

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.

```
vboxuser@Ubuntu:~/NetworkLab$ pwd
/home/vboxuser/NetworkLab
vboxuser@Ubuntu:~/NetworkLab$
```

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.

```
vboxuser@Ubuntu:~$ cd Network
vboxuser@Ubuntu:~/Network$ cd ..
vboxuser@Ubuntu:~$ cd -
/home/vboxuser/Network
vboxuser@Ubuntu:~/Network$ cd
vboxuser@Ubuntu:~$
```

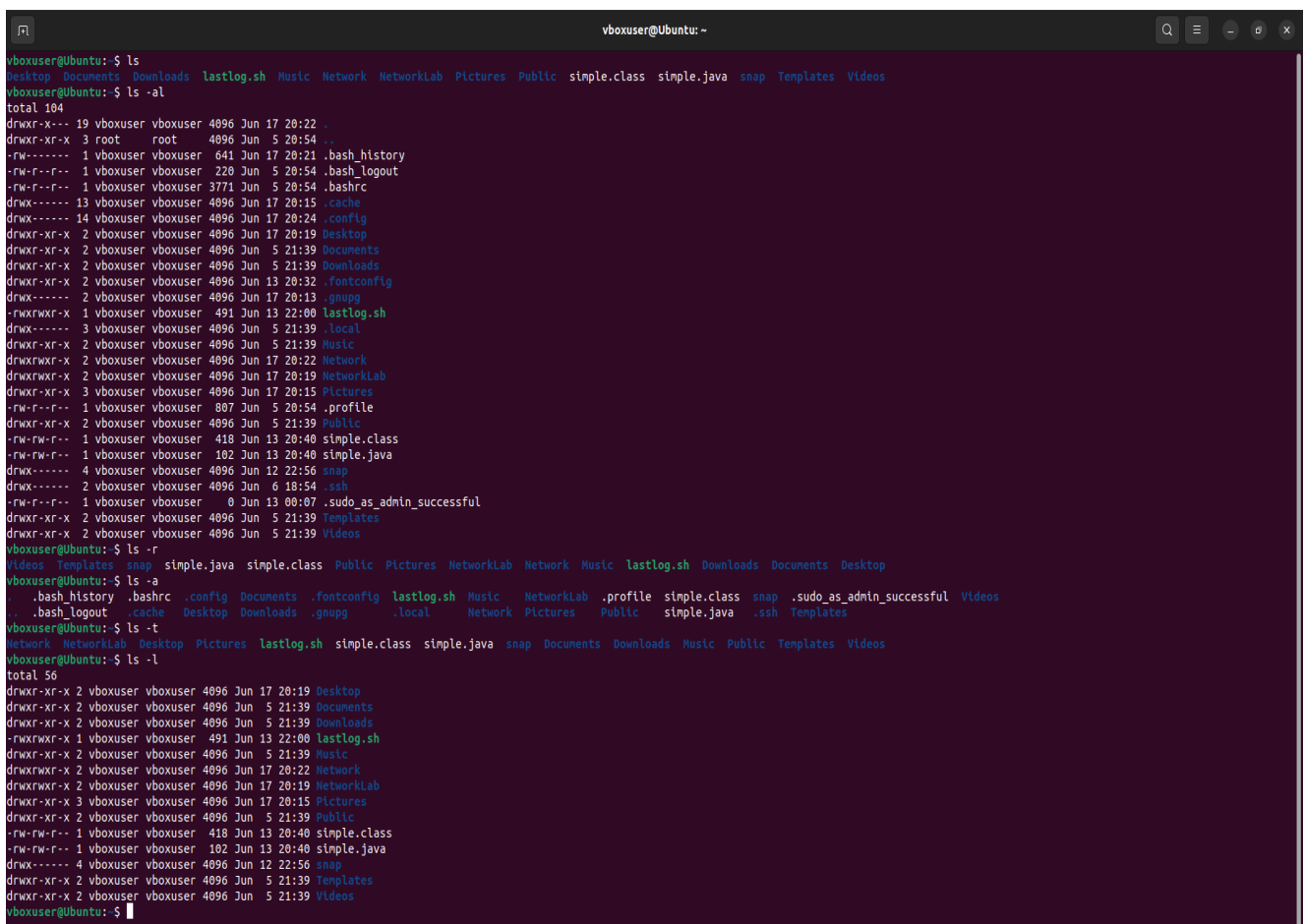
3. ls

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.



```
vboxuser@Ubuntu: ~
vboxuser@Ubuntu:~$ ls
Desktop Documents Downloads lastlog.sh Music Network NetworkLab Pictures Public simple.class simple.java snap Templates Videos
vboxuser@Ubuntu:~$ ls -al
total 104
drwxr-x--- 19 vboxuser vboxuser 4096 Jun 17 20:22 .
drwxr-xr-x  3 root      root      4096 Jun  5 20:54 ..
-rw-r----- 1 vboxuser vboxuser 641 Jun 17 20:21 .bash_history
-rw-r--r--  1 vboxuser vboxuser 220 Jun  5 20:54 .bash_logout
-rw-r--r--  1 vboxuser vboxuser 3771 Jun  5 20:54 .bashrc
drwx----- 13 vboxuser vboxuser 4096 Jun 17 20:15 .cache
drwx----- 14 vboxuser vboxuser 4096 Jun 17 20:24 .config
drwxr-xr-x  2 vboxuser vboxuser 4096 Jun 17 20:19 Desktop
drwxr-xr-x  2 vboxuser vboxuser 4096 Jun  5 21:39 Documents
drwxr-xr-x  2 vboxuser vboxuser 4096 Jun  5 21:39 Downloads
drwxr-xr-x  2 vboxuser vboxuser 4096 Jun 13 20:32 .fontconfig
drwx----- 2 vboxuser vboxuser 4096 Jun 17 20:13 .gnupg
-rwxrwxr-x  1 vboxuser vboxuser 491 Jun 13 22:00 lastlog.sh
drwx----- 3 vboxuser vboxuser 4096 Jun  5 21:39 .local
drwxr-xr-x  2 vboxuser vboxuser 4096 Jun  5 21:39 Music
drwxrwxr-x  2 vboxuser vboxuser 4096 Jun 17 20:22 Network
drwxrwxr-x  2 vboxuser vboxuser 4096 Jun 17 20:19 NetworkLab
drwxr-xr-x  3 vboxuser vboxuser 4096 Jun 17 20:15 Pictures
-rw-r--r--  1 vboxuser vboxuser 807 Jun  5 20:54 .profile
drwxr-xr-x  2 vboxuser vboxuser 4096 Jun  5 21:39 Public
-rw-rw-r--  1 vboxuser vboxuser 418 Jun 13 20:40 simple.class
-rw-rw-r--  1 vboxuser vboxuser 102 Jun 13 20:40 simple.java
drwx----- 4 vboxuser vboxuser 4096 Jun 12 22:56 snap
drwx----- 2 vboxuser vboxuser 4096 Jun  6 10:54 .ssh
-rw-r--r--  1 vboxuser vboxuser  0 Jun 13 00:07 .sudo_as_admin_successful
drwxr-xr-x  2 vboxuser vboxuser 4096 Jun  5 21:39 Templates
drwxr-xr-x  2 vboxuser vboxuser 4096 Jun  5 21:39 Videos
vboxuser@Ubuntu:~$ ls -R
Videos Templates snap simple.class simple.class Public Pictures NetworkLab Network Music lastlog.sh Downloads Documents Desktop
vboxuser@Ubuntu:~$ ls -a
.bash_history .bashrc .config Documents .fontconfig lastlog.sh Music NetworkLab .profile simple.class snap .sudo_as_admin_successful Videos
.bash_logout .cache Desktop Downloads .gnupg .local Network Pictures Public simple.java .ssh Templates
vboxuser@Ubuntu:~$ ls -t
Network NetworkLab Desktop Pictures lastlog.sh simple.class simple.java snap Documents Downloads Music Public Templates Videos
vboxuser@Ubuntu:~$ ls -l
total 56
drwxr-xr-x 2 vboxuser vboxuser 4096 Jun 17 20:19 Desktop
drwxr-xr-x 2 vboxuser vboxuser 4096 Jun  5 21:39 Documents
drwxr-xr-x 2 vboxuser vboxuser 4096 Jun  5 21:39 Downloads
-rwxrwxr-x 1 vboxuser vboxuser 491 Jun 13 22:00 lastlog.sh
drwxr-xr-x 2 vboxuser vboxuser 4096 Jun  5 21:39 Music
drwxrwxr-x 2 vboxuser vboxuser 4096 Jun 17 20:22 Network
drwxrwxr-x 2 vboxuser vboxuser 4096 Jun 17 20:19 NetworkLab
drwxr-xr-x 3 vboxuser vboxuser 4096 Jun 17 20:15 Pictures
drwxr-xr-x 2 vboxuser vboxuser 4096 Jun  5 21:39 Public
-rw-rw-r-- 1 vboxuser vboxuser 418 Jun 13 20:40 simple.class
-rw-rw-r-- 1 vboxuser vboxuser 102 Jun 13 20:40 simple.java
drwx----- 4 vboxuser vboxuser 4096 Jun 12 22:56 snap
drwxr-xr-x 2 vboxuser vboxuser 4096 Jun  5 21:39 Templates
drwxr-xr-x 2 vboxuser vboxuser 4096 Jun  5 21:39 Videos
vboxuser@Ubuntu:~$
```

```

vboxuser@Ubuntu:~$ ls -R
.:
Desktop Documents Downloads lastlog.sh Music Network NetworkLab Pictures Public simple.class simple.java snap Templates Videos
./Desktop:
pwd.png
./Documents:
./Downloads:
./Music:
./Network:
./NetworkLab:
./Pictures:
Screenshots
./Pictures/Screenshots:
cd.png ls.png pwd.png
./Public:
./snap:
firefox snapd-desktop-integration
./snap/firefox:
2356 common current
./snap/firefox/2356:
./snap/firefox/common:
./snap/snapd-desktop-integration:
49 83 common current
./snap/snapd-desktop-integration/49:
./snap/snapd-desktop-integration/83:
./snap/snapd-desktop-integration/common:
./Templates:
./Videos:
vboxuser@Ubuntu:~$

```

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.

```

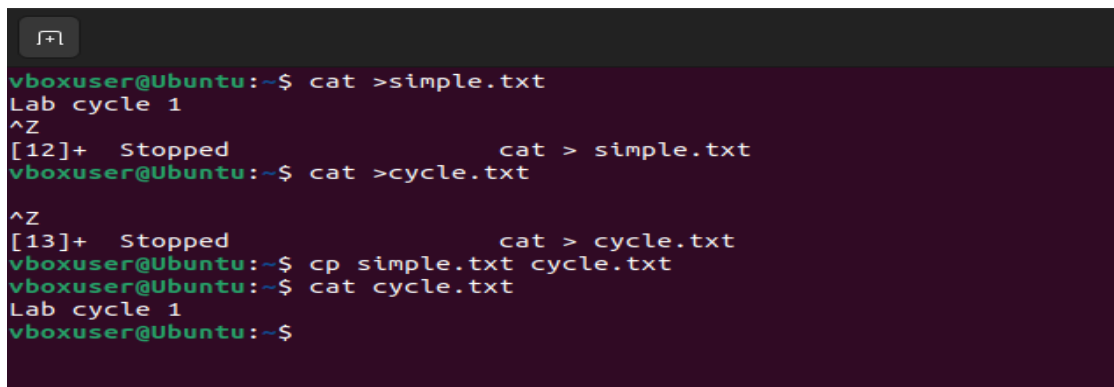
vboxuser@Ubuntu:~$ cat >a.txt
Network Lab
^Z
[10]+  Stopped                  cat > a.txt
vboxuser@Ubuntu:~$ cat >b.txt
linux commands
^Z
[11]+  Stopped                  cat > b.txt
vboxuser@Ubuntu:~$ cat a.txt >c.txt
vboxuser@Ubuntu:~$ cat c.txt
Network Lab
vboxuser@Ubuntu:~$ cat a.txt
Network Lab
vboxuser@Ubuntu:~$ cat b.txt
linux commands
vboxuser@Ubuntu:~$ cat >a.txt b.txt >d.txt
vboxuser@Ubuntu:~$ cat d.txt
linux commands
vboxuser@Ubuntu:~$

```

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.

A terminal window with a dark purple background. The user 'vboxuser@Ubuntu' is in the home directory. They run 'cat > simple.txt' and enter 'Lab cycle 1' followed by a Ctrl-Z (^Z) to stop the cat command. The prompt changes to '[12]+ Stopped cat > simple.txt'. Then they run 'cat > cycle.txt' and enter 'Lab cycle 1' followed by another Ctrl-Z (^Z). The prompt changes to '[13]+ Stopped cat > cycle.txt'. Finally, they run 'cp simple.txt cycle.txt' and 'cat cycle.txt', which outputs 'Lab cycle 1'.

```
vboxuser@Ubuntu:~$ cat >simple.txt
Lab cycle 1
^Z
[12]+  Stopped                  cat > simple.txt
vboxuser@Ubuntu:~$ cat >cycle.txt
^Z
[13]+  Stopped                  cat > cycle.txt
vboxuser@Ubuntu:~$ cp simple.txt cycle.txt
vboxuser@Ubuntu:~$ cat cycle.txt
Lab cycle 1
vboxuser@Ubuntu:~$
```

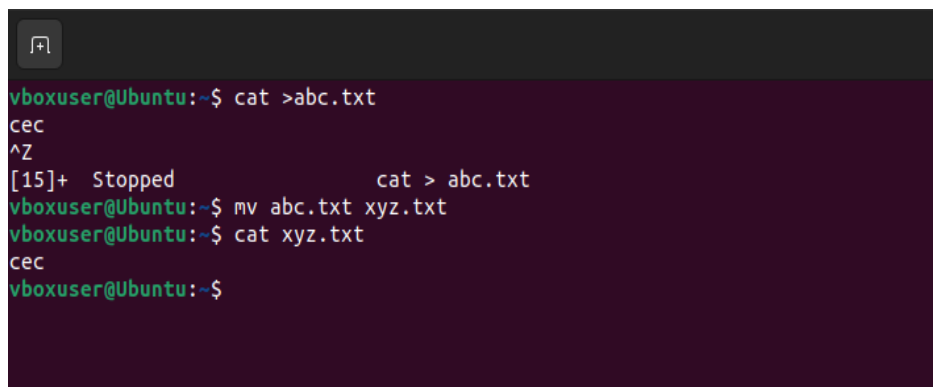
6. mv

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.extnewname.ext`.

A terminal window with a dark purple background. The user 'vboxuser@Ubuntu' is in the home directory. They run 'cat > abc.txt' and enter 'cec' followed by a Ctrl-Z (^Z) to stop the cat command. The prompt changes to '[15]+ Stopped cat > abc.txt'. Then they run 'mv abc.txt xyz.txt' and 'cat xyz.txt', which outputs 'cec'.

```
vboxuser@Ubuntu:~$ cat >abc.txt
cec
^Z
[15]+  Stopped                  cat > abc.txt
vboxuser@Ubuntu:~$ mv abc.txt xyz.txt
vboxuser@Ubuntu:~$ cat xyz.txt
cec
vboxuser@Ubuntu:~$
```

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.

Forexample, mkdir -p Music/2020/Newfile will create the new “2020” file.

```
vboxuser@Ubuntu:~$ mkdir Networking
vboxuser@Ubuntu:~$ cd Networking
vboxuser@Ubuntu:~/Networking$
```

8. rmdir

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

```
vboxuser@Ubuntu:~$ rmdir Networking
vboxuser@Ubuntu:~$
```

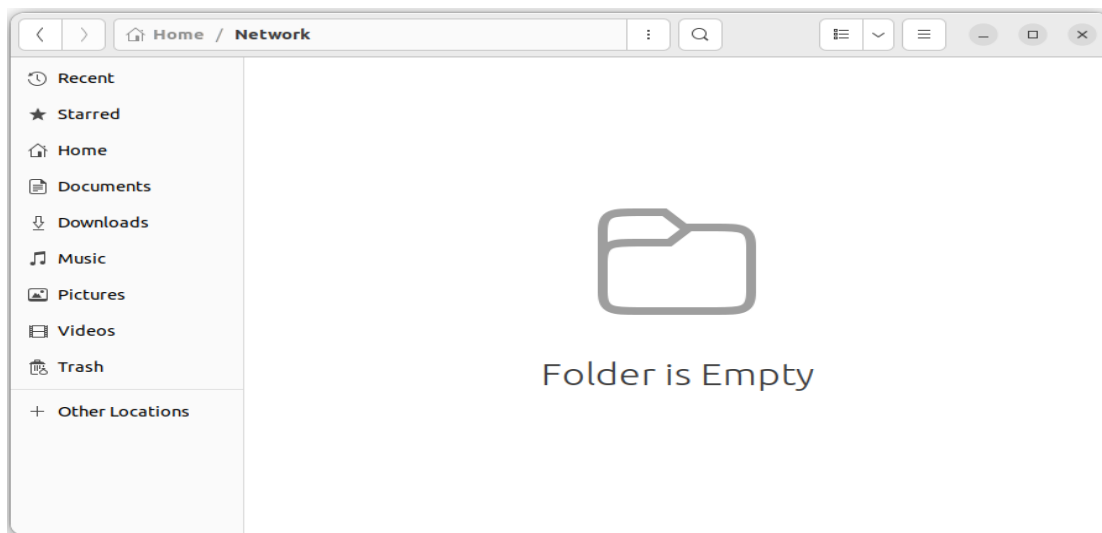
9. rm

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.

```
vboxuser@Ubuntu:~/Network$ rm lab.txt
vboxuser@Ubuntu:~/Network$ cat lab.txt
cat: lab.txt: No such file or directory
vboxuser@Ubuntu:~/Network$
```

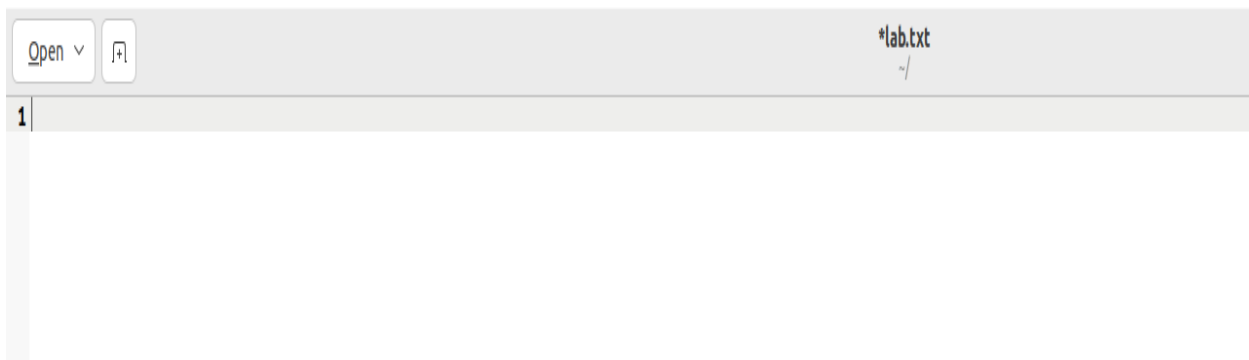


10. touch

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.

```
vboxuser@Ubuntu:~$ touch lab.txt
vboxuser@Ubuntu:~$
```



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.

```
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.

```
vboxuser@Ubuntu:~$ find a.txt
a.txt
vboxuser@Ubuntu:~$
```

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”

```
vboxuser@Ubuntu:~$ cat >mail.txt
arya@gmail.com
anju@gmail.com
lakshmi@gmail.com
megha@gmail.com
sruthy@outlook.com

^Z
[1]+  Stopped                  cat > mail.txt
vboxuser@Ubuntu:~$ grep gmail mail.txt
arya@gmail.com
anju@gmail.com
lakshmi@gmail.com
megha@gmail.com
vboxuser@Ubuntu:~$
```

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.

```
vboxuser@Ubuntu:~$ sudo -h
sudo - execute a command as another user

usage: sudo -h | -K | -k | -V
usage: sudo -v [-AbknS] [-g group] [-h host] [-p prompt] [-u user] [-u user] [command]
usage: sudo -l [-AbknS] [-g group] [-h host] [-p prompt] [-U user] [-u user] [command]
usage: sudo [-AbBEHknPS] [-r role] [-t type] [-C num] [-D directory] [-g group] [-h host] [-p prompt] [-R directory] [-T timeout] [-u user] [VAR=value] [-i|-s] [<command>]
usage: sudo -e [-AbknS] [-r role] [-t type] [-C num] [-D directory] [-g group] [-h host] [-p prompt] [-R directory] [-T 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
-D, --chdir=directory change the working directory before running command
-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)
-i, --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, --chroot=directory change the root directory before running command
-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

vboxuser@Ubuntu:~$
```

15. df

se 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.

```
vboxuser@Ubuntu:~$ df
Filesystem      1K-blocks      Used Available Use% Mounted on
tmpfs           202268         1472    200796   1% /run
/dev/sda3       25106692 13584776   10221232  58% /
tmpfs           1011324          0    1011324   0% /dev/shm
tmpfs           5120           4        5116   1% /run/lock
/dev/sda2       524252         6216    518036   2% /boot/efi
tmpfs           202264         100    202164   1% /run/user/1000
vboxuser@Ubuntu:~$
```



```
vboxuser@Ubuntu:~$ df -m
Filesystem      1M-blocks    Used Available Use% Mounted on
tmpfs            198         2        197    1% /run
/dev/sda3       24519    13267        9982   58% /
tmpfs           988         0        988    0% /dev/shm
tmpfs            5         1         5    1% /run/lock
/dev/sda2       512         7        506    2% /boot/efi
tmpfs           198         1        198    1% /run/user/1000
vboxuser@Ubuntu:~$
```

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.

```
vboxuser@Ubuntu:~$ du
392  ./cache/gstreamer-1.0
36  ./cache/thumbnails/normal
340  ./cache/thumbnails/large
380  ./cache/thumbnails
4  ./cache/libus-table
56  ./cache/fontconfig
8  ./cache/mesa_shader_cache/bc
8  ./cache/mesa_shader_cache/14
8  ./cache/mesa_shader_cache/4f
8  ./cache/mesa_shader_cache/a6
8  ./cache/mesa_shader_cache/8a
8  ./cache/mesa_shader_cache/d5
8  ./cache/mesa_shader_cache/26
12  ./cache/mesa_shader_cache/72
8  ./cache/mesa_shader_cache/2b
16  ./cache/mesa_shader_cache/42
8  ./cache/mesa_shader_cache/57
8  ./cache/mesa_shader_cache/06
8  ./cache/mesa_shader_cache/91
8  ./cache/mesa_shader_cache/b5
8  ./cache/mesa_shader_cache/d3
12  ./cache/mesa_shader_cache/18
8  ./cache/mesa_shader_cache/94
8  ./cache/mesa_shader_cache/20
8  ./cache/mesa_shader_cache/d7
8  ./cache/mesa_shader_cache/0d
12  ./cache/mesa_shader_cache/dd
8  ./cache/mesa_shader_cache/bf
8  ./cache/mesa_shader_cache/8f
8  ./cache/mesa_shader_cache/1f
8  ./cache/mesa_shader_cache/93
8  ./cache/mesa_shader_cache/61
12  ./cache/mesa_shader_cache/3e
8  ./cache/mesa_shader_cache/0b
8  ./cache/mesa_shader_cache/68
8  ./cache/mesa_shader_cache/de
8  ./cache/mesa_shader_cache/cc
12  ./cache/mesa_shader_cache/32
8  ./cache/mesa_shader_cache/e6
8  ./cache/mesa_shader_cache/6c
8  ./cache/mesa_shader_cache/2f
12  ./cache/mesa_shader_cache/b4
12  ./cache/mesa_shader_cache/4e
```

```
vboxuser@Ubuntu:~$ du -b
398325 ./cache/gstreamer-1.0
11895  ./cache/thumbnails/normal
276555 ./cache/thumbnails/large
292546 ./cache/thumbnails
4896   ./cache/libus-table
51632  ./cache/fontconfig
6682   ./cache/mesa_shader_cache/bc
5700   ./cache/mesa_shader_cache/14
7254   ./cache/mesa_shader_cache/4f
6879   ./cache/mesa_shader_cache/a6
7674   ./cache/mesa_shader_cache/8a
6982   ./cache/mesa_shader_cache/d5
7691   ./cache/mesa_shader_cache/26
7343   ./cache/mesa_shader_cache/72
6319   ./cache/mesa_shader_cache/2b
12917  ./cache/mesa_shader_cache/42
6768   ./cache/mesa_shader_cache/57
6607   ./cache/mesa_shader_cache/06
6837   ./cache/mesa_shader_cache/91
6696   ./cache/mesa_shader_cache/b5
7195   ./cache/mesa_shader_cache/d3
8752   ./cache/mesa_shader_cache/18
6561   ./cache/mesa_shader_cache/94
6605   ./cache/mesa_shader_cache/20
7409   ./cache/mesa_shader_cache/d7
7103   ./cache/mesa_shader_cache/0d
9464   ./cache/mesa_shader_cache/dd
7180   ./cache/mesa_shader_cache/bf
7355   ./cache/mesa_shader_cache/8f
6964   ./cache/mesa_shader_cache/1f
6604   ./cache/mesa_shader_cache/93
7782   ./cache/mesa_shader_cache/61
9419   ./cache/mesa_shader_cache/3e
6031   ./cache/mesa_shader_cache/0b
7589   ./cache/mesa_shader_cache/68
6099   ./cache/mesa_shader_cache/de
6260   ./cache/mesa_shader_cache/cc
8214   ./cache/mesa_shader_cache/32
6224   ./cache/mesa_shader_cache/e6
6604   ./cache/mesa_shader_cache/6c
6224   ./cache/mesa_shader_cache/2f
7642   ./cache/mesa_shader_cache/b4
10230  ./cache/mesa_shader_cache/4e
7468   ./cache/mesa_shader_cache/f6
```

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)

```
vboxuser@Ubuntu:~$ cat mail.txt
arya@gmail.com
anju@gmail.com
lakshmi@gmail.com
megha@gmail.com
sruthy@outlook.com

vboxuser@Ubuntu:~$ head -n 1 mail.txt
arya@gmail.com
vboxuser@Ubuntu:~$
```

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`.

```
vboxuser@Ubuntu:~$ cat mail.txt
arya@gmail.com
anju@gmail.com
megha@gmail.com
lakshmi@gmail.com
sruthy@outlook.com

vboxuser@Ubuntu:~$ tail -n 2 mail.txt
lakshmi@gmail.com
sruthy@outlook.com
vboxuser@Ubuntu:~$
```

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`

```
vboxuser@Ubuntu:~$ cat simple.txt
Lab cycle 1
vboxuser@Ubuntu:~$ cat mail.txt
arya@gmail.com
anju@gmail.com
megha@gmail.com
lakshmi@gmail.com
sruthy@outlook.com
vboxuser@Ubuntu:~$ diff simple.txt mail.txt
1c1,5
< Lab cycle 1
---
> arya@gmail.com
> anju@gmail.com
> megha@gmail.com
> lakshmi@gmail.com
> sruthy@outlook.com
vboxuser@Ubuntu:~$
```

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.

```
vboxuser@Ubuntu:~$ tar cvf file.tar *.txt
a.txt
b.txt
cycle.txt
d.txt
file5.txt
lab.txt
mail.txt
simple.txt
xyz.txt
vboxuser@Ubuntu:~$ tar xvf file.tar
a.txt
b.txt
cycle.txt
d.txt
file5.txt
lab.txt
mail.txt
simple.txt
xyz.txt
vboxuser@Ubuntu:~$
```

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 .

```
vboxuser@Ubuntu:~$ cat a.txt
Network Lab
vboxuser@Ubuntu:~$ chmod -r a.txt
vboxuser@Ubuntu:~$ cat a.txt
cat: a.txt: Permission denied
vboxuser@Ubuntu:~$ chmod +r a.txt
vboxuser@Ubuntu:~$ cat a.txt
Network Lab
vboxuser@Ubuntu:~$
```

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.

```
user@user-HP-Laptop-15-da0xxx:~$ cat >cn.txt
computer
networks
programming
lab
hello
world
^Z
[1]+  Stopped                  cat > cn.txt
user@user-HP-Laptop-15-da0xxx:~$ cat cn.txt
computer
networks
programming
lab
hello
world
user@user-HP-Laptop-15-da0xxx:~$ ls
a.txt      cn.txt      Desktop    file2.txt  hello.txt  '#newfile.txt#'  primes.c      Public
bitstring.c  computer  Documents  file7.txt  lab.txt    new.txt          programming   Templates
cn          cpgns     file1.txt  GIT        Music      Pictures          programming.txt  Videos
user@user-HP-Laptop-15-da0xxx:~$ ls -l cn.txt
-rw-rw-r-- 1 user user 46 Jun 13 17:45 cn.txt
user@user-HP-Laptop-15-da0xxx:~$ chown gowri cn.txt
chown: changing ownership of 'cn.txt': Operation not permitted
user@user-HP-Laptop-15-da0xxx:~$
```

23. ps

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

```
vboxuser@Ubuntu:~$ ps
  PID TTY          TIME CMD
  2358 pts/0    00:00:00 bash
  11426 pts/0    00:00:00 cat
  20953 pts/0    00:00:00 cat
  21314 pts/0    00:00:00 cat
  21652 pts/0    00:00:00 cat
  22525 pts/0    00:00:00 cat
  22800 pts/0    00:00:00 cat
  32699 pts/0    00:00:00 ps
vboxuser@Ubuntu:~$ ps -h
  PID TTY          Ss+  TIME COMMAND
  1450 tty2      Ss+  0:00 /usr/libexec/gdm-wayland-session env GNOME_SHELL_SESSION_MODE=ubuntu /usr/bin/gnome-session --session=ubuntu
  1457 tty2      Sl+  0:00 /usr/libexec/gnome-session-binary --session=ubuntu
  2358 pts/0      Ss   0:00 bash
  11426 pts/0      T    0:00 cat
  20953 pts/0      T    0:00 cat
  21314 pts/0      T    0:00 cat
  21652 pts/0      T    0:00 cat
  22525 pts/0      T    0:00 cat
  22800 pts/0      T    0:00 cat
  32806 pts/0      R+   0:00 ps -h
vboxuser@Ubuntu:~$
```

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.

```
vboxuser@Ubuntu:~$ ping amazon.in
PING amazon.in (104.18.15.146) 56(84) bytes of data:
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=1 ttl=53 time=79.5 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=2 ttl=53 time=391 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=3 ttl=53 time=58.1 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=4 ttl=53 time=123 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=5 ttl=53 time=89.2 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=6 ttl=53 time=124 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=7 ttl=53 time=1294 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=8 ttl=53 time=3152 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=9 ttl=53 time=1203 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=10 ttl=53 time=2723 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=11 ttl=53 time=3448 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=12 ttl=53 time=3689 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=13 ttl=53 time=2317 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=14 ttl=53 time=1600 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=15 ttl=53 time=928 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=16 ttl=53 time=392 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=17 ttl=53 time=106 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=18 ttl=53 time=82.1 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=19 ttl=53 time=453 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=20 ttl=53 time=453 ms
From 192.168.43.160 icmp_seq=28 Destination Host Unreachable
From 192.168.43.160 (192.168.43.160) icmp_seq=41 Destination Host Unreachable
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=42 ttl=53 time=383 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=43 ttl=53 time=272 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=44 ttl=53 time=391 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=45 ttl=53 time=189 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=46 ttl=53 time=89.5 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=47 ttl=53 time=87.7 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=48 ttl=53 time=85.5 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=49 ttl=53 time=84.3 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=50 ttl=53 time=123 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=51 ttl=53 time=167 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=52 ttl=53 time=928 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=53 ttl=53 time=210 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=54 ttl=53 time=222 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=55 ttl=53 time=1237 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=56 ttl=53 time=217 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=57 ttl=53 time=277 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=58 ttl=53 time=124 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=59 ttl=53 time=63.0 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=60 ttl=53 time=118 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=61 ttl=53 time=215 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=62 ttl=53 time=125 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=63 ttl=53 time=360 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=64 ttl=53 time=256 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=65 ttl=53 time=367 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=66 ttl=53 time=158 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=67 ttl=53 time=124 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=68 ttl=53 time=225 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=69 ttl=53 time=270 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=70 ttl=53 time=59.4 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=71 ttl=53 time=155 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=72 ttl=53 time=298 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=73 ttl=53 time=129 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=74 ttl=53 time=72.6 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=75 ttl=53 time=72.6 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=76 ttl=53 time=72.6 ms
64 bytes from 104.18.15.146 (104.18.15.146): icmp_seq=77 ttl=53 time=72.6 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.

```
vboxuser@Ubuntu:~$ wget https://www.amazon.in/index.html
--2023-06-18 14:29:11-- https://www.amazon.in/index.html
Resolving www.amazon.in (www.amazon.in)... 75.2.51.62, 99.83.179.101
Connecting to www.amazon.in (www.amazon.in)|75.2.51.62|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://www.amazon.fr/443/ [following]
--2023-06-18 14:29:14-- https://www.amazon.fr/
Resolving www.amazon.fr (www.amazon.fr)... 18.161.243.163, 2600:9000:257a:800:6:f4ed:9992:2361, 2600:9000:257a:200:6:f4ed:9992:2361, ...
Connecting to www.amazon.fr (www.amazon.fr)|18.161.243.163|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'index.html.2'

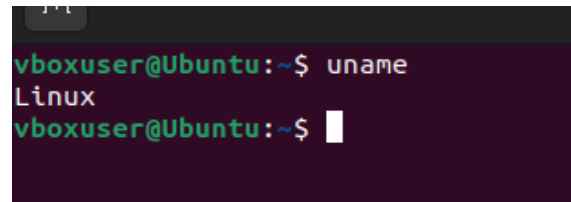
index.html.2          [          <=>          ] 351.17K  63.6KB/s   in 5.5s

2023-06-18 14:29:22 (63.6 KB/s) - 'index.html.2' saved [359599]

vboxuser@Ubuntu:~$
```

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.



```
vboxuser@Ubuntu:~$ uname
Linux
vboxuser@Ubuntu:~$
```

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.

```
top - 14:33:15 up 54 min, 1 user, load average: 0.32, 0.44, 0.40
Tasks: 197 total, 1 running, 187 sleeping, 9 stopped, 0 zombie
%Cpu(s): 3.4 us, 1.4 sy, 0.0 ni, 94.7 id, 0.3 wa, 0.0 hi, 0.2 st, 0.0 sr
MiB Mem : 1975.2 total, 132.6 free, 1181.5 used, 661.2 buff/cache
MiB Swap: 2680.0 total, 2661.6 free, 18.4 used, 613.5 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1548	vboxuser	20	0	4422268	451444	146404	S	4.7	22.3	8:32.85	gnome-shell
954	mysql	20	0	1783600	391492	33660	S	1.7	19.4	0:44.71	mysqld
2201	vboxuser	20	0	568696	58284	45624	S	0.7	2.9	0:32.64	gnome-terminal-
277	root	-51	0	0	0	0	S	0.3	0.0	0:01.02	irq/18-vmwgfx
590	systemd+	20	0	14828	6304	5512	S	0.3	0.3	0:06.05	systemd-oond
639	root	20	0	2812	1136	1048	S	0.3	0.1	0:00.47	acpid
1394	vboxuser	20	0	17940	10808	8212	S	0.3	0.5	0:09.22	systemd
41215	vboxuser	20	0	21872	4264	3404	R	0.3	0.2	0:00.13	top
1	root	20	0	240448	11992	8368	S	0.0	0.6	0:03.71	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.01	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
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	slub_flushwq
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns
8	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H-events_highpri
10	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq
11	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_kthread
12	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_rude_kthread
13	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_trace_kthread
14	root	20	0	0	0	0	S	0.0	0.0	0:00.80	ksoftirqd/0
15	root	20	0	0	0	0	I	0.0	0.0	0:03.37	rcu_preempt
16	root	rt	0	0	0	0	S	0.0	0.0	0:00.04	migration/0
17	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_inject/0
19	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
20	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/1
21	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_inject/1
22	root	rt	0	0	0	0	S	0.0	0.0	0:00.24	migration/1
23	root	20	0	0	0	0	S	0.0	0.0	0:01.03	ksoftirqd/1
25	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/1:0H-events_highpri
26	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdevtmpfs
27	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	inet_frag_wq
28	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kauditd
29	root	20	0	0	0	0	S	0.0	0.0	0:00.00	khungtaskd
31	root	20	0	0	0	0	S	0.0	0.0	0:00.00	oom_reaper
33	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	writeback
34	root	20	0	0	0	0	S	0.0	0.0	0:00.24	kcompactd0
35	root	25	5	0	0	0	S	0.0	0.0	0:00.00	ksmd
36	root	39	19	0	0	0	S	0.0	0.0	0:00.00	khugepaged
37	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kintegrityd
38	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kblockd
39	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	blkcg_punt_bio
41	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	tpm_dev_wq
42	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	ata_sff
43	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	md
44	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	edac-poller
45	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	devfreq_wq
46	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	watchdogd
47	root	0	-20	0	0	0	I	0.0	0.0	0:01.12	kworker/1:1H-kblockd

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.

```
vboxuser@Ubuntu:~$ history
 1  sudo apt update
 2  sudo apt install mysql-server
 3  sudo
 4  sudo apt install mysql-server
 5  sudo apt install default-jdk
 6  clear
 7  who
 8  vi/etc/sudoers
 9  sudo vi /etc/sudoers
10  su -
11  su
12  restart
13  reboot
14  sudo apt install mysql-server
15  sudo systemctl status mysql
16  sudo mysql_secure_installation
17  sudo mysql
18  clear
19  sudo apt install default-jdk
20  clear
21  sudo apt install default-jdk
22  gedit simple.java
23  javac simple.java
24  gedit simple.java
25  javac simple.java
26  gedit simple.java
27  javac simple.java
28  gedit simple.java
29  javac simple.java
30  java simple
31  sudo mysql
32  pwd
33  clear
34  cd
35  clear
36  mkdir NetworkLab
37  pwd
38  cd NetworkLab
39  clear
```

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 to start learning about man utility.

```
vboxuser@Ubuntu:~$ man
What manual page do you want?
For example, try 'man man'.
vboxuser@Ubuntu:~$ man pwd
```



```

PWD(1) User Commands PWD(1)
NAME
    pwd - print name of current/working directory
SYNOPSIS
    pwd [OPTION]...
DESCRIPTION
    Print the full filename of the current working directory.

    -L, --logical
        use PWD from environment, even if it contains symlinks

    -P, --physical
        avoid all symlinks

    --help
        display this help and exit

    --version
        output version information and exit

    If no option is specified, -P is assumed.

    NOTE: your shell may have its own version of pwd, which usually supersedes the version described here. Please refer to your shell's documentation for details about the options it supports.
AUTHOR
    Written by Jim Meyering.
REPORTING BUGS
    GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
    Report any translation bugs to <https://translationproject.org/team/>
COPYRIGHT
    Copyright © 2020 Free Software Foundation, Inc. License GPLv3+: GNU GPL version 3 or later <https://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.
SEE ALSO
    getcwd(3)

    Full documentation <https://www.gnu.org/software/coreutils/pwd>
    or available locally via: info '(coreutils) pwd invocation'
GNU coreutils 8.32 February 2022 PWD(1)
Manual page pwd(1) line 1/44 (END) (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

```

vboxuser@Ubuntu:~$ echo hello Linux
hello Linux
vboxuser@Ubuntu:~$

```

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)

```

vboxuser@Ubuntu:~$ 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 mmddyyyy] [-n suffixes] [zipfile list] [-xi 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
-o store only                       -l convert LF to CR LF (-ll CR LF to LF)
-i 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
vboxuser@Ubuntu:~$

```



```
vboxuser@Ubuntu:~$ 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
vboxuser@Ubuntu:~$
```

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.

```
vboxuser@Ubuntu:~$ hostname
Ubuntu
vboxuser@Ubuntu:~$ hostname -I
10.0.2.15
vboxuser@Ubuntu:~$
```

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

```
vboxuser@Ubuntu:~$ 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
--extrausers          Use the extra users database

vboxuser@Ubuntu:~$
```

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.

```
vboxuser@Ubuntu:~$ expr --version
expr (GNU coreutils) 8.32
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://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.
vboxuser@Ubuntu:~$
```

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.

```
vboxuser@Ubuntu:~$ cat marks.txt
arya-88
megha-92
anju-90
lakshmi-94
vboxuser@Ubuntu:~$ cut -d- -f2 marks.txt
88
92
90
94
vboxuser@Ubuntu:~$ cut -d- -f1 marks.txt
arya
megha
anju
lakshmi
vboxuser@Ubuntu:~$
```

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.

```
vboxuser@Ubuntu:~$ cat states
kerala
karnataka
jharkhand
Madhya pradesh
vboxuser@Ubuntu:~$ cat capital
Thiruvananthapuram
Bangaluru
Ranchi
Bhopal
vboxuser@Ubuntu:~$ paste states capital
kerala Thiruvananthapuram
karnataka Bangaluru
jharkhand Ranchi
Madhya pradesh Bhopal
vboxuser@Ubuntu:~$
```

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.

```
vboxuser@Ubuntu:~$ ssh
usage: ssh [-46ACFGgKKkMNnqsTtVvXxYy] [-B bind_interface]
          [-b bind_address] [-c cipher_spec] [-D [bind_address:]port]
          [-E log_file] [-e escape_char] [-F configfile] [-I pkcs11]
          [-i identity_file] [-J [user@]host[:port]] [-L address]
          [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]
          [-Q query_option] [-R address] [-S ctl_path] [-W host:port]
          [-w local_tun[:remote_tun]] destination [command [argument ...]]

vboxuser@Ubuntu:~$ scp
usage: scp [-346ABCOPpqRrsTv] [-c cipher] [-D sftp_server_path] [-F ssh_config]
          [-i identity_file] [-J destination] [-l limit]
          [-o ssh_option] [-P port] [-S program] source ... target

vboxuser@Ubuntu:~$
```

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.

```
vboxuser@Ubuntu:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/vboxuser/.ssh/id_rsa): vbox
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in vbox
Your public key has been saved in vbox.pub
The key fingerprint is:
SHA256: tH9RJnn6vRZiVjkkcD87mDi4rKzgv/gdhcxMSHfESE vboxuser@Ubuntu
The key's randomart image is:
+--[RSA 3072]-----+
|      .            |
|     . . . 0 .     |
|    . . . .B 0     |
|   . . . . = .     |
|  + S . . +       |
|   . . . = .0+ .   |
|  . . . 0 . 0+ + 0 |
| ..000. . . + 0    |
|  . . * . . . 0 0   |
|  . . . [SHA256]-----+
vboxuser@Ubuntu:~$ ssh-copy-id
Usage: /usr/bin/ssh-copy-id [-h-?] [-f] [-n] [-s] [-i [identity_file]] [-p port] [-F alternative_ssh_config_file] [[-o <ssh -o options>] ...] [user@]hostname
-f: force mode -- copy keys without trying to check if they are already installed
-n: dry run -- no keys are actually copied
-s: use sftp -- use sftp instead of executing remote-commands. Can be useful if the remote only allows sftp
-h|-?: print this help
vboxuser@Ubuntu:~$
```

Result: Familiarized with Linux commands.

