

HW 1, Ganesh Prasad Chandra Shekar - A20557831

1. Setup VM, Linux, and basic testing – must take screenshots at each step to receive points

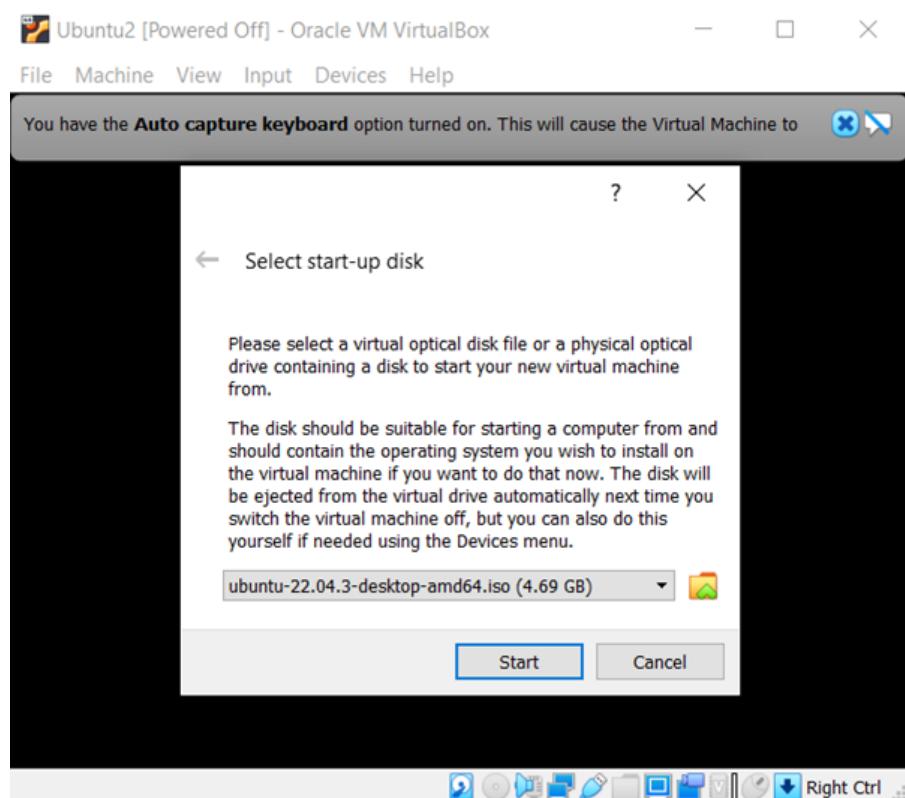
1.b. Dowload Oracle VirtualBox 6.1.36

- **VirtualBox 6.1.36** (*released July 19 2022*)
 - ↗ Windows hosts
 - ↗ Solaris hosts
 - ↗ Solaris 11 IPS hosts
 - Linux Hosts:
 - ↗ Oracle Linux 9 / Red Hat Enterprise Linux 9
 - ↗ Oracle Linux 8 / Red Hat Enterprise Linux 8
 - ↗ Oracle Linux 7 / Red Hat Enterprise Linux 7 / CentOS 7
 - ↗ Oracle Linux 6 / Red Hat Enterprise Linux 6 / CentOS 6
 - ↗ Ubuntu 22.04
 - ↗ Ubuntu 20.04
 - ↗ Ubuntu 19.10 / 20.10 / 21.04
 - ↗ Ubuntu 18.04 / 18.10 / 19.04
 - ↗ Ubuntu 16.04
 - ↗ Debian 11
 - ↗ Debian 10
 - ↗ Debian 9
 - ↗ openSUSE 15.3 / 15.4 / 15.5
 - ↗ openSUSE 15.0
 - ↗ openSUSE 13.2 / Leap 42
 - ↗ Fedora 36 / 37 / 38
 - ↗ Fedora 35
 - ↗ Fedora 33 / 34
 - ↗ Fedora 32
 - ↗ All distributions
 - ↗ Extension Pack
 - ↗ Sources

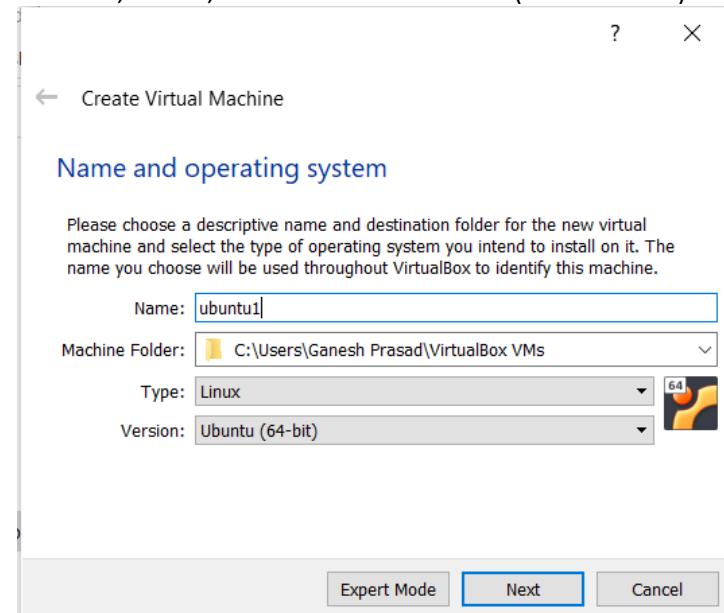
1.c. Install Virtual Box

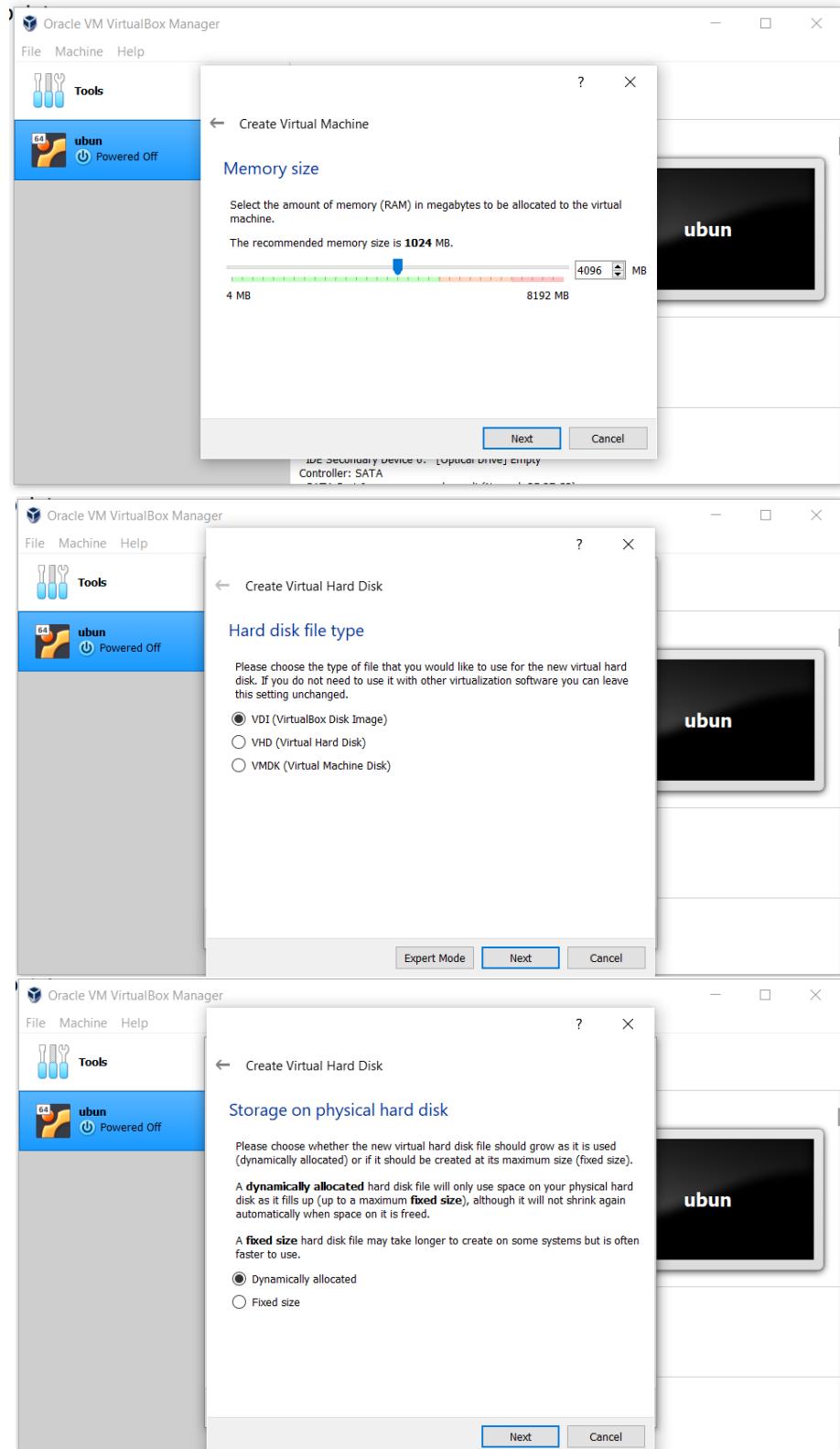


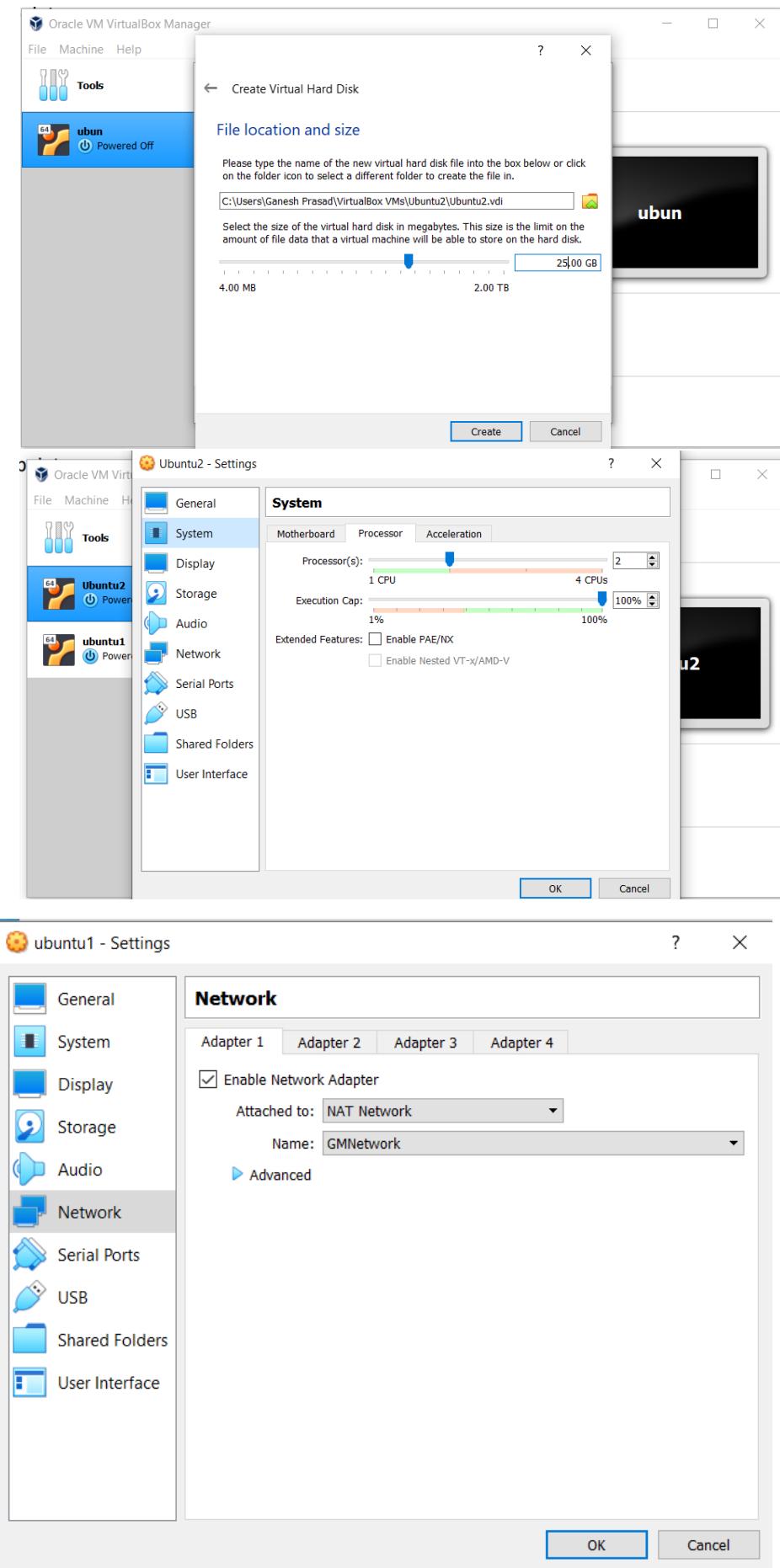
1.d. Download and install Ubuntu Desktop 22.04.3



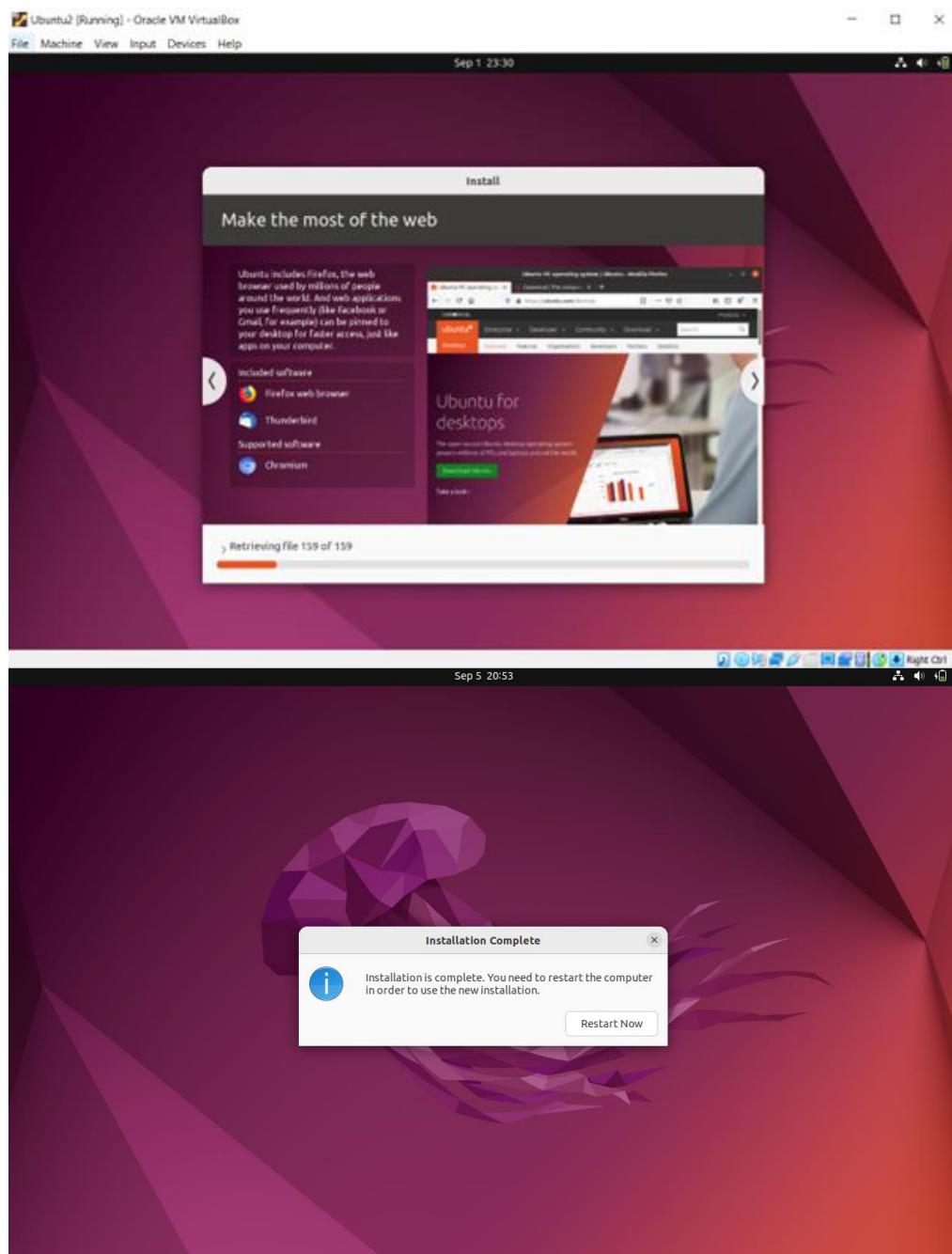
1.e. Create Virtual Machine (VM), to support Linux, Ubuntu, 64-bit, 4GB RAM, Virtual Disk 25GB, VDI image, dynamically allocated, 2-core, and a network interface (1GbE or WiFi) with NAT support.



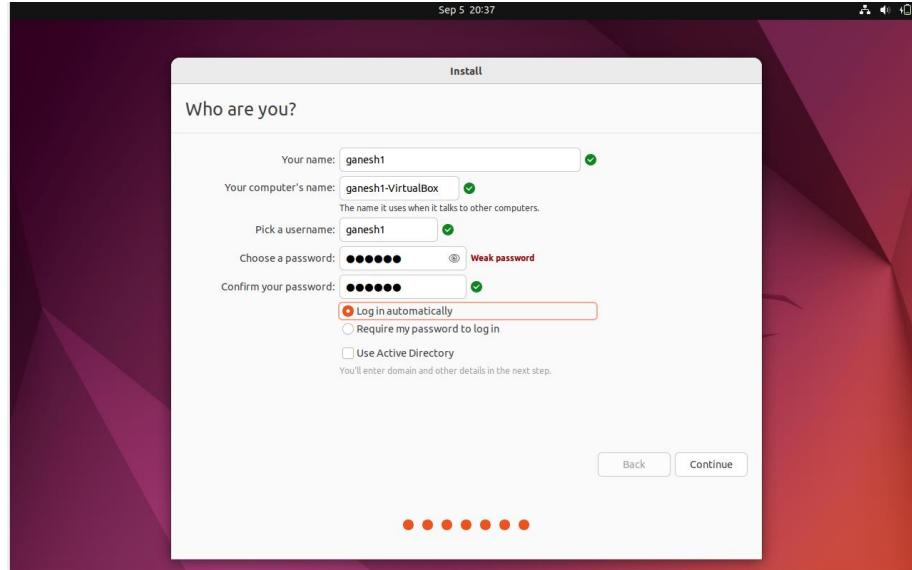




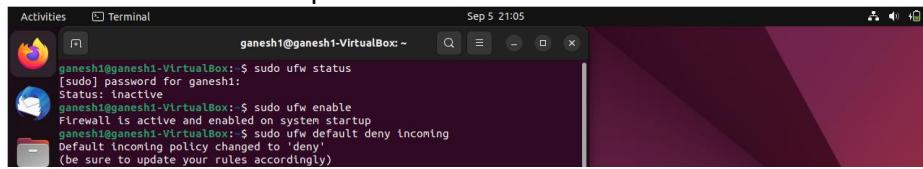
1.f. Install Linux from the ISO image



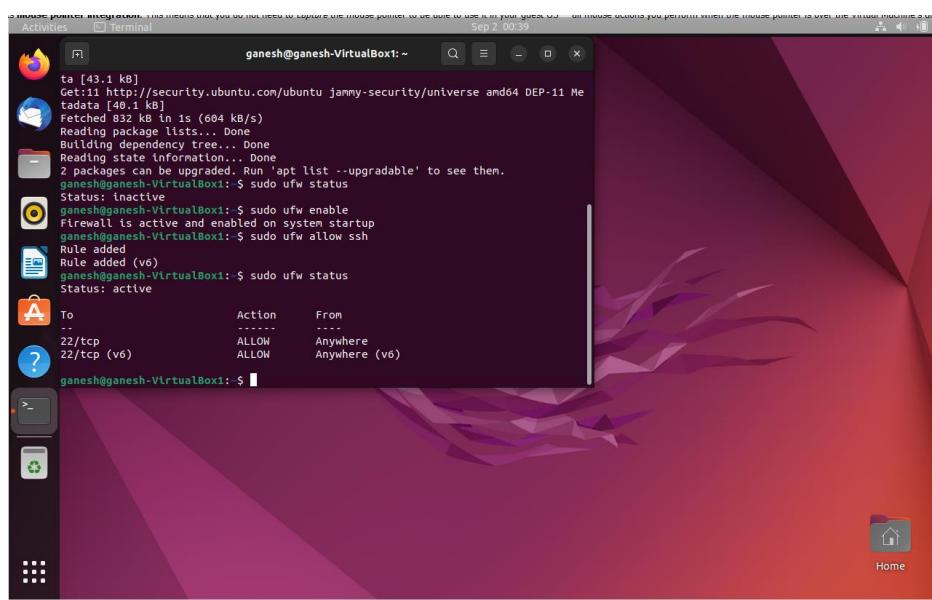
1.g. Create a user id and password.



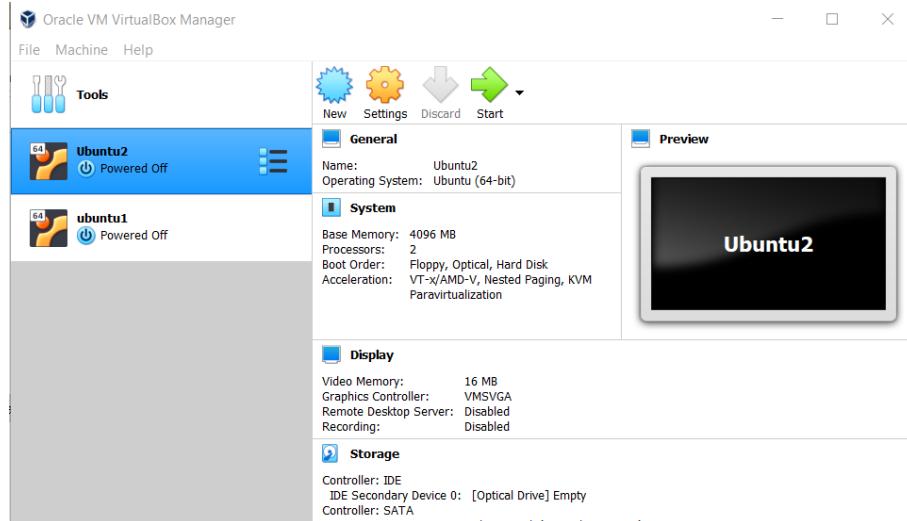
1.h. Turn on Firewall and block all ports.



1.i. Enable SSH access to your new Linux installation; open SSH port in firewall



1.j. Repeat steps e through i, and create another VM with the same specifications as the first one.



1.k. Create private/public keys and install them properly in both of your new VMs. Test that you can connect remotely to your VMs with your keys, from one VM to the other VM.

A screenshot of a terminal window titled "ganesh1@ganesh1: ~". The user runs the command "ssh-keygen". The output shows the generation of an RSA key pair. It asks for a file to save the key, noting that "/home/ganesh1/.ssh/id_rsa" already exists and asking if it should be overwritten (y/n). The user types "y". It then prompts for a passphrase and asks to enter it again. After saving the key, it displays the key fingerprint and the public key location. Finally, it shows the randomart image of the key.

```
ganesh1@ganesh1:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ganesh1/.ssh/id_rsa):
/home/ganesh1/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ganesh1/.ssh/id_rsa
Your public key has been saved in /home/ganesh1/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:Ma2fE7Fo8aWKQssNPRkQzbs8BeHfiH5yKpUGmncwppo ganesh1@ganesh1
The key's randomart image is:
----[RSA 3072]----
|   o+..
|   o+ .
|   oo+ o .
| =..=.X =
| =o==+S *
| +o.=o+ o o
| o .++=.+ +
| E   ... =   .
| ...
----[SHA256]-----
ganesh1@ganesh1:~$
```

Copy public key from VM1 to VM2

```
ganesh1@ganesh1: ~
gshadow          passwd-      wpa_supplicant
gshadow-         pcmcia       X11
gss              perl         xattr.conf
gtk-2.0          pki          xdg
gtk-3.0          pm           xml
hdparm.conf      pam2ppa.conf zsh_command_not_found
host.conf        polkit-1
hostid          ppp
ganesh1@ganesh1:/etc$ cd ssh
ganesh1@ganesh1:/etc/ssh$ sudo nano ssh_config
[sudo] password for ganesh1:
ganesh1@ganesh1:/etc/ssh$ cd
ganesh1@ganesh1:~$ cd
ganesh1@ganesh1:~$ ssh-copy-id -f -i ~/.ssh/id_rsa.pub ganesh2@192.168.1.3
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ganesh1/.ssh/id_rsa.pub"
ganesh2@192.168.1.3's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'ganesh2@192.168.1.3'"
and check to make sure that only the key(s) you wanted were added.

ganesh1@ganesh1:~$
```

Testing connection from VM1 to VM2 and creating directory in VM2

```
Activities Terminal Sep 11 10:25
ganesh1@ganesh1:~$ ssh ganesh2@192.168.1.3
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-32-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

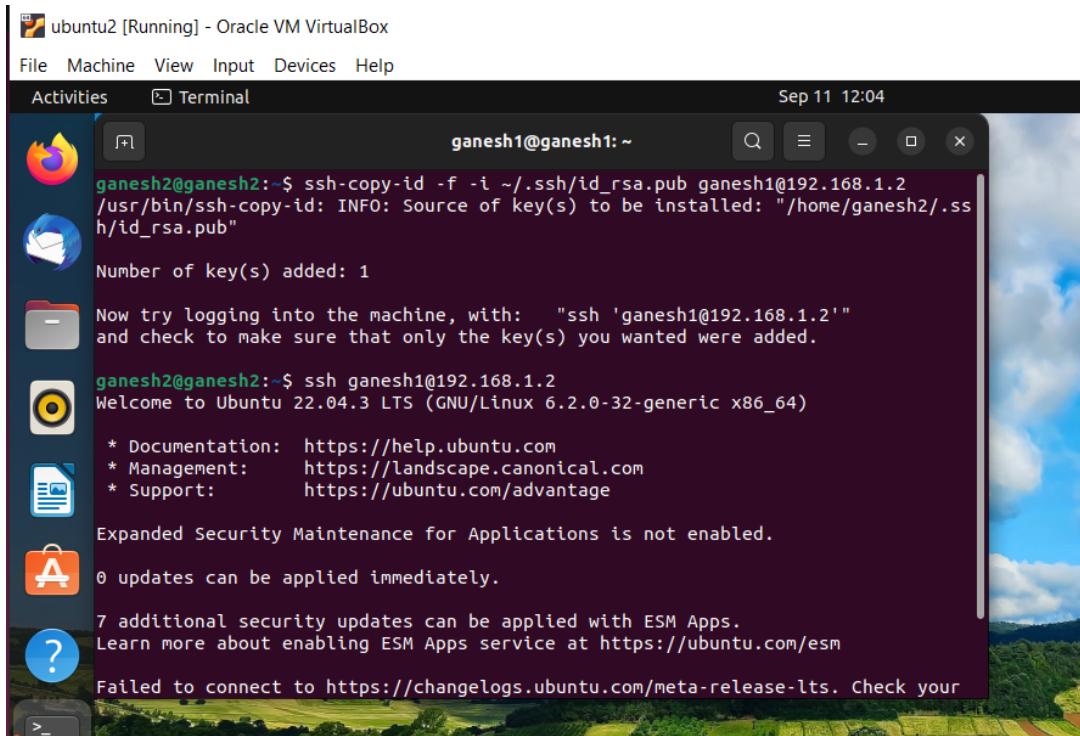
Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

7 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Last login: Mon Sep 11 02:38:05 2023 from 192.168.1.2
ganesh2@ganesh2:~$ cd ~/Desktop
ganesh2@ganesh2:~/Desktop$ mkdir Ganii
ganesh2@ganesh2:~/Desktop$
```

Repeat the same for Test connection VM2 to VM1



```
ubuntu2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Sep 11 12:04
ganesh1@ganesh1: ~
ganesh2@ganesh2:~$ ssh-copy-id -f -i ~/.ssh/id_rsa.pub ganesh1@192.168.1.2
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ganesh2/.ssh/id_rsa.pub"
Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'ganesh1@192.168.1.2'"
and check to make sure that only the key(s) you wanted were added.

ganesh2@ganesh2:~$ ssh ganesh1@192.168.1.2
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-32-generic x86_64)

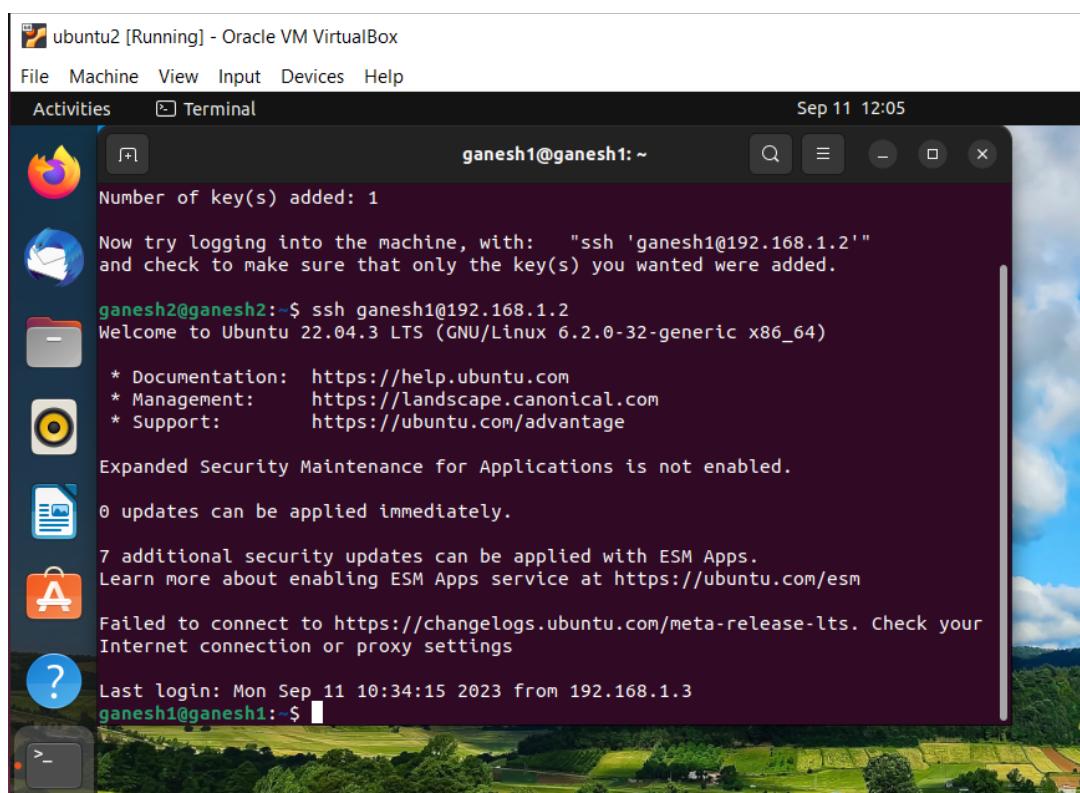
 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

7 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
Internet connection or proxy settings.
```



```
ubuntu2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Sep 11 12:05
ganesh1@ganesh1: ~
Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'ganesh1@192.168.1.2'"
and check to make sure that only the key(s) you wanted were added.

ganesh2@ganesh2:~$ ssh ganesh1@192.168.1.2
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-32-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

Expanded Security Maintenance for Applications is not enabled.

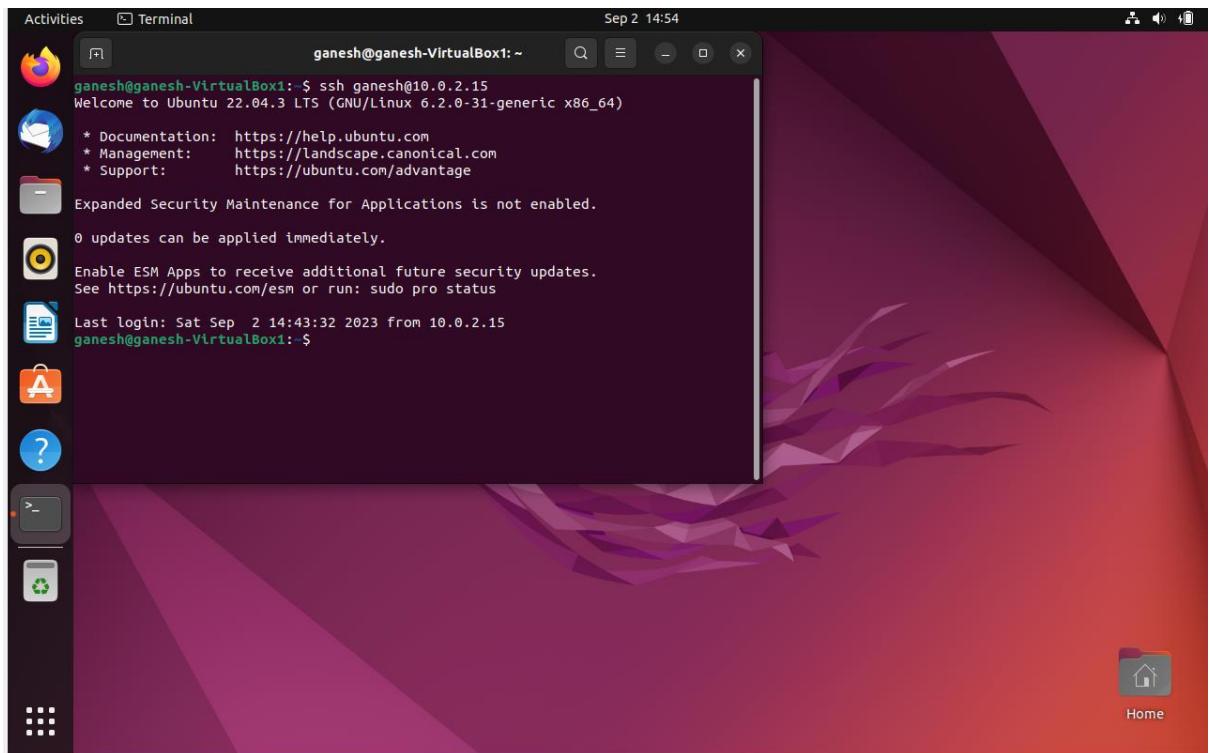
0 updates can be applied immediately.

7 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
Internet connection or proxy settings

Last login: Mon Sep 11 10:34:15 2023 from 192.168.1.3
ganesh1@ganesh1: $
```

2.a. ssh – this command is to connect to a remote SSH server



A screenshot of an Ubuntu desktop environment. A terminal window titled "ganesh@ganesh-VirtualBox1: ~" is open, showing the output of the "ssh" command. The terminal shows a welcome message for Ubuntu 22.04.3 LTS, system documentation, management, and support links, and a note about expanded security maintenance. It also shows the last login information and ends with a "\$" prompt. The desktop background is a purple and red abstract design. The Unity interface is visible, including the Dash icon bar on the left and a "Home" button on the right.

```
ganesh@ganesh-VirtualBox1: ~$ ssh ganesh@10.0.2.15
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-31-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

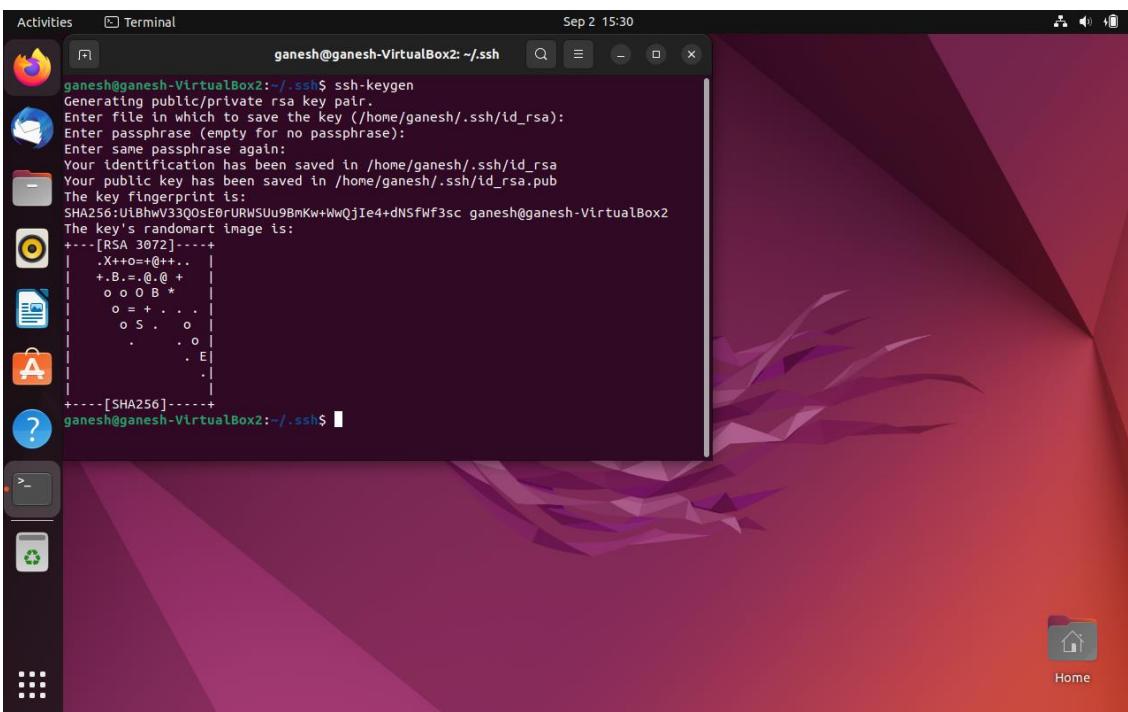
Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Sat Sep  2 14:43:32 2023 from 10.0.2.15
ganesh@ganesh-VirtualBox1: ~$
```

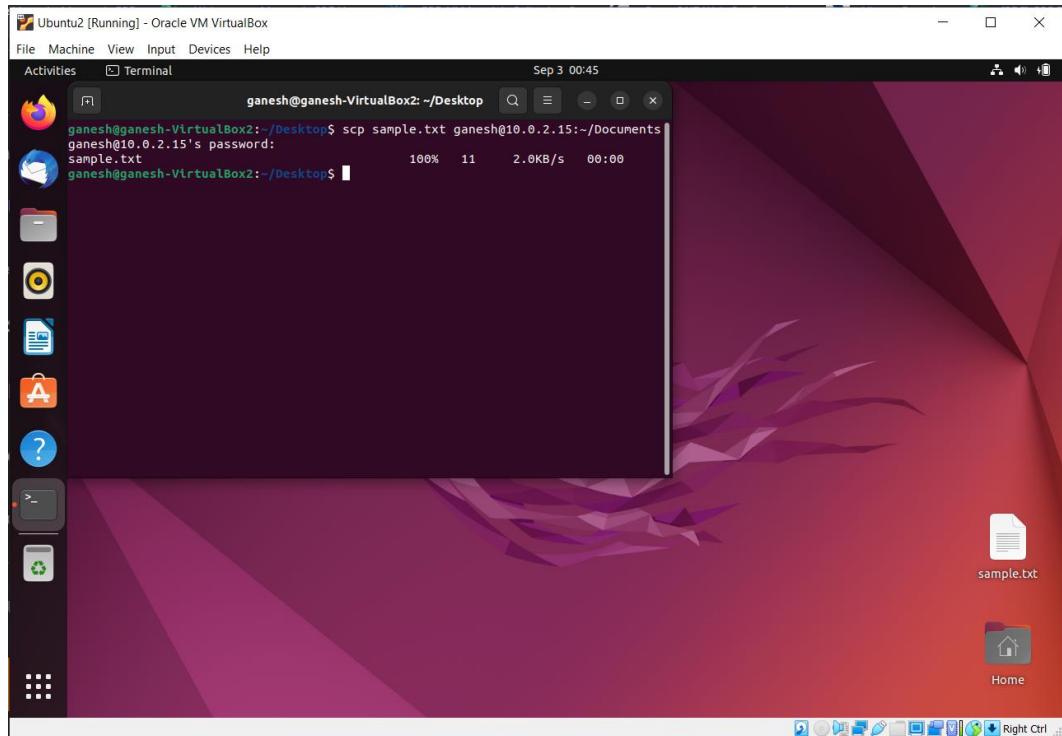
2.b. ssh-keygen – this command is to create a public and private key pair



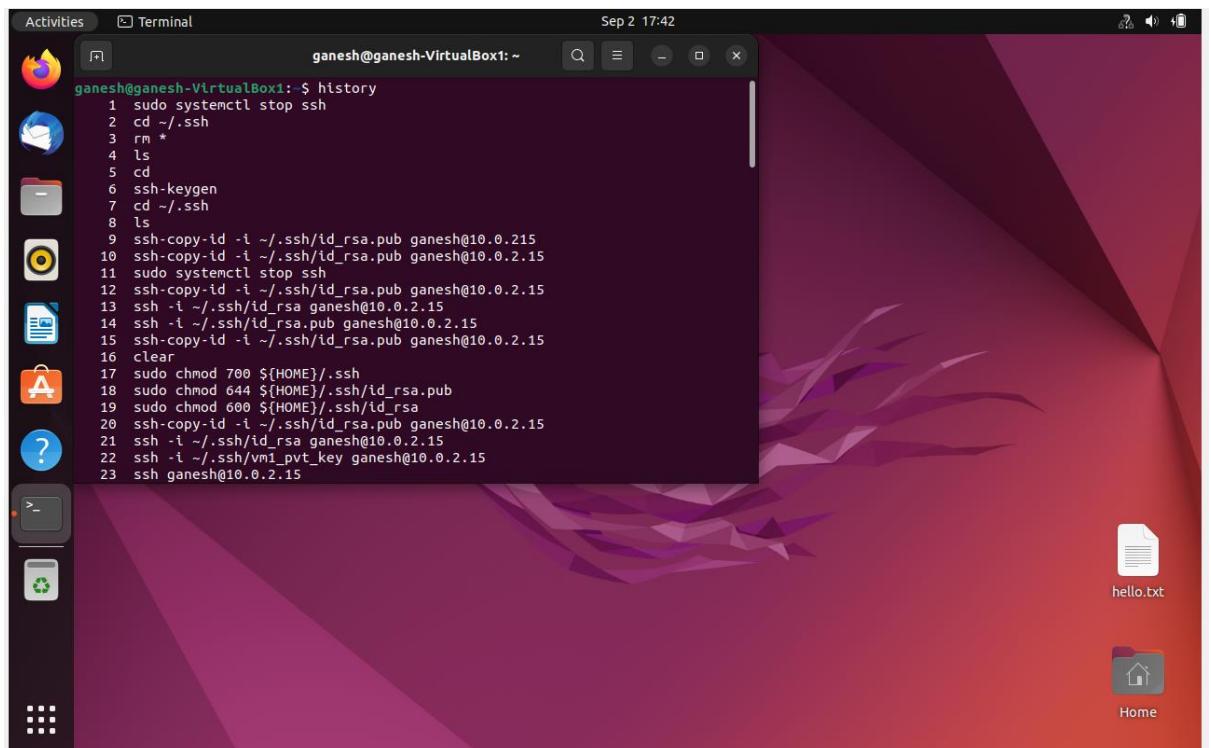
A screenshot of an Ubuntu desktop environment. A terminal window titled "ganesh@ganesh-VirtualBox2: ~/.ssh" is open, showing the output of the "ssh-keygen" command. The command generates an RSA key pair, prompting for a passphrase and saving the keys to the specified directory. It then displays the key fingerprint and the randomart image. The desktop background is a purple and red abstract design. The Unity interface is visible, including the Dash icon bar on the left and a "Home" button on the right.

```
ganesh@ganesh-VirtualBox2: ~/.ssh$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ganesh/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ganesh/.ssh/id_rsa
Your public key has been saved in /home/ganesh/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:UiBhwV3jQSe0rURWSUu9BmKw+WwQjIe4+dNSfWf3sc ganesh@ganesh-VirtualBox2
The key's randomart image is:
+---[RSA 3072]---+
| .X++o=@++.. |
| +B.=.0@ + |
| o o O B * |
| o = + . . |
| o S . o |
| . . o |
| . E |
+---[SHA256]---+
ganesh@ganesh-VirtualBox2: ~/.ssh$
```

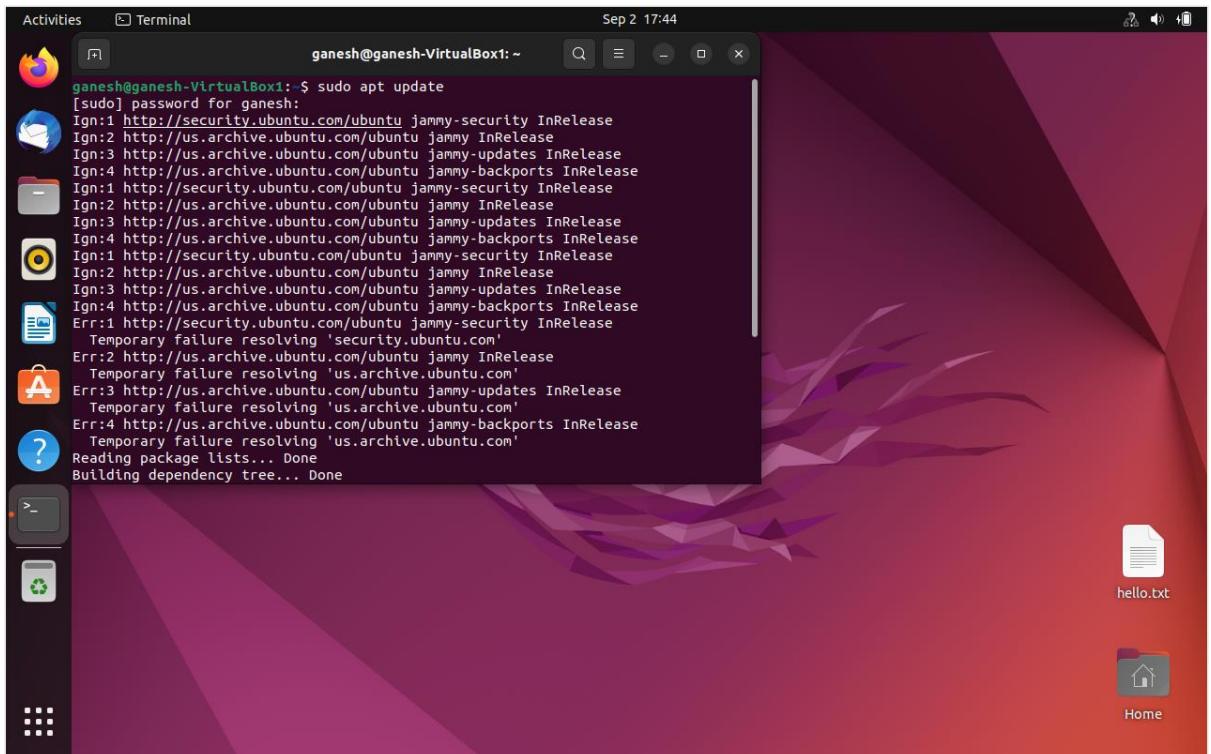
2.c. scp - used to transfer files from one system to a remote server.



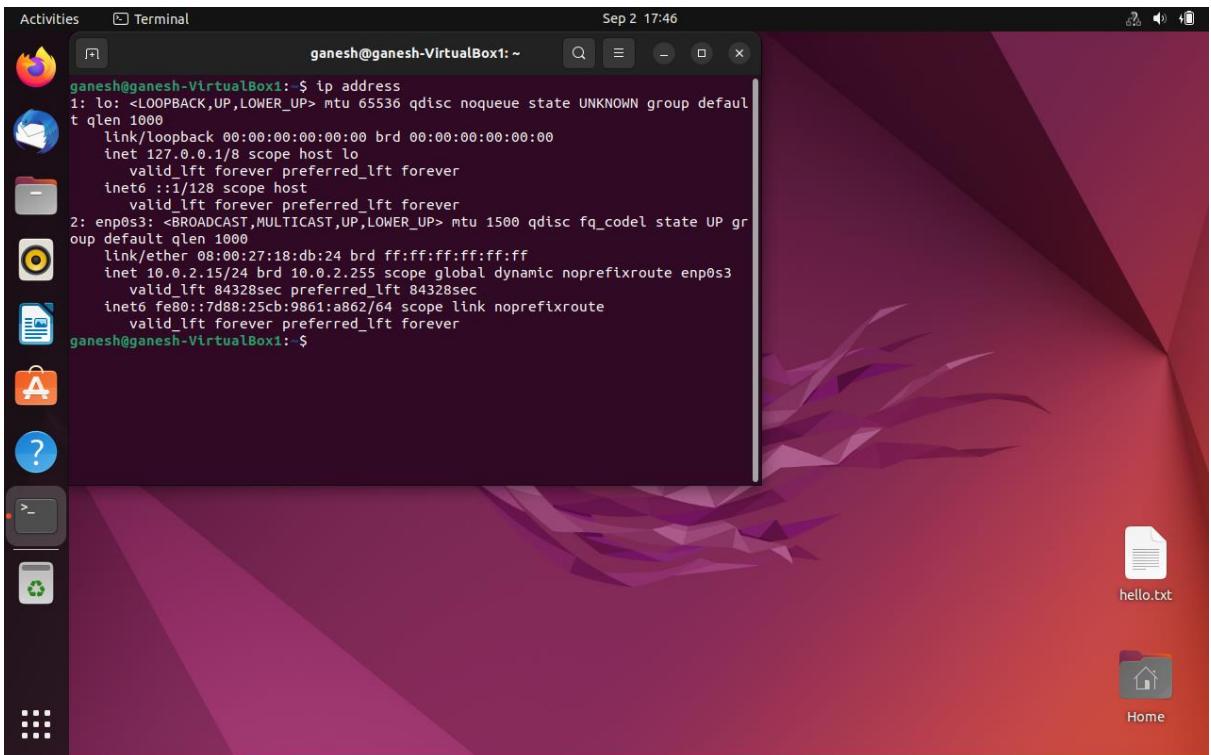
2.d. history - lists all the commands entered in the past



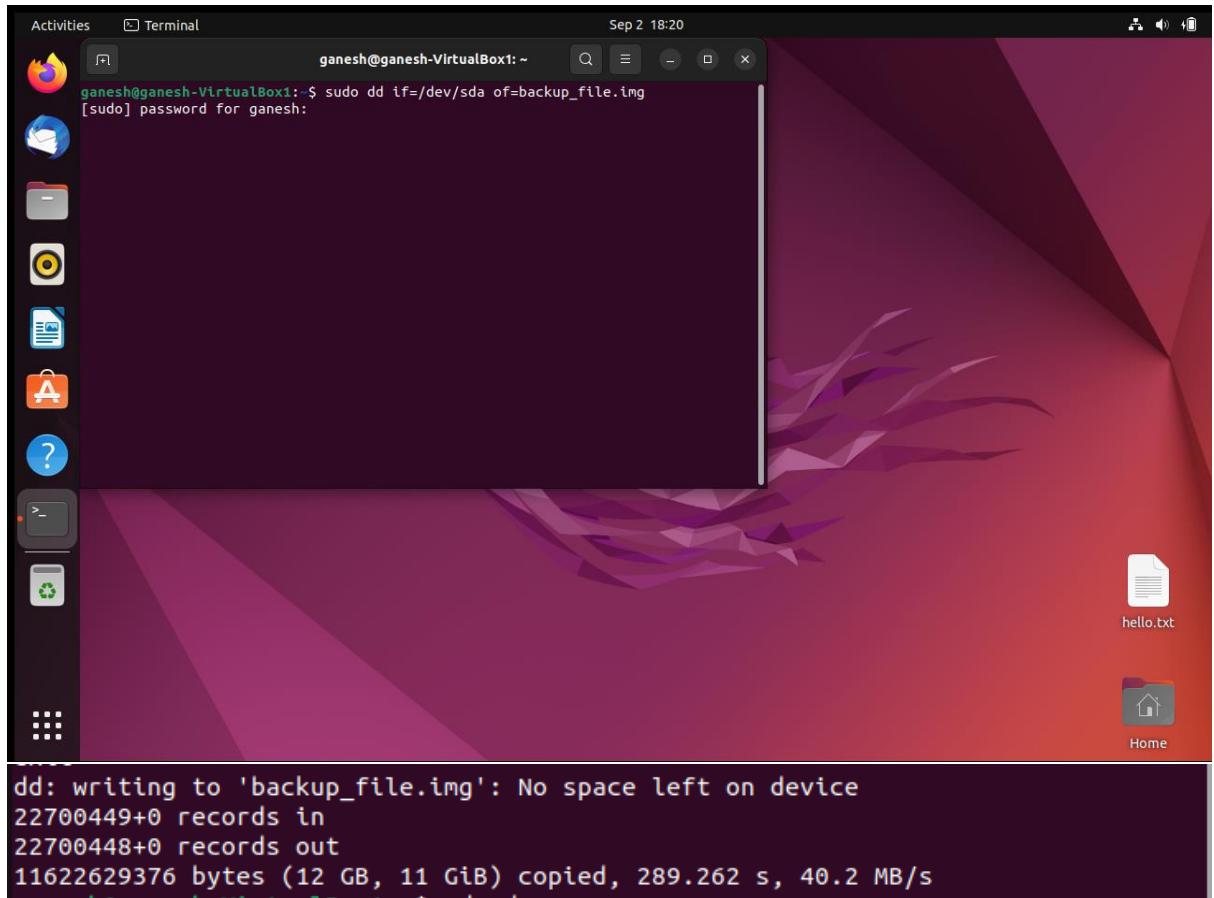
2.e. sudo - Running any command with SuperUser Privilege



2.f. ip – fetch network related information



2.g. dd – This command is used for creating disk images, copying data from one machine to another.

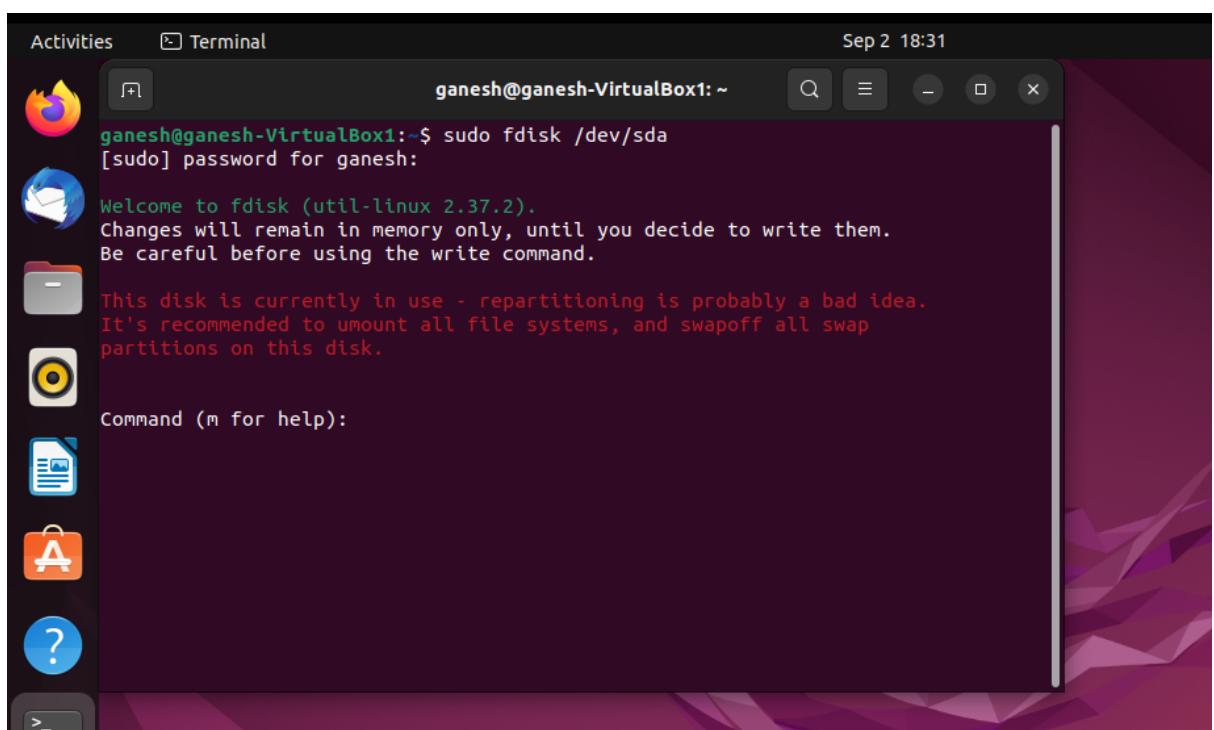


A screenshot of an Ubuntu desktop environment. A terminal window is open in the top-left corner, showing the command `sudo dd if=/dev/sda of=backup_file.img` being run. The terminal window has a dark background and white text. The desktop background is a purple and red abstract design. On the desktop, there is a file icon labeled "hello.txt" and a folder icon labeled "Home". The system tray at the bottom shows icons for battery, signal, and volume. The taskbar at the bottom has icons for the Dash, Home, and other applications.

```
ganesh@ganesh-VirtualBox1: ~$ sudo dd if=/dev/sda of=backup_file.img
[sudo] password for ganesh:

dd: writing to 'backup_file.img': No space left on device
22700449+0 records in
22700448+0 records out
11622629376 bytes (12 GB, 11 GiB) copied, 289.262 s, 40.2 MB/s
ganesh@ganesh-VirtualBox1: ~ cd sda
```

2.h. fdisk- this commands helps us to partition a disk



A screenshot of an Ubuntu desktop environment. A terminal window is open in the top-left corner, showing the command `sudo fdisk /dev/sda` being run. The terminal window has a dark background and white text. The desktop background is a purple and red abstract design. On the desktop, there is a file icon labeled "hello.txt" and a folder icon labeled "Home". The system tray at the bottom shows icons for battery, signal, and volume. The taskbar at the bottom has icons for the Dash, Home, and other applications.

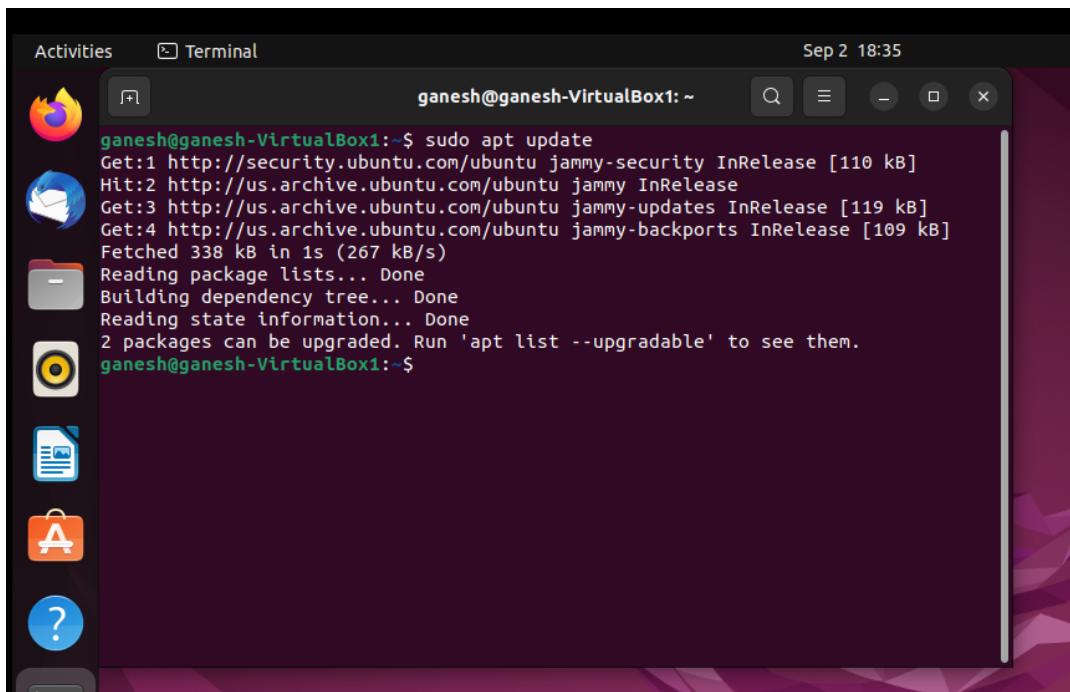
```
ganesh@ganesh-VirtualBox1: ~$ sudo fdisk /dev/sda
[sudo] password for ganesh:

Welcome to fdisk (util-linux 2.37.2).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

This disk is currently in use - repartitioning is probably a bad idea.
It's recommended to umount all file systems, and swapoff all swap
partitions on this disk.

Command (m for help):
```

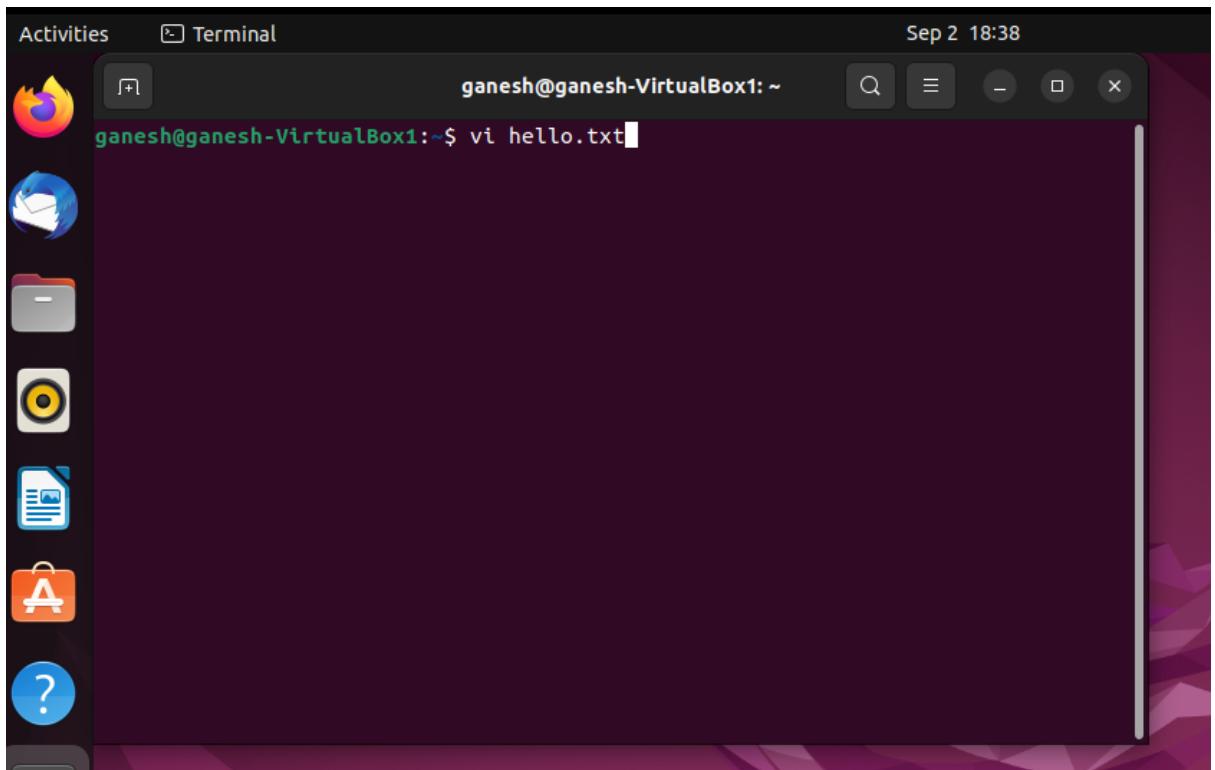
2.i. apt – it is used to manage updates for Linux systems, software packages. apt is a package management tool



A screenshot of an Ubuntu desktop environment. On the left is a dock with icons for Dash, Home, Applications, Files, and Help. In the center is a terminal window titled "Terminal" with the command "ganesh@ganesh-VirtualBox1: ~". The terminal output shows the execution of "sudo apt update", which fetches packages from various repositories and finds 2 upgradeable packages.

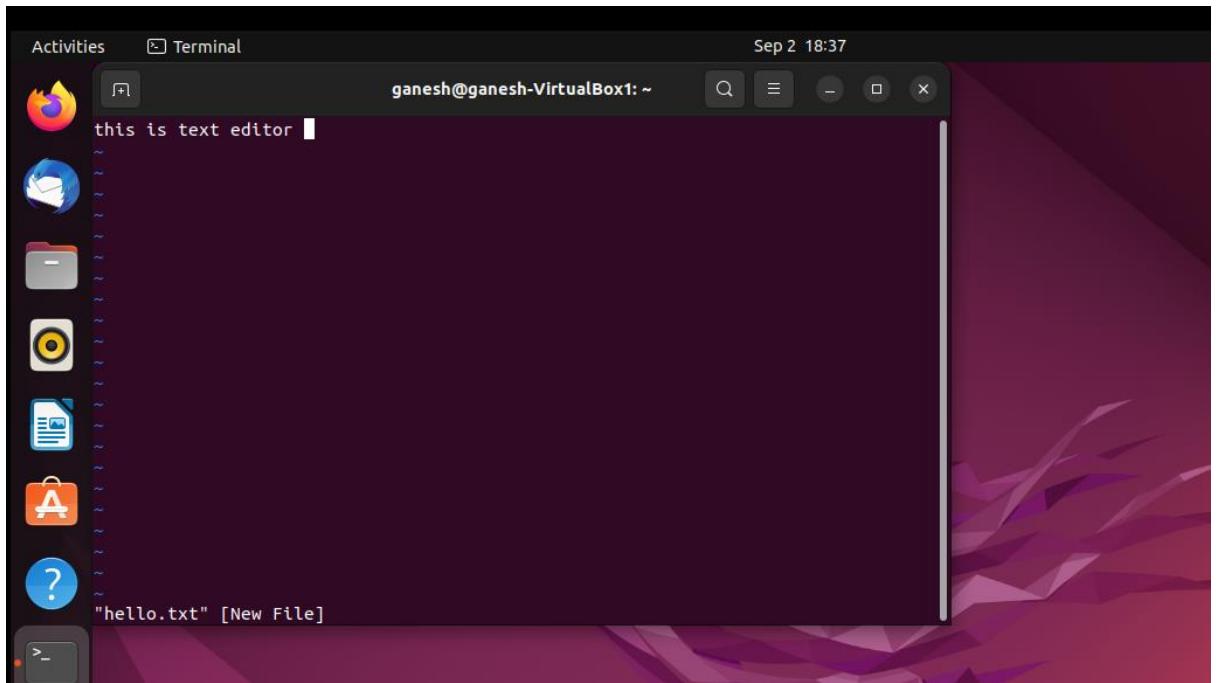
```
ganesh@ganesh-VirtualBox1:~$ sudo apt update
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Hit:2 http://us.archive.ubuntu.com/ubuntu jammy InRelease
Get:3 http://us.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Fetched 338 kB in 1s (267 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
2 packages can be upgraded. Run 'apt list --upgradable' to see them.
ganesh@ganesh-VirtualBox1:~$
```

2.j. vi- this is text editor used to edit content of files.

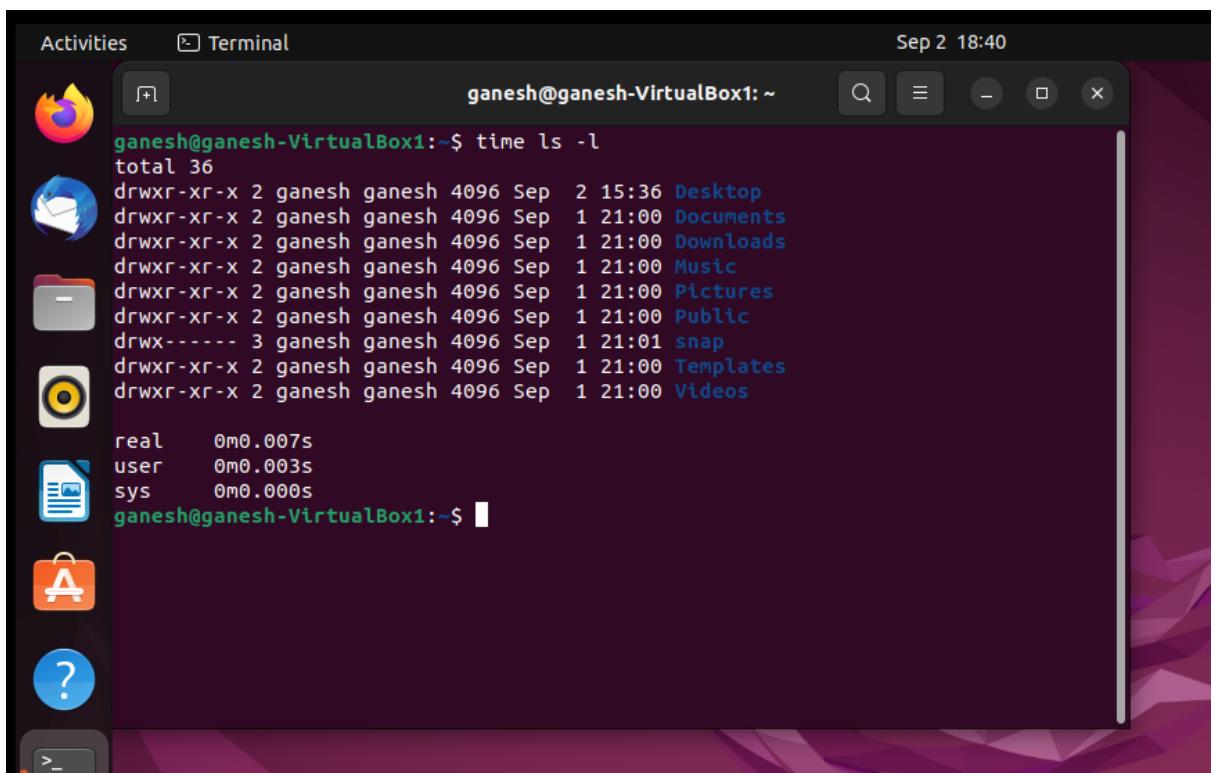


A screenshot of an Ubuntu desktop environment. On the left is a dock with icons for Dash, Home, Applications, Files, and Help. In the center is a terminal window titled "Terminal" with the command "ganesh@ganesh-VirtualBox1: ~\$ vi hello.txt". The terminal is in insert mode, indicated by the cursor icon.

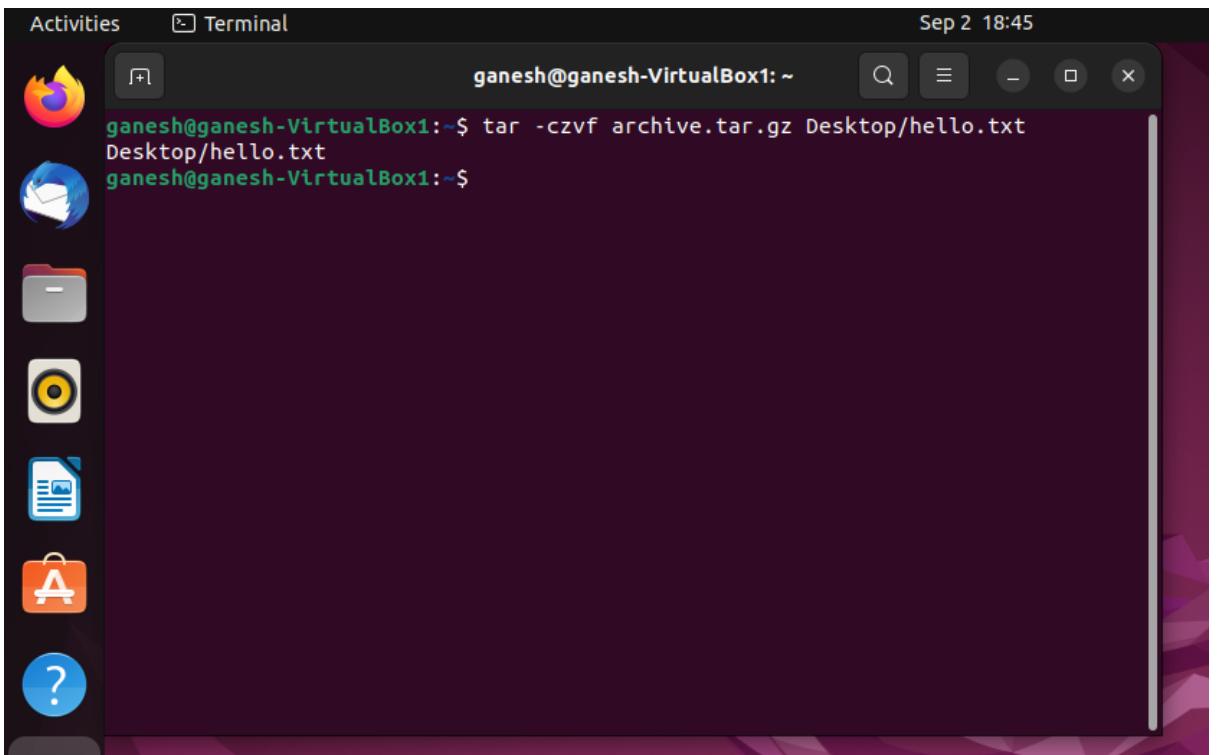
```
ganesh@ganesh-VirtualBox1:~$ vi hello.txt
```



2.k. time- it is used to measure the time taken to execute a particular command, which includes system time to too.



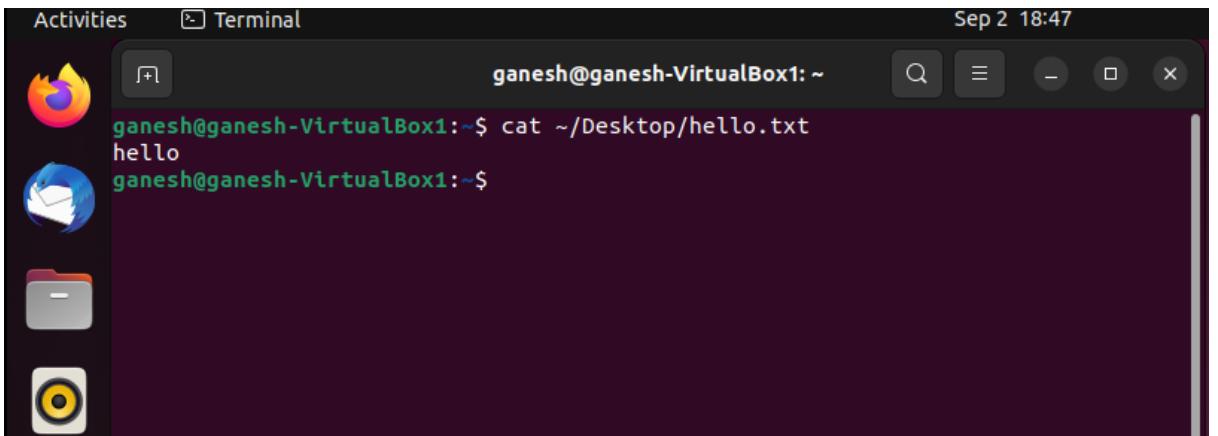
2.l. tar – is used to create compress and archive files



A screenshot of a Linux desktop environment. On the left is a vertical dock with icons for the Dash, Home, Applications, and Help. The main window is a terminal titled "Terminal" with the command "ganesh@ganesh-VirtualBox1: ~". The terminal output shows:

```
ganesh@ganesh-VirtualBox1:~$ tar -czvf archive.tar.gz Desktop/hello.txt
Desktop/hello.txt
ganesh@ganesh-VirtualBox1:~$
```

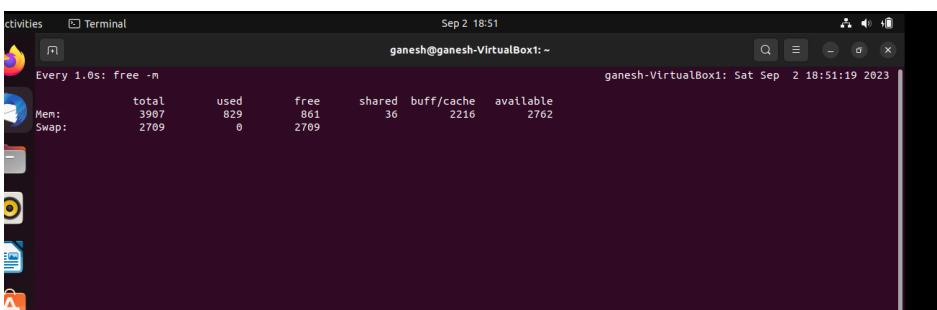
2.m. cat – this command displays the contents of the file or concatenate multiple files



A screenshot of a Linux desktop environment. On the left is a vertical dock with icons for the Dash, Home, Applications, and Help. The main window is a terminal titled "Terminal" with the command "ganesh@ganesh-VirtualBox1: ~". The terminal output shows:

```
ganesh@ganesh-VirtualBox1:~$ cat ~/Desktop/hello.txt
hello
ganesh@ganesh-VirtualBox1:~$
```

2.n. watch- it is used to repeatedly execute a command and display result in a terminal window

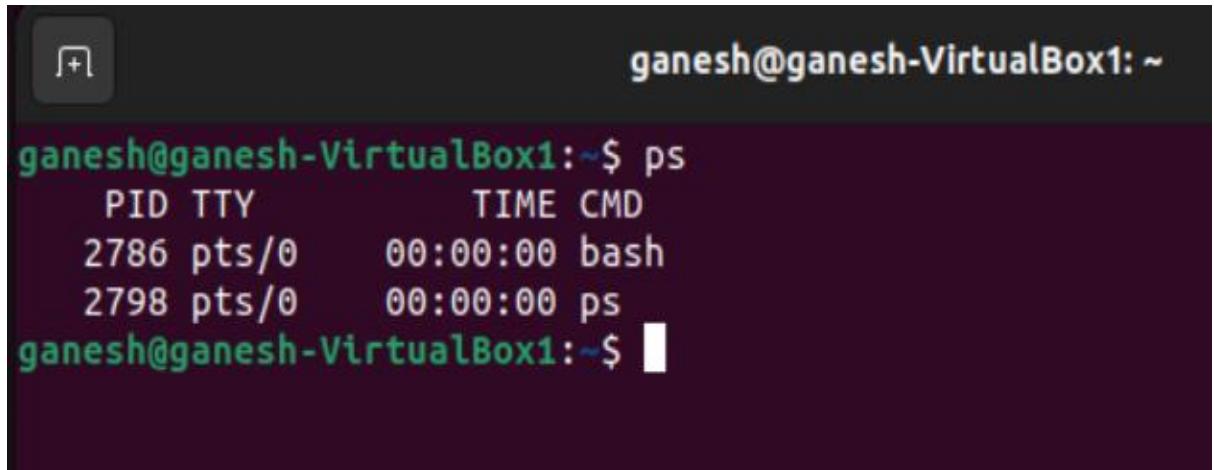


A screenshot of a Linux desktop environment. On the left is a vertical dock with icons for the Dash, Home, Applications, and Help. The main window is a terminal titled "Terminal" with the command "ganesh@ganesh-VirtualBox1: ~". The terminal output shows:

```
Every 1.0s: free -m
              total        used        free      shared  buff/cache   available
Mem:       3907         829         861          36        2216        2762
Swap:      2709          0         2709

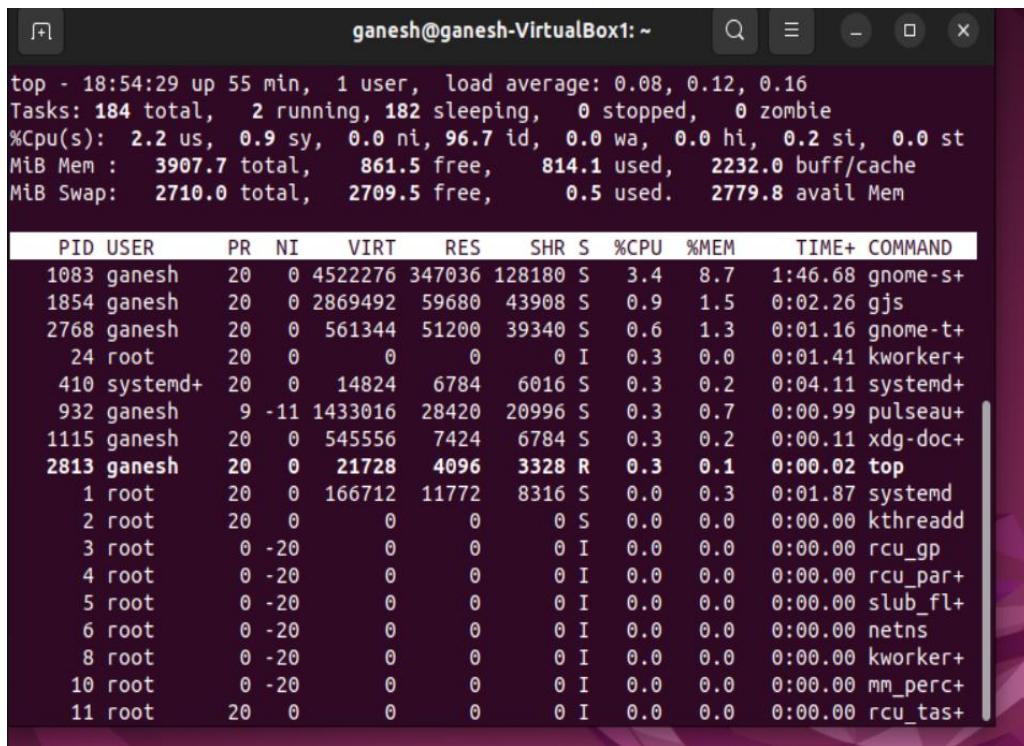
ganesh@ganesh-VirtualBox1:~$
```

2.o. ps – ps lists all running process.



```
ganesh@ganesh-VirtualBox1:~$ ps
  PID TTY          TIME CMD
 2786 pts/0    00:00:00 bash
 2798 pts/0    00:00:00 ps
ganesh@ganesh-VirtualBox1:~$
```

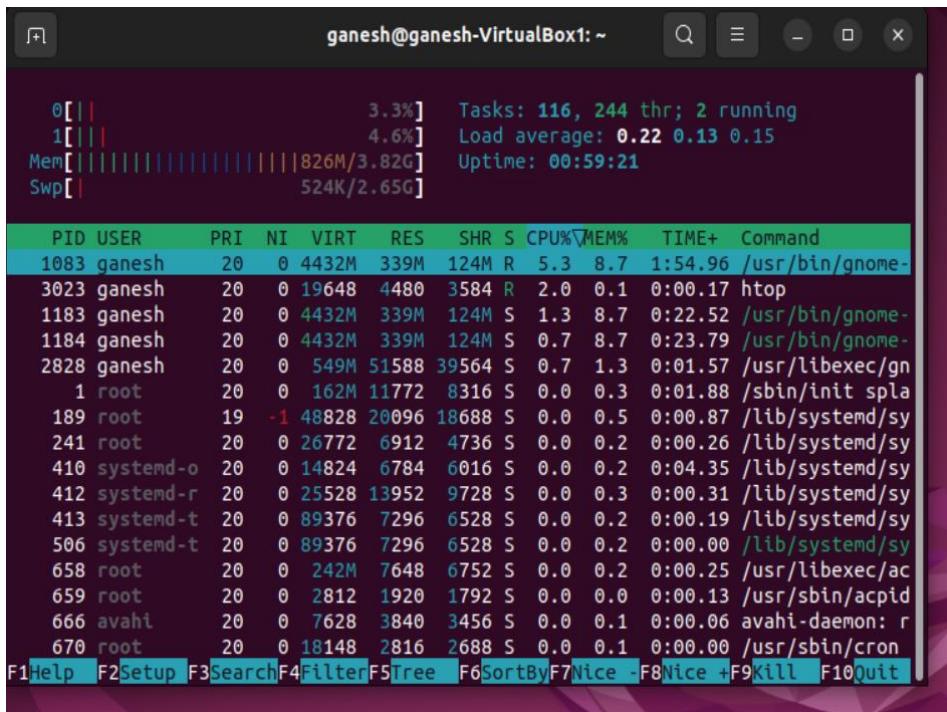
2.p. top – get resource usage details in real time.



```
top - 18:54:29 up 55 min,  1 user,  load average: 0.08, 0.12, 0.16
Tasks: 184 total,   2 running, 182 sleeping,   0 stopped,   0 zombie
%CPU(s): 2.2 us, 0.9 sy, 0.0 ni, 96.7 id, 0.0 wa, 0.0 hi, 0.2 si, 0.0 st
MiB Mem : 3907.7 total,   861.5 free,   814.1 used, 2232.0 buff/cache
MiB Swap: 2710.0 total,  2709.5 free,      0.5 used. 2779.8 avail Mem

      PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM TIME+ COMMAND
 1083 ganesh    20   0 4522276 347036 128180 S  3.4  8.7 1:46.68 gnome-s+
 1854 ganesh    20   0 2869492  59680  43908 S  0.9  1.5  0:02.26 gjs
 2768 ganesh    20   0  561344  51200  39340 S  0.6  1.3  0:01.16 gnome-t+
 24 root       20   0      0      0      0 I  0.3  0.0  0:01.41 kworker+
 410 systemd+  20   0  14824   6784   6016 S  0.3  0.2  0:04.11 systemd+
 932 ganesh     9 -11 1433016 28420 20996 S  0.3  0.7  0:00.99 pulseau+
1115 ganesh    20   0  545556   7424   6784 S  0.3  0.2  0:00.11 xdg-doc+
2813 ganesh    20   0  21728  4096  3328 R  0.3  0.1  0:00.02 top
 1 root       20   0 166712 11772   8316 S  0.0  0.3  0:01.87 systemd
 2 root       20   0      0      0      0 S  0.0  0.0  0:00.00 kthreadd
 3 root       0 -20      0      0      0 I  0.0  0.0  0:00.00 rcu_gp
 4 root       0 -20      0      0      0 I  0.0  0.0  0:00.00 rcu_par+
 5 root       0 -20      0      0      0 I  0.0  0.0  0:00.00 slub_f+
 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+
10 root      0 -20      0      0      0 I  0.0  0.0  0:00.00 mm_perc+
11 root      20   0      0      0      0 I  0.0  0.0  0:00.00 rcu_tas+
```

2.q. htop – provides interactive view of the system processes for monitoring.

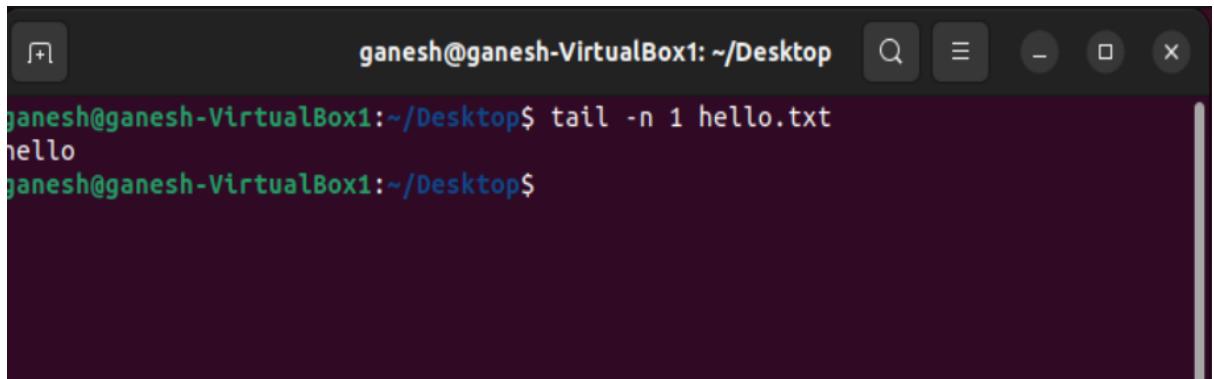


2.r. gcc – gcc is a compiler, used to compile languages including C /C++ codes.

The screenshot shows a terminal window with the following command and output:

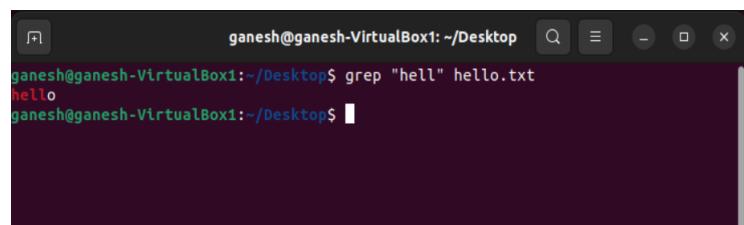
```
janesh@ganesh-VirtualBox1:~/Desktop$ gcc -o myoutup mycode.c
janesh@ganesh-VirtualBox1:~/Desktop$
```

2.s. tail – it is used to display last lines of a text file



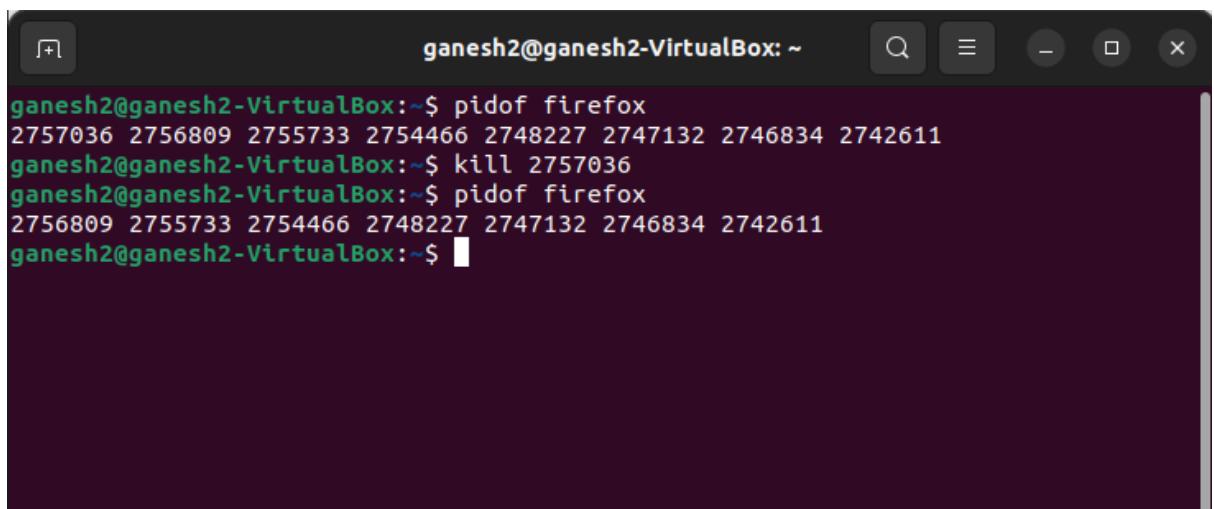
```
ganesh@ganesh-VirtualBox1:~/Desktop$ tail -n 1 hello.txt
hello
ganesh@ganesh-VirtualBox1:~/Desktop$
```

2.t. grep – search for a string pattern in a text file.



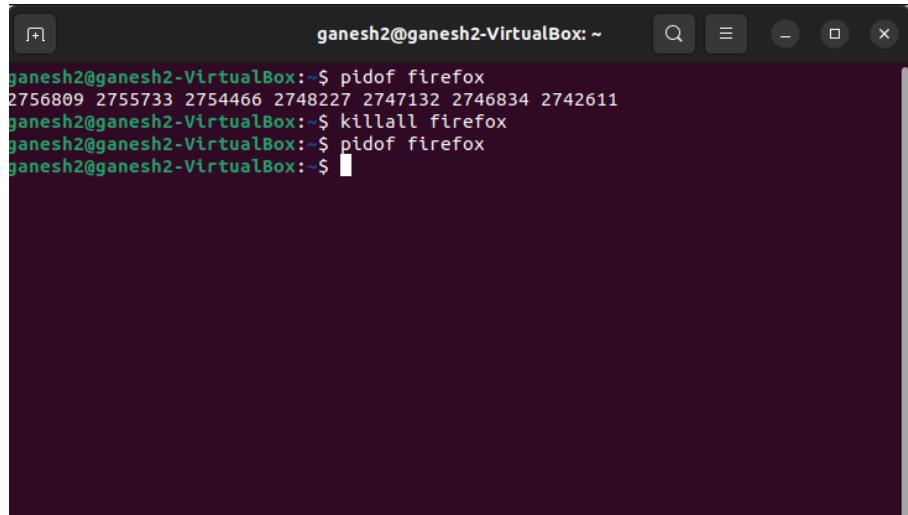
```
ganesh@ganesh-VirtualBox1:~/Desktop$ grep "hell" hello.txt
hello
ganesh@ganesh-VirtualBox1:~/Desktop$
```

2.u. kill – to end a process with the help of a process ID



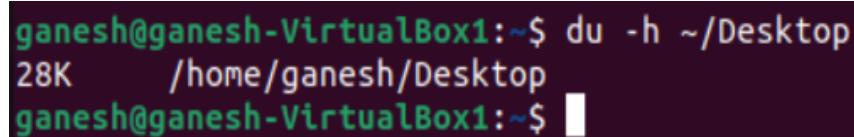
```
ganesh2@ganesh2-VirtualBox:~$ pidof firefox
2757036 2756809 2755733 2754466 2748227 2747132 2746834 2742611
ganesh2@ganesh2-VirtualBox:~$ kill 2757036
ganesh2@ganesh2-VirtualBox:~$ pidof firefox
2756809 2755733 2754466 2748227 2747132 2746834 2742611
ganesh2@ganesh2-VirtualBox:~$
```

2.v. killall – similar to kill, used to terminate a process by name



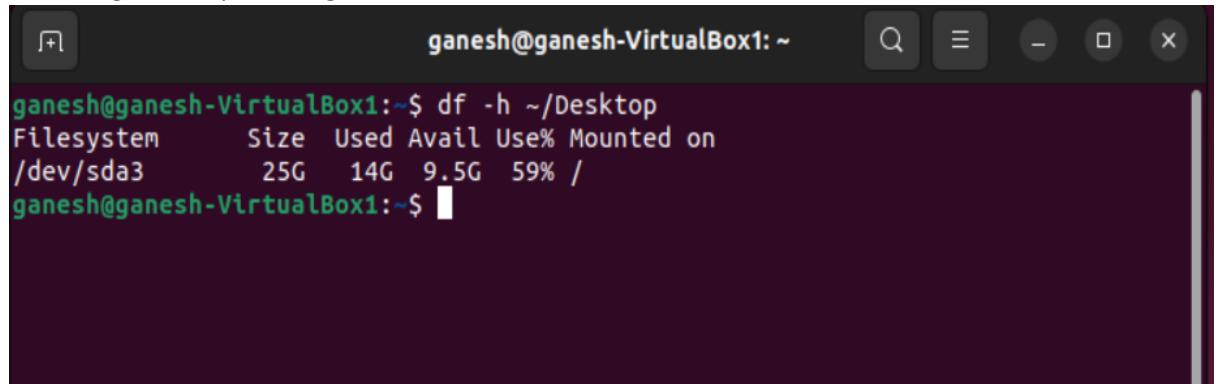
```
ganesh2@ganesh2-VirtualBox:~$ pidof firefox
2756809 2755733 2754466 2748227 2747132 2746834 2742611
ganesh2@ganesh2-VirtualBox:~$ killall firefox
ganesh2@ganesh2-VirtualBox:~$ pidof firefox
ganesh2@ganesh2-VirtualBox:~$
```

2.w. du – get current disk usage status



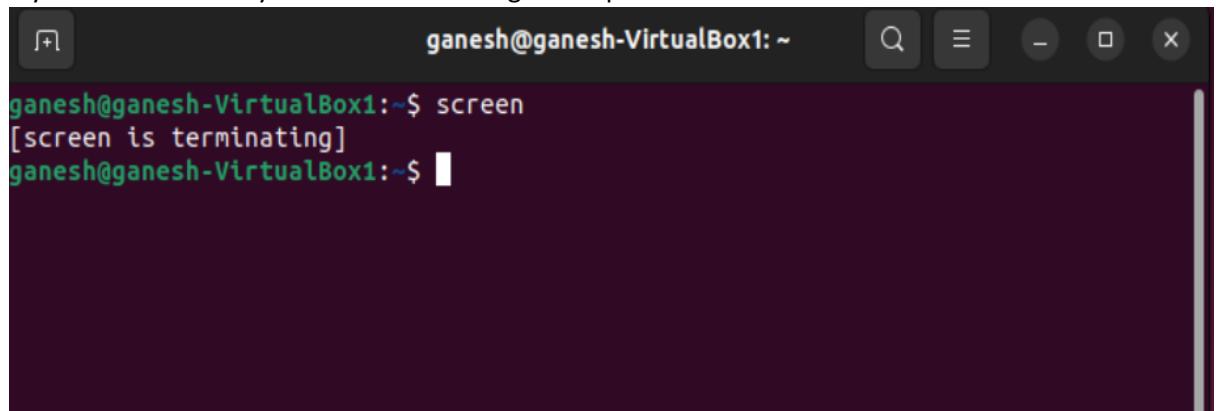
```
ganesh@ganesh-VirtualBox1:~$ du -h ~/Desktop
28K      /home/ganesh/Desktop
ganesh@ganesh-VirtualBox1:~$
```

2.x. df – get disk space usage



```
ganesh@ganesh-VirtualBox1:~$ df -h ~/Desktop
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda3        25G   14G   9.5G  59% /
ganesh@ganesh-VirtualBox1:~$
```

2.y. screen – flexibility to create and manage multiple terminal sessions



```
ganesh@ganesh-VirtualBox1:~$ screen
[screen is terminating]
ganesh@ganesh-VirtualBox1:~$
```

2.z. vim – vim is a text editor

The screenshot shows a terminal window with a dark background. At the top, the title bar reads "ganesh@ganesh-VirtualBox1: ~/Desktop". The command line shows two commands: "cd ~/Desktop" and "Vim hello.txt". Below the command line, the Vim editor is open, displaying the word "hello" on a single line. The status bar at the bottom indicates the file is "hello.txt" [readonly] with 1L, 6B, and the cursor is at 1,1. The Vim interface includes standard navigation keys like h, j, k, l, and movement keys like gg, G, and f.

2.aa. chmod – provide/change file permission such as read, write, execute to owner, user, group

The screenshot shows a terminal window with a dark background. The title bar reads "ganesh@ganesh-VirtualBox1: ~/Desktop". The command "chmod 644 mycode.c" is entered and executed. The command line ends with a "\$" prompt.

2.bb. chown – update ownership/group for a particular file/directory.

The screenshot shows a terminal window with a dark background. The title bar reads "ganesh@ganesh-VirtualBox1: ~/Desktop". The command "chown ganesh mycode.c" is entered and executed. The command line ends with a "\$" prompt.

2.cc. useradd – create a new user profile

A screenshot of a terminal window titled "Firefox Web Browser". The command entered is "sudo useradd roger". A password prompt "[sudo] password for ganesh:" is displayed. The terminal window has a dark background with light-colored text.

```
ganesh@ganesh-VirtualBox1: ~/Desktop
[sudo] password for ganesh:
ganesh@ganesh-VirtualBox1: ~/Desktop$
```

2.dd. man – to display manual/more information for a command and much more.

A screenshot of a terminal window titled "User Commands". The command entered is "man ls". The output shows the manual page for the ls command, including its NAME, SYNOPSIS, DESCRIPTION, and various options like -a, -A, -b, -c, -l, -t, and --color. The terminal window has a dark background with light-colored text.

```
ls(1)                               User Commands                               LS(1)

NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       List information about the FILES (the current directory by default). Sort entries alphabetically if none of -cftuvSUX
       nor --sort is specified.

       Mandatory arguments to long options are mandatory for short options too.

       -a, --all
              do not ignore entries starting with .

       -A, --almost-all
              do not list implied . and ..

       --author
              with -l, print the author of each file

       -b, --escape
              print C-style escapes for nongraphic characters

       --block-size=SIZE
              with -l, scale sizes by SIZE when printing them; e.g., '--block-size=M'; see SIZE format below

       -B, --ignore-backups
              do not list implied entries ending with -

       -c      with -lt: sort by, and show, ctime (time of last modification of file status information); with -l: show ctime and
              sort by name; otherwise: sort by ctime, newest first

       -C      list entries by columns

       --color[=WHEN]
              colorize the output; WHEN can be 'always' (default if omitted), 'auto', or 'never'; more info below

Manual page ls(1) line 1 (press h for help or q to quit)
```

2.ee. locate – quickly find items by its name

A screenshot of a terminal window titled "ganesh@ganesh-VirtualBox1: ~". The command entered is "locate hello.txt". The output shows two file paths: "/home/ganesh/.hello.txt.swp" and "/home/ganesh/Desktop/hello.txt". The terminal window has a dark background with light-colored text.

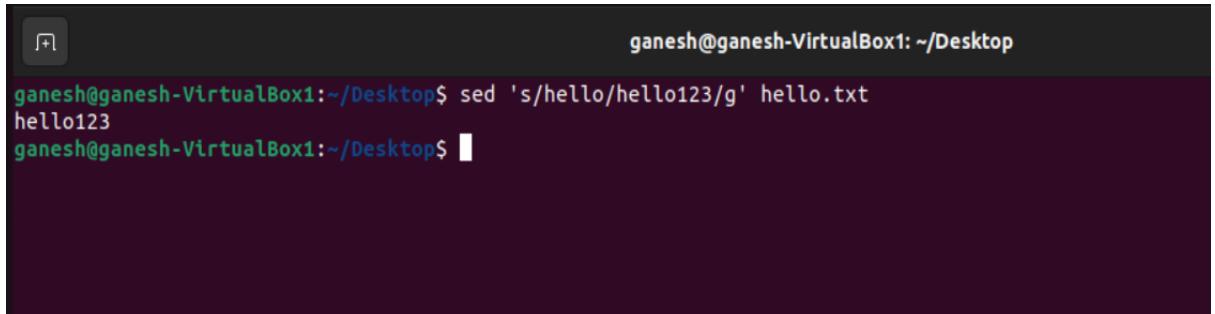
```
ganesh@ganesh-VirtualBox1:~$ locate hello.txt
/home/ganesh/.hello.txt.swp
/home/ganesh/Desktop/hello.txt
ganesh@ganesh-VirtualBox1:~$
```

2.ff. find – this command helps to find files and directories.

A screenshot of a terminal window titled "ganesh@ganesh-VirtualBox1: ~". The command entered is "find ~/Desktop/ -name hello.txt". The output shows the single file path "/home/ganesh/Desktop/hello.txt". The terminal window has a dark background with light-colored text.

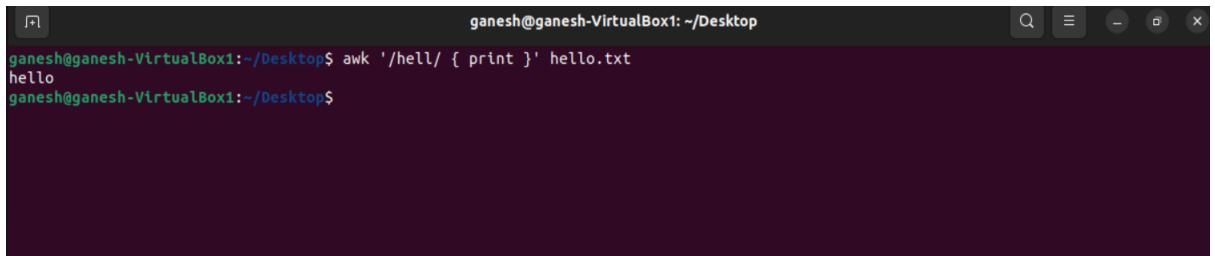
```
ganesh@ganesh-VirtualBox1:~/Desktop/
find ~/Desktop/ -name hello.txt
/home/ganesh/Desktop/hello.txt
ganesh@ganesh-VirtualBox1:~/Desktop/$
```

2.gg. sed – stands stream editor used for string manipulation and used perform text related operations



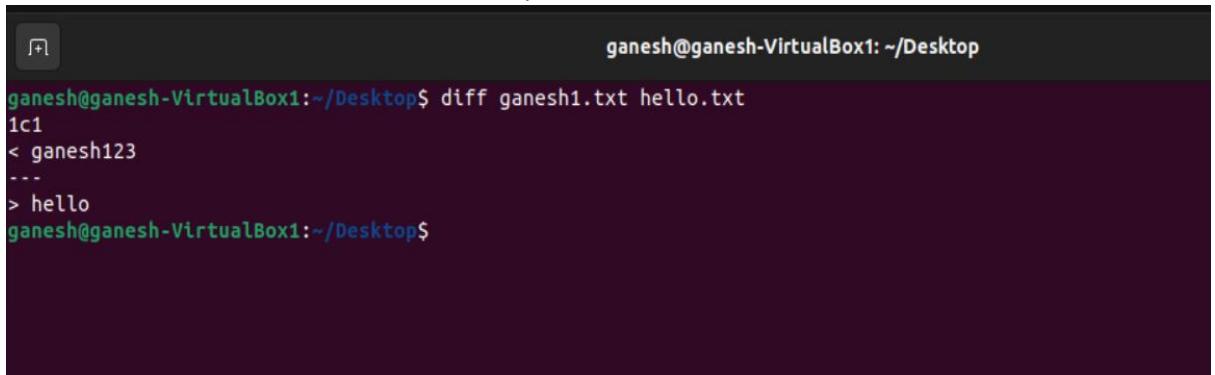
```
ganesh@ganesh-VirtualBox1:~/Desktop$ sed 's/hello/hello123/g' hello.txt
hello123
ganesh@ganesh-VirtualBox1:~/Desktop$
```

2.hh. awk – this is used for text analysis/manipulation like pattern matching, filtering data from files



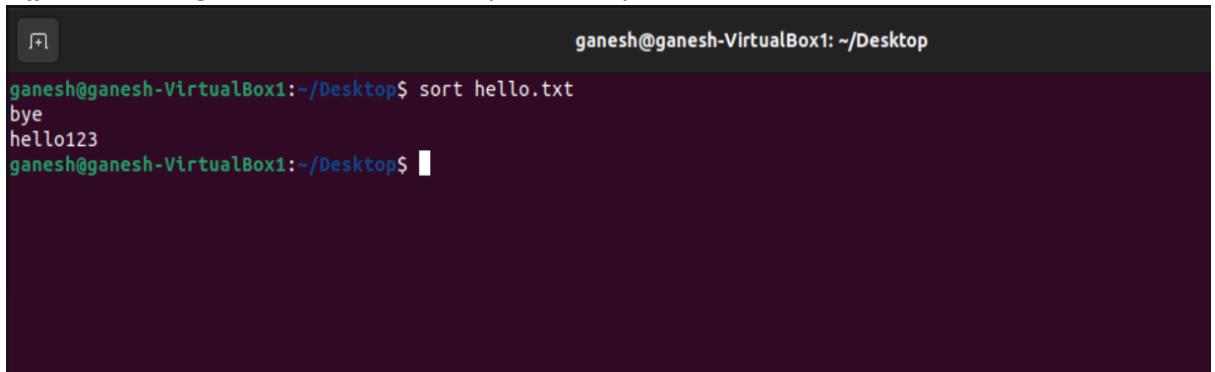
```
ganesh@ganesh-VirtualBox1:~/Desktop$ awk '/hell/ { print }' hello.txt
hello
ganesh@ganesh-VirtualBox1:~/Desktop$
```

2.ii. diff – this command is used for file comparison.



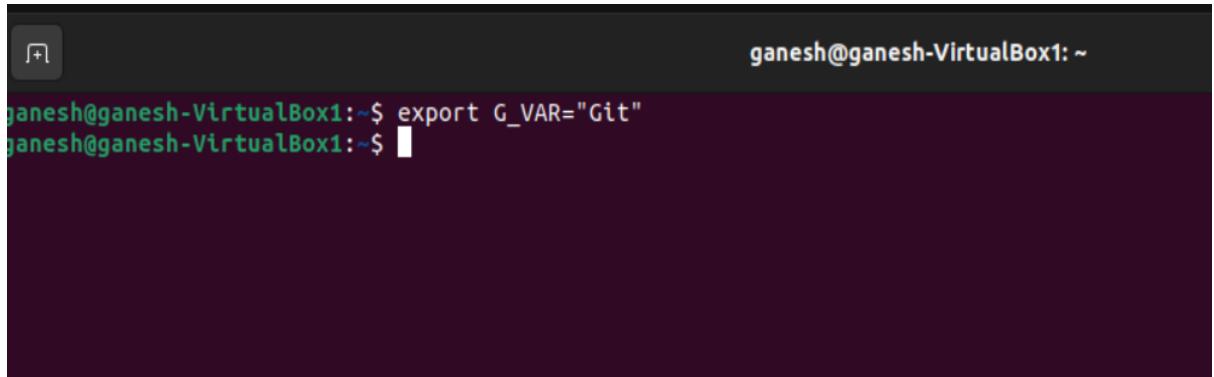
```
ganesh@ganesh-VirtualBox1:~/Desktop$ diff ganesh1.txt hello.txt
1c1
< ganesh123
---
> hello
ganesh@ganesh-VirtualBox1:~/Desktop$
```

2.jj. sort – arrange the lines of the text alphabetically



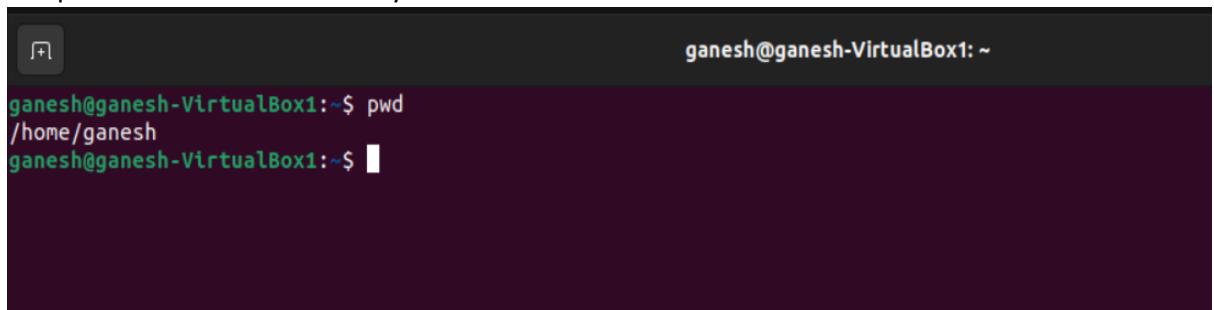
```
ganesh@ganesh-VirtualBox1:~/Desktop$ sort hello.txt
bye
hello123
ganesh@ganesh-VirtualBox1:~/Desktop$
```

2.kk. export – it is used to set environment variables and make the same available for child processes too.



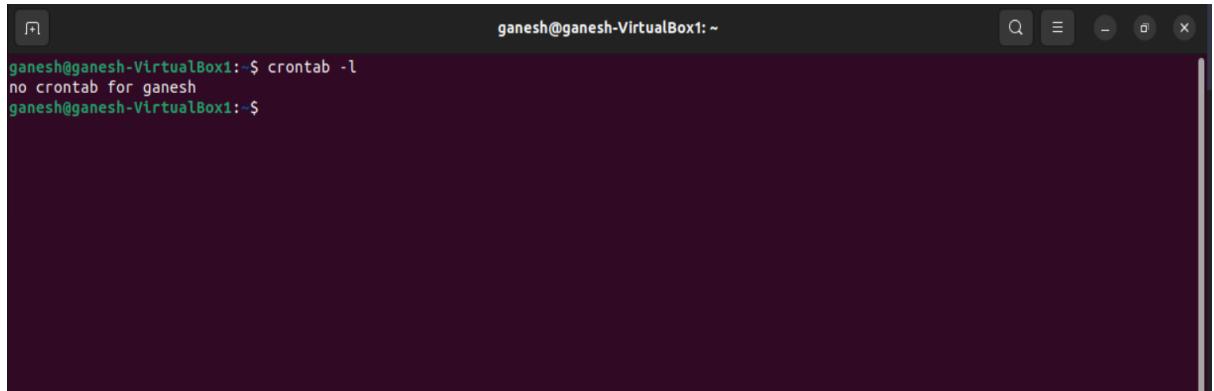
```
ganesh@ganesh-VirtualBox1:~$ export G_VAR="Git"
ganesh@ganesh-VirtualBox1:~$
```

2.ll. pwd – fetch current directory



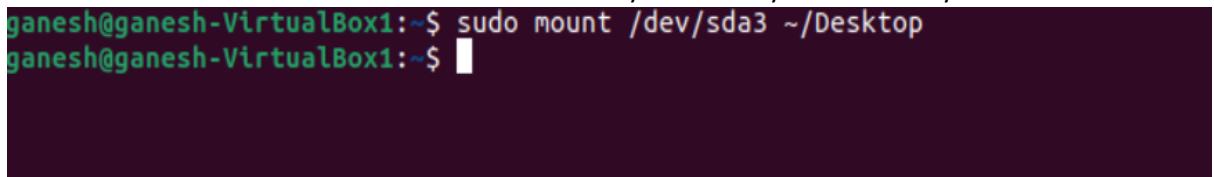
```
ganesh@ganesh-VirtualBox1:~$ pwd
/home/ganesh
ganesh@ganesh-VirtualBox1:~$
```

2.mm. crontab – used to schedule and manage tasks that execute at configured intervals on a regular basis.



```
ganesh@ganesh-VirtualBox1:~$ crontab -l
no crontab for ganesh
ganesh@ganesh-VirtualBox1:~$
```

2.nn. mount – this command is used to attach a filesystem to a system directory.



```
ganesh@ganesh-VirtualBox1:~$ sudo mount /dev/sda3 ~/Desktop
ganesh@ganesh-VirtualBox1:~$
```

2.oo. passwd – used to change user account password.

```
ganesh@ganesh-VirtualBox1:~$ passwd ganesh
Changing password for ganesh.
Current password:
New password:
```

2.pp. uname – to display information about the system.

```
ganesh@ganesh-VirtualBox1:~$ uname -a
Linux ganesh-VirtualBox1 6.2.0-31-generic #31~22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Wed Aug 16 13:45:26 UTC 2023 x86_64 x86_64 x86_64 GNU/Linux
ganesh@ganesh-VirtualBox1:~$
```

2.qq. whereis – this helps in getting the binary, documentation, source code for a command

```
ganesh@ganesh-VirtualBox1:~$ whereis ls
ls: /usr/bin/ls /usr/share/man/man1/ls.1.gz
ganesh@ganesh-VirtualBox1:~$
```

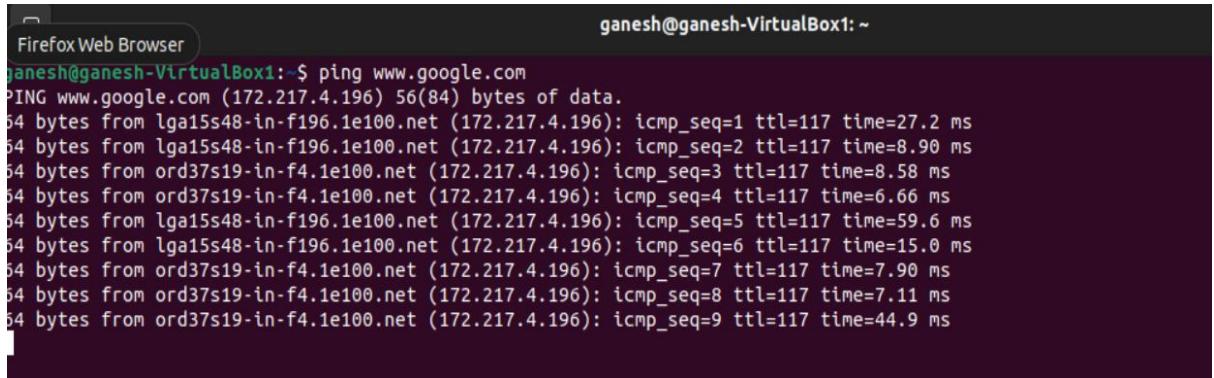
2.rr. whatis – gives a brief description of the command given

```
ganesh@ganesh-VirtualBox1:~$ whatis ls
ls (1)           - list directory contents
ganesh@ganesh-VirtualBox1:~$
```

2.ss. su – used to switch to another available user

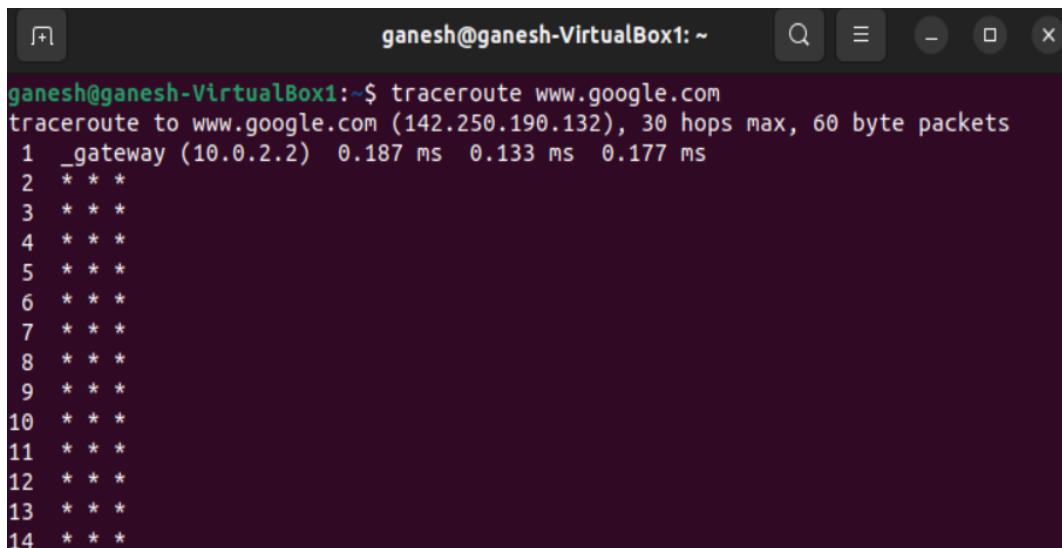
```
ganesh@ganesh-VirtualBox1:~$ su - roger
Password:
```

2.tt. ping – used for testing network connectivity



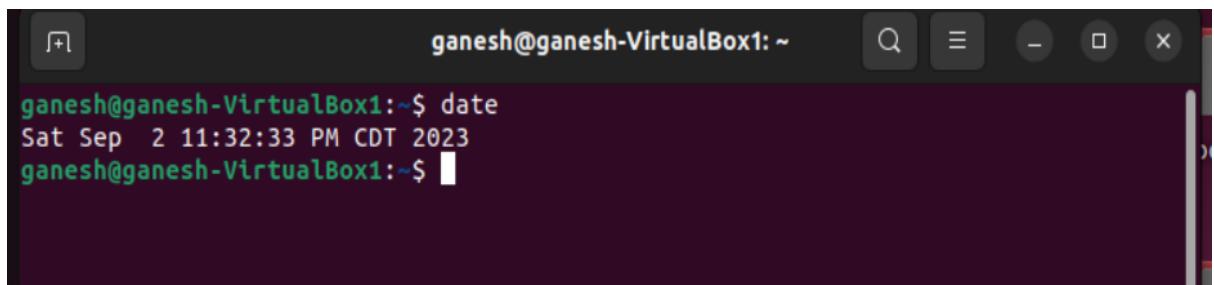
```
ganesh@ganesh-VirtualBox1:~$ ping www.google.com
PING www.google.com (172.217.4.196) 56(84) bytes of data.
64 bytes from lga15s48-in-f196.1e100.net (172.217.4.196): icmp_seq=1 ttl=117 time=27.2 ms
64 bytes from lga15s48-in-f196.1e100.net (172.217.4.196): icmp_seq=2 ttl=117 time=8.90 ms
64 bytes from ord37s19-in-f4.1e100.net (172.217.4.196): icmp_seq=3 ttl=117 time=8.58 ms
64 bytes from ord37s19-in-f4.1e100.net (172.217.4.196): icmp_seq=4 ttl=117 time=6.66 ms
64 bytes from lga15s48-in-f196.1e100.net (172.217.4.196): icmp_seq=5 ttl=117 time=59.6 ms
64 bytes from lga15s48-in-f196.1e100.net (172.217.4.196): icmp_seq=6 ttl=117 time=15.0 ms
64 bytes from ord37s19-in-f4.1e100.net (172.217.4.196): icmp_seq=7 ttl=117 time=7.90 ms
64 bytes from ord37s19-in-f4.1e100.net (172.217.4.196): icmp_seq=8 ttl=117 time=7.11 ms
64 bytes from ord37s19-in-f4.1e100.net (172.217.4.196): icmp_seq=9 ttl=117 time=44.9 ms
```

2.uu. traceroute – used to track packets and the hops that is taken from source to destination host in a network



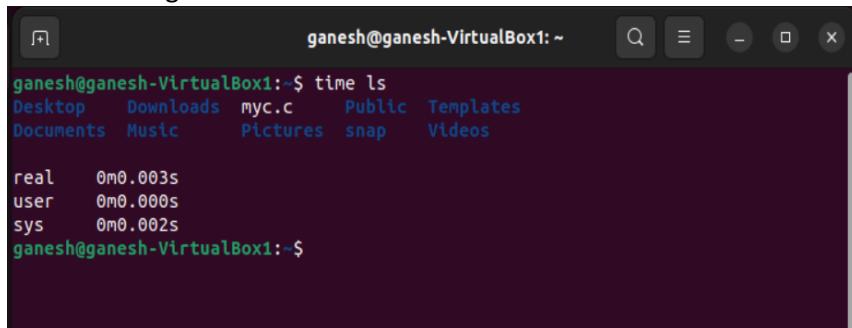
```
ganesh@ganesh-VirtualBox1:~$ traceroute www.google.com
traceroute to www.google.com (142.250.190.132), 30 hops max, 60 byte packets
 1 _gateway (10.0.2.2)  0.187 ms  0.133 ms  0.177 ms
 2 * * *
 3 * * *
 4 * * *
 5 * * *
 6 * * *
 7 * * *
 8 * * *
 9 * * *
10 * * *
11 * * *
12 * * *
13 * * *
14 * * *
```

2.vv. date – fetch current date and time



```
ganesh@ganesh-VirtualBox1:~$ date
Sat Sep  2 11:32:33 PM CDT 2023
ganesh@ganesh-VirtualBox1:~$
```

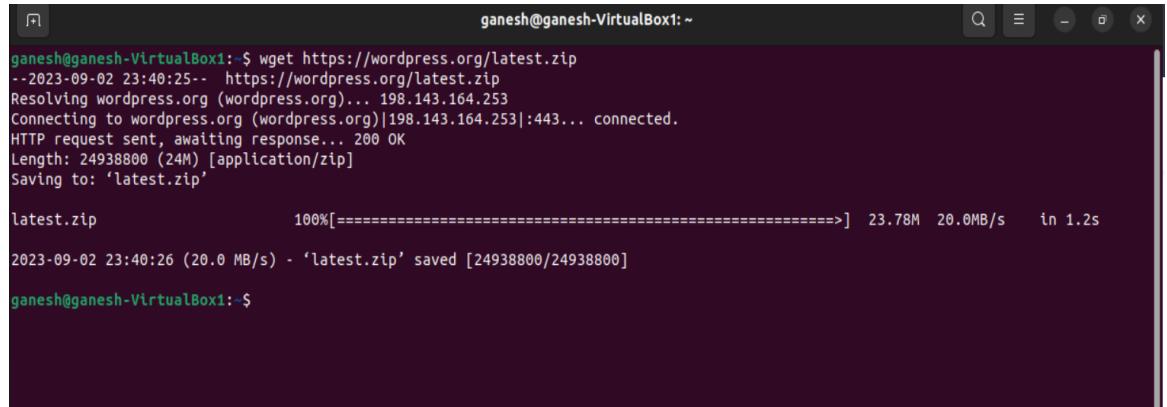
2.ww. time – get the command execution time



```
ganesh@ganesh-VirtualBox1:~$ time ls
Desktop  Downloads  myc.c    Public  Templates
Documents  Music      Pictures  snap     Videos

real    0m0.003s
user    0m0.000s
sys     0m0.002s
ganesh@ganesh-VirtualBox1:~$
```

2.xx. wget – download files over the internet, web servers



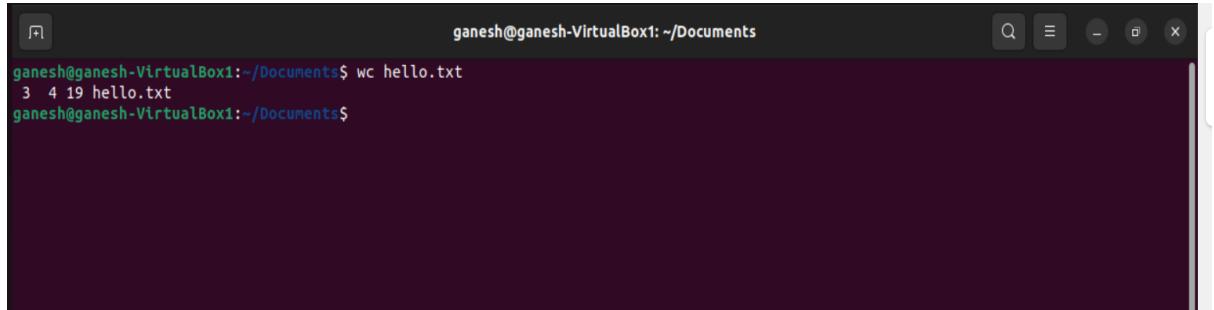
```
ganesh@ganesh-VirtualBox1:~$ wget https://wordpress.org/latest.zip
--2023-09-02 23:40:25-- https://wordpress.org/latest.zip
Resolving wordpress.org (wordpress.org)... 198.143.164.253
Connecting to wordpress.org (wordpress.org)|198.143.164.253|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 24938800 (24M) [application/zip]
Saving to: 'latest.zip'

latest.zip          100%[=====] 23.78M  20.0MB/s   in 1.2s

2023-09-02 23:40:26 (20.0 MB/s) - 'latest.zip' saved [24938800/24938800]

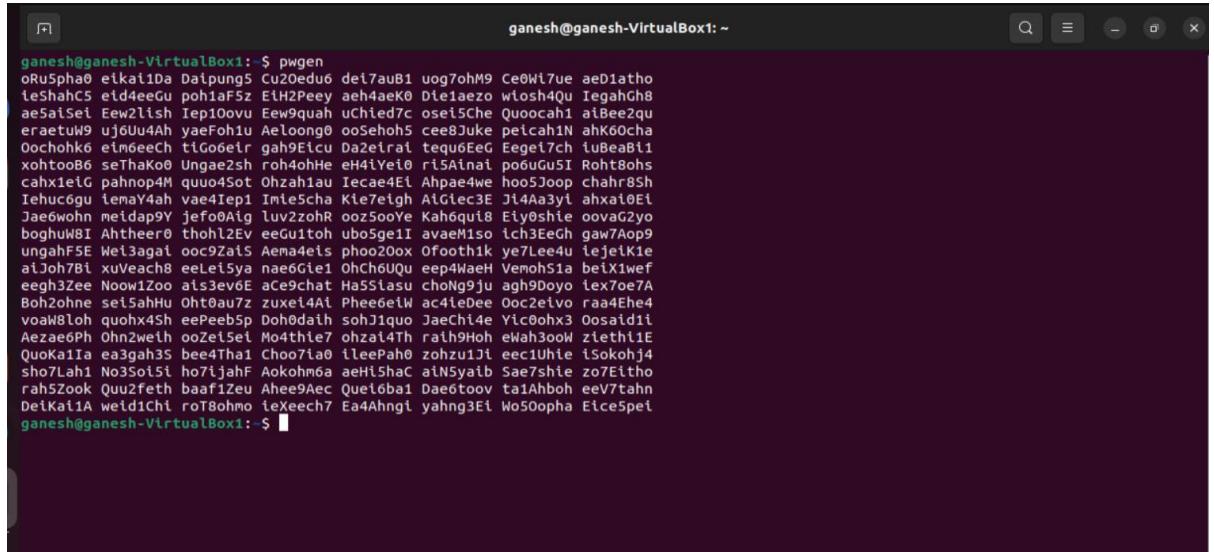
ganesh@ganesh-VirtualBox1:~$
```

2.yy. wc – used to count the number of lines, words, total characters in the file specified.



```
ganesh@ganesh-VirtualBox1:~/Documents$ wc hello.txt
3 4 19 hello.txt
ganesh@ganesh-VirtualBox1:~/Documents$
```

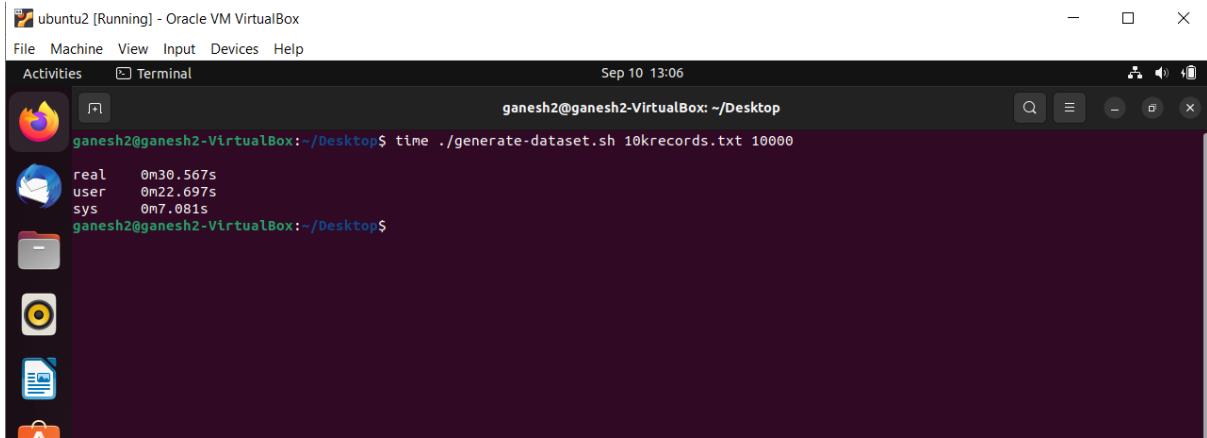
2.zz. pwgen – generate random passwords



```
ganesh@ganesh-VirtualBox1:~$ pwgen
oRuSpHa0 eikaiDa Daipung5 Cu20edu6 dei7auB1 uog7ohM9 Ce0Wi7ue aeDiatho
teShahC5 eid4eeGu poh1aF5z ElH2Peey aeh4aeK0 Die1aezo wlsh4Qu IegahGh8
ae5aiSei Eew2lish Iep1Oovu Eew9quah uchted7c osei5che Quocah1 aiBee2qu
eraetu9 uj6Uu4Ah yaeFoh1u Aeloong8 ooSehoh5 cee8Juke pelcah1N ahK6ocha
Oochokh6 eim6eech tlGodeir gah9Eicu Da2eirai tequ6EeG Eege17ch luBeaB1
xohtooB6 seThak08 Ungae2sh roh4ohHe eH4iYe10 riSAinal po6guG51 RohtBohs
cahx1elG pahnop4M quuo4Sot Ohzah1au Iecae4El Ahpaed4we hoo5Joop chahr8Sh
Iehuc6gu iemaY4ah vae4Iep1 Imie5cha KieTeigh AiGiec3E Jl4Aa3yl ahxa10Et
Jae6wohn meidap9Y jefo0Aig luvzzohR oozSooYe Kah6qui8 Ety0shie oovaG2yo
boghuw8I Ahtheero thohl2Ev eeGu1toh ubo5geII avaaeMiso Ich3EeGh gaw7Aop9
unghahF5E Wei3agai ooc9Zais Aema4ets phoo20ox Ofoothik ye7Lee4u ieje1k1e
aiJoh7Bi xuVeach8 eeLei5ya nae6cie1 OhCh6Uqu eep4WaeH Vemoh51a beix1wef
eegh3Zee Noow1Zoo als3ev6E aCe9chat Ha5Siisu choNg9ju agh9Doyo iex7oe7A
BohZohne se1sahHu Oht0au7z zuxet4At Phee6etW ac4leDee Ooc2etvo raa4he4
voakW8loh quohx45h eePeeb5p Doh0dah sohJ1quo JaeChi4c Yic0ohx3 Oosaidil
Aezae6Ph Ohn2weiH ooZei5ei Mo4thie7 ohzai4th ralh9Hoh ewah3ooW ziethi1E
Quoka1ia ea3gah3S bee4Tha1 Choo7ia8 illepah0 zohzuJ1i eec1Uhie isokohj4
sho7Lah1 No3so15i ho7ijahF Aokohm6a aeh15hac alN5yatl Sae7shie zo7Etho
rah5Zook Quu2feth baaf1zeu Ahee9Aec Quet6ba1 Dae6tovo t1Ahboh eeV7tahn
DetKaiIA weidichi ro78ohmo iexeech7 Ea4Ahngi yahng3El Wo50opoha Eice5pe1
ganesh@ganesh-VirtualBox1:~$
```

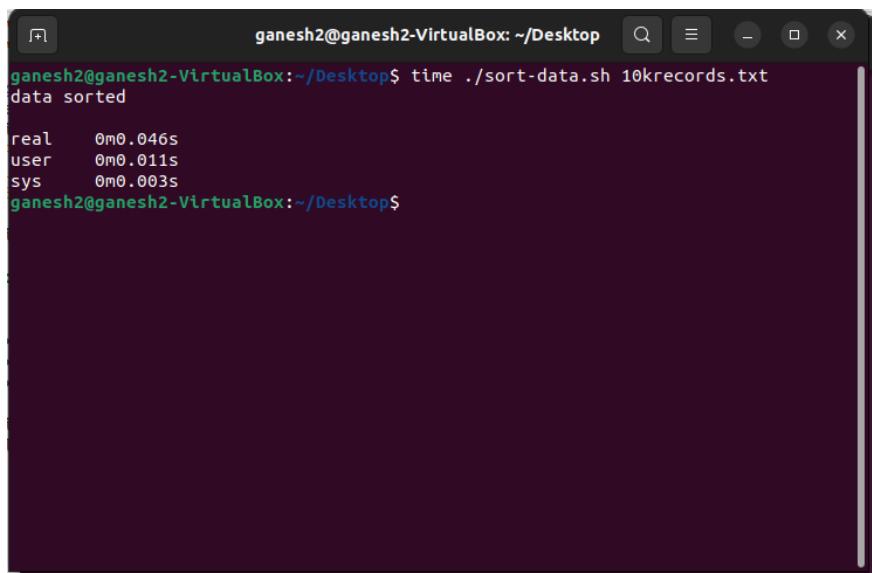
3.a. bash scripting

To generate 10,000 records it took around 30 seconds.



A screenshot of a terminal window titled "ubuntu2 [Running] - Oracle VM VirtualBox". The window shows a dark-themed desktop environment with icons for a browser, file manager, terminal, and others. The terminal itself has a light background. It displays the command "time ./generate-dataset.sh 10krecords.txt 10000" followed by its output: "real 0m30.567s user 0m22.697s sys 0m7.081s". The timestamp at the top right of the terminal window is "Sep 10 13:06".

3.b. sorting the data with 10,000 records took 0.046 seconds.



A screenshot of a terminal window titled "ganesh2@ganesh2-VirtualBox: ~/Desktop". The terminal shows the command "time ./sort-data.sh 10krecords.txt" followed by its output: "data sorted" and "real 0m0.046s user 0m0.011s sys 0m0.003s". The timestamp at the top right of the terminal window is "Sep 10 13:06".

3.c. to generate and sort 1000, 100000 and 10000000 records

```
ganesh2@ganesh2-VirtualBox:~/Desktop$ time ./generate-dataset.sh 1krecords.txt 1000
real    0m12.699s
user    0m9.277s
sys     0m3.053s
ganesh2@ganesh2-VirtualBox:~/Desktop$ time ./generate-dataset.sh 100krecords.txt 100000
real    5m30.109s
user    3m56.217s
sys     1m24.562s
ganesh2@ganesh2-VirtualBox:~/Desktop$ time ./generate-dataset.sh 10milrecords.txt 10000000

ganesh2@ganesh2-VirtualBox:~/Desktop$ time ./generate-dataset.sh 10milrecords.txt 100000
0

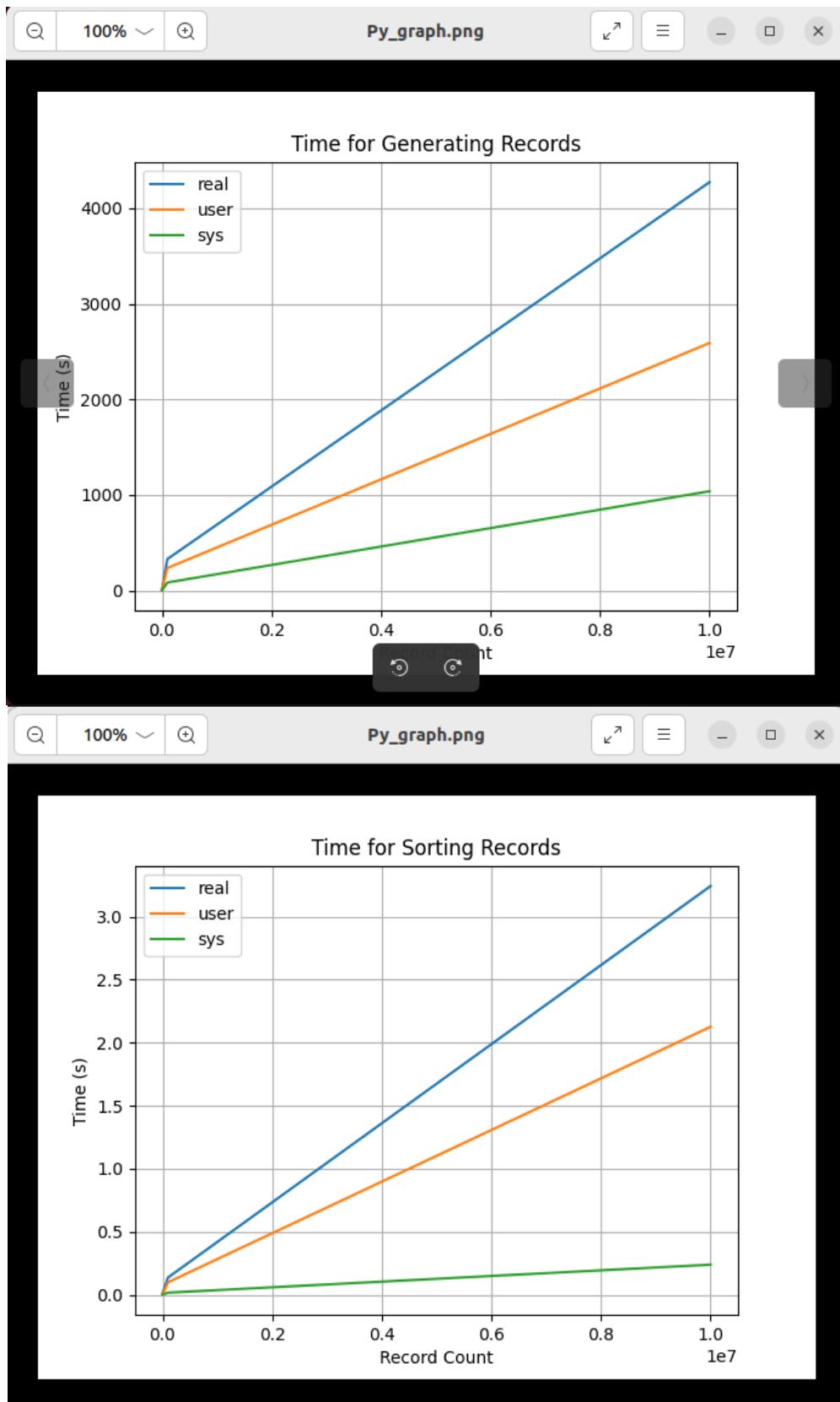
real    71m24.371s
user    43m18.865s
sys     17m32.824s
ganesh2@ganesh2-VirtualBox:~/Desktop$ time ./sort-data.sh 100krecords.txt
data sorted

real    0m0.138s
user    0m0.100s
sys     0m0.018s
ganesh2@ganesh2-VirtualBox:~/Desktop$ time ./sort-data.sh 10milrecords.txt
data sorted

real    0m3.242s
user    0m2.126s
sys     0m0.240s
ganesh2@ganesh2-VirtualBox:~/Desktop$ time ./sort-data.sh 1krecords.txt
data sorted

real    0m0.014s
user    0m0.005s
sys     0m0.003s
ganesh2@ganesh2-VirtualBox:~/Desktop$
```

Below images show the graph for generating and sorting records respectively.



4.a. In the system configuration of the VM, explain how changing the number of processors changes the behavior of your VM. Explain a scenario where you want to set this to the minimum, and a scenario where you want to set it to the maximum. Why is setting it to the maximum potentially a bad idea?

- Processors determine the computing speed of any given task and the count of cores reflect the ability to run parallel tasks. In some instances, we might have to keep the number of processors allocated to a VM as minimal as possible when the workload doesn't demand higher computing power so that even the host Operating System or other VMs can have significant CPU resources to run its processes. In addition to this, having minimum number of processors can help save costs when setting up a cloud environment. On the other hand, allocating maximum number of processors can help in high workload environments such that the workload can be parallelly distributed on multiple processors/threads thus assisting in reducing time to complete a task. An example of this type could be for graphics processing. But allocating maximum number of processors may not be a good idea every time as it may result in higher cost for setting up and in environments where multiple VMs are deployed it would lead to CPU resource crunch, overhead and ultimately negative performance.

4.b. In the system configuration of the VM, under the Acceleration Tab, explain the difference between the paravirtualization options: None, Legacy, Minimal, Hyper-V, and KVM. Explain which one would be best to use with Ubuntu Linux, and why.

- Acceleration tab is used for improving the performance. The difference is highlighted as below:

- *None – Paravirtualization is not used.*
- *Legacy – this is used for old VMs usually for backward compatibility.*
- *Minimal – minimal is used when there is no much modification required on the guest OS. The benefits lie between full and paravirtualization.*
- *Hyper-V – developed by Microsoft used for having multiple VMs generally for Windows guests.*
- *KVM – KVM is for Linux systems, makes the physical systems into Hypervisor and allowing the privilege of having multiple VMs.
KVM would suit well for Ubuntu Linux as KVM is well integrated with Linux Kernel and would perform better with Linux environments.*

4.c. In storage devices when configuring the VM, there are multiple types of storage controllers: explain the difference between the IDE, SATA, and NVMe controller. Give an example for each type of storage controller of a scenario where you may want to use this type of controller.

- *IDE – IDEs are legacy controllers introduced in 1980s, typically used in Hard Disk Drives, Floppy drives. These lack speed and are not performance oriented. An example for this type would be when there is a need to run old Operating Systems as it supports only legacy controller types.*
- *SATA – SATAs are better than IDE in performance aspect and they even offer support to more devices compared to IDEs. SATA even support SSD devices. Example of this application would be for modern day Operating Systems and VMs.*
- *NVMe – NVMes' are faster and latest compared to the latter (SATA and IDE), used for SSDs, they offer low latency along with the support for parallelism too. Typical use case for NVMe would be for Data Intensive, performance focused database like SQL Server.*

4.d. In the network configuration of the VM, there are multiple types of network adapters: explain the difference between NAT, Bridged Adapter, Internal Network, and Host-only Network. Give an example for each type of network of a scenario where you may want to use this type of network.

- *NAT – stands for Network Address Translation, NAT provides the flexibility to assign multiple devices a single IP address. This blocks all external networks from accessing the VM, thus protecting the VMs. An example of this would be to allow internet connection for a virtual machine.*
- *Bridged Adapter – this configuration is used when external networks need to access the VMs, each VMs will have their own unique IP address. Bridged Adapter configuration would be required when hosting a Web Server as it should be available to the public.*
- *Internal Network – This is required when we have multiple VMs and secure communication is required among them. This is used to create a private network in the Hypervisor. These types of networks can be useful for clustered databases.*
- *Host-only Network – Similar to Internal Networks, these types of networks can communicate with the host system and other VMs only and not to external networks since they are not connected to physical networking interface. Host-only is used when there is coordination is required between VMs for instance, like a web server VM and a database VM.*

4.e. For the USB configuration of the VM, explain the difference between USB 1.1, 2.0, and 3.0 controllers

- *USB 1.1 – this provides universal support and a legacy technology, offering transfer speed of up to 12 Mbps.*
- *USB 2.0 – much faster than USB 1.1, this is a balance of speed and compatibility. This has transfer speeds of up to 480Mbps.*
- *USB 3.0 – this supports speed of up to 5Gbps, used for high-speed and large data transfer applications.*

5.a. Describe what a core and hardware thread is on a modern processor, and the difference between them?

- Core is a processing unit which is part of a Central Processing Unit (CPU), a CPU can have multiple cores to support multitasking. A CPU can have up to 12 cores. Large application machines can have chores of around 48.

- Hardware thread is the ability of each core of a CPU to operate multiple tasks simultaneously. Each core can have multiple threads thus aiding in CPU utilization and performance(throughput).

5.b. How many cores do the fastest processors from each manufacturer have? Give an example (specific model, specs, and price). (a) Intel CPU (x86) (b) AMD CPU (x86) (c) IBM CPU (Power9) (d) ThunderX CPU (ARM) (e) NVIDIA GPU

	Intel CPU (x86)	AMD CPU (x86)	IBM CPU (POWER9)	THUNDERX CPU(ARM)	NVIDIA GPU
Model	Core i9- 13900KS	Ryzen 9 3990X	IBM Power9	Cavium ThunderX	RTX 4090
Cores	24	64	24	48	CUDA Cores-

					16384
Specs	6GHz, 32 threads, 36Mb Cache	4.3 GHz, 128 threads, Cache 288Mb	512Kb L2 Cache, 10Mb L3 Cache	SoC, 2.5 GHz, 78K I- Cache, 32K D-Cache, 16MB of L2 Cache	2.52 GHz, memory – 24GB GDDR6X
Price - \$	699	3,995	4,000	1,795	1,599

5.c. Why do we not have processors running at 1THz today (as might have been predicted in the year 2000)?

- *There are multiple reasons for not having processors running at 1000GHz currently due to many reasons, few are listed below:*

- *Higher frequencies mean more power (more power consumption) which equates to higher heat dissipation. Even top-notch cooling systems will face challenge in cooling such high temperatures.*
- *Much software's at the moment are not configured to fully utilize the processors of such high clock speed.*
- *Present day storage systems cannot cope up with the throughput of such fast CPUs.*
- *The time, resources and manufacturing costs involved for this development would not be viable.*

5.d. Describe Moore's Law. Is it going to go on forever? If not, when will it end? Justify your answer to why it will end and when.

- *Moore's Law is an observation made in 1965 by Gordon Moore, he stated that the number of transistors on an Integrated Circuit will double every two years. However, this might not hold true in the long run due to the current limitations and challenges:*

- *When transistors become smaller and smaller, they are prone to quantum effects, which will disrupt the behaviour of the transistor.*
- *As the transistor size decreases, the manufacturing costs will scale up and even the difficulty in fabricating them.*
- *Limitations of silicon like controlling leakage currents and heat management will be major challenge.*

The validity of Moore's law may face challenges in the near future, potentially becoming obsolete by around 2025.

6.a. Why is threading useful on a single-core processor?

- *Threading aids in efficient use of CPU Cores. Multithreading is possible for a single-core processor also, threading really helps in switching between tasks more faster, giving the illusion of parallel execution and maximizing utilization. Thus, assisting in responsiveness of the system and reducing latency effects.*

6.b. Do more threads always mean better performance?

- *This might not hold true every time. The most basic way is to match the count of threads to the count of CPU. But, having less threads would not fully utilize the CPU cores. Threading helps in throughput but not make the CPU faster, it helps in executing multiple tasks simultaneously and make efficient use of the CPU. On the other hand, having more threads would also add up to queuing hence leading to longer wait time for program execution. A CPU with 4 cores and 4 threads would perform better than 2 cores and 4 threads.*

6.c. Is super-linear speedup possible? Explain why or why not.

- *Super-linear speedup would not be possible theoretically, but in practice we do see such cases rarely. For instance, when working in a set of sequential program doesn't fit in the cache, resulting in numerous cache misses, parallel execution with two threads may avoid cache misses when the working set fits in two caches. This can lead to super-linear speedup, where the serial execution time exceeds twice the parallel execution time. Amdahl's Law limits the maximum speedup. There are three key causes for super-linear speedup in parallel computing:*

- *Caching effects – Adding more processors increases total cache size, improving memory locality and reducing misses.*
- *Early backtracking – One thread finding an exception can trigger backtracking in others before more work is done.*
- *Optimization pruning, earlier termination.*

6.d. Why are locks needed in a multi-threaded program?

- *Locks provide access to shared resources, without them there would be data corruption as multiple threads would be trying to access the data (race condition) hence enforcing synchronization, they are also used to prevent Deadlocks from happening. Locks also help in distributing files for one thread at a time thus avoiding conflicts. In addition to this, locks also enforce that the threads run in order.*

6.e. Would it make sense to limit the number of threads in a server process?

- *It would be sensible to limit the number of threads in a server process. It mainly depends on the workload, resource availability. Having too many threads can limit the number of CPU cores resulting in conflict for the threads to share resources, resulting in performance degradation. Having lot of active threads means having large number of context switching which can cause overhead and diminish performance. For efficient use of resources, it is better to limit the threads based on operating conditions and use case.*