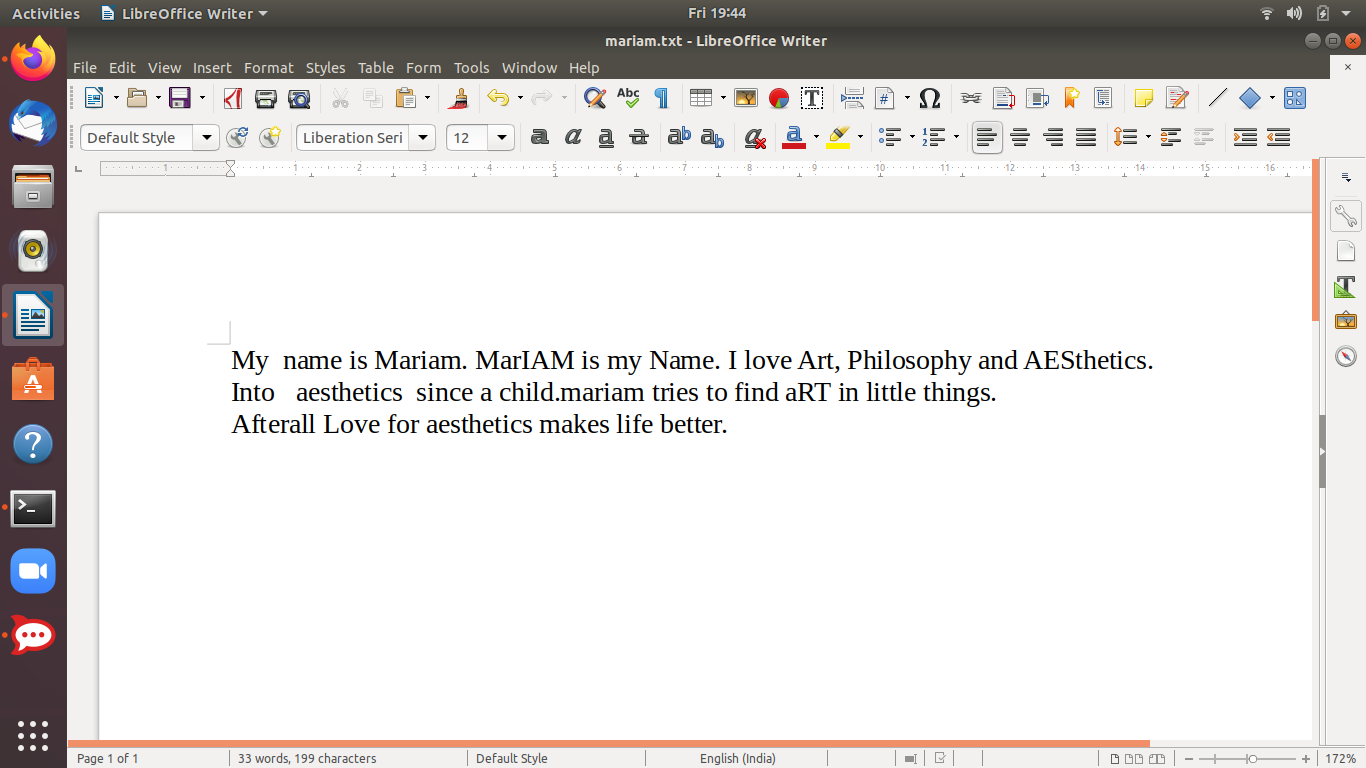
**Grep Command:**

The grep filter searches a file for a particular pattern of characters, and displays all lines that contain that pattern. The pattern that is searched in the file is referred to as the regular expression (grep stands for globally search for regular expression and print out).

Syntax: $ grep [options] pattern [files]

Eg; $ grep -i “Mariam” mariam.txt *(for case insensitive string matching)*

$ grep -c “aesthetics” mariam.txt *(for counting the no. of times a pattern exists in the file so on and so forth...)*

Example:I will take an example of a text file mariam.txt that is clear in the image below.

**Options Description**

**-c** : This prints only a count of the lines that match a pattern

**-h :** Display the matched lines, but do not display the filenames.

**-i :** Ignores case for matching,case insensitive.

**-l :** Displays list of a filenames only.

**-n :** Display the matched lines and their line numbers.

**-v :** This prints out all the lines that do not matches the pattern

**-e exp :** Specifies expression with this option. Can use multiple times.

**-f file :** Takes patterns from file, one per line.

**-E :** Treats pattern as an extended regular expression (ERE)

**-w :** Match whole word

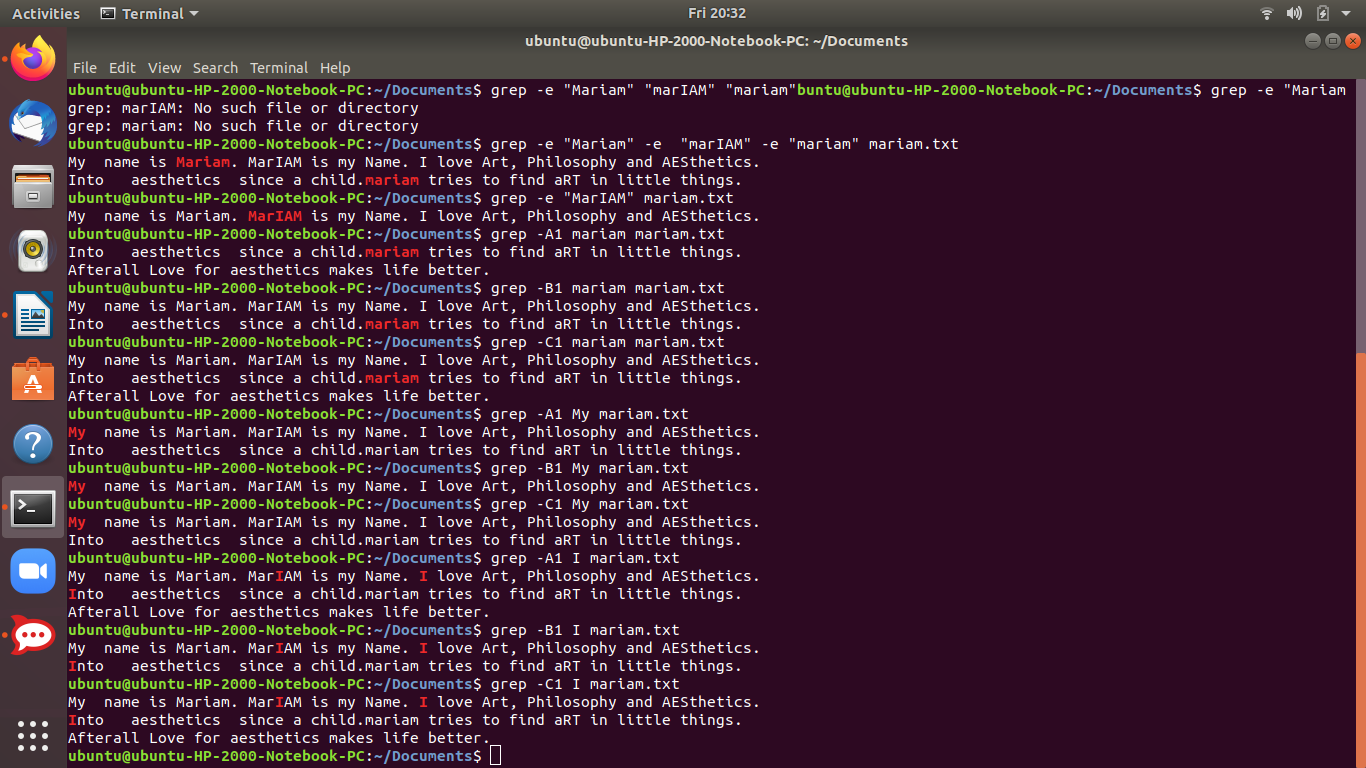
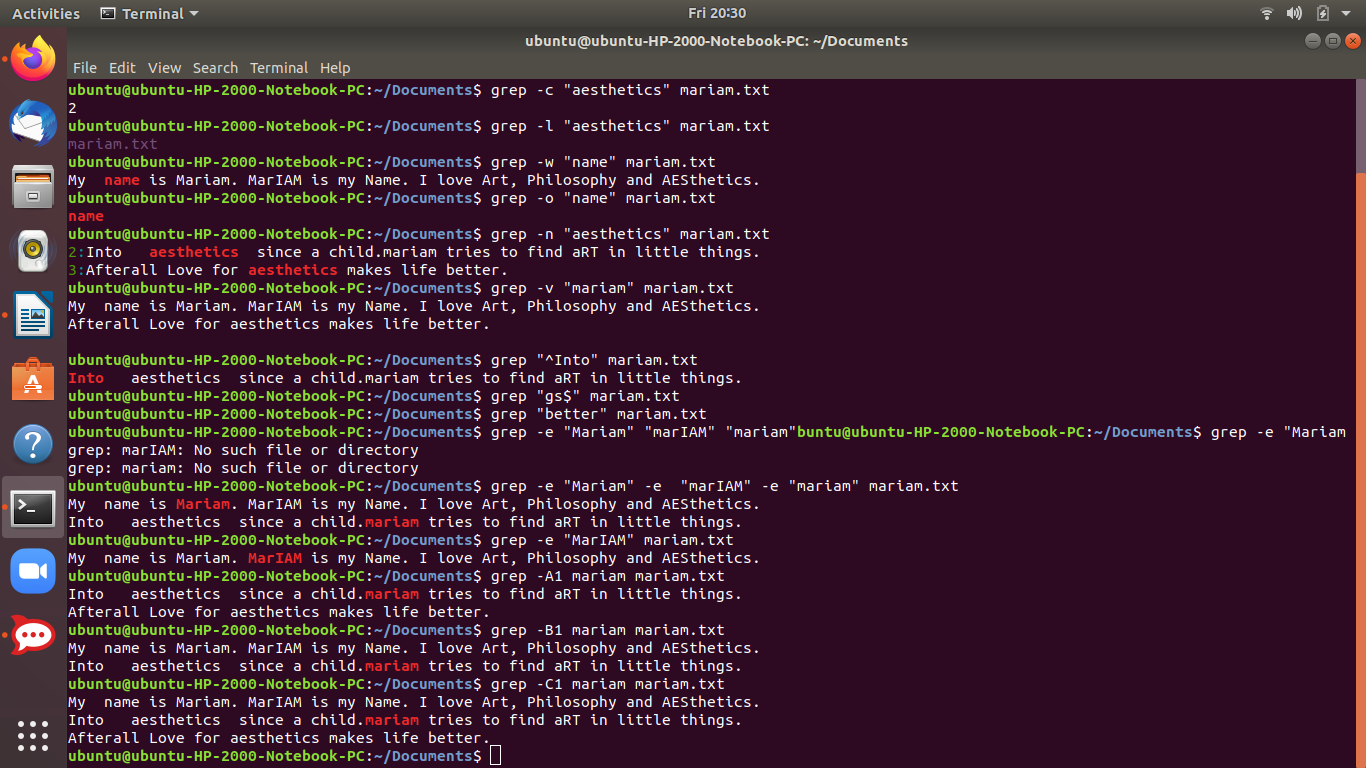
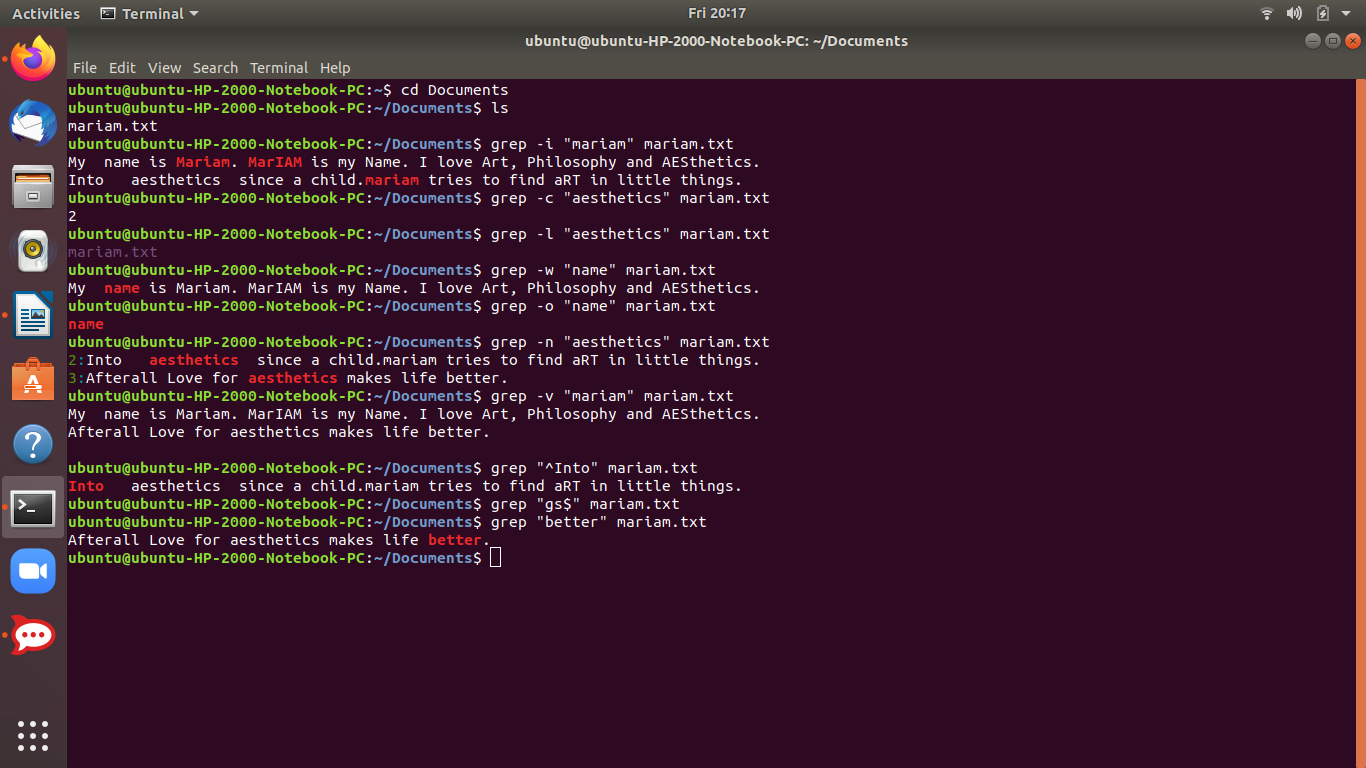
**-o :** Print only the matched parts of a matching line,

with each such part on a separate output line.

**-A n** **:** Prints searched line and nlines after the result.

**-B n :** Prints searched line and n line before the result.

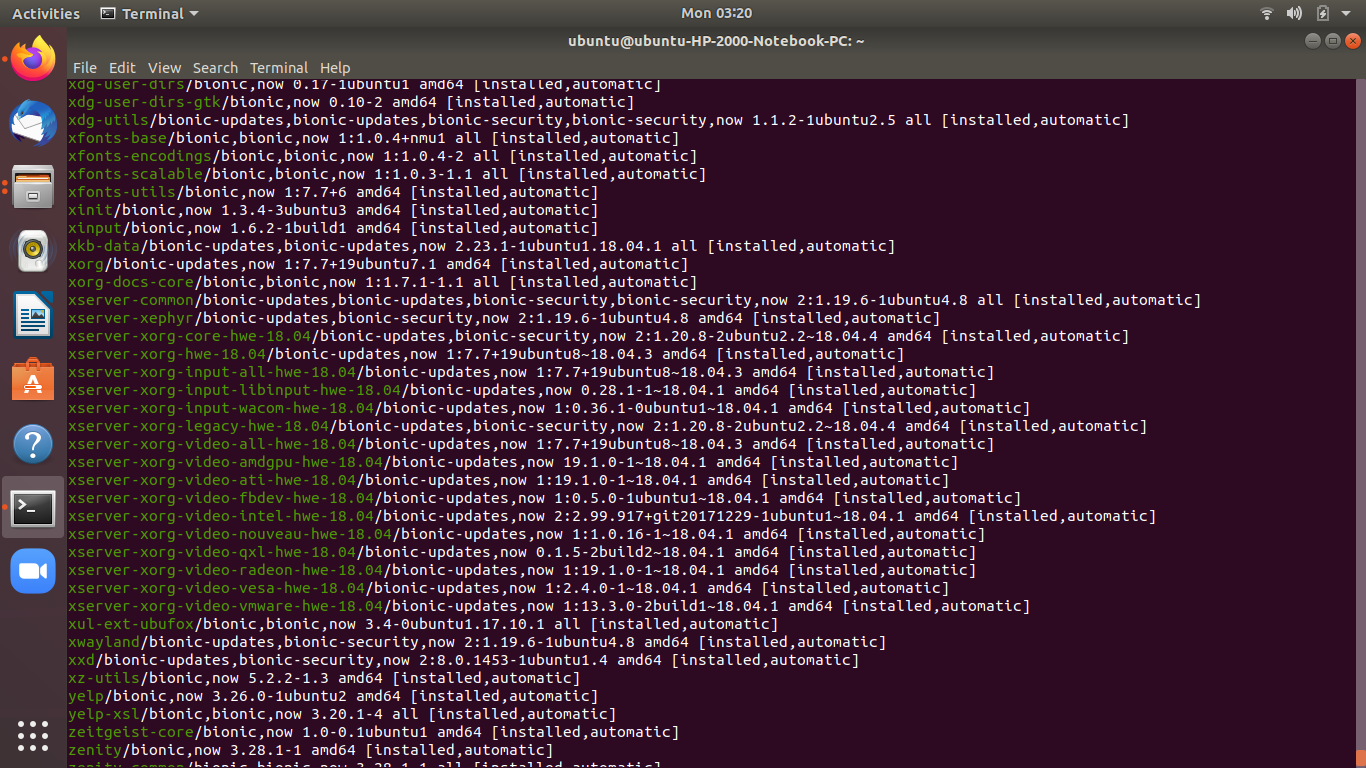
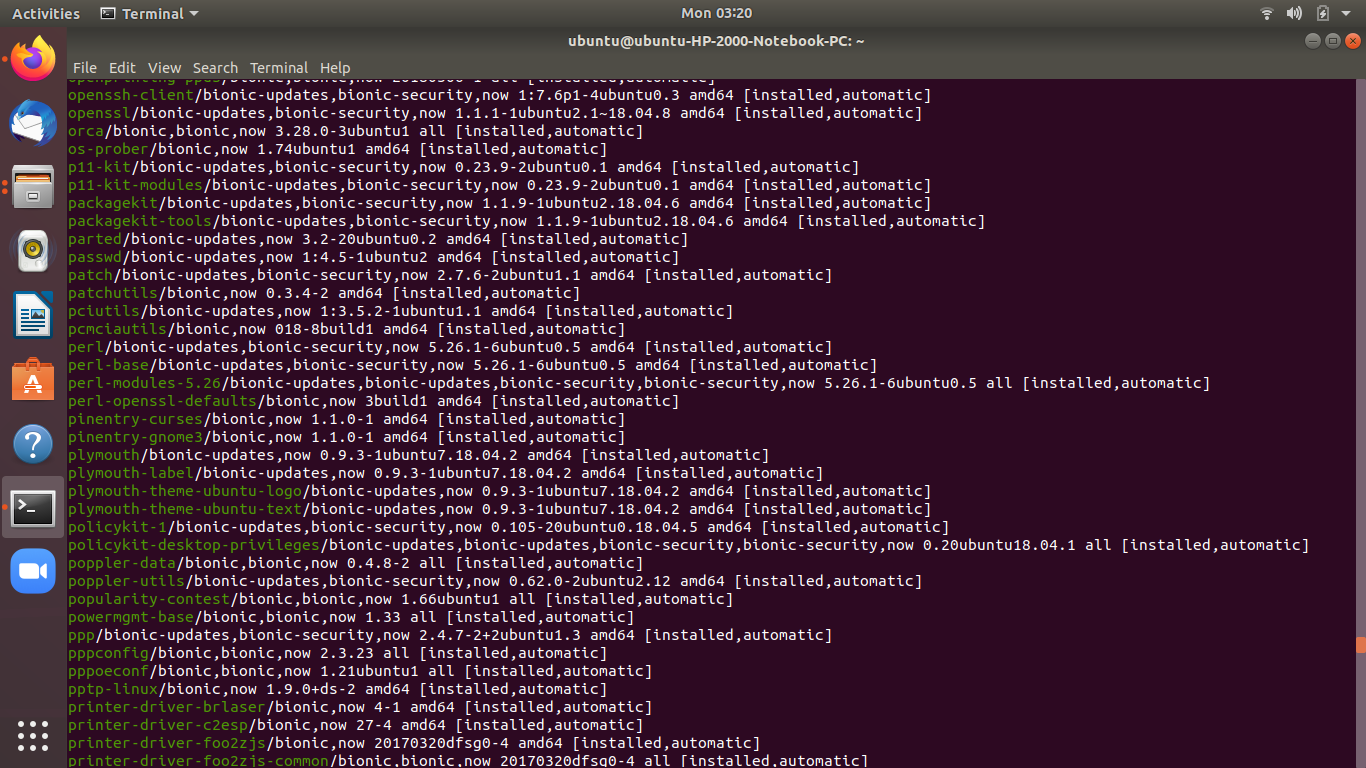
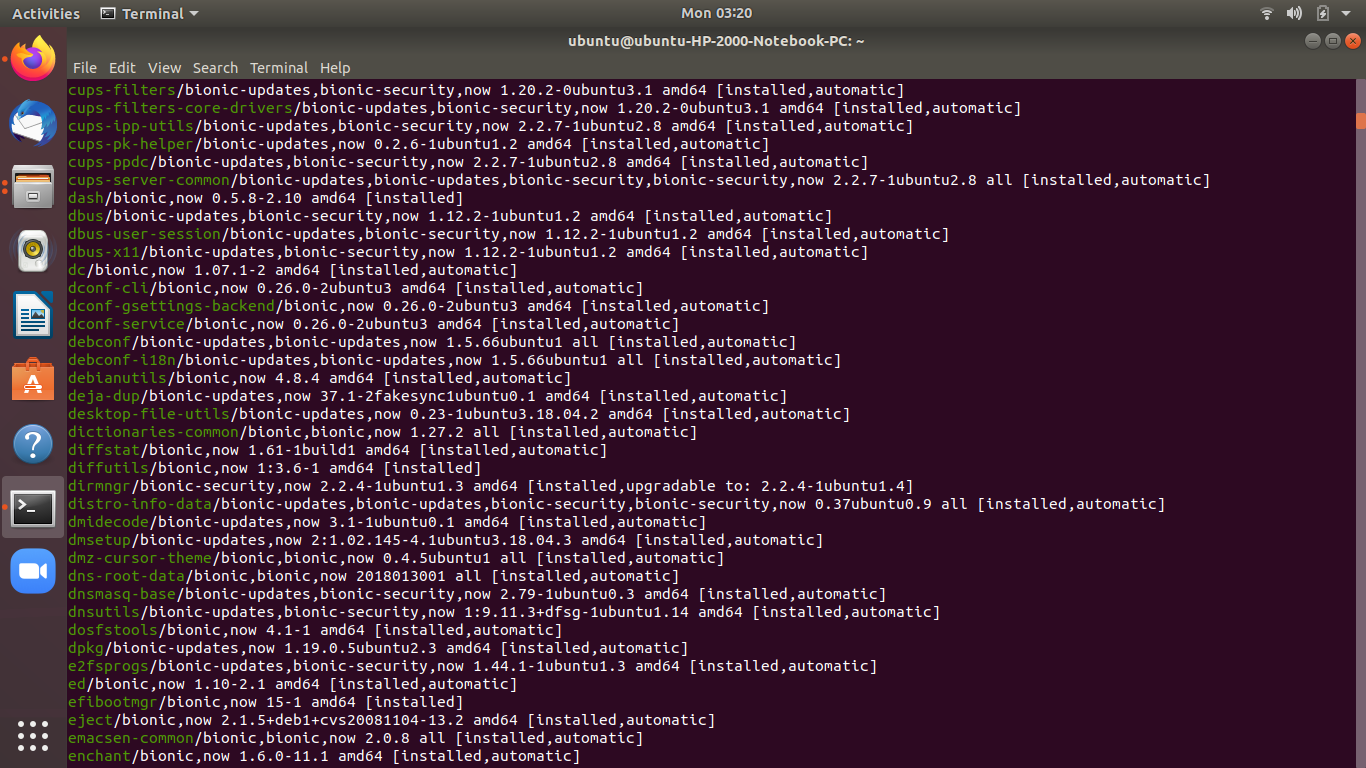
**-C n :** Prints searched line and n lines after before the result.

All these commands will be very clear from the below screenshots:

**How to check list of packages installed on your system:**

The procedure to list what packages are installed on Ubuntu:

1. Open the terminal application.
2. Run command apt list --installed to list all installed packages on Ubuntu
3. To display a list of packages satisfying certain criteria such as show matching apache2 packages, run apt list apache.



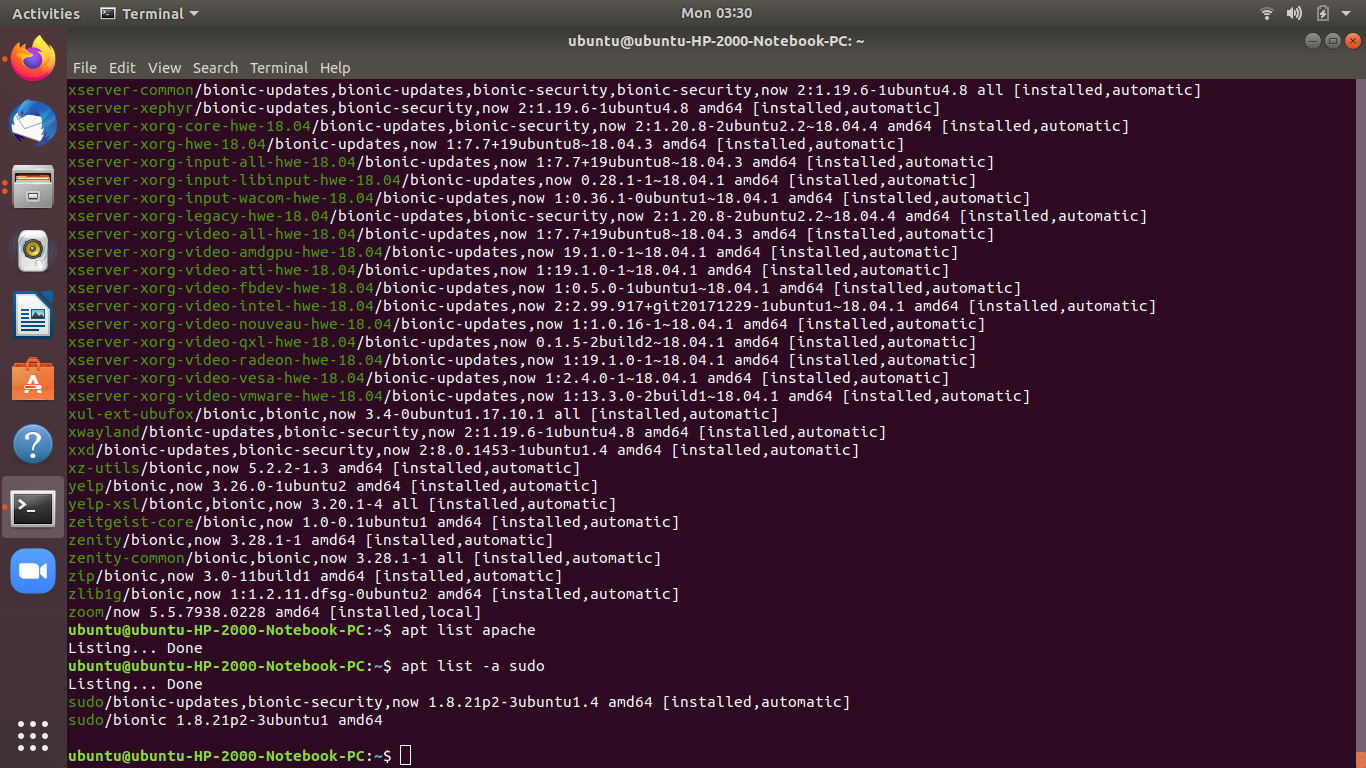
**How to check if a certain package is installed on our machine :**

Run package policy as follows:

$ apt list -a pkgNameHere

For eg if we want to see that sudo package is installed or not

$ apt list -a sudo

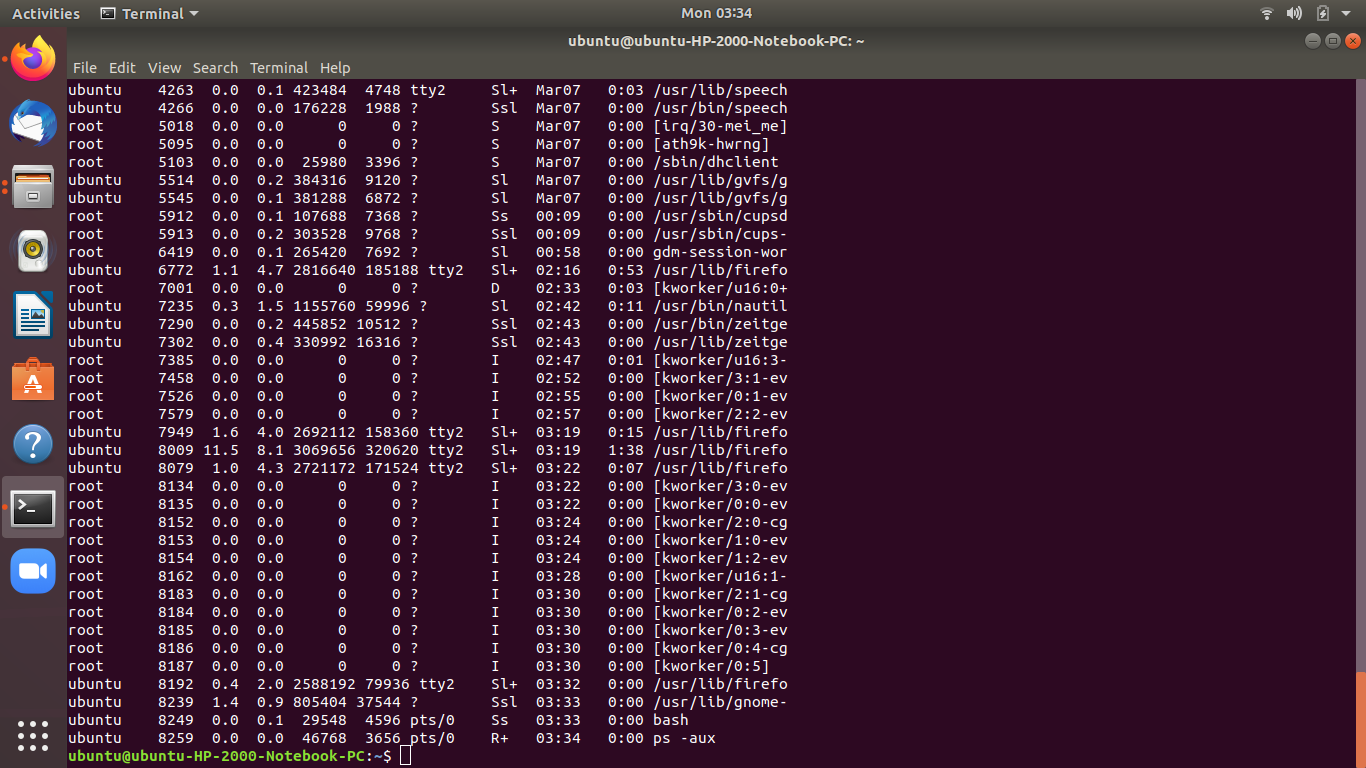
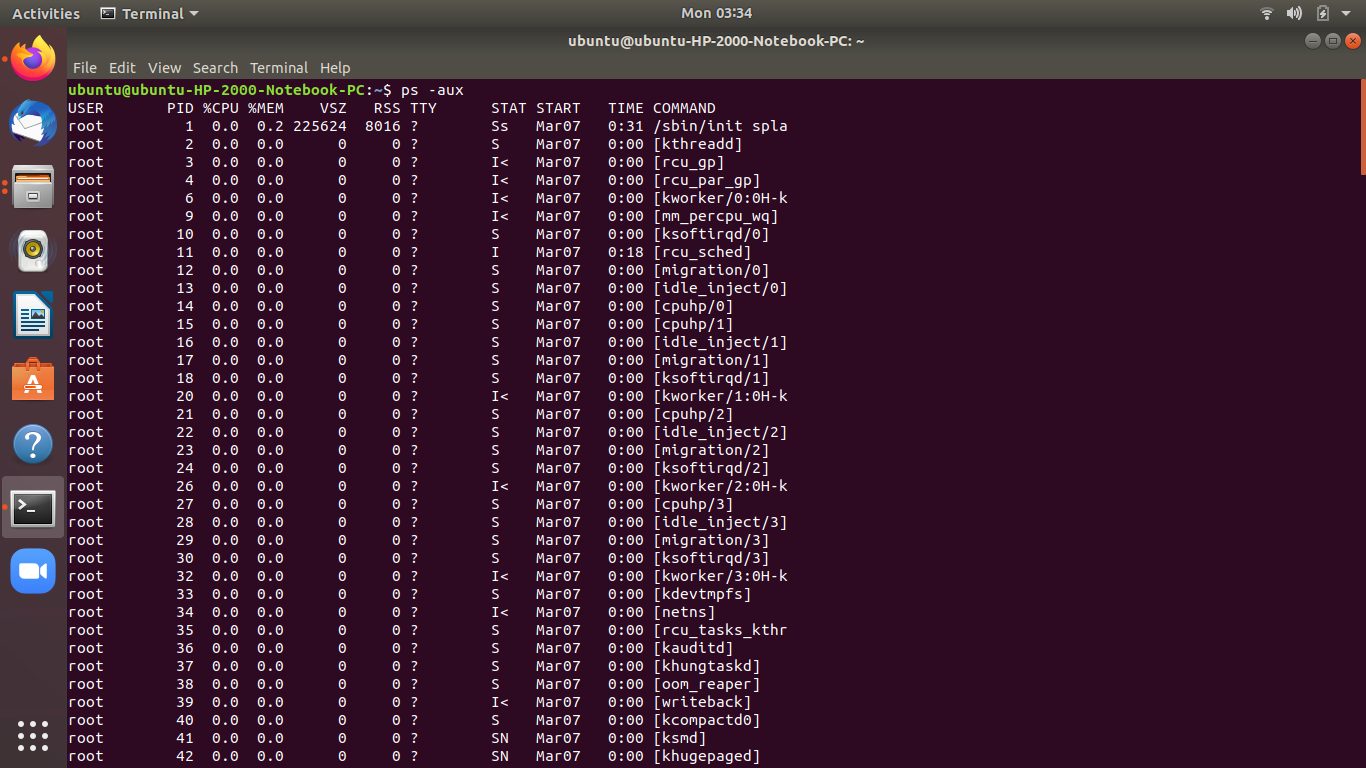


## 

## **Finding running processes in Linux**

## The procedure to monitor the running process in Linux using the command line is as follows:

1. Open the terminal window on Linux
2. Type the ps aux command to see all running process in Linux
3. Alternatively, you can issue the top command or htop command to view running process in Linux



**Combining Commands**

If you use Linux, you know how useful the command line can be for working with files, installing software, and launching programs. But it can be even more efficient if you run multiple commands at once.

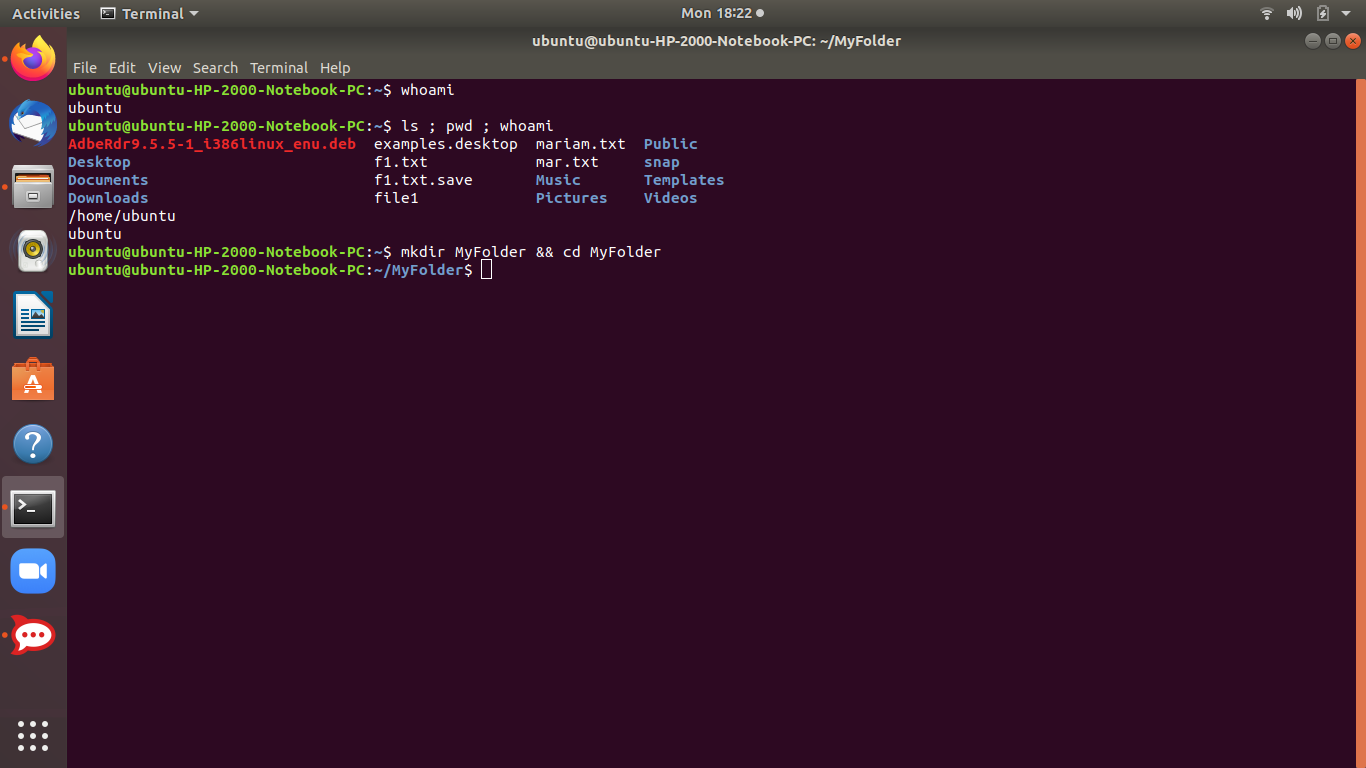
Combining two or more commands on the command line is also known as “command chaining”. We’ll show you different ways you can combine commands on the command line.

## **Option One: The Semicolon (;) Operator**

**The semicolon (;)** operator allows you to execute multiple commands in succession, regardless of whether each previous command succeeds. For example, open a Terminal window .Then, type the following three

commands on one line, separated by semicolons, and press Enter. This will give you a listing of the current directory ( ls ), find out which directory you’re currently in ( pwd ), and display your login name ( whoami ) all at once.

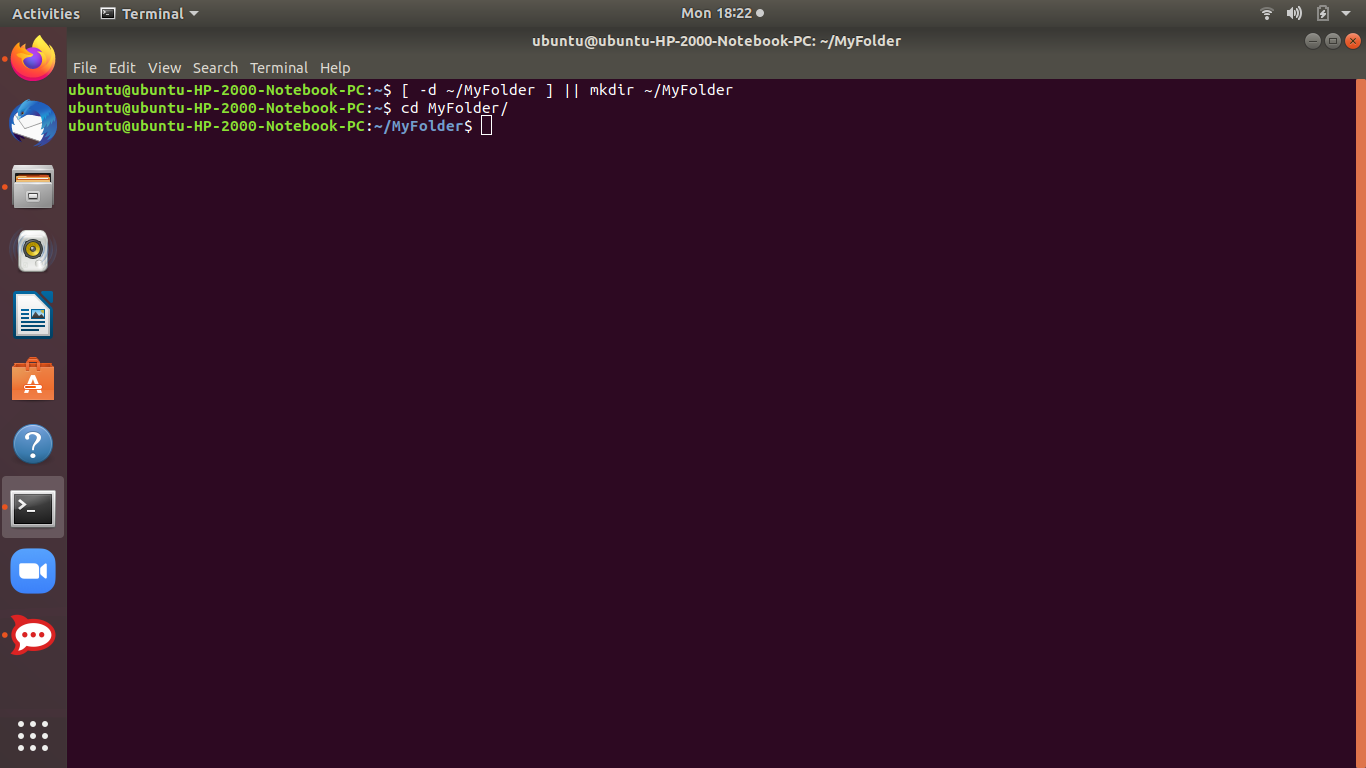
ls ; pwd ; whoami



## **Option Two: The Logical AND Operator (&&)**

If we want the second command to only run if the first command is successful, separate the commands with the logical AND operator, which is two ampersands **( && )**. For example, we want to make a directory called MyFolder and then change to that directory–provided it was successfully created. So, we type the following on the command line and press Enter.

mkdir MyFolder && cd MyFolder



The folder was successfully created, so the cd command was executed and we are now in the new folder.

*We recommend using the logical AND operator rather than the semicolon operator most of the time (;). This ensures that you don’t do anything disastrous.*

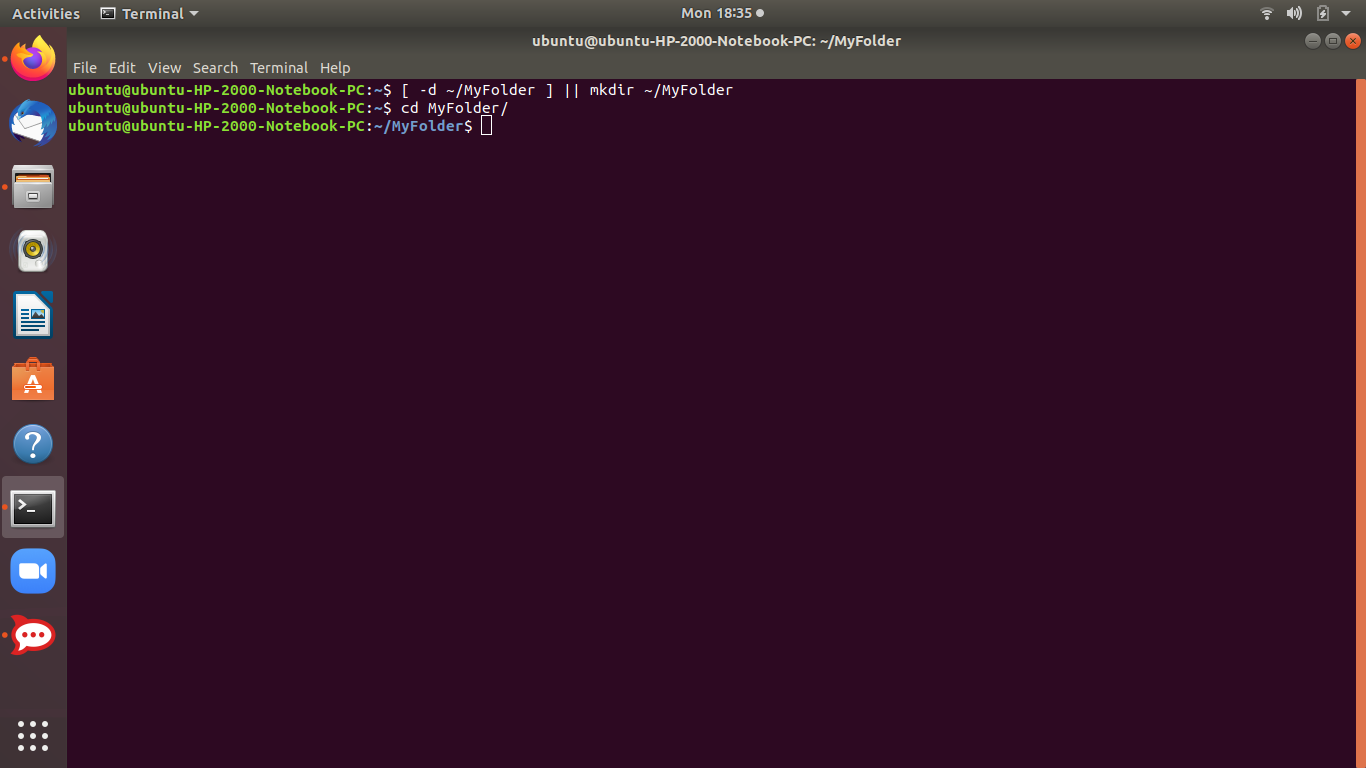
For example, if we run a command to change to a directory and then force remove everything in that directory recursively ( cd /some\_directory ; rm -Rf \* ), you could end up ruining your system if the directory change didn’t happen. e.

## 

## **Option Three: The Logical OR Operator (||)**

Sometimes you might want to execute a second command only if the first command does *not* succeed. To do this, we use the logical OR operator, ortwo vertical bars **( || )**. For example, we want to check to see if the MyFolder directory exists ( [ -d ~/MyFolder ] ) and create it if it doesn’t ( mkdir ~/MyFolder ). So, we type the following command at the prompt and press Enter.

[ -d ~/MyFolder ] || mkdir ~/MyFolder



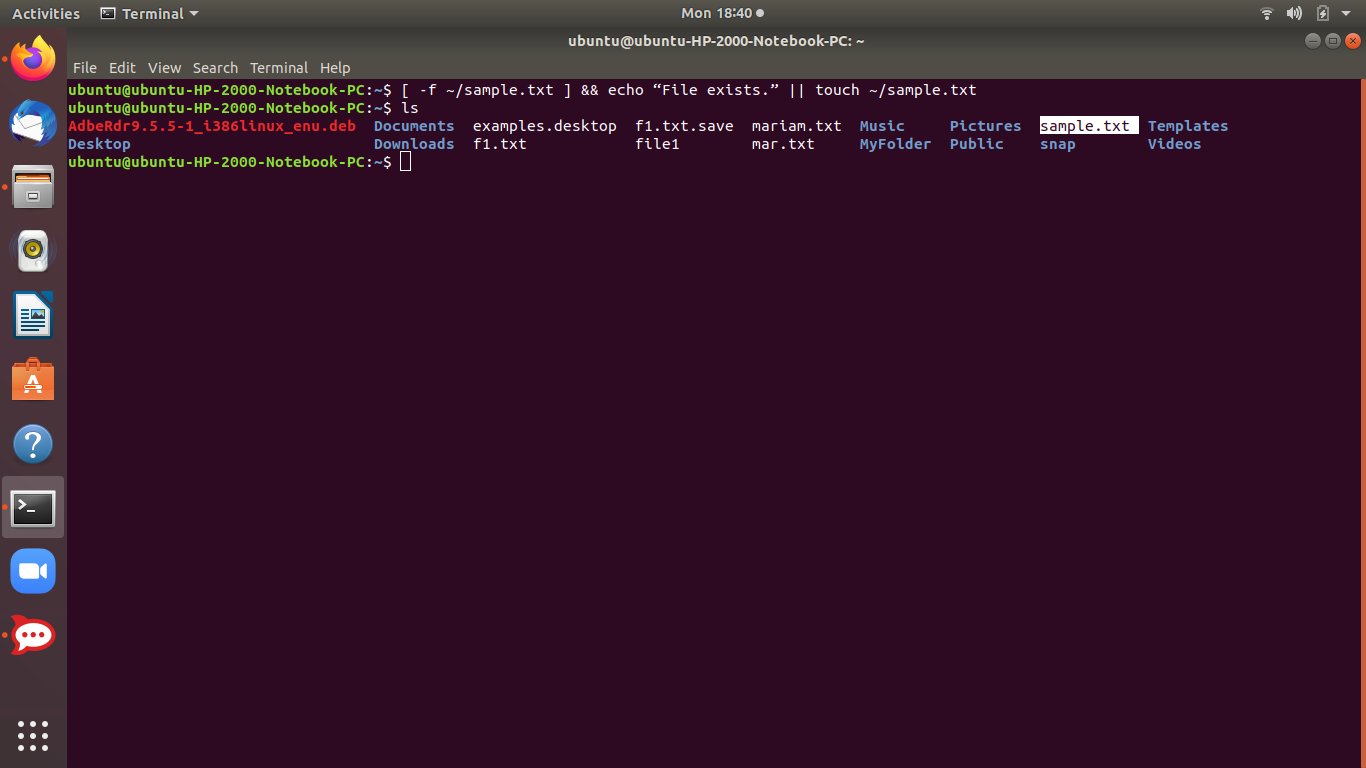
Be sure there is a space after the first bracket and before the second bracket or the first command that checks if the directory exists will not work.

In our example, the MyFolder directory does not exist, so the second command creates the directory.

## **Combining Multiple Operators**

We can combine multiple operators on the command line, too. For example, we want to first check if a file exists ( [ -f ~/sample.txt ] ). If it does, we print a message to the screen saying so ( echo “File exists.” ). If not, we create the file ( touch ~/sample.txt ). So, we type the following at the command prompt and press Enter.

[ -f ~/sample.txt ] && echo “File exists.” || touch ~/sample.txt



In my example, the file didn’t exist, so it was created.

Reference link:

<https://www.howtogeek.com/269509/how-to-run-two-or-more-terminal-commands-at-once-in-linux/>

Another important is pipe command “|”

<https://www.geeksforgeeks.org/piping-in-unix-or-linux/>

<https://www.guru99.com/linux-pipe-grep.html>