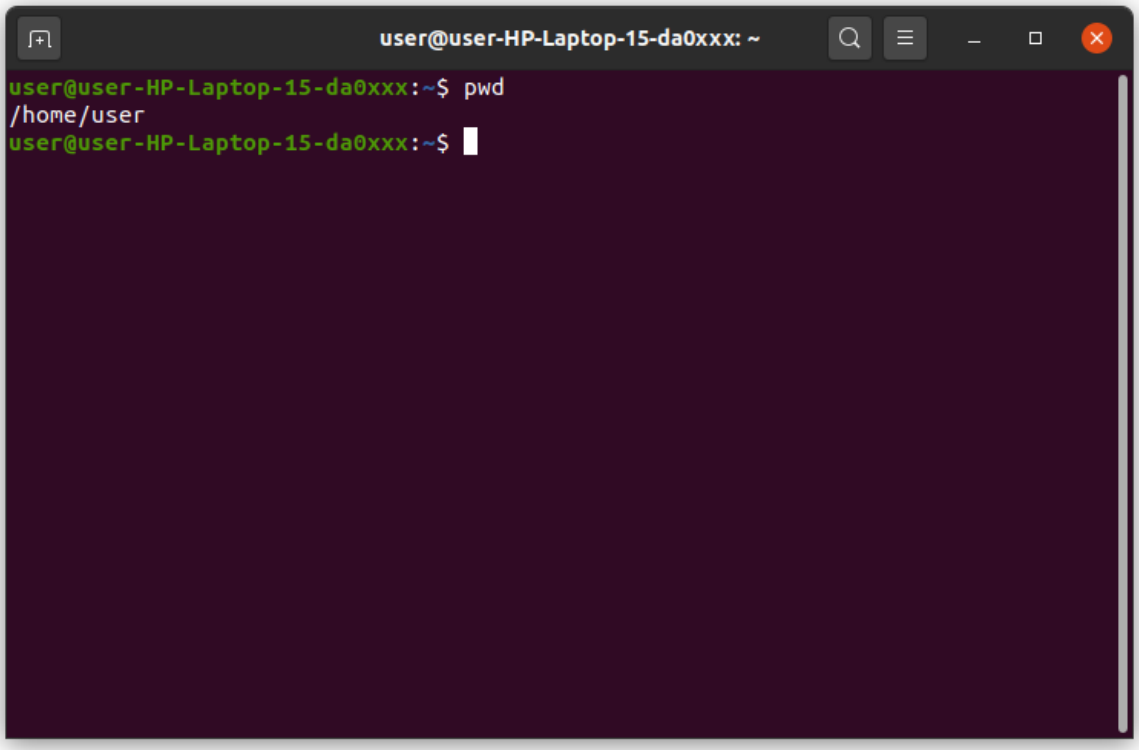
EXPERIMENT NO:4 **DATE:**

FAMILIARISATION WITH LINUX COMMANDS

AIM: 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.



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.

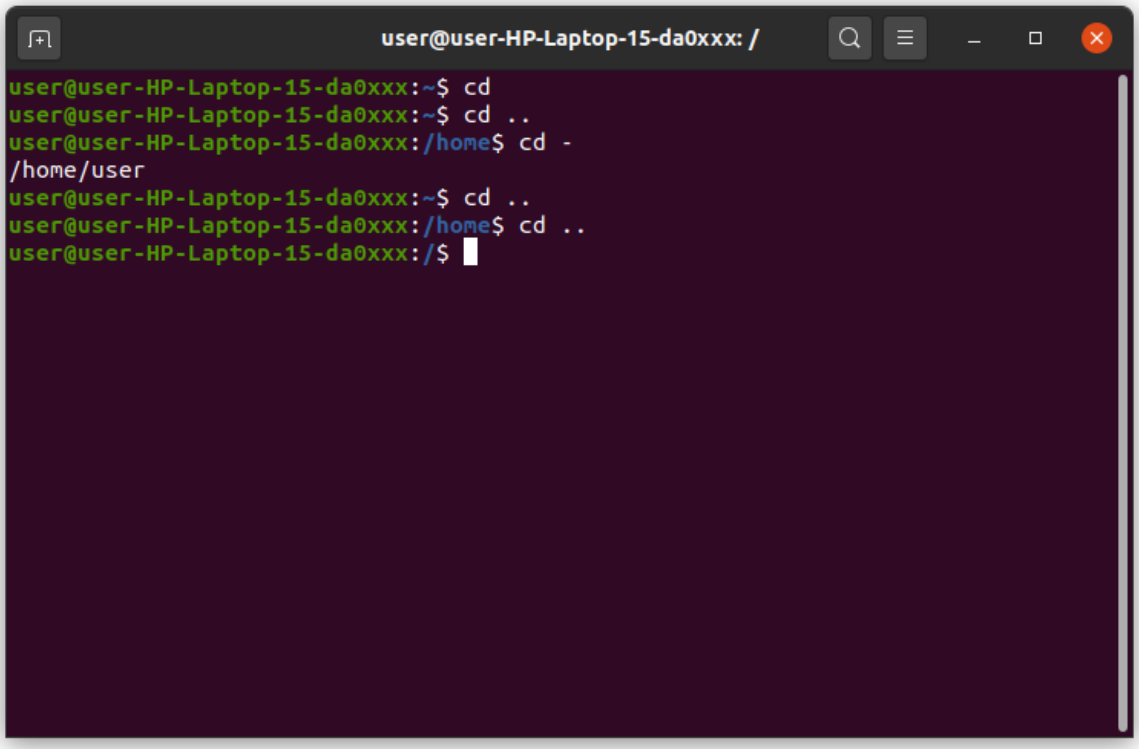
Let's say you're in /home/username/Documents and you want to go to Photos, a subdirectory of Documents. To do so, simply type the following command: cd Photos. Another scenario is if you want to switch to a completely new directory, for

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.



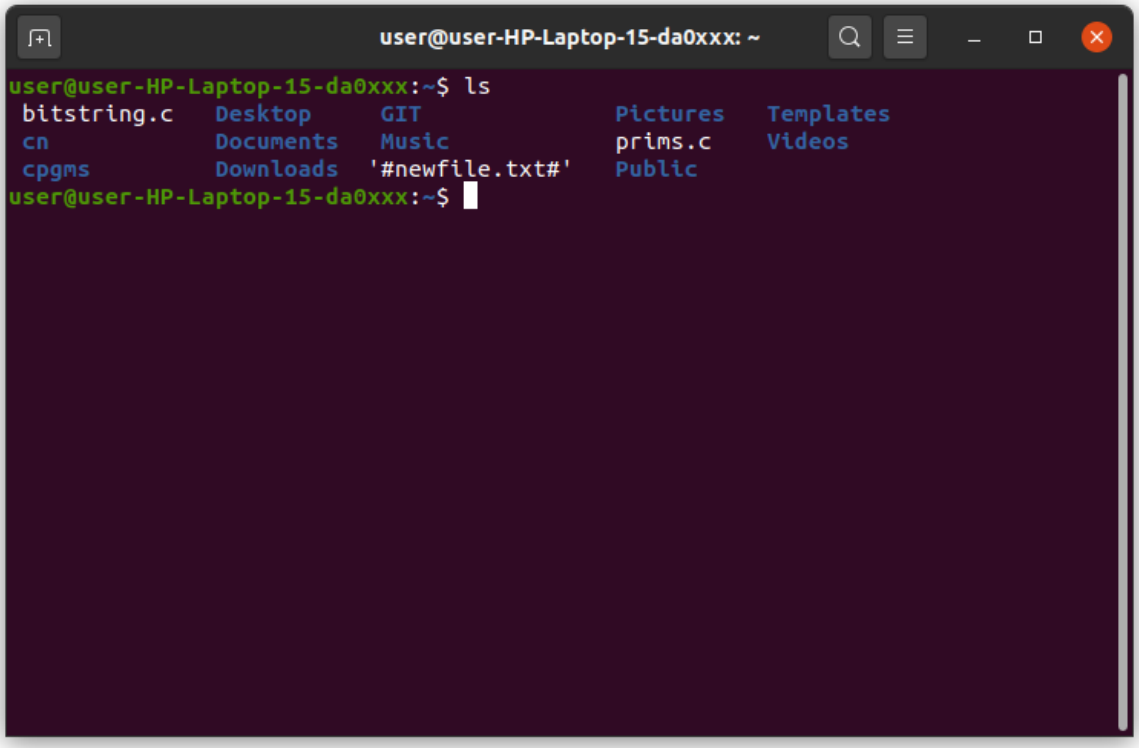
3. ls

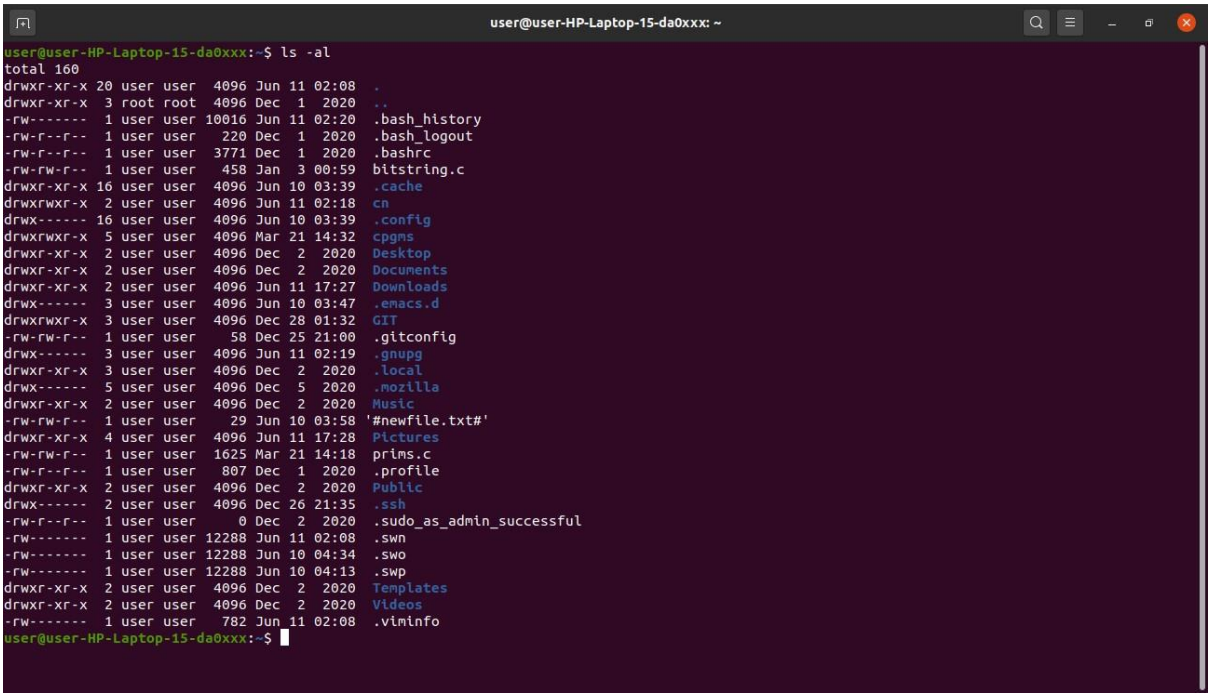
The ls command is used to view the contents of a directory. By default, this command will display the contents of your current working directory.

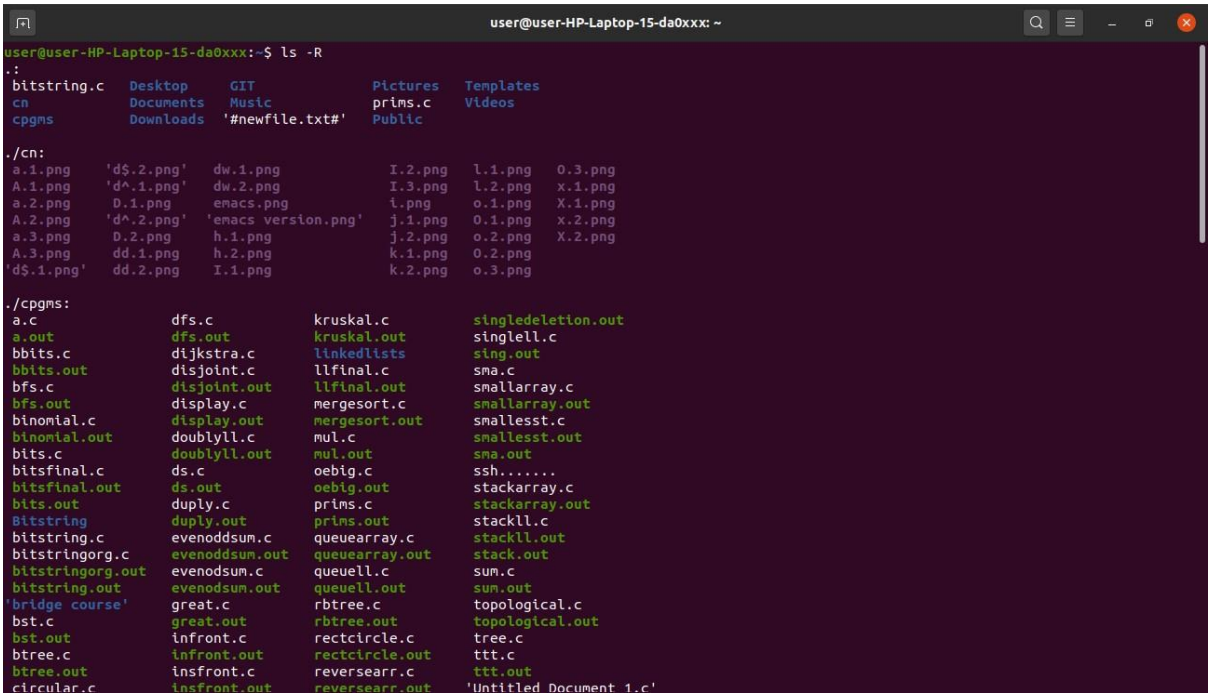
If you want to see the content of other directories, type ls and then the directory's path. For example, enter ls /home/username/Documents to view the content of Documents.

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:~$

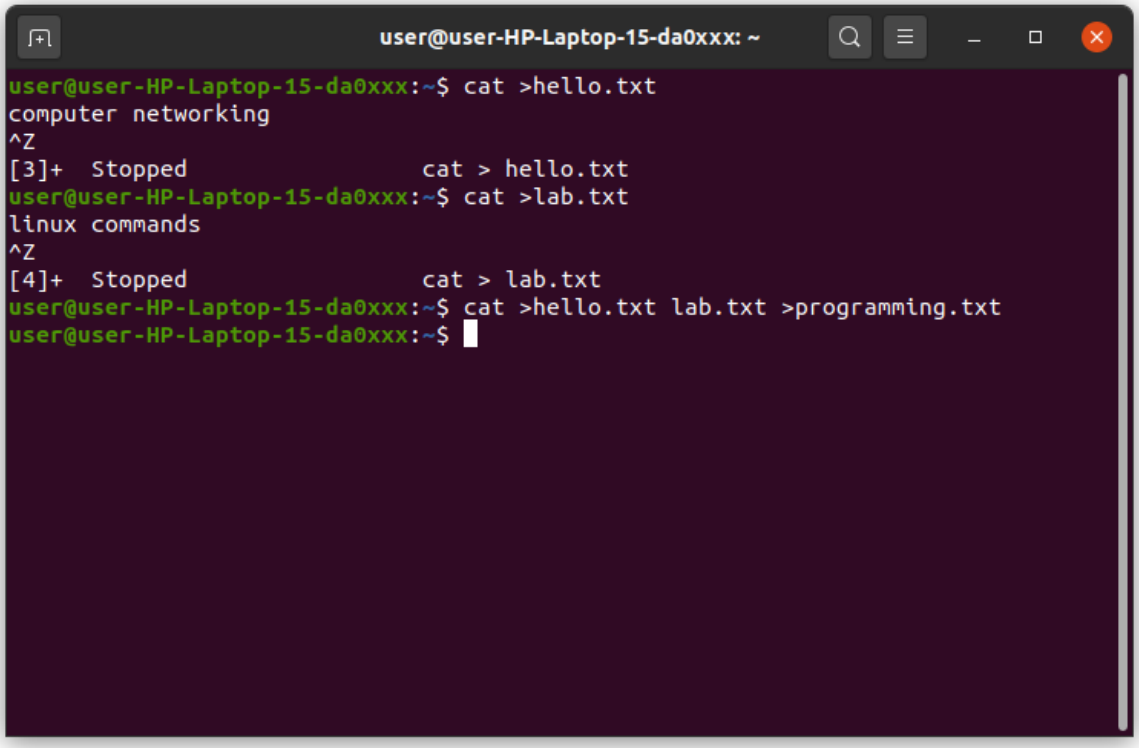

user@user-HP-Laptop-15-da0xxx: ~
user@user-HP-Laptop-15-da0xxx:~$ ls -al
total 160
drwxr-xr-x 20 user user 4096 Jun 11 02:08 .
drwxr-xr-x 3 root root 4096 Dec 1 2020 ..
-rw------- 1 user user 10016 Jun 11 02:20 .bash_history
-rw-r--r-- 1 user user 220 Dec 1 2020 .bash_logout
-rw-r--r-- 1 user user 3771 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 .swn
-rw------- 1 user user 12288 Jun 10 04:34 .swo
-rw------- 1 user user 12288 Jun 10 04:13 .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:~$

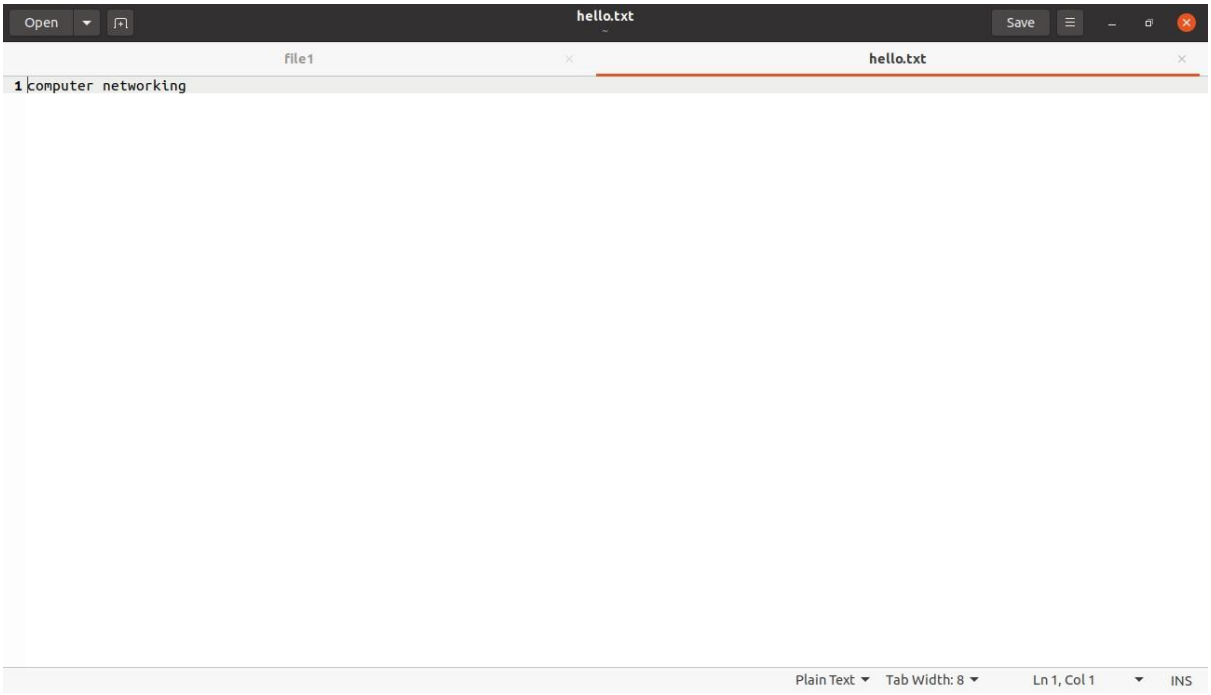


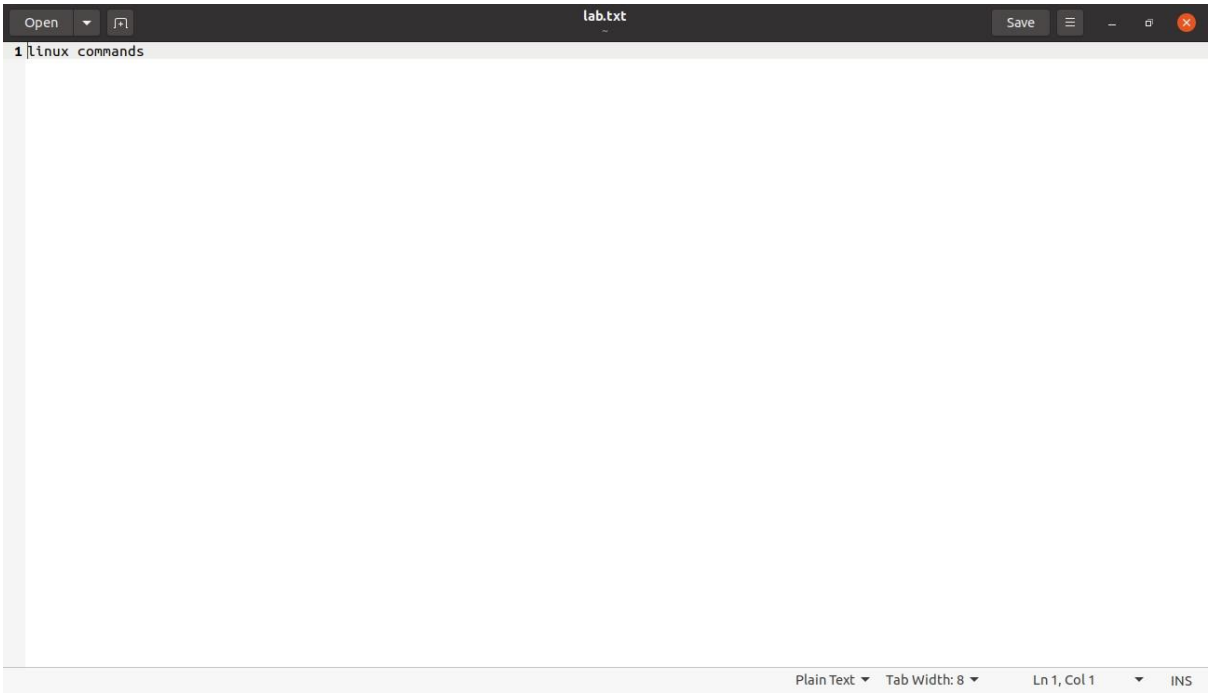
4. cat cat (short for concatenate) is one of the most frequently used commands in Linux. It is used to list the contents of a file on the standard output stdout . To run this command, type cat followed by the file's name and its extension. For instance: cat file.txt.

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


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
INS



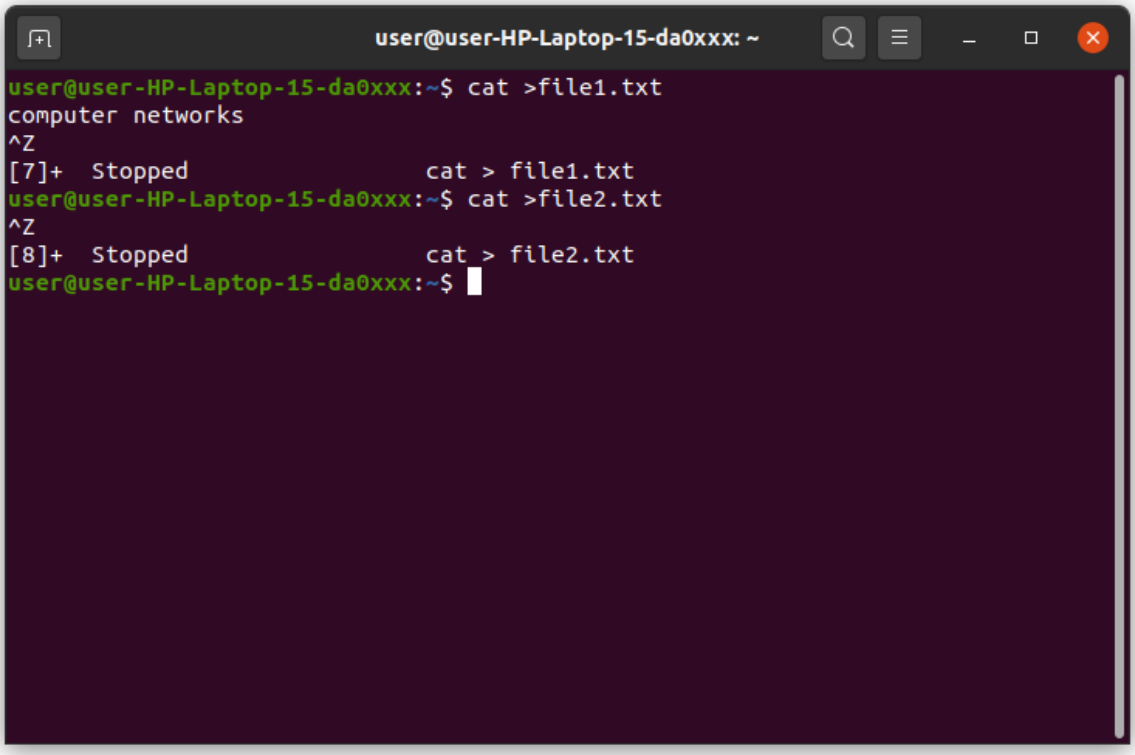
5. cp

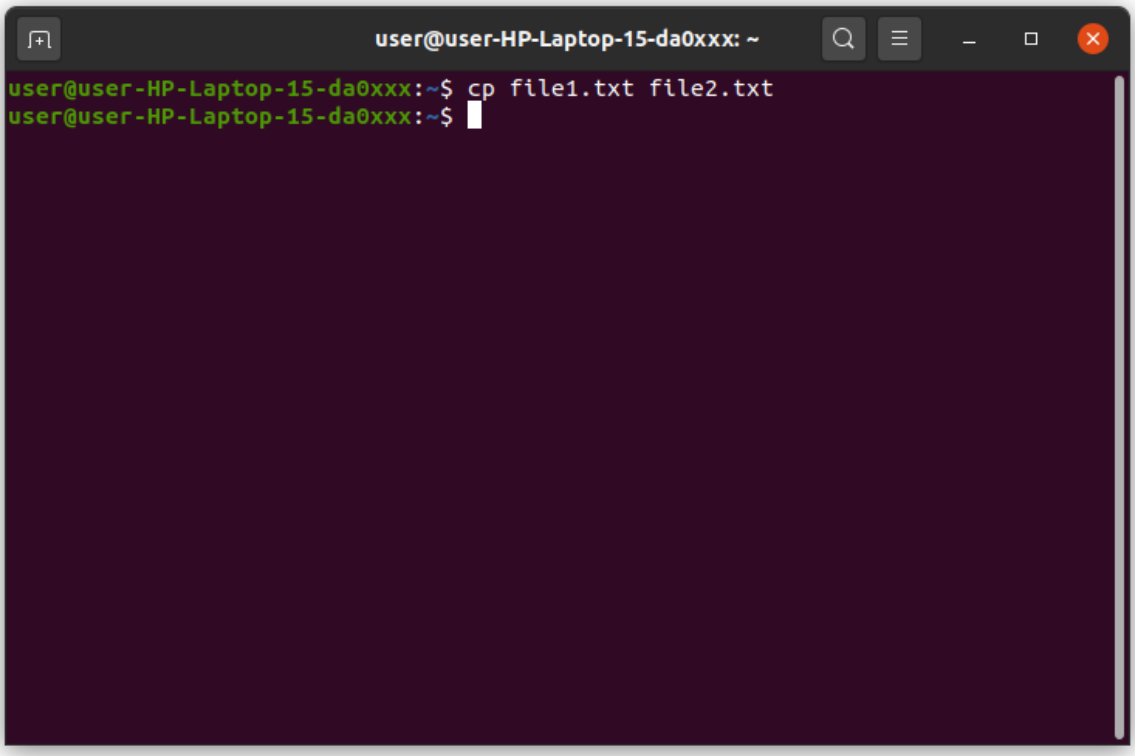
Use the cp command to copy files from the current directory to a different directory.

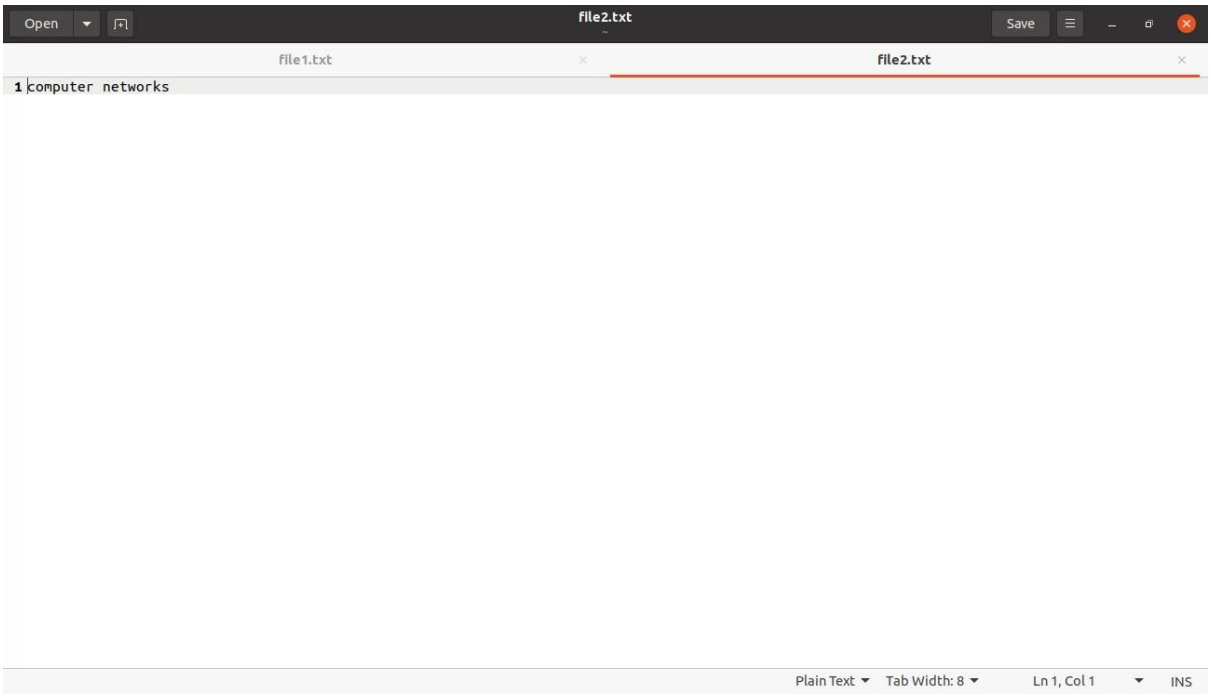
For instance, the command cp scenery.jpg /home/username/Pictures would create a copy of scenery.jpg (from your current directory) into the Pictures directory.

- 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.





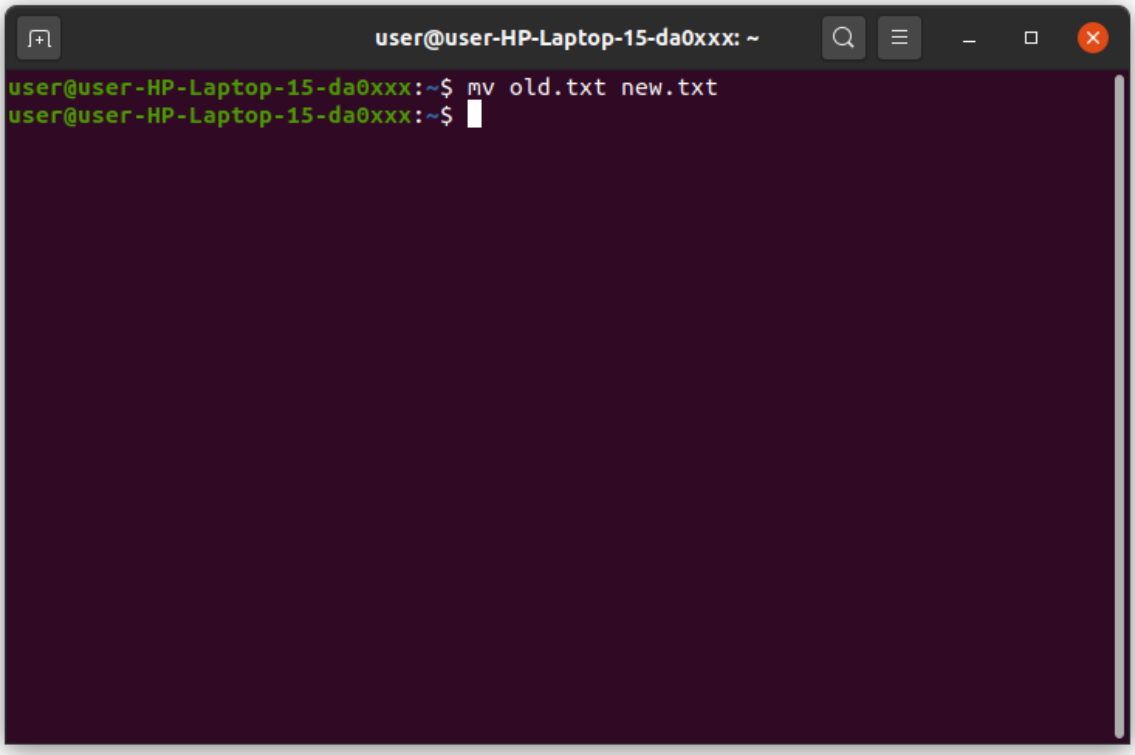


6. mv

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

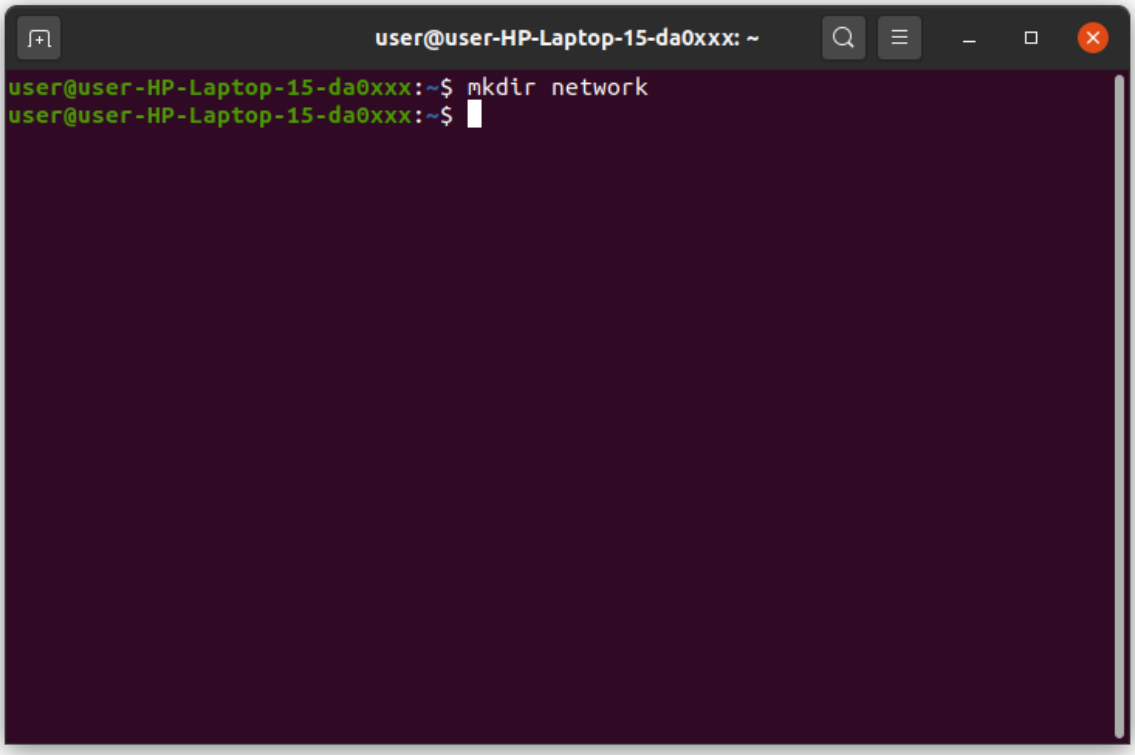
Music it will create a directory called Music.

There are extra mkdir commands as well:

- To generate a new directory inside another directory, use this Linux basic command

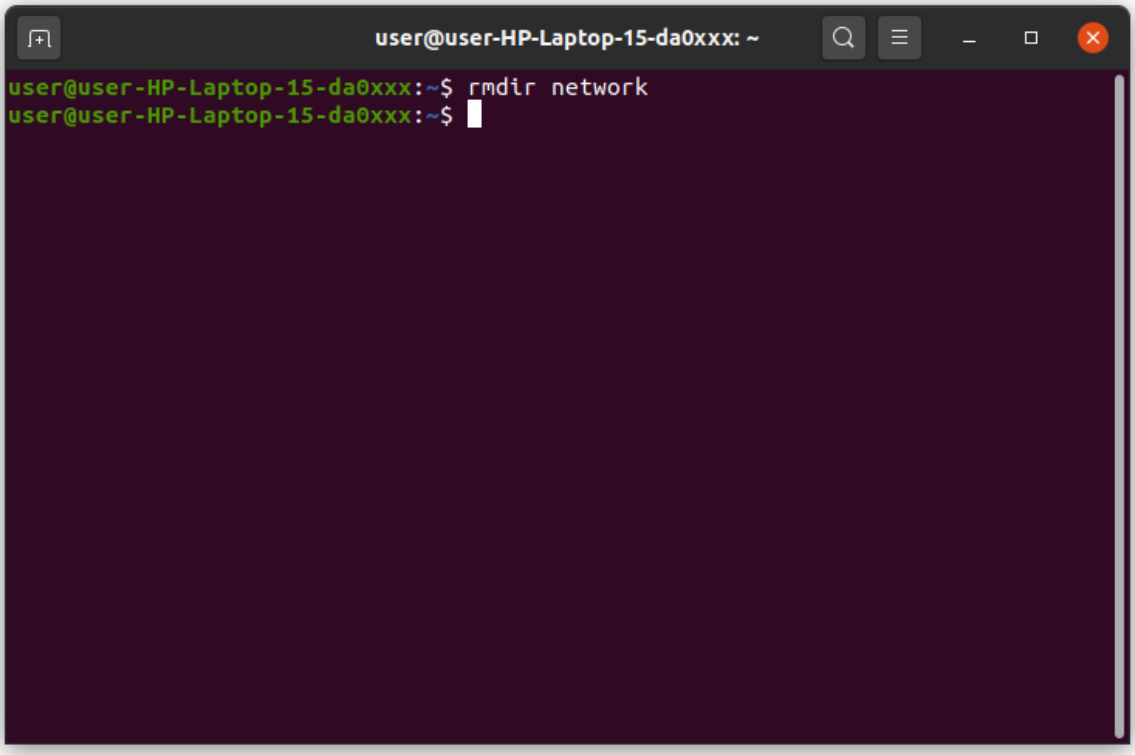
mkdir Music/Newfile

- 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

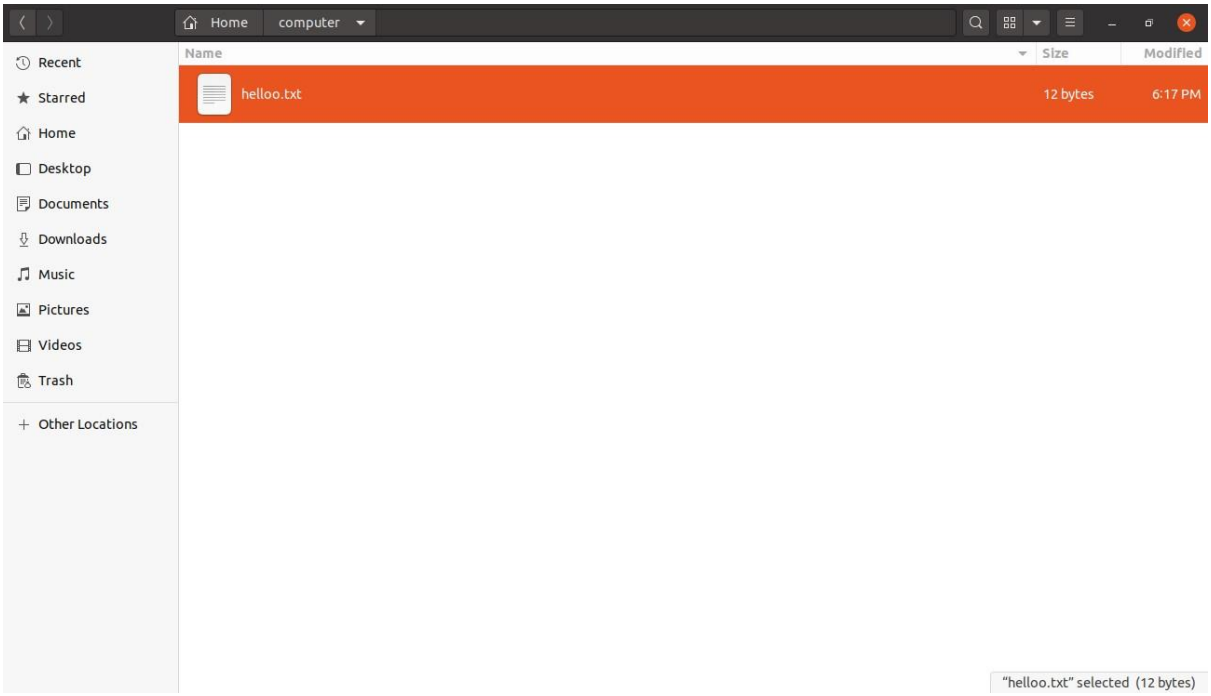
If you need to delete a directory, use the rmdir command. However, rmdir only allows you to delete empty directories.

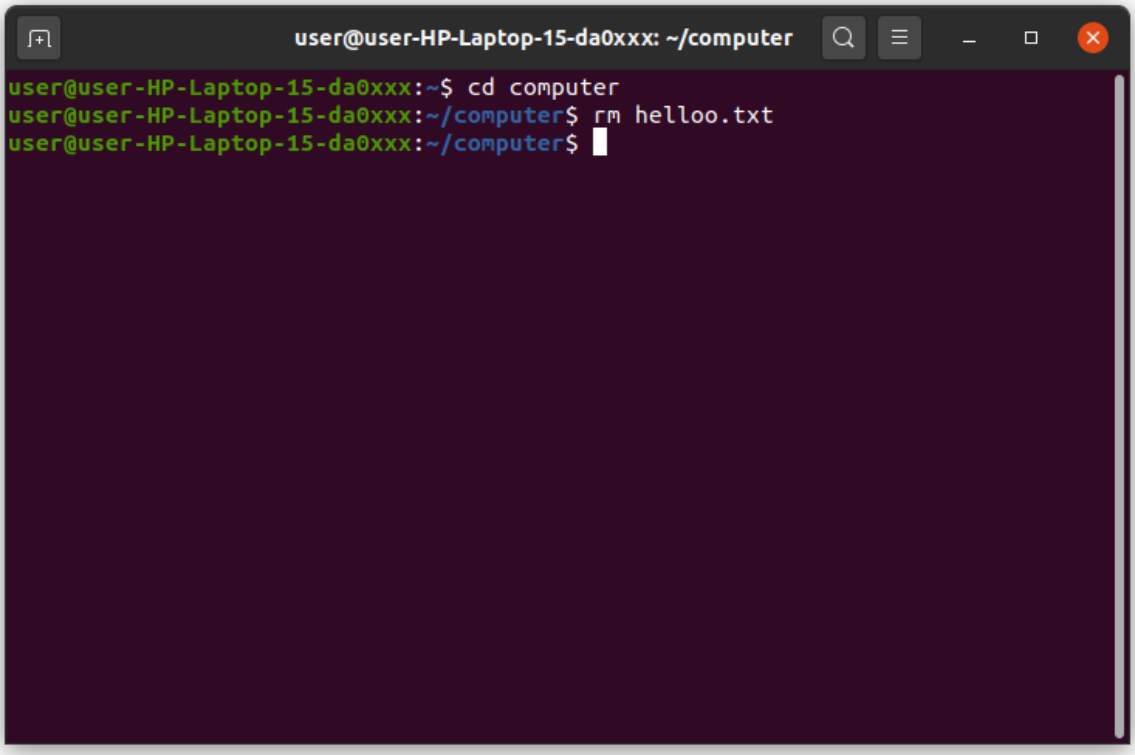


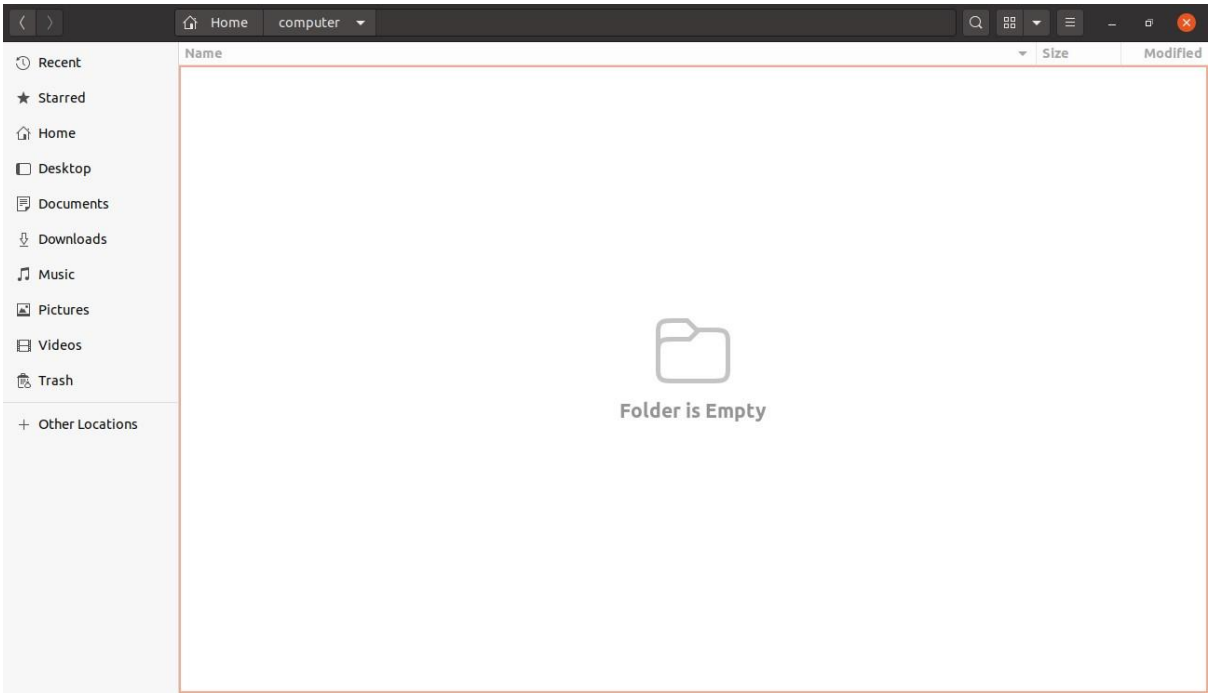
9. rm

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.



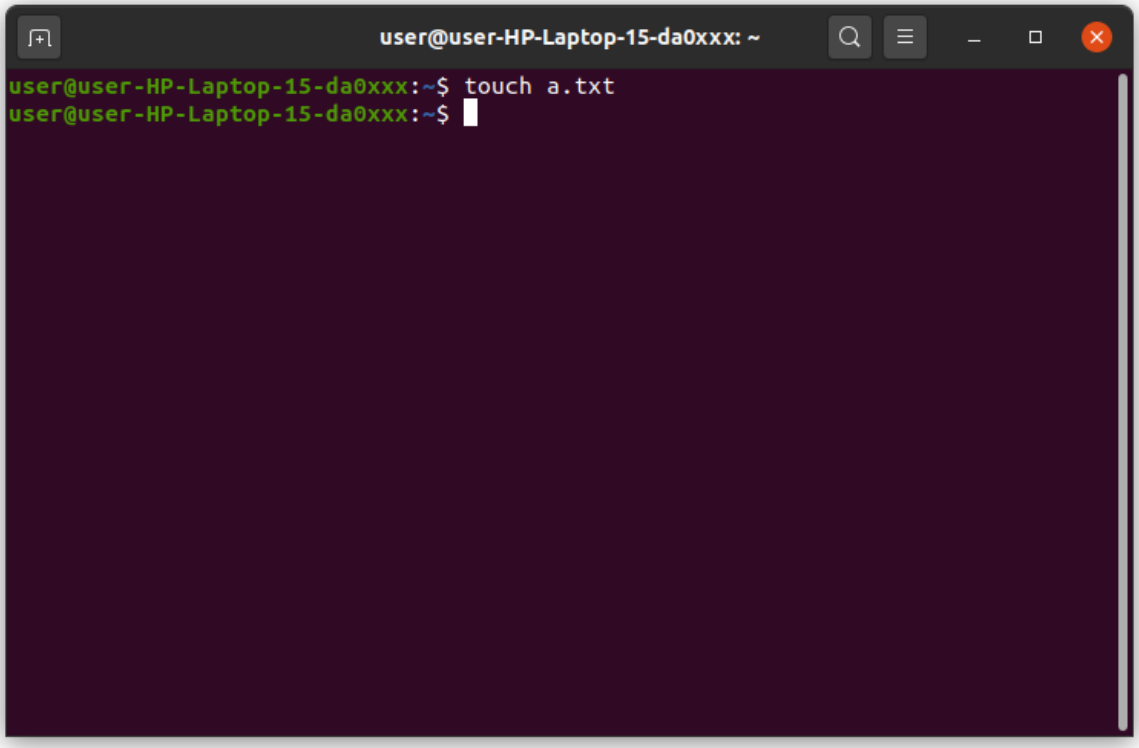




10. touch

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

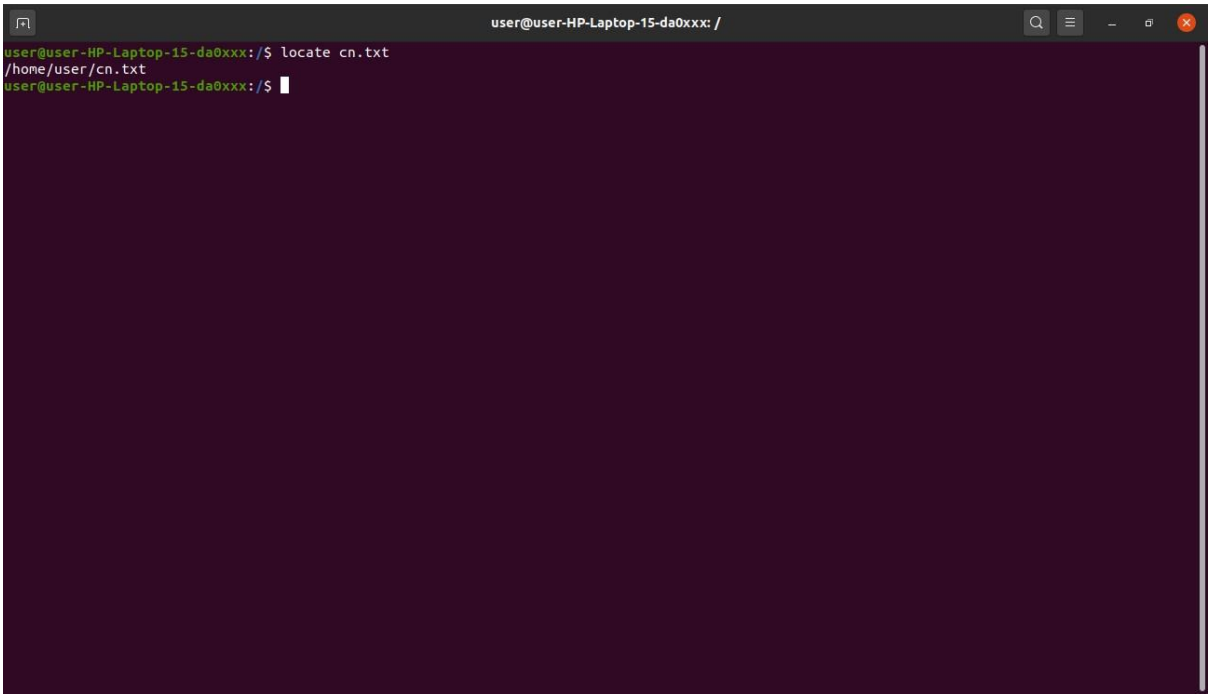
As an example, enter touch /home/username/Documents/Web.html to create an HTML file entitled Web under the Documents directory.



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 case-insensitive, 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.



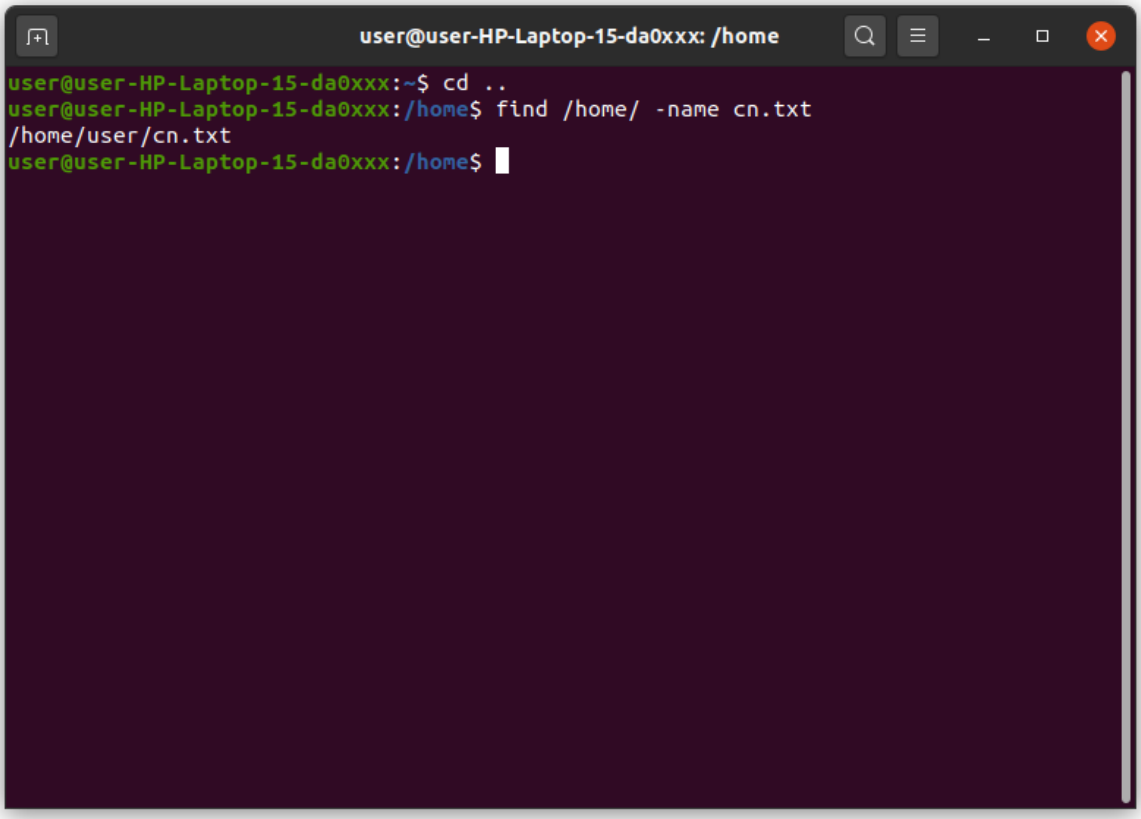
12. find

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

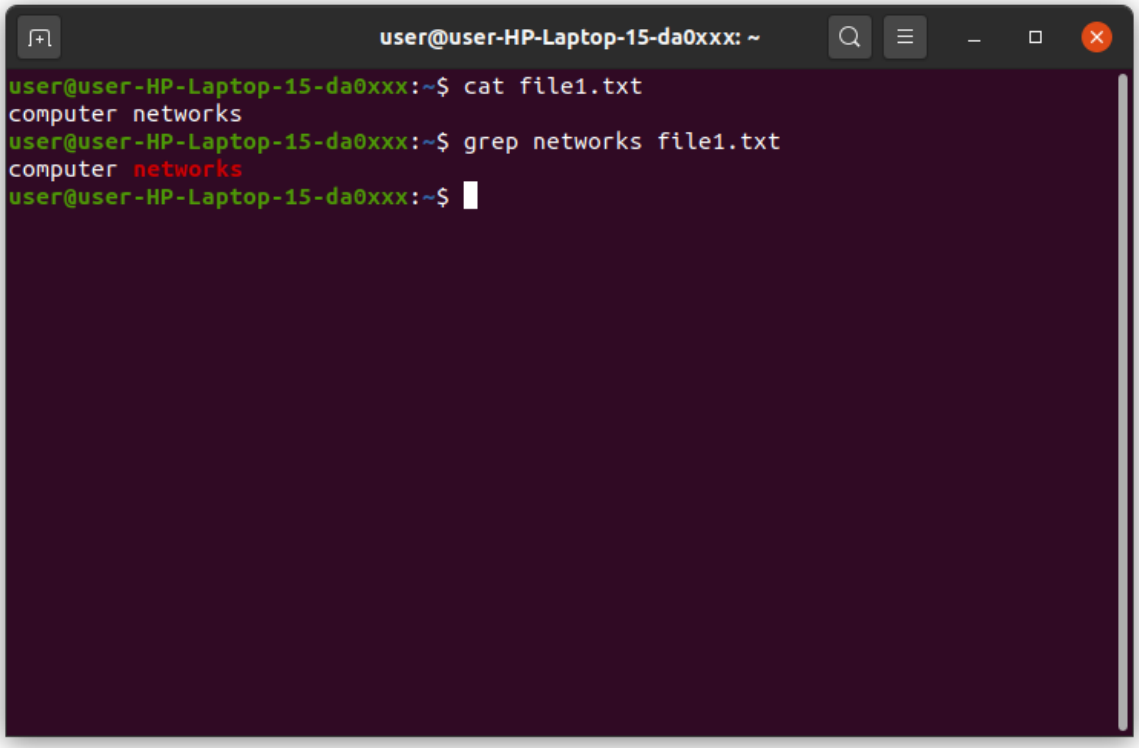


13. grep

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.

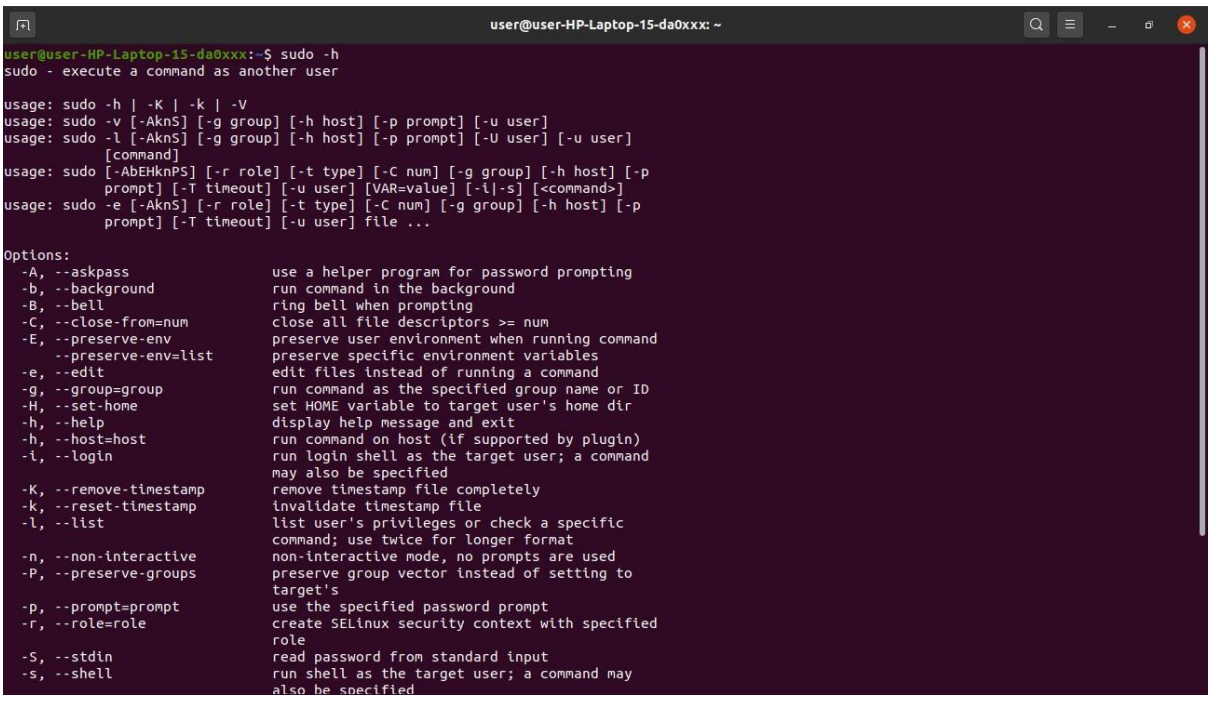
To illustrate, grep blue notepad.txt will search for the word blue in the notepad file. Lines that contain the searched word will be displayed fully. You should refer to some grep tutorial

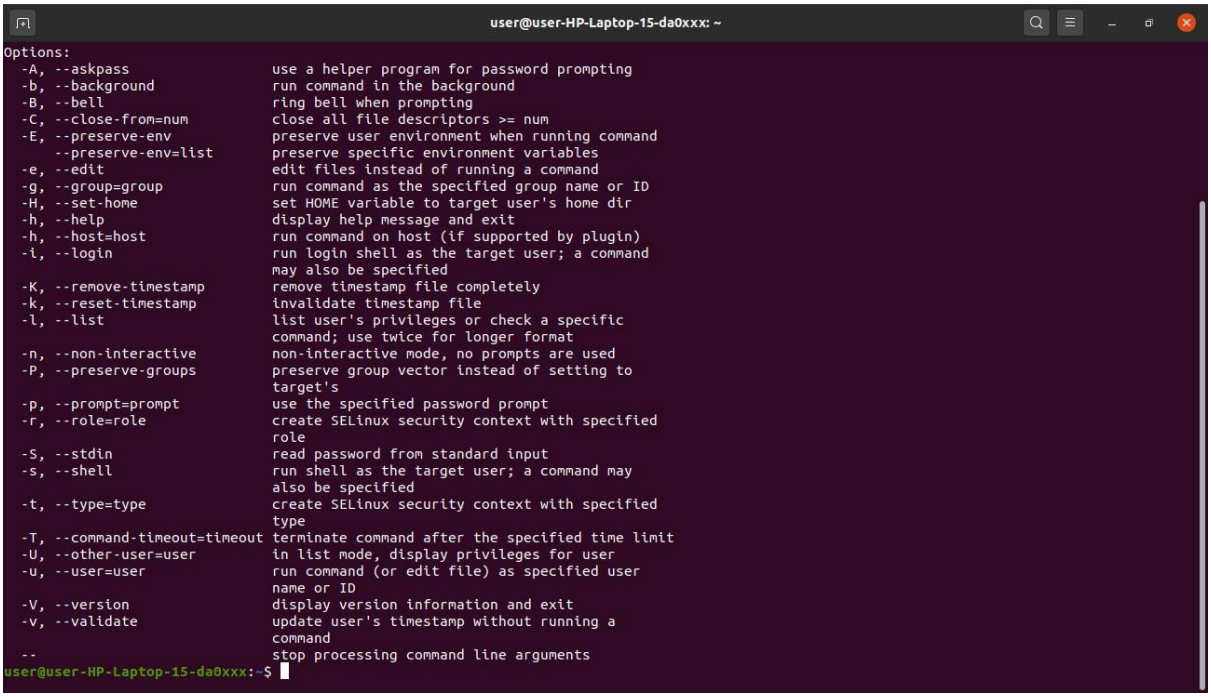
Useful for command line use as well. Usually output of a previous command is piped into the grep command. For example ls -l | grep “kernel”



14. sudo

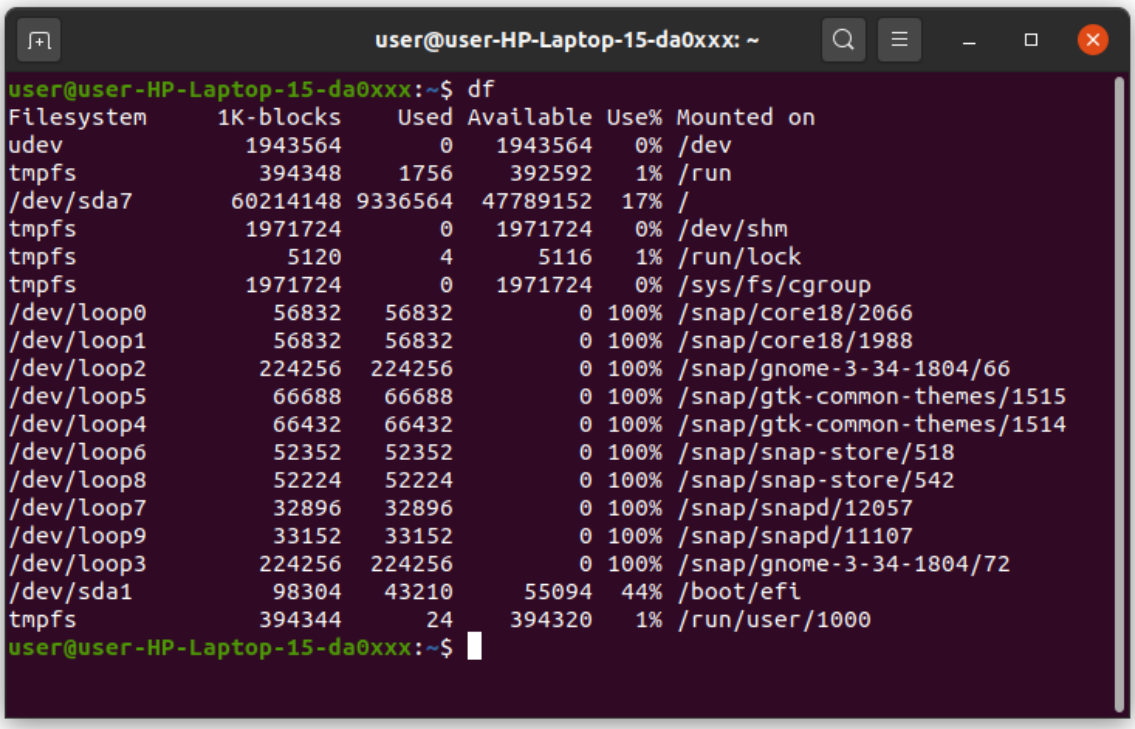
Short for “SuperUser Do”, this command enables you to perform tasks that require administrative or root permissions. You must have sufficient permissions to use this command.

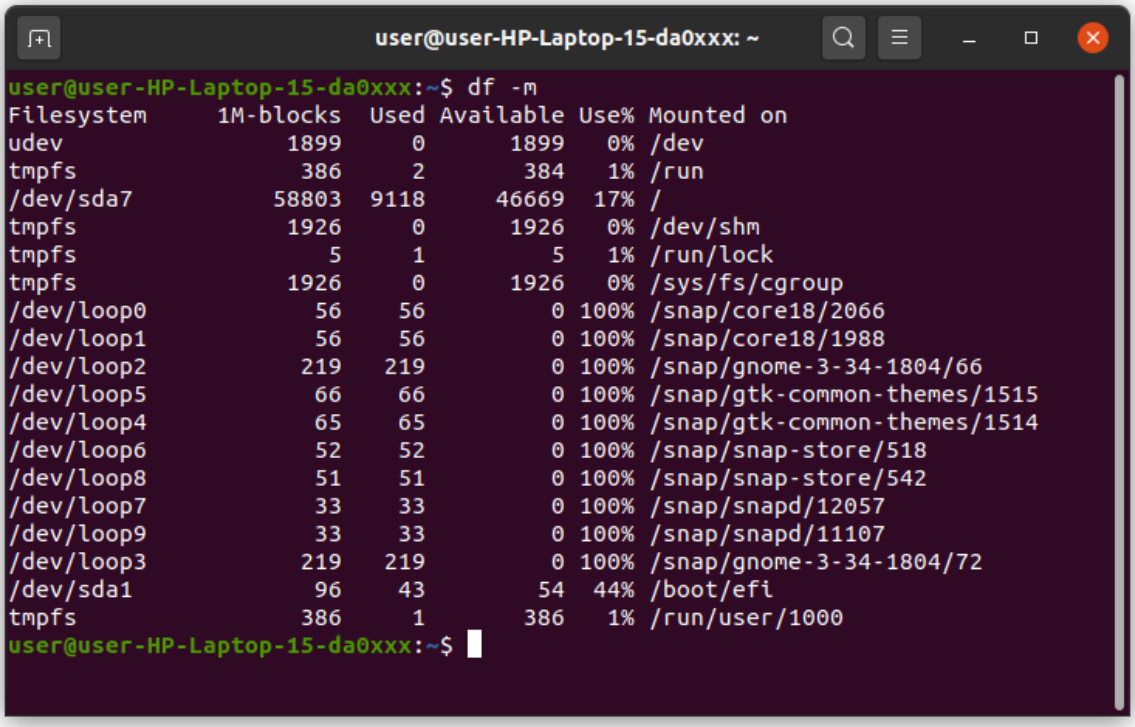




15. df

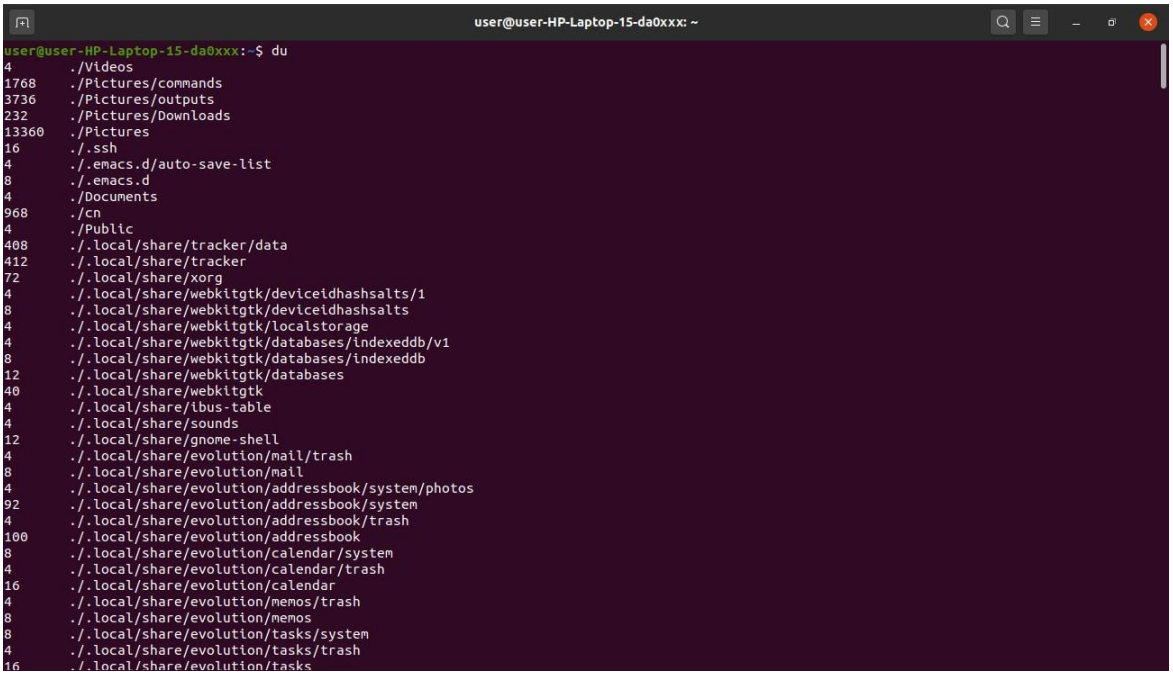
Use df command to get a report on the system's disk space usage, shown in percentage and KBs. If you want to see the report in megabytes, type df -m.

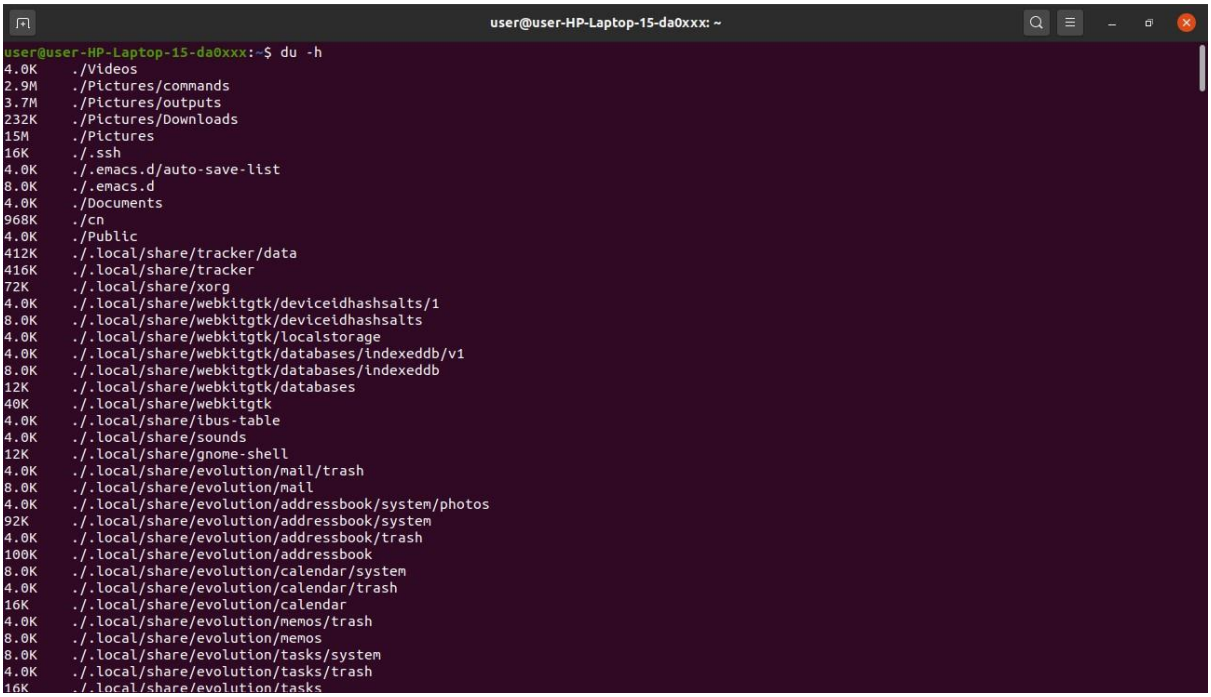




16. du

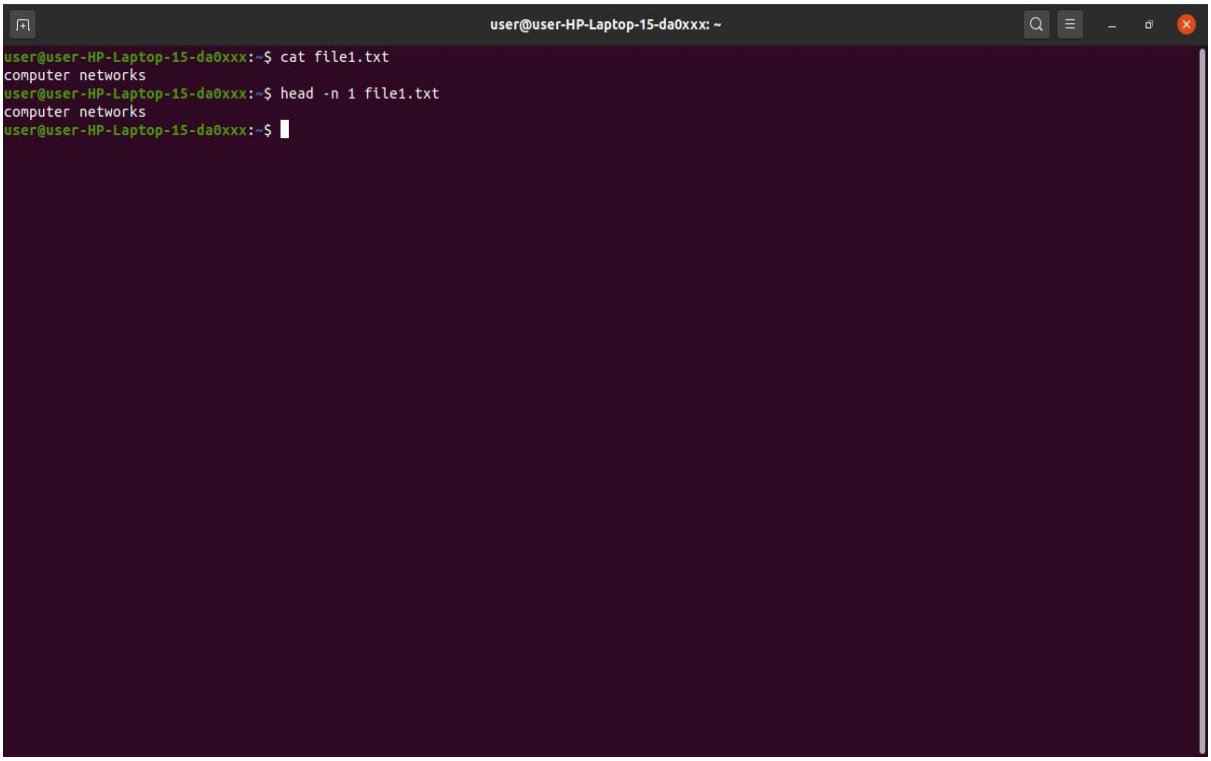
If you want to check how much space a file or a directory takes, the du (Disk Usage) command is the answer. However, the disk usage summary will show disk block numbers instead of the usual size format. If you want to see it in bytes, kilobytes, and megabytes, add the -h argument to the command line.





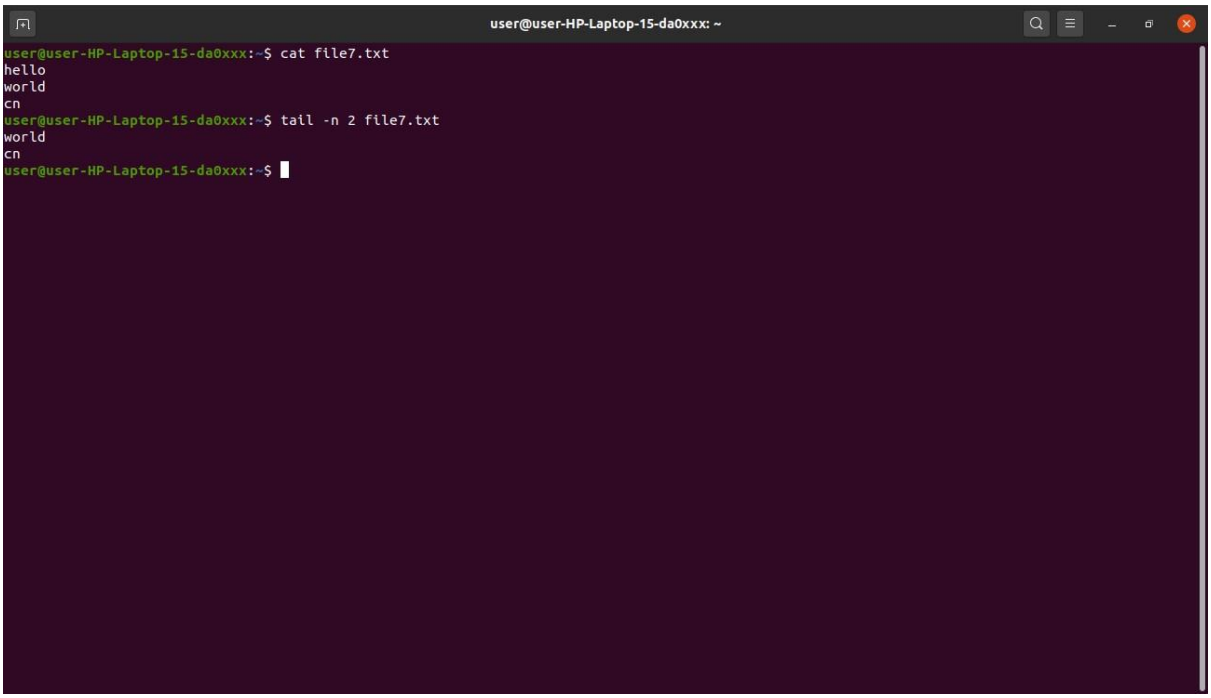
17. head

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. (Read the manual)



18. tail

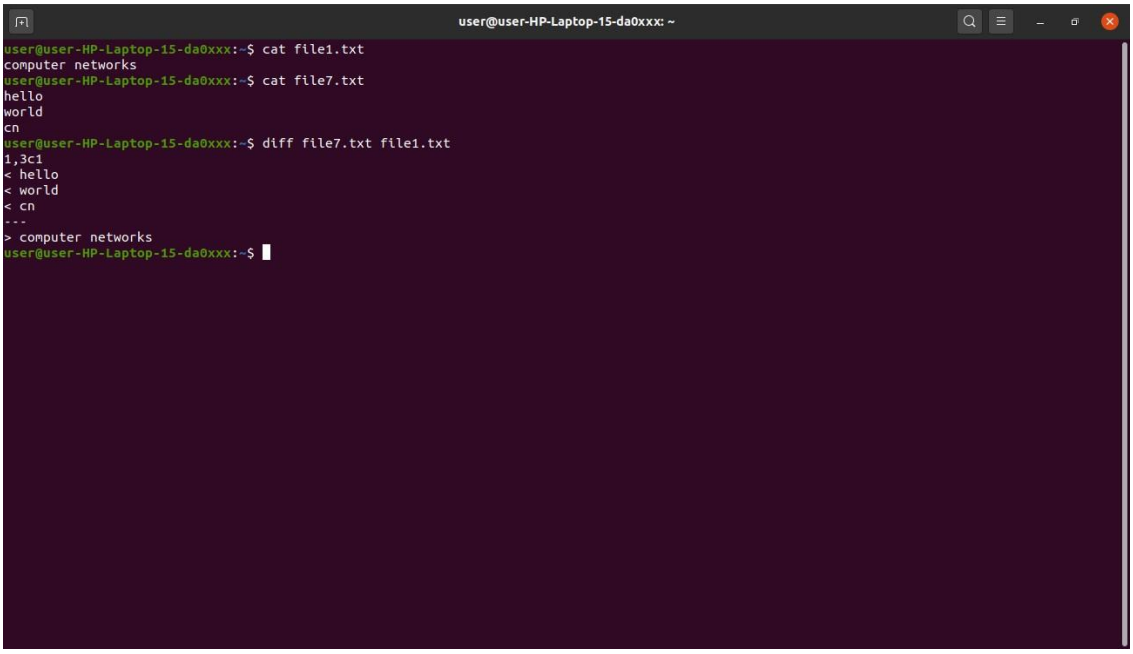
This one has a similar function to the head command, but instead of showing the first lines, the tail command will display the last ten lines of a text file. For example, tail -n filename.ext.



19. diff

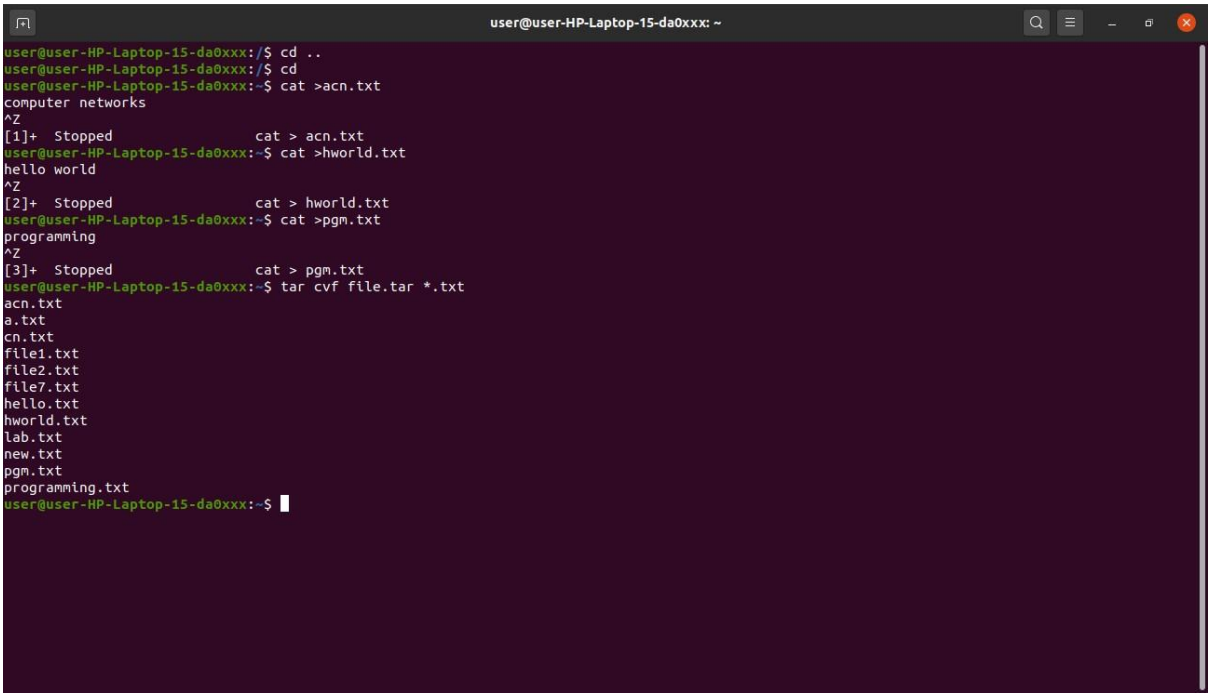
Short for difference, the diff command compares the contents of two files line by line. After analyzing the files, it will output the lines that do not match. Programmers often use this command when they need to make program alterations instead of rewriting the entire source code.

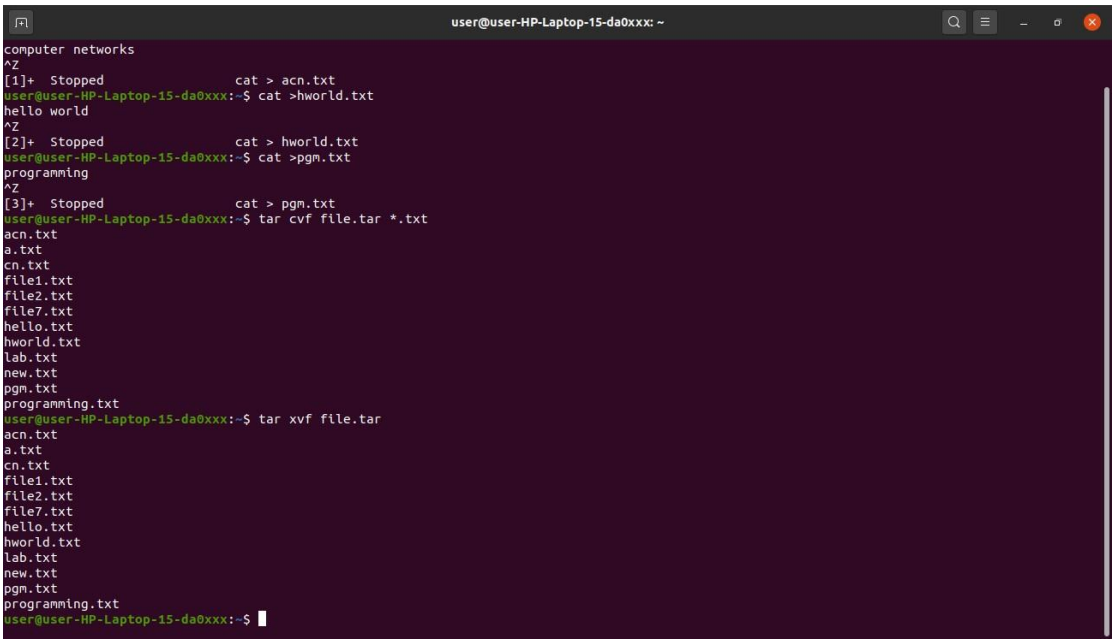
The simplest form of this command is diff file1.ext file2.ext



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.

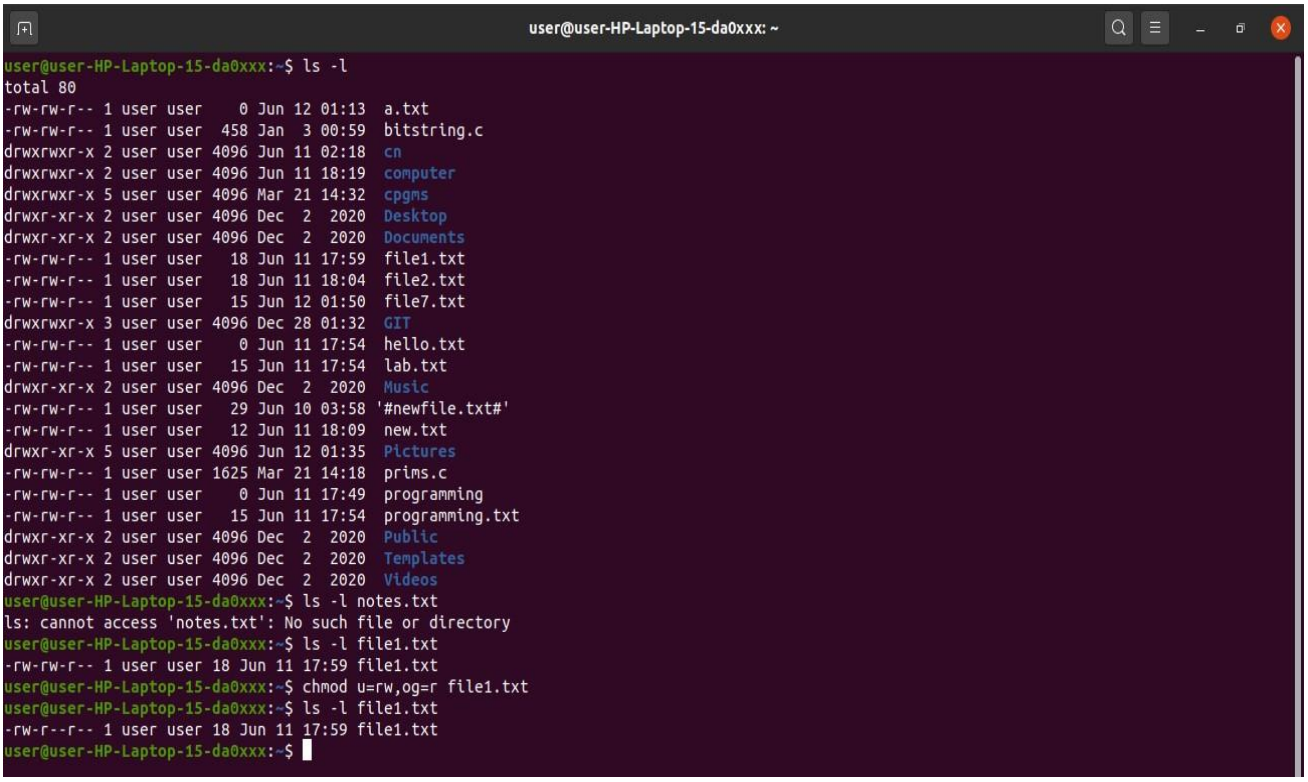




21. chmod

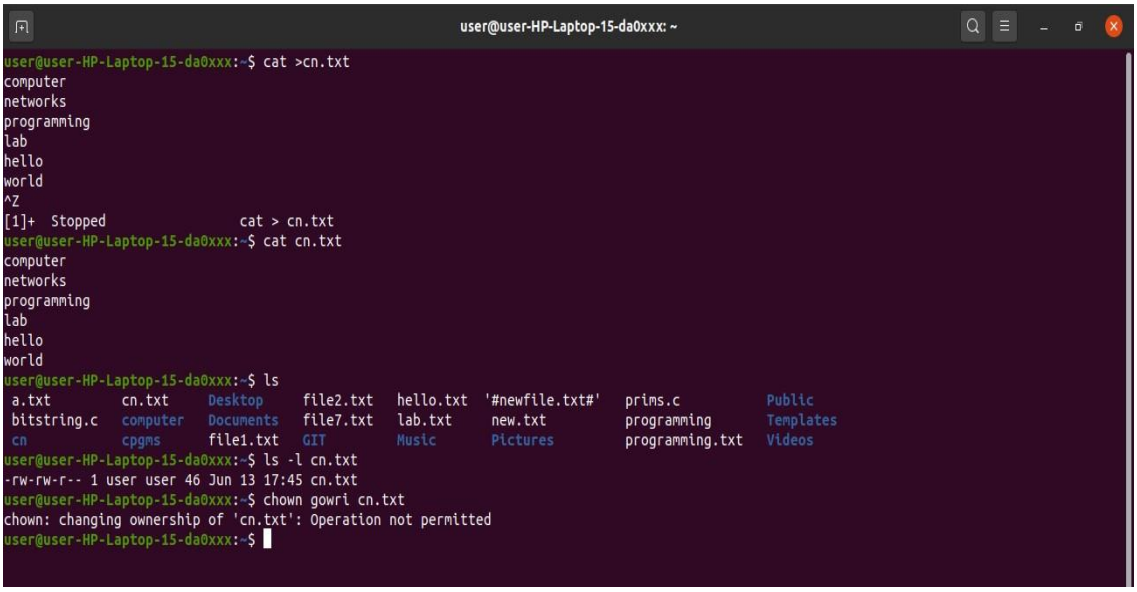
chmod is another Linux command, used to change the read, write, and execute permissions of files and directories. Read about permissions and how to manipulate them

.



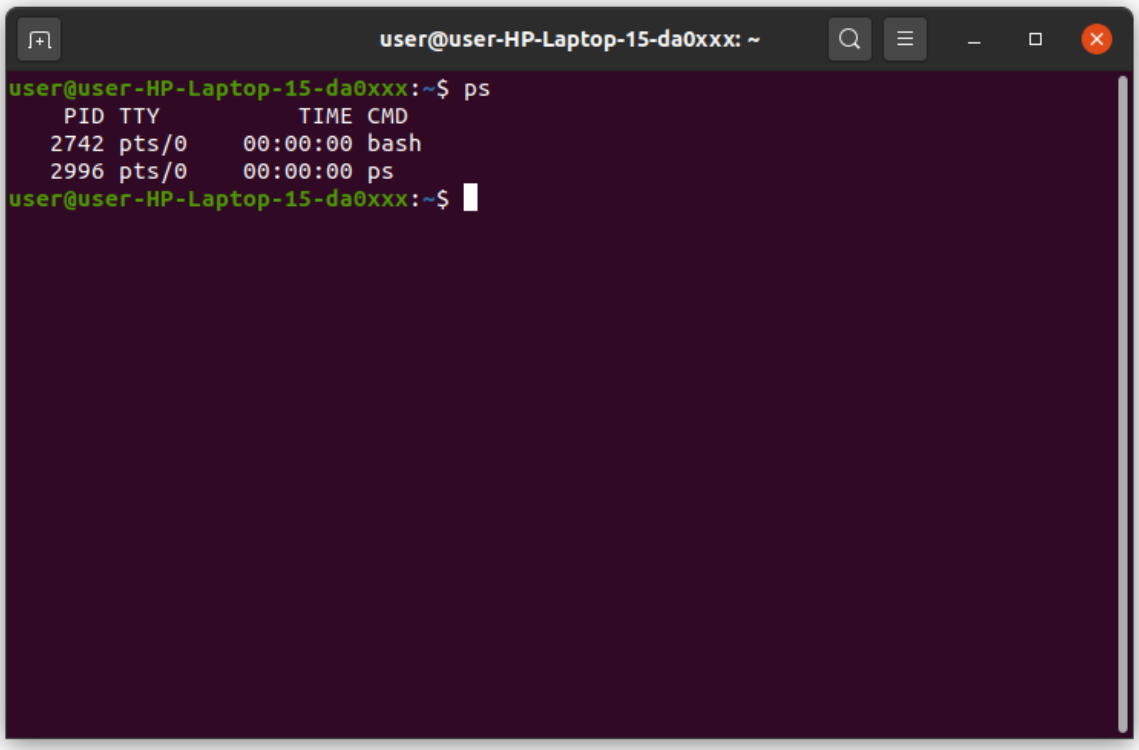
22. chown

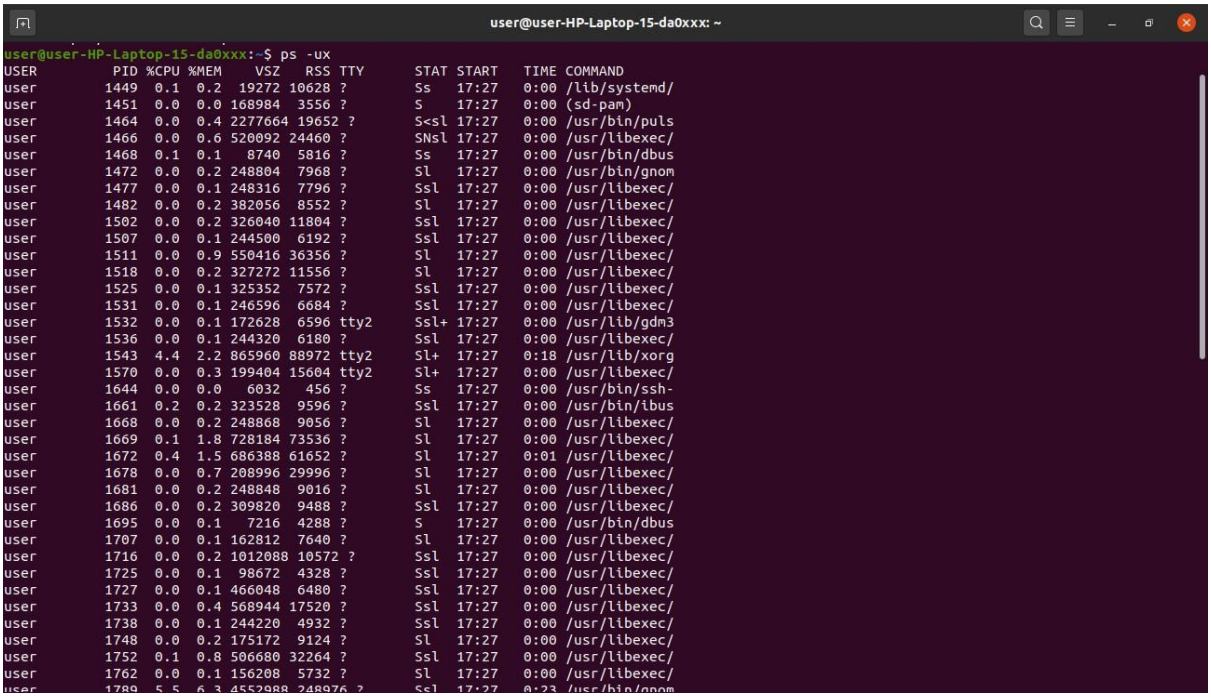
In Linux, all files are owned by a specific user. The chown command enables you to change or transfer the ownership of a file to the specified username. For instance, chown linuxuser2 file.ext will make linuxuser2 as the owner of the file.ext.



23. ps

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


user@user-HP-Laptop-15-da0xxx: ~
user@user-HP-Laptop-15-da0xxx:~$ ps
PID TTY TIME CMD
2742 pts/0 00:00:00 bash
2996 pts/0 00:00:00 ps
user@user-HP-Laptop-15-da0xxx:~$


user@user-HP-Laptop-15-da0xxx: ~
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 168984 3556 ? S 17:27 0:00 (sd-pam)
user 1464 0.0 0.4 2277664 19652 ? S<sl 17:27 0:00 /usr/bin/puls
user 1466 0.0 0.6 520092 24460 ? SNsl 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 248316 7796 ? Ssl 17:27 0:00 /usr/libexec/
user 1482 0.0 0.2 382056 8552 ? Sl 17:27 0:00 /usr/libexec/
user 1502 0.0 0.2 326040 11804 ? Ssl 17:27 0:00 /usr/libexec/
user 1507 0.0 0.1 244500 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/gdm3
user 1536 0.0 0.1 244320 6180 ? Ssl 17:27 0:00 /usr/libexec/
user 1543 4.4 2.2 865960 88972 tty2 Sl+ 17:27 0:18 /usr/lib/xorg
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 323528 9596 ? Ssl 17:27 0:00 /usr/bin/ibus
user 1668 0.0 0.2 248868 9056 ? Sl 17:27 0:00 /usr/libexec/
user 1669 0.1 1.8 728184 73536 ? Sl 17:27 0:00 /usr/libexec/
user 1672 0.4 1.5 686388 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 248848 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 568944 17520 ? Ssl 17:27 0:00 /usr/libexec/
user 1738 0.0 0.1 244220 4932 ? 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 32264 ? 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:

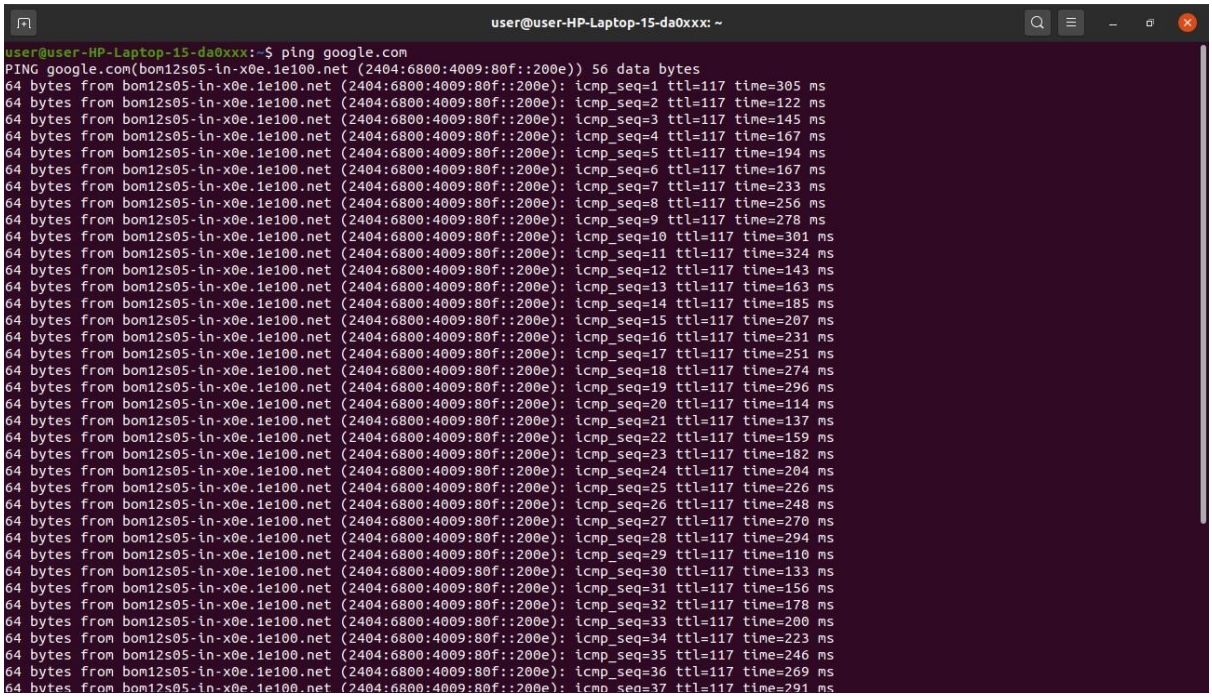
- SIGTERM (15) — requests a program to stop running and gives it some time to save all of its progress. If you don't specify the signal when entering the kill command, this signal will be used.
- SIGKILL (9) — forces programs to stop immediately. Unsaved progress will be lost. Besides knowing the signals, 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

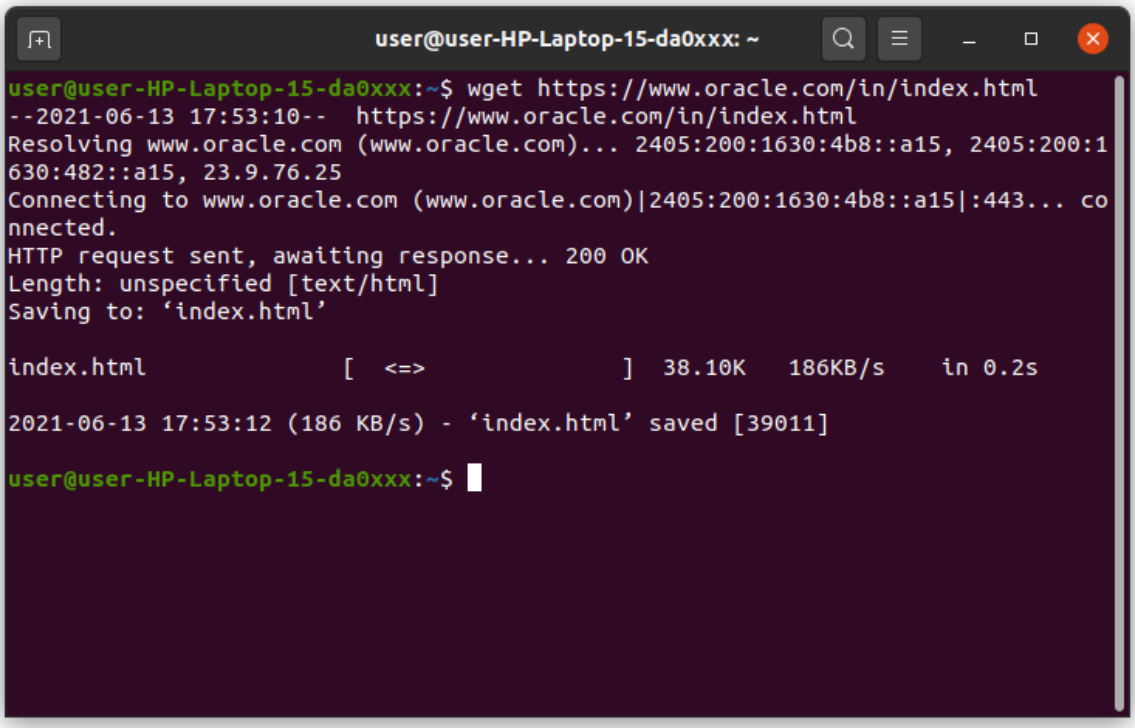
25. ping

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.



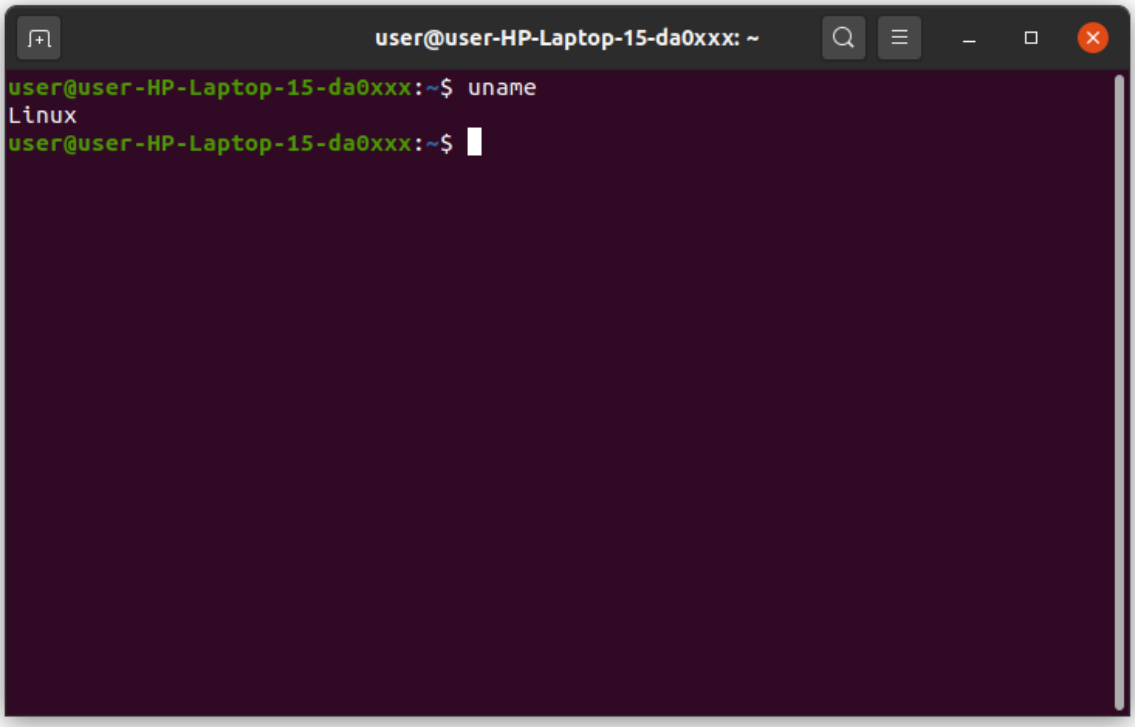
26. wget

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.



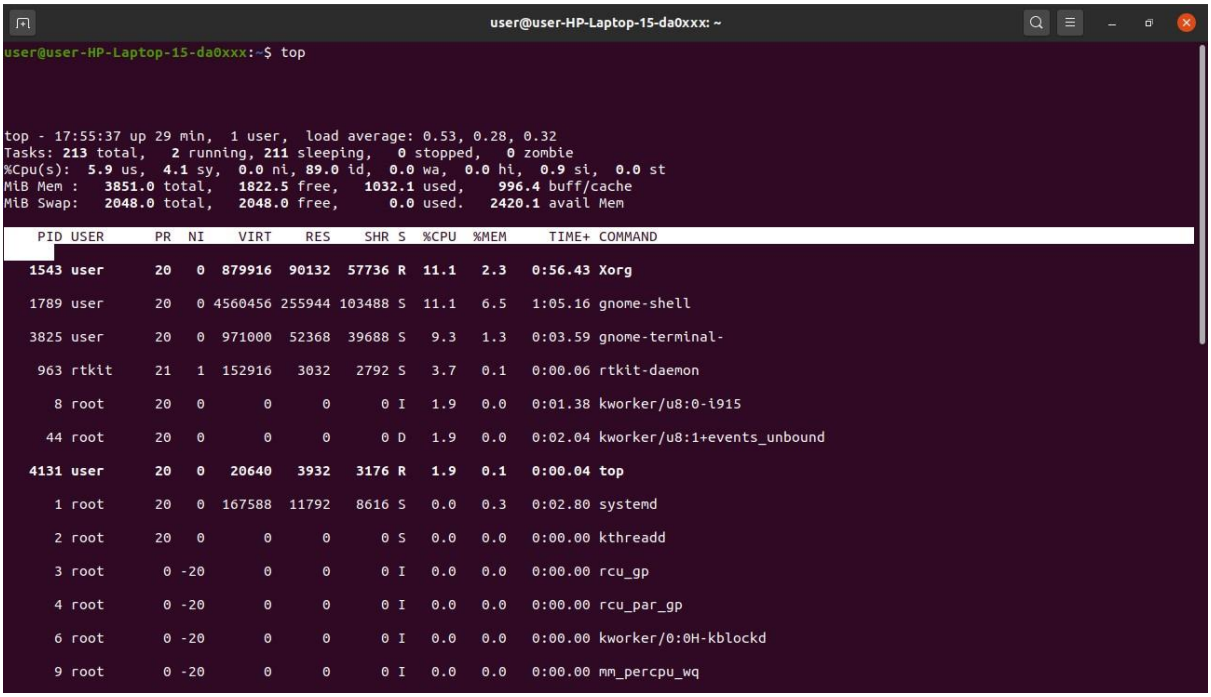
27. uname

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.



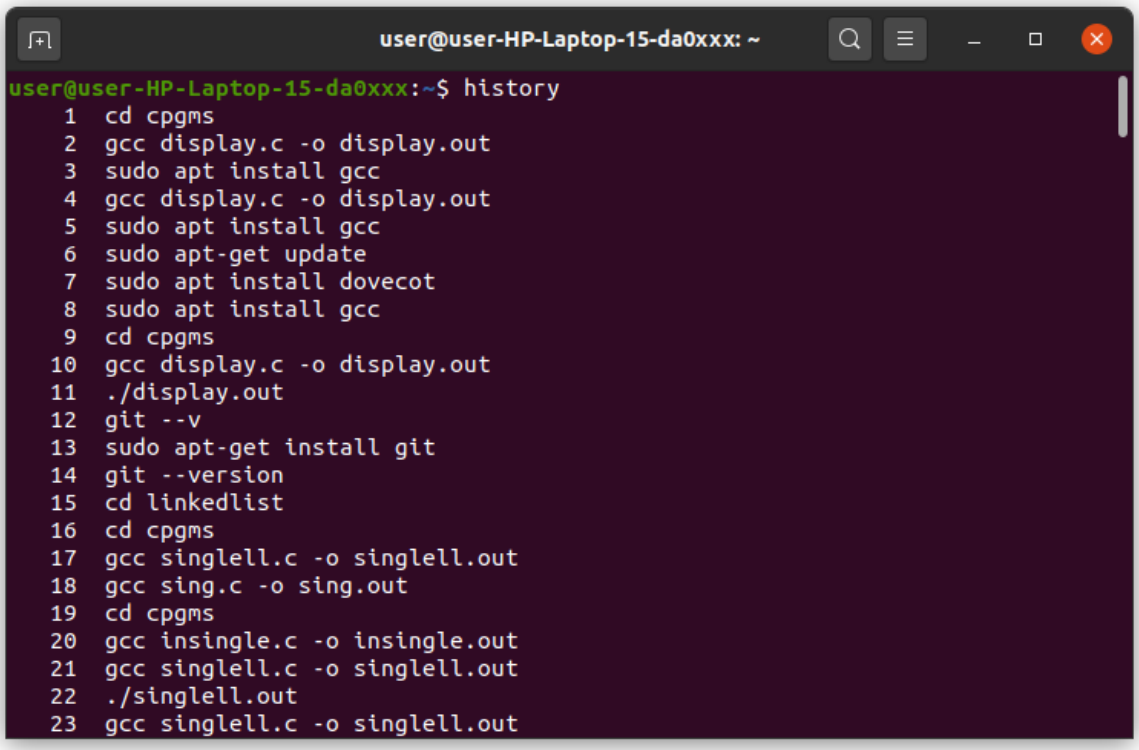
28. top

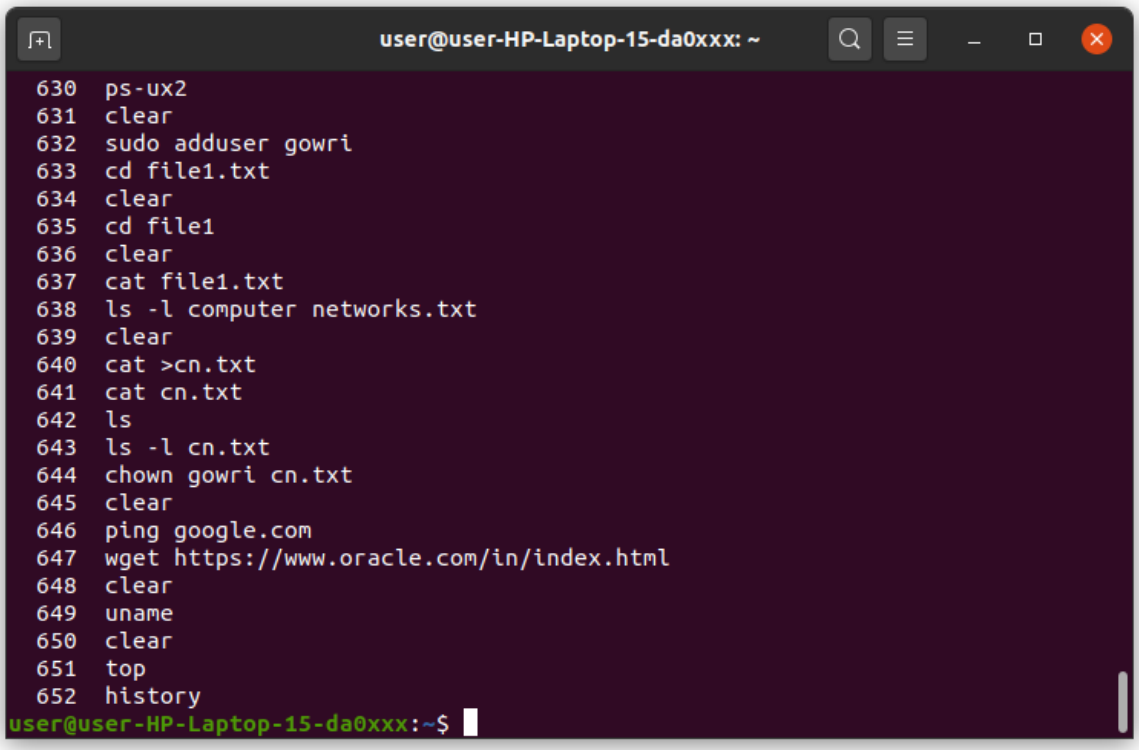
As a terminal equivalent to Task Manager in Windows, the top command will display a list of running processes and how much CPU each process uses. It's very useful to monitor system resource usage, especially knowing which process needs to be terminated because it consumes too many resources.



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.

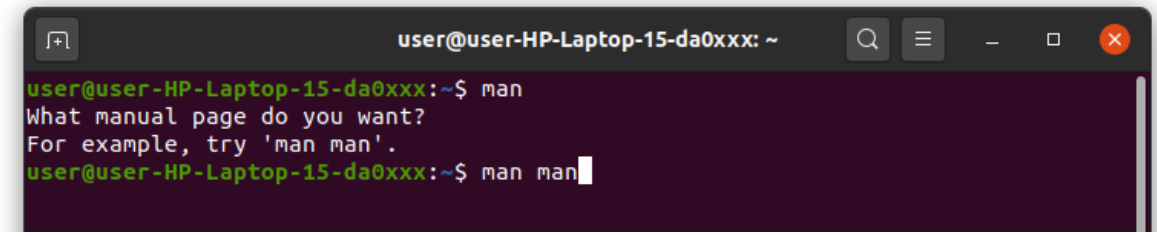


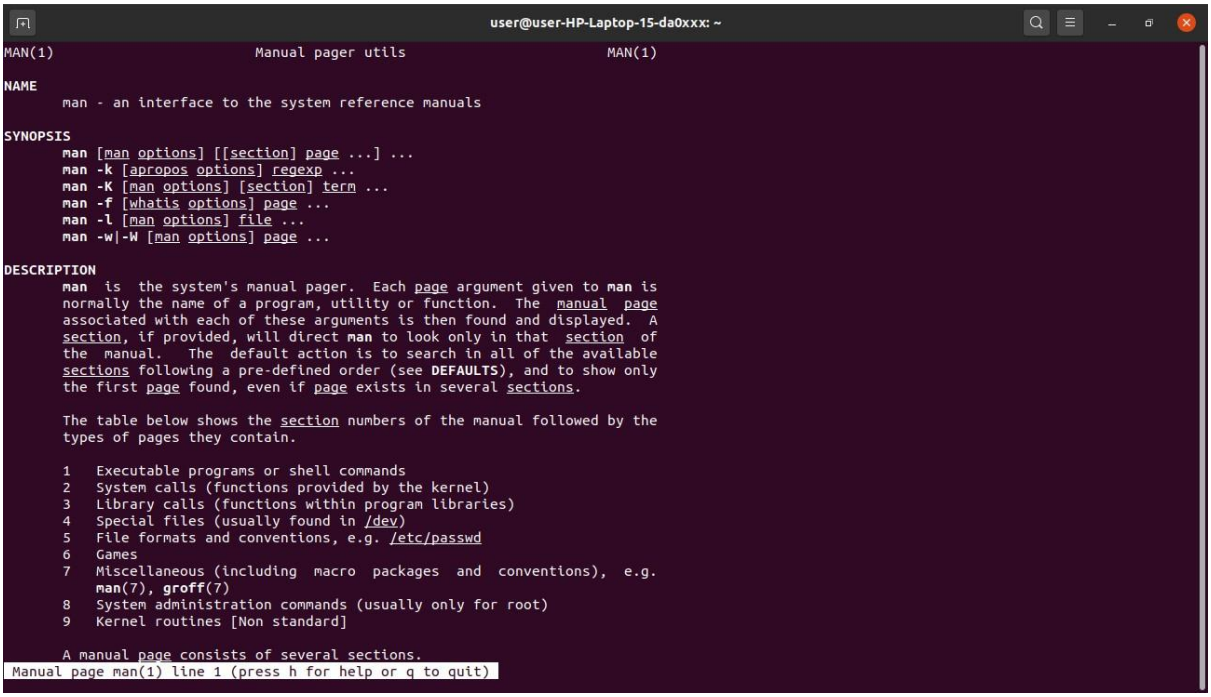


30. man

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.

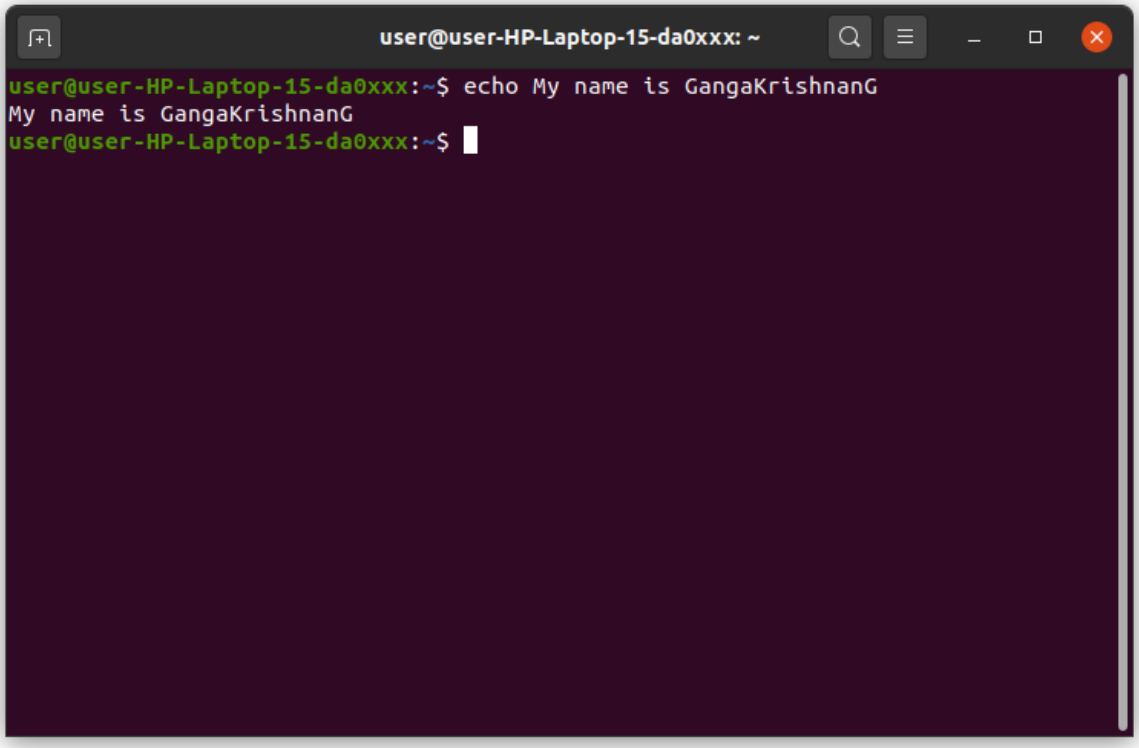
Use the command: man man to start learning about man utility.





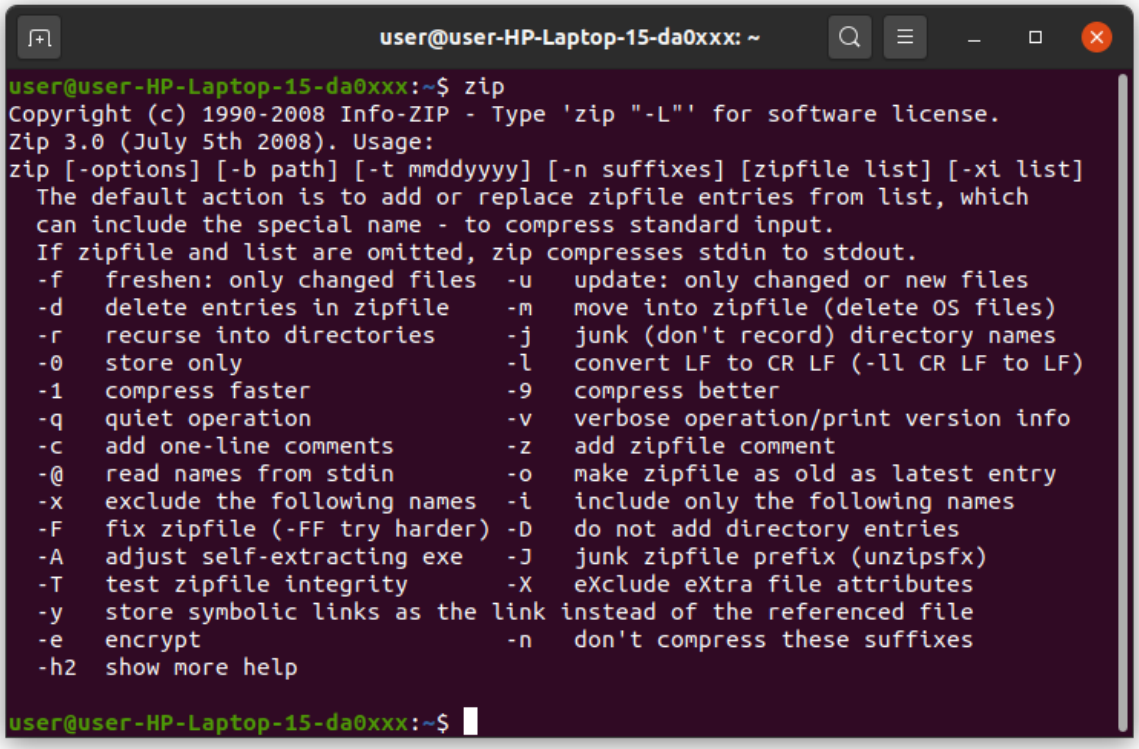
31. echo

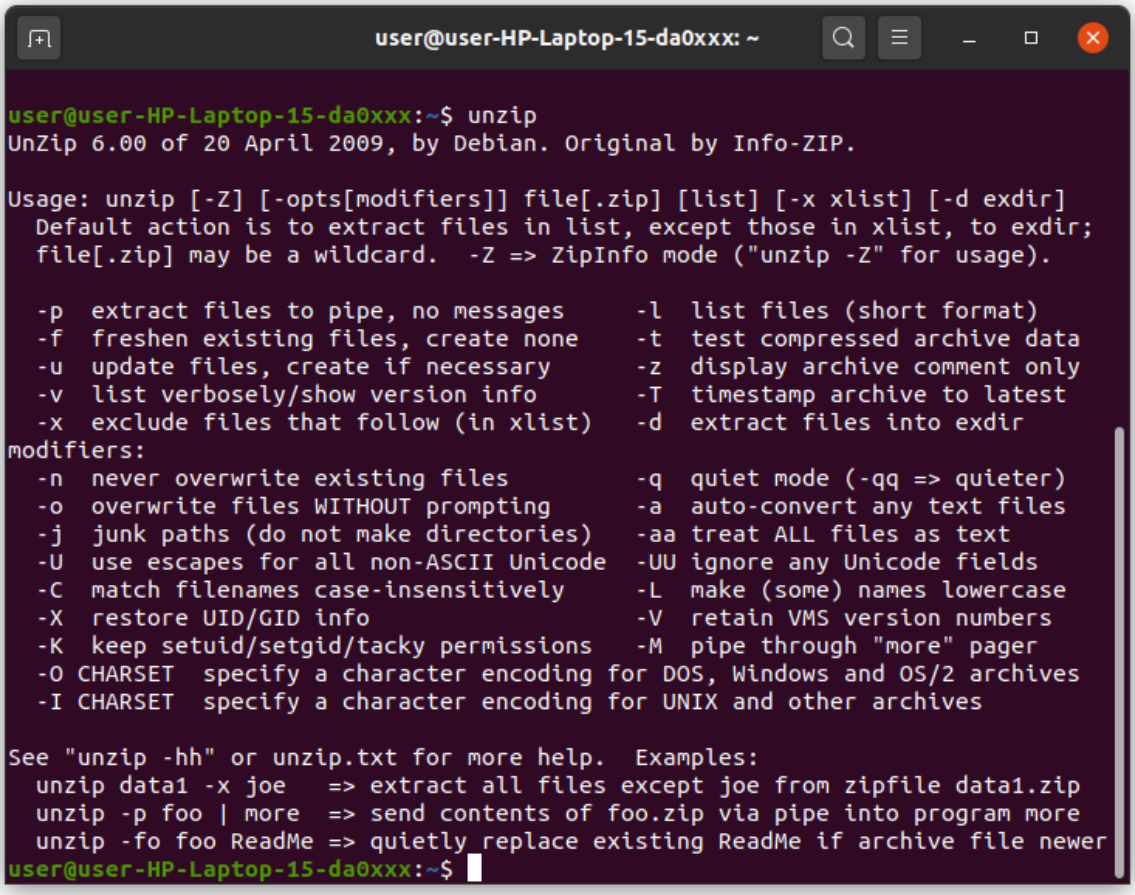
This command is used to move some data into a file. For example, if you want to add the text, “Hello, my name is John” into a file called name.txt, you would type echo Hello, my name is John >> name.txt



32. zip, unzip

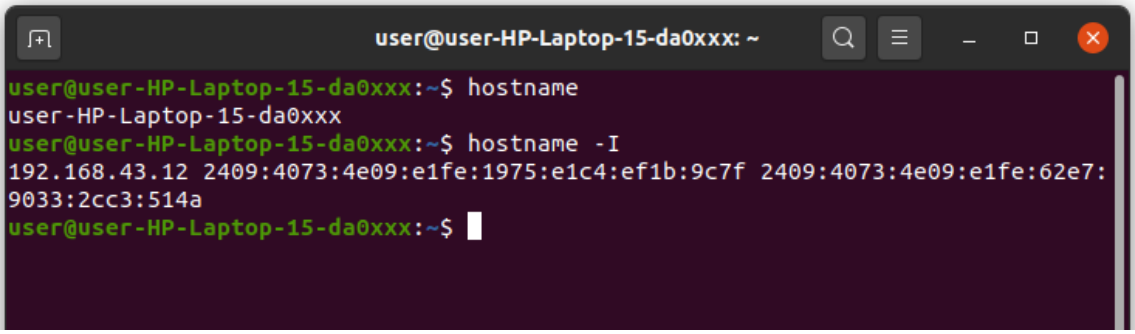
Use the zip command to compress your files into a zip archive, and use the unzip command to extract the zipped files from a zip archive. (This program should be installed , some distributions may not have them. You can also look at gzip and bzip commands)





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.

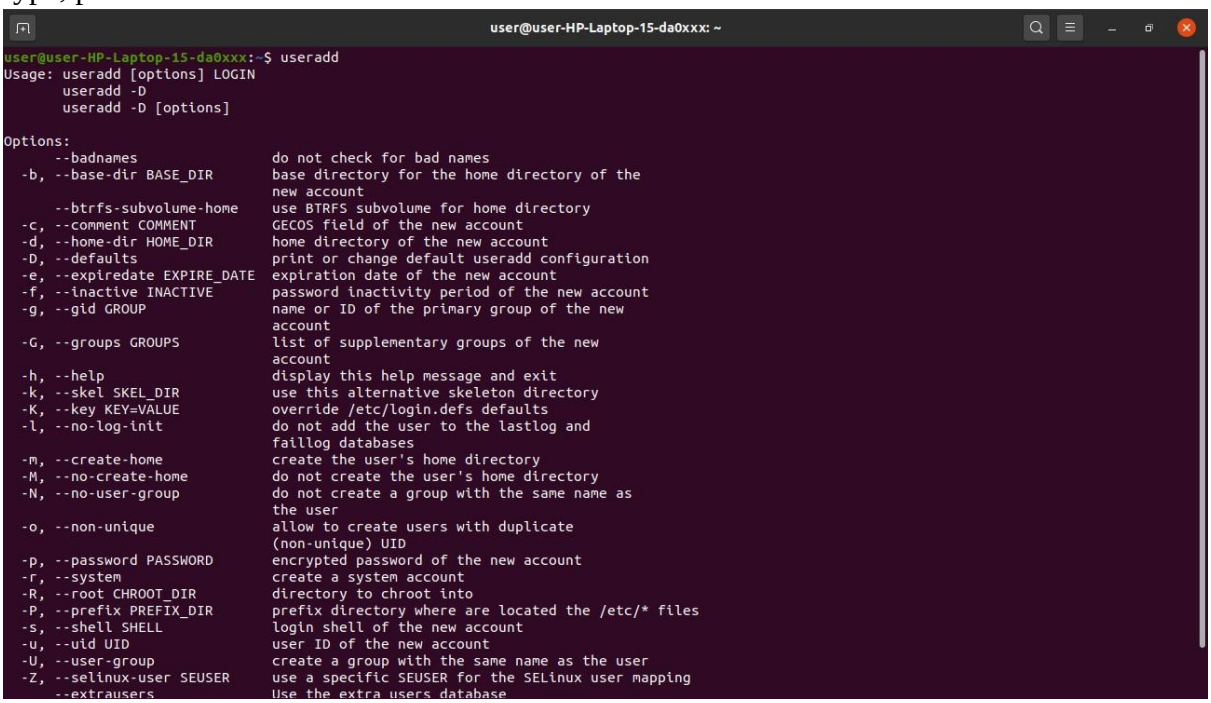


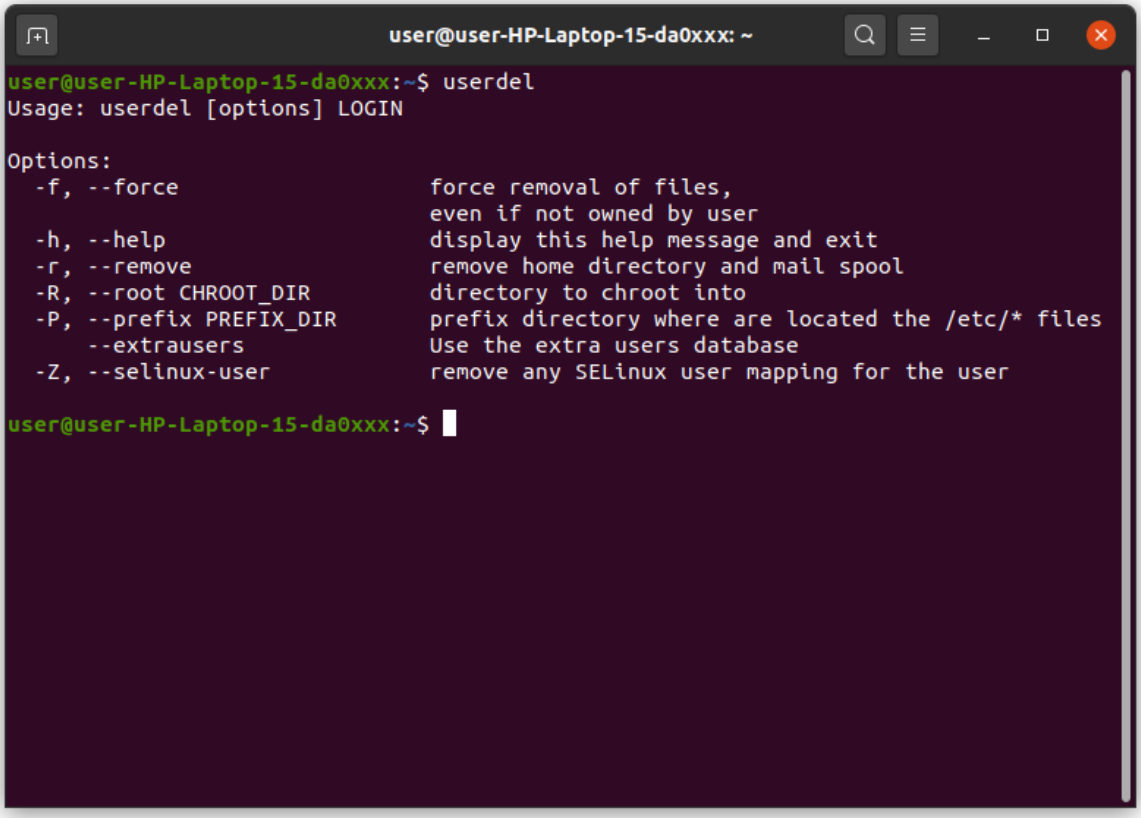
34. useradd, userdel

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.





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.

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.

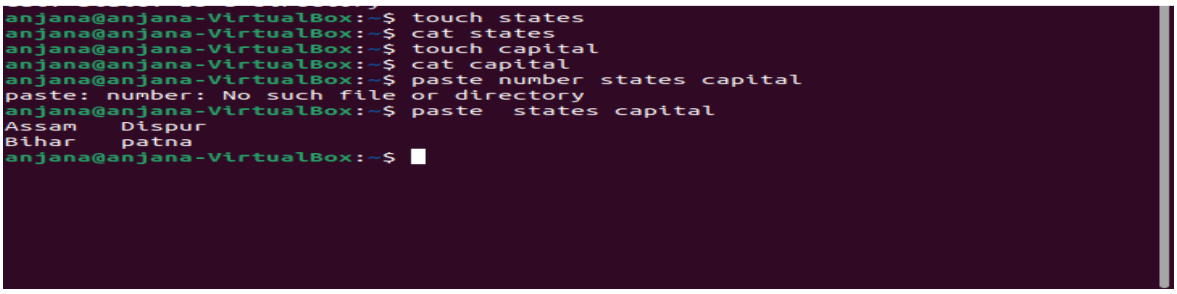
37.cut

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 precedes 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

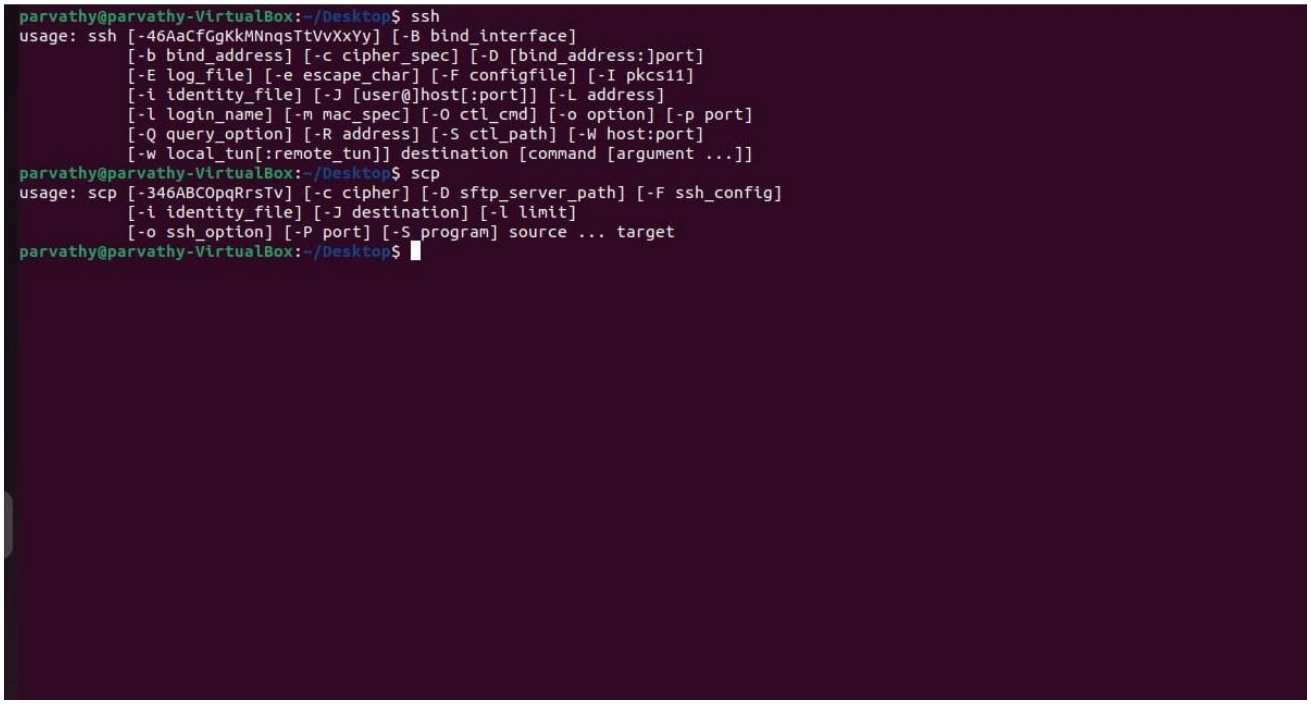
Paste command is one of the useful commands in Unix or Linux operating system. It is used to join files horizontally (parallel merging) by outputting lines consisting of lines from each file specified, separated by tab as delimiter, to the standard output. When no file is specified, or put dash ("-")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



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. ssh-keygen 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.