

## Assignment -1

Name: - Pranav Murali

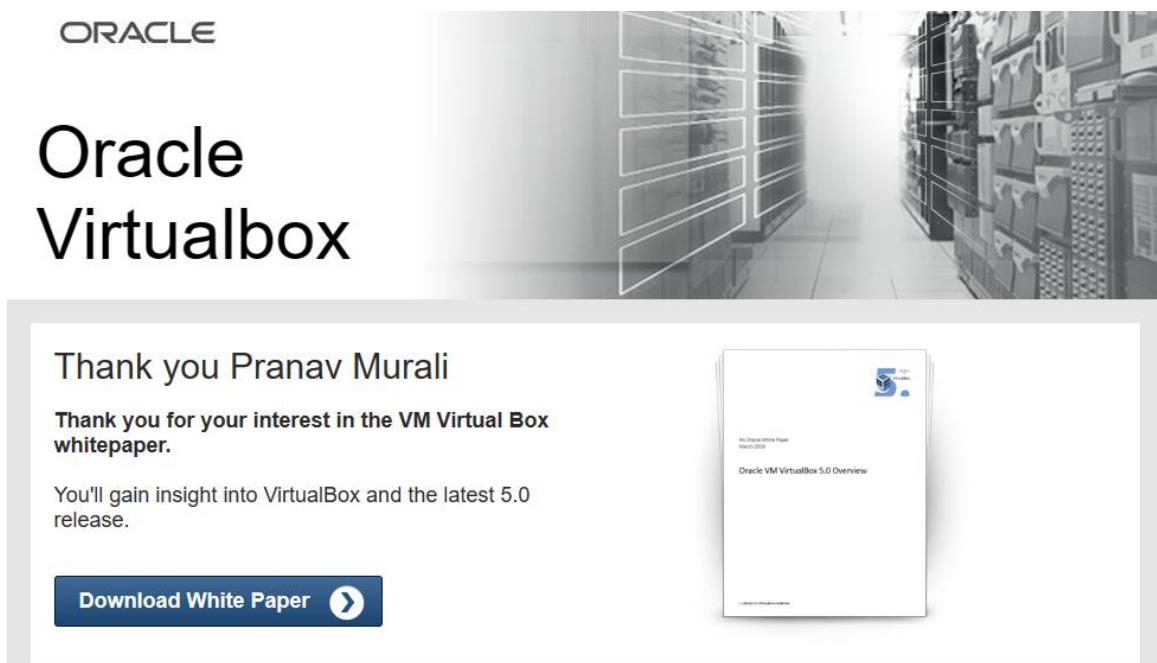
CWID: - A20555824

1.

### **a) Read Oracle VirtualBox White Paper**

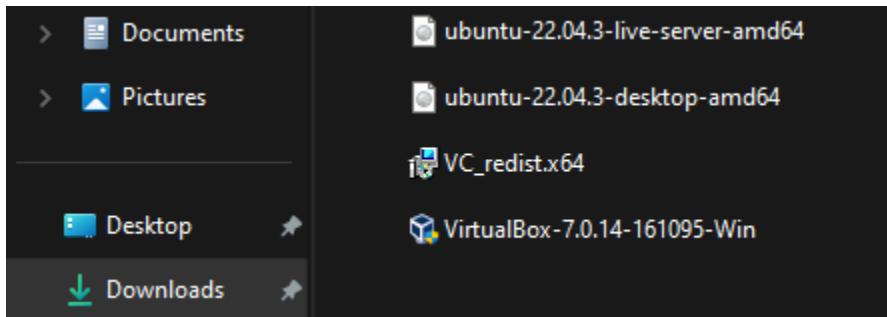
I read the paper, created an email account, and received the mail below. Information on the Oracle VM virtual box is provided in the document below.

- Its two parts, the virtual box expansion pack, and the VM virtual box: While the virtual box is free, the extended version has a few extra capabilities and requires a paid subscription.
- Virtual box uses include software upgrades, testing new modifications, troubleshooting and developing multiple applications that require diverse environments, performing demos, and running demo apps in parallel.
- Information shared between virtual machines (VMs) or from your system is safe since encryption is used.
- We can also utilize virtual boxes for training.



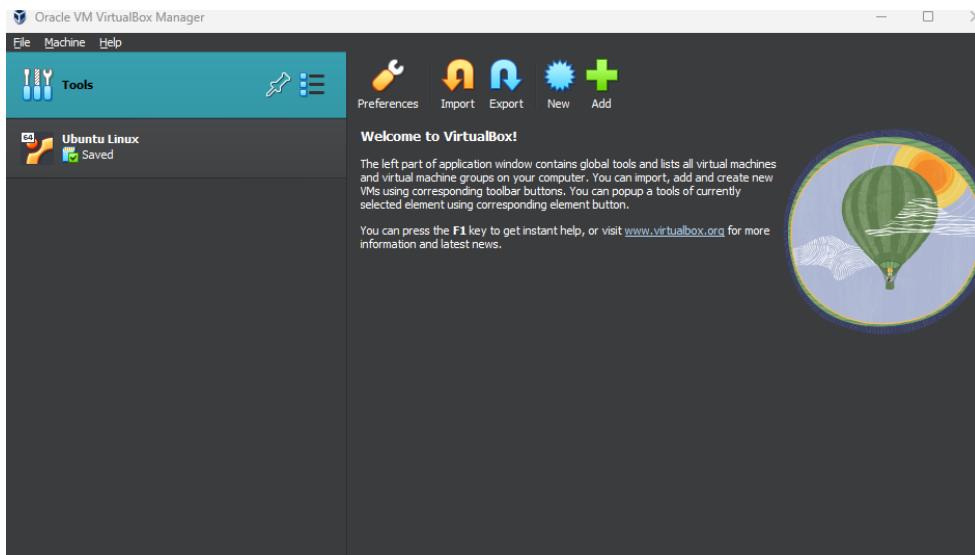
The image is a screenshot of an email from Oracle. The subject line is "Thank you Pranav Murali". The body of the email starts with "Thank you for your interest in the VM Virtual Box whitepaper." It then says "You'll gain insight into VirtualBox and the latest 5.0 release." Below this, there is a blue button with the text "Download White Paper" and a white arrow icon. To the right of the text, there is a thumbnail image of the white paper cover, which features the Oracle logo and the title "Oracle VM VirtualBox 5.0 Overview".

**b) Download Oracle VirtualBox 6.0.4**



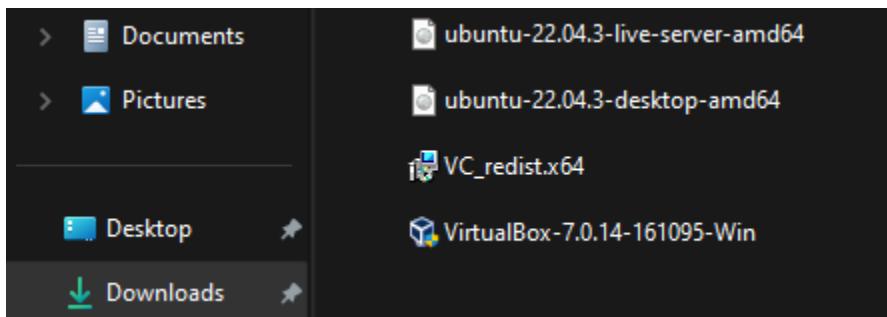
**c) Install VirtualBox**

Installed Virtual box on my machine below is the screenshot for the same

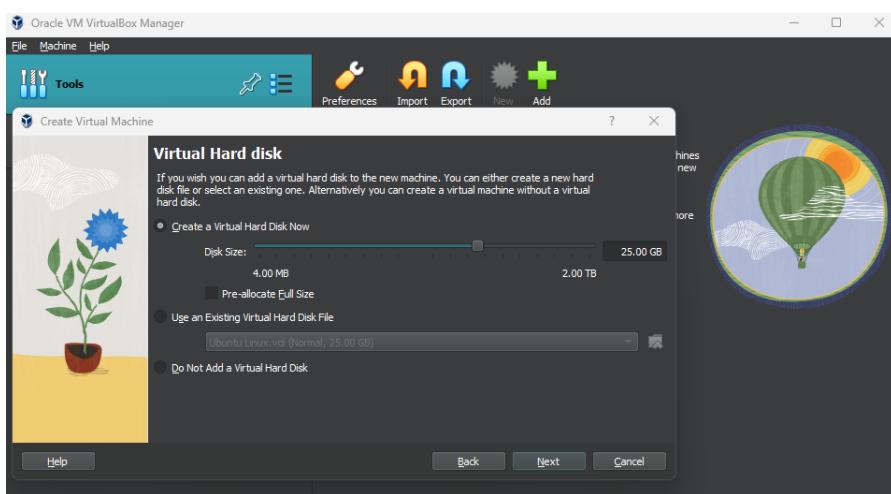
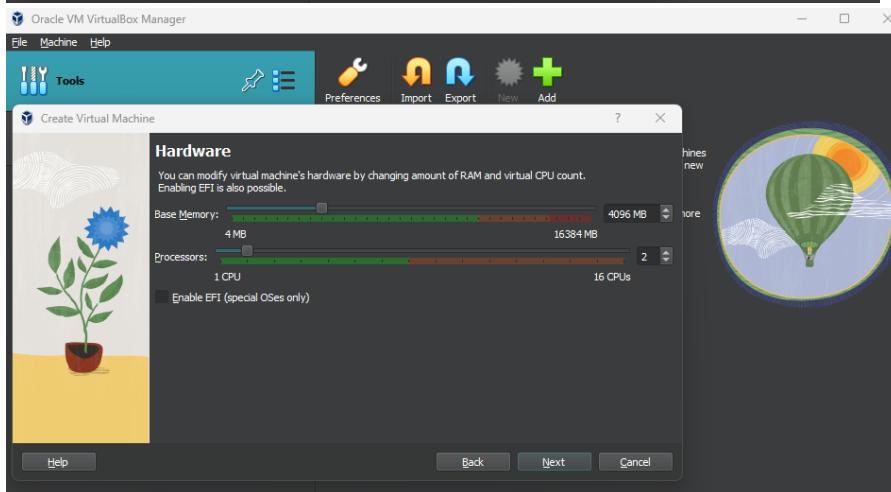
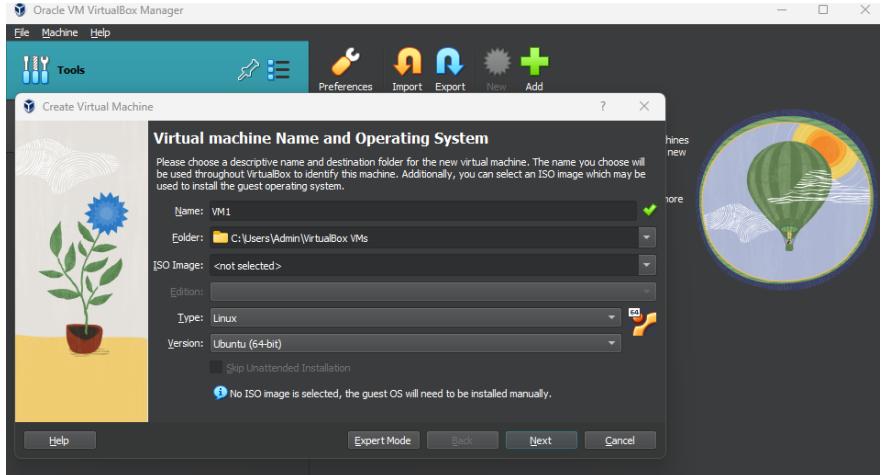


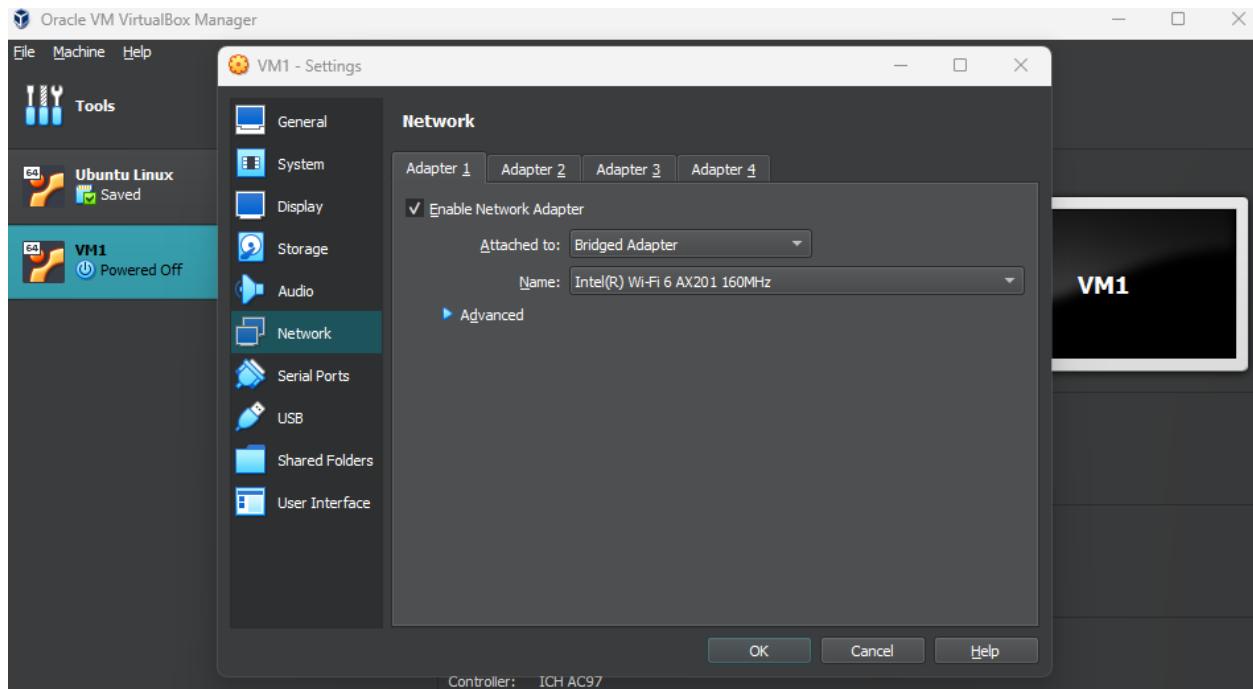
**d) Download Ubuntu 18.04.1 Linux ISO image**

Downloaded Ubuntu 18.04.1 below is the screenshot for the same

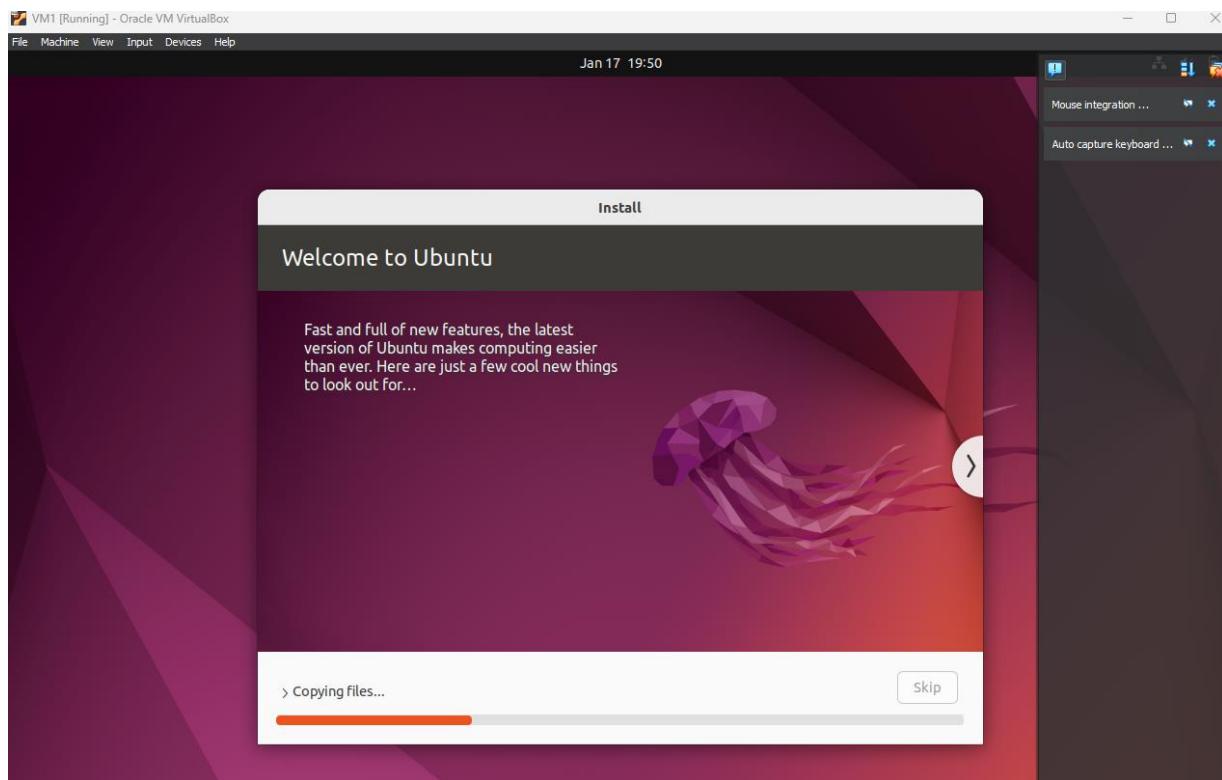


**e) Create Virtual Machine (VM), to support Linux, Ubuntu, 64-bit, 1GB RAM, Virtual Disk 20GB, VDI image, dynamically allocated, 1-core, and a network interface (1GbE or WiFi) with Bridged Adapter**

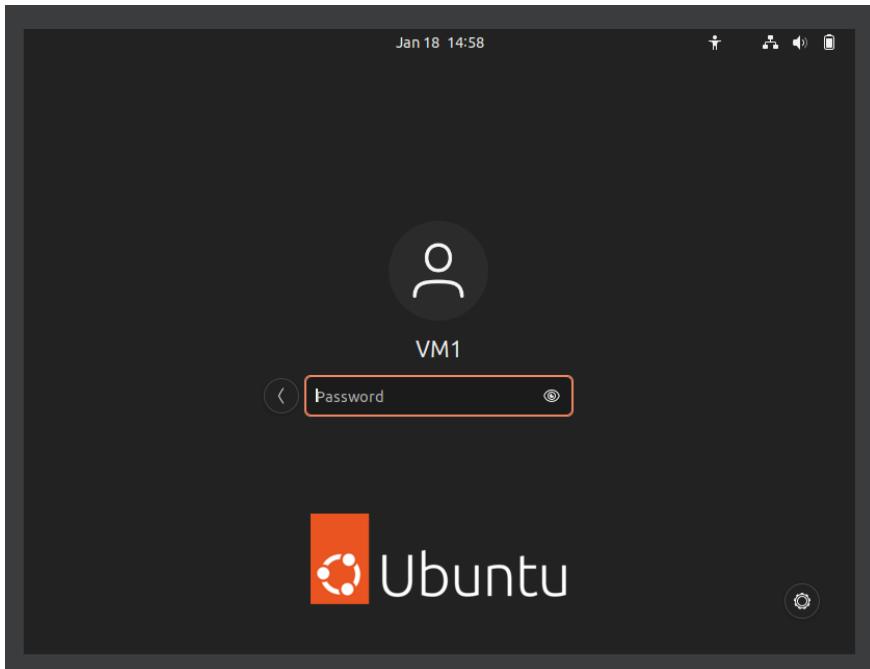




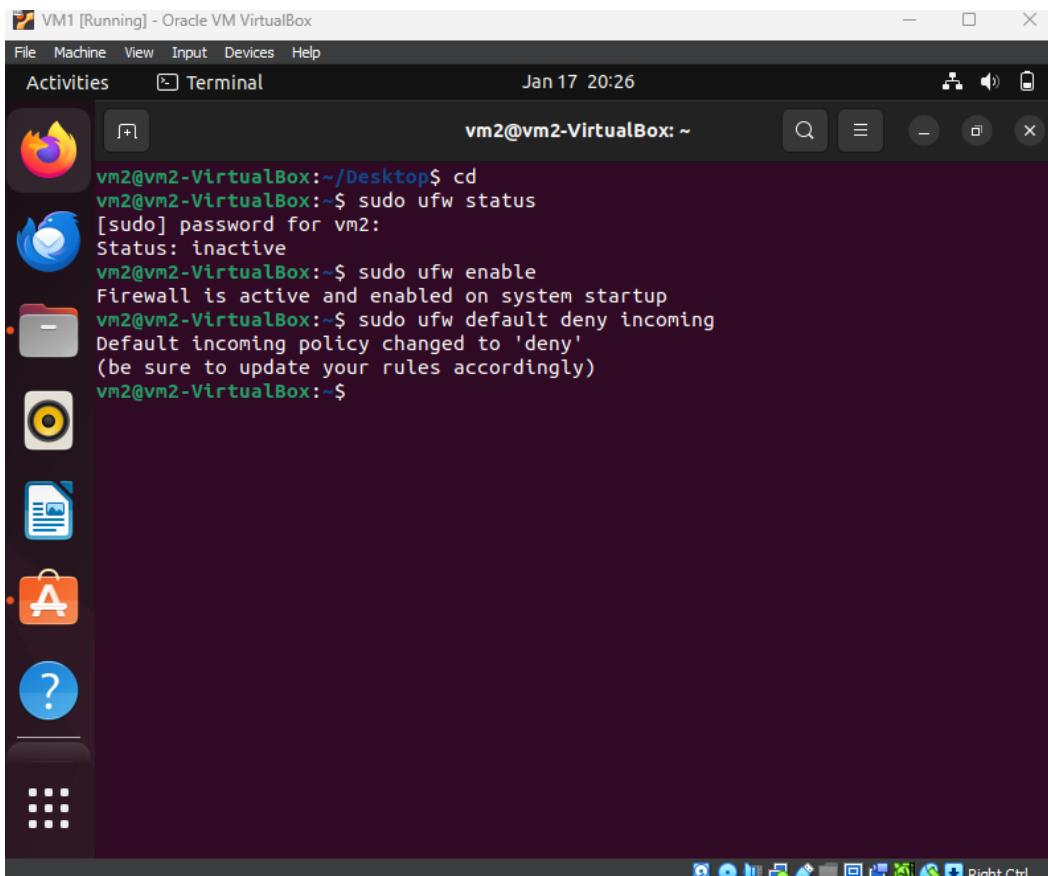
## f) Install Linux from the ISO image



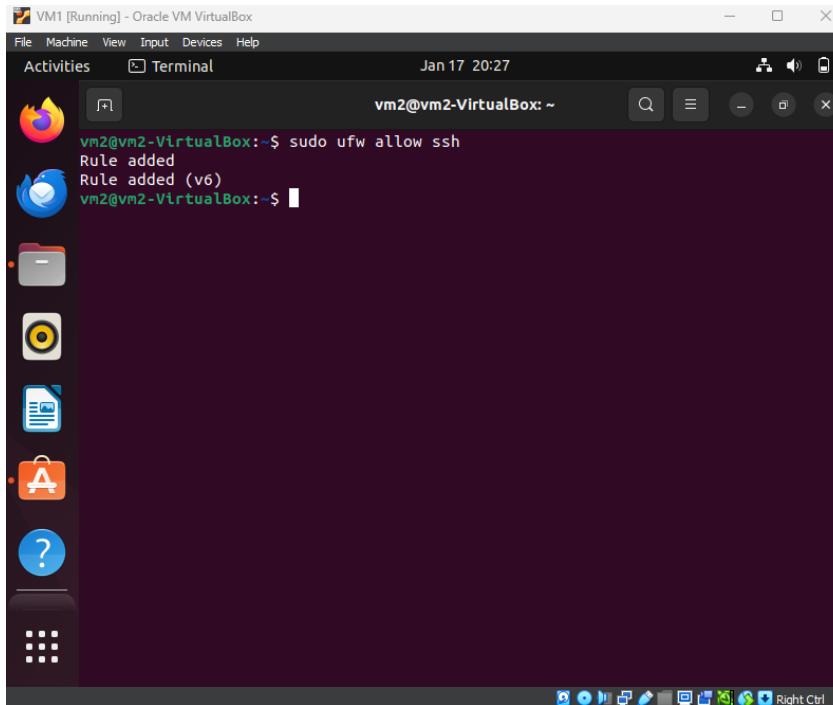
**g) Create a user id and password**



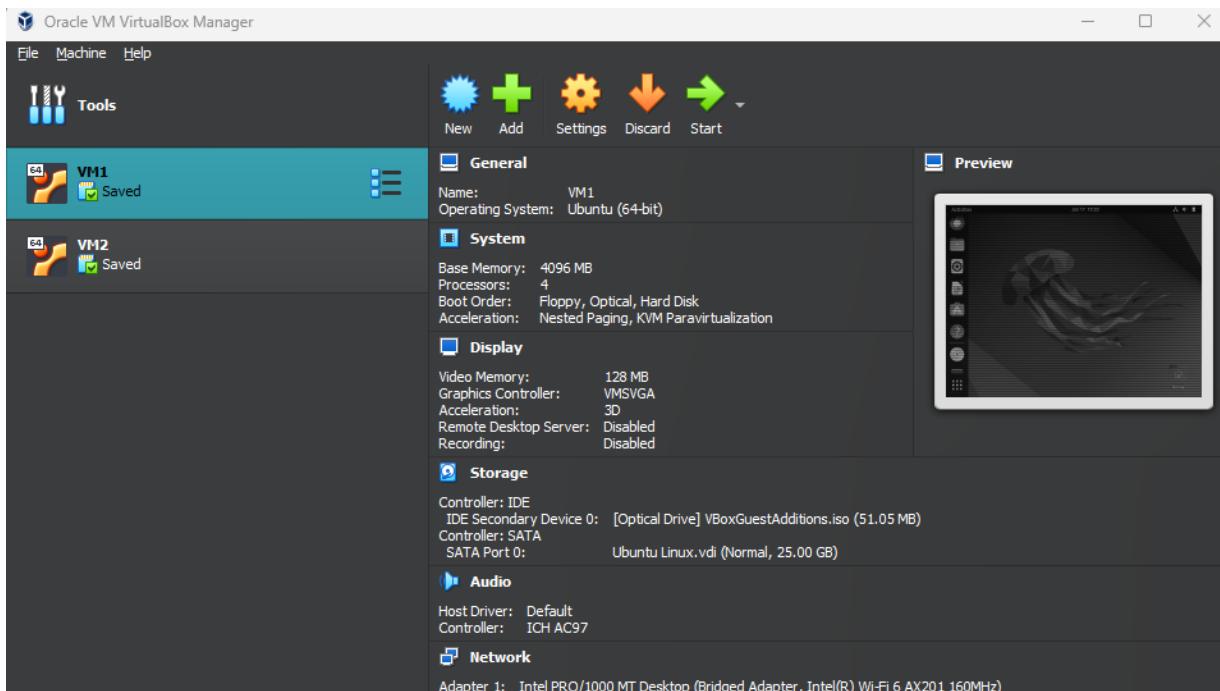
**h) Turn on Firewall and block all ports**



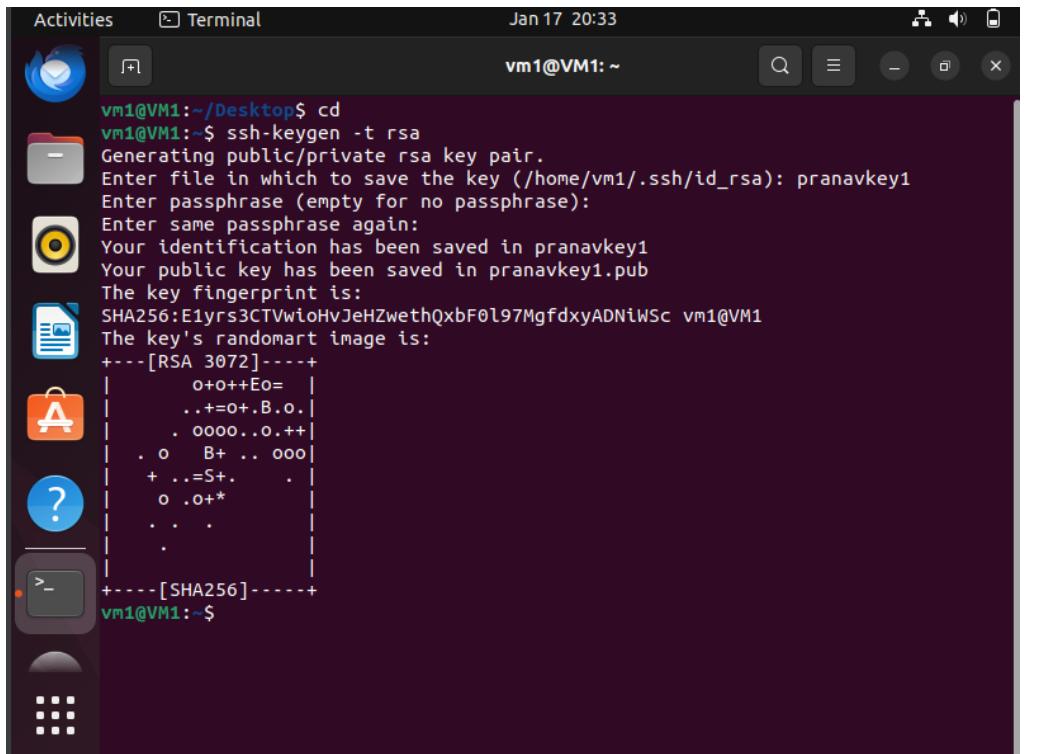
**i) Enable SSH access to your new Linux installation; open SSH port in firewall**



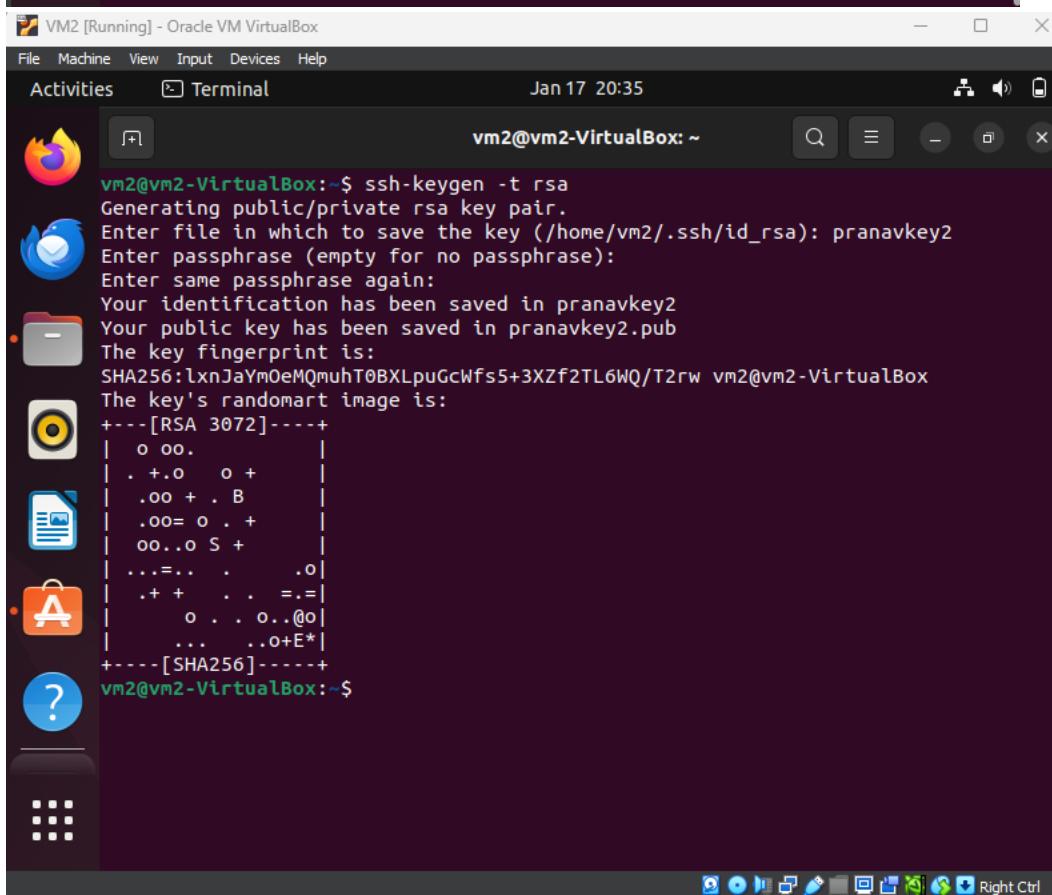
**j) Repeat steps 5 through 9, and create another VM with the same specifications as the first one** **Created another VM with the same details**



**k) Create private/public keys and install them properly in both of your new VMs**

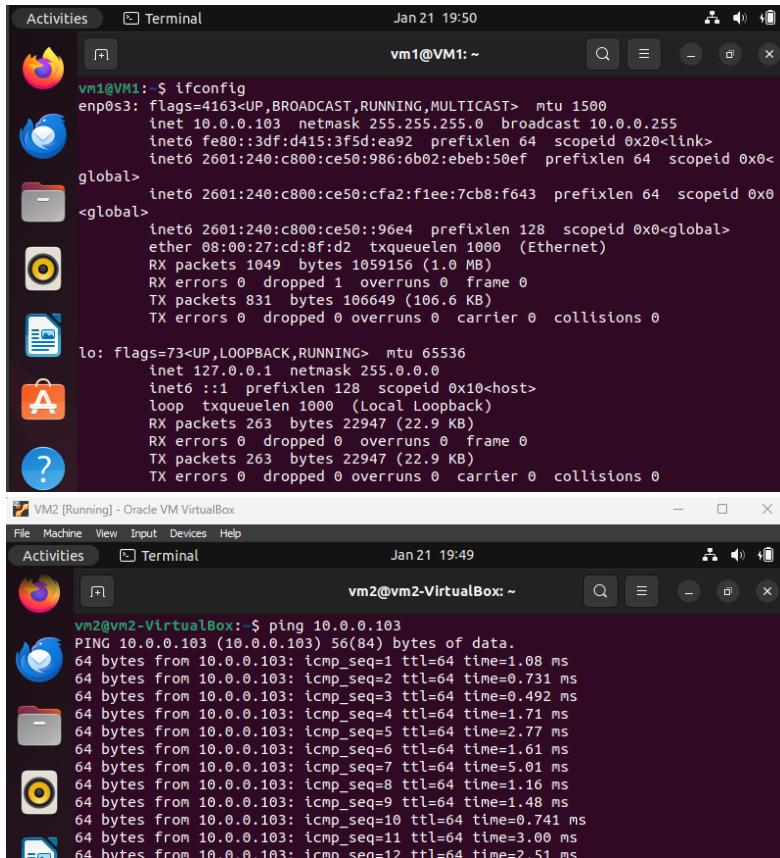


```
Activities Terminal Jan 17 20:33
vm1@VM1:~$ cd
vm1@VM1:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/vm1/.ssh/id_rsa): pranavkey1
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in pranavkey1
Your public key has been saved in pranavkey1.pub
The key fingerprint is:
SHA256:E1yrs3CTVwloHvJeHZwethQxbF0l97MgfdxyADNiWSc vm1@VM1
The key's randomart image is:
+---[RSA 3072]---+
|   o+o++Eo= |
|   ..+o+.B.o.|
|   . oooo...o++|
|   . o   B+ .. ooo|
|   + ..=S+.   .|
|   o .o+*      |
|   . . .      |
|   . . .      |
+---[SHA256]---+
vm1@VM1:~$
```



```
VM2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Jan 17 20:35
vm2@vm2-VirtualBox:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/vm2/.ssh/id_rsa): pranavkey2
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in pranavkey2
Your public key has been saved in pranavkey2.pub
The key fingerprint is:
SHA256:lxnJaYm0eMQuhT0BXLpuGcWfs5+3XZf2TL6WQ/T2rw vm2@vm2-VirtualBox
The key's randomart image is:
+---[RSA 3072]---+
|   o oo.      |
|   .+.o  o +   |
|   .oo + . B   |
|   .oo= o . +   |
|   oo..o S +   |
|   ...=... . .o |
|   .+ + . . =.=|
|   o . . o..@o |
|   ... . .o+E* |
+---[SHA256]---+
vm2@vm2-VirtualBox:~$
```

**I) Test that you can connect remotely to your VMs with your keys, from one VM to the other VM**



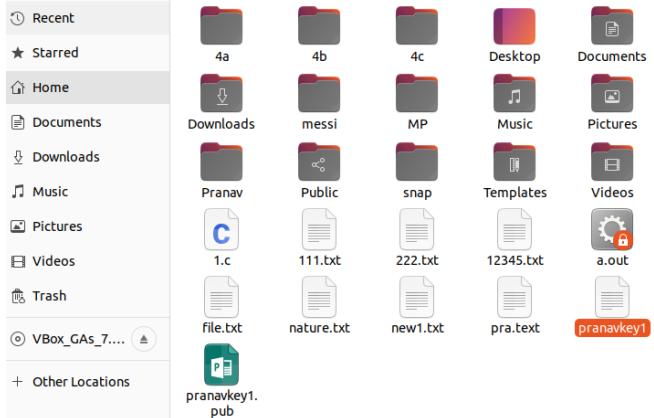
```
vm1@VM1: ~
```

```
ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.103 netmask 255.255.255.0 broadcast 10.0.0.255
        inet6 fe80::3df:d415:3f5d:ea92/128 scopeid 0x20<link>
            inet6 2601:240:c800:ce50:986:6b02:eb:50ef/64 scopeid 0x0<global>
                inet6 2601:240:c800:ce50::96e4/128 scopeid 0x0<global>
                    ether 08:00:27:cd:8f:d2 txqueuelen 1000 (Ethernet)
                    RX packets 1049 bytes 1059156 (1.0 MB)
                    RX errors 0 dropped 1 overruns 0 frame 0
                    TX packets 831 bytes 106649 (106.6 KB)
                    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
            RX packets 263 bytes 22947 (22.9 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 263 bytes 22947 (22.9 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

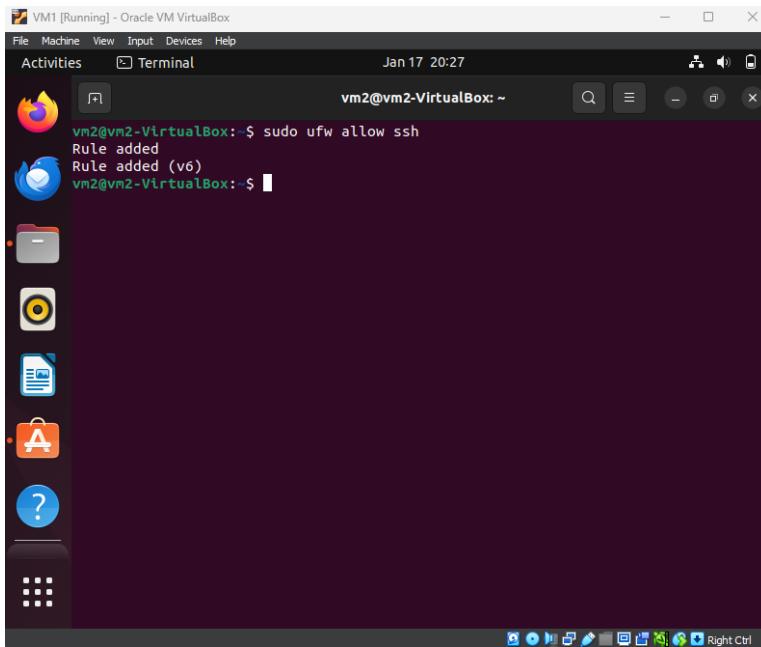
```
vm2@vm2-VirtualBox: ~
```

```
ping 10.0.0.103
PING 10.0.0.103 (10.0.0.103) 56(84) bytes of data.
64 bytes from 10.0.0.103: icmp_seq=1 ttl=64 time=1.08 ms
64 bytes from 10.0.0.103: icmp_seq=2 ttl=64 time=0.731 ms
64 bytes from 10.0.0.103: icmp_seq=3 ttl=64 time=0.492 ms
64 bytes from 10.0.0.103: icmp_seq=4 ttl=64 time=1.71 ms
64 bytes from 10.0.0.103: icmp_seq=5 ttl=64 time=2.77 ms
64 bytes from 10.0.0.103: icmp_seq=6 ttl=64 time=1.61 ms
64 bytes from 10.0.0.103: icmp_seq=7 ttl=64 time=5.01 ms
64 bytes from 10.0.0.103: icmp_seq=8 ttl=64 time=1.16 ms
64 bytes from 10.0.0.103: icmp_seq=9 ttl=64 time=1.48 ms
64 bytes from 10.0.0.103: icmp_seq=10 ttl=64 time=0.741 ms
64 bytes from 10.0.0.103: icmp_seq=11 ttl=64 time=3.00 ms
64 bytes from 10.0.0.103: icmp_seq=12 ttl=64 time=2.51 ms
```

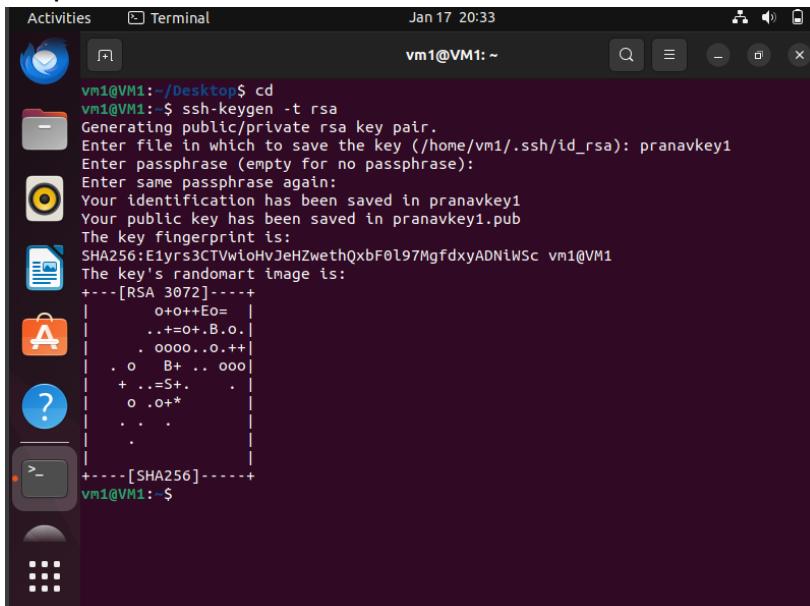
**2. Show an example of using the following commands (hint: you can use man to find more information about each one); take screenshots of your commands; make sure to clear the screen between each command; explain in your own words what these commands do.**

- a) **Ssh:** Given command is used to establish a secure network between two machines.



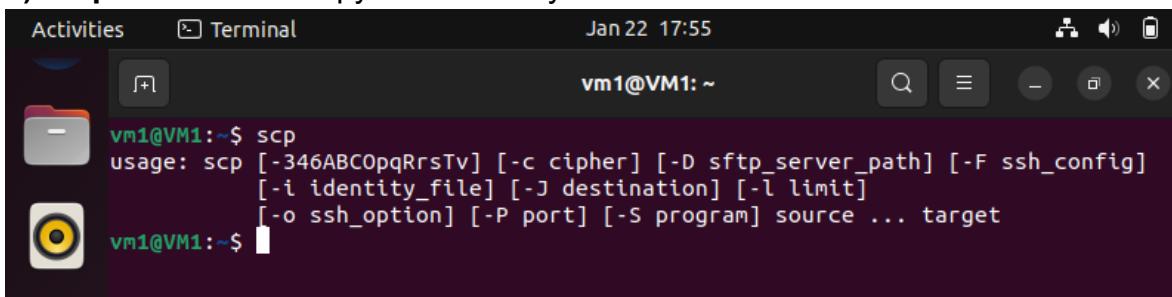
```
VM1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Jan 17 20:27
vm2@vm2-VirtualBox: ~
vm2@vm2-VirtualBox: $ sudo ufw allow ssh
Rule added
Rule added (v6)
vm2@vm2-VirtualBox: ~
```

b) **Ssh-keygen**: It is used for key generation in the public key authentication protocol.



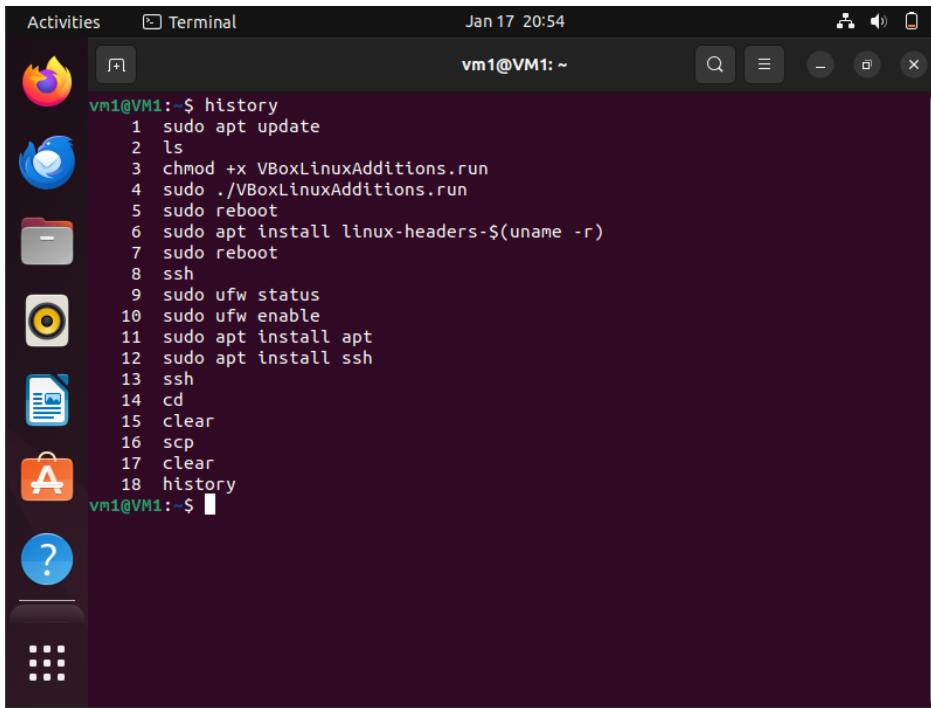
```
Activities Terminal Jan 17 20:33
vm1@VM1: ~
vm1@VM1: ~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/vm1/.ssh/id_rsa): pranavkey1
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in pranavkey1
Your public key has been saved in pranavkey1.pub
The key fingerprint is:
SHA256:E1yrs3CTVwloHvJeHZwethQxbF0l97MgfaxyADNlWSc vm1@VM1
The key's randomart image is:
+---[RSA 3072]---+
|   o+o+Eo=  |
|   ..+=o+B.o.|
|   . oooo...o.++|
|   . o  B+ ... oo|
|   + ...=S+.   .|
|   o ..o+*   .   |
|   . . .   .   |
|   . . .   .   |
+---[SHA256]---+
vm1@VM1: ~$
```

c) **Scp**: It is used to copy files securely from one host to another



```
Activities Terminal Jan 22 17:55
vm1@VM1: ~
vm1@VM1: ~$ scp
usage: scp [-346ABC0pqRrsTv] [-c cipher] [-D sftp_server_path] [-F ssh_config]
           [-i identity_file] [-J destination] [-l limit]
           [-o ssh_option] [-P port] [-S program] source ... target
vm1@VM1: ~$
```

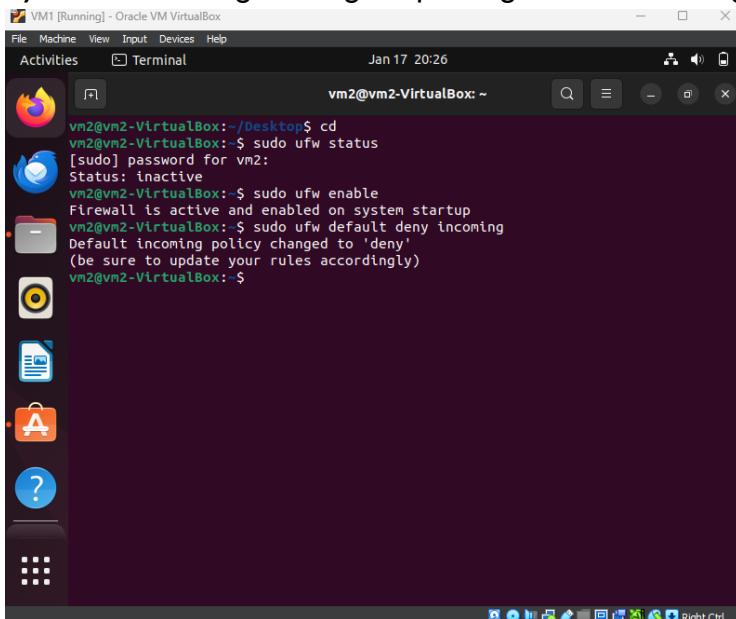
d) **History**: It provides list of commands which we used in past



A screenshot of an Ubuntu desktop environment. On the left is a dock with icons for the Dash, Home, Applications, and Help. The main window is a terminal window titled 'Terminal' with the title bar showing 'vm1@VM1: ~' and the date 'Jan 17 20:54'. The terminal window displays a command history:

```
vm1@VM1:~$ history
 1 sudo apt update
 2 ls
 3 chmod +x VBoxLinuxAdditions.run
 4 sudo ./VBoxLinuxAdditions.run
 5 sudo reboot
 6 sudo apt install linux-headers-$(uname -r)
 7 sudo reboot
 8 ssh
 9 sudo ufw status
10 sudo ufw enable
11 sudo apt install apt
12 sudo apt install ssh
13 ssh
14 cd
15 clear
16 scp
17 clear
18 history
vm1@VM1:~$
```

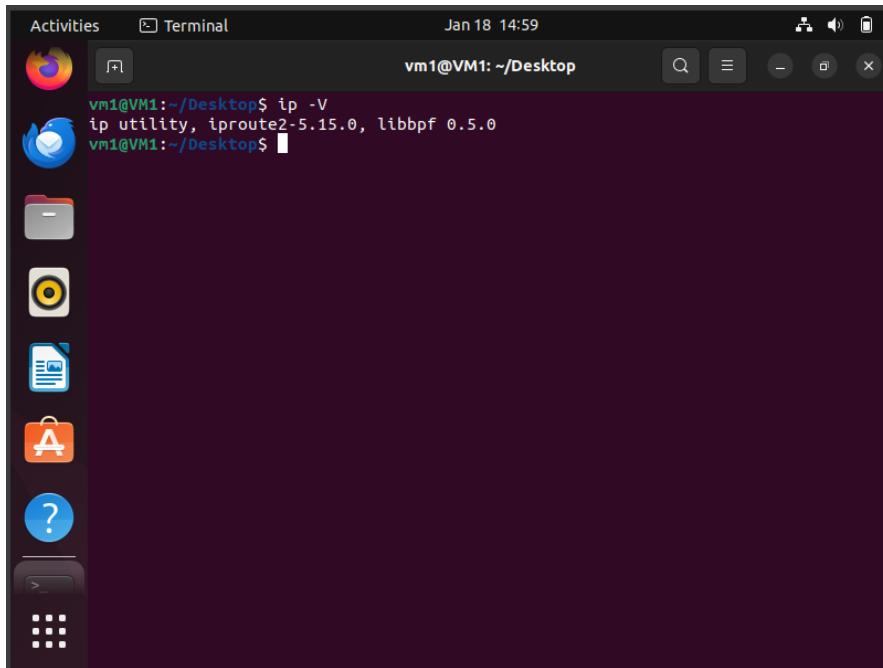
e) **Sudo**: Sudo gives higher privileges while running the commands.



A screenshot of an Ubuntu desktop environment. On the left is a dock with icons for the Dash, Home, Applications, and Help. The main window is a terminal window titled 'Terminal' with the title bar showing 'vm2@vm2-VirtualBox: ~' and the date 'Jan 17 20:26'. The terminal window displays the following commands:

```
VM1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Jan 17 20:26
vm2@vm2-VirtualBox:~/Desktop$ cd
vm2@vm2-VirtualBox:~$ sudo ufw status
[sudo] password for vm2:
Status: inactive
vm2@vm2-VirtualBox:~$ sudo ufw enable
Firewall is active and enabled on system startup
vm2@vm2-VirtualBox:~$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
vm2@vm2-VirtualBox:~$
```

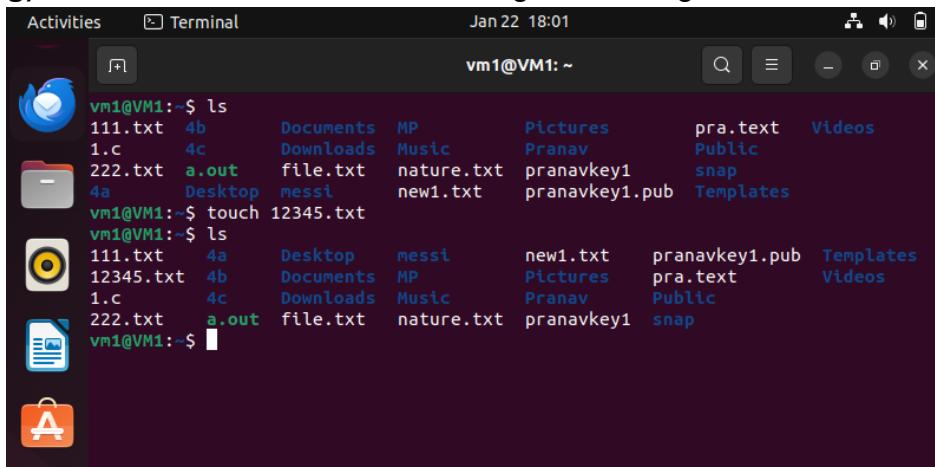
f) **Ip**: It is used to get information about Ip utility, Ip -v can be used to get the version of Ip utility.



A screenshot of an Ubuntu desktop environment. On the left is a vertical dock with icons for Dash, Home, Dash to Dock, and Help. The main window is a terminal window titled 'Terminal' with the status bar showing 'Jan 18 14:59' and the command line 'vm1@VM1: ~/Desktop'. The terminal output shows:

```
vm1@VM1:~/Desktop$ ip -V
ip utility, iproute2-5.15.0, libbpf 0.5.0
vm1@VM1:~/Desktop$
```

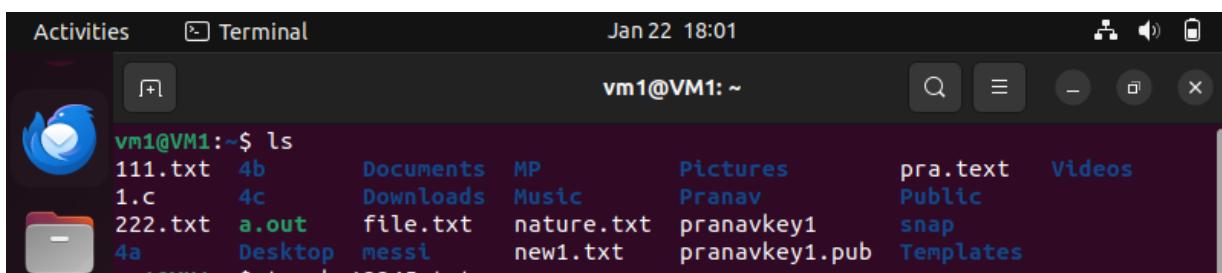
g) **Touch**: Touch is used for creating files through the terminal.



A screenshot of an Ubuntu desktop environment. The terminal window shows the following session:

```
vm1@VM1:~$ ls
111.txt 4b  Documents  MP  Pictures  pra.text  Videos
1.c  4c  Downloads  Music  Pranav  Public
222.txt  a.out  file.txt  nature.txt  pranavkey1  snap
4a  Desktop  messi  new1.txt  pranavkey1.pub  Templates
vm1@VM1:~$ touch 12345.txt
vm1@VM1:~$ ls
111.txt  4a  Desktop  messi  new1.txt  pranavkey1.pub  Templates
12345.txt  4b  Documents  MP  Pictures  pra.text  Videos
1.c  4c  Downloads  Music  Pranav  Public
222.txt  a.out  file.txt  nature.txt  pranavkey1  snap
vm1@VM1:~$
```

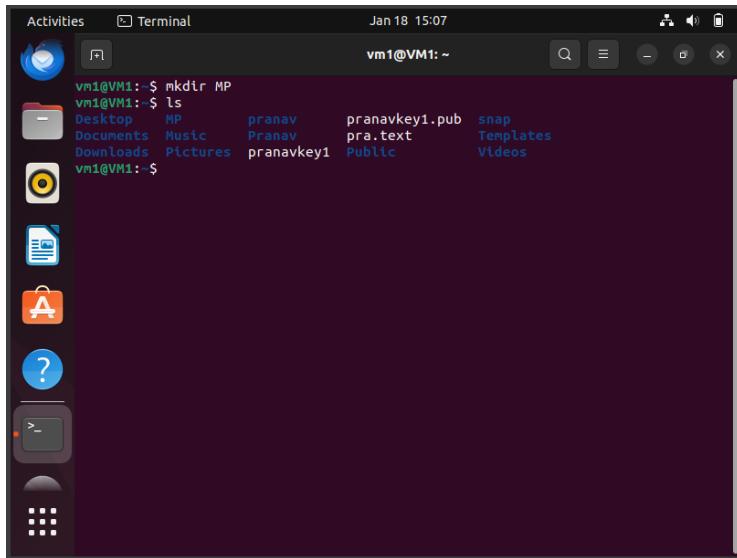
h) **Ls**: It is used to print all content present in given directory.



A screenshot of an Ubuntu desktop environment. The terminal window shows the following session:

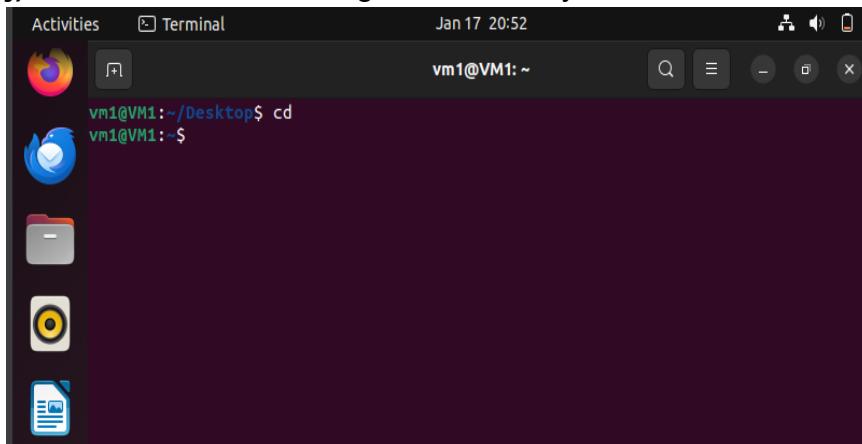
```
vm1@VM1:~$ ls
111.txt  4b  Documents  MP  Pictures  pra.text  Videos
1.c  4c  Downloads  Music  Pranav  Public
222.txt  a.out  file.txt  nature.txt  pranavkey1  snap
4a  Desktop  messi  new1.txt  pranavkey1.pub  Templates
```

i) **Mkdir**: It is used to create a directory.



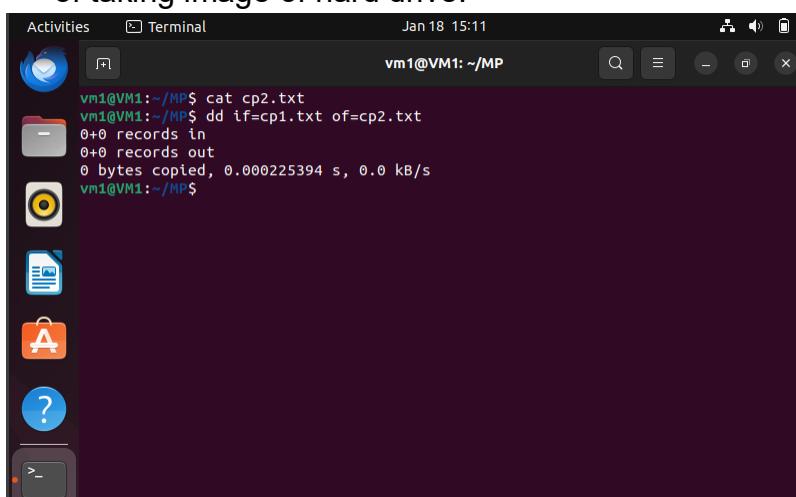
```
Activities Terminal Jan 18 15:07
vm1@VM1:~$ ls
Desktop MP pranav pranavkey1.pub snap
Documents Music Pranav pra.text Templates
Downloads Pictures pranavkey1 Public Videos
vm1@VM1:~$
```

j) **Cd:** It is used to change the directory.



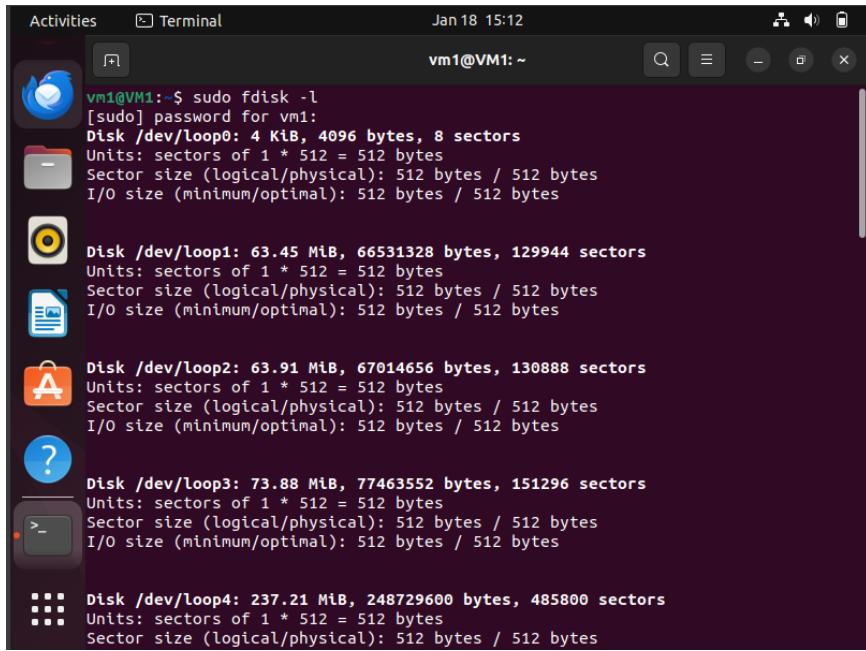
```
Activities Terminal Jan 17 20:52
vm1@VM1:~/Desktop$ cd
vm1@VM1:~$
```

k) **Dd:** It is used to copy or convert the files, Even it is used for backup, restore of taking image of hard drive.



```
Activities Terminal Jan 18 15:11
vm1@VM1:~/MP$ cat cp2.txt
vm1@VM1:~/MP$ dd if=cp1.txt of=cp2.txt
0+0 records in
0+0 records out
0 bytes copied, 0.000225394 s, 0.0 kB/s
vm1@VM1:~/MP$
```

l) **Fdisk:** fdisk is used for disk partitioning or getting details about disks in your system.



```
Activities Terminal Jan 18 15:12
vm1@VM1:~$ sudo fdisk -l
[sudo] password for vm1:
Disk /dev/loop0: 4 KiB, 4096 bytes, 8 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

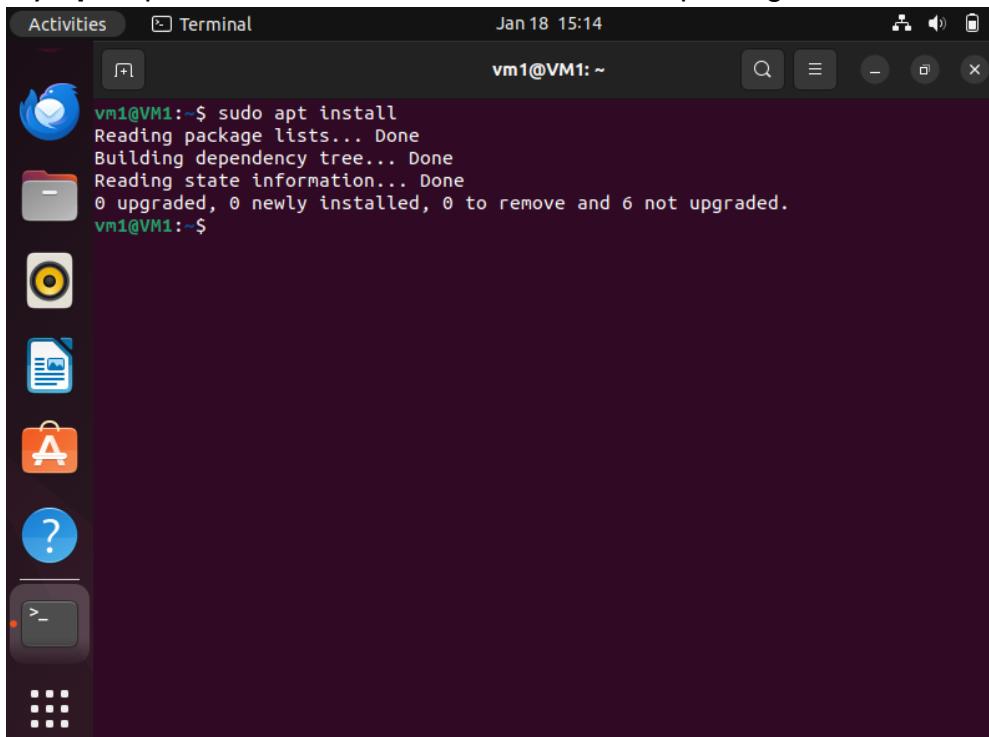
Disk /dev/loop1: 63.45 MiB, 66531328 bytes, 129944 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop2: 63.91 MiB, 67014656 bytes, 130888 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop3: 73.88 MiB, 77463552 bytes, 151296 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop4: 237.21 MiB, 248729600 bytes, 485800 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
```

m) **Apt**: Apt is a tool which is used to install the packages.



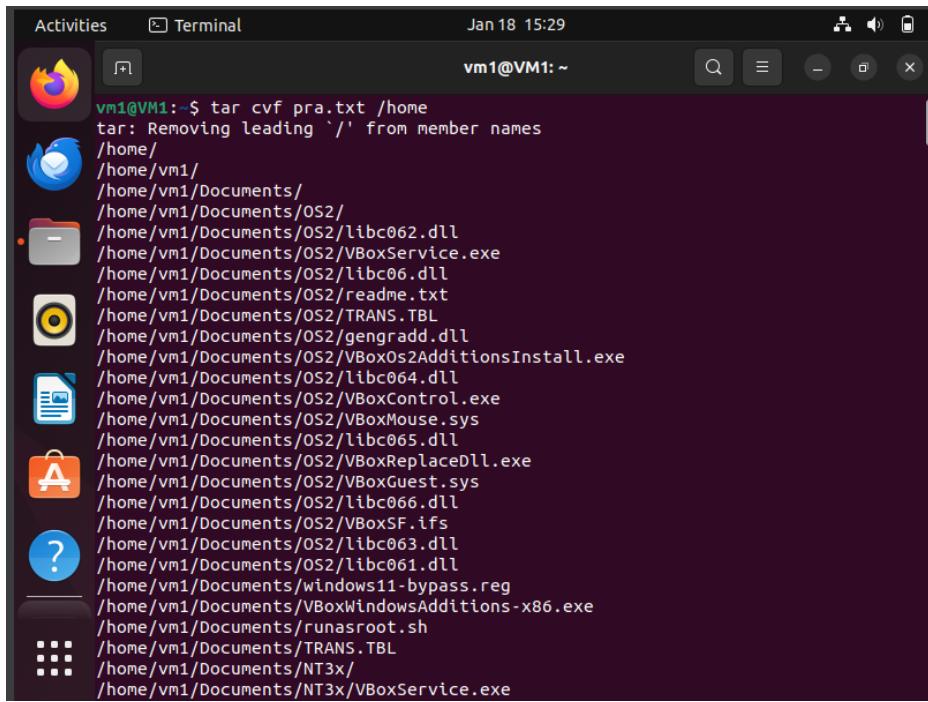
```
Activities Terminal Jan 18 15:14
vm1@VM1:~$ sudo apt install
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 6 not upgraded.
vm1@VM1:~$
```

n) **Vi**: It is used to insert updated text or files in the console. If the file is not present, then it will create that file and the vi user can write into the file.

- o) **Time:** Given command is used to measure the time difference between given requests

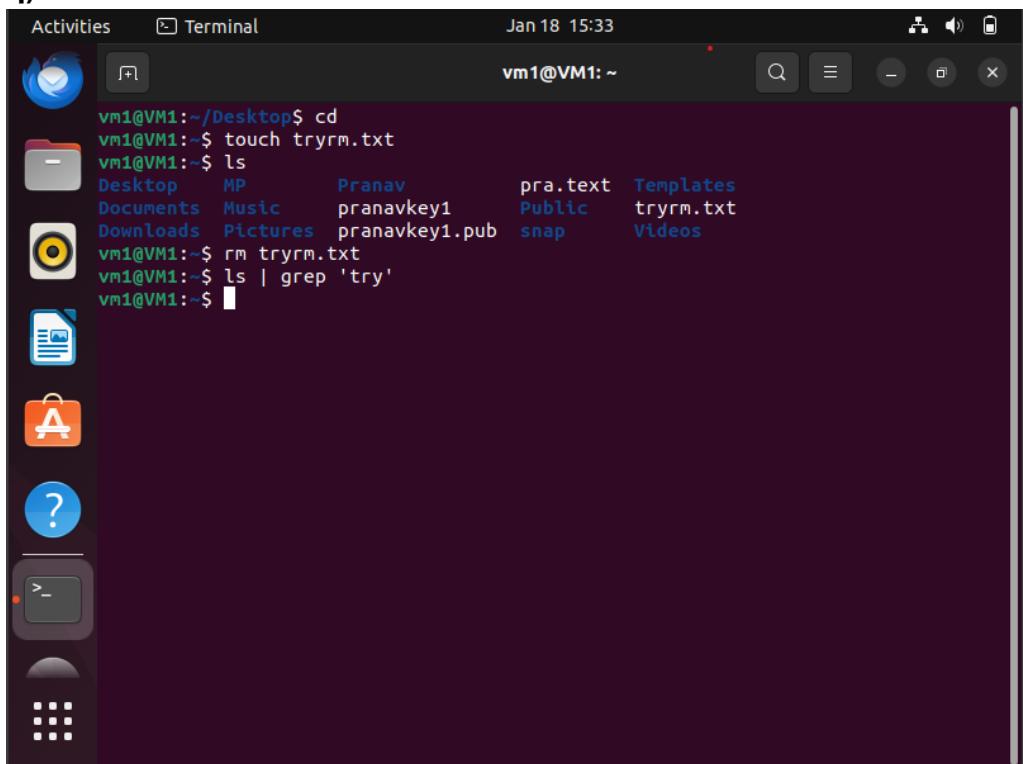
```
Activities Terminal Jan 18 15:21
vm1@VM1: ~
vm1@VM1:~$ time
real    0m0.000s
user    0m0.000s
sys     0m0.000s
vm1@VM1:~$ time www.google.com
www.google.com: command not found
real    0m0.253s
user    0m0.063s
sys     0m0.115s
vm1@VM1:~$
```

p) Tar: It is used to archive the file



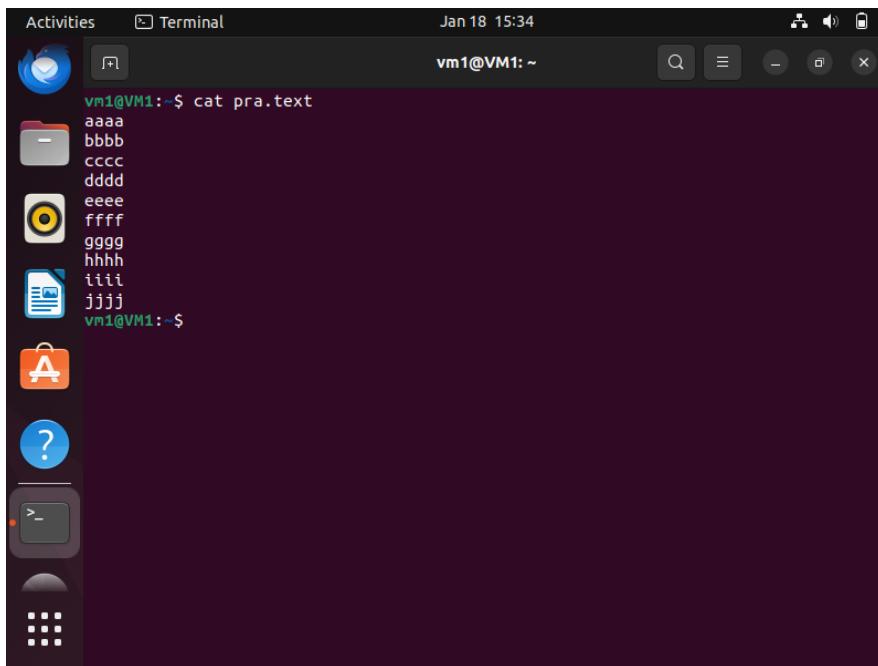
```
Activities Terminal Jan 18 15:29
vm1@VM1:~$ tar cvf pra.txt /home
tar: Removing leading '/' from member names
/home/
/home/vm1/
/home/vm1/Documents/
/home/vm1/Documents/OS2/
/home/vm1/Documents/OS2/libc062.dll
/home/vm1/Documents/OS2/VBoxService.exe
/home/vm1/Documents/OS2/libc06.dll
/home/vm1/Documents/OS2/readme.txt
/home/vm1/Documents/OS2/TRANS.TBL
/home/vm1/Documents/OS2/gengradd.dll
/home/vm1/Documents/OS2/VboxOs2AdditionsInstall.exe
/home/vm1/Documents/OS2/libc064.dll
/home/vm1/Documents/OS2/VBoxControl.exe
/home/vm1/Documents/OS2/VBoxMouse.sys
/home/vm1/Documents/OS2/libc065.dll
/home/vm1/Documents/OS2/VBoxReplaceDll.exe
/home/vm1/Documents/OS2/VBoxGuest.sys
/home/vm1/Documents/OS2/libc066.dll
/home/vm1/Documents/OS2/VBoxSF.ifs
/home/vm1/Documents/OS2/libc063.dll
/home/vm1/Documents/OS2/libc061.dll
/home/vm1/Documents/windows11-bypass.reg
/home/vm1/Documents/VBoxWindowsAdditions-x86.exe
/home/vm1/Documents/runasroot.sh
/home/vm1/Documents/TRANS.TBL
/home/vm1/Documents/NT3x/
/home/vm1/Documents/NT3x/VBoxService.exe
```

q) **Rm:** Command rm is used to remove or delete the files.



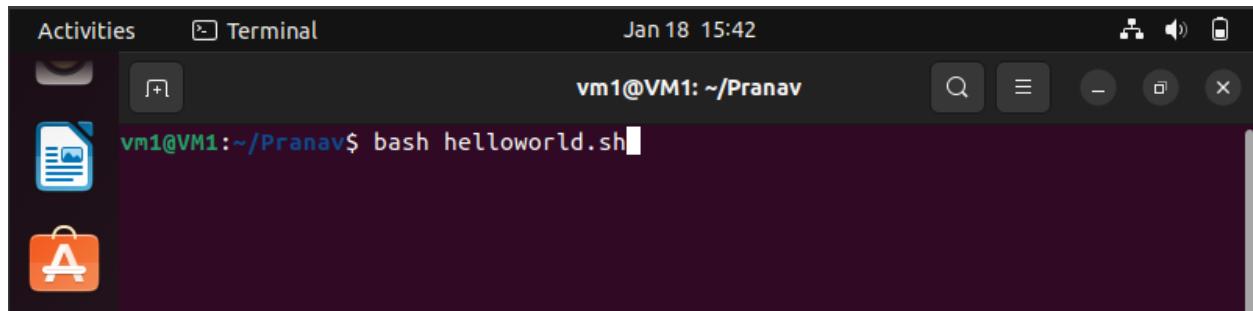
```
Activities Terminal Jan 18 15:33
vm1@VM1:~/Desktop$ cd
vm1@VM1:~$ touch tryrm.txt
vm1@VM1:~$ ls
Desktop  MP  Pranav  pra.text  Templates
Documents  Music  pranavkey1  Public  tryrm.txt
Downloads  Pictures  pranavkey1.pub  snap  Videos
vm1@VM1:~$ rm tryrm.txt
vm1@VM1:~$ ls | grep 'try'
vm1@VM1:~$
```

r) **Cat:** cat is used to print the content of the file in the output window.

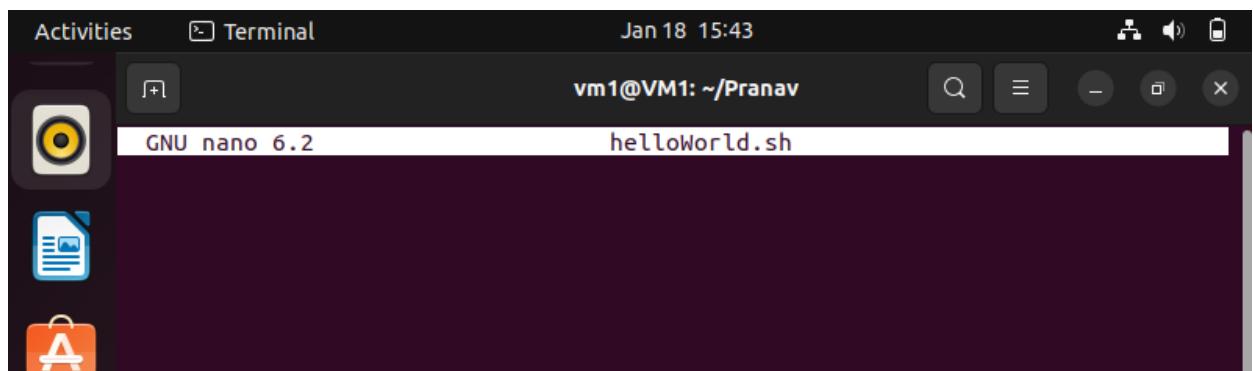


```
Activities Terminal Jan 18 15:34
vm1@VM1: ~
aaaa
bbbb
cccc
dddd
eeee
ffff
gggg
hhhh
iiii
jjjj
vm1@VM1: ~
```

s) **Bash**: Command bash is used to run shell scripts



```
Activities Terminal Jan 18 15:42
vm1@VM1: ~/Pranav
vm1@VM1: ~/Pranav$ bash helloworld.sh
```



```
Activities Terminal Jan 18 15:43
vm1@VM1: ~/Pranav
GNU nano 6.2
```

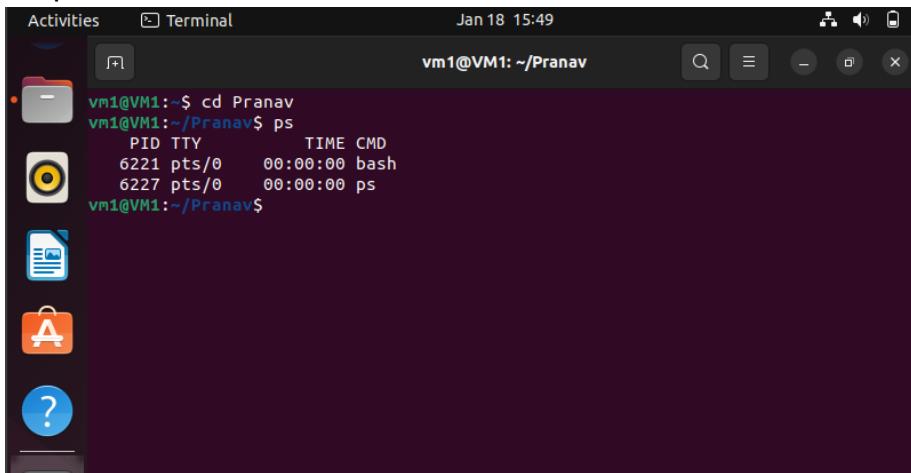
t) **More**: We can say it is a manual scroll, Without more system prints every output on your screen, and the user has to scroll the screen to see the outputs.

```
File Edit View Search Terminal Help
=
@
@]]
111.txt
1.c
1.sh
222.txt
2.sh
55.sh
abc.tar
checkinte.sh
cp1.txt
cp2.txt
dd.txt
Desktop
disk-benchmark-background-log.txt
disk-benchmark-background.sh
Documents
Downloads
examples.desktop
example.txt
firstshell.sh
Hello
helloworld.py
matplotlib.py
Music
nested-loops.sh
--More--
```

- u) **Watch:** Given command will display the output of the given command after a given interval and it will run into a loop.

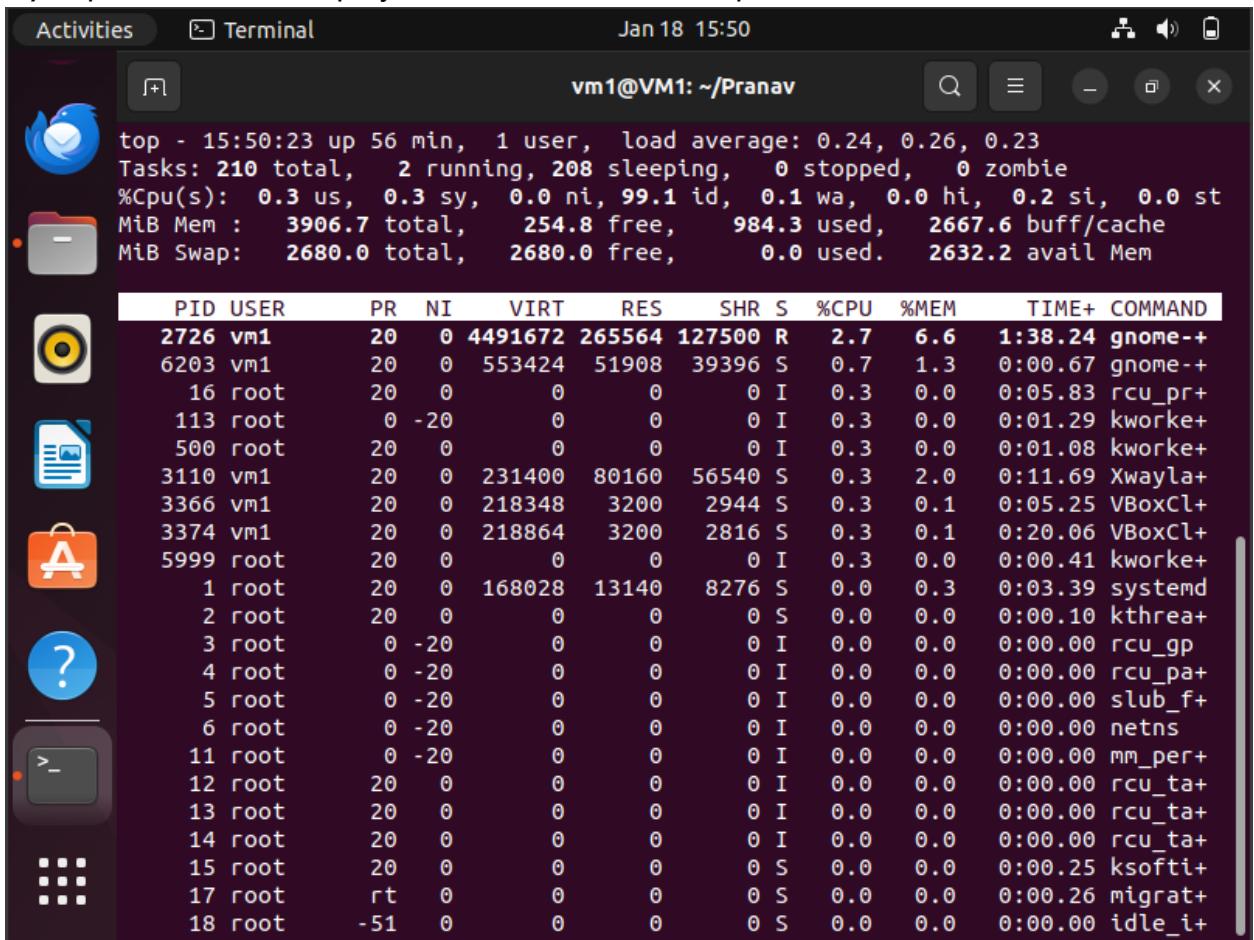
```
Activities Terminal Jan 18 15:48
vm1@VM1: ~ VM1: Thu Jan 18 15:48:10 2024
Every 2.0s: free -m
total        used        free      shared  buff/cache   availabl
e
Mem:      3906         979        261          42        2665        263
7
Swap:     2679          0        2679
```

- v) **Ps:** It is used to get information about currently running processes and their process ids.



```
vm1@VM1:~$ cd Pranav
vm1@VM1:~/Pranav$ ps
  PID TTY      TIME CMD
 6221 pts/0    00:00:00 bash
 6227 pts/0    00:00:00 ps
vm1@VM1:~/Pranav$
```

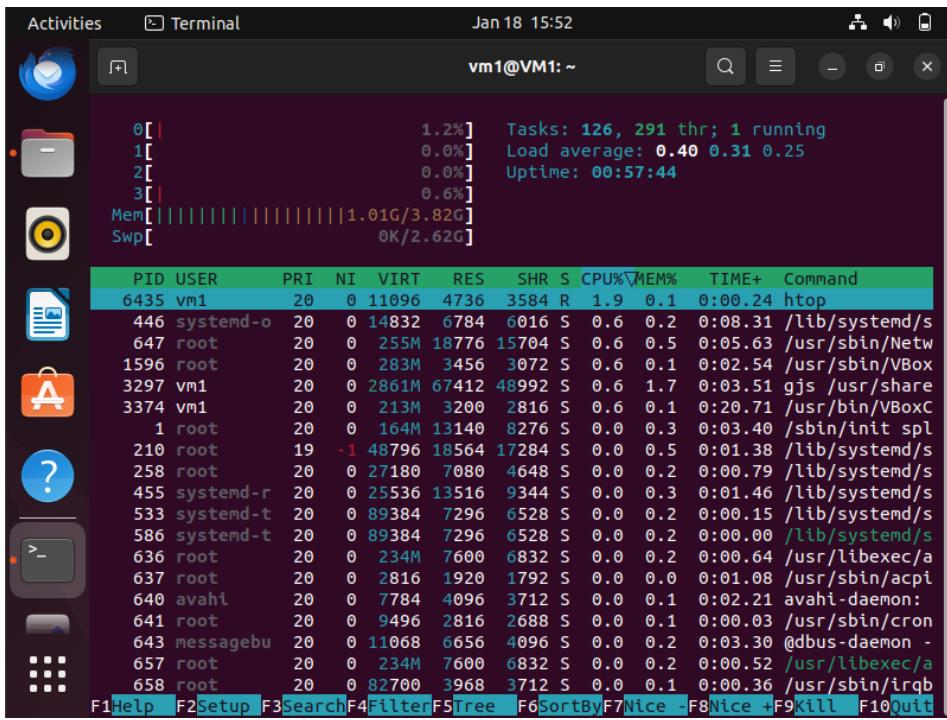
- w) **Top:** It is used to display information about CPU processes



```
top - 15:50:23 up 56 min,  1 user,  load average: 0.24, 0.26, 0.23
Tasks: 210 total,  2 running, 208 sleeping,  0 stopped,  0 zombie
%Cpu(s): 0.3 us, 0.3 sy, 0.0 ni, 99.1 id, 0.1 wa, 0.0 hi, 0.2 si, 0.0 st
MiB Mem : 3906.7 total, 254.8 free, 984.3 used, 2667.6 buff/cache
MiB Swap: 2680.0 total, 2680.0 free, 0.0 used. 2632.2 avail Mem

          PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM TIME+ COMMAND
 2726 vm1      20   0 4491672 265564 127500 R  2.7  6.6 1:38.24 gnome+-+
 6203 vm1      20   0 553424 51908 39396 S  0.7  1.3 0:00.67 gnome+-+
 16 root      20   0      0      0      0 I  0.3  0.0 0:05.83 rcu_pr+
 113 root     0 -20      0      0      0 I  0.3  0.0 0:01.29 kworker+
 500 root     20   0      0      0      0 I  0.3  0.0 0:01.08 kworker+
 3110 vm1      20   0 231400 80160 56540 S  0.3  2.0 0:11.69 Xwayla+
 3366 vm1      20   0 218348 3200 2944 S  0.3  0.1 0:05.25 VBoxCl+
 3374 vm1      20   0 218864 3200 2816 S  0.3  0.1 0:20.06 VBoxCl+
 5999 root     20   0      0      0      0 I  0.3  0.0 0:00.41 kworker+
 1 root      20   0 168028 13140 8276 S  0.0  0.3 0:03.39 systemd
 2 root      20   0      0      0      0 S  0.0  0.0 0:00.10 kthrea+
 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_pa+
 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
 11 root    0 -20      0      0      0 I  0.0  0.0 0:00.00 mm_per+
 12 root    20   0      0      0      0 I  0.0  0.0 0:00.00 rcu_ta+
 13 root    20   0      0      0      0 I  0.0  0.0 0:00.00 rcu_ta+
 14 root    20   0      0      0      0 I  0.0  0.0 0:00.00 rcu_ta+
 15 root    20   0      0      0      0 S  0.0  0.0 0:00.25 ksofti+
 17 root    rt   0      0      0      0 S  0.0  0.0 0:00.26 migrat+
 18 root   -51   0      0      0      0 S  0.0  0.0 0:00.00 idle_i+
```

x) **Htop**: It's a user-friendly process viewer, That interactively shows process details.

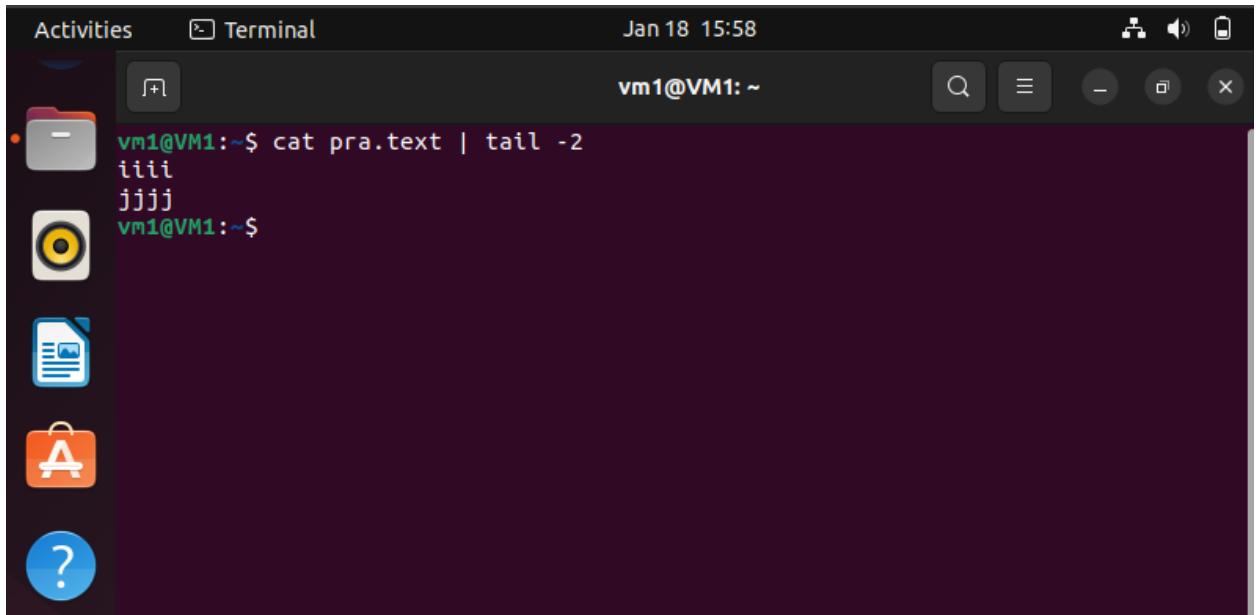


y) **Gcc**: It is used to execute c programs.

The screenshot shows a terminal window with the title "vm1@VM1: ~". The terminal prompt is "vm1@VM1:~\$". The user has run the command "sudo gcc 1.c", which is displayed in the terminal window. The terminal window is part of a desktop environment with a dark theme, showing icons for various applications on the left.

```
vm1@VM1:~$ sudo gcc 1.c
```

z) **Tail:** Tail is used to get the last rows or bottom rows from file.

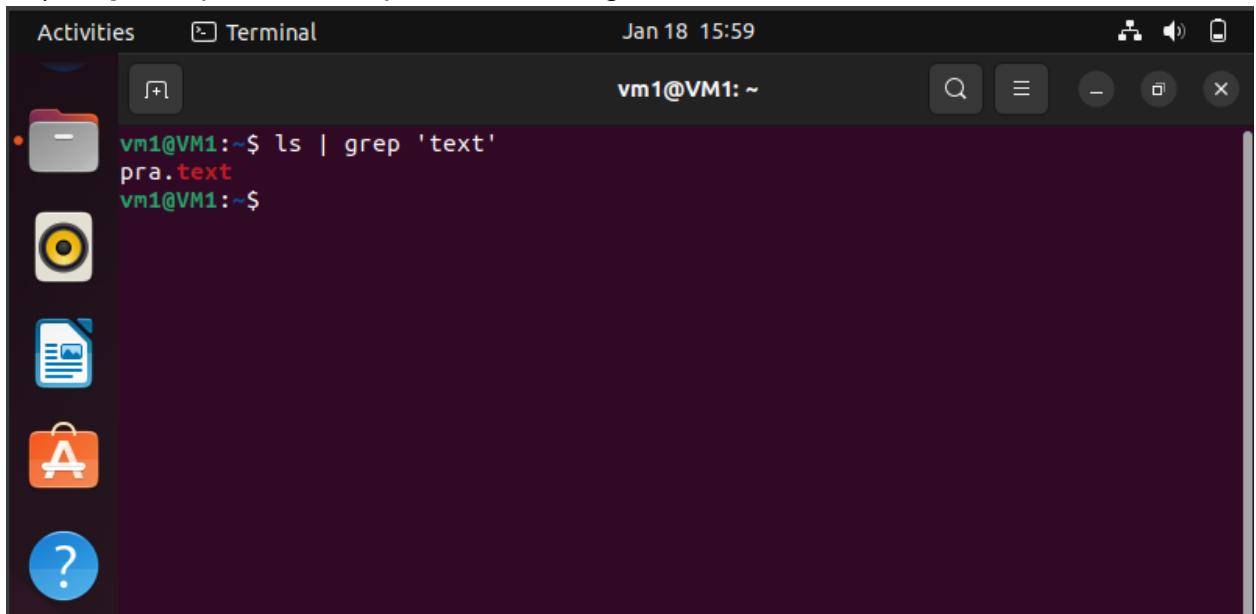


A screenshot of a Linux desktop environment showing a terminal window. The terminal window title is "Terminal" and the status bar shows "Jan 18 15:58" and "vm1@VM1: ~". The terminal content is as follows:

```
Activities Terminal Jan 18 15:58
vm1@VM1:~$ cat pra.text | tail -2
iiit
jjjj
vm1@VM1:~$
```

The terminal window is part of a desktop interface with a dark theme, showing other application icons in the dock.

aa) **Grep:** Grep is used for pattern matching.

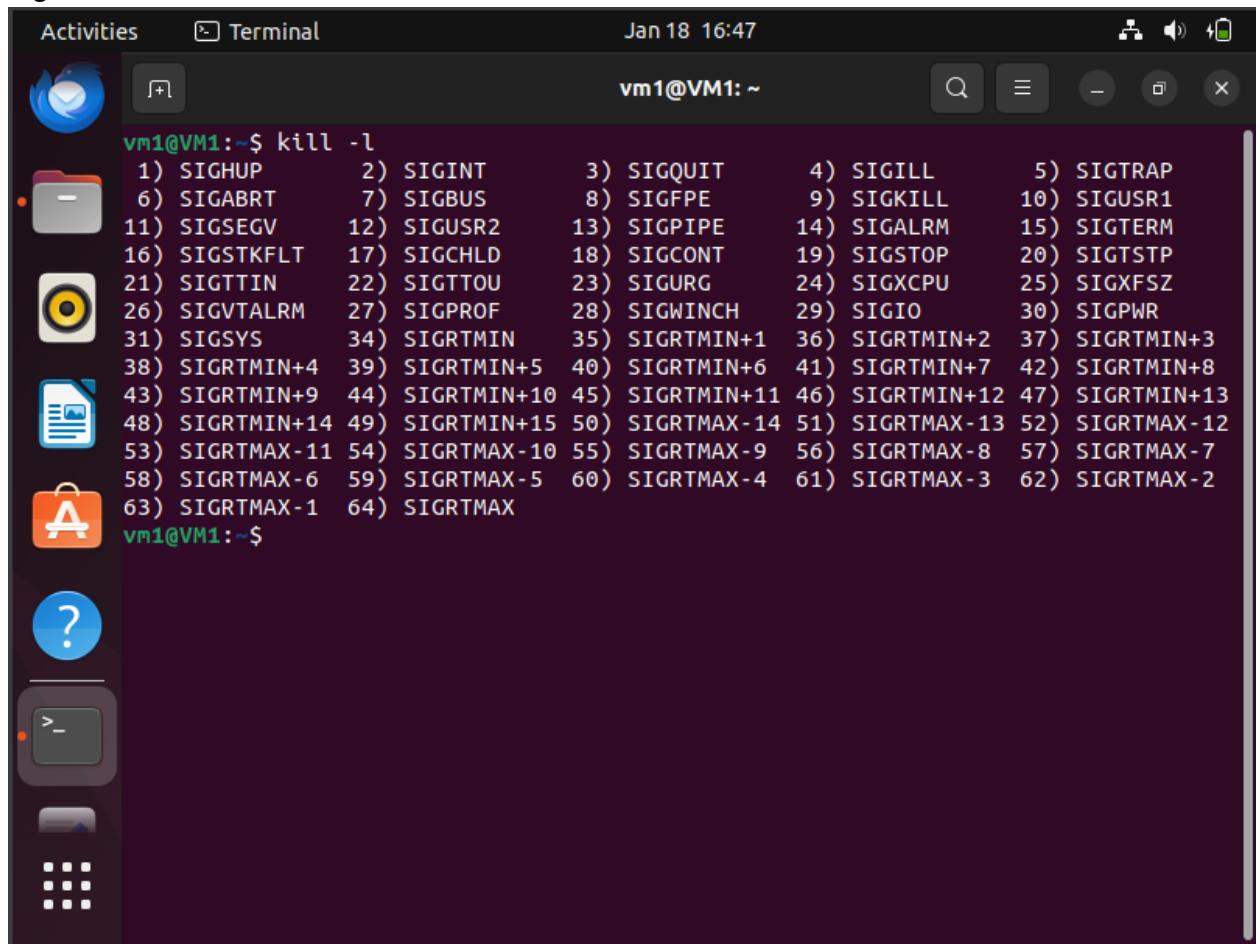


A screenshot of a Linux desktop environment showing a terminal window. The terminal window title is "Terminal" and the status bar shows "Jan 18 15:59" and "vm1@VM1: ~". The terminal content is as follows:

```
Activities Terminal Jan 18 15:59
vm1@VM1:~$ ls | grep 'text'
pra.text
vm1@VM1:~$
```

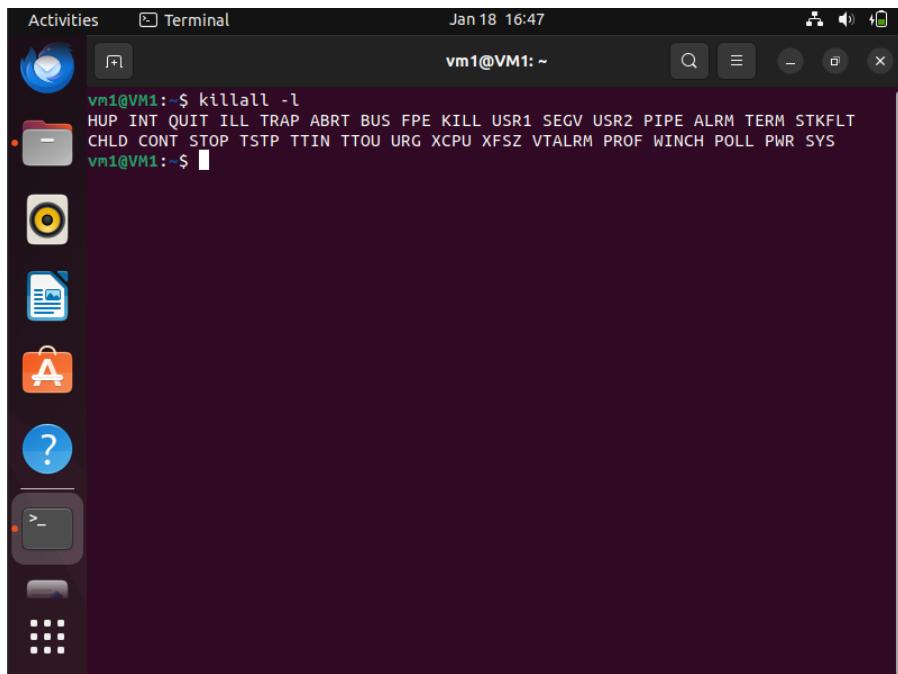
The terminal window is part of a desktop interface with a dark theme, showing other application icons in the dock.

bb) **Kill**: It is used to get information about the signals and to kill the specific signals.

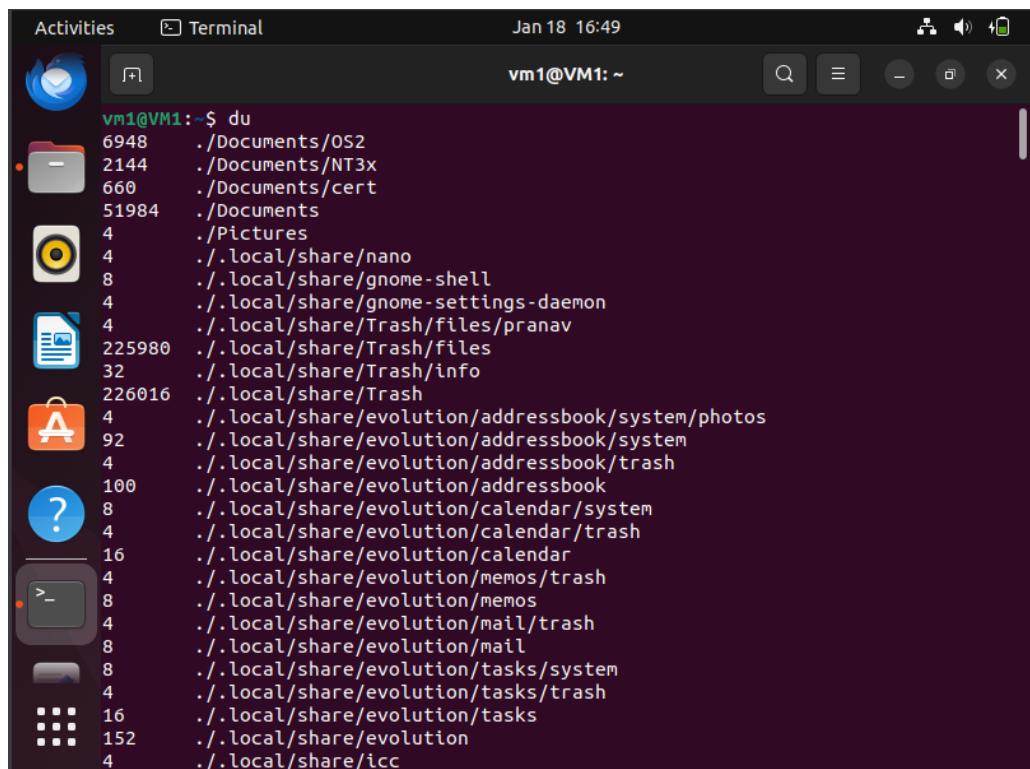


```
Activities Terminal Jan 18 16:47
vm1@VM1:~$ kill -l
 1) SIGHUP      2) SIGINT      3) SIGQUIT      4) SIGILL      5) SIGTRAP
 6) SIGABRT     7) SIGBUS      8) SIGFPE       9) SIGKILL     10) SIGUSR1
11) SIGSEGV     12) SIGUSR2     13) SIGPIPE     14) SIGALRM     15) SIGTERM
16) SIGSTKFLT   17) SIGCHLD     18) SIGCONT     19) SIGSTOP     20) SIGTSTP
21) SIGTTIN     22) SIGTTOU     23) SIGURG      24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM   27) SIGPROF     28) SIGWINCH    29) SIGIO       30) SIGPWR
31) SIGSYS      34) SIGRTMIN    35) SIGRTMIN+1  36) SIGRTMIN+2  37) SIGRTMIN+3
38) SIGRTMIN+4  39) SIGRTMIN+5  40) SIGRTMIN+6  41) SIGRTMIN+7  42) SIGRTMIN+8
43) SIGRTMIN+9  44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7
58) SIGRTMAX-6  59) SIGRTMAX-5  60) SIGRTMAX-4  61) SIGRTMAX-3  62) SIGRTMAX-2
63) SIGRTMAX-1  64) SIGRTMAX
vm1@VM1:~$
```

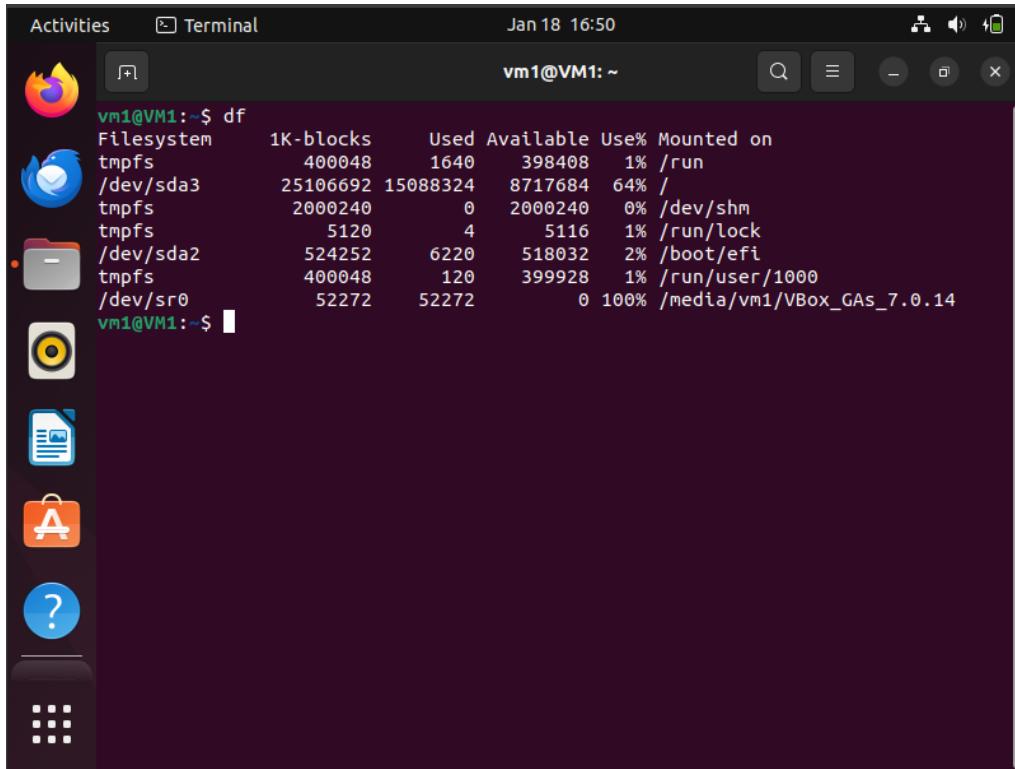
cc) **Killall** : Its function is to kill all the process using the command.



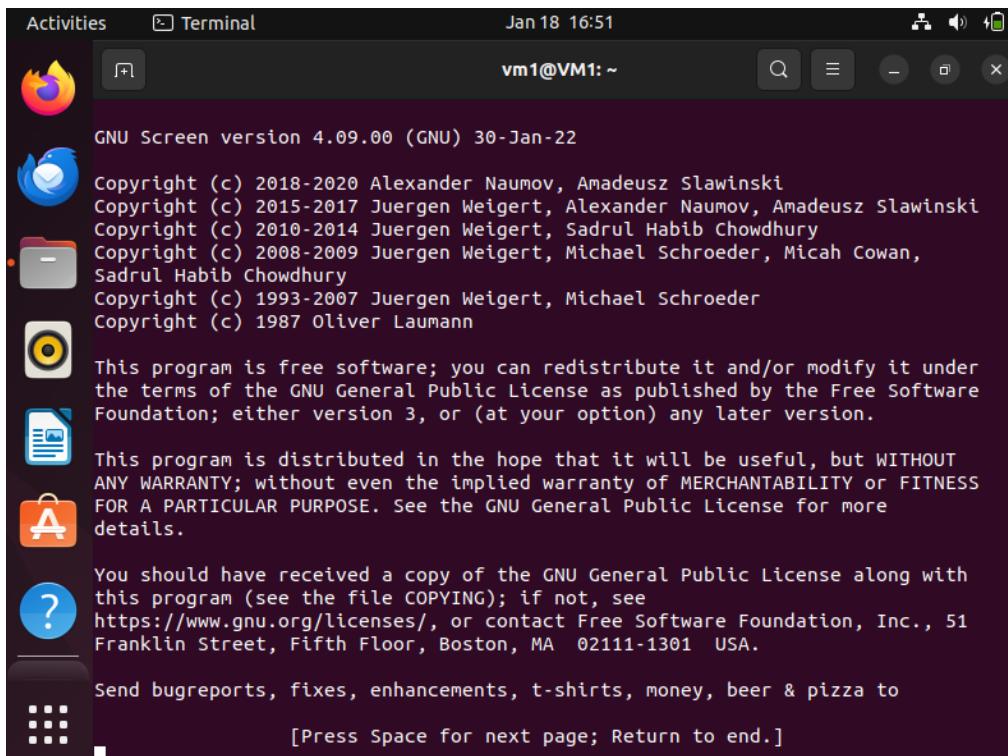
dd)**Du**: It shows the file space usage.



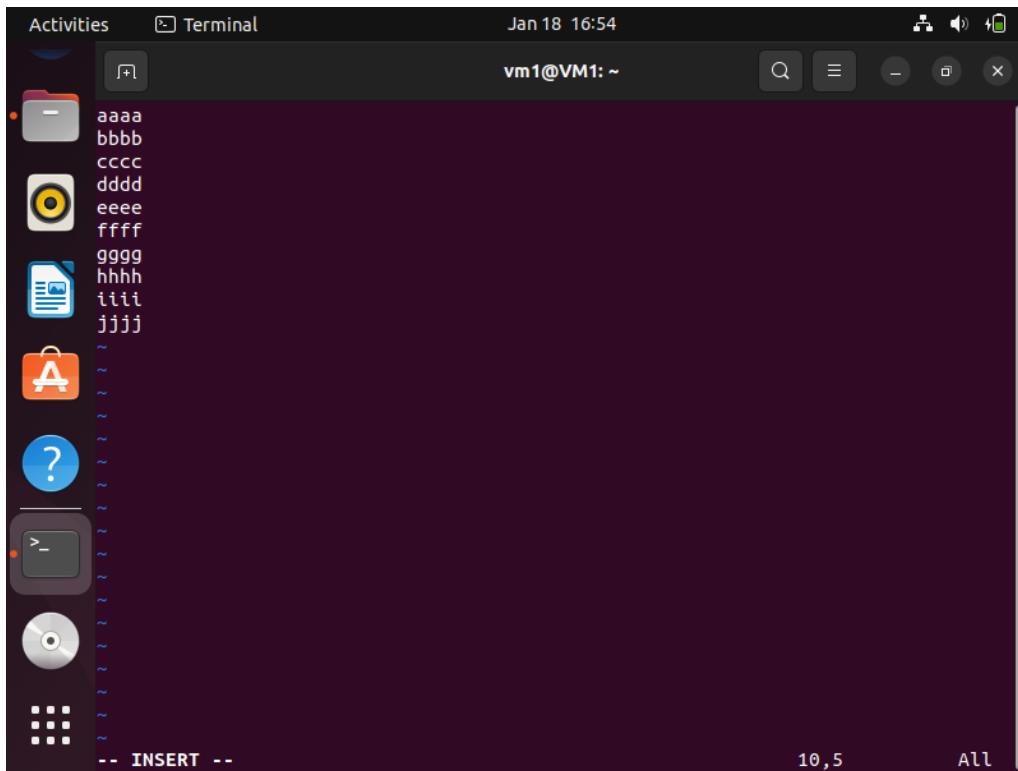
ee)**Df**: It is the same as that of DU, which is used to get disk space usage.



ff) **Screen**: Multiply given terminal for multiple processes.

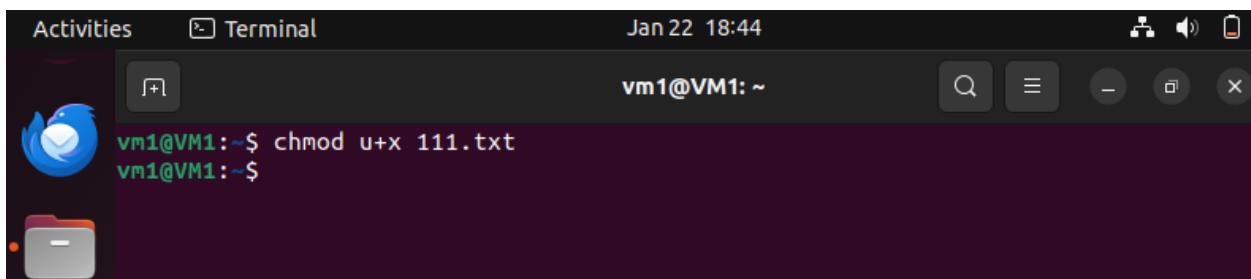


gg)Vim: Used to print or edit content of file using console.



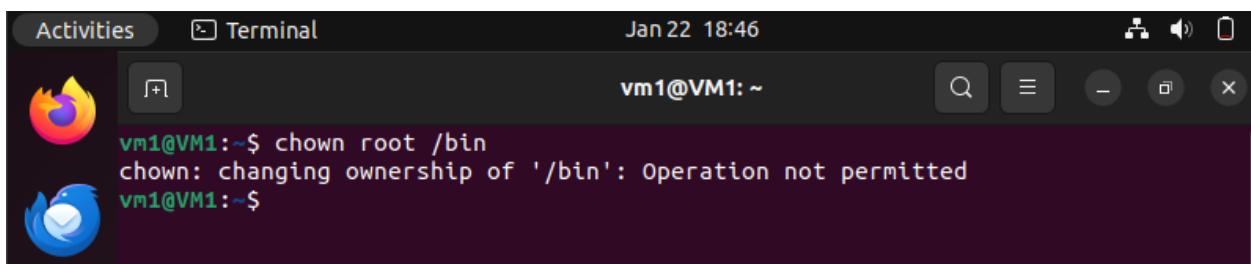
```
aaaa
bbbb
cccc
dddd
eeee
ffff
gggg
hhhh
iiii
jjjj
~
~
~
~
?
>_
~
```

hh) **Chmod**: Chmod is used to give permissions to file, permissions like read write and execute access.



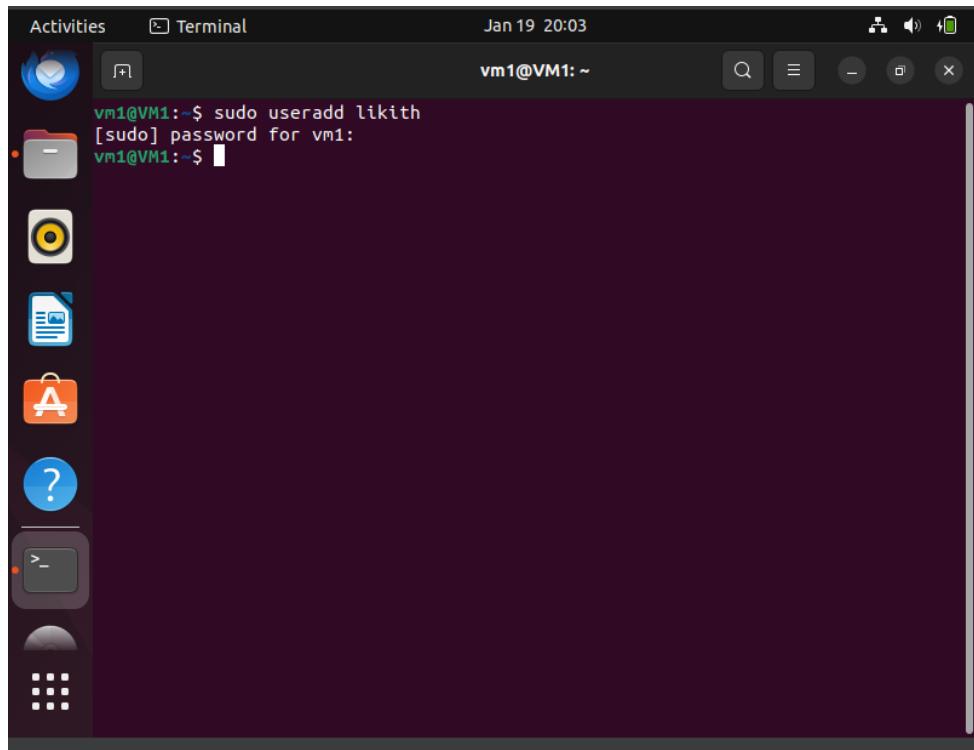
```
vm1@VM1:~$ chmod u+x 111.txt
vm1@VM1:~$
```

ii) Chown : It is used to change file owner or group.



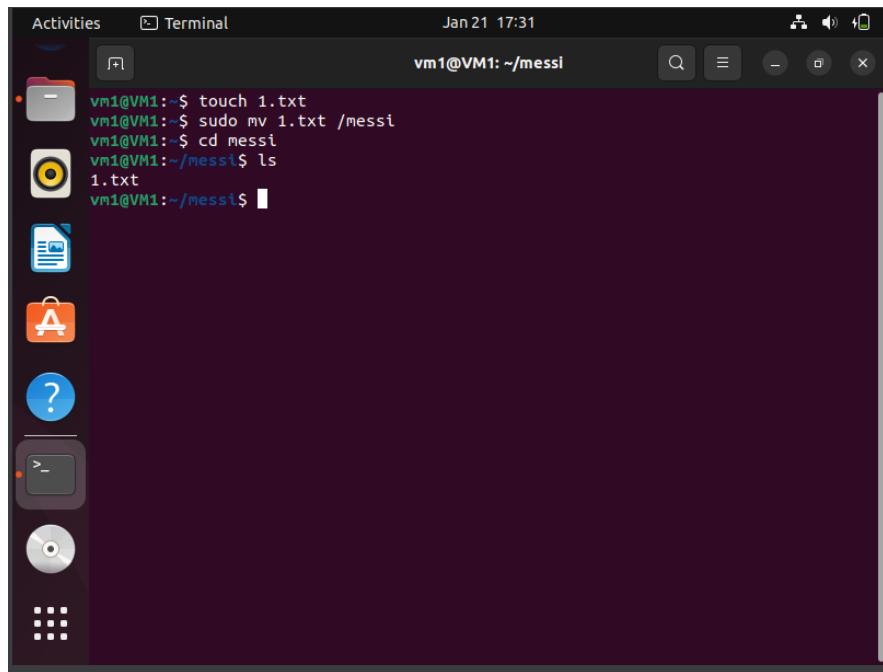
```
vm1@VM1:~$ chown root /bin
chown: changing ownership of '/bin': Operation not permitted
vm1@VM1:~$
```

jj) Useradd : Used to add new user to existing system.



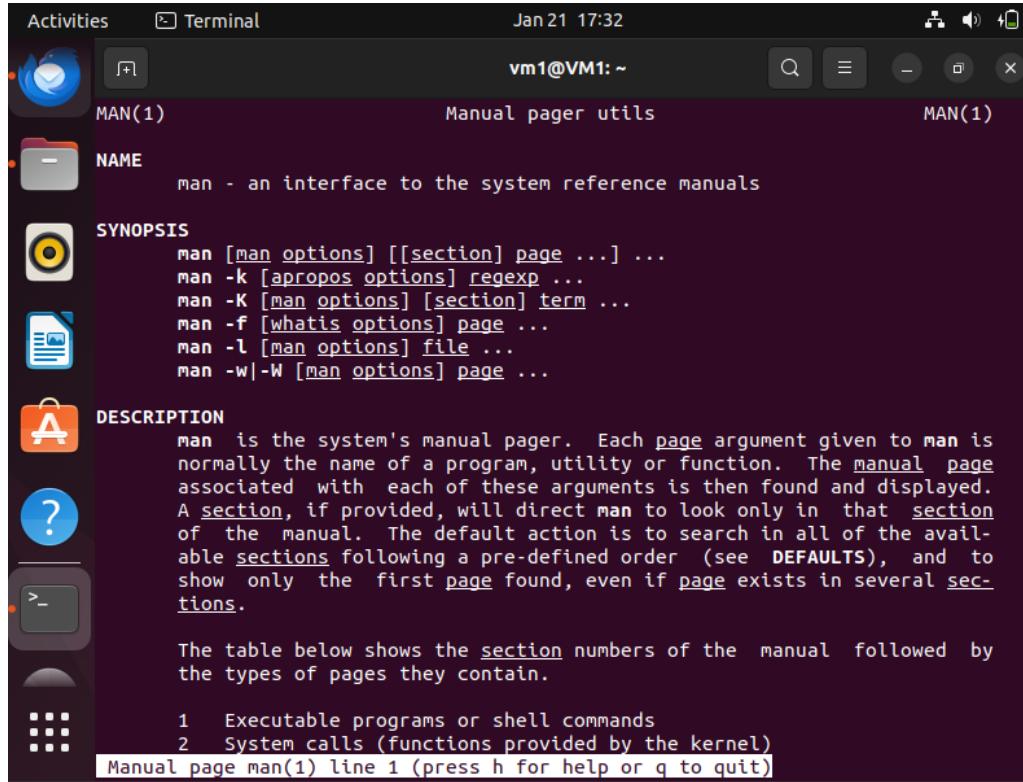
```
Activities Terminal Jan 19 20:03
vm1@VM1:~$ sudo useradd likith
[sudo] password for vm1:
vm1@VM1:~$
```

kk) **Mv**: It is used to move the file from one folder to another.



```
Activities Terminal Jan 21 17:31
vm1@VM1:~$ touch 1.txt
vm1@VM1:~$ sudo mv 1.txt /messi
vm1@VM1:~$ cd messi
vm1@VM1:~/messi$ ls
1.txt
vm1@VM1:~/messi$
```

II) **Man:** It is used to get user manual information for given command, See below output for man ls command.



Activities Terminal Jan 21 17:32

MAN(1) Manual pager utils MAN(1)

NAME man - an interface to the system reference manuals

SYNOPSIS

```
man [man options] [[section] page ...] ...
man -k [apropos options] regexp ...
man -K [man options] [section] term ...
man -f [whatis options] page ...
man -l [man options] file ...
man -w|-W [man options] page ...
```

DESCRIPTION

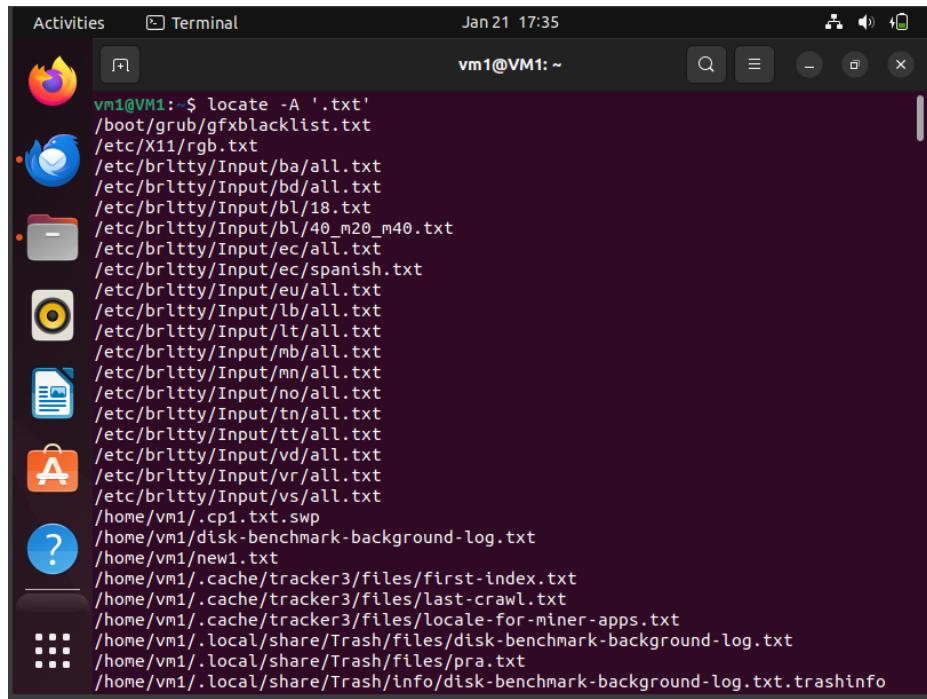
`man` is the system's manual pager. Each `page` argument given to `man` is normally the name of a program, utility or function. The `manual page` associated with each of these arguments is then found and displayed. A `section`, if provided, will direct `man` to look only in that `section` of the manual. The default action is to search in all of the available `sections` following a pre-defined order (see `DEFAULTS`), and to show only the first `page` found, even if `page` exists in several `sections`.

The table below shows the `section` numbers of the manual followed by the types of pages they contain.

1	Executable programs or shell commands
2	System calls (functions provided by the kernel)

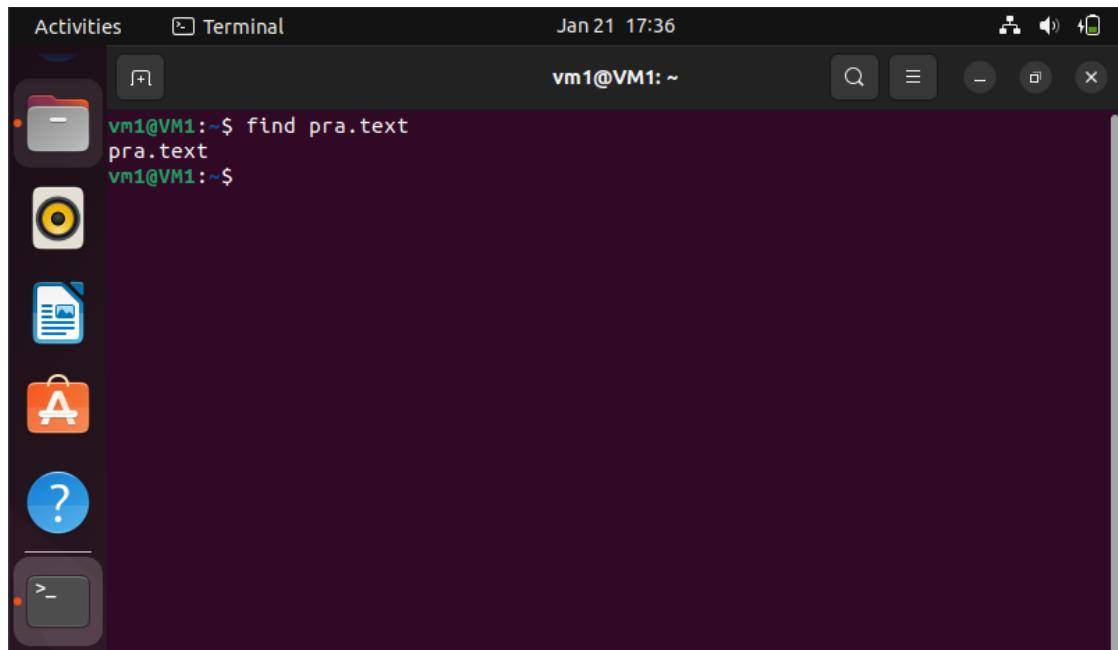
Manual page man(1) line 1 (press h for help or q to quit)

mm) **Locate:** It is used to find the files which matches given pattern.



```
Activities Terminal Jan 21 17:35
vm1@VM1:~$ locate -A '.txt'
/boot/grub/gfxblacklist.txt
/etc/X11/rgb.txt
/etc/brltty/Input/ba/all.txt
/etc/brltty/Input/bd/all.txt
/etc/brltty/Input/bl/18.txt
/etc/brltty/Input/bl/40_m20_m40.txt
/etc/brltty/Input/ec/all.txt
/etc/brltty/Input/ec/spanish.txt
/etc/brltty/Input/eu/all.txt
/etc/brltty/Input/lb/all.txt
/etc/brltty/Input/lt/all.txt
/etc/brltty/Input(mb/all.txt
/etc/brltty/Input/mn/all.txt
/etc/brltty/Input/no/all.txt
/etc/brltty/Input/tn/all.txt
/etc/brltty/Input/tt/all.txt
/etc/brltty/Input/vd/all.txt
/etc/brltty/Input/vr/all.txt
/etc/brltty/Input/vs/all.txt
/home/vm1/.cp1.txt.swp
/home/vm1/disk-benchmark-background-log.txt
/home/vm1/new1.txt
/home/vm1/.cache/tracker3/files/first-index.txt
/home/vm1/.cache/tracker3/files/last-crawl.txt
/home/vm1/.cache/tracker3/files/locale-for-miner-apps.txt
/home/vm1/.local/share/Trash/files/disk-benchmark-background-log.txt
/home/vm1/.local/share/Trash/files/pr.a.txt
/home/vm1/.local/share/Trash/info/disk-benchmark-background-log.txt.trashinfo
```

nn) **Find:** Used to search a file in given folder.



```
Activities Terminal Jan 21 17:36
vm1@VM1:~$ find pra.text
pra.text
vm1@VM1:~$
```

oo) **Sed:** it is used to edit or transform the text in console.

Activities Terminal Jan 21 17:41

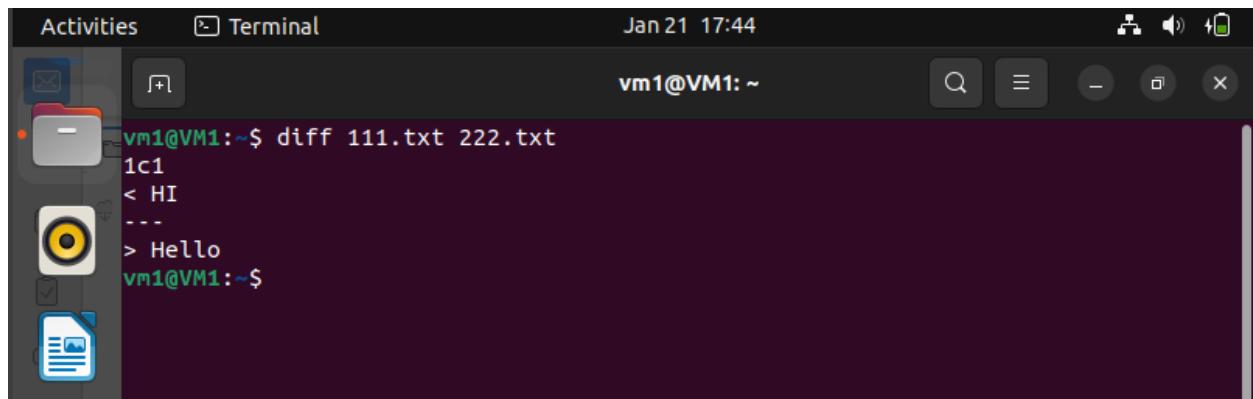
```
vm1@VM1:~$ sed 's/unix/linux/' nature.txt
Nature is what we see and feel around us.
Nature is like our mother who never scolds us unless we do anything wrong.
I am happy to interact with nature and also happy to be a part of this earth.
The nature is very attractive and full of my favorite green color.
It provides us the air to breathe, the water to drink, the soil to make a home,
and the land to stay.
Nature gives us fruits, vegetables, and grains to eat.
We should enjoy nature without disturbing its balance.
Nature is a precious and notable gift given to us by god.
Nature is our best friend. We should do plantation and nurturing to keep it healthy.
We must keep our nature healthy, and peaceful.
```

pp) Awk: Used for pattern matching.

Activities Terminal Jan 21 17:42

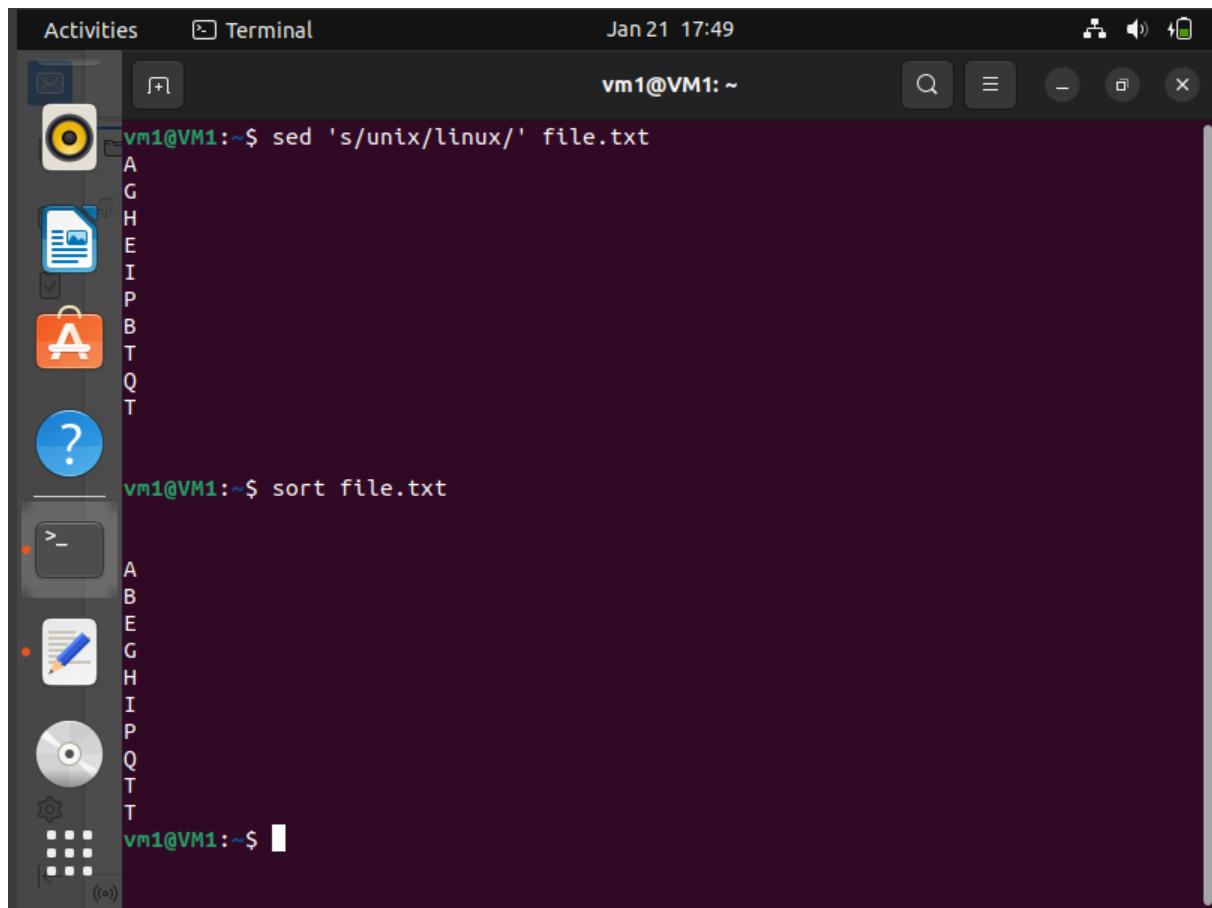
```
vm1@VM1:~$ awk '{print $1,$4}' nature.txt
Nature we
Nature our
I to
I to
The very
It the
Nature fruits,
We nature
Nature precious
Nature best
We our
vm1@VM1:~$
```

qq) Diff: It is used to compare files line by line.



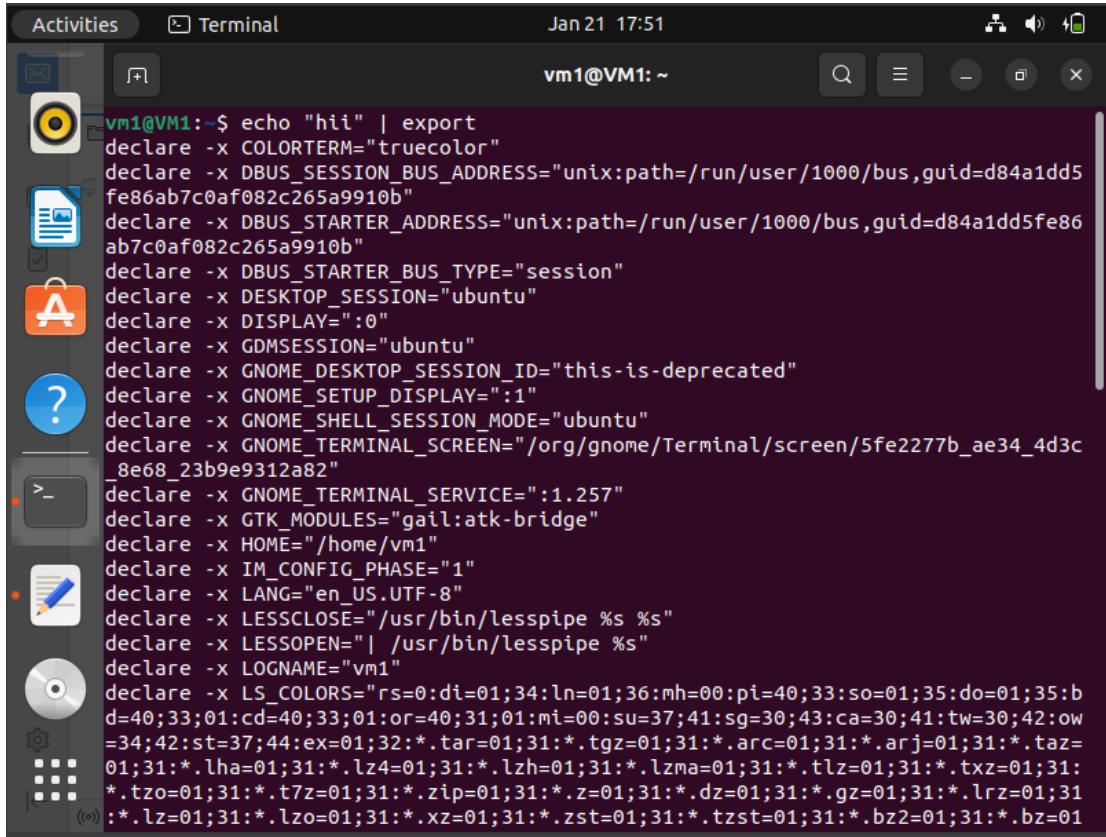
```
vm1@VM1:~$ diff 111.txt 222.txt
1c1
< HI
---
> Hello
vm1@VM1:~$
```

rr) **Sort**: Sort is used to sort records according to ascending or descending order.



```
vm1@VM1:~$ sed 's/unix/linux/' file.txt
A
G
H
E
I
P
B
T
Q
T
vm1@VM1:~$ sort file.txt
A
B
E
G
H
I
P
Q
T
T
vm1@VM1:~$
```

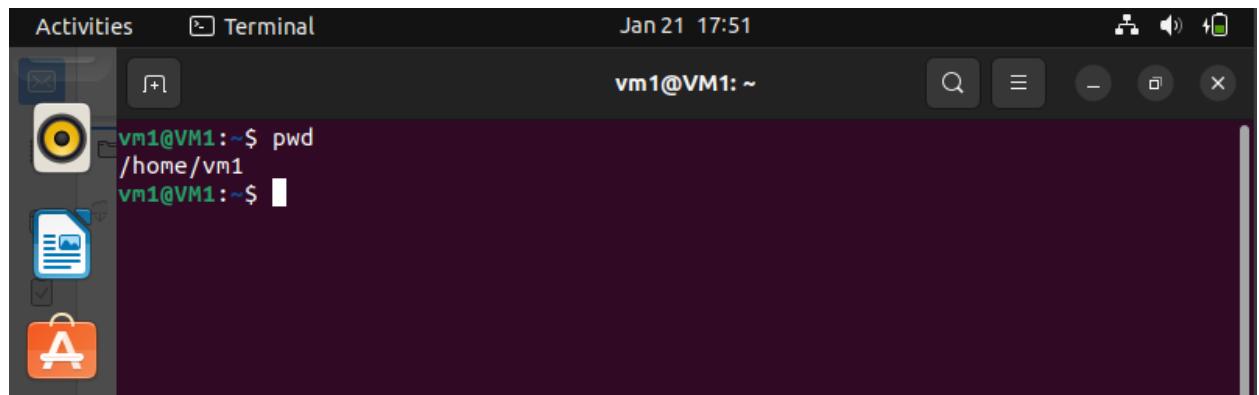
ss) **Export**: Used to export the value of variables available to child processes or other resources.



A screenshot of an Ubuntu desktop environment. A terminal window is open in the center, showing the command 'echo "hi" | export' and its output. The output lists various environment variables being set, such as 'COLORTERM', 'DBUS\_SESSION\_BUS\_ADDRESS', 'DBUS\_STARTER\_ADDRESS', 'DISPLAY', 'GDMSESSION', 'GNOME\_DESKTOP\_SESSION\_ID', 'GNOME\_SETUP\_DISPLAY', 'GNOME\_SHELL\_SESSION\_MODE', 'GNOME\_TERMINAL\_SCREEN', 'GNOME\_TERMINAL\_SERVICE', 'GTK\_MODULES', 'HOME', 'IM\_CONFIG\_PHASE', 'LANG', 'LESSCLOSE', 'LESSOPEN', and 'LOGNAME'. The terminal window is titled 'vm1@VM1: ~'. The desktop background shows a dark theme with icons for the Dash, Home, and other applications.

```
vm1@VM1:~$ echo "hi" | export
declare -x COLORTERM="truecolor"
declare -x DBUS_SESSION_BUS_ADDRESS="unix:path=/run/user/1000/bus,guid=d84a1dd5
fe86ab7c0af082c265a9910b"
declare -x DBUS_STARTER_ADDRESS="unix:path=/run/user/1000/bus,guid=d84a1dd5fe86
ab7c0af082c265a9910b"
declare -x DBUS_STARTER_BUS_TYPE="session"
declare -x DESKTOP_SESSION="ubuntu"
declare -x DISPLAY=:0"
declare -x GDMSESSION="ubuntu"
declare -x GNOME_DESKTOP_SESSION_ID="this-is-deprecated"
declare -x GNOME_SETUP_DISPLAY=:1"
declare -x GNOME_SHELL_SESSION_MODE="ubuntu"
declare -x GNOME_TERMINAL_SCREEN="/org/gnome/Terminal/screen/5fe2277b_ae34_4d3c
_8e68_23b9e9312a82"
declare -x GNOME_TERMINAL_SERVICE=:1.257"
declare -x GTK_MODULES="gail:atk-bridge"
declare -x HOME="/home/vm1"
declare -x IM_CONFIG_PHASE="1"
declare -x LANG="en_US.UTF-8"
declare -x LESSCLOSE="/usr/bin/lesspipe %s %s"
declare -x LESSOPEN="| /usr/bin/lesspipe %s"
declare -x LOGNAME="vm1"
declare -x LS_COLORS="rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:b
d=40;33:01:cd=40;33:01:or=40;31:01:mi=00:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow
=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=
01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:
*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31
:*.lz=01;31:*.lzo=01;31:*.xz=01;31:*.zst=01;31:*.tzst=01;31:*.bz2=01;31:*.bz=01
```

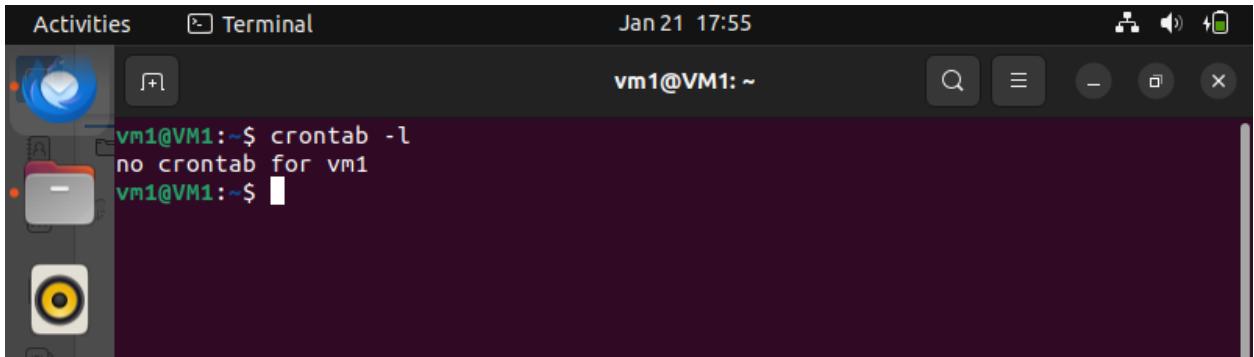
tt) **Pwd**: Used to print working directory.



A screenshot of an Ubuntu desktop environment. A terminal window is open in the center, showing the command 'pwd' and its output. The output shows the current working directory as '/home/vm1'. The terminal window is titled 'vm1@VM1: ~'. The desktop background shows a dark theme with icons for the Dash, Home, and other applications.

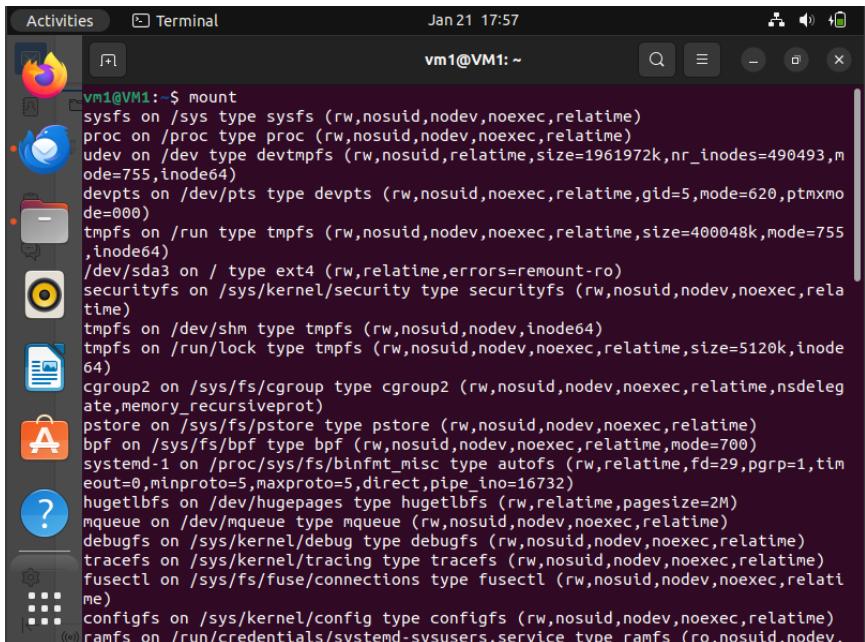
```
vm1@VM1:~$ pwd
/home/vm1
vm1@VM1:~$
```

uu) **Crontab**: Used to maintain crontab program files for individual users.



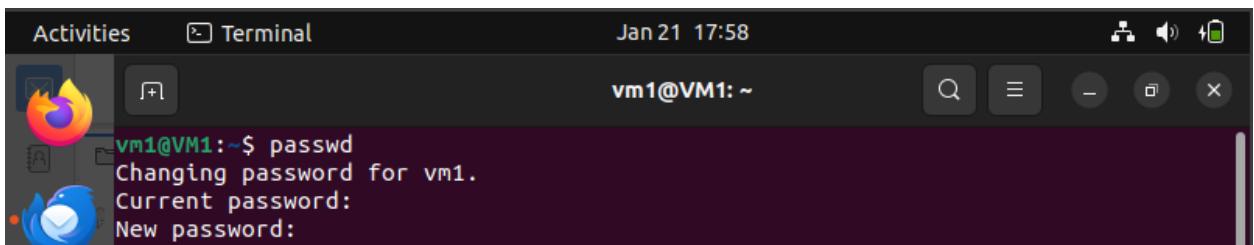
```
Activities Terminal Jan 21 17:55
vm1@VM1:~$ crontab -l
no crontab for vm1
vm1@VM1:~$
```

vv) **Mount**: All files in Ubuntu are maintained in a tree, Mount is used to attach filesystem found on other system to tree.



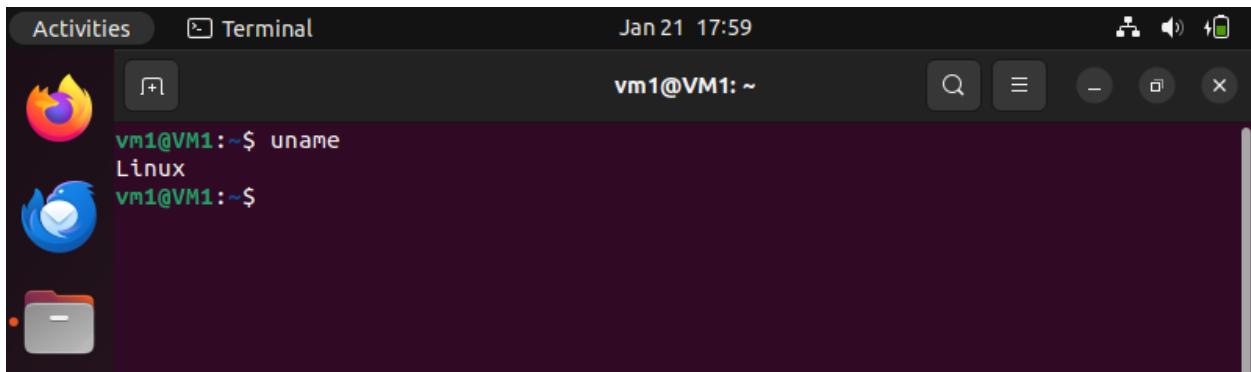
```
Activities Terminal Jan 21 17:57
vm1@VM1:~$ mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,nosuid,relatime,size=1961972k,nr_inodes=490493,mode=755,inode64)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,noexec,relatime,size=400048k,mode=755,inode64)
/dev/sda3 on / type ext4 (rw,relatime,errors=remount-ro)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,inode64)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k,inode64)
cgroup2 on /sys/fs/cgroup type cgroup2 (rw,nosuid,nodev,noexec,relatime,nsdelegate,memory_recursiveprot)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
bpf on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=29,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=16732)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,pagesize=2M)
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime)
debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relatime)
tracefs on /sys/kernel/tracing type tracefs (rw,nosuid,nodev,noexec,relatime)
fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relatime)
configfs on /sys/kernel/config type configfs (rw,nosuid,nodev,noexec,relatime)
ramfs on /run/credentials/systemd-sysusers.service type ramfs (ro,nosuid,nodev,
```

ww) **Passwd**: Given command is used to change the password.



```
Activities Terminal Jan 21 17:58
vm1@VM1:~$ passwd
Changing password for vm1.
Current password:
New password:
```

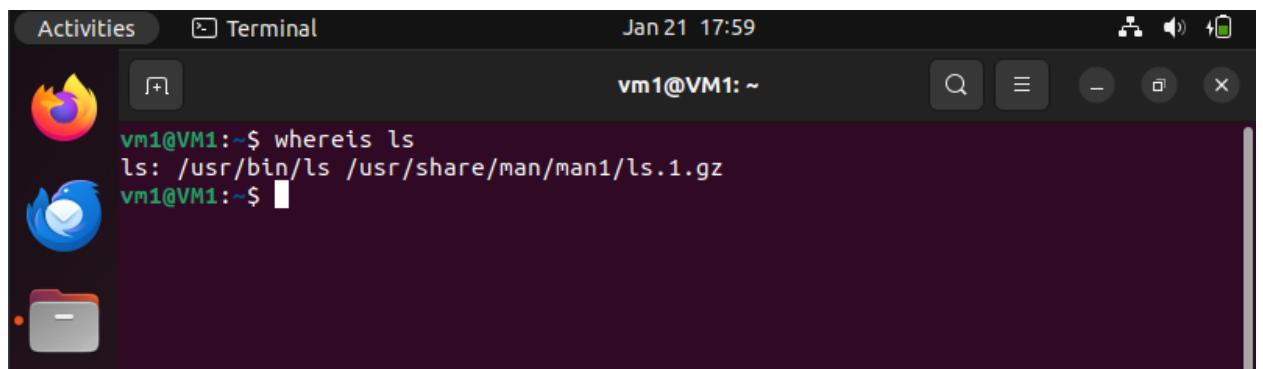
xx) **Uname**: It is used to get username.



A screenshot of an Ubuntu desktop environment. On the left, there is a dock with icons for the Dash, Home, and a folder. The main window is a terminal window titled 'Terminal' with the status bar showing 'Jan 21 17:59' and 'vm1@VM1: ~'. The terminal window contains the following text:

```
vm1@VM1:~$ uname
Linux
vm1@VM1:~$
```

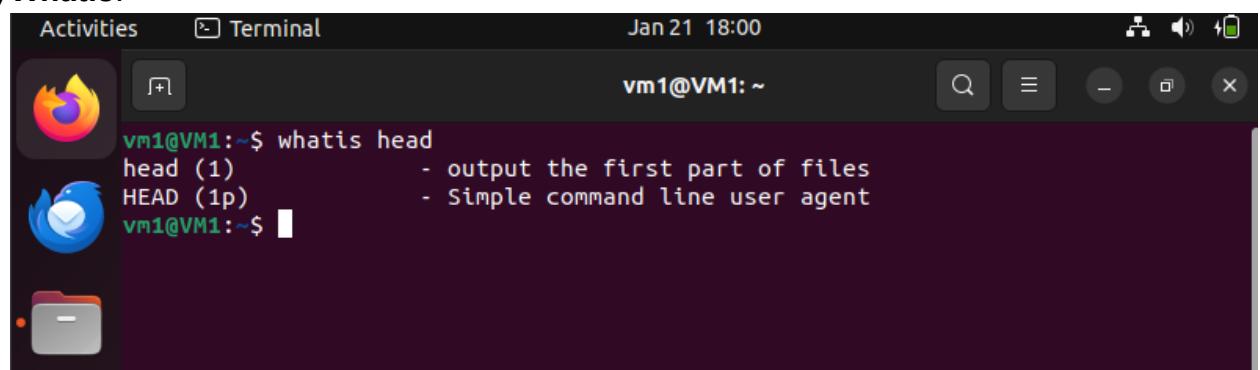
yy) **Whereis**: It is used to get location of executable source code for given command.



A screenshot of an Ubuntu desktop environment. On the left, there is a dock with icons for the Dash, Home, and a folder. The main window is a terminal window titled 'Terminal' with the status bar showing 'Jan 21 17:59' and 'vm1@VM1: ~'. The terminal window contains the following text:

```
vm1@VM1:~$ whereis ls
ls: /usr/bin/ls /usr/share/man/man1/ls.1.gz
vm1@VM1:~$
```

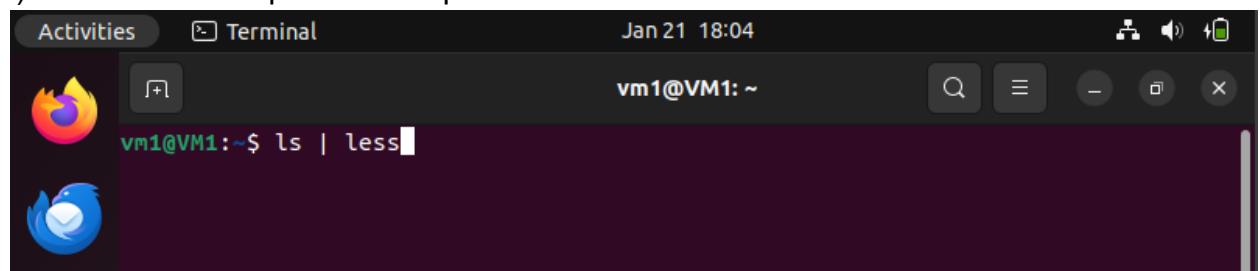
zz) **Whatis**:



A screenshot of an Ubuntu desktop environment. On the left, there is a dock with icons for the Dash, Home, and a folder. The main window is a terminal window titled 'Terminal' with the status bar showing 'Jan 21 18:00' and 'vm1@VM1: ~'. The terminal window contains the following text:

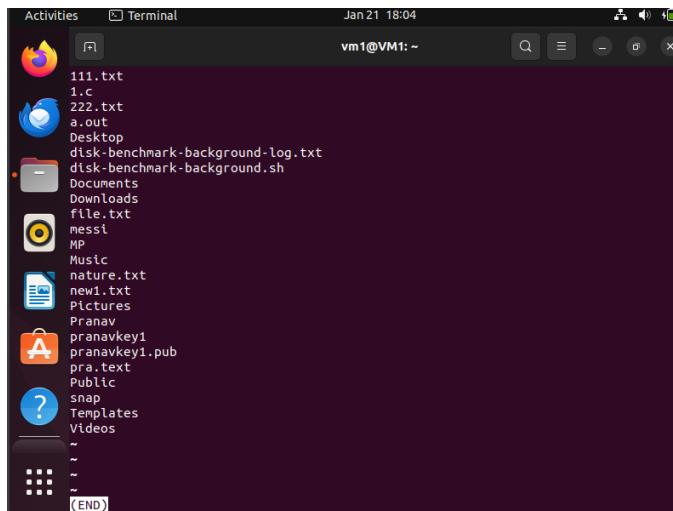
```
vm1@VM1:~$ whatis head
head (1)          - output the first part of files
HEAD (1p)         - Simple command line user agent
vm1@VM1:~$
```

aaa) **Less**: Used to print less outputs on console screen.

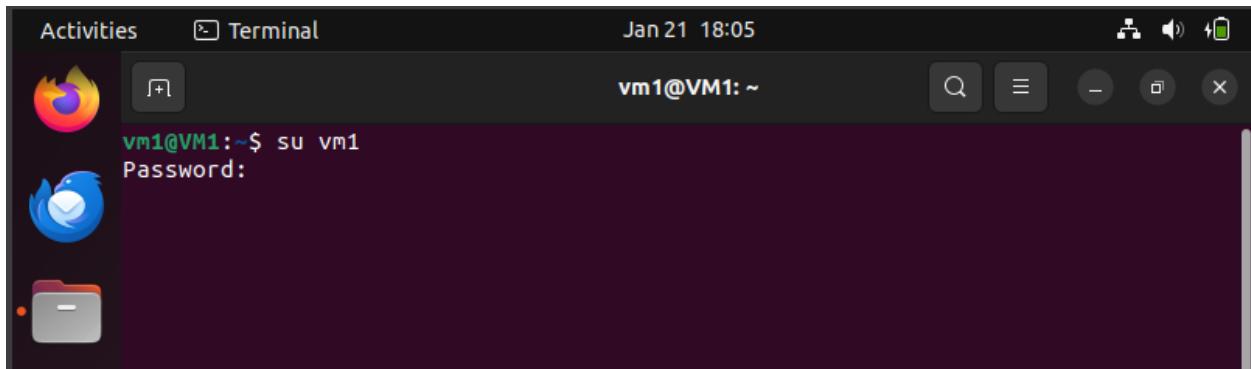


A screenshot of an Ubuntu desktop environment. On the left, there is a dock with icons for the Dash, Home, and a folder. The main window is a terminal window titled 'Terminal' with the status bar showing 'Jan 21 18:04' and 'vm1@VM1: ~'. The terminal window contains the following text:

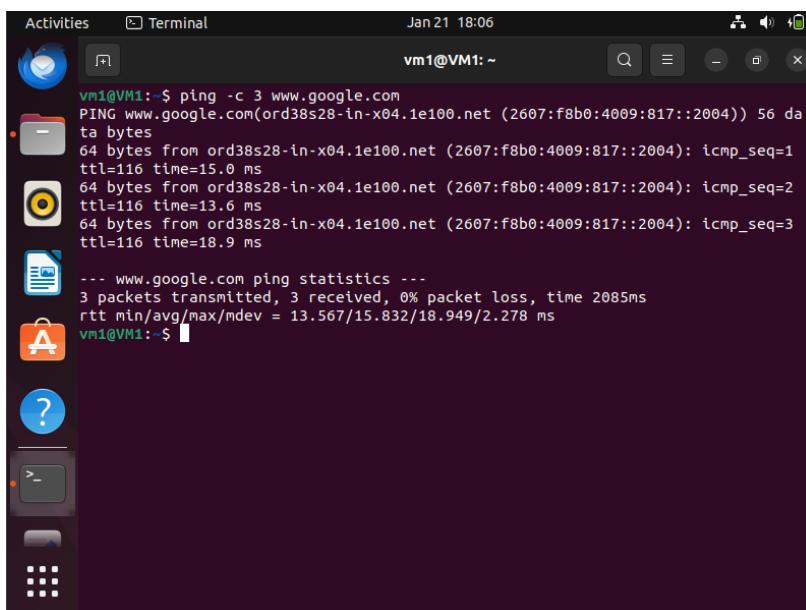
```
vm1@VM1:~$ ls | less
```



bbb) **Su** : It is used to change the user for given session.



ccc) **Ping**: Ping command is used to get response time from a server or DNS which is also called latency.



ddd) **Traceroute**: It is used to see how our packet is travelling on the internet given all the server names from which our packet is going.

```
Activities Terminal Jan 21 18:08
vm1@VM1: ~ traceroute www.google.com
traceroute to www.google.com (142.250.190.36), 30 hops max, 60 byte packets
 1 _gateway (10.0.0.1) 4.377 ms 5.286 ms 6.084 ms
 2 96.120.25.5 (96.120.25.5) 18.180 ms 18.103 ms 18.028 ms
 3 po-317-1282-rur202.area4.il.chicago.comcast.net (96.110.167.205) 16.167 ms
 4 17.913 ms 21.128 ms
 5 po-2-rur201.area4.il.chicago.comcast.net (162.151.44.1) 22.603 ms 22.554 ms
 6 22.924 ms
 7 po-200-xar01.area4.il.chicago.comcast.net (24.153.88.109) 22.859 ms 23.20
 8 4 ms 21.588 ms
 9 * * *
10 be-502-ar01.area4.il.chicago.comcast.net (96.216.150.118) 14.316 ms be-501
  -ar01.area4.il.chicago.comcast.net (96.216.150.114) 13.506 ms 14.495 ms
11 c-50-171-101-230.unallocated.comcastbusiness.net (50.171.101.230) 14.660 ms
12 s 16.358 ms c-50-171-101-226.unallocated.comcastbusiness.net (50.171.101.226)
 16.648 ms
13 * * *
14 216.239.47.128 (216.239.47.128) 18.548 ms 108.170.243.174 (108.170.243.174)
15 21.532 ms 142.251.60.208 (142.251.60.208) 21.011 ms
16 108.170.243.197 (108.170.243.197) 21.409 ms 142.251.60.215 (142.251.60.215)
17 20.364 ms 142.251.60.213 (142.251.60.213) 21.148 ms
18 ord37s33-in-f4.1e100.net (142.250.190.36) 20.276 ms 209.85.241.124 (209.85
  .241.124) 21.413 ms 142.251.233.230 (142.251.233.230) 22.316 ms
vm1@VM1: ~
```

eee) **Date:** It prints date stamp along with zone.

A screenshot of a Linux desktop environment. The top bar shows 'Activities' and 'Terminal' in the title bar, the date 'Jan 21 18:08', and system icons for volume, brightness, and battery. The terminal window has a dark background and shows the command 'date' being run, displaying the current date and time: 'Sun Jan 21 06:08:39 PM CST 2024'. The window title is 'vm1@VM1: ~'. The desktop background is visible behind the terminal window.

fff) **Wget**: It's a non-interactive network downloader.

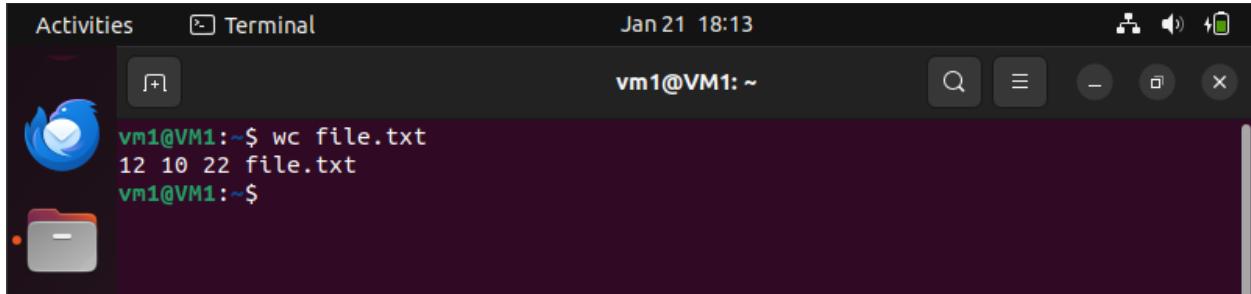
```
Activities Terminal Jan 21 18:12
vm1@VM1: ~
vm1@VM1: $ wget -V
GNU Wget 1.21.2 built on linux-gnu.

-ares +digest -gpgme +https +ipv6 +iri +large-file -metalink +nls
+ntlm +opie +psl +ssl +openssl

Wgetrc:
  /etc/wgetrc (system)
  Locale:
    /usr/share/locale
Compile:
  gcc -DHAVE_CONFIG_H -DSYSTEM_WGETRC="/etc/wgetrc"
  -DLOCATEDIR="/usr/share/locale" -I. -I. -I.. -I../lib
  -I.. -I./lib -Ddate-time -D_FORTIFY_SOURCE=2 -DHAVE_LIBSSL -DNDEBUG
  -g -O2 -ffile-prefix-map=/build/wget-8g5eY0/wget-1.21.2=
  -fflags=auto -ffat-lto-objects -fno-auto -ffat-lto-objects
  -fstack-protector-strong -Wformat -Werror=format-security
  -DNOSLVL2 -D_FILE_OFFSET_BITS=64 -g -Wall
Link:
  gcc -DHAVE_LIBSSL -DNDEBUG -g -O2
  -ffile-prefix-map=/build/wget-8g5eY0/wget-1.21.2= -fflags=auto
  -ffat-lto-objects -fno-auto -ffat-lto-objects
  -fstack-protector-strong -Wformat -Werror=format-security
  -DNOSLVL2 -D_FILE_OFFSET_BITS=64 -g -Wall -Wl,-Bsymbolic-functions
  -fflags=auto -ffat-lto-objects -fno-auto -Wl,-z,relro -Wl,-z,now
  -lpcres2 -luid -lidn2 -lssl -lcrypto -lz -lpsl ftp-opie.o
  openssl http-nntl.o .. /lib/libgnu.a

Copyright (C) 2015 Free Software Foundation, Inc.
```

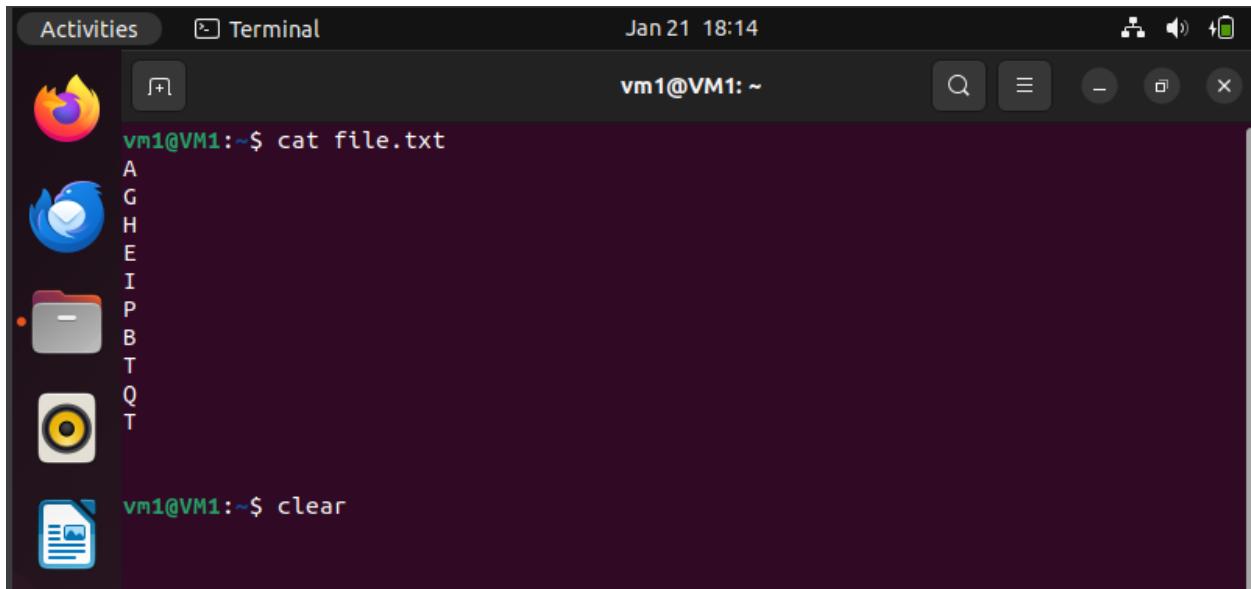
ggg) **Wc**: It is used to get information about users and their instances.



A screenshot of a Linux desktop environment showing a terminal window. The terminal window title is "Terminal" and the status bar shows "Jan 21 18:13" and "vm1@VM1: ~". The terminal content shows the command "wc file.txt" being run, followed by its output: "12 10 22 file.txt". The desktop interface includes a dock with icons for the terminal, file manager, and browser, and a system tray with icons for battery, signal, and volume.

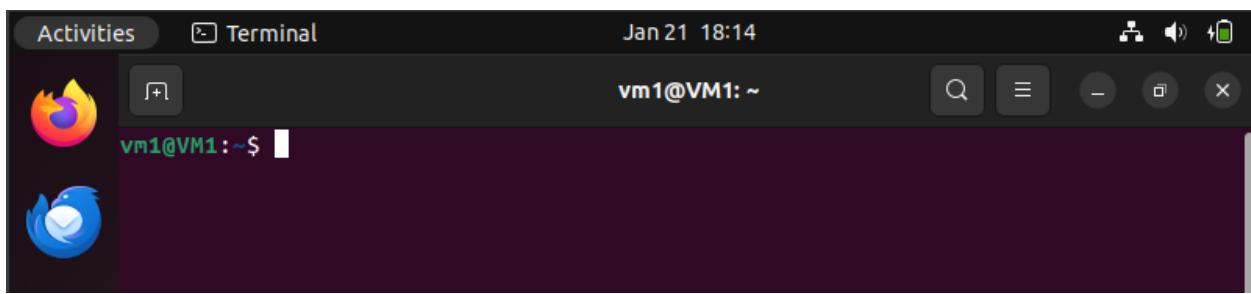
```
Activities Terminal Jan 21 18:13
vm1@VM1: ~
vm1@VM1:~$ wc file.txt
12 10 22 file.txt
vm1@VM1:~$
```

hhh) **Clear**: Command clear is used to clear the content of console.



A screenshot of a Linux desktop environment showing a terminal window. The terminal window title is "Terminal" and the status bar shows "Jan 21 18:14" and "vm1@VM1: ~". The terminal content shows the command "cat file.txt" being run, followed by its output: "A G H E I P B T Q T". Below this, the command "clear" is run. The desktop interface includes a dock with icons for the terminal, file manager, and browser, and a system tray with icons for battery, signal, and volume.

```
Activities Terminal Jan 21 18:14
vm1@VM1: ~
vm1@VM1:~$ cat file.txt
A
G
H
E
I
P
B
T
Q
T
vm1@VM1:~$ clear
```



A screenshot of a Linux desktop environment showing a terminal window. The terminal window title is "Terminal" and the status bar shows "Jan 21 18:14" and "vm1@VM1: ~". The terminal content shows the command "clear" being run. The desktop interface includes a dock with icons for the terminal, file manager, and browser, and a system tray with icons for battery, signal, and volume.

```
Activities Terminal Jan 21 18:14
vm1@VM1: ~
vm1@VM1:~$ clear
```

3. (25 points) Write bash scripts to do the following:

- Write a script called "disk-benchmark-background.sh" that uses the dd command to run a benchmark against the local disk in the background, that captures all the output (both standard out and error output) to a file "disk-benchmark-background-log.txt". Use the "time" command to show how long the benchmark took to complete. The benchmark

should run for at least 10 seconds, and it should complete even if the ssh (or bash) session is terminated.

- b. Write a script called “network-test.sh” that takes input a file “network-test-machine list.txt” with a list of DNS names (e.g. google.com, iit.edu, anl.gov), each name on a separate line, and runs the ping utility collecting 3 samples from each DNS name, and writing the RTT (round trip time) average latency into a file “network-test-latency.txt” where each line will have the DNS name and average RTT separated by a space. Make sure it works with at least 10 DNS names, but it should work for an unspecified number of DNS names.
- c. Write a Python matplotlib script to generate a graph of the “network-test-latency.txt” data. The graph should automatically adjust to the number of entries, and the scale of the data.

## **ALL THE CODING ANSWERS ARE IN REPOSITORY WITH OUTPUT SCREENSHOTS**

### **4. Answer the following questions:**

- 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?**

It will boost your system's performance to some amount if your RAM remains constant, but to improve it further, we must increase it. If you allocate more than one CPU to the system but the BIOS does not have virtualization techniques, it will not assist you in maximising performance. Allocating more processors to the VM may result in a spin lock, in which the host is halted by the multiprocessor guest system. Sharing more CPUs with guests might lead to issues such as reduced host responsiveness and malfunctions. If the user wants his or her host to run optimally, it is best to assign the bare minimum of processors to the guest system.

- 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.**

Given options provide the partial virtualization options which we can provide to guest operating system

1. None: No paravirtualisation
2. Legacy: It is intended for older virtual box applications.

3. Minimal: This is for Mac OS guest users and offers TSC and APIC frequency for guest operating systems.
4. Hyper-V: This applies to Windows guest PCs, which are typically Windows 7 or newer systems. Features include Para virtualized clocks, guest crash reporting, and relaxed timer clocks.
5. KVM: This is helpful for Linux systems because it supports Para virtualized clocks and SMP spin locks.

KVM is good for Linux reasons:

- Smaller and faster.
- Applicable to use by additional visitors. Has the ability to shut down and save the machine state on the hard disk.

**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.**

Points	IDE	SATA	NVMe
Pin	40	9	Around 70
Connections	Two devices	Allows only one connection	Two
Speed	133 mebibytes/second	1.5Gbits/second	3.5GB/sec
Developed by	Western Digital Electronics in association	Serial ATA Working Group	NVM express
Cost	Better value for money	Least expensive	Higher cost

We must employ NVMe when processing large databases in essential applications. SATA is needed for high capacity, low availability, and sequential needs, while IDE is needed for regular processing.

**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.**

Connectivity	NAT	Bridged Adapter	Internal Network	Host Only
VM & Host connectivity	No	Yes	No	Yes
Between two VM's	No	Yes	Yes	Yes
VM to internet	Yes	Yes	No	No
Internet to VM	No (Can be done by port forwarding)	Yes	No	No
Network activities	Mask all network activities	Replicates another node in current network	Can directly communicate to outside network	Network operations with host OS

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

Attributes	USB 1.1	USB 2.0	USB 3.0
Bandwidth	12 Mbps	480 Mbps	4.8 Gbps
Ideal For	Keyboard, Mouse, Printers	Mass storage devices, Video adapters, Data transfer cables	Large mass storage devices, Video adapters
Power required for configured devices	500mA	500mA	900-1000mA
Power required for non-configured devices	100mA	100mA	150mA
Speed	Average speed	High speed	Super high speed
Backward compatible	NA	USB 1.1	USB 2.0/USB 1.1