

COMSATS UNIVERSITY ISLAMABAD WAH CAMPUS



LAB MANUAL # 02

DEV OPS FOR CLOUD COMPUTING

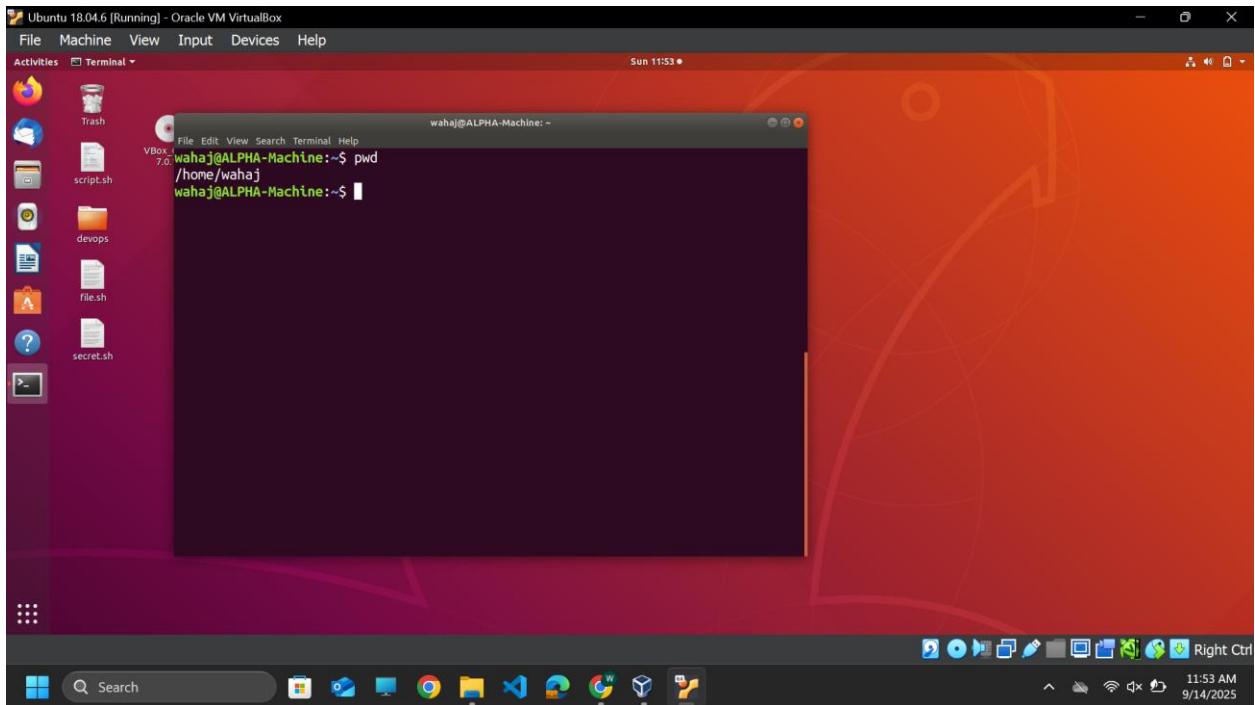
Submitted By: M. Wahaj .Yasin

Registration #: SP22-BSE-040

Submitted to: Mr Muhammad Ibrahim

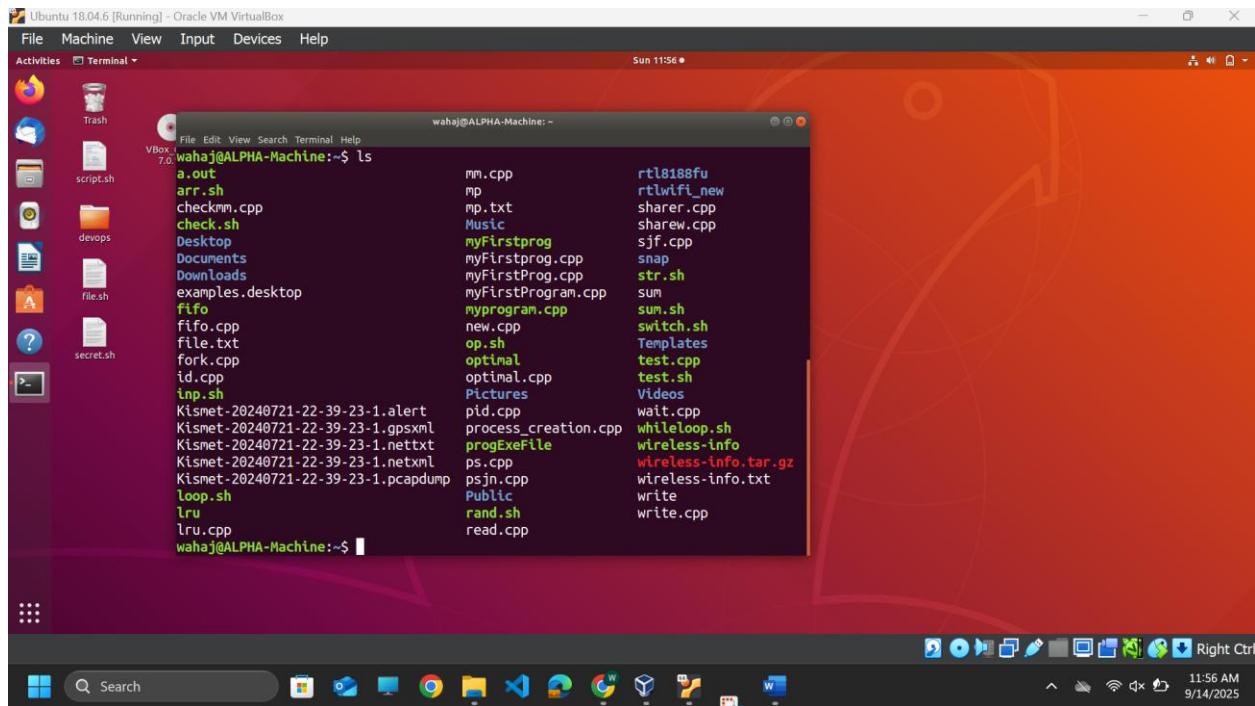
Basic Linux Commands

pwd

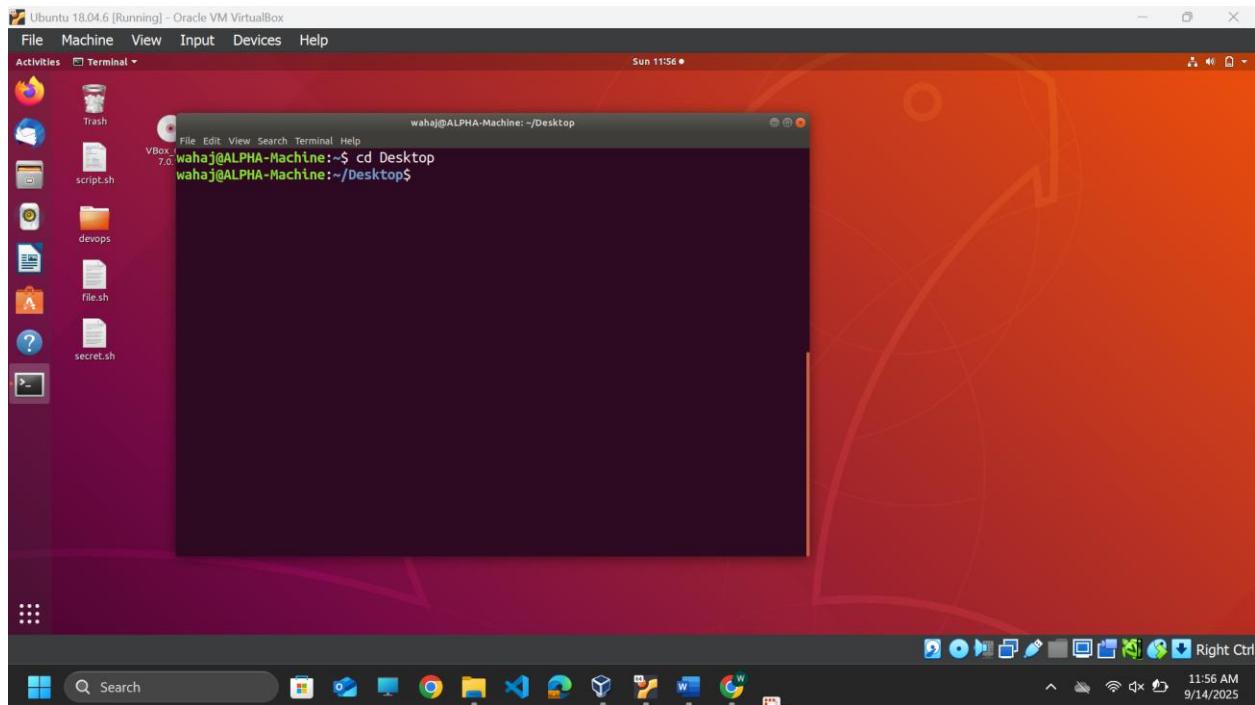


ls

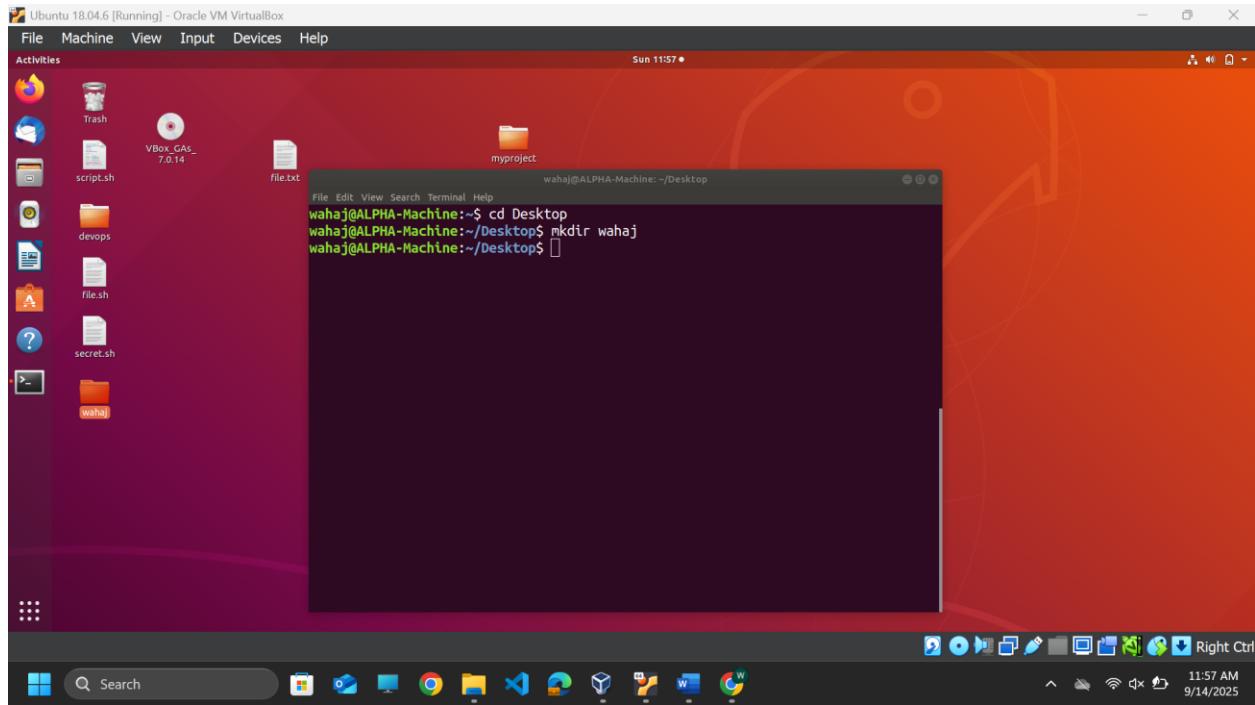
3



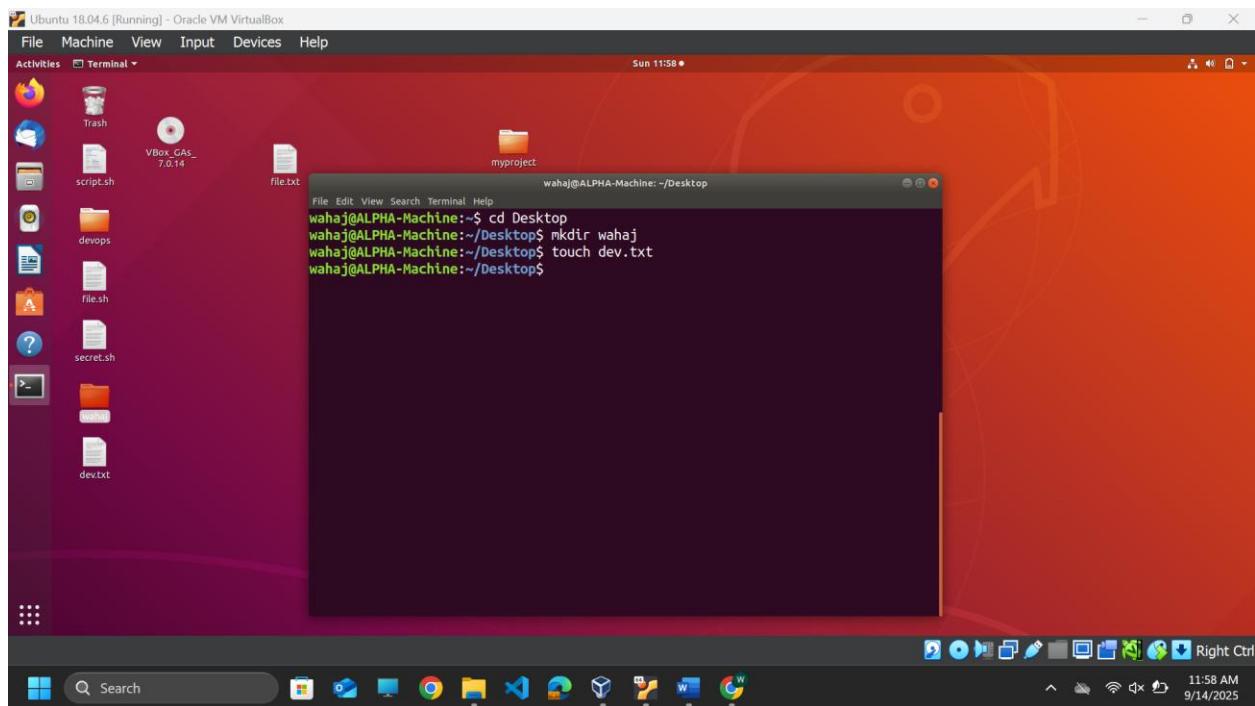
Cd



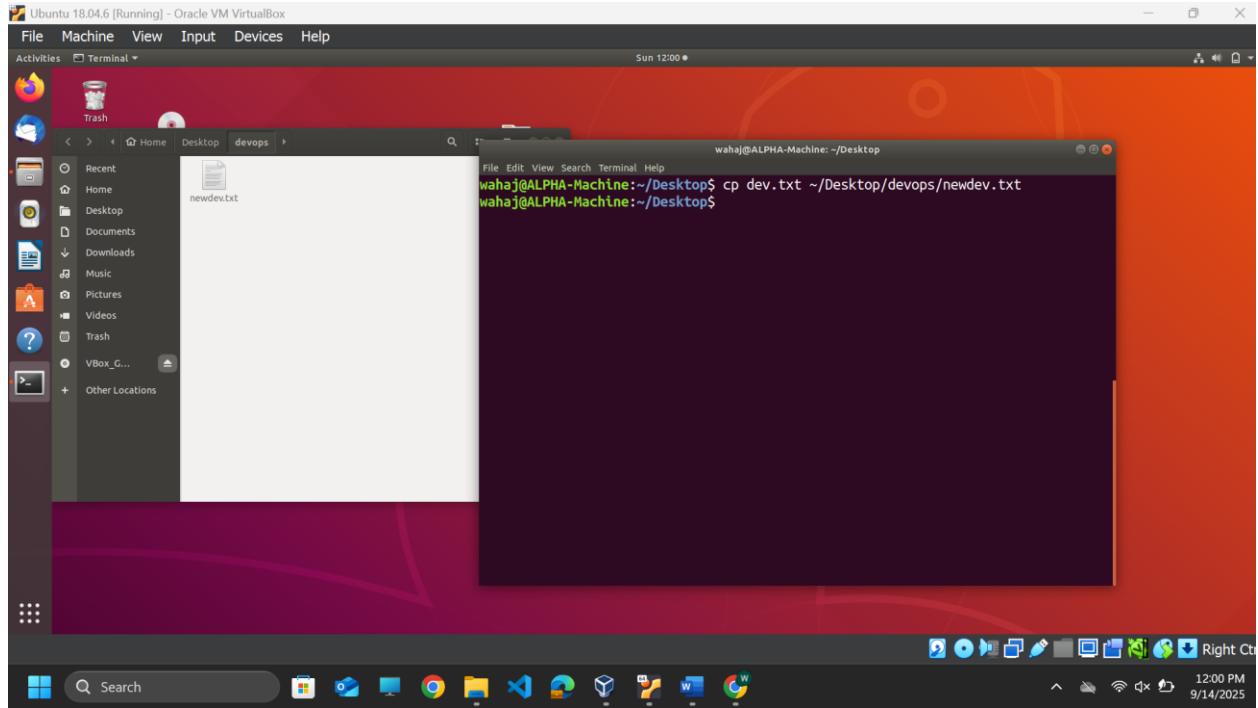
mkdir



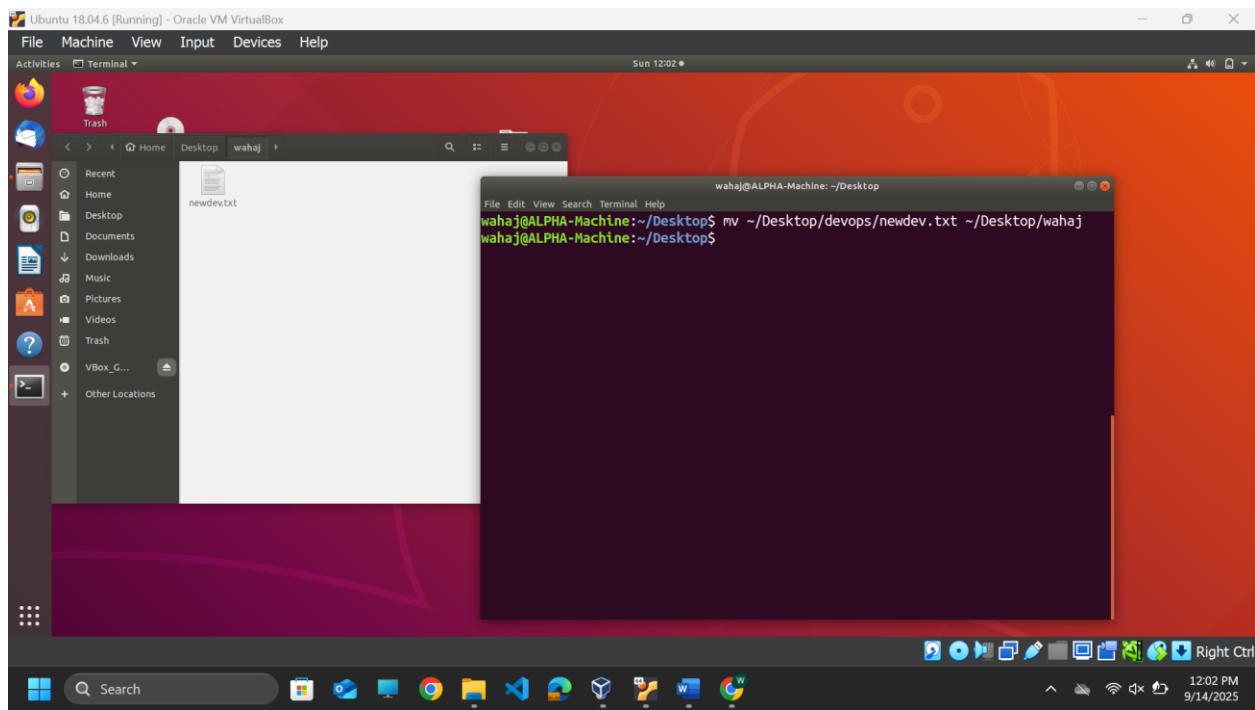
touch



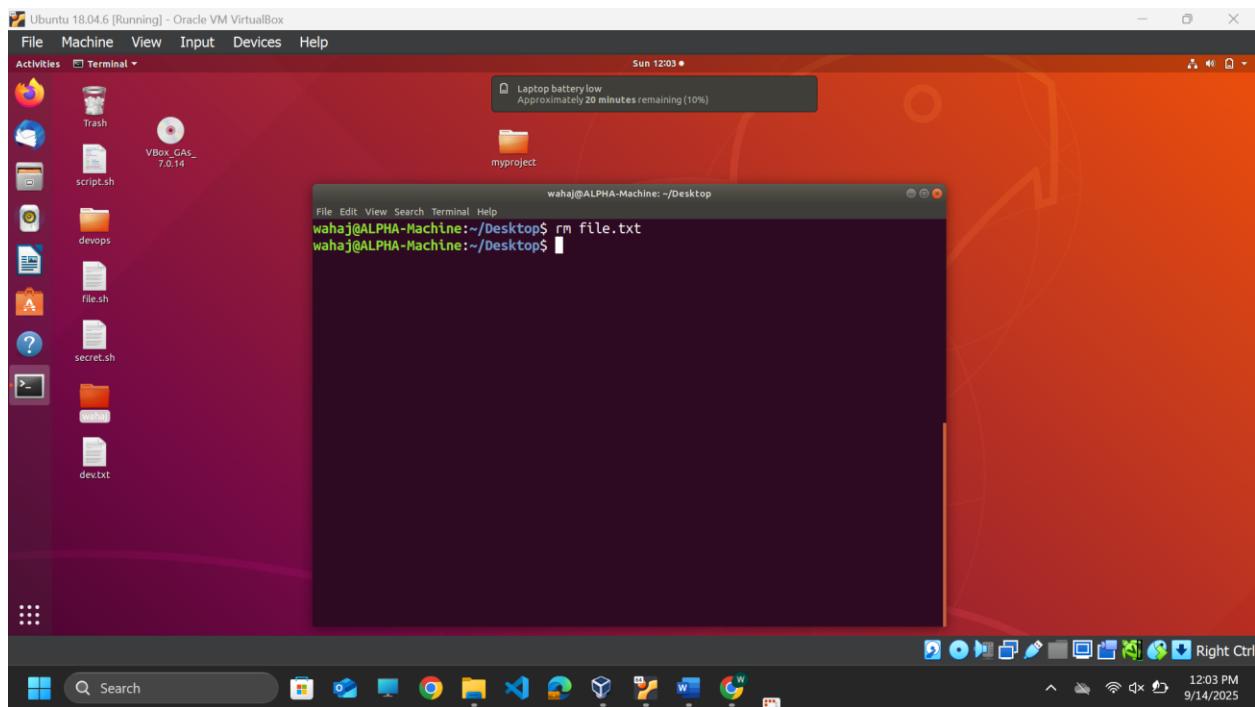
cp



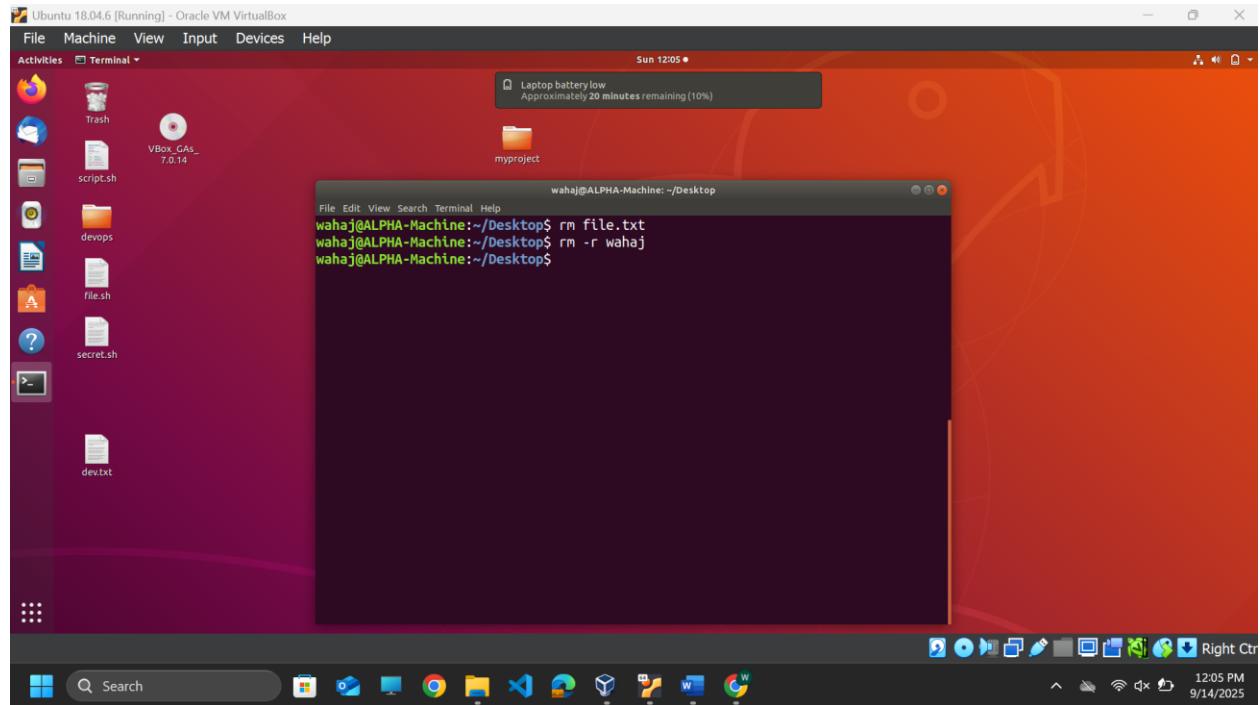
mv



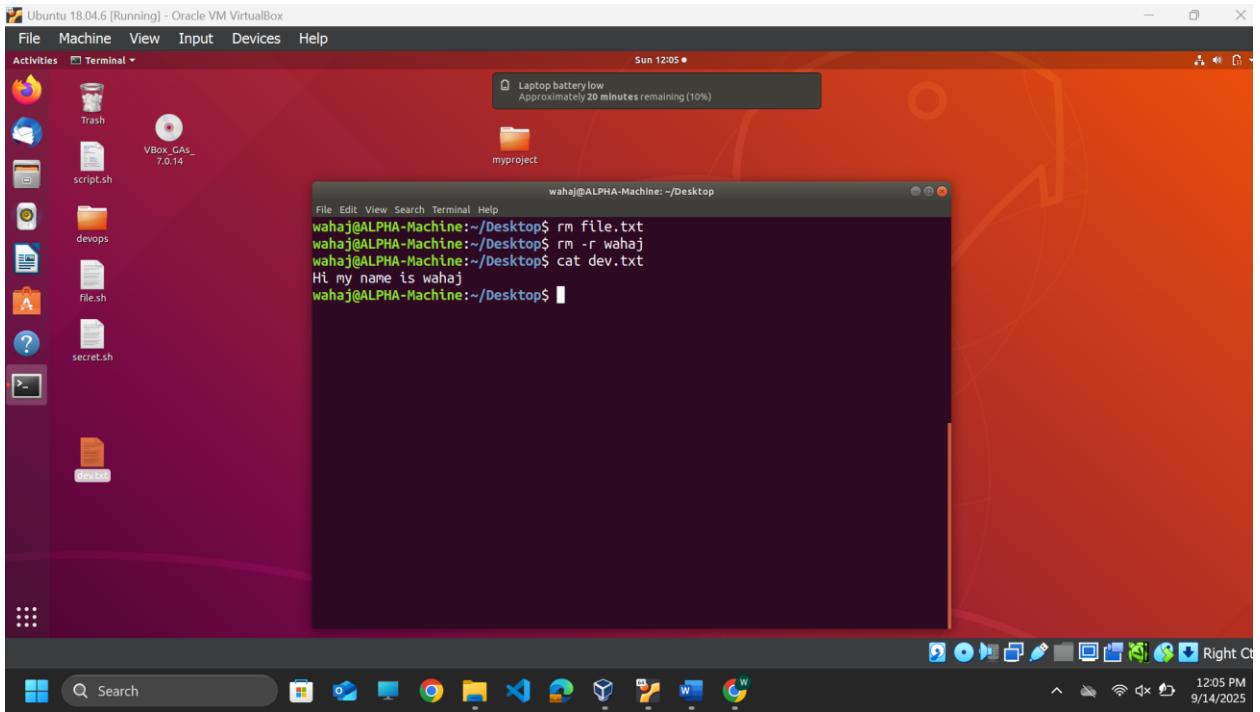
rm



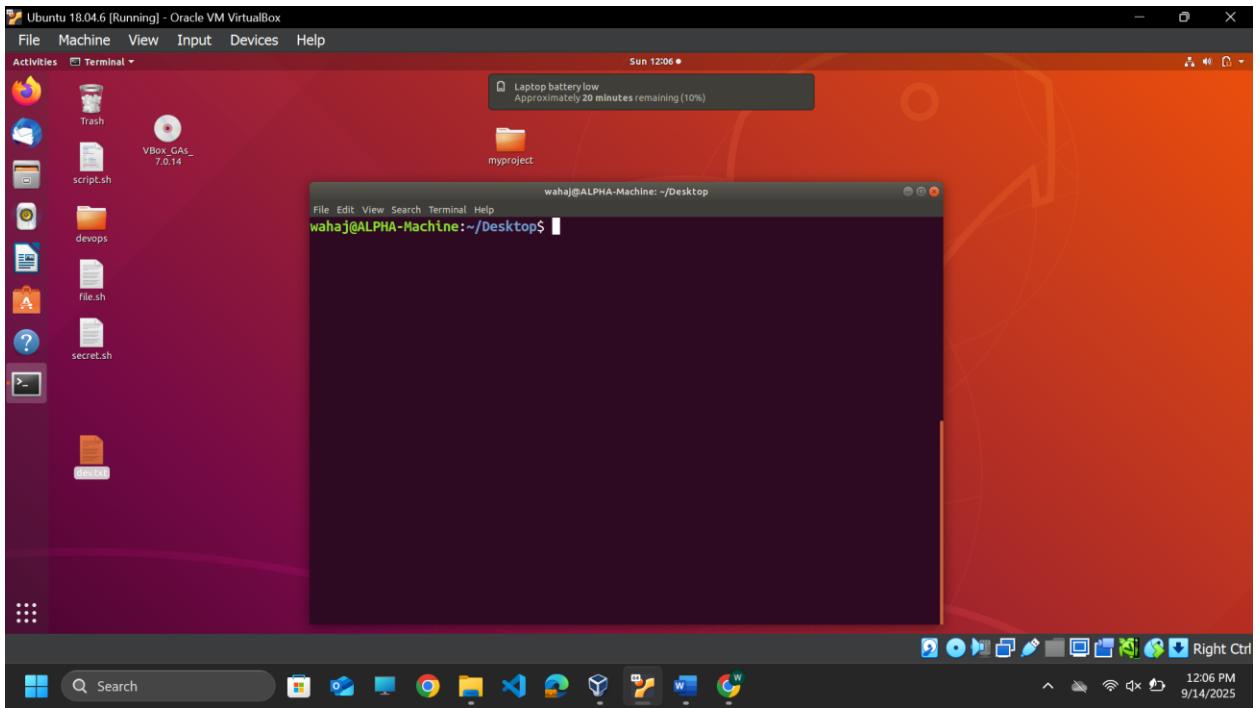
rm -r remove recursively



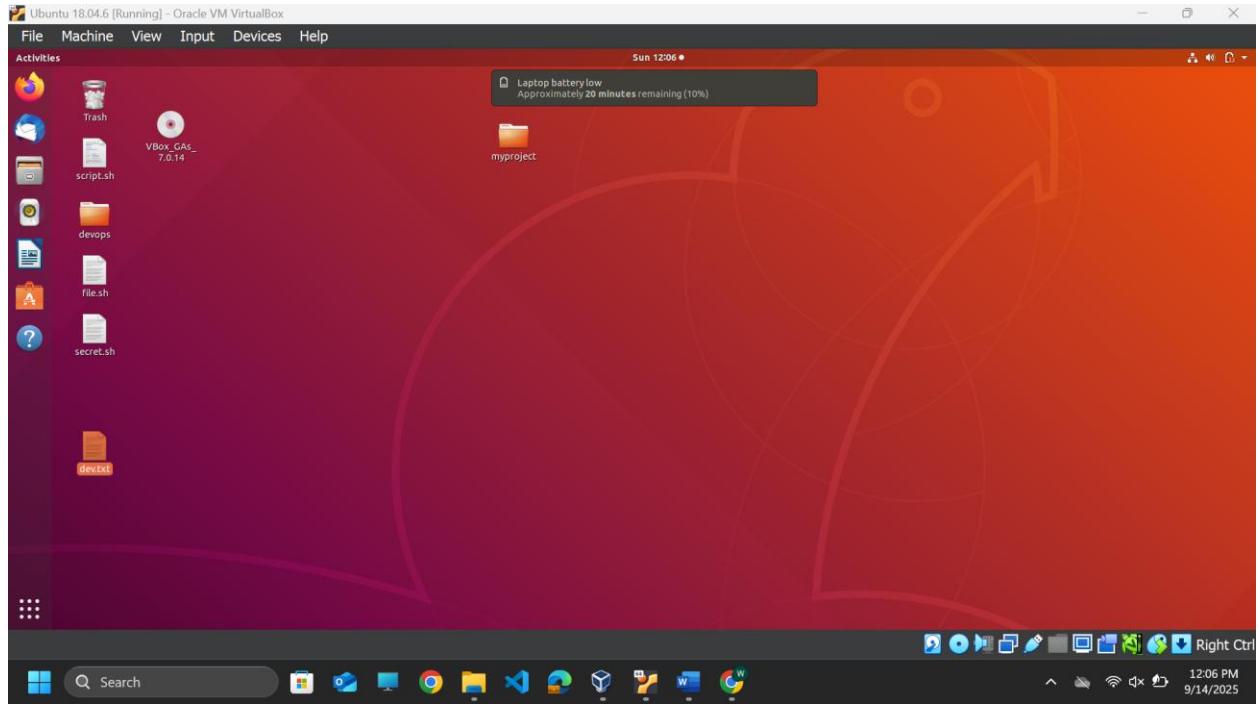
cat Show file contents



clear Clear the terminal

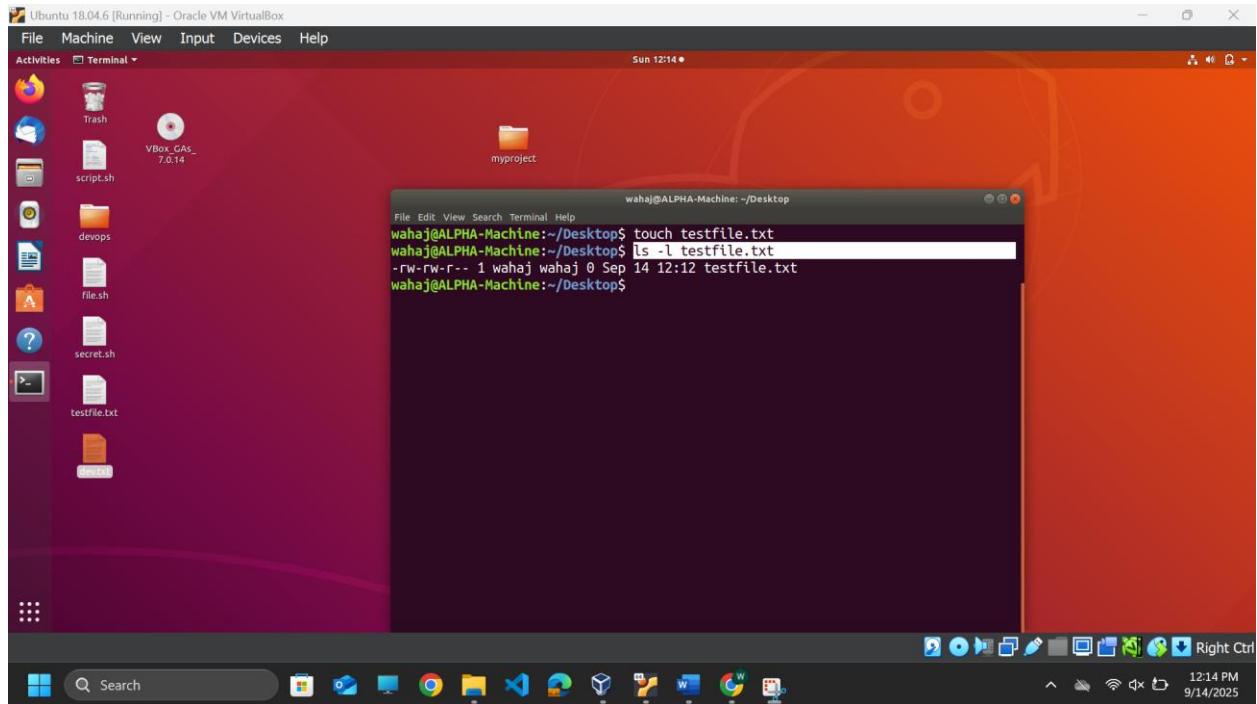


exit Close the terminal or shell

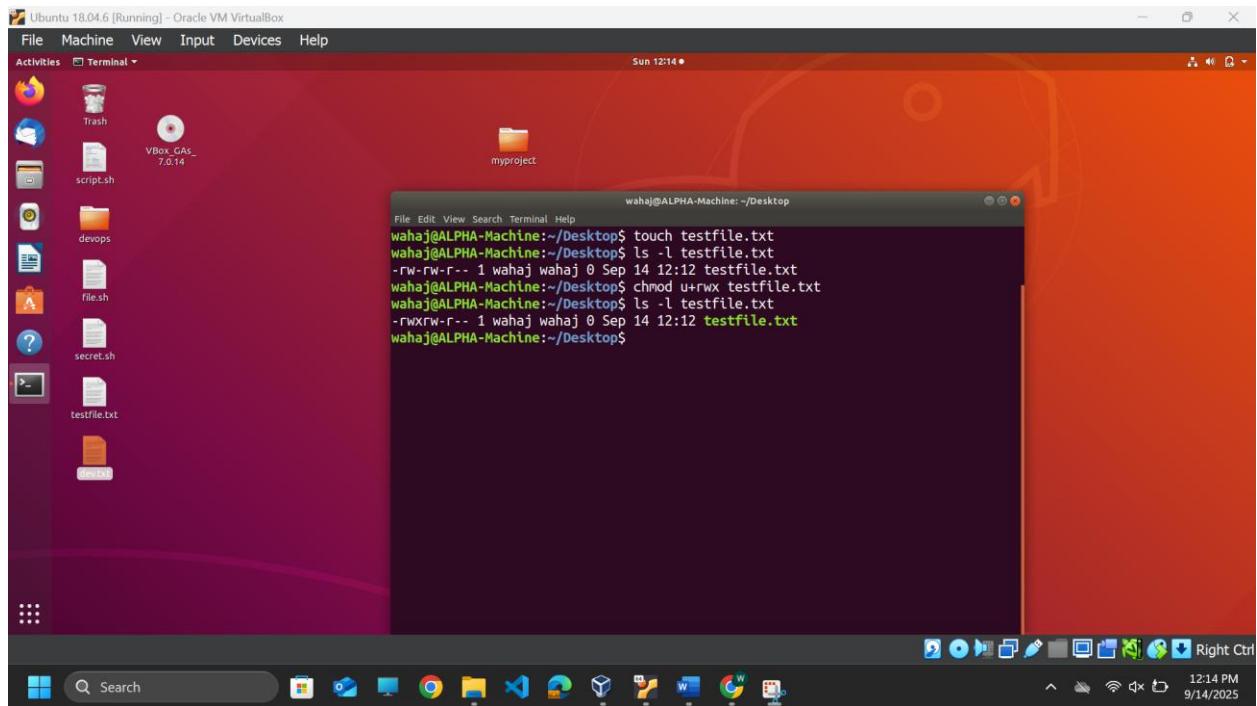


>User and Permissions

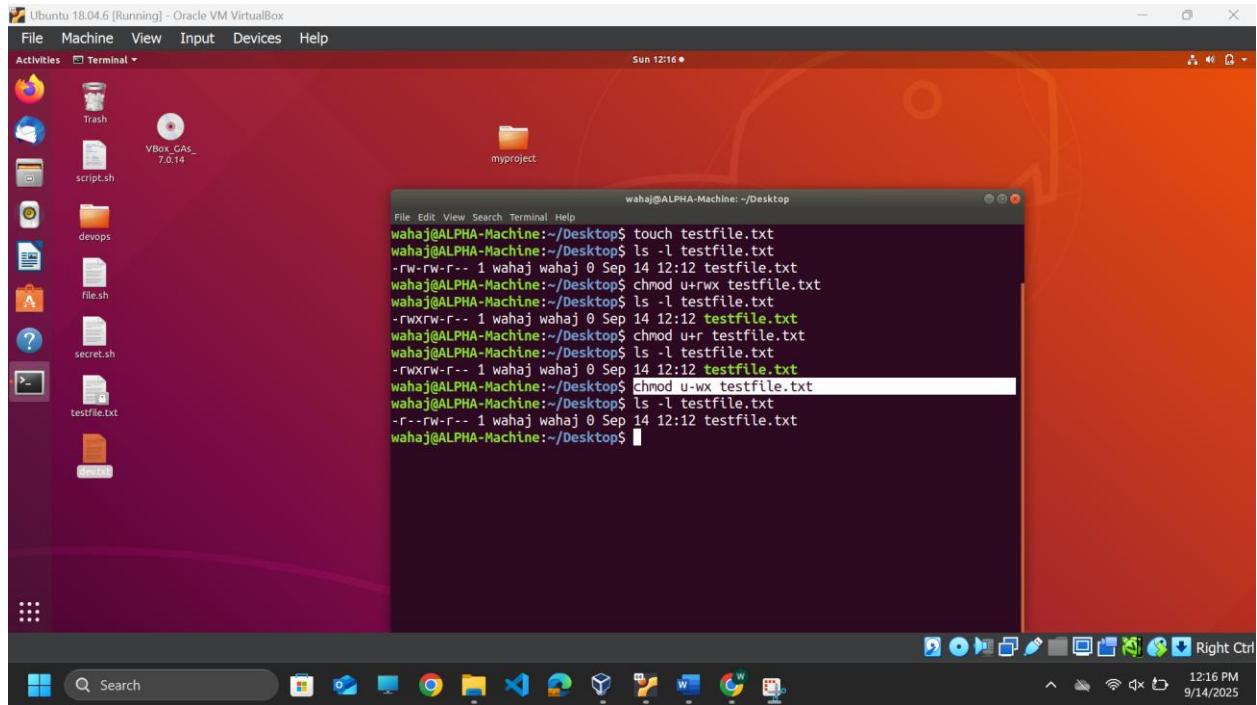
ls -l testfile.txt



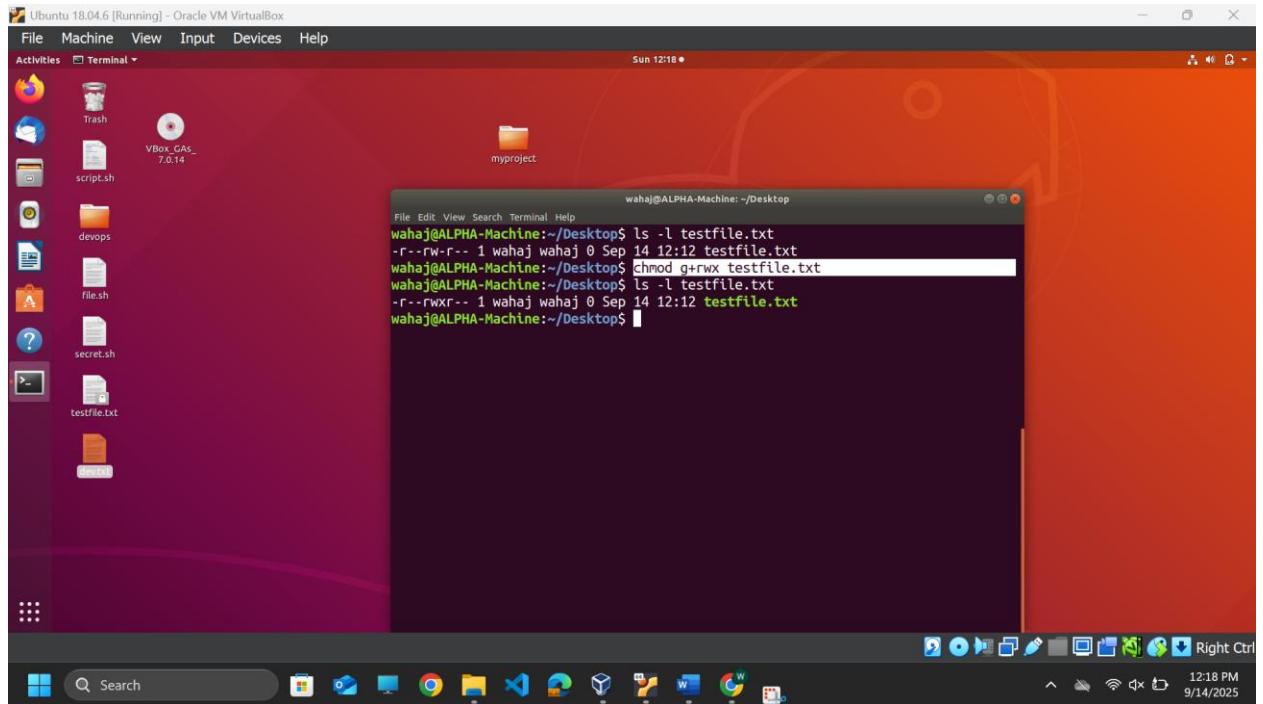
chmod u+rwx testfile.txt



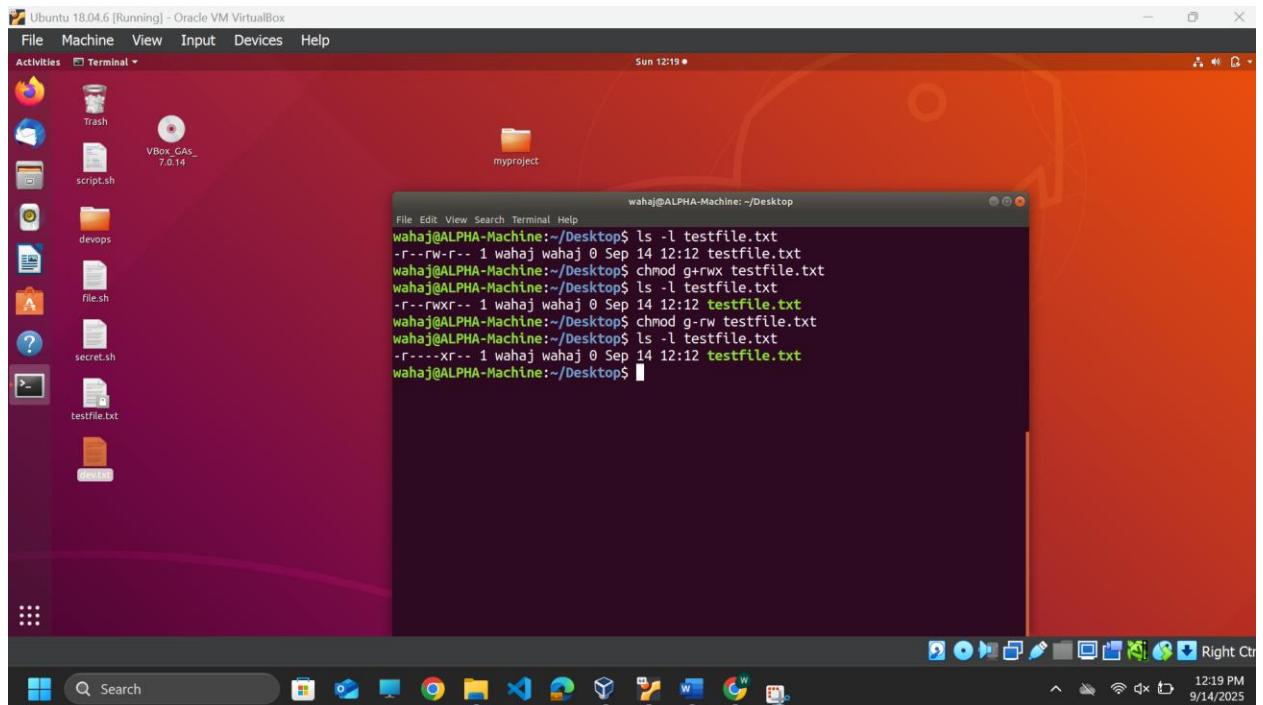
chmod u-wx testfile.txt



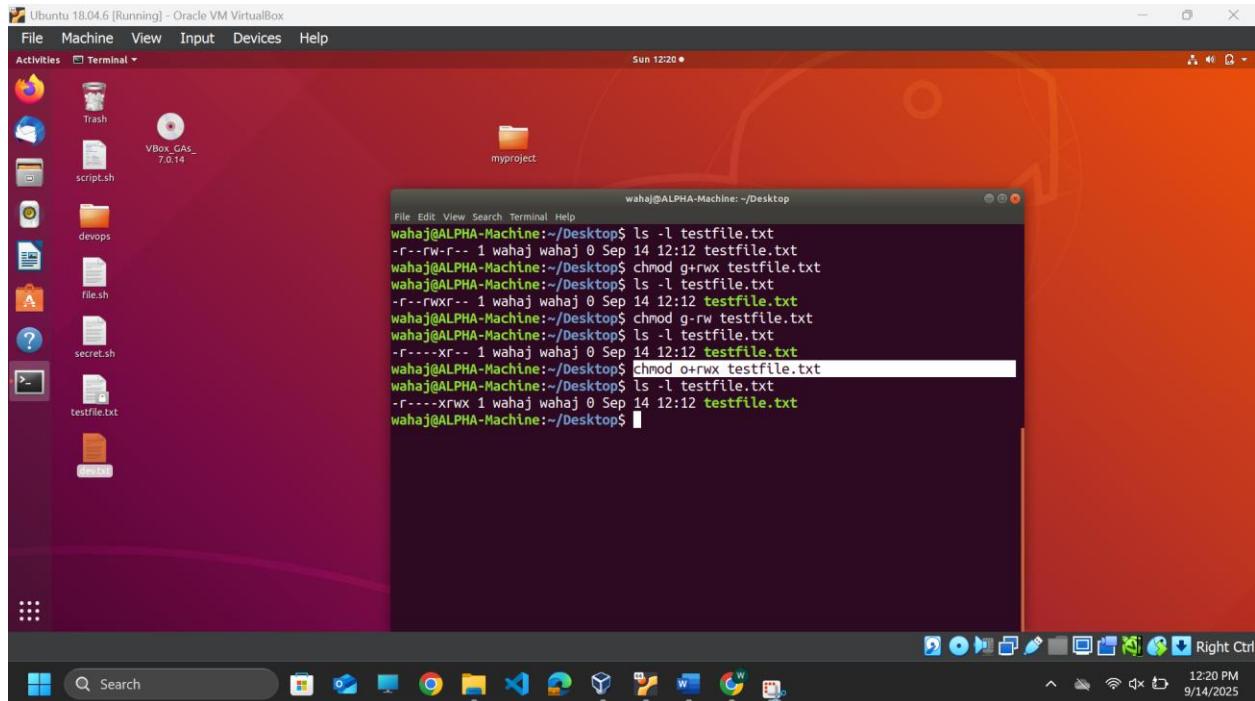
chmod g+rwx testfile.txt



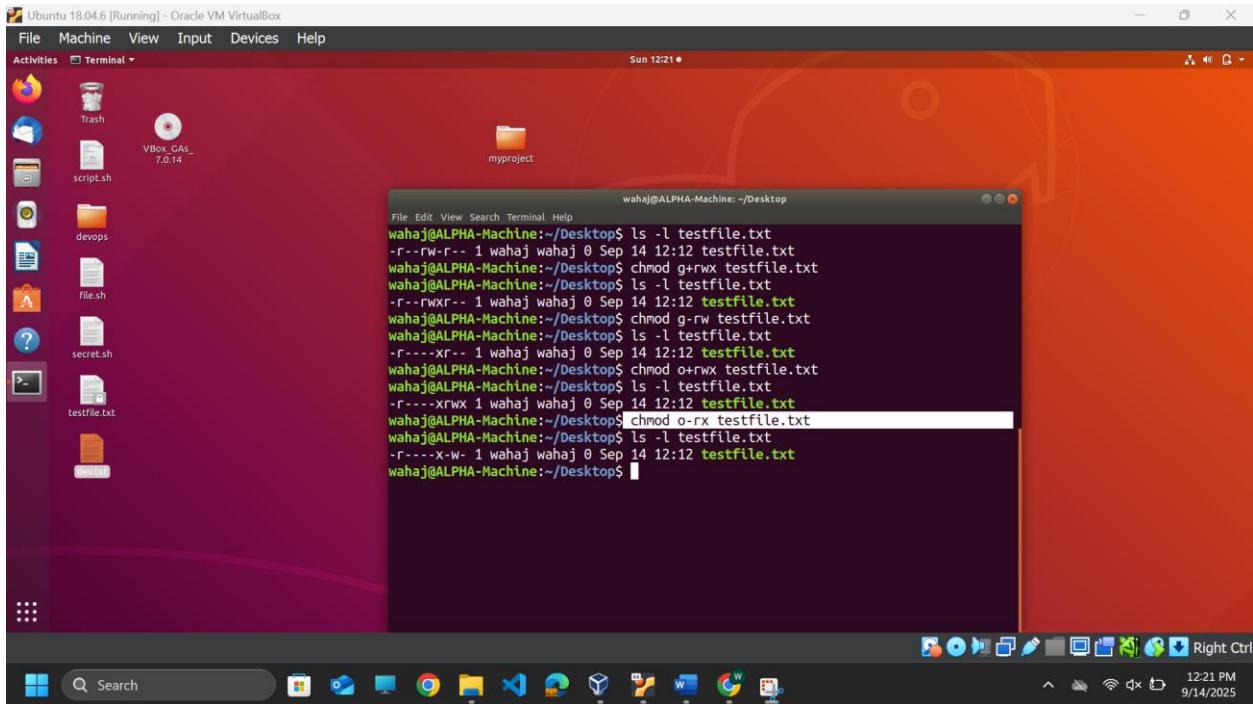
chmod g-rw testfile.txt



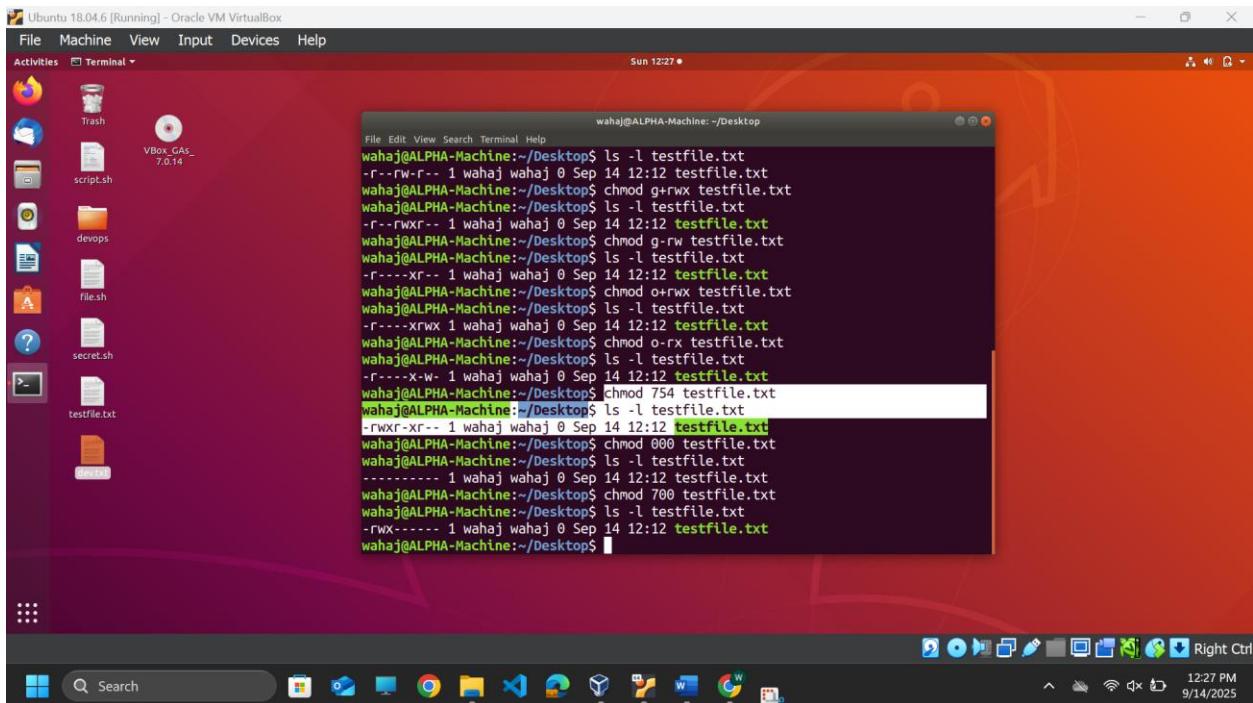
chmod o+rwx testfile.txt



chmod o-rx testfile.txt



chmod 754 testfile.txt



chmod 000 testfile.txt

The screenshot shows a desktop environment with a terminal window open. The terminal window title is "wahaj@ALPHA-Machine: ~/Desktop". The terminal output shows the following sequence of commands and their results:

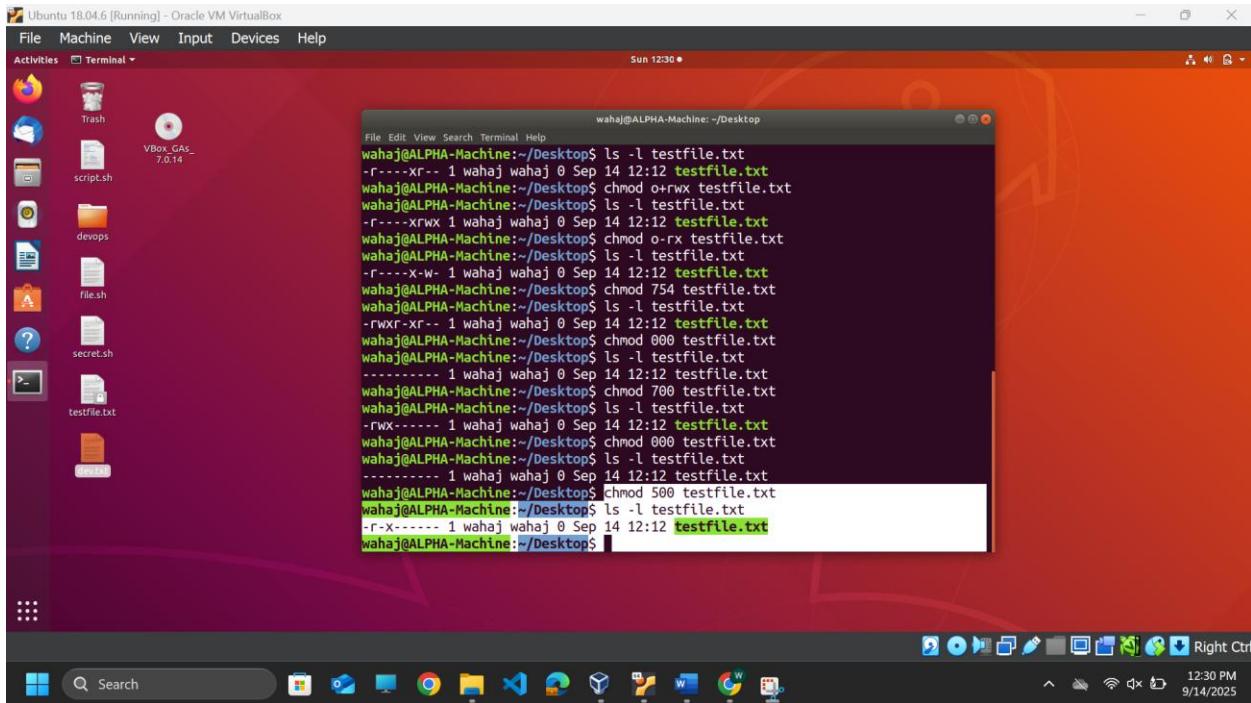
```
wahaj@ALPHA-Machine:~/Desktop$ ls -l testfile.txt
-r--rw-r-- 1 wahaj wahaj 0 Sep 14 12:12 testfile.txt
wahaj@ALPHA-Machine:~/Desktop$ chmod g+rwx testfile.txt
wahaj@ALPHA-Machine:~/Desktop$ ls -l testfile.txt
-r--rwxr- 1 wahaj wahaj 0 Sep 14 12:12 testfile.txt
wahaj@ALPHA-Machine:~/Desktop$ chmod g-rw testfile.txt
wahaj@ALPHA-Machine:~/Desktop$ ls -l testfile.txt
-r-----xr- 1 wahaj wahaj 0 Sep 14 12:12 testfile.txt
wahaj@ALPHA-Machine:~/Desktop$ chmod o+rwx testfile.txt
wahaj@ALPHA-Machine:~/Desktop$ ls -l testfile.txt
-r-----xw- 1 wahaj wahaj 0 Sep 14 12:12 testfile.txt
wahaj@ALPHA-Machine:~/Desktop$ chmod 754 testfile.txt
wahaj@ALPHA-Machine:~/Desktop$ ls -l testfile.txt
-rwxr--xr- 1 wahaj wahaj 0 Sep 14 12:12 testfile.txt
wahaj@ALPHA-Machine:~/Desktop$ chmod 000 testfile.txt
wahaj@ALPHA-Machine:~/Desktop$ ls -l testfile.txt
----- 1 wahaj wahaj 0 Sep 14 12:12 testfile.txt
wahaj@ALPHA-Machine:~/Desktop$ chmod 700 testfile.txt
wahaj@ALPHA-Machine:~/Desktop$ ls -l testfile.txt
-rwx----- 1 wahaj wahaj 0 Sep 14 12:12 testfile.txt
wahaj@ALPHA-Machine:~/Desktop$
```

chmod 700 testfile.txt

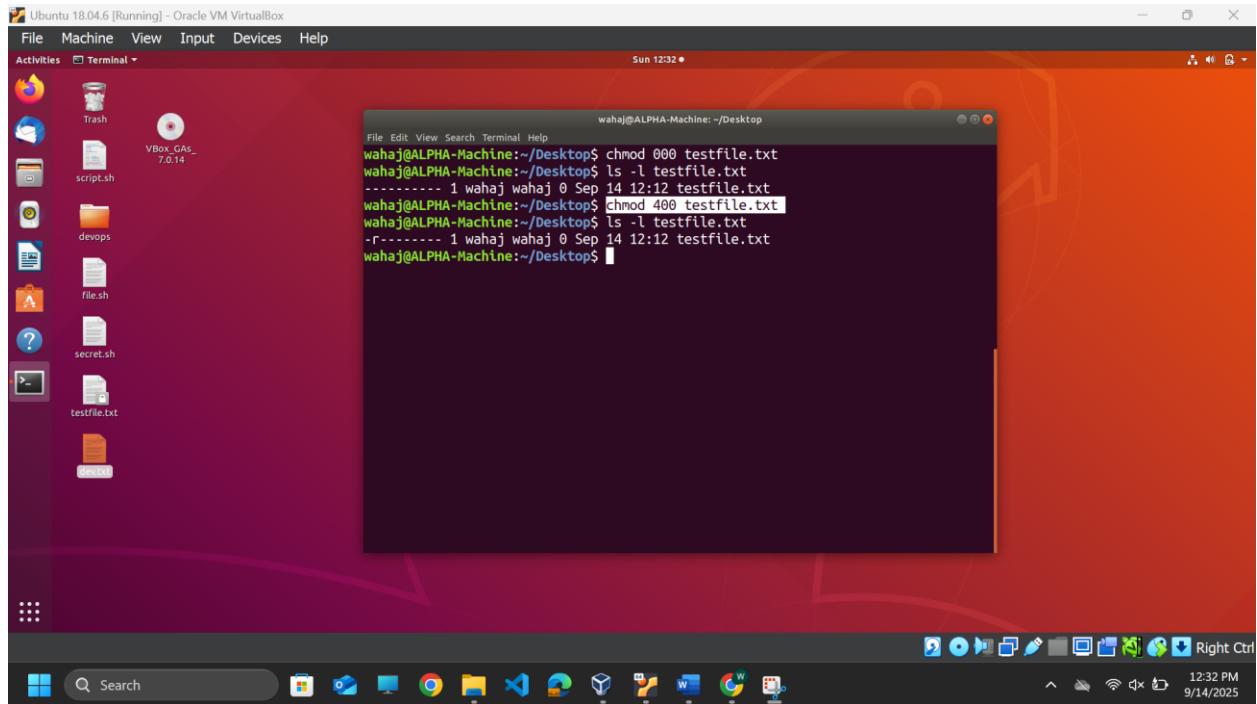
The screenshot shows a desktop environment with a terminal window open. The terminal window title is "wahaj@ALPHA-Machine: ~/Desktop". The terminal output shows the following sequence of commands and their results:

```
wahaj@ALPHA-Machine:~/Desktop$ ls -l testfile.txt
-r--rw-r-- 1 wahaj wahaj 0 Sep 14 12:12 testfile.txt
wahaj@ALPHA-Machine:~/Desktop$ chmod g+rwx testfile.txt
wahaj@ALPHA-Machine:~/Desktop$ ls -l testfile.txt
-r--rwxr- 1 wahaj wahaj 0 Sep 14 12:12 testfile.txt
wahaj@ALPHA-Machine:~/Desktop$ chmod g-rw testfile.txt
wahaj@ALPHA-Machine:~/Desktop$ ls -l testfile.txt
-r-----xr- 1 wahaj wahaj 0 Sep 14 12:12 testfile.txt
wahaj@ALPHA-Machine:~/Desktop$ chmod o+rwx testfile.txt
wahaj@ALPHA-Machine:~/Desktop$ ls -l testfile.txt
-r-----xw- 1 wahaj wahaj 0 Sep 14 12:12 testfile.txt
wahaj@ALPHA-Machine:~/Desktop$ chmod 754 testfile.txt
wahaj@ALPHA-Machine:~/Desktop$ ls -l testfile.txt
-rwxr--xr- 1 wahaj wahaj 0 Sep 14 12:12 testfile.txt
wahaj@ALPHA-Machine:~/Desktop$ chmod 000 testfile.txt
wahaj@ALPHA-Machine:~/Desktop$ ls -l testfile.txt
----- 1 wahaj wahaj 0 Sep 14 12:12 testfile.txt
wahaj@ALPHA-Machine:~/Desktop$ chmod 700 testfile.txt
wahaj@ALPHA-Machine:~/Desktop$ ls -l testfile.txt
-rwx----- 1 wahaj wahaj 0 Sep 14 12:12 testfile.txt
wahaj@ALPHA-Machine:~/Desktop$
```

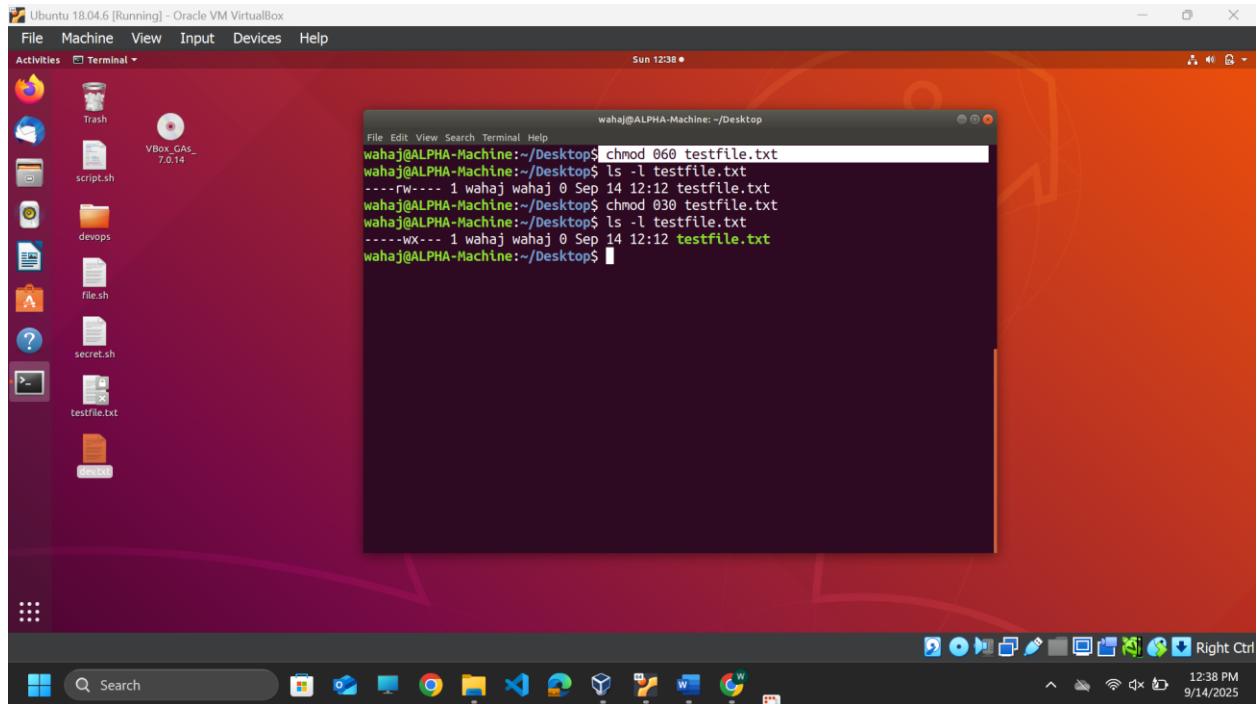
chmod 500 testfile.txt



chmod 400 testfile.txt

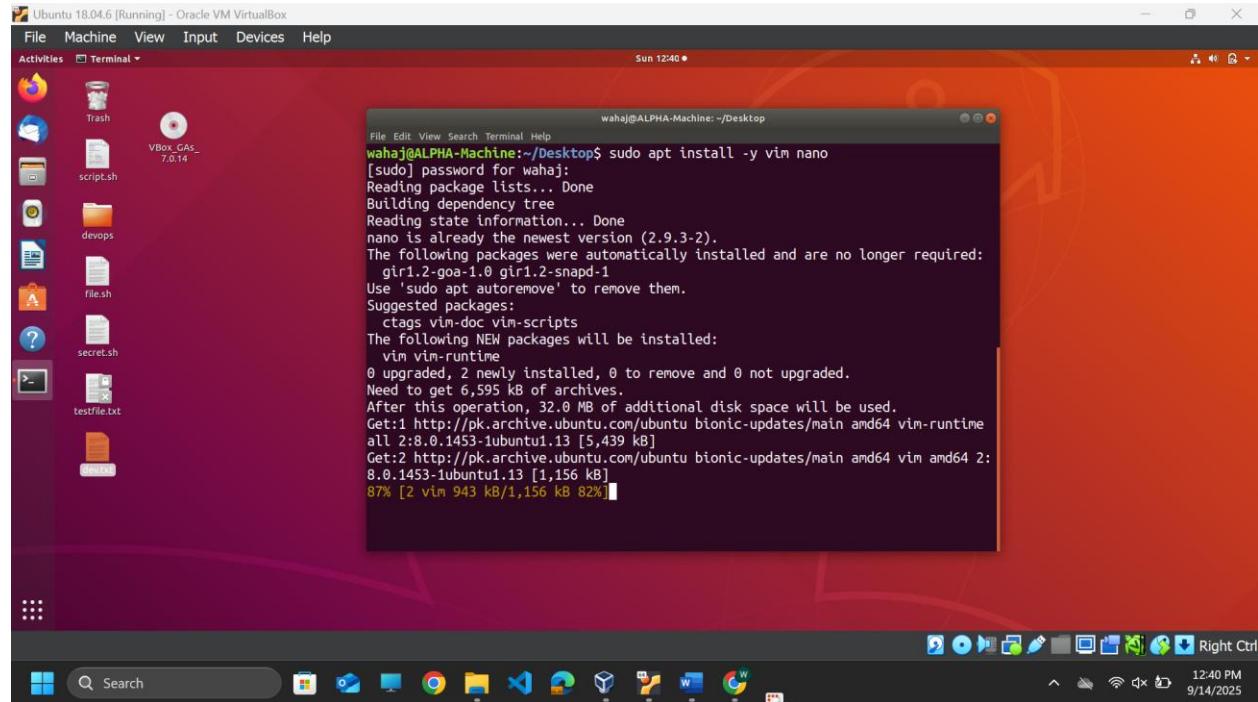


chmod 060 testfile.txt



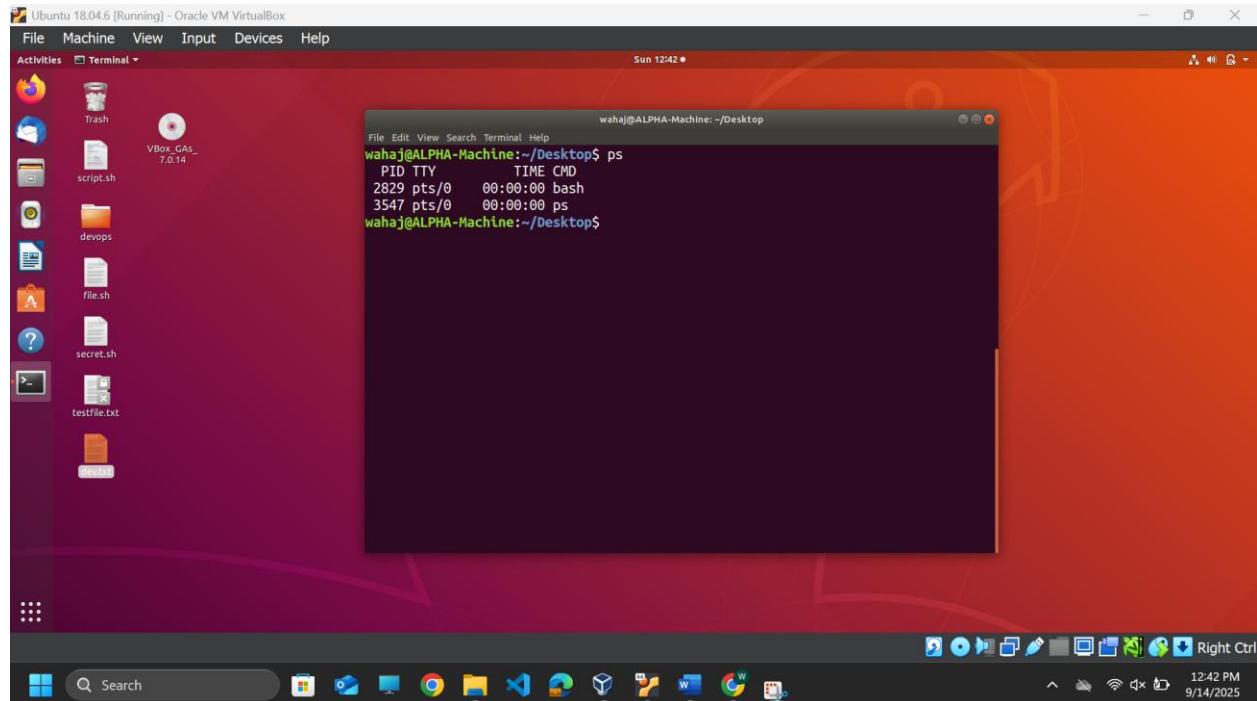
Package Management (Installing Software)

sudo apt install -y vim nano -y (assume yes to all prompts)

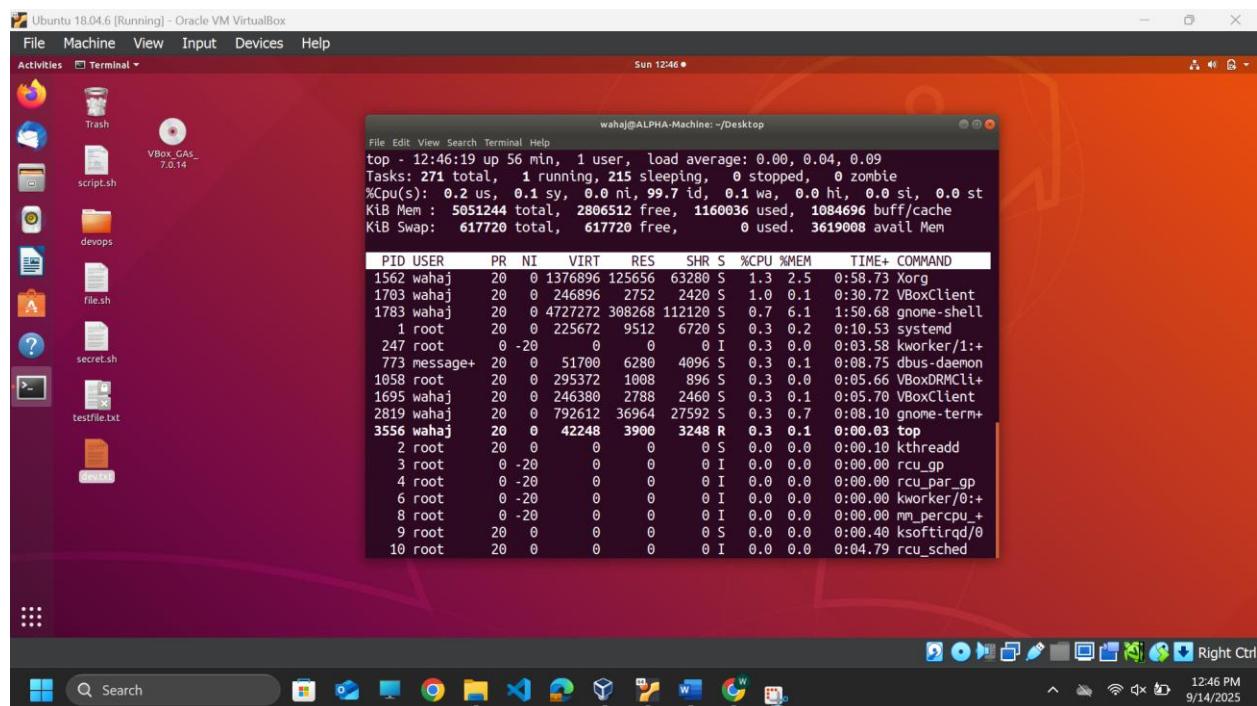


Processes and System Monitoring

ps Show running processes



top Live view of system usage (CPU, memory)



kill Stop a process

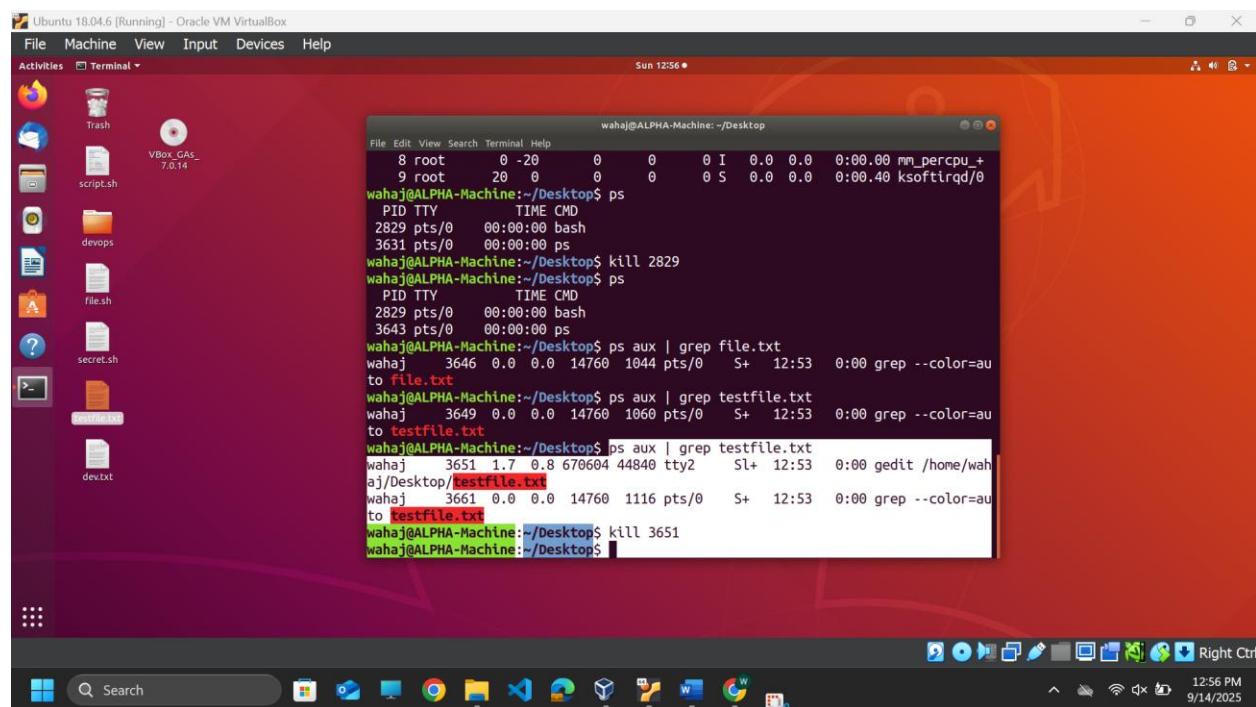
1. ps aux

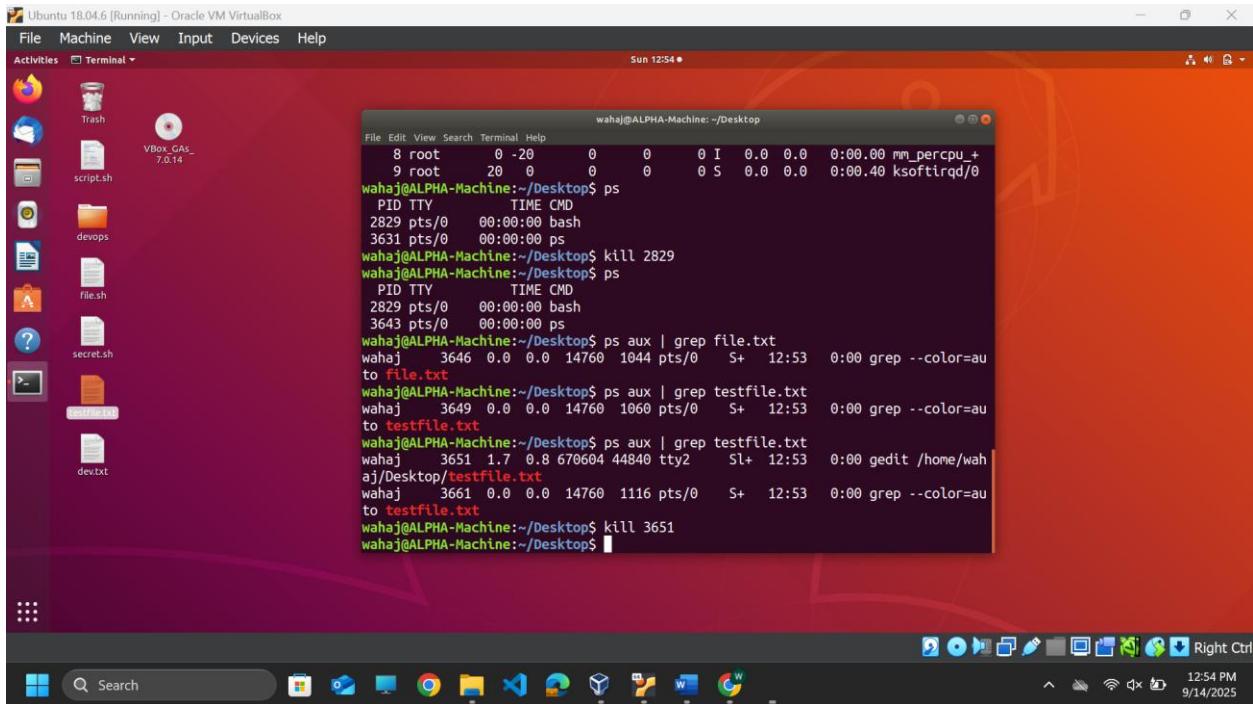
- ps → “process status” → lists running processes.
- a → show processes of all users, not just you.
- u → show user/owner of the process.
- x → show processes not attached to a terminal (background processes).

In short: lists all running processes with details like PID, CPU usage, memory, and command.

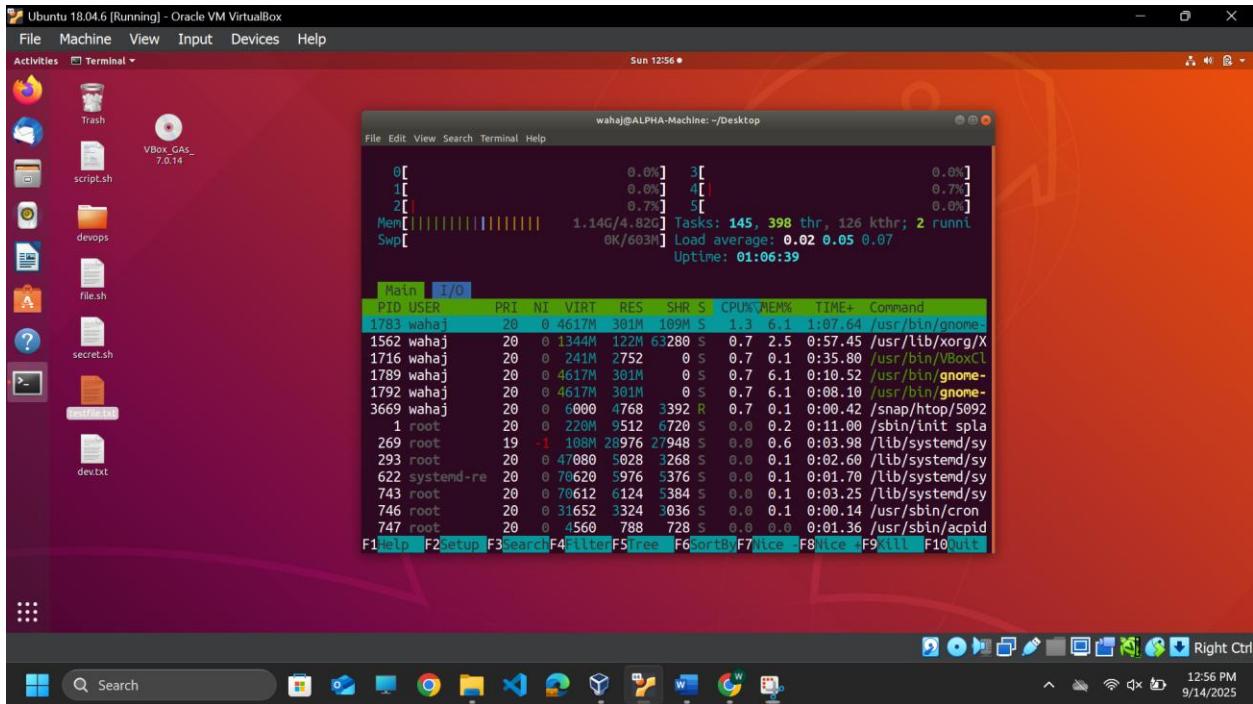
2. | grep testfile.txt

- | → pipe → takes the output of the first command (ps aux) and sends it to the next command.
- grep testfile.txt → searches for the word testfile.txt in the list of processes.

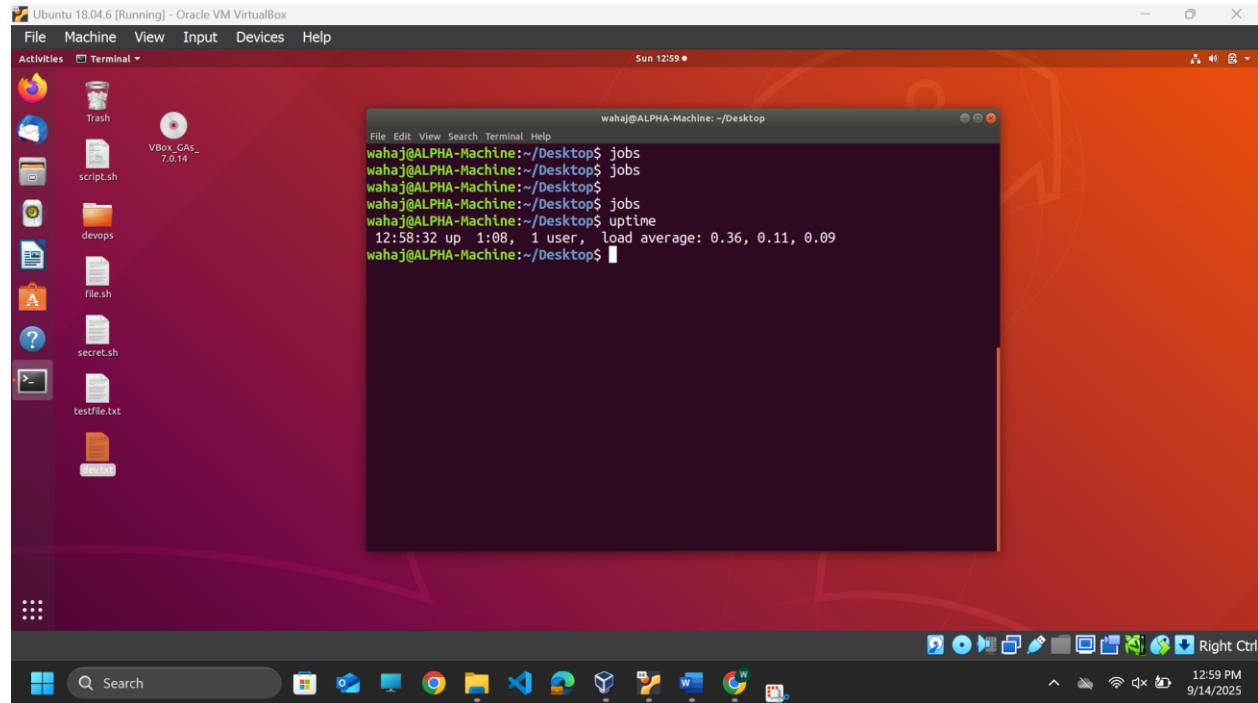




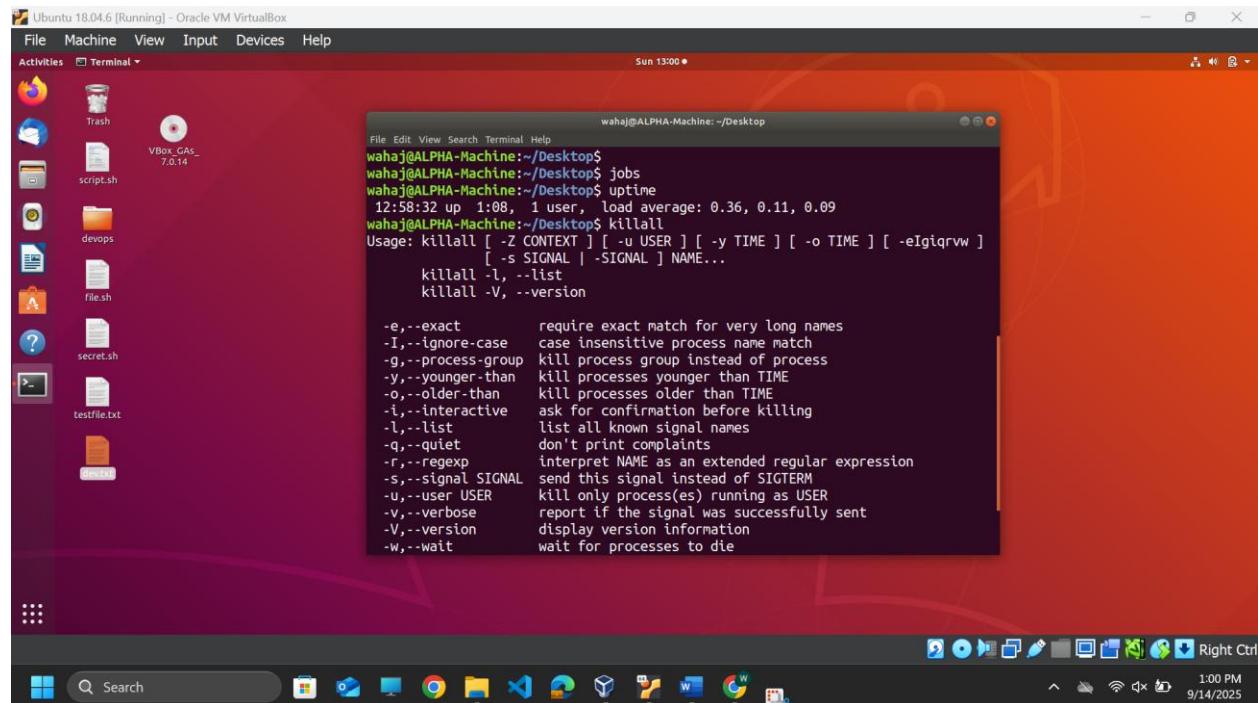
htop Better version of top (if installed)



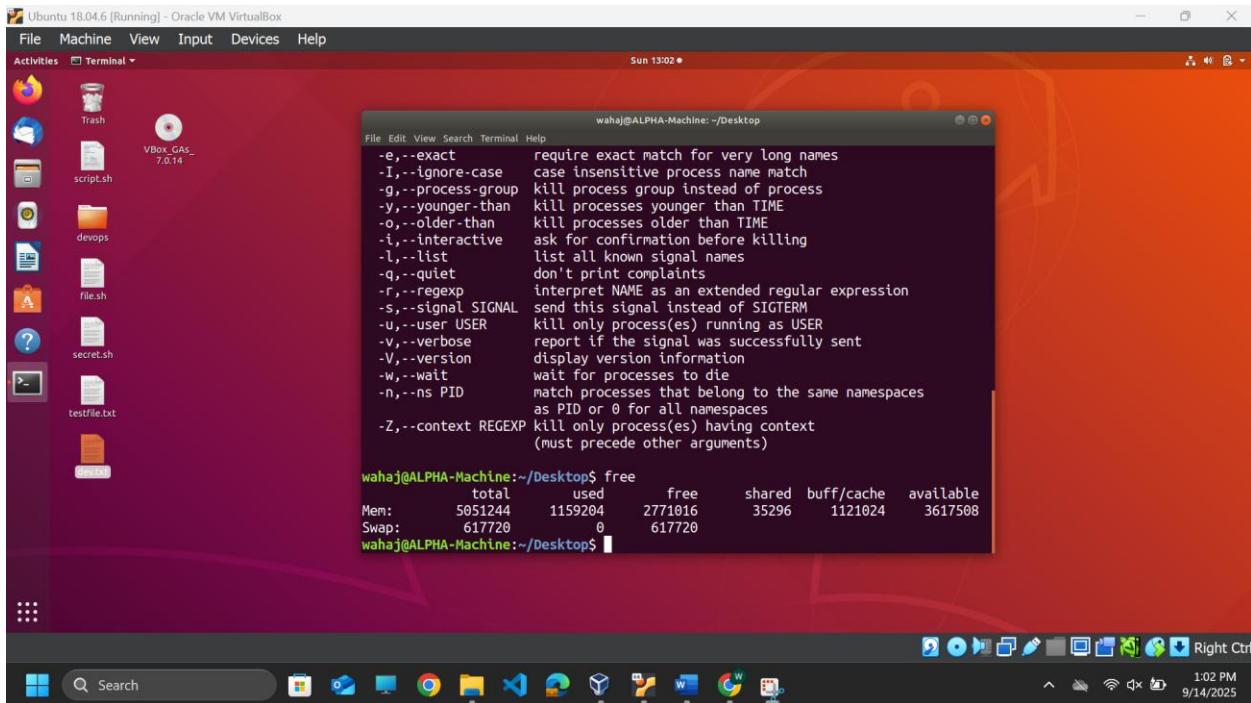
uptime



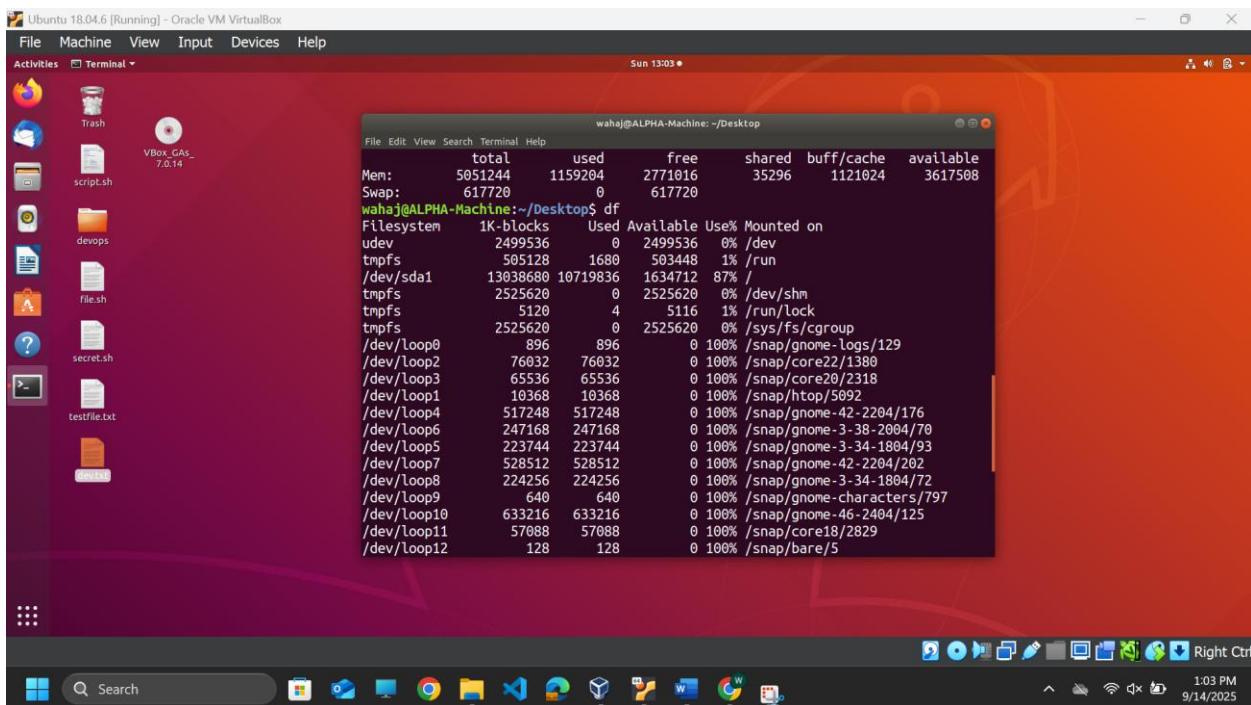
killall (Stop all processes by name)



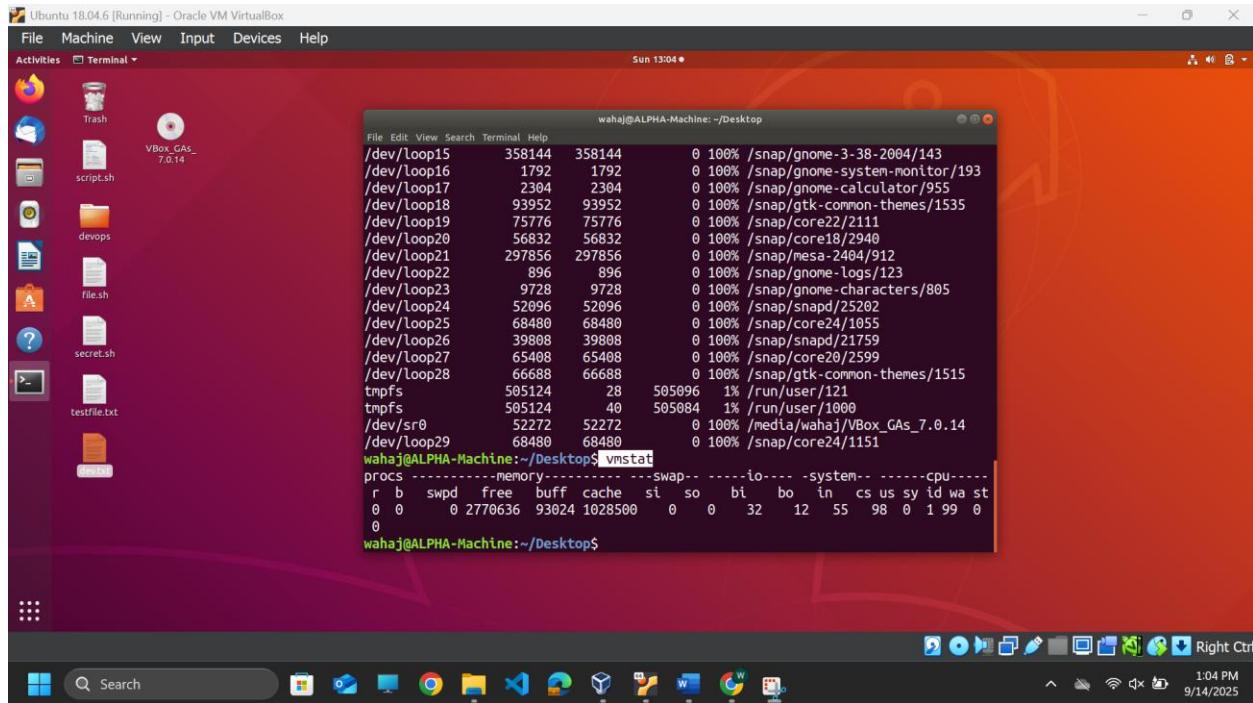
free show memory usage (RAM and swap)



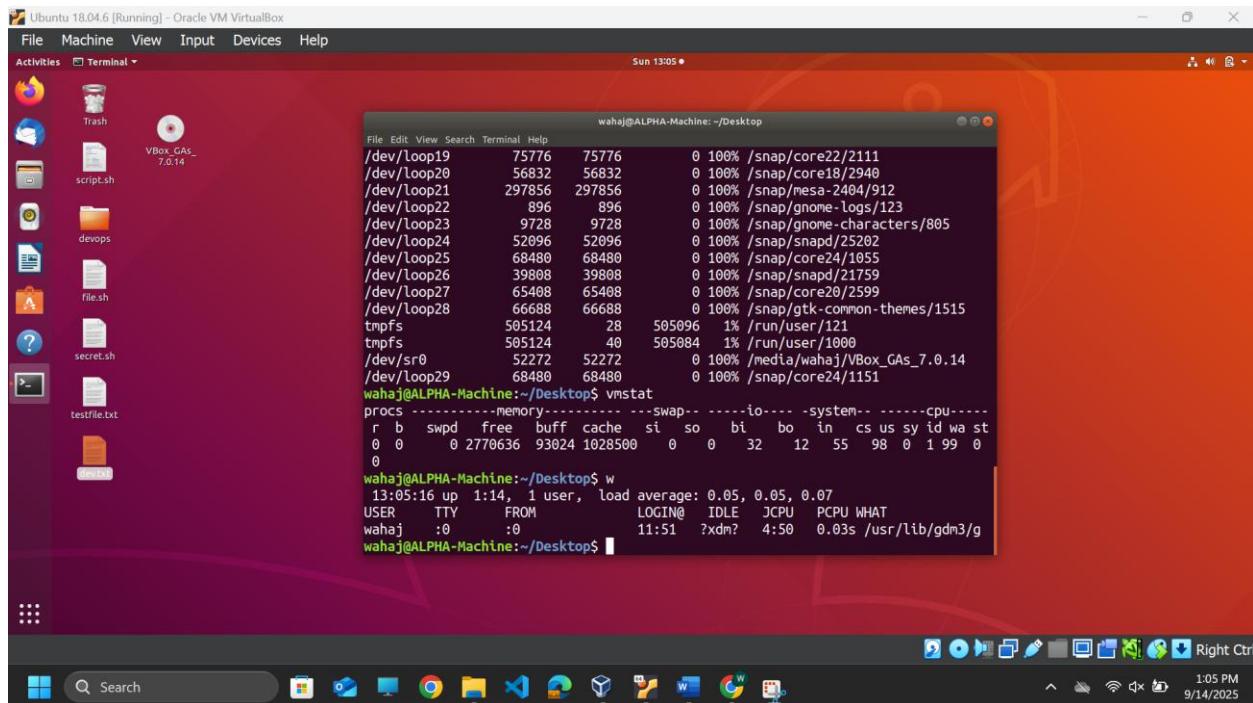
df Show disk space usage for mounted filesystem



vmstat



w Show who is logged in and what they are doing



File Compression and Archiving

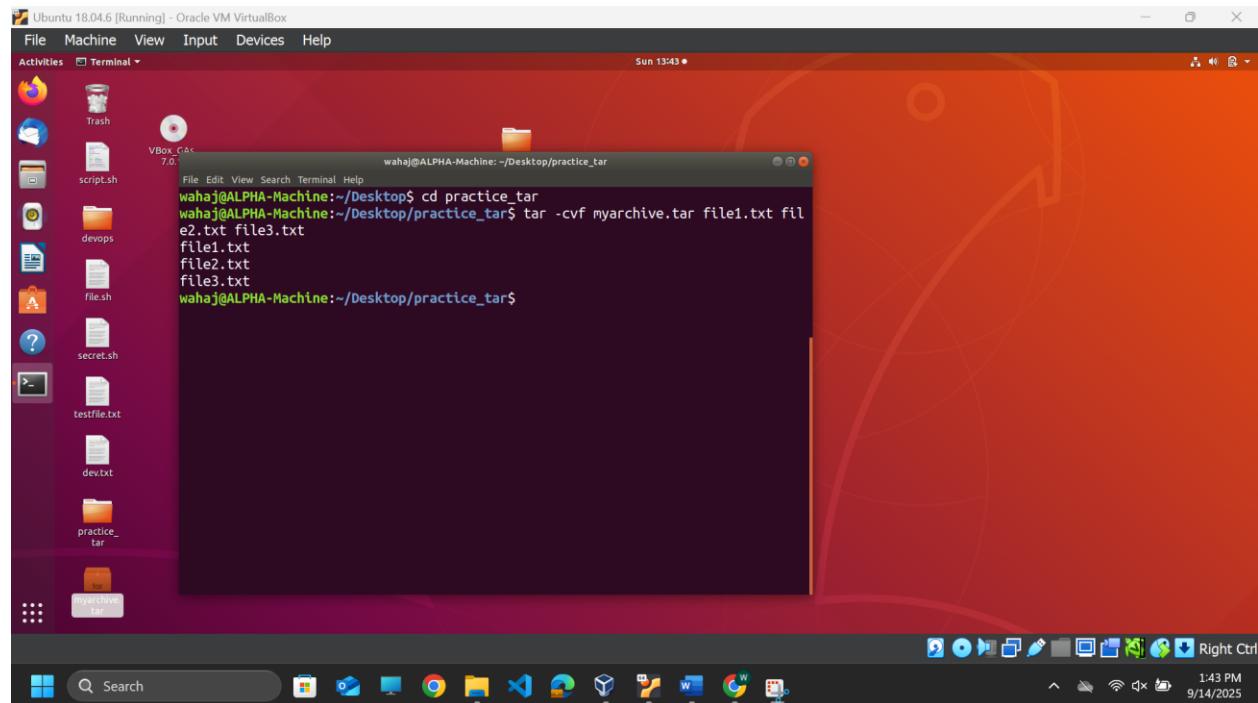
2. Create a tar archive(must navigate to the folder to create tar)

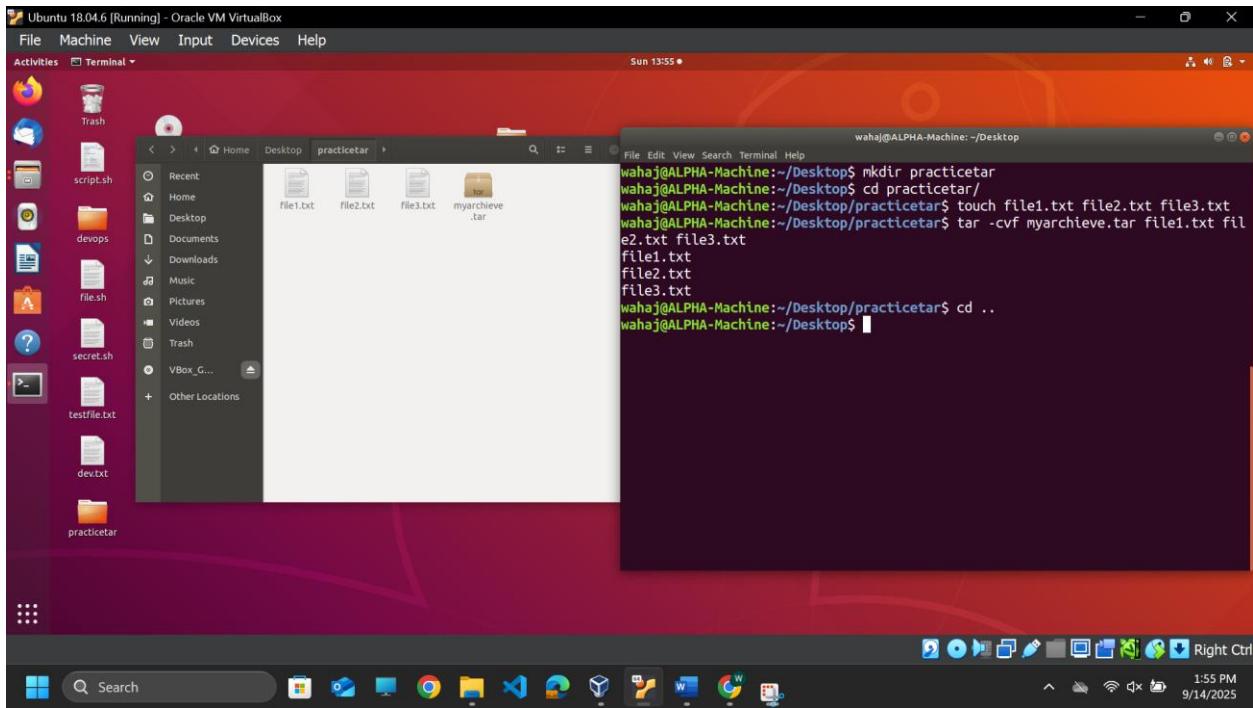
```
tar -cvf myarchive.tar file1.txt file2.txt file3.txt
```

- c → create
- v → verbose (shows files being added)
- f → filename of the archive
- Result: myarchive.tar contains the 3 files.

Check:

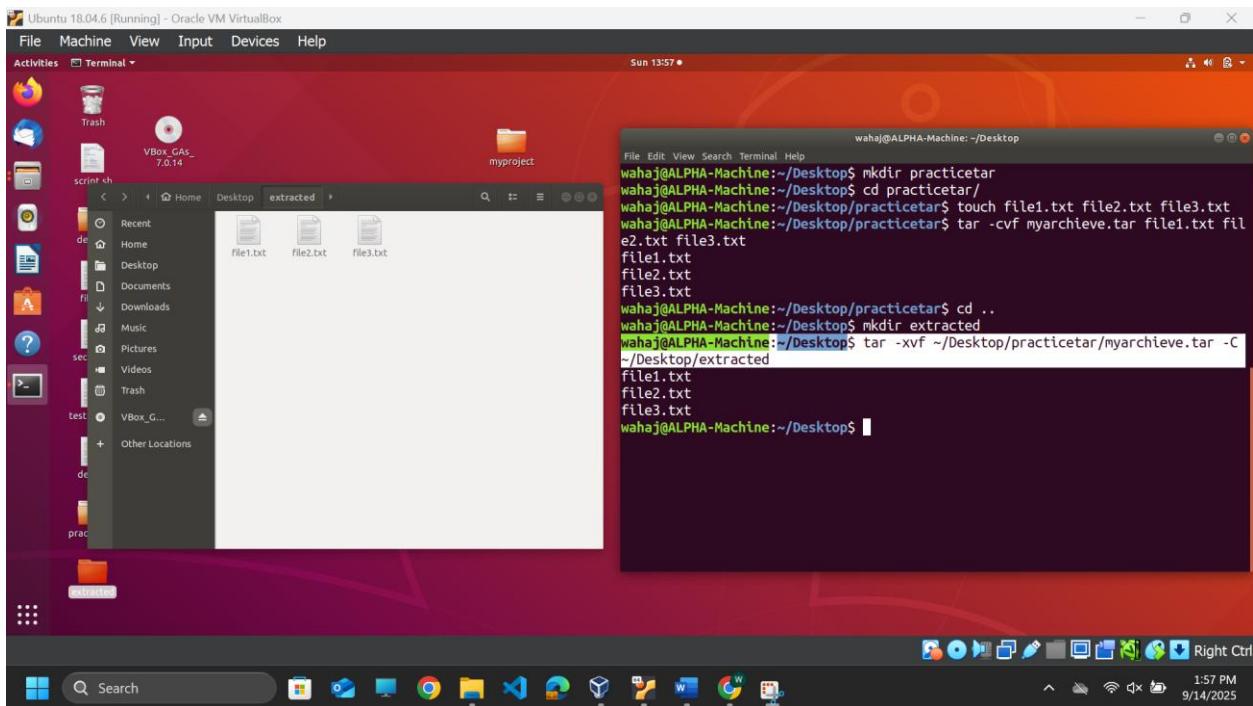
```
wahaj@ALPHA-Machine:~/Desktop/practicetar$ tar -cvf myarchieve.tar  
file1.txt file2.txt file3.txt
```



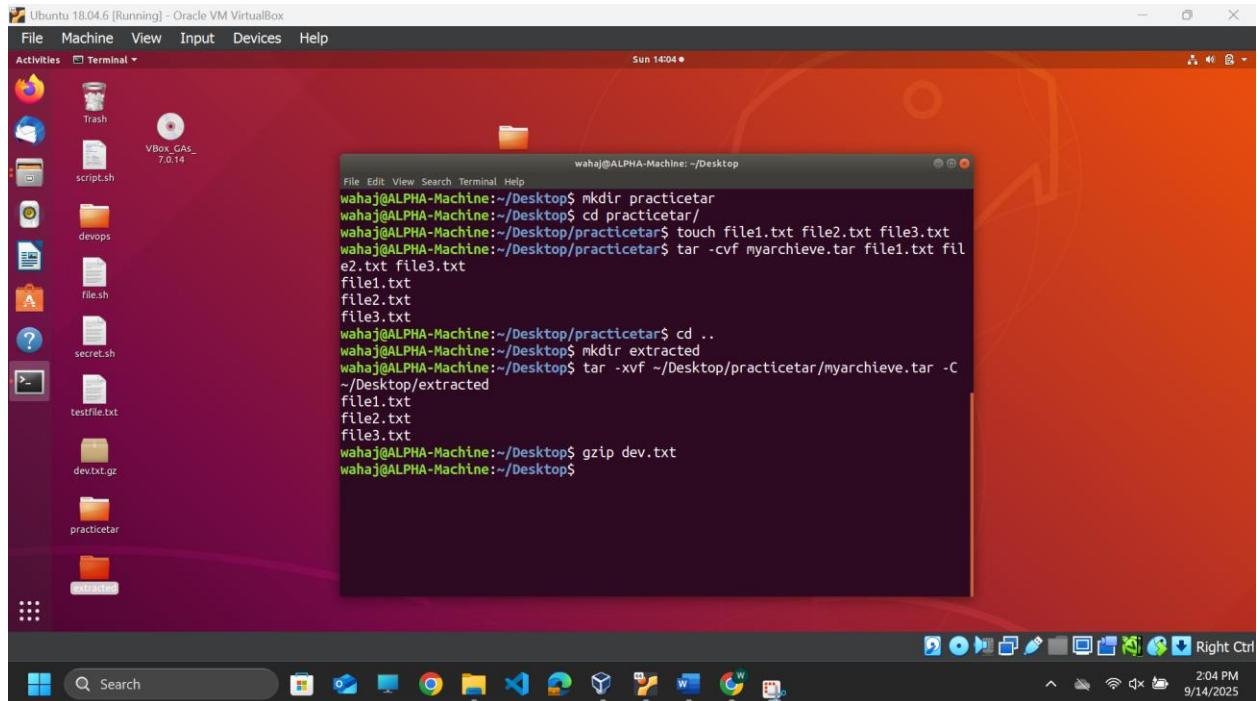


untar

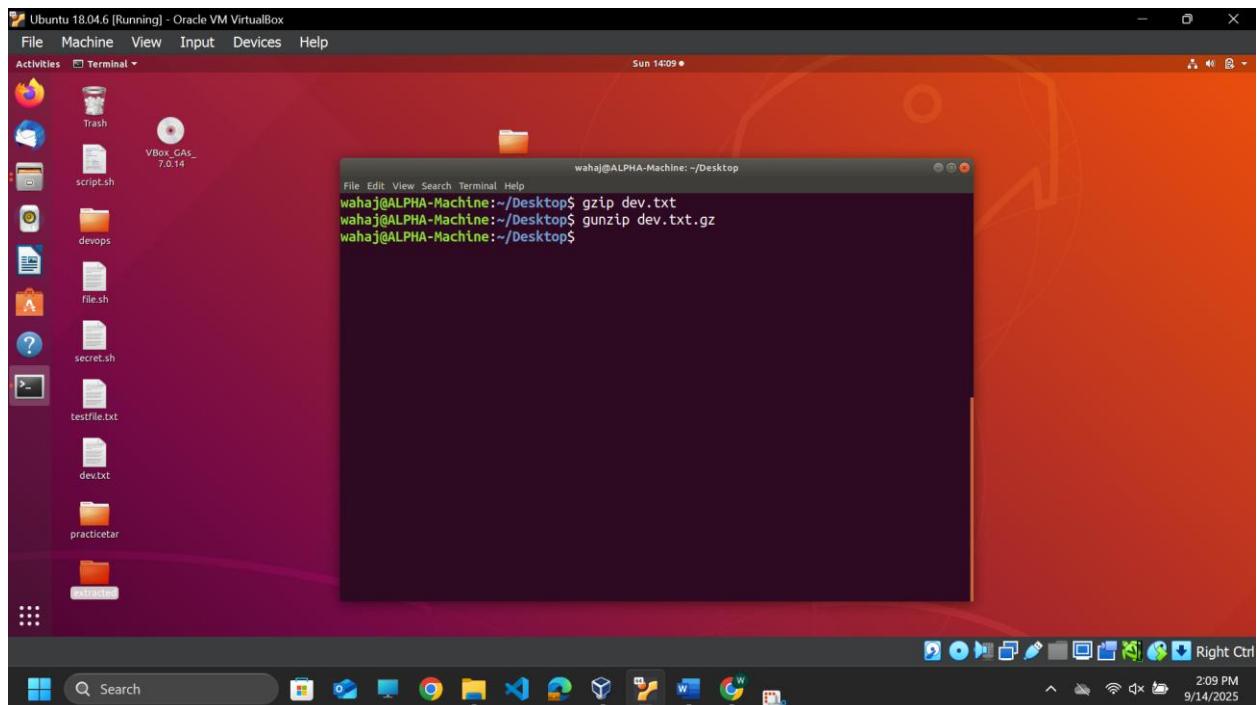
```
wahaj@ALPHA-Machine:~/Desktop$ tar -xvf
~/Desktop/practicetar/myarchive.tar -C ~/Desktop/extracted
```



gzip dev.txt (compress file)



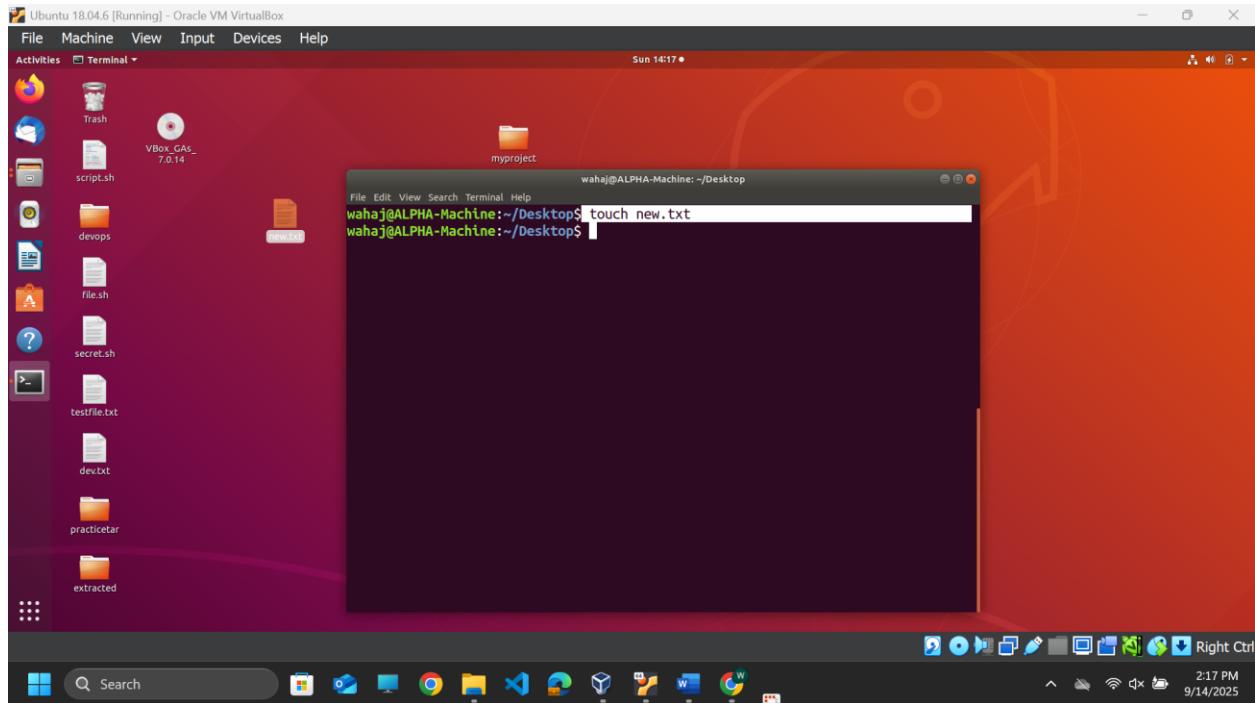
gunzip dev.txt.gz



CRUD Operations

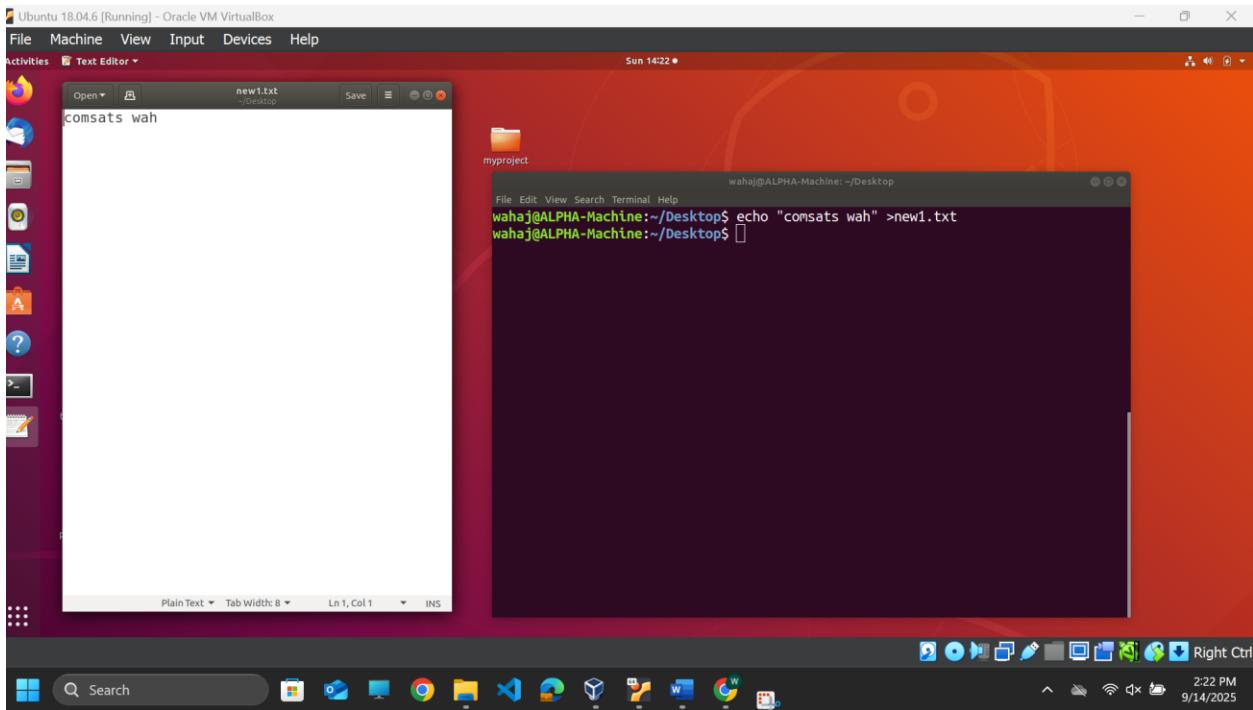
1- Create a File

touch new.txt

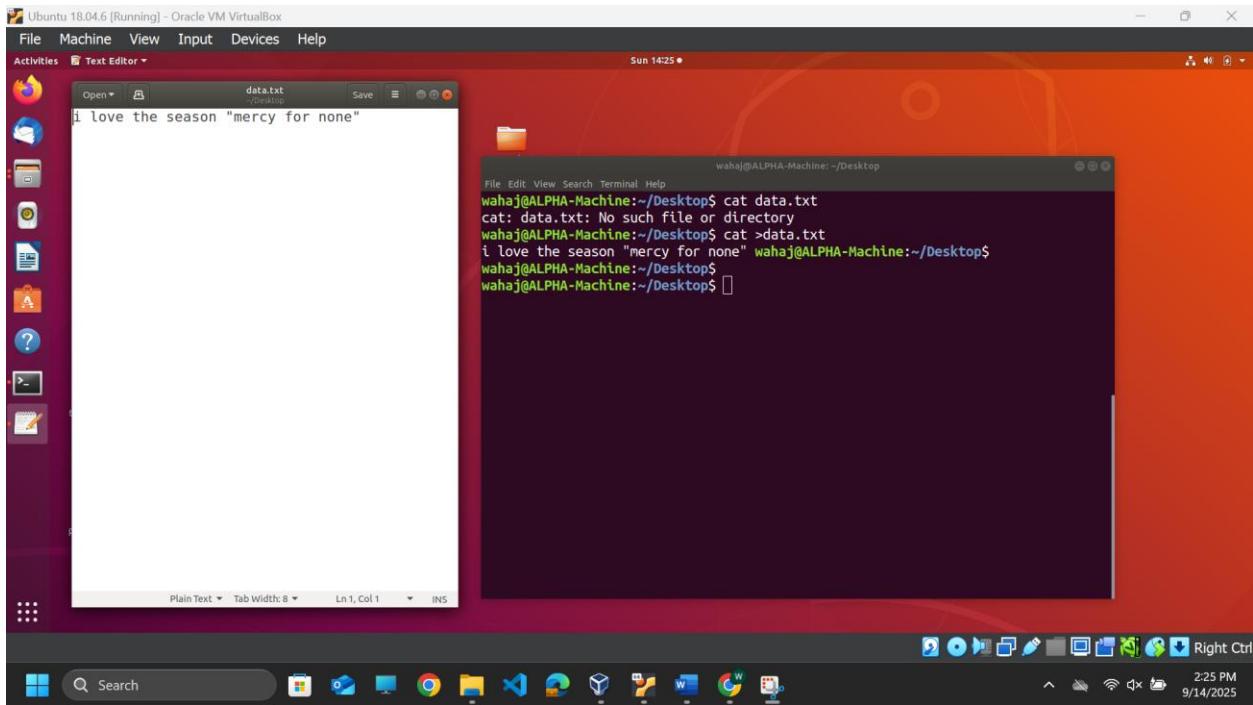


Create a File with Content

echo "comsats wah" >new1.txt

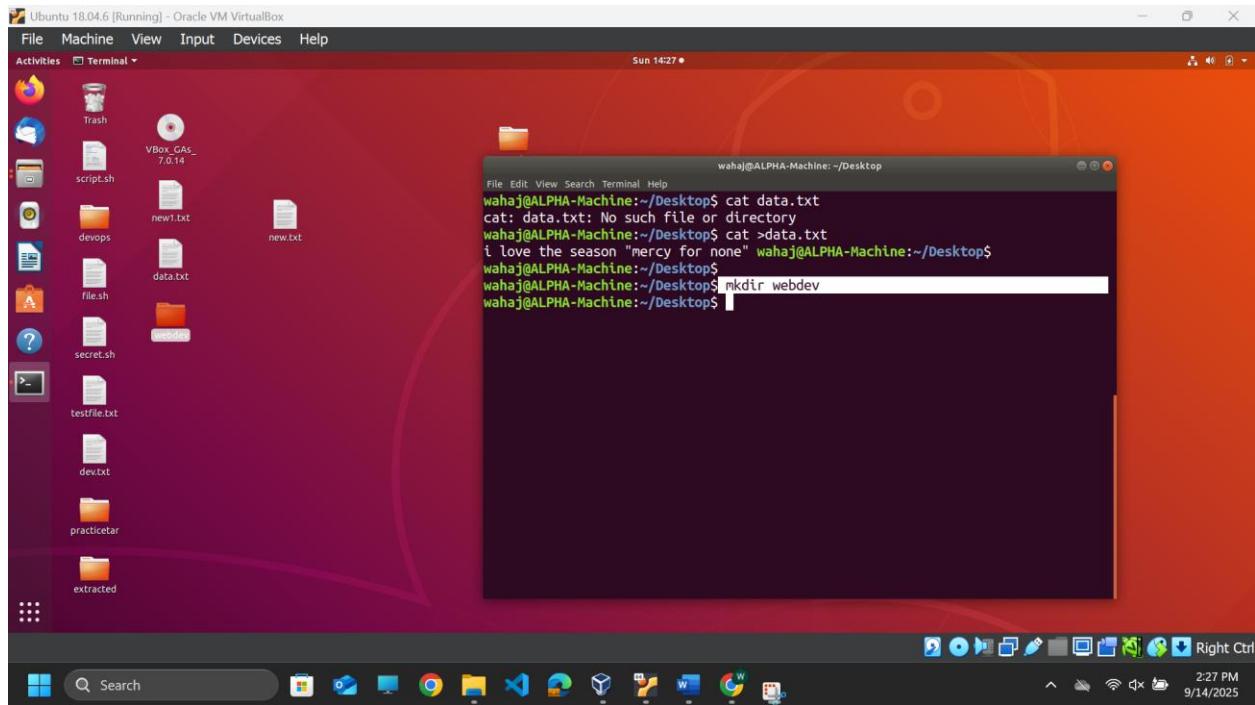


cat > data.txt (Press Ctrl+D to save)



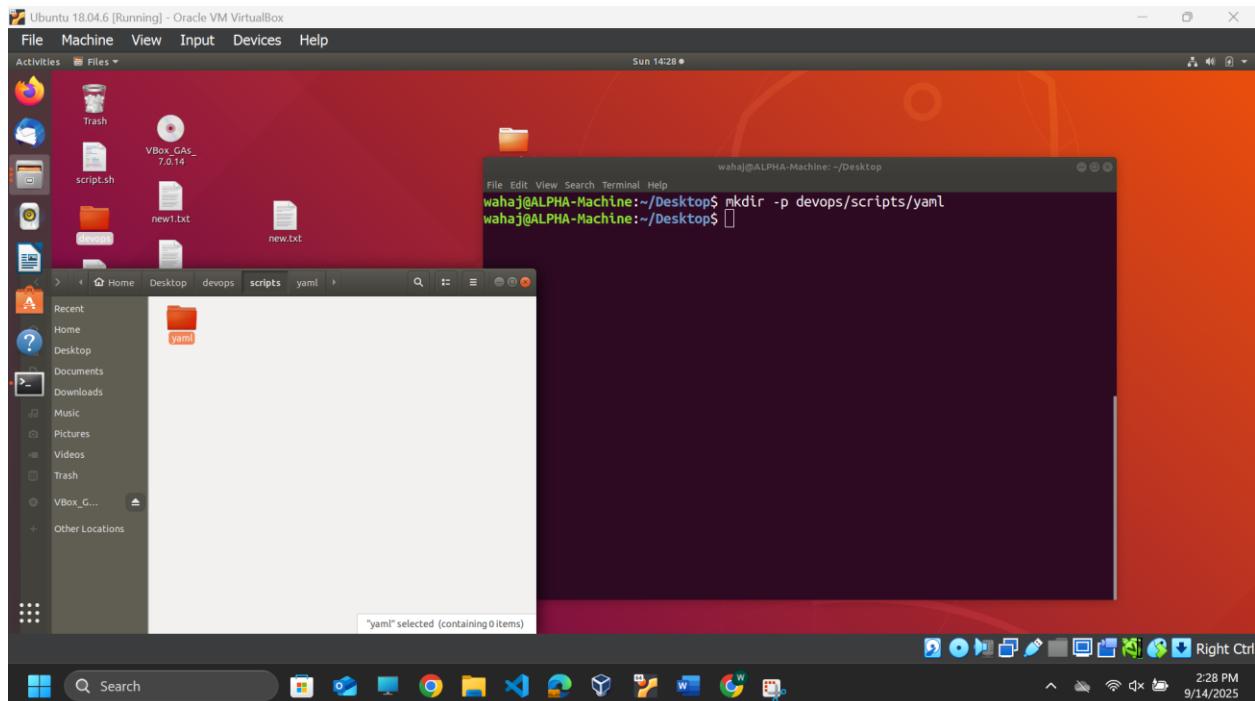
Create a Directory (Folder)

`mkdir webdev`



Create Nested Directories

`mkdir -p devops/scripts/yaml`



Example without -p:

`mkdir devops/scripts/yaml`

This will fail if devops or scripts don't already exist.

Example with -p:

`mkdir -p devops/scripts/yaml`

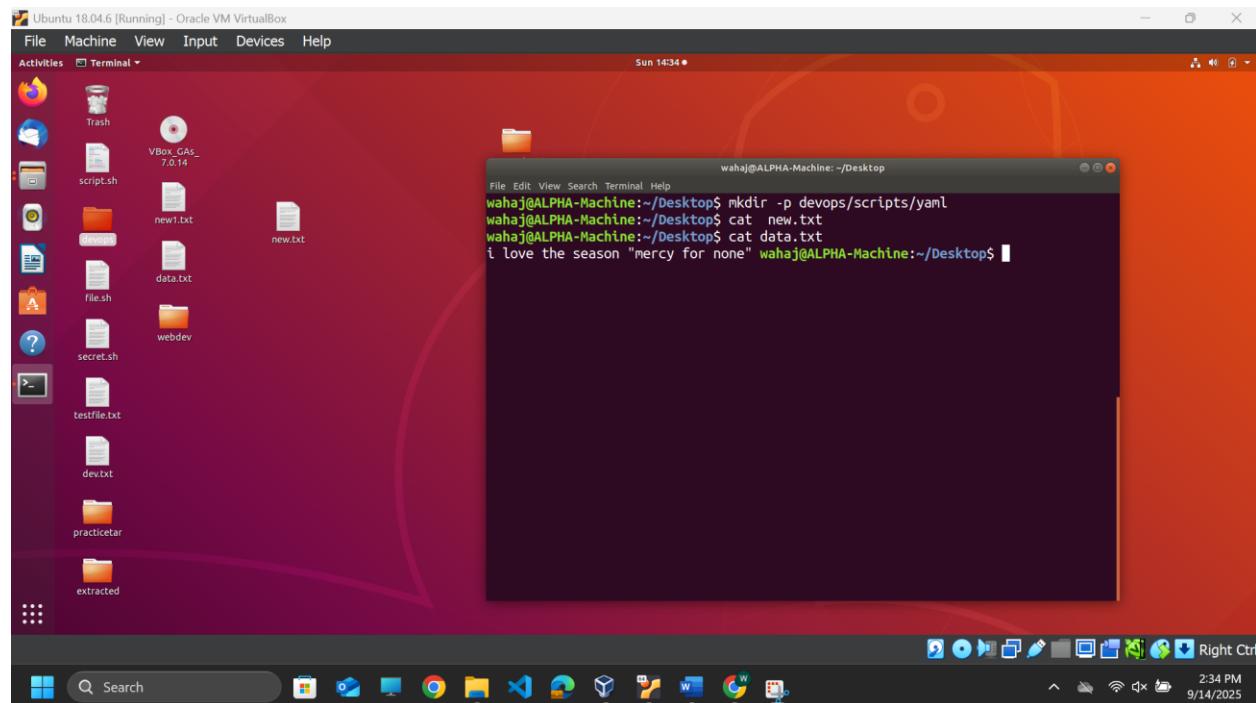
This will automatically create the whole path (devops, then scripts, then yaml) even if none of them exist before.

Also, if the directory already exists, it won't give an error.

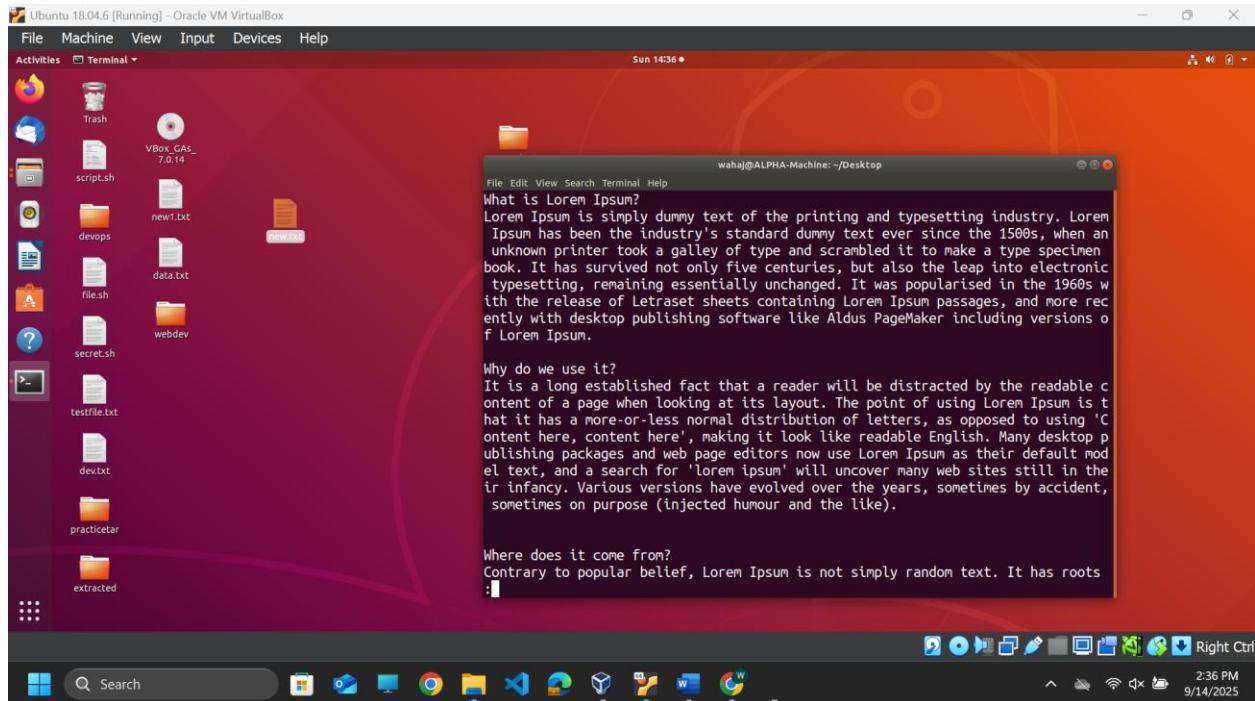
2- R – READ Operations in Linux

View File Content

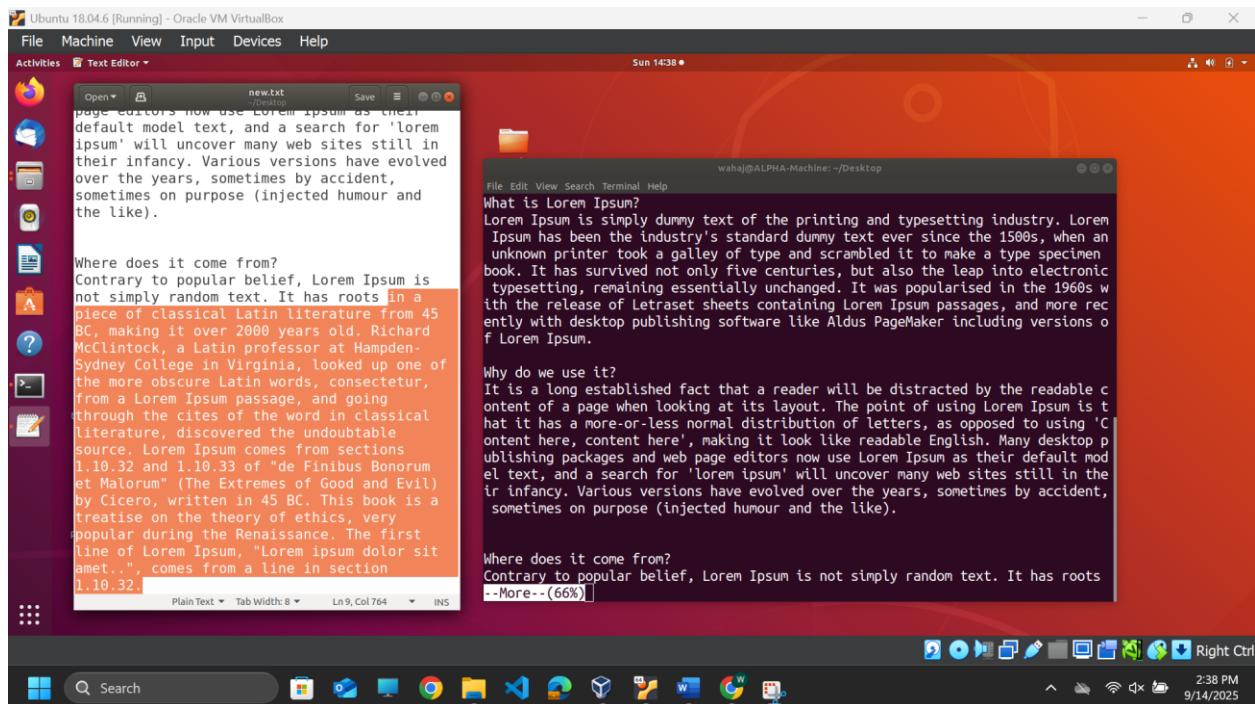
cat filename.txt



less filename.txt # Scrollable in terminal



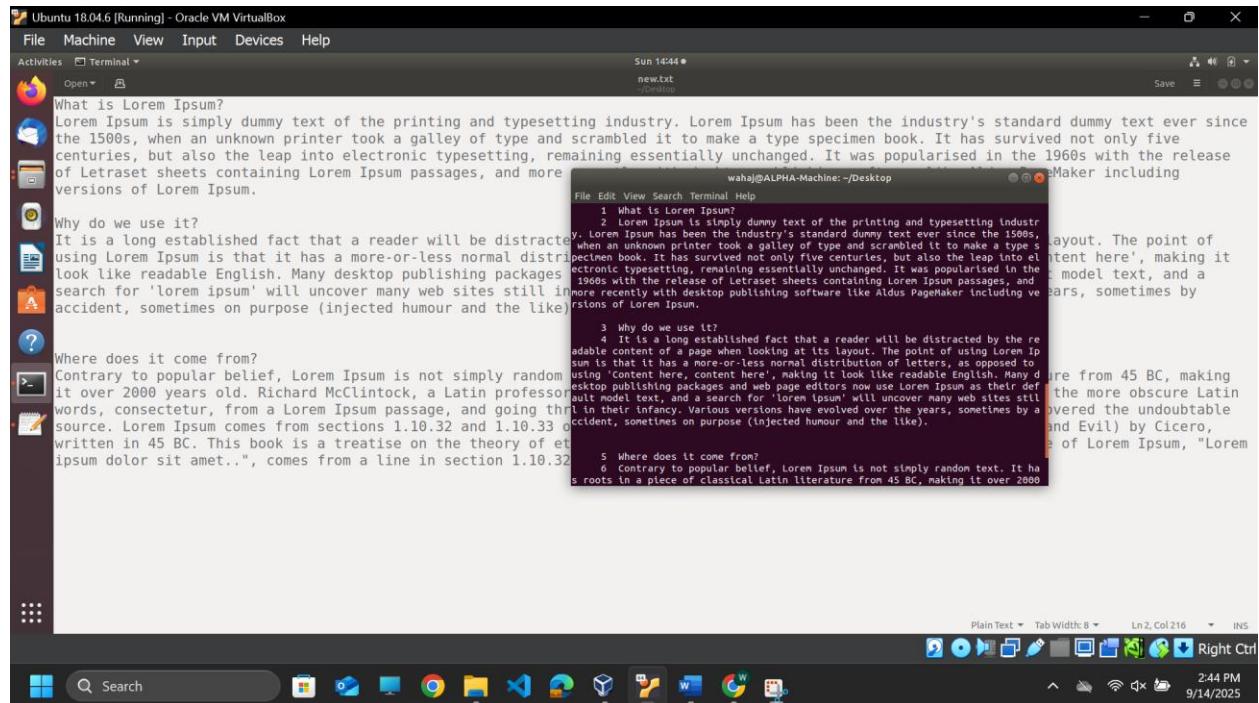
more filename.txt # Page-wise output (only show page 1)



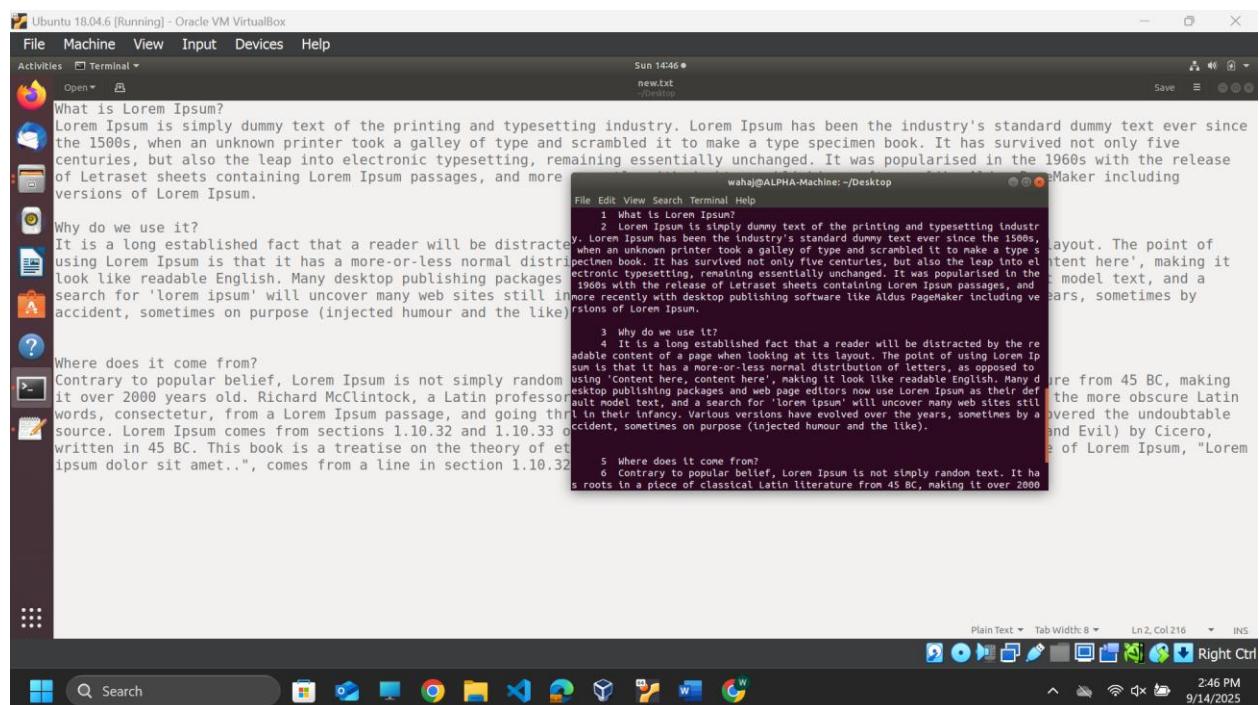
not shown the text after page 1

View First or Last Lines

head filename.txt # First 10 lines (must see how it read first 10 lines)



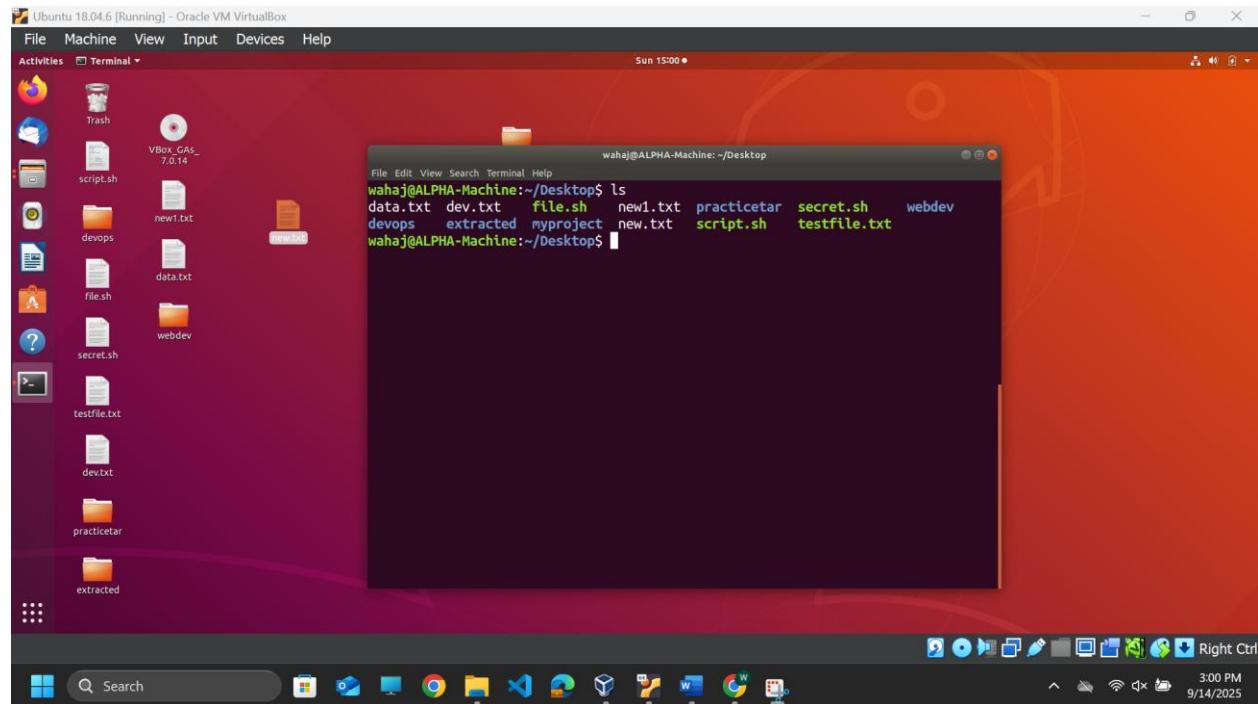
tail filename.txt # Last 10 lines(must see how it read last10 lines)



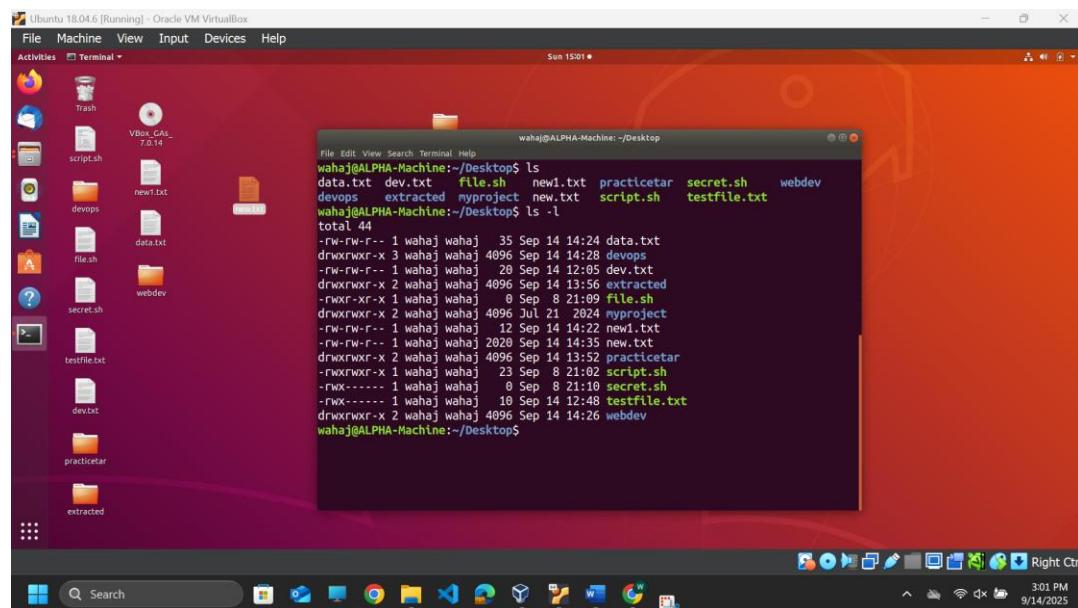
or head new.txt |nl **or** tail new.txt|nl **or** (nl means print line number along with lines)

Read a Directory

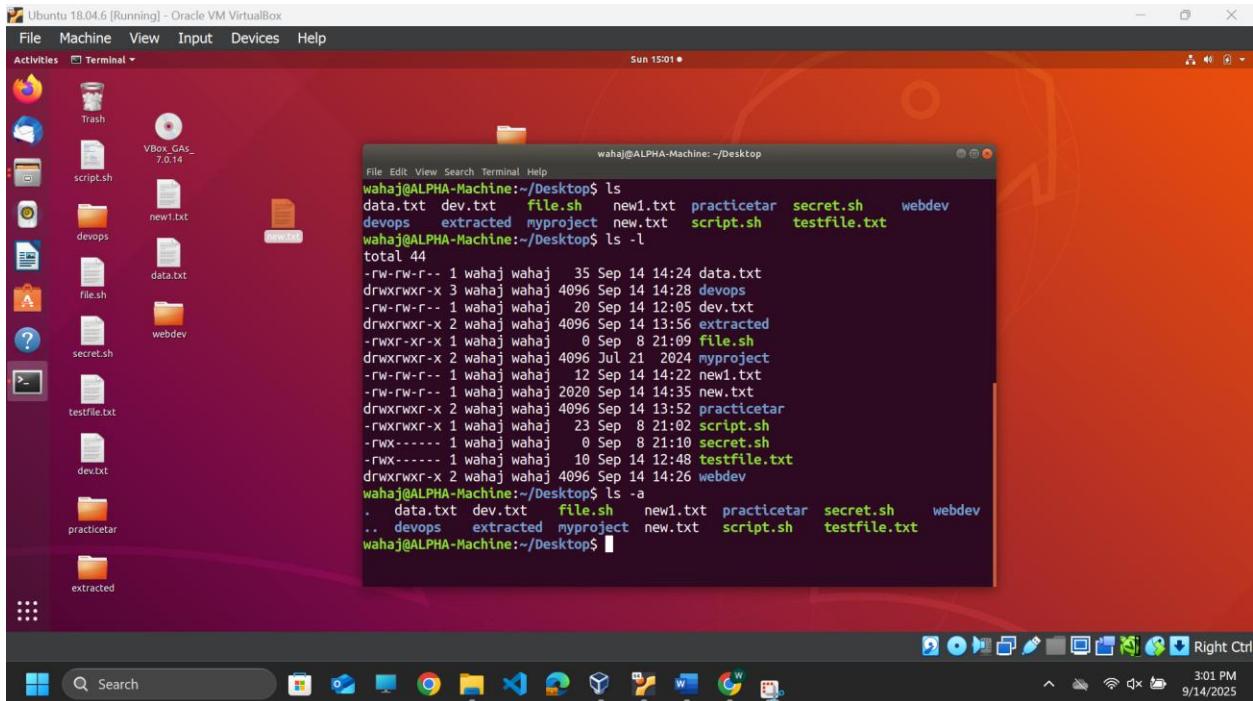
ls # List files



ls -l # Detailed view



ls -a # Show hidden files

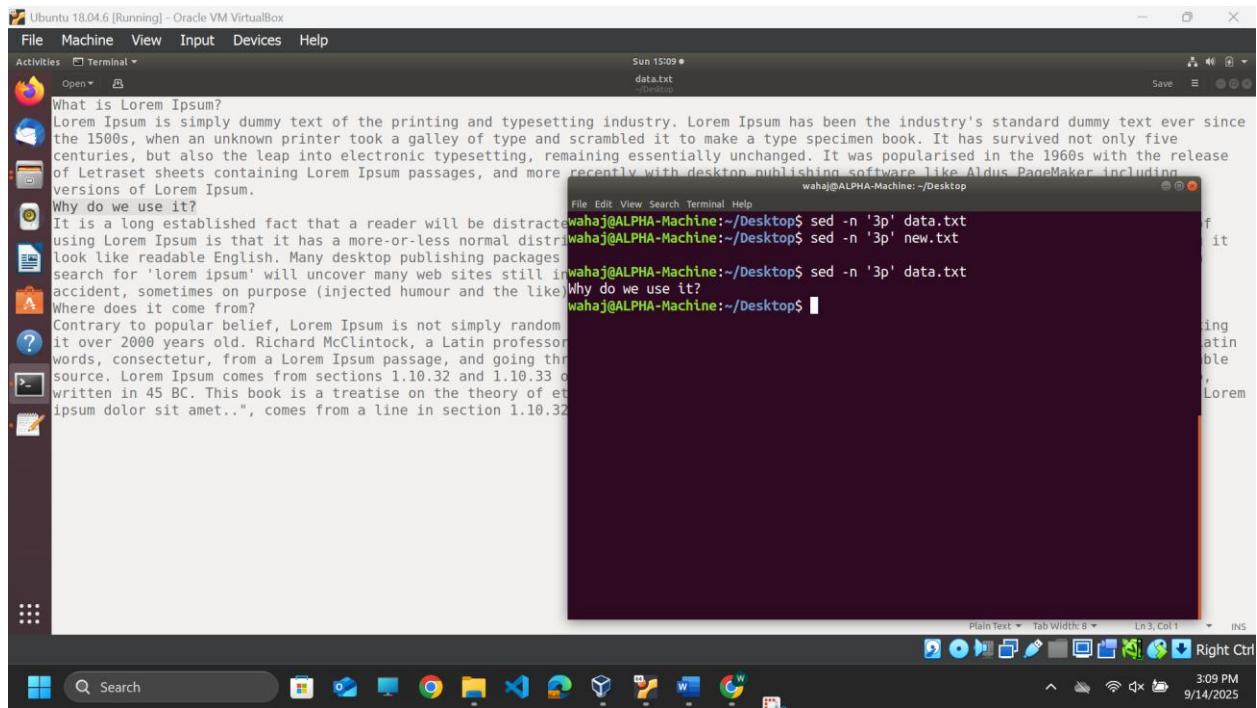


Read a Specific Line (Using sed or awk)

sed -n '3p' filename.txt # Show 3rd line

- **sed** → stream editor (used for editing text in a stream/file).
- **-n** → suppresses (hides) the default output. Normally sed prints the whole file.
- **'3p'** → means:
 - 3 = line number 3
 - p = print

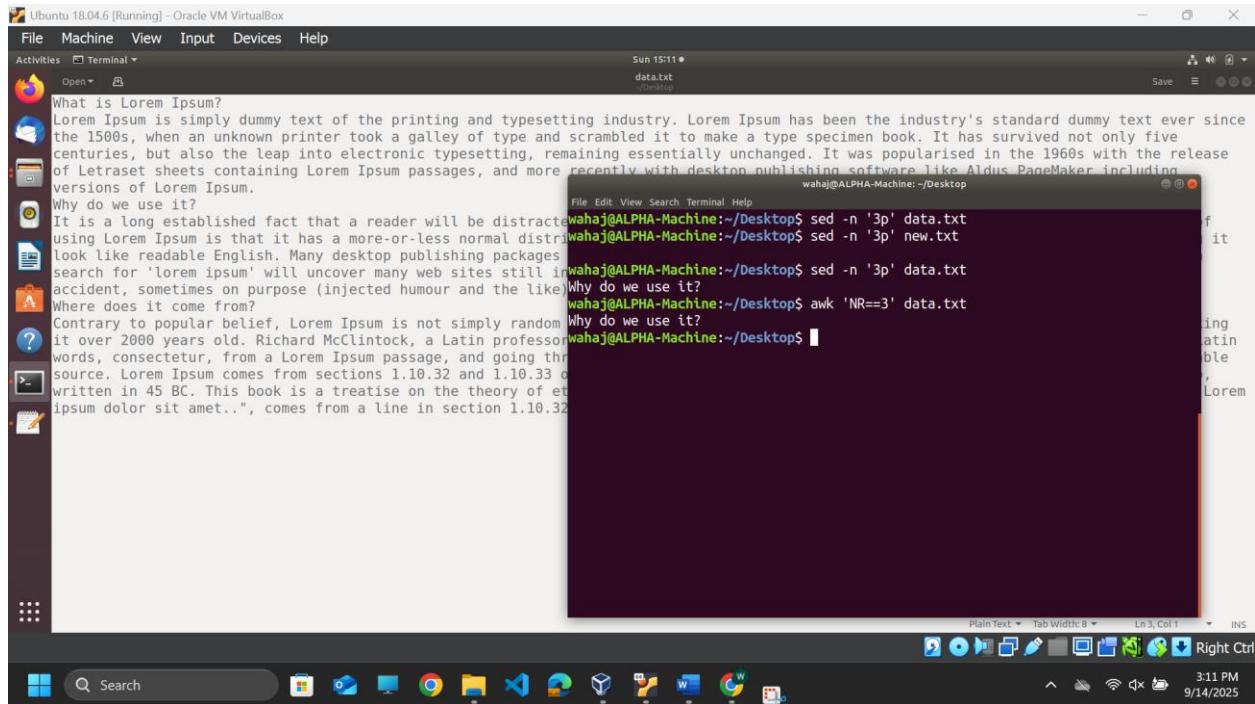
So together → **print only the 3rd line of the file.**



```
awk 'NR==3' filename.txt # Another method
```

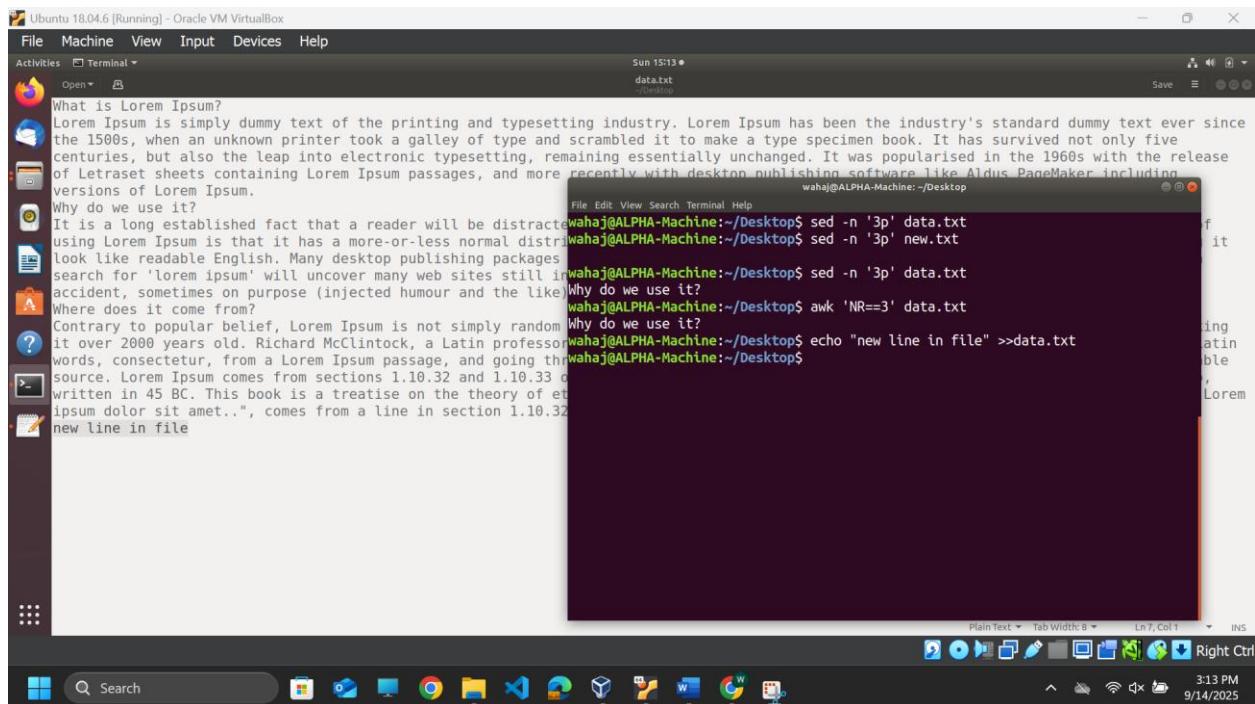
Explanation:

- awk → a text processing tool, works line by line.
- NR → *Number of Record* → basically the current line number.
- NR==3 → means *when the line number is 3*.
- Action (by default) → print that line.



3-U – UPDATE Operations in Linux

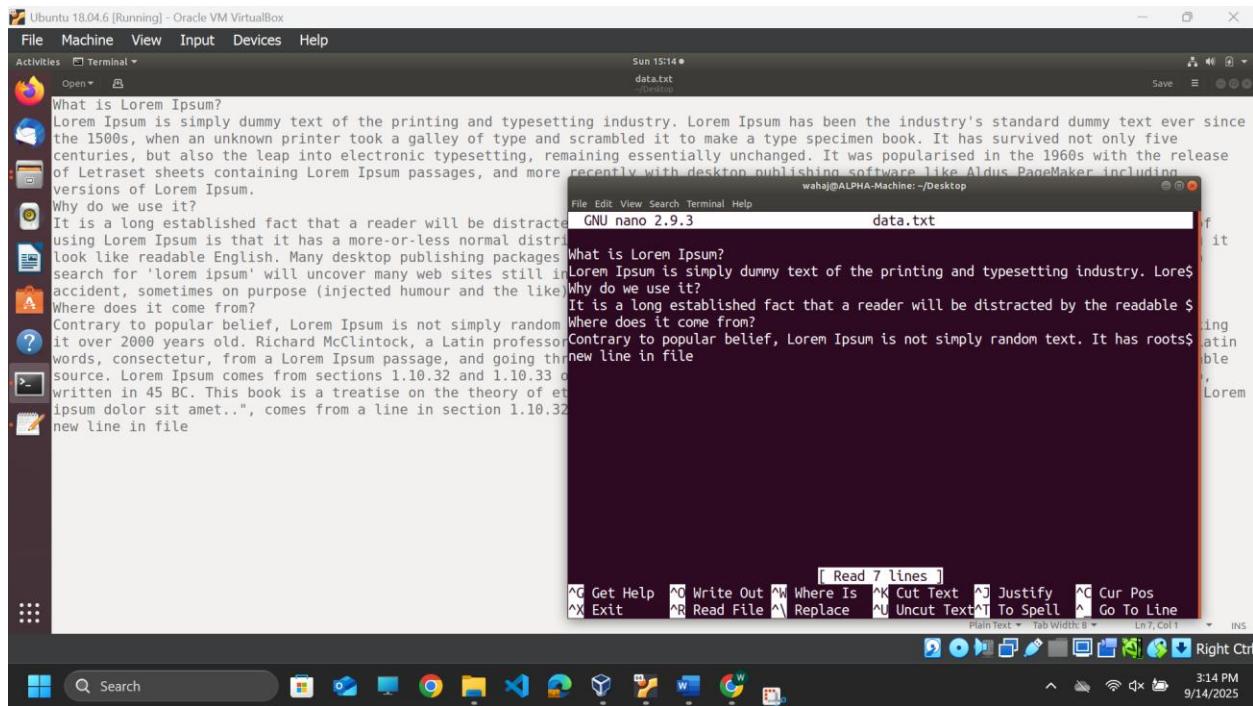
echo "new line in file" >>data.txt add new line in file(append)



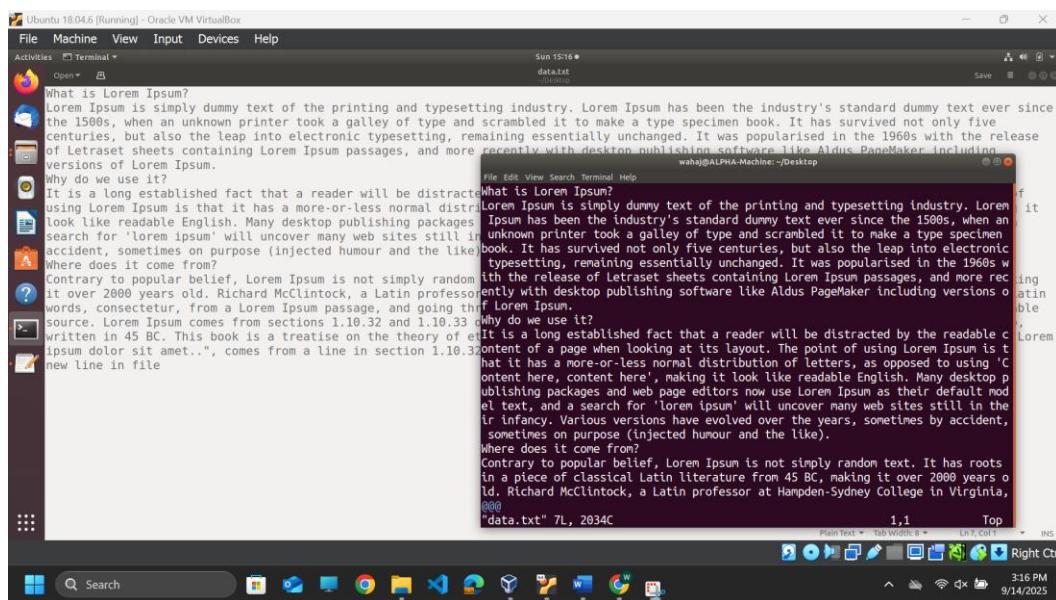
Edit File (Using Editors)

nano data.txt

beginner-friendly terminal editor

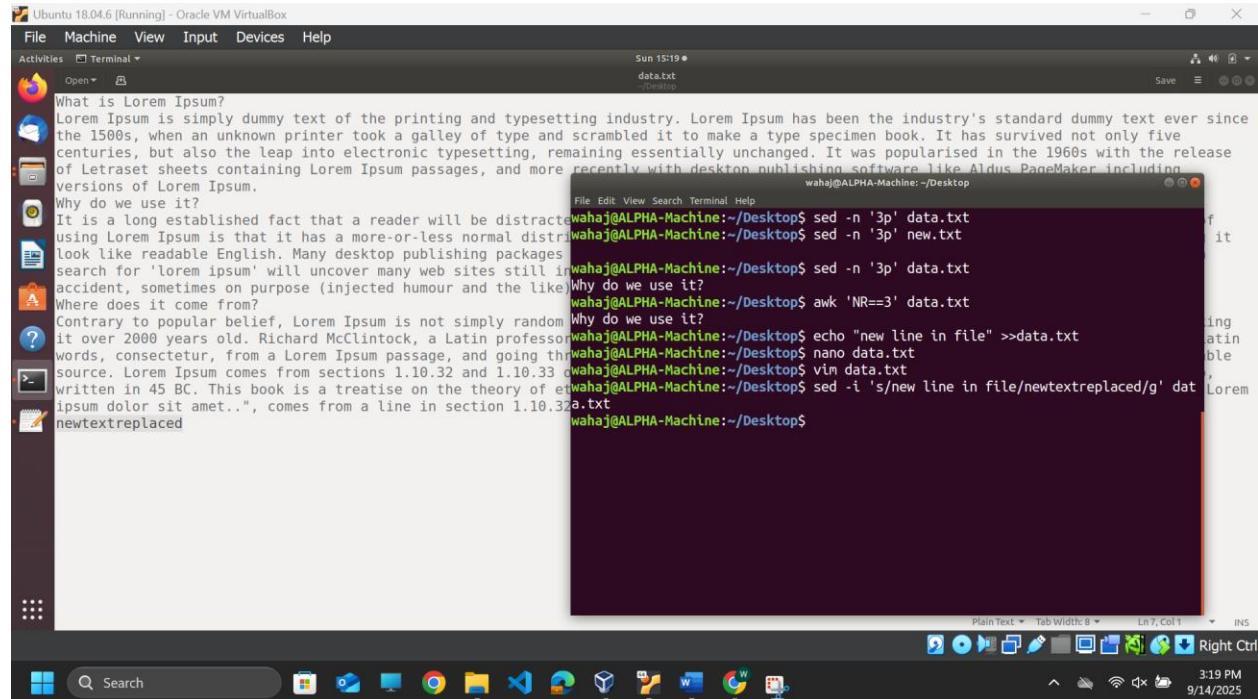


vim filename.txt – powerful but advanced editor



Replace Text in File

sed -i 's/oldtext/newtext/g' filename.txt

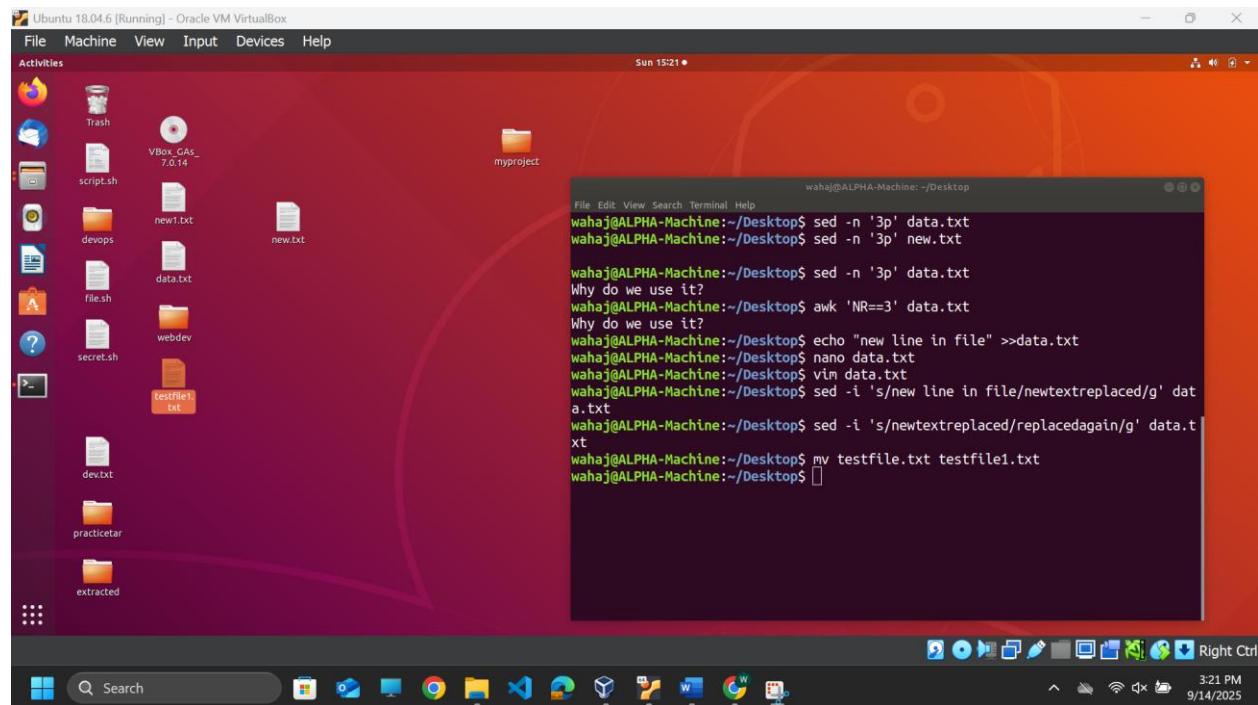


The screenshot shows a desktop environment with a terminal window open. The terminal window title is "data.txt" and it contains the following command history:

```
waha@ALPHA-Machine:~/Desktop$ sed -n '3p' data.txt
waha@ALPHA-Machine:~/Desktop$ sed -n '3p' new.txt
waha@ALPHA-Machine:~/Desktop$ sed -n '3p' data.txt
waha@ALPHA-Machine:~/Desktop$ awk 'NR==3' data.txt
waha@ALPHA-Machine:~/Desktop$ echo "new line in file" >>data.txt
waha@ALPHA-Machine:~/Desktop$ nano data.txt
waha@ALPHA-Machine:~/Desktop$ vim data.txt
waha@ALPHA-Machine:~/Desktop$ sed -i 's/new line in file/newtextreplaced/g' data.txt
waha@ALPHA-Machine:~/Desktop$
```

Rename File or Folder

mv testfile.txt testfile1.txt

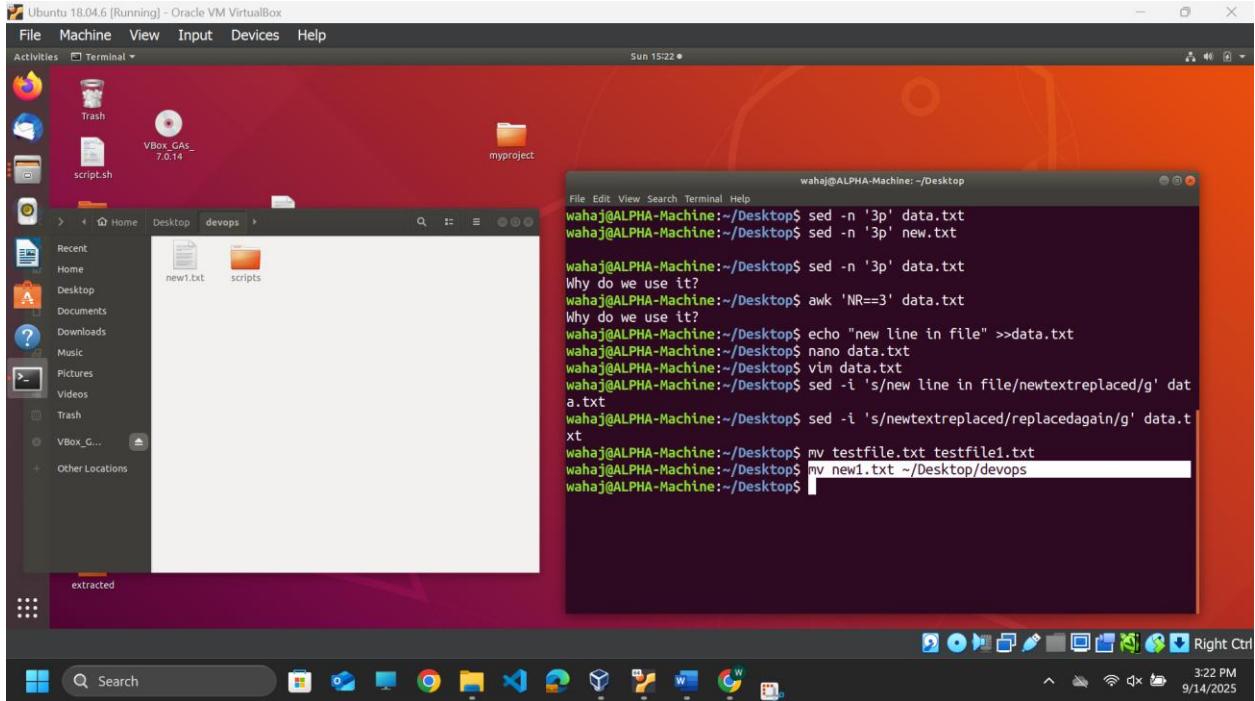


The screenshot shows a desktop environment with a terminal window open. The terminal window title is "waha@ALPHA-Machine:~/Desktop" and it contains the following command history:

```
waha@ALPHA-Machine:~/Desktop$ sed -n '3p' data.txt
waha@ALPHA-Machine:~/Desktop$ sed -n '3p' new.txt
waha@ALPHA-Machine:~/Desktop$ sed -n '3p' data.txt
waha@ALPHA-Machine:~/Desktop$ awk 'NR==3' data.txt
waha@ALPHA-Machine:~/Desktop$ echo "new line in file" >>data.txt
waha@ALPHA-Machine:~/Desktop$ nano data.txt
waha@ALPHA-Machine:~/Desktop$ vim data.txt
waha@ALPHA-Machine:~/Desktop$ sed -i 's/new line in file/newtextreplaced/g' data.txt
waha@ALPHA-Machine:~/Desktop$ sed -i 's/newtextreplaced/replacedagain/g' data.txt
waha@ALPHA-Machine:~/Desktop$ mv testfile.txt testfile1.txt
waha@ALPHA-Machine:~/Desktop$
```

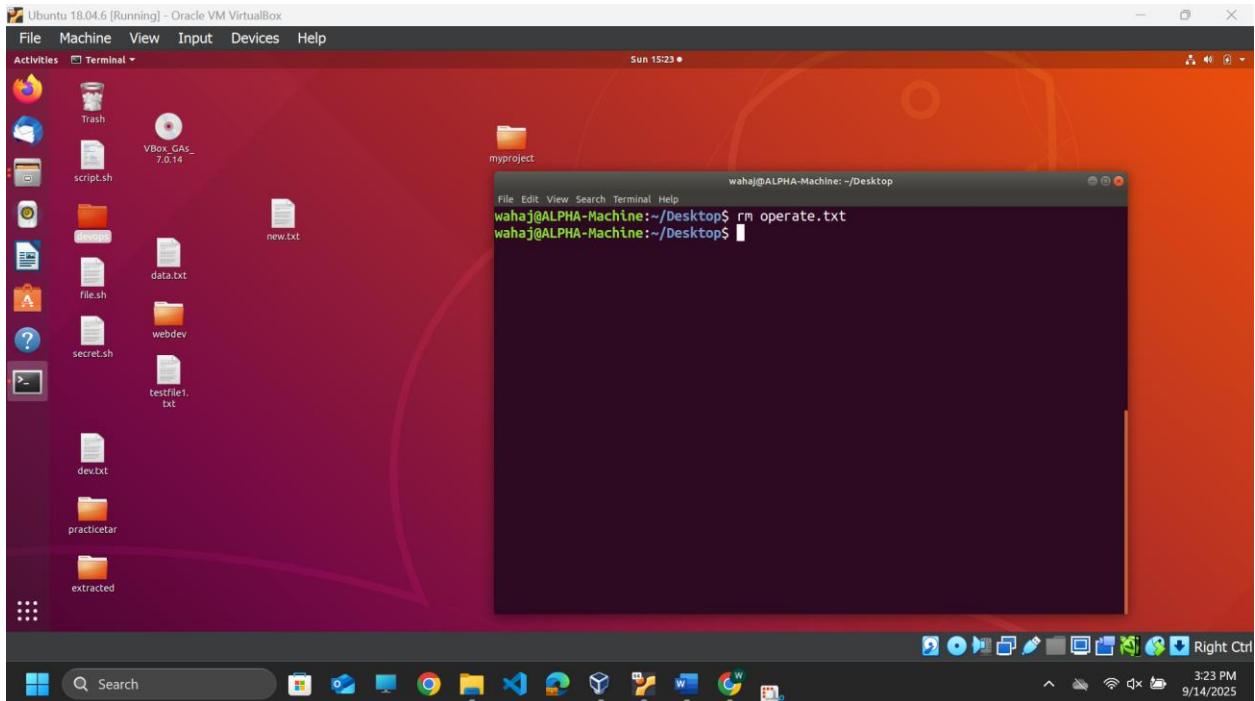
Move File to Different Location

mv new1.txt ~/Desktop/devops



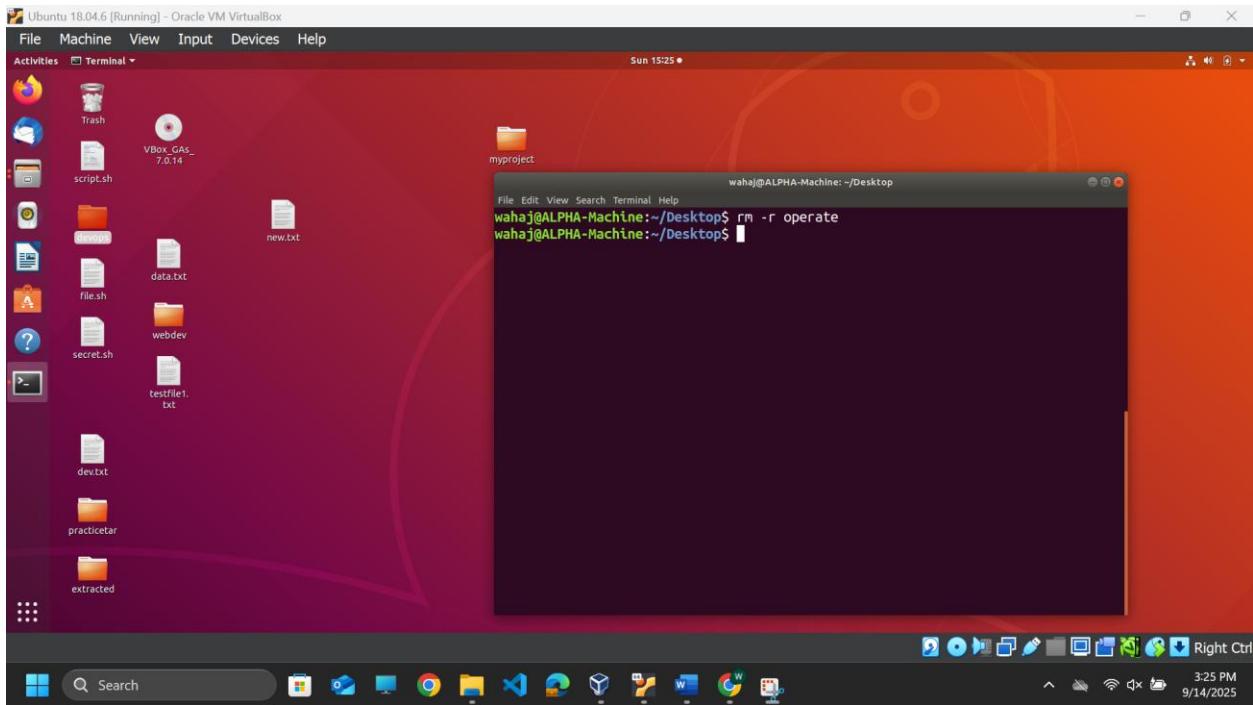
Delete a File

rm filename.txt



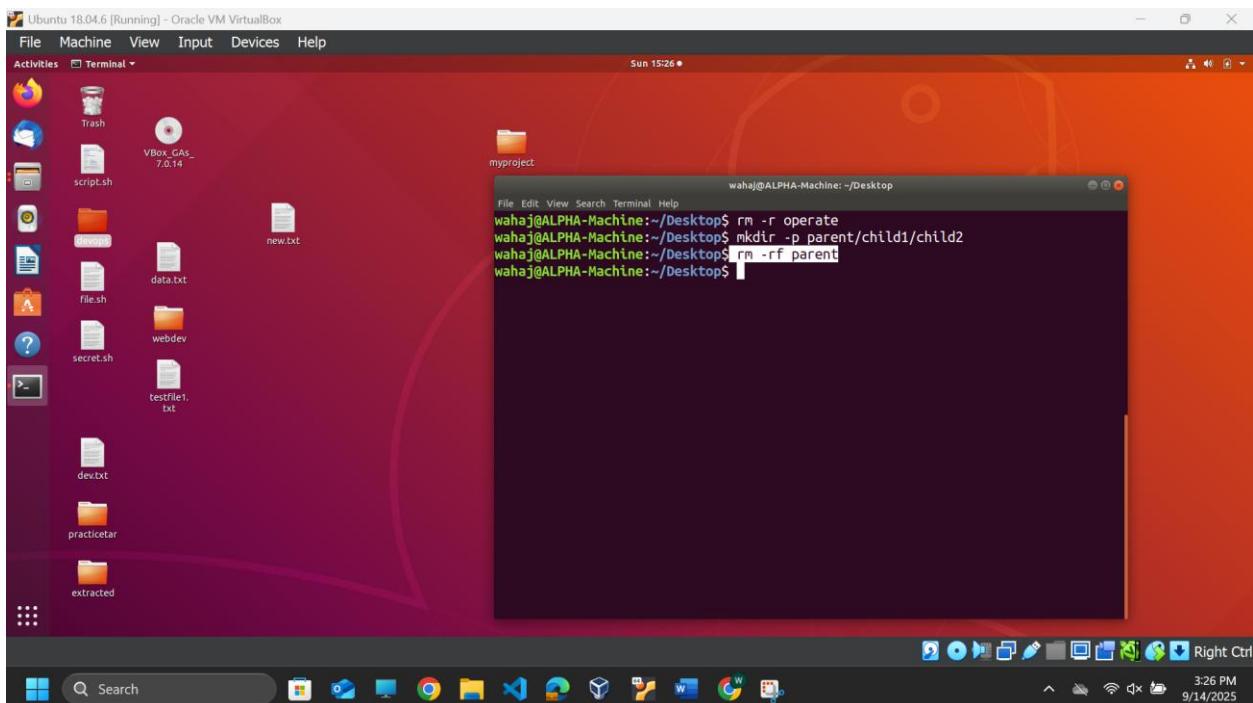
Delete a Directory

rm -r operate



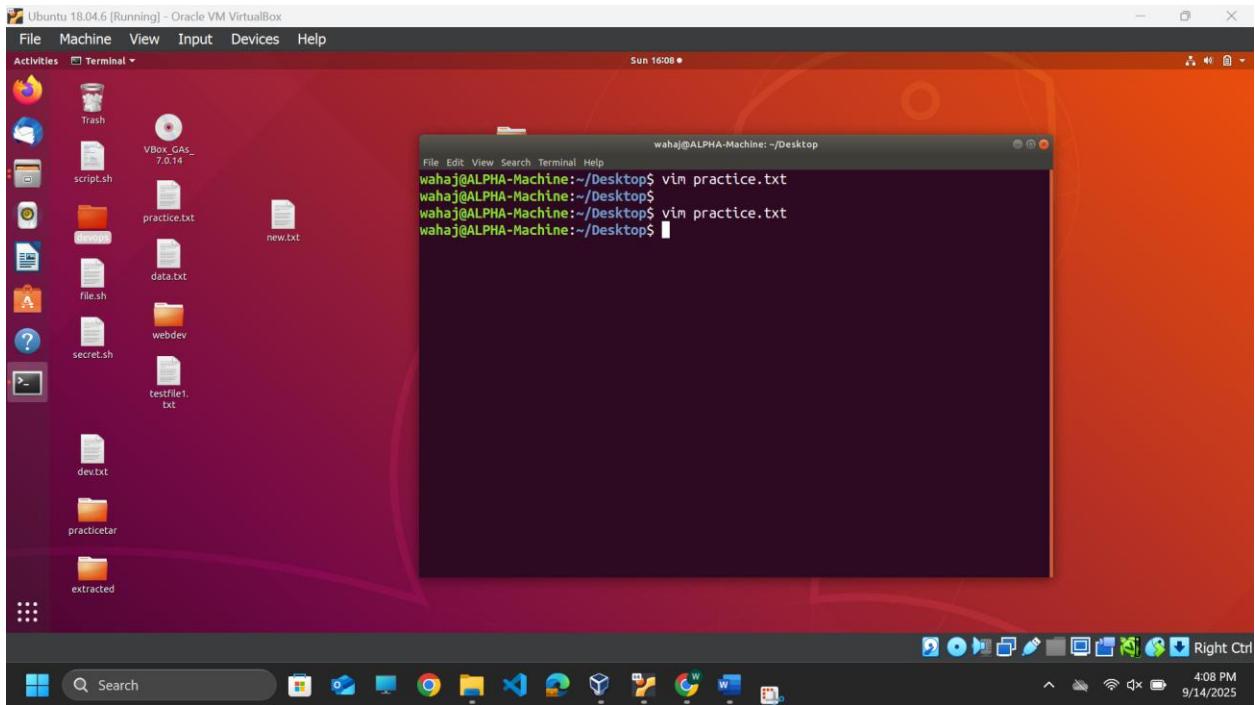
Force Delete

rm -rf foldername



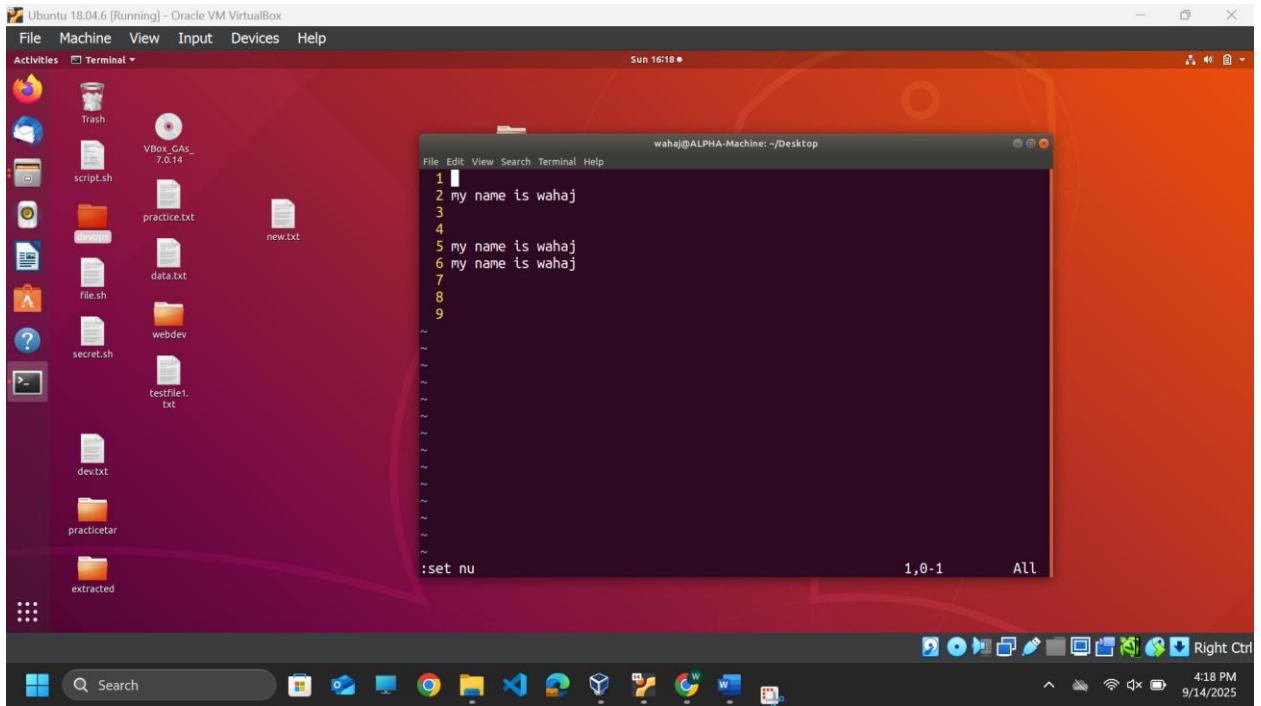
VIM Editor in Linux

vim practice.txt

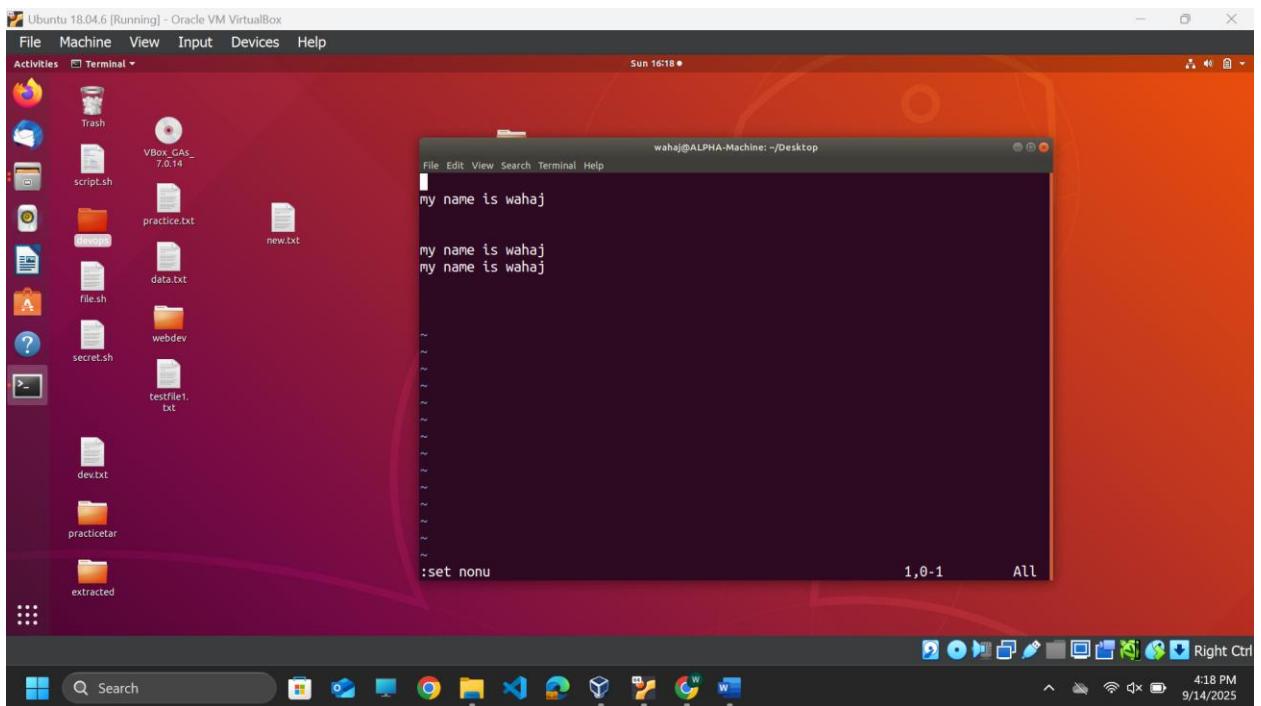


COMMAND-LINE MODE – For Save, Quit, Search

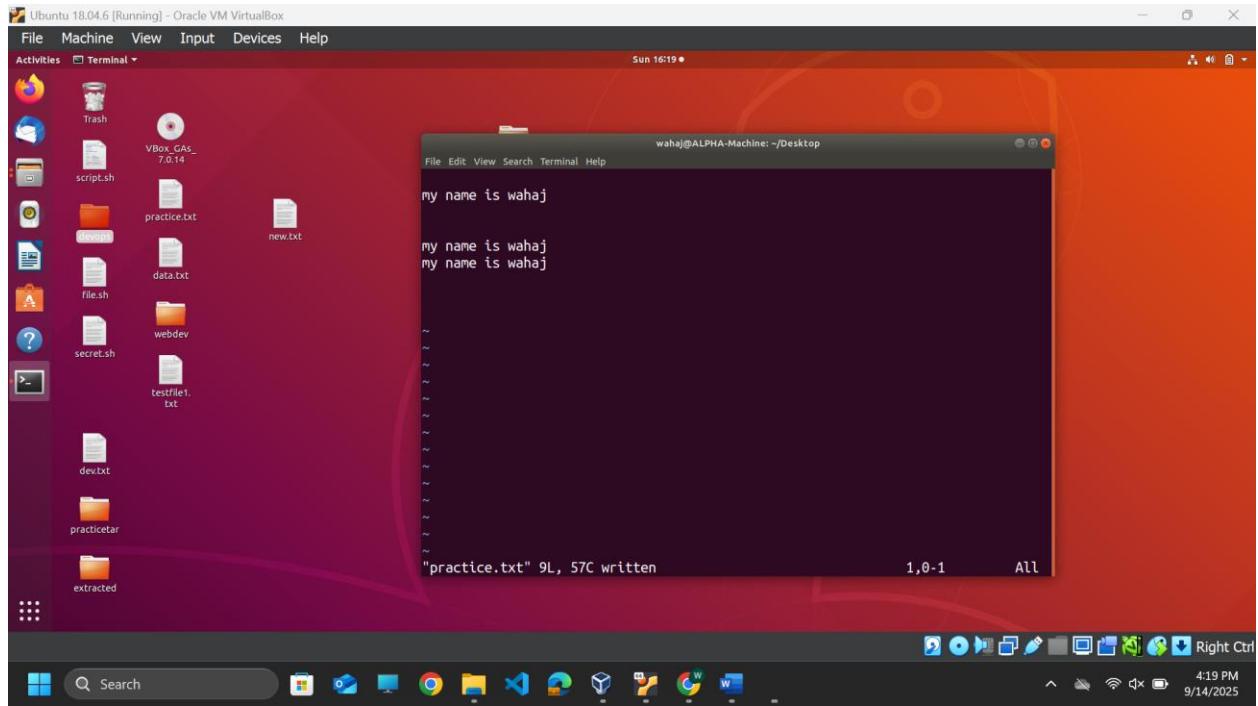
set nu



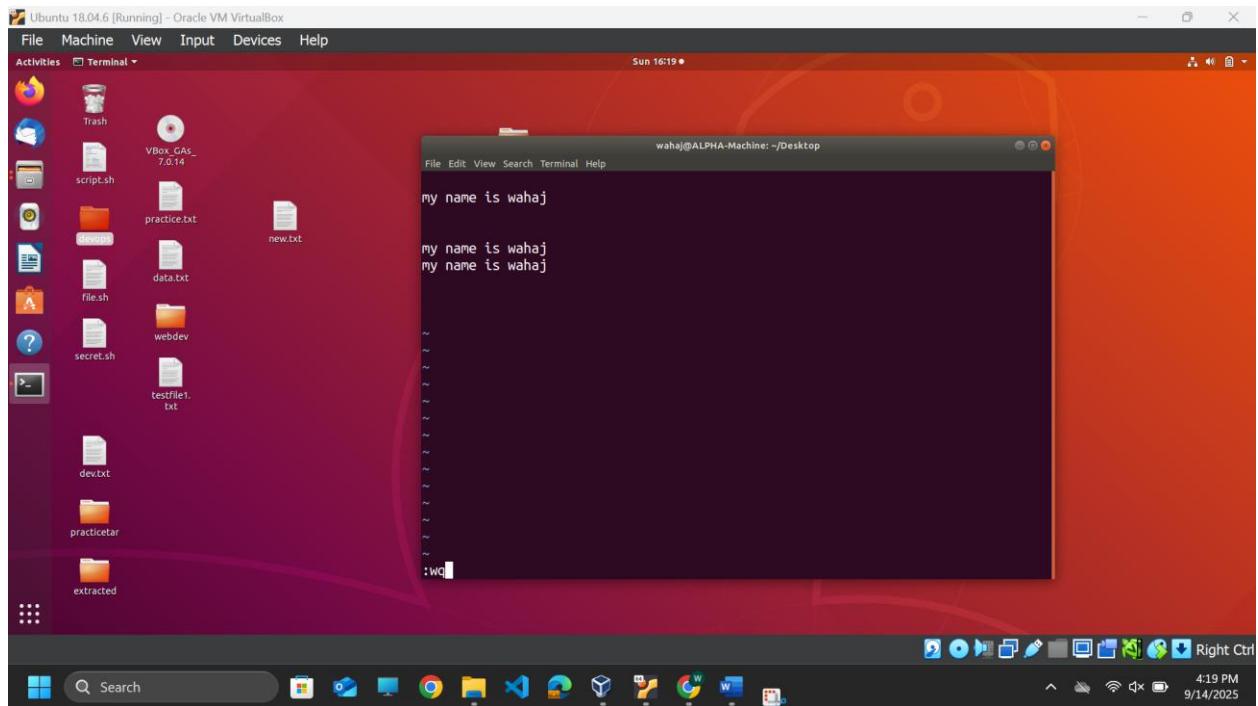
set nonu

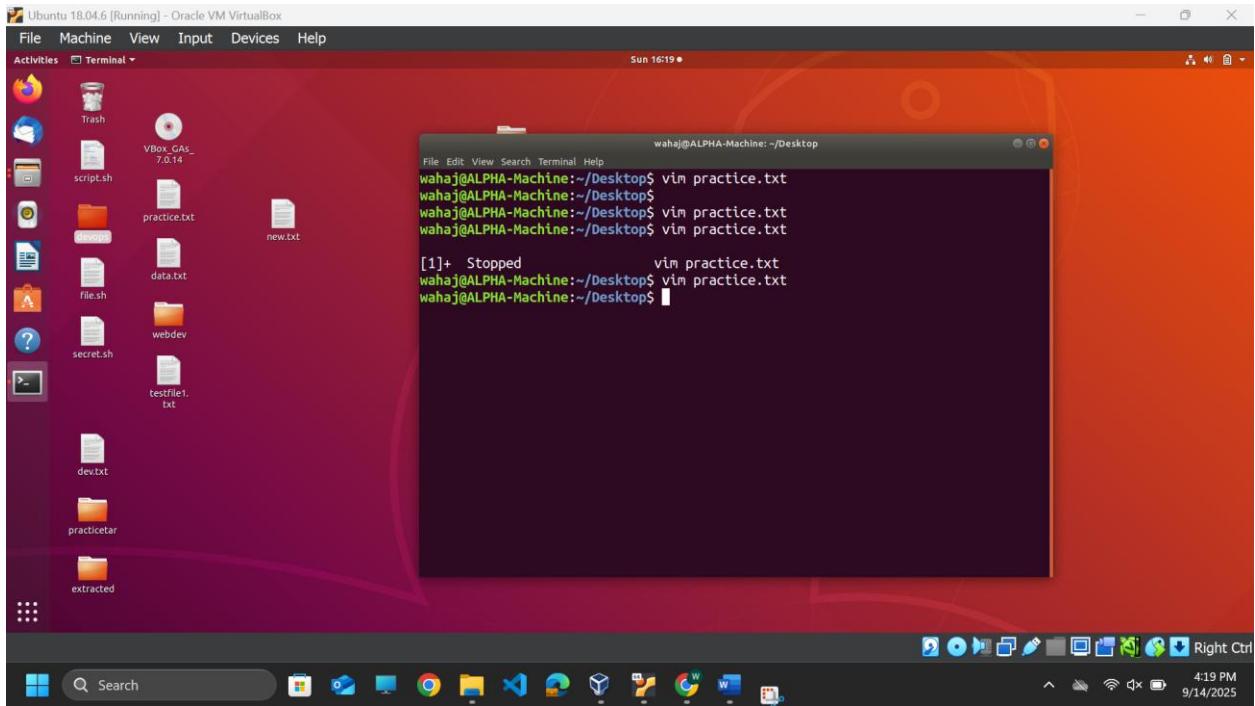


:w (save)



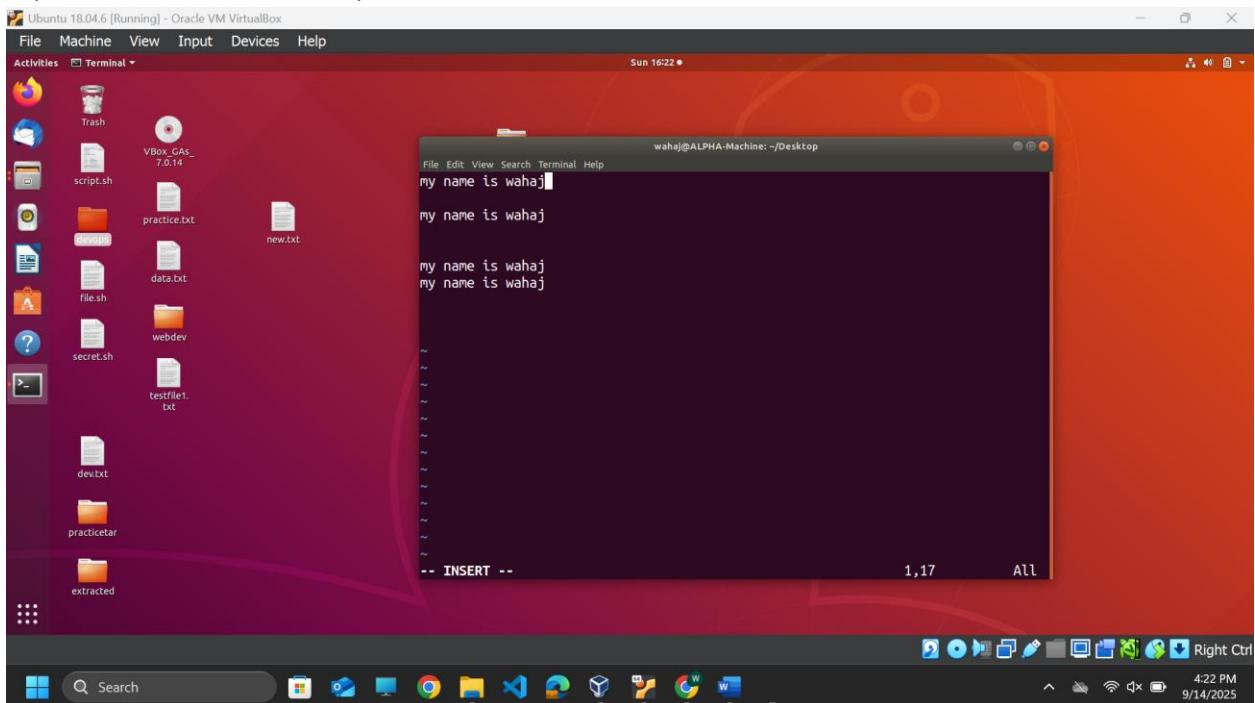
:wq Save and quit or :x



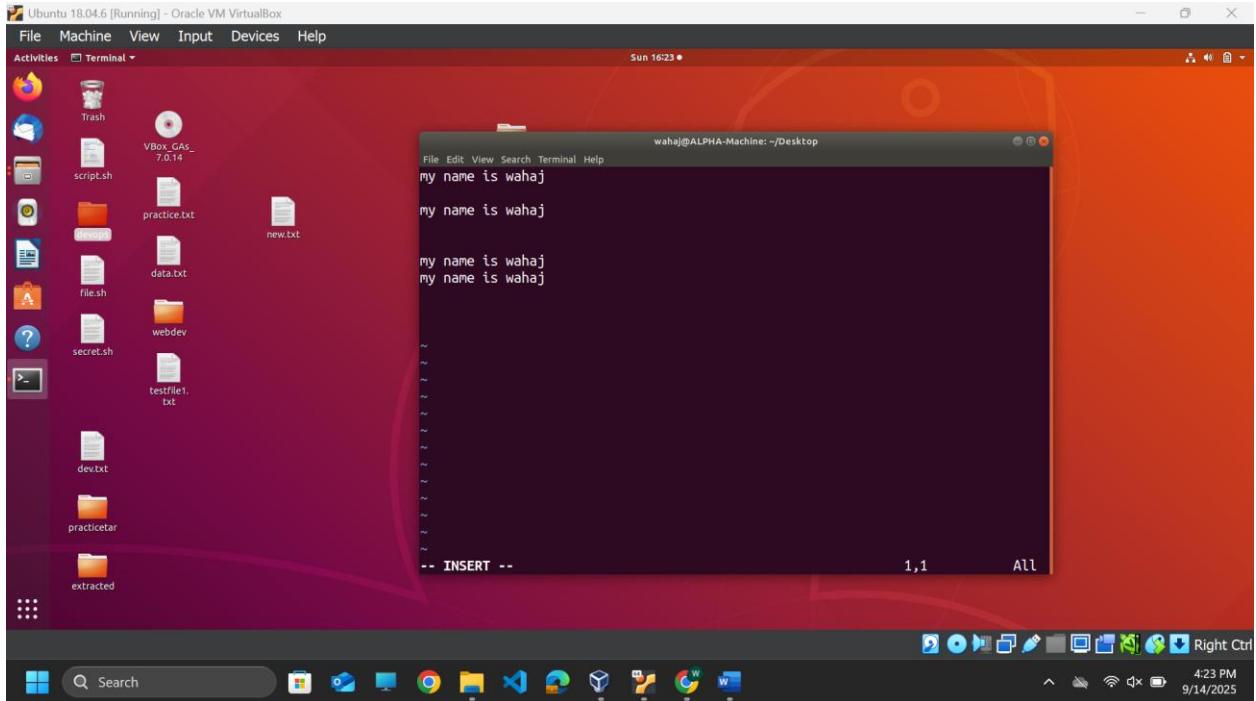


NORMAL MODE

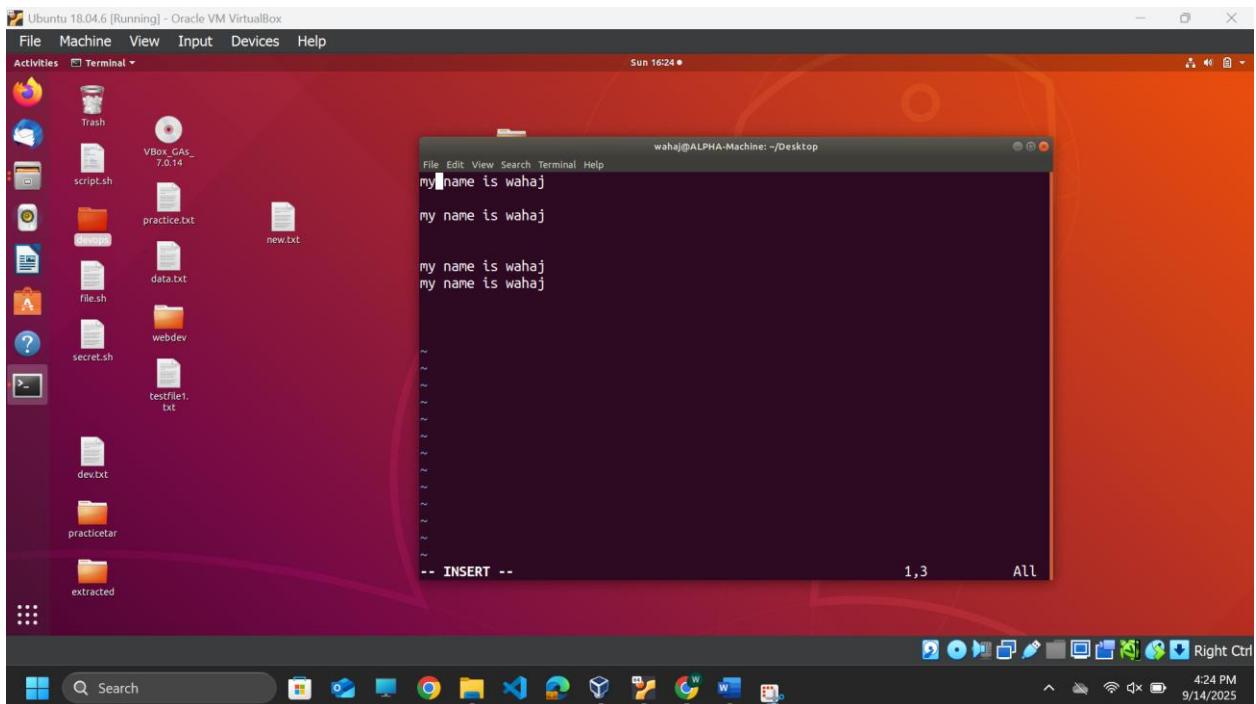
i (Insert before cursor)



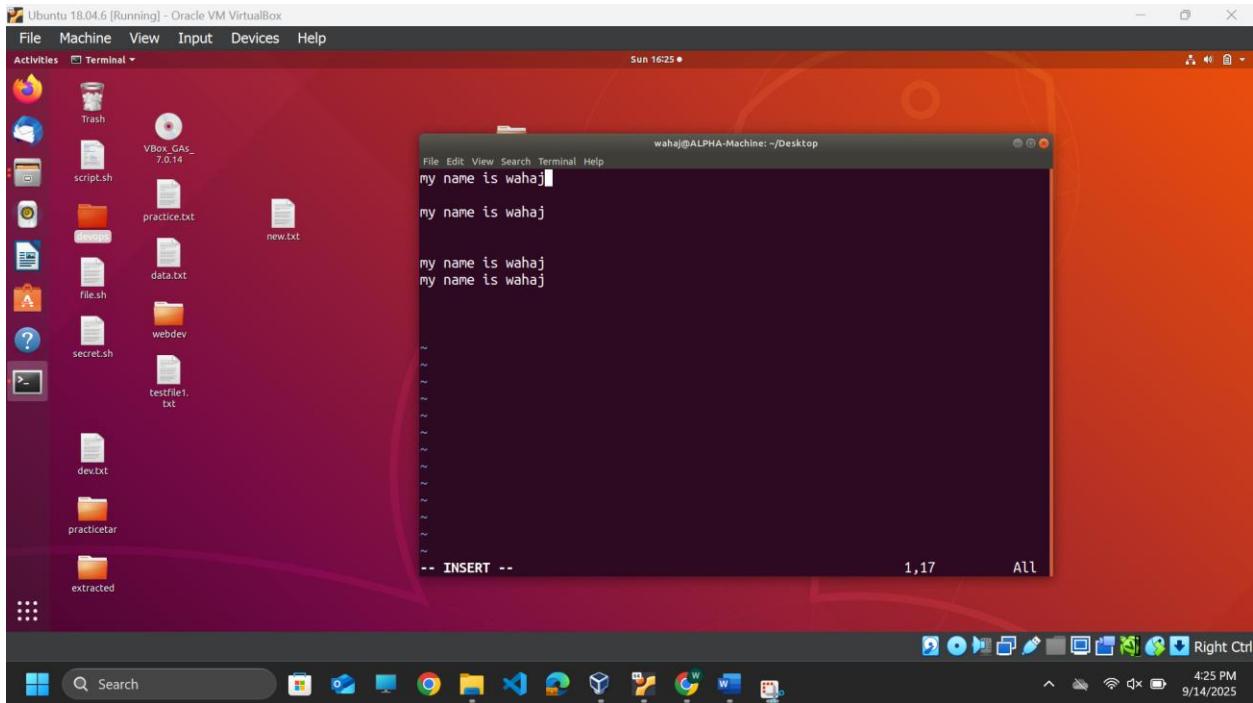
I (Insert at start of line)



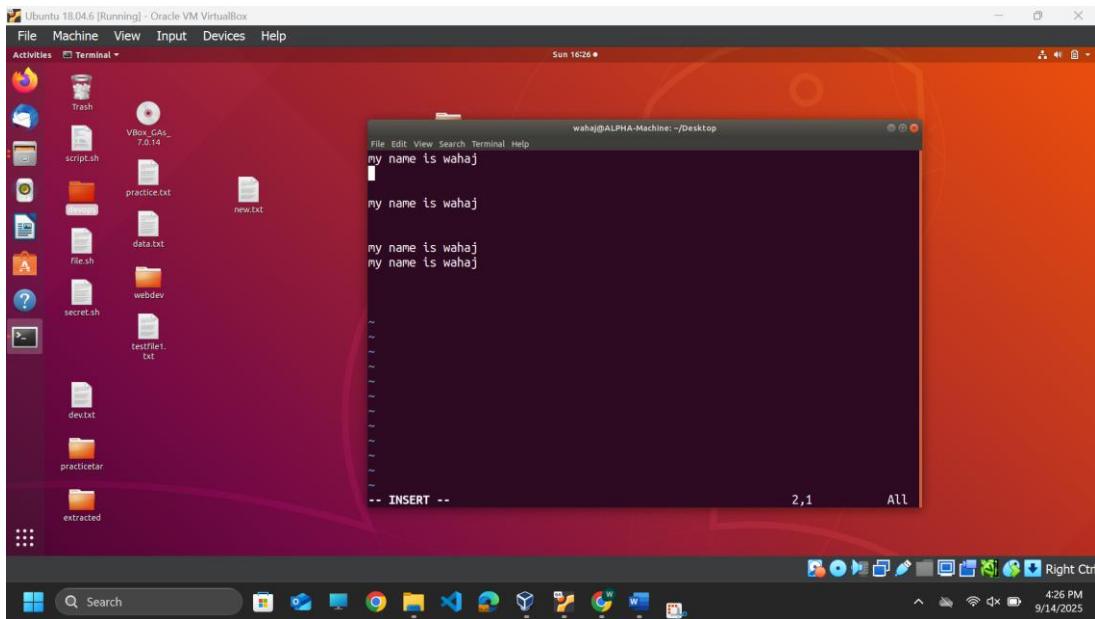
a (insert after cursor)



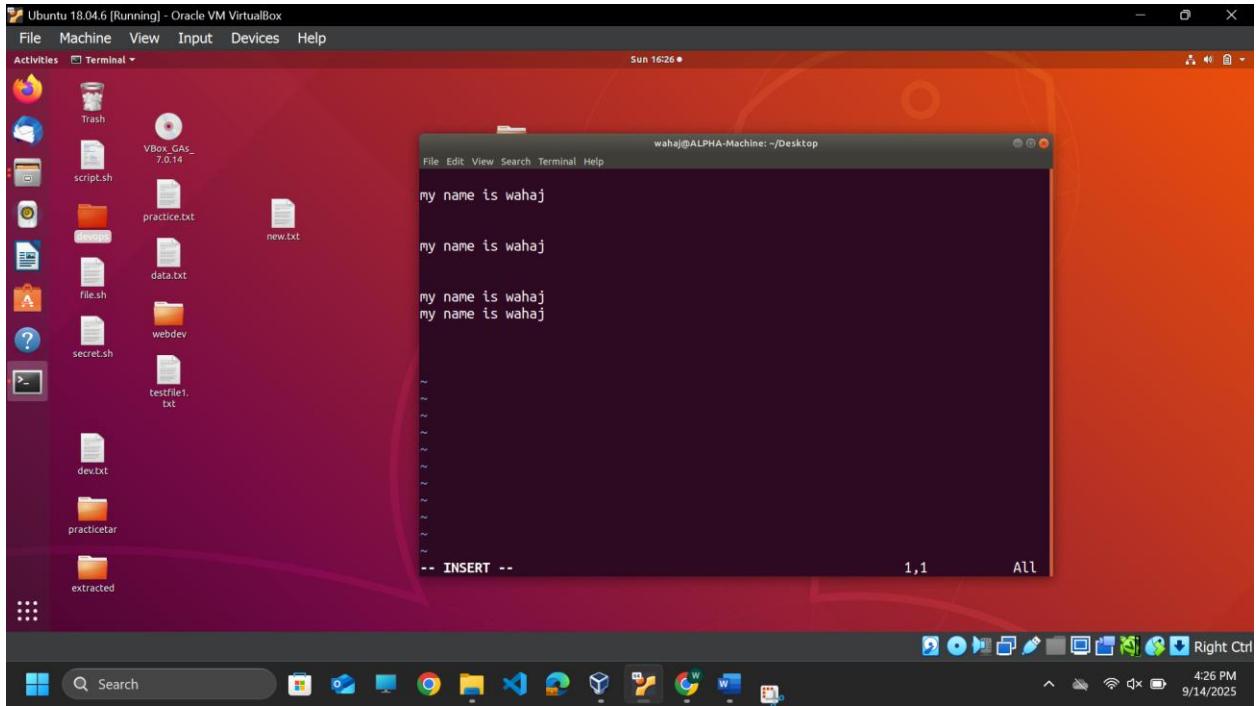
A Insert at end of line



o Open a new line below

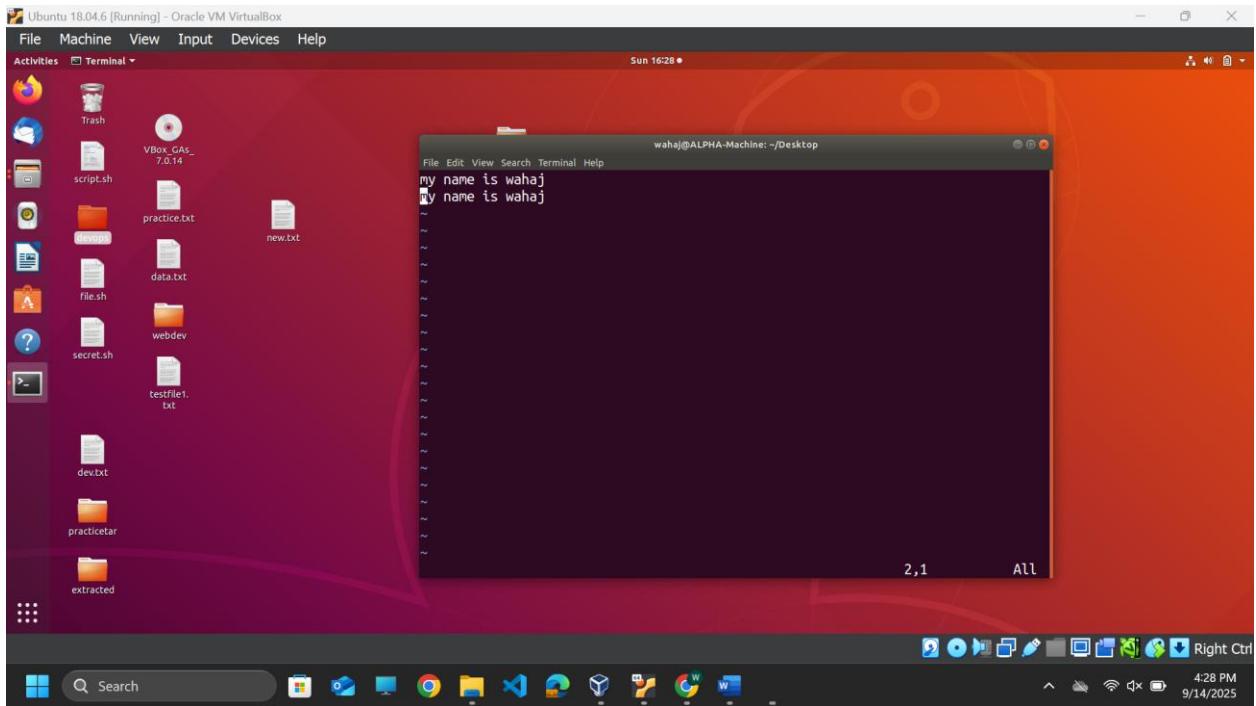


O Open a new line above

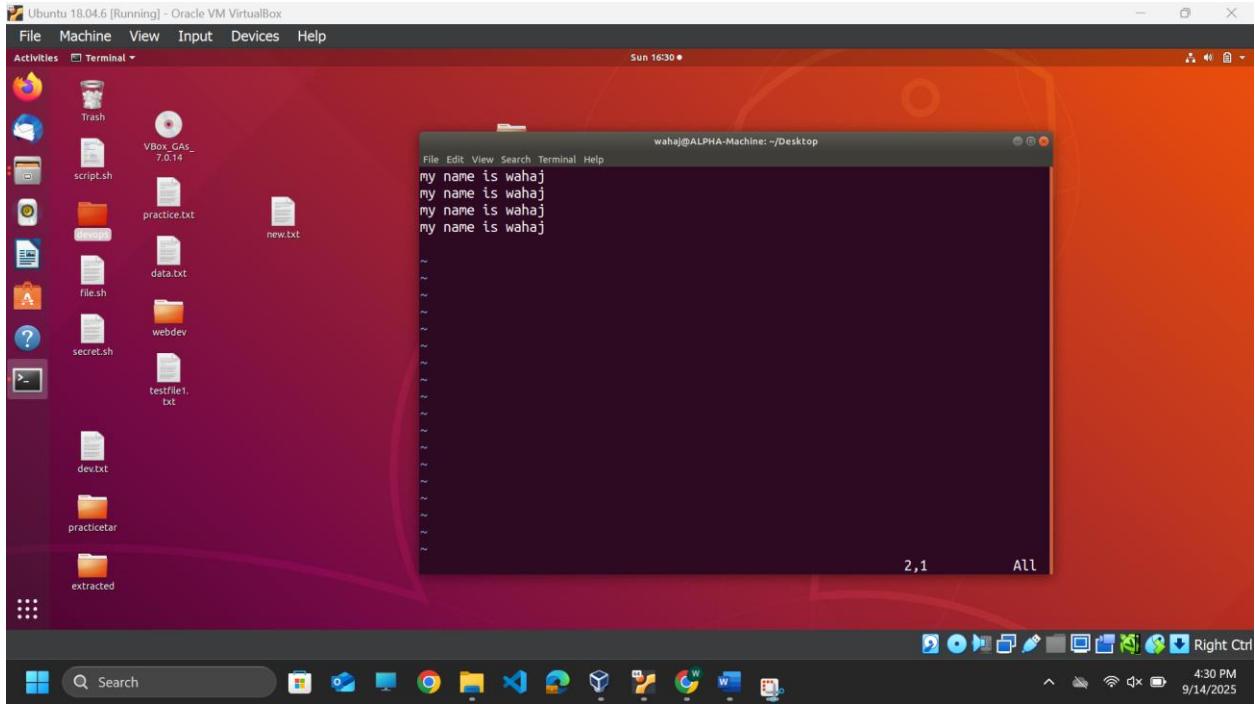


Copy, Cut, Paste

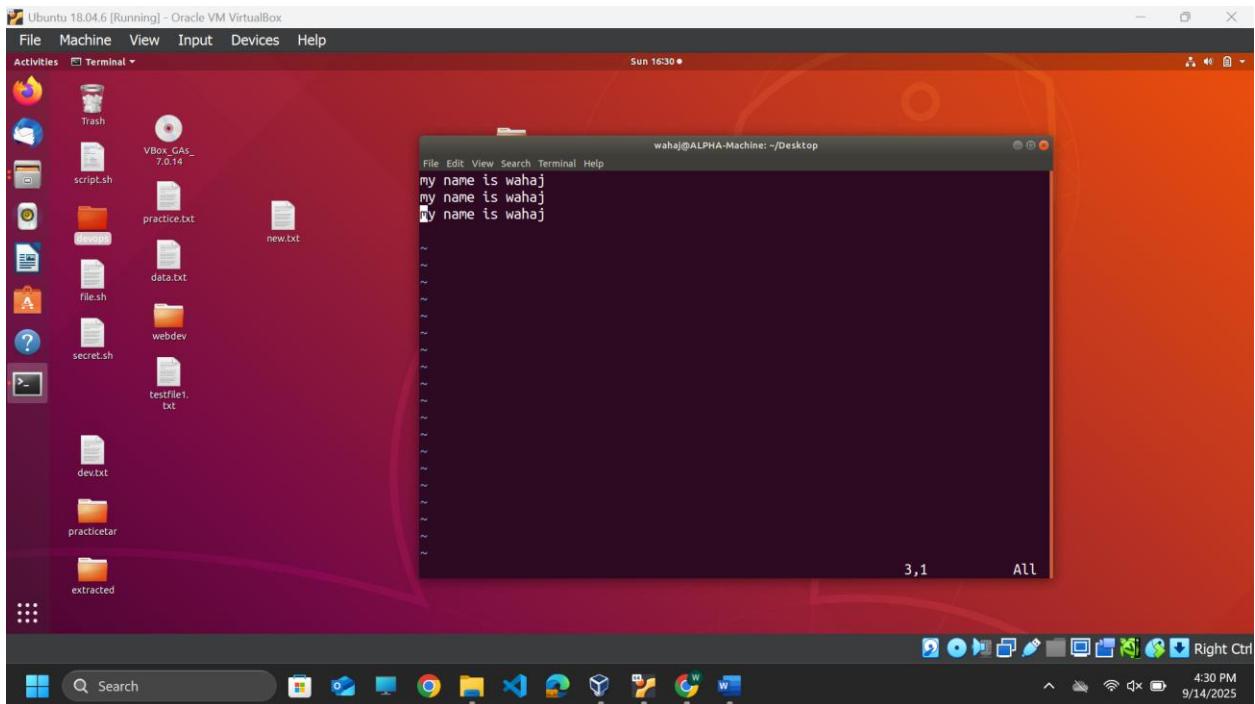
Copy (line) yy



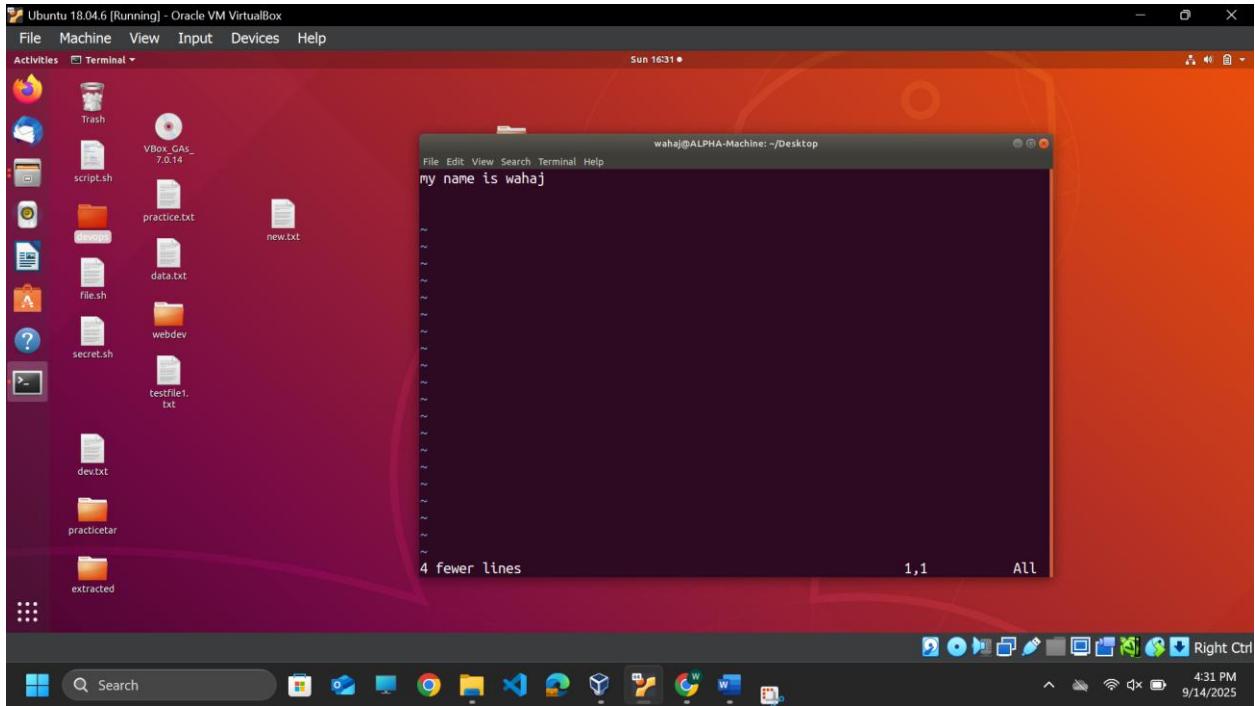
Copy (2 lines) 2yy



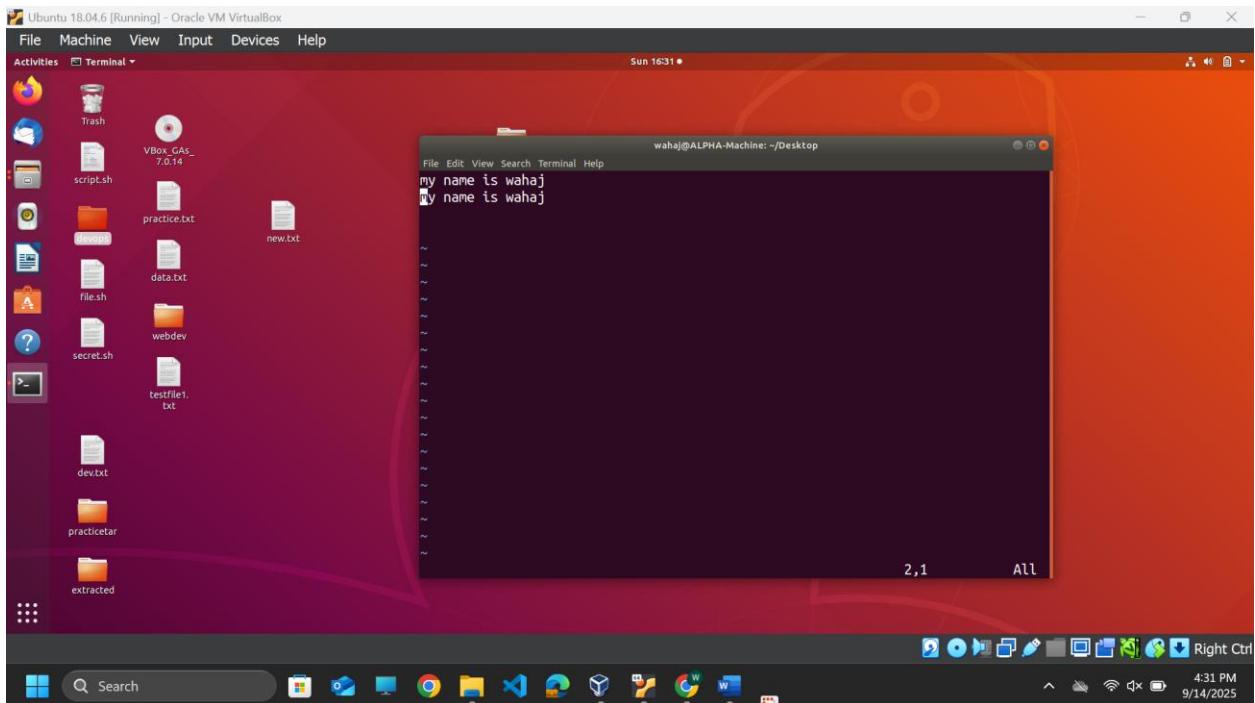
Delete (cut) line dd



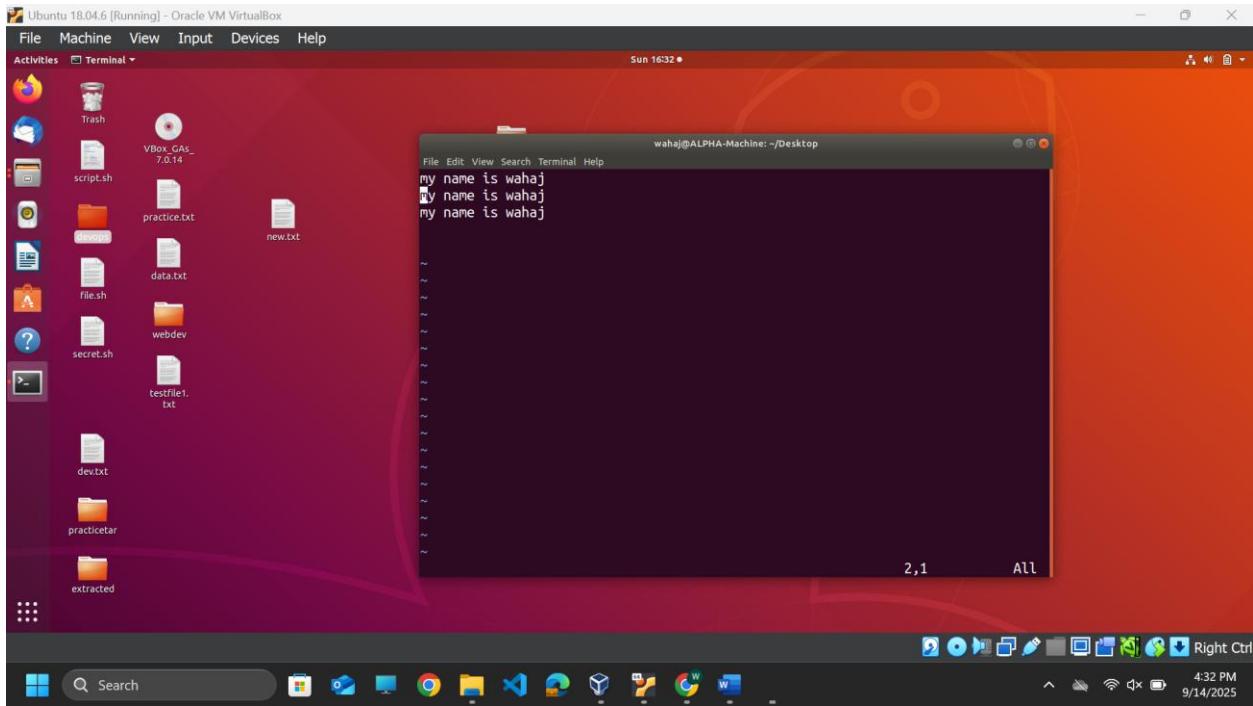
Delete 4 lines 4dd



Paste below p

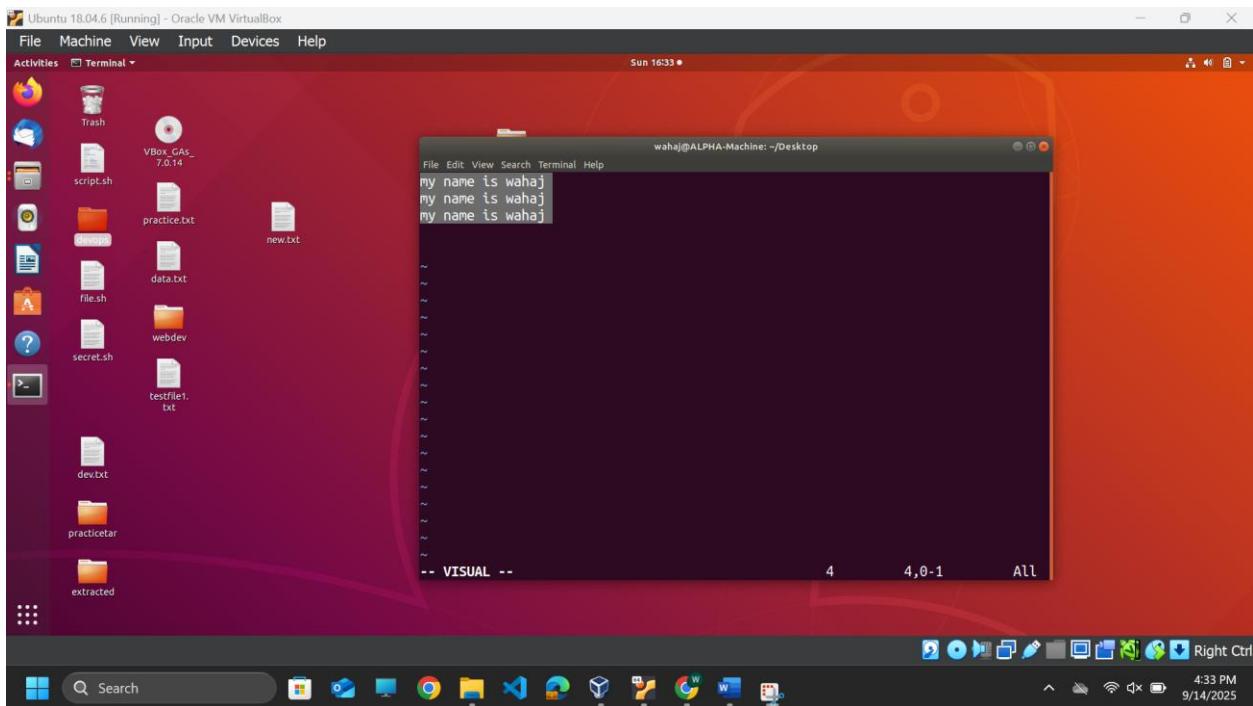


Paste above P

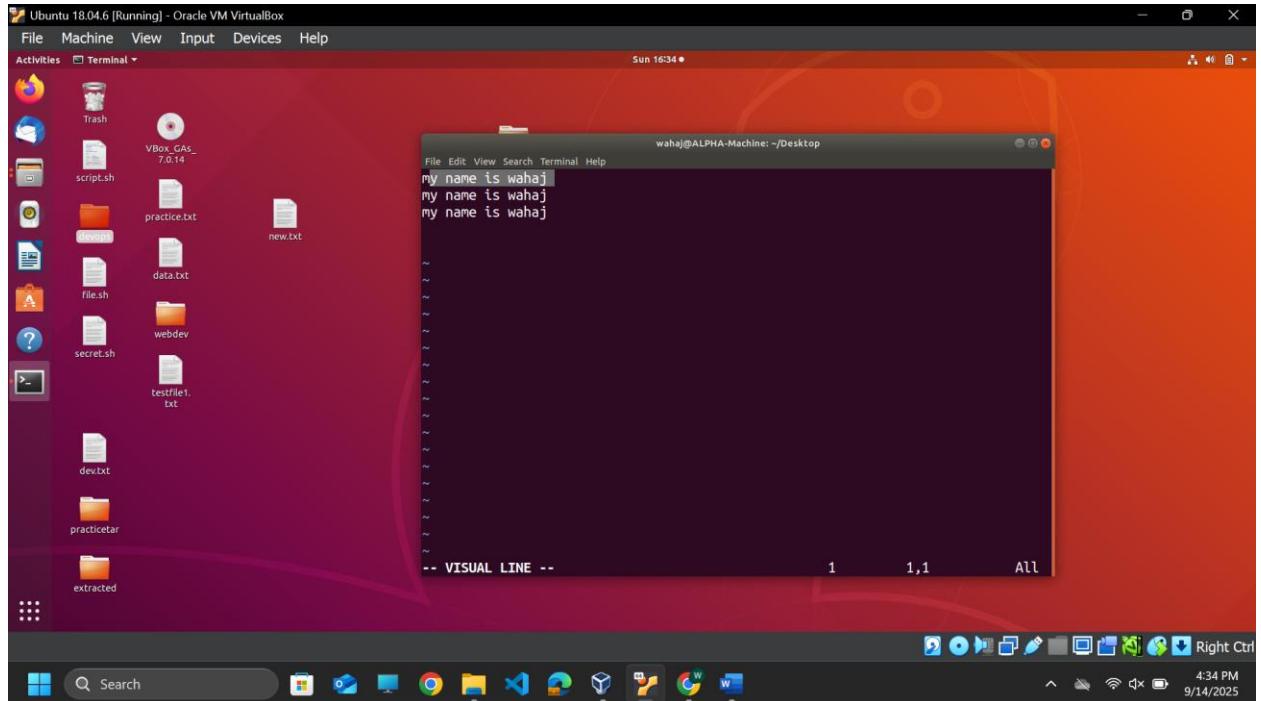


Line and Block Editing

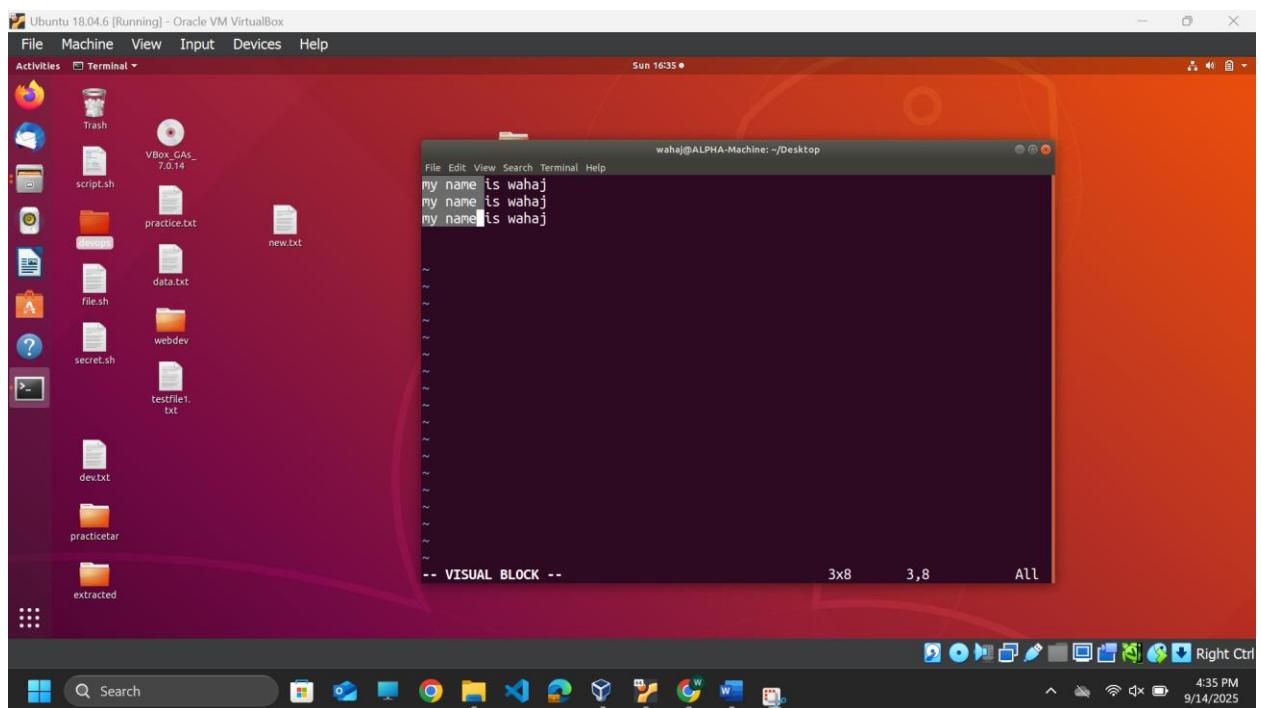
Visual Mode (v): Select characters/words



Visual Line Mode (V): Select entire line



Visual Block Mode (Ctrl + v): Select columns



Sample Practice

