    0.0   0.0   0:00.00 rcu_gp  
    4 root        0 -20     0     0     0  I    0.0   0.0   0:00.00 rcu_par_gp  
    6 root        0 -20     0     0     0  I    0.0   0.0   0:00.00 kworker/0:0H-kblockd  
    9 root        0 -20     0     0     0  I    0.0   0.0   0:00.00 mm_percpu_wq
```

29. history

When you've been using Linux for a certain period of time, you'll quickly notice that you can run hundreds of commands every day. As such, running history command is particularly useful if you want to review the s you've entered before.

A terminal window titled 'user@user-HP-Laptop-15-da0xxx: ~' with search, menu, and window control icons. The terminal shows the command 'history' being executed, which lists 23 previous commands with line numbers. The commands include directory changes, compiling C programs, installing software like gcc and git, and running system updates.

```
user@user-HP-Laptop-15-da0xxx:~$ history
 1 cd cpgms
 2 gcc display.c -o display.out
 3 sudo apt install gcc
 4 gcc display.c -o display.out
 5 sudo apt install gcc
 6 sudo apt-get update
 7 sudo apt install dovecot
 8 sudo apt install gcc
 9 cd cpgms
10 gcc display.c -o display.out
11 ./display.out
12 git --v
13 sudo apt-get install git
14 git --version
15 cd linkedlist
16 cd cpgms
17 gcc singlell.c -o singlell.out
18 gcc sing.c -o sing.out
19 cd cpgms
20 gcc insingle.c -o insingle.out
21 gcc singlell.c -o singlell.out
22 ./singlell.out
23 gcc singlell.c -o singlell.out
```

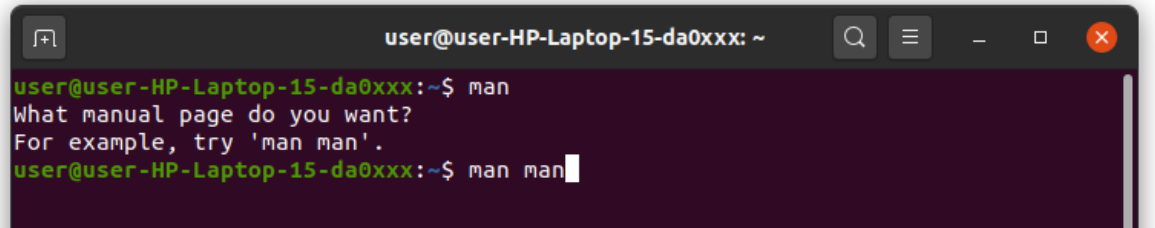
A terminal window titled 'user@user-HP-Laptop-15-da0xxx: ~' with search, menu, and window control icons. The terminal shows a continuation of the command history, starting from line 630. The commands include clearing the screen, adding a user, creating and editing files, running network tests, and finally running the 'history' command again.

```
user@user-HP-Laptop-15-da0xxx:~$
630 ps-ux2
631 clear
632 sudo adduser gowri
633 cd file1.txt
634 clear
635 cd file1
636 clear
637 cat file1.txt
638 ls -l computer networks.txt
639 clear
640 cat >cn.txt
641 cat cn.txt
642 ls
643 ls -l cn.txt
644 chown gowri cn.txt
645 clear
646 ping google.com
647 wget https://www.oracle.com/in/index.html
648 clear
649 uname
650 clear
651 top
652 history
user@user-HP-Laptop-15-da0xxx:~$
```

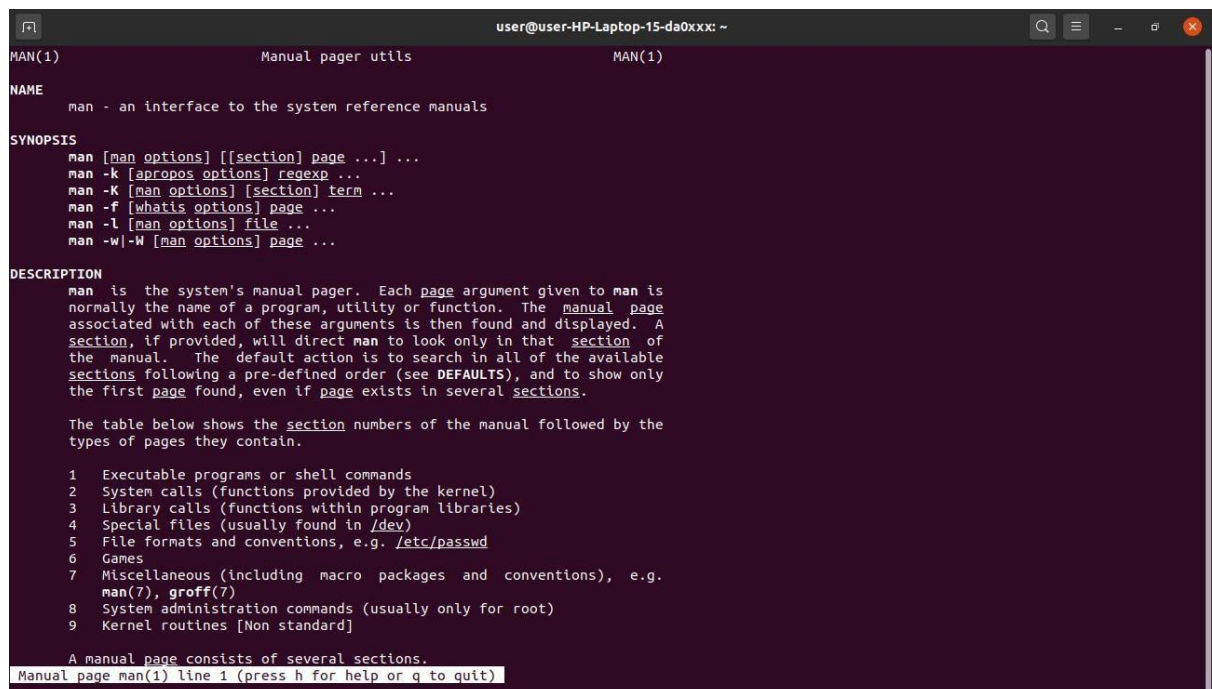

30. man

Confused about the function of certain Linux commands? Don't worry, you can easily learn how to use them right from Linux's shell by using the man command. For instance, entering man tail will show the manual instruction of the tail command.

Use the command: man man to start learning about man utility.



```
user@user-HP-Laptop-15-da0xxx: ~  
user@user-HP-Laptop-15-da0xxx:~$ man  
What manual page do you want?  
For example, try 'man man'.  
user@user-HP-Laptop-15-da0xxx:~$ man man
```



```
MAN(1) Manual pager utils MAN(1)  
NAME  
man - an interface to the system reference manuals  
SYNOPSIS  
man [man options] [[section] page ...] ...  
man -k [apropos options] regexp ...  
man -K [man options] [section] term ...  
man -f [whatIs options] page ...  
man -l [man options] file ...  
man -w|-W [man options] page ...  
DESCRIPTION  
man is the system's manual pager. Each page argument given to man is normally the name of a program, utility or function. The manual page associated with each of these arguments is then found and displayed. A section, if provided, will direct man to look only in that section of the manual. The default action is to search in all of the available sections following a pre-defined order (see DEFAULTS), and to show only the first page found, even if page exists in several sections.  
  
The table below shows the section numbers of the manual followed by the types of pages they contain.  
  
1 Executable programs or shell commands  
2 System calls (functions provided by the kernel)  
3 Library calls (functions within program libraries)  
4 Special files (usually found in /dev)  
5 File formats and conventions, e.g. /etc/passwd  
6 Games  
7 Miscellaneous (including macro packages and conventions), e.g. man(7), groff(7)  
8 System administration commands (usually only for root)  
9 Kernel routines [Non standard]  
  
A manual page consists of several sections.  
Manual page man(1) line 1 (press h for help or q to quit)
```

31. echo

This command is used to move some data into a file. For example, if you want to add the text, "Hello, my name is John" into a file called name.txt, you would type echo

Hello, my name is John >> name.txt

```
user@user-HP-Laptop-15-da0xxx: ~  
user@user-HP-Laptop-15-da0xxx:~$ echo My name is GangaKrishnanG  
My name is GangaKrishnanG  
user@user-HP-Laptop-15-da0xxx:~$
```

32. zip, unzip

Use the zip command to compress your files into a zip archive, and use the unzip command to extract the zipped files from a zip archive. (This program should be installed , some distributions may not have them. You can also look at gzip and bzip commands).

```
user@user-HP-Laptop-15-da0xxx: ~  
user@user-HP-Laptop-15-da0xxx:~$ zip  
Copyright (c) 1990-2008 Info-ZIP - Type 'zip -L' for software license.  
Zip 3.0 (July 5th 2008). Usage:  
zip [-options] [-b path] [-t mmdyyy] [-n suffixes] [zipfile list] [-xl list]  
The default action is to add or replace zipfile entries from list, which  
can include the special name - to compress standard input.  
If zipfile and list are omitted, zip compresses stdin to stdout.  
-f freshen: only changed files      -u update: only changed or new files  
-d delete entries in zipfile        -m move into zipfile (delete OS files)  
-r recurse into directories          -j junk (don't record) directory names  
-0 store only                        -l convert LF to CR LF (-ll CR LF to LF)  
-1 compress faster                  -9 compress better  
-q quiet operation                  -v verbose operation/print version info  
-c add one-line comments            -z add zipfile comment  
-@ read names from stdin             -o make zipfile as old as latest entry  
-x exclude the following names      -i include only the following names  
-F fix zipfile (-FF try harder)     -D do not add directory entries  
-A adjust self-extracting exe       -J junk zipfile prefix (unzipsfx)  
-T test zipfile integrity            -X eXclude eXtra file attributes  
-y store symbolic links as the link  instead of the referenced file  
-e encrypt                          -n don't compress these suffixes  
-h2 show more help  
user@user-HP-Laptop-15-da0xxx:~$
```

```
user@user-HP-Laptop-15-da0xxx: ~  
user@user-HP-Laptop-15-da0xxx:~$ unzip  
UnZip 6.00 of 20 April 2009, by Debian. Original by Info-ZIP.  
  
Usage: unzip [-Z] [-opts[modifiers]] file[.zip] [list] [-x xlist] [-d exdir]  
Default action is to extract files in list, except those in xlist, to exdir;  
file[.zip] may be a wildcard. -Z => ZipInfo mode ("unzip -Z" for usage).  
  
-p extract files to pipe, no messages      -l list files (short format)  
-f freshen existing files, create none     -t test compressed archive data  
-u update files, create if necessary        -z display archive comment only  
-v list verbosely/show version info        -T timestamp archive to latest  
-x exclude files that follow (in xlist)    -d extract files into exdir  
  
modifiers:  
-n never overwrite existing files          -q quiet mode (-qq => quieter)  
-o overwrite files WITHOUT prompting       -a auto-convert any text files  
-j junk paths (do not make directories)    -aa treat ALL files as text  
-U use escapes for all non-ASCII Unicode   -UU ignore any Unicode fields  
-C match filenames case-insensitively     -L make (some) names lowercase  
-X restore UID/GID info                   -V retain VMS version numbers  
-K keep setuid/setgid/tacky permissions   -M pipe through "more" pager  
-O CHARSET specify a character encoding for DOS, Windows and OS/2 archives  
-I CHARSET specify a character encoding for UNIX and other archives  
  
See "unzip -hh" or unzip.txt for more help. Examples:  
unzip data1 -x joe => extract all files except joe from zipfile data1.zip  
unzip -p foo | more => send contents of foo.zip via pipe into program more  
unzip -fo foo ReadMe => quietly replace existing ReadMe if archive file newer  
user@user-HP-Laptop-15-da0xxx:~$
```

33. hostname

If you want to know the name of your host/network simply type `hostname`. Adding a `-I` to the end will display the IP address of your network.

```
user@user-HP-Laptop-15-da0xxx: ~  
user@user-HP-Laptop-15-da0xxx:~$ hostname  
user-HP-Laptop-15-da0xxx  
user@user-HP-Laptop-15-da0xxx:~$ hostname -I  
192.168.43.12 2409:4073:4e09:e1fe:1975:e1c4:ef1b:9c7f 2409:4073:4e09:e1fe:62e7:  
9033:2cc3:514a  
user@user-HP-Laptop-15-da0xxx:~$
```

34. useradd, userdel

This is available only to system admins. Since Linux is a multi-user system, this means more than one person can interact with the same system at the same time. `useradd` is used to create a new user, while `passwd` is adding a password to that user's account. To add a new person named John type, `useradd John` and then to add his password type, `passwd 123456789`.

```
user@user-HP-Laptop-15-da0xxx:~$ useradd
Usage: useradd [options] LOGIN
       useradd -D
       useradd -D [options]

Options:
  --badnames           do not check for bad names
  -b, --base-dir BASE_DIR base directory for the home directory of the
                        new account
  --btrfs-subvolume-home use BTRFS subvolume for home directory
  -c, --comment COMMENT GECOS field of the new account
  -d, --home-dir HOME_DIR home directory of the new account
  -D, --defaults        print or change default useradd configuration
  -e, --expiredate EXPIRE_DATE expiration date of the new account
  -f, --inactive INACTIVE password inactivity period of the new account
  -g, --gid GROUP        name or ID of the primary group of the new
                        account
  -G, --groups GROUPS    list of supplementary groups of the new
                        account
  -h, --help            display this help message and exit
  -k, --skel SKEL_DIR   use this alternative skeleton directory
  -K, --key KEY-VALUE    override /etc/login.defs defaults
  -l, --no-log-init      do not add the user to the lastlog and
                        faillog databases
  -m, --create-home      create the user's home directory
  -M, --no-create-home   do not create the user's home directory
  -N, --no-user-group     do not create a group with the same name as
                        the user
  -o, --non-unique       allow to create users with duplicate
                        (non-unique) UID
  -p, --password PASSWORD encrypted password of the new account
  -r, --system          create a system account
  -R, --root CHROOT_DIR directory to chroot into
  -P, --prefix PREFIX_DIR prefix directory where are located the /etc/* files
  -s, --shell SHELL      login shell of the new account
  -u, --uid UID          user ID of the new account
  -U, --user-group       create a group with the same name as the user
  -Z, --selinux-user SEUSER use a specific SEUSER for the SELinux user mapping
                        like the extra users database
  --extrausers           use the extra users database
```

```
user@user-HP-Laptop-15-da0xxx:~$ userdel
Usage: userdel [options] LOGIN

Options:
  -f, --force           force removal of files,
                        even if not owned by user
  -h, --help            display this help message and exit
  -r, --remove          remove home directory and mail spool
  -R, --root CHROOT_DIR directory to chroot into
  -P, --prefix PREFIX_DIR prefix directory where are located the /etc/* files
  --extrausers          Use the extra users database
  -Z, --selinux-user    remove any SELinux user mapping for the user

user@user-HP-Laptop-15-da0xxx:~$
```

35. passwd :

passwd command in Linux is used to change the user account passwords. The root user reserves the privilege to change the password for any user on the system, while a normal user can only change the account password for his or her own account.

```
himanshu@ansh:~$ passwd
Changing password for himanshu.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
himanshu@ansh:~$
```

36. expr

The *expr* command in Unix evaluates a given expression and displays its corresponding output. It is used for:

- Basic operations like addition, subtraction, multiplication, division, and modulus on integers.
- Evaluating regular expressions, string operations like substring, length of strings etc.

```
anshul@anshul-VirtualBox:~/Desktop$ expr --version
expr (GNU coreutils) 8.28
Copyright (C) 2017 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Written by Mike Parker, James Youngman, and Paul Eggert.
anshul@anshul-VirtualBox:~/Desktop$
```

37.cut

The *cut* command in UNIX is a command for cutting out the sections from each line of files and writing the result to standard output. It can be used to cut parts of a line by byte position, character and field. Basically the *cut* command slices a line and extracts the text. It is necessary to specify option with command otherwise it gives error. If more than one file name is provided then data from each file is not preceded by its file name.


```
javatpoint@javatpoint-Inspiron-3542:~$ cat marks.txt
alex-50
alen-70
jon-75
carry-85
celena-90
justin-80
javatpoint@javatpoint-Inspiron-3542:~$ cut -d- -f2 marks.txt
50
70
75
85
90
80
javatpoint@javatpoint-Inspiron-3542:~$ cut -d- -f1 marks.txt
alex
alen
jon
carry
celena
justin
```

38.paste

Paste command is one of the useful commands in Unix or Linux operating system. It is used to join files horizontally (parallel merging) by outputting lines consisting of lines from each file specified, separated by tab as delimiter, to the standard output. When no file is specified, or put dash (“-“) instead of file name, paste reads from standard input and gives output as it is until a interrupt command

```
exam23@cec-Veriton-M200-H81:~$ touch states
exam23@cec-Veriton-M200-H81:~$ cat states
exam23@cec-Veriton-M200-H81:~$ touch capital
exam23@cec-Veriton-M200-H81:~$ cat capital
exam23@cec-Veriton-M200-H81:~$ paste states capital
Assam      Dispur
Bihar      Patna
exam23@cec-Veriton-M200-H81:~$
```

39.ssh,scp :

`ssh` stands for “**Secure Shell**”. It is a protocol used to securely connect to a remote server/system. `ssh` is secure in the sense that it transfers the data in encrypted form between the host and the client. It transfers inputs from the client to the host and relays back the output. `ssh` runs at TCP/IP port 22.

`scp` (secure copy) command in Linux system is used to copy file(s) between servers in a secure way. The SCP command or secure copy allows secure transferring of files in between the local host and the remote host or between two remote hosts. It uses the same authentication and security as it is used in the Secure Shell (SSH) protocol. SCP is known for its simplicity, security and pre-installed availability.

40.ssh-keygen, ssh-copy-id

`ssh-keygen` is the utility used to generate, manage, and convert authentication keys for SSH. `sshkeygen` comes installed with SSH in most of the operating systems. `ssh-keygen` is able to generate a key using one of three different digital signature algorithms.

- RSA
- DSA
- ECDSA

The `ssh-copy-id` command is a simple tool that allows you to install an SSH key on a remote server’s authorized keys. This command facilitates SSH key login, which removes the need for a password for each login, thus ensuring a password-less, automatic login process. The `ssh-copy-id` command is part of OpenSSH, a tool for performing remote system administrations using encrypted SSH connections.

```
javatpoint@javatpoint-Inspiron-3542:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/javatpoint/.ssh/id_rsa): key@ssh
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in key@ssh.
Your public key has been saved in key@ssh.pub.
The key fingerprint is:
SHA256:GJRjZ6aTbsE5ejZ5HRaHBKm7sIne6Yqc/aF7NBqBTVM javatpoint@javatpoint-Inspiron-3542
The key's randomart image is:
+---[RSA 2048]-----+
|  .E...+..          |
|  o .+ = o .        |
|  + .o.X   o        |
|  . o Xo  o         |
|  .o.*So .          |
|  .ooO . .          |
|  .+O.+             |
|  ..+o+oo           |
|  .+o*B.            |
+---[SHA256]-----+
javatpoint@javatpoint-Inspiron-3542:~$
```