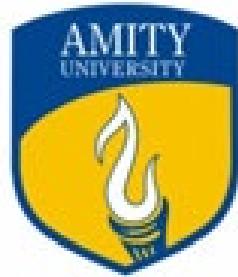


## **Lab File**

### **Linux for Devices [CSE438]**

DEPARTMENT  
OF  
COMPUTER SCIENCE AND ENGINEERING

BACHELOR OF TECHNOLOGY  
IN  
COMPUTER SCIENCE AND ENGINEERING



**Submitted To:**

Dr. Vibha Nehra

**Submitted By:**

Ritik Verma  
A2305218247  
B. Tech (CSE)  
7CSE4Y

AMITY SCHOOL OF ENGINEERING AND TECHNOLOGY  
AMITY UNIVERSITY UTTAR PRADESH  
NOIDA-201301

## **INDEX**

S No.	Name of Practical	Date of Conduct	Maximum Marks	Marks Obtained	Signature of faculty
1.	Installation of Linux Operating System using Virtual Box.	20/07/2021			
2.	Study of Unix/Linux general purpose utility command list obtained from (man, cat, cd, cp, ps, ls, mv, rm, mkdir, rmdir, date, time, chmod, pwd, cal) commands.	27/07/2021			
3.	To perform various operations on vi editor.	03/08/2021			
4.	Study of Bash shell, Bonnie shell and C shell in Unix/Linux operating system	03/08/2021			
5.	Write a shell script program to a calculator, check whether the given number is even or odd and, search whether element is present is in the list or not.	10/08/2021			
6.	Install Docker and run Start, stop, push, pull, log, clocker ps, docker ps -a,create account, docker compose, docker build, docker run	17/08/2021			
7.	Write a shell script program to check whether given file is a directory or not, to count number of files in a Directory, program to copy content of one file to another and, write content on that and Copy to a suitable location in your home directory.	24/08/2021			
8.	Write a shell script program A) to use a pipeline and command substitution to set the length of a line in file to a variable. B) to print duplicated lines of Input using SED command.	31/08/2021			
9.	Write a shell script program to illustrate the implementation of following: Integer Comparison, String comparison, Logical operators, File tests, Conditional control structure, Loop control structures.	07/09/2021			
10.	Write a shell script program to 1. to print a number in reverse order. 2. to reverse the list of strings and reverse each string further in the list. 3. to find the sum of all numbers in a file in Linux 4. to find out the unique words in a file and also count the occurrence of each word	21/09/2021			
11.	Write a shell script program to 1. Write a Case Study (Any Real-Life Problem)	28/09/2021			
12.	Design and develop a "Birthday Reminder" that can automatically send birthday wishes with a personalized message via email.	05/10/2021			
13.	Implement NIC bonding and Teaming.	12/10/2021			
14.	Configuration of NFS Server and Client	19/10/2021			

## Experiment-1

### AIM:

Installing UBUNTU OS using virtual machine.

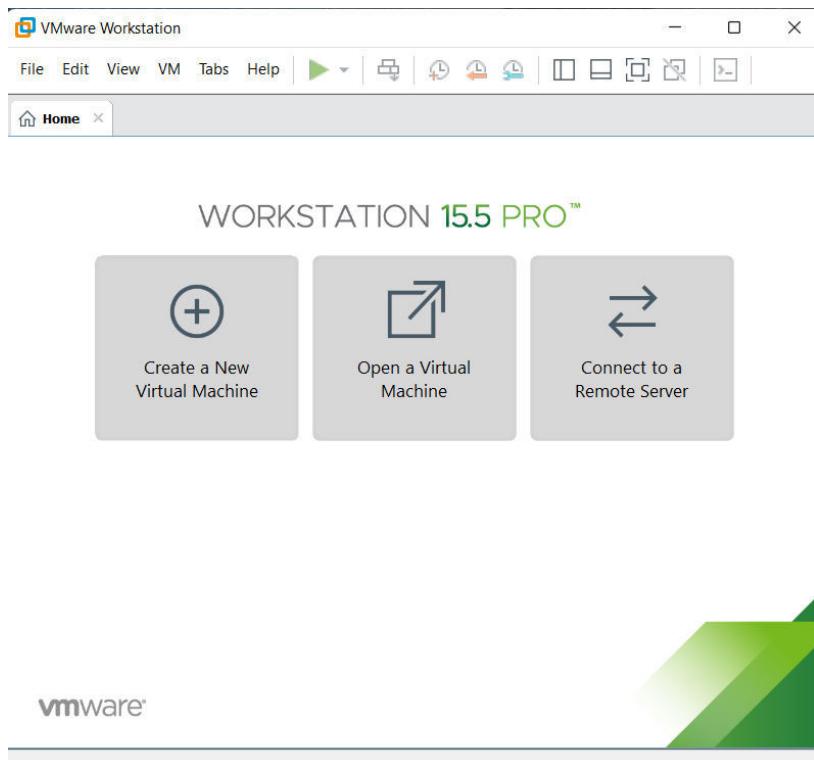
### Theory:

**LINUX** is a kernel distributed under an open-source license. Its functionality list is quite like UNIX. The kernel is a program at the heart of the Linux operating system that takes care of fundamental stuff, like letting hardware communicate with software. Linux is an operating system or a kernel which germinated as an idea in the mind of young and bright Linus Torvalds when he was a computer science student. He used to work on the UNIX OS (proprietary software) and thought that it needed improvements. However, when his suggestions were rejected by the designers of UNIX, he thought of launching an OS which will be receptive to changes, modifications suggested by its users.

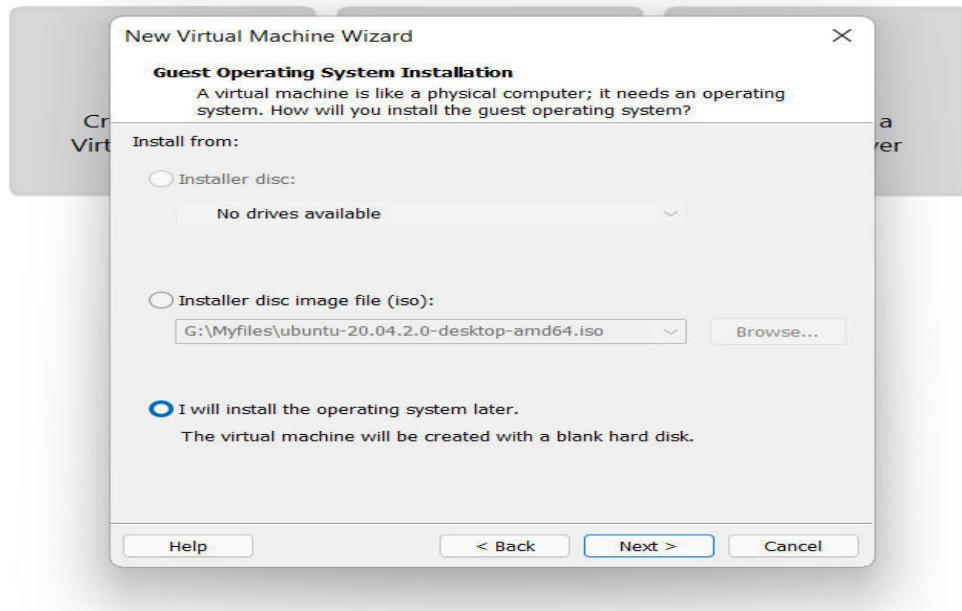
Ubuntu is a complete Linux operating system, freely available with both community and professional support. The Ubuntu community is built on the ideas enshrined in the Ubuntu Manifesto: that software should be available free of charge, that software tools should be usable by people in their local language and despite any disabilities, and that people should have the freedom to customize and alter their software in whatever way they see fit.



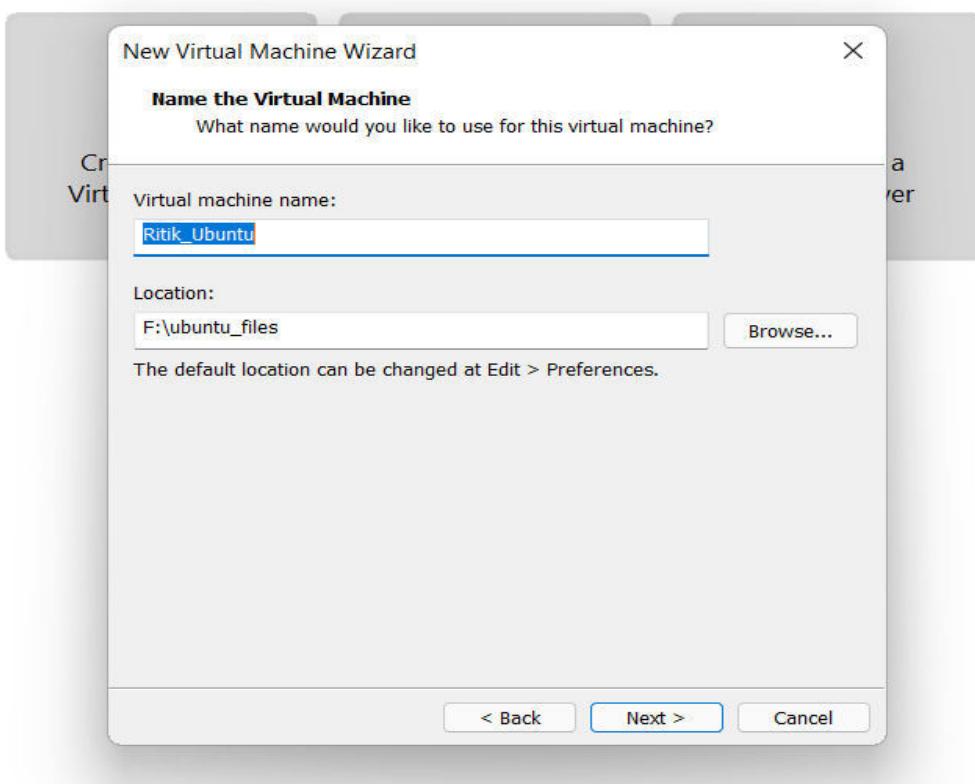
## **PROCEDURE:-**



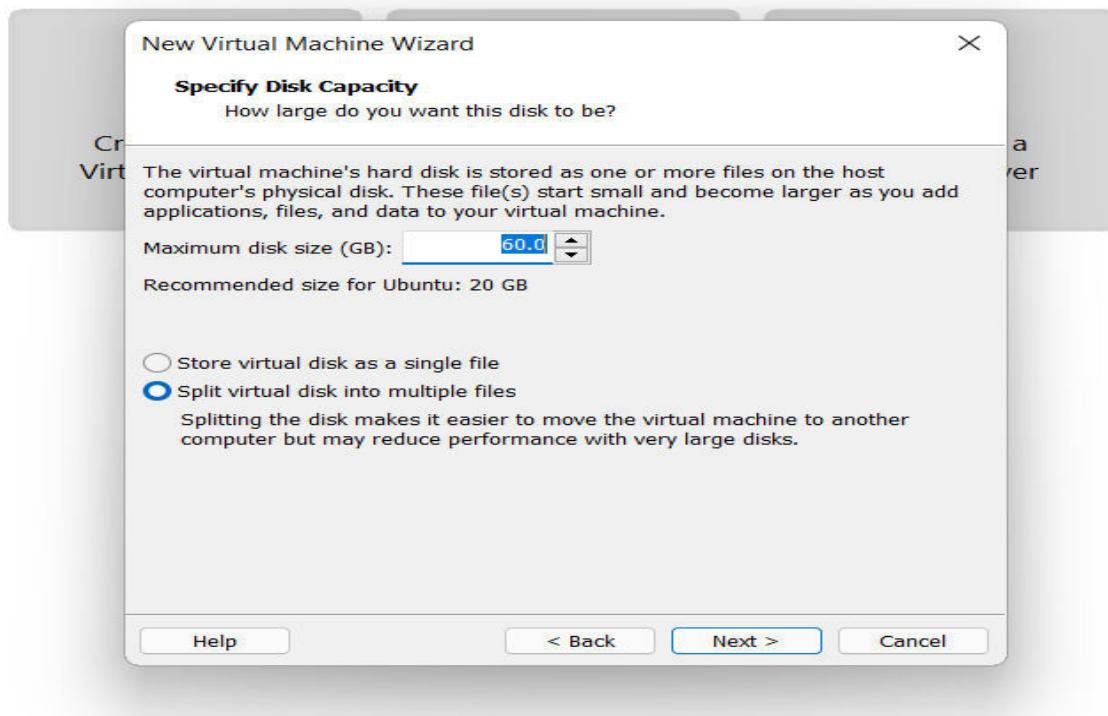
# WORKSTATION 15.5 PRO™



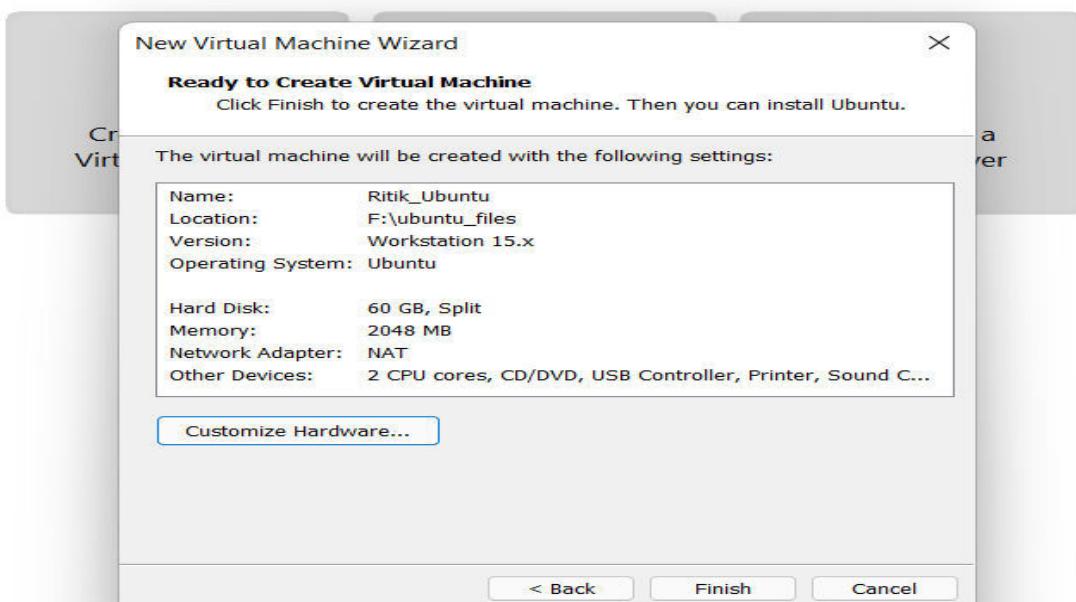
# WORKSTATION 15.5 PRO™

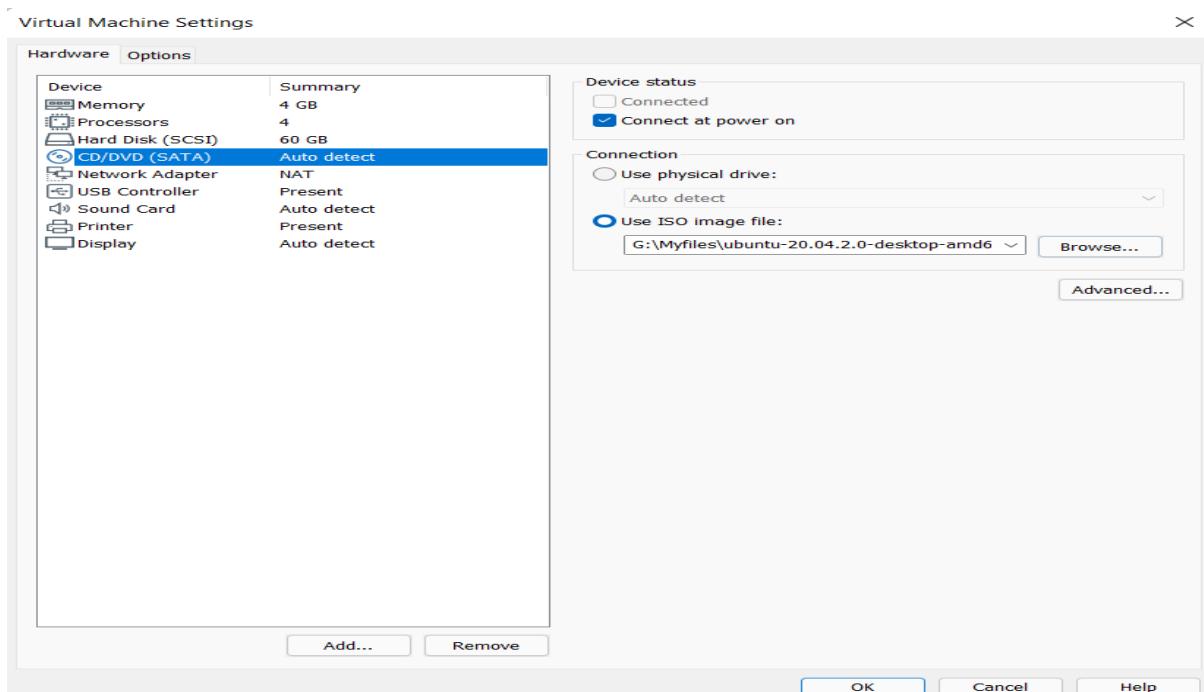
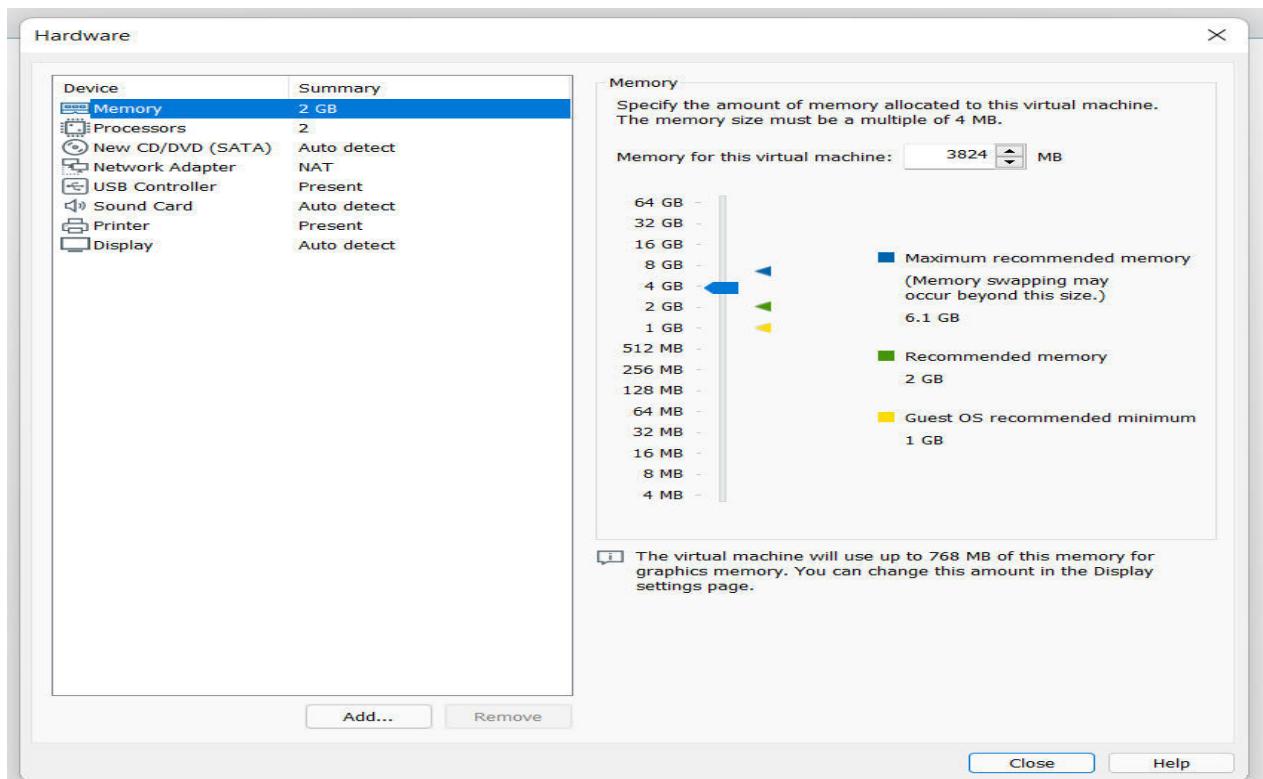


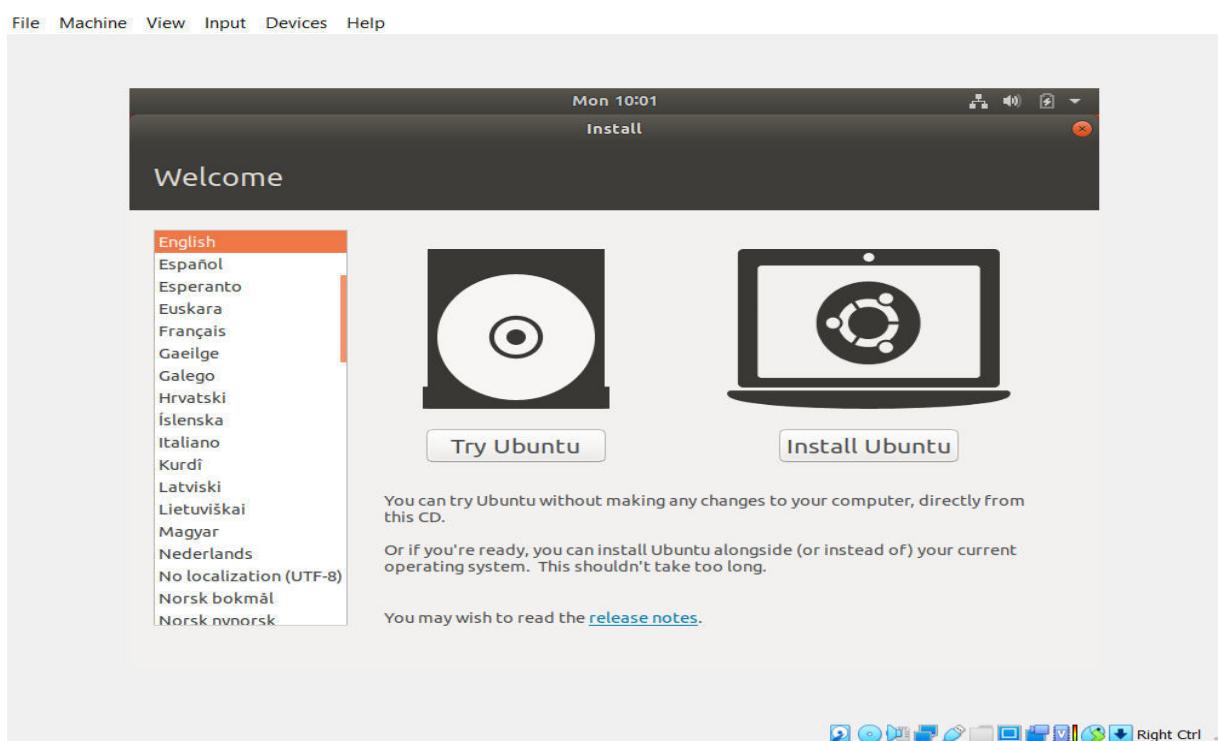
# WORKSTATION 15.5 PRO™

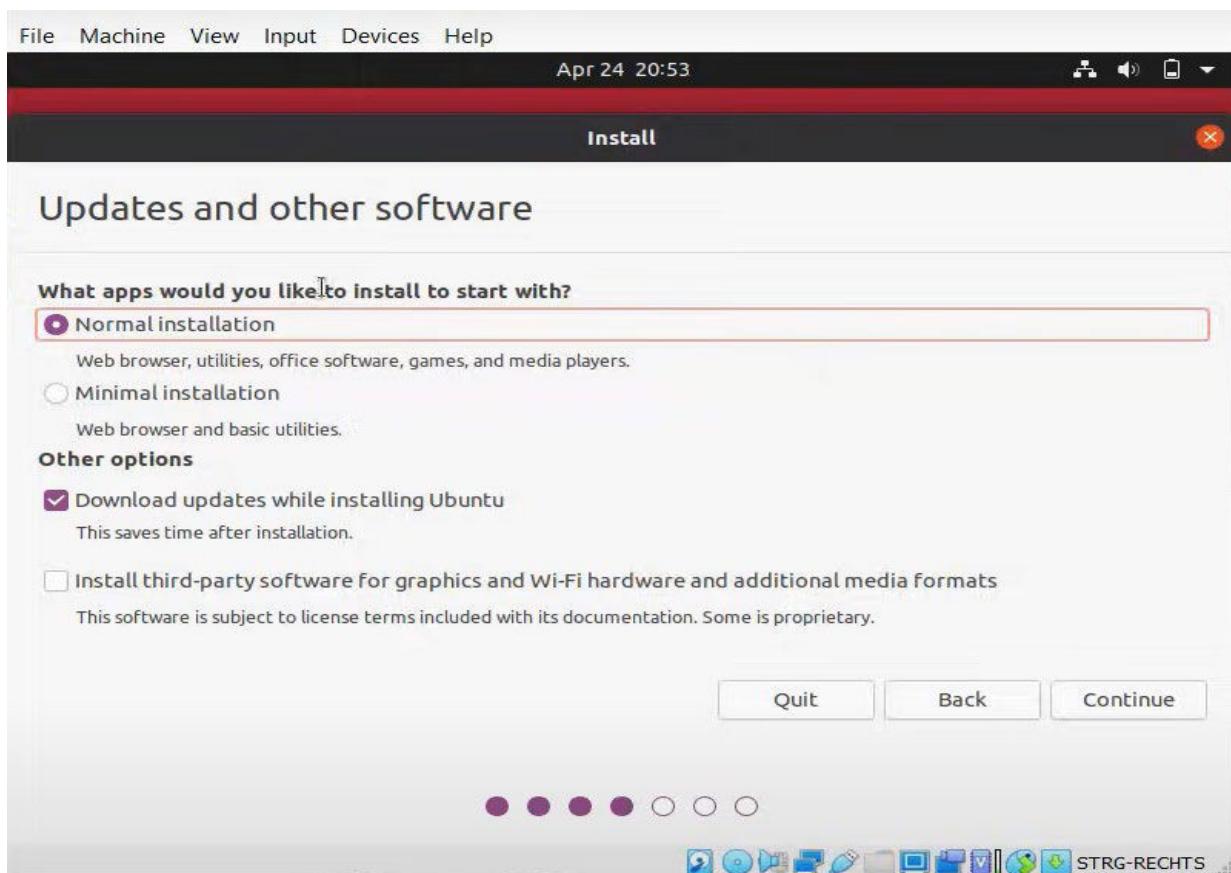
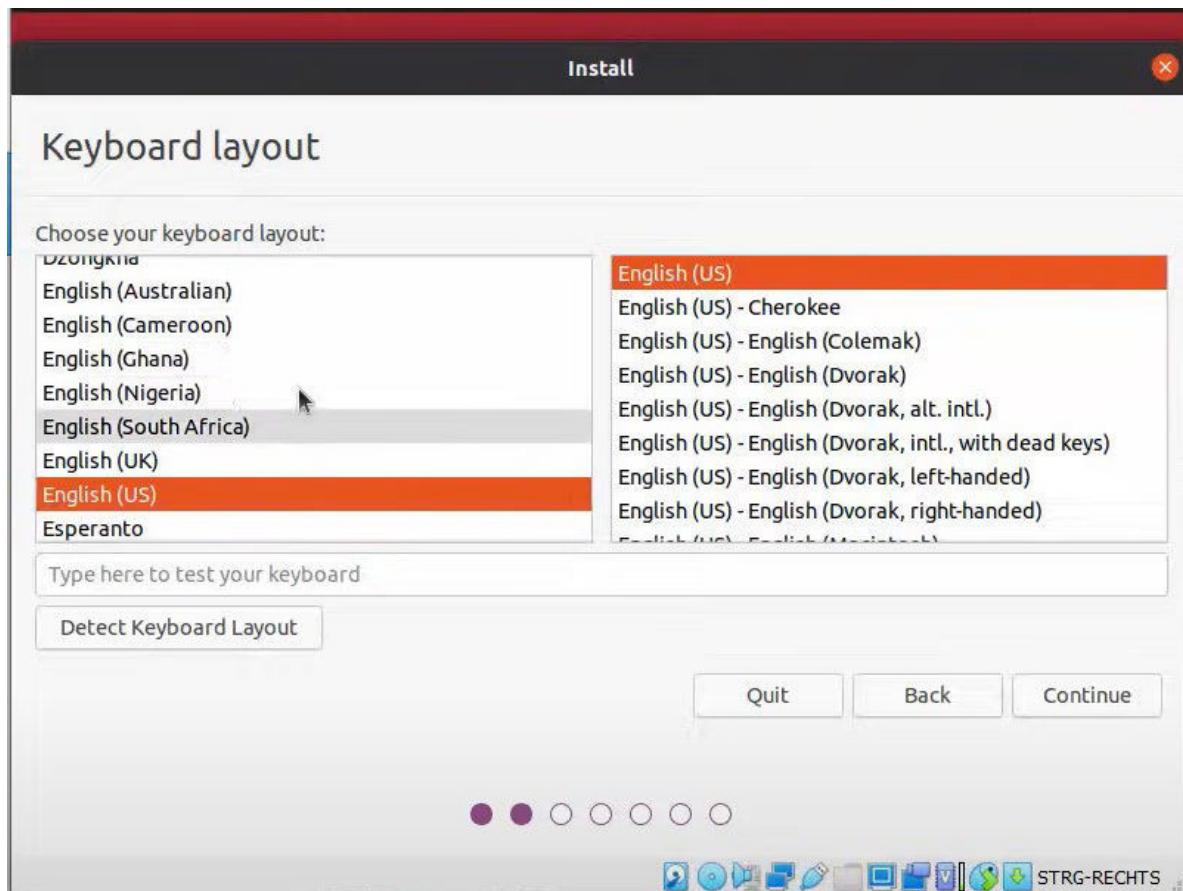


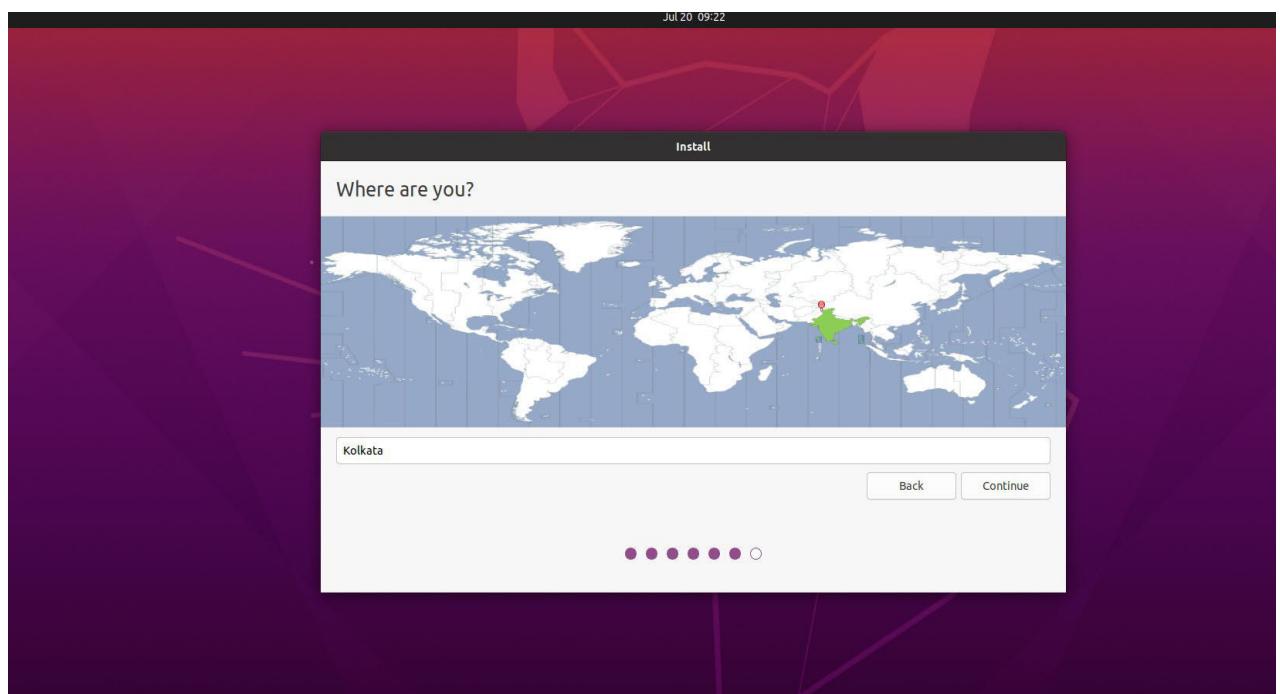
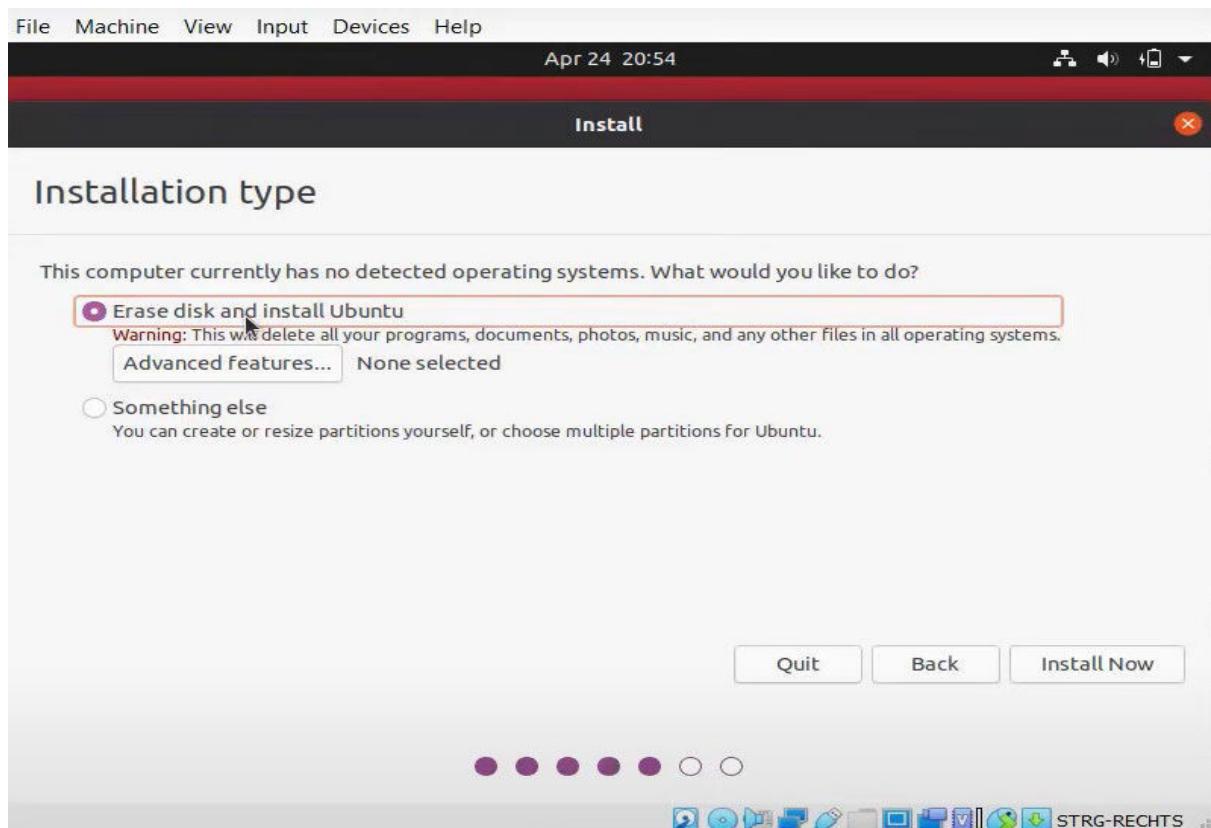
# WORKSTATION 15.5 PRO™

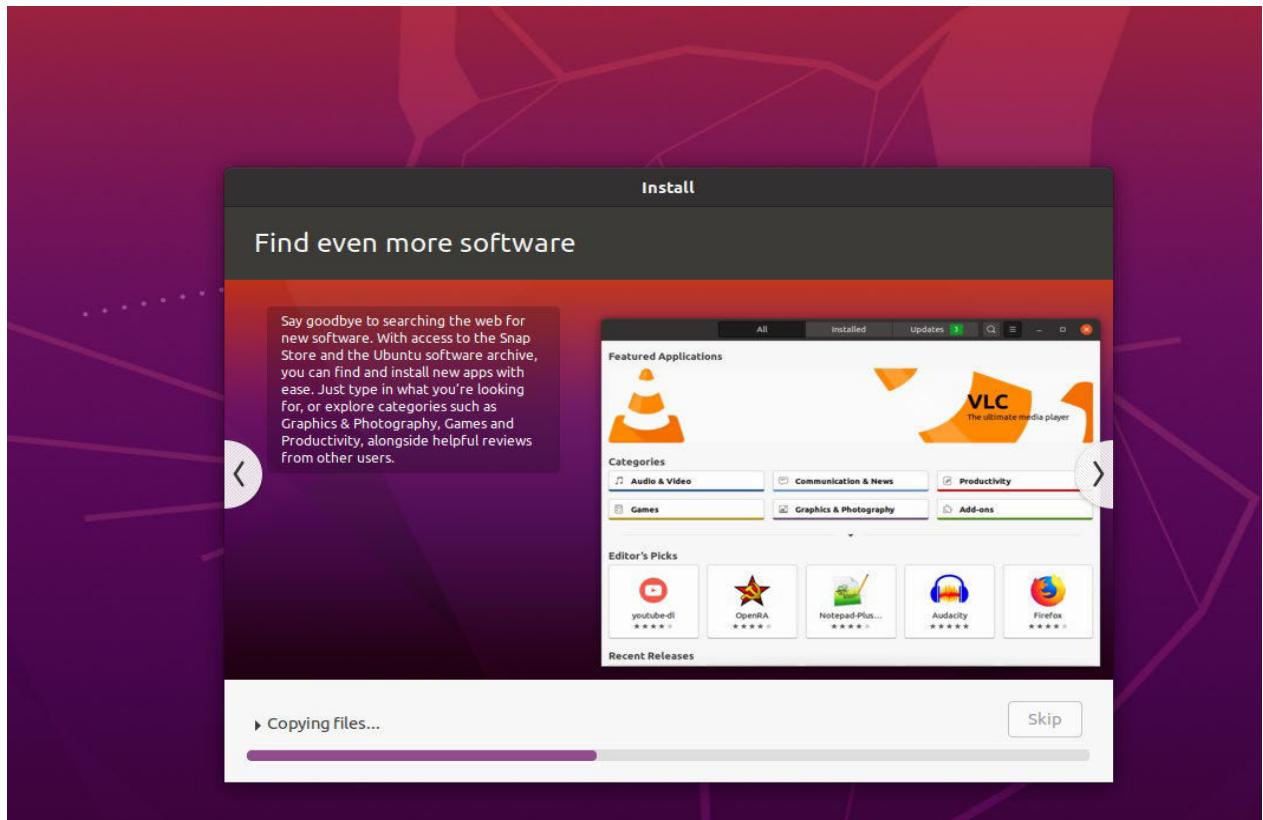






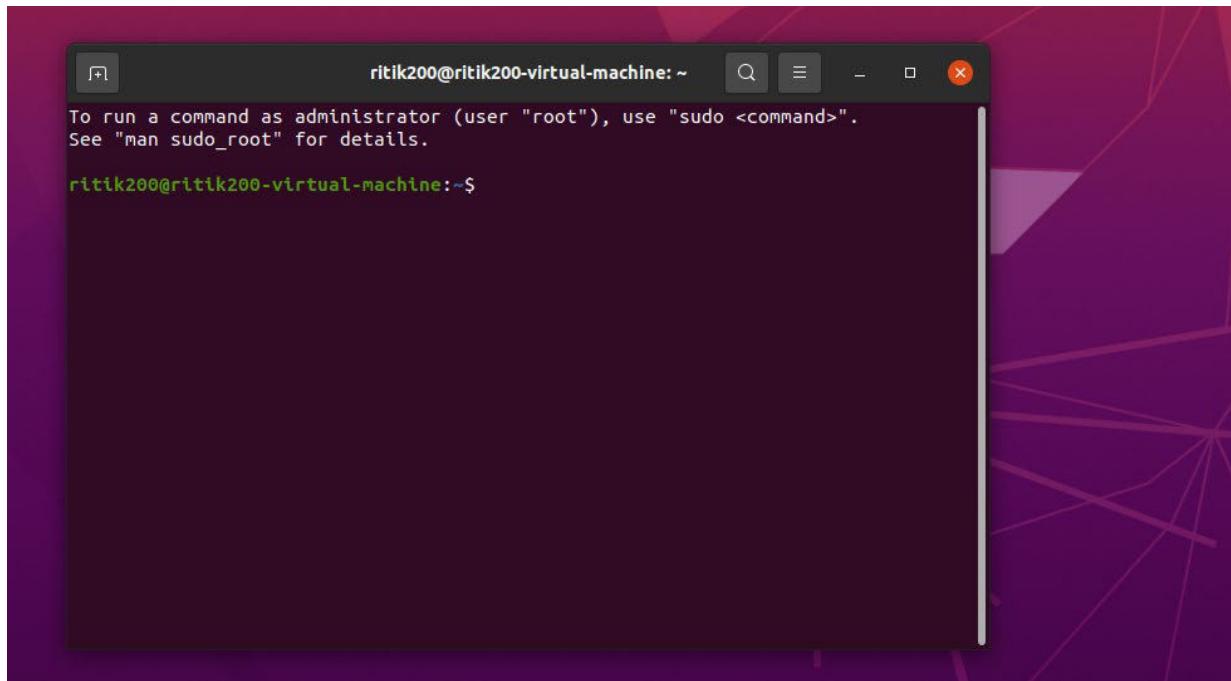






## CONCLUSION-

Virtual machine is installed.



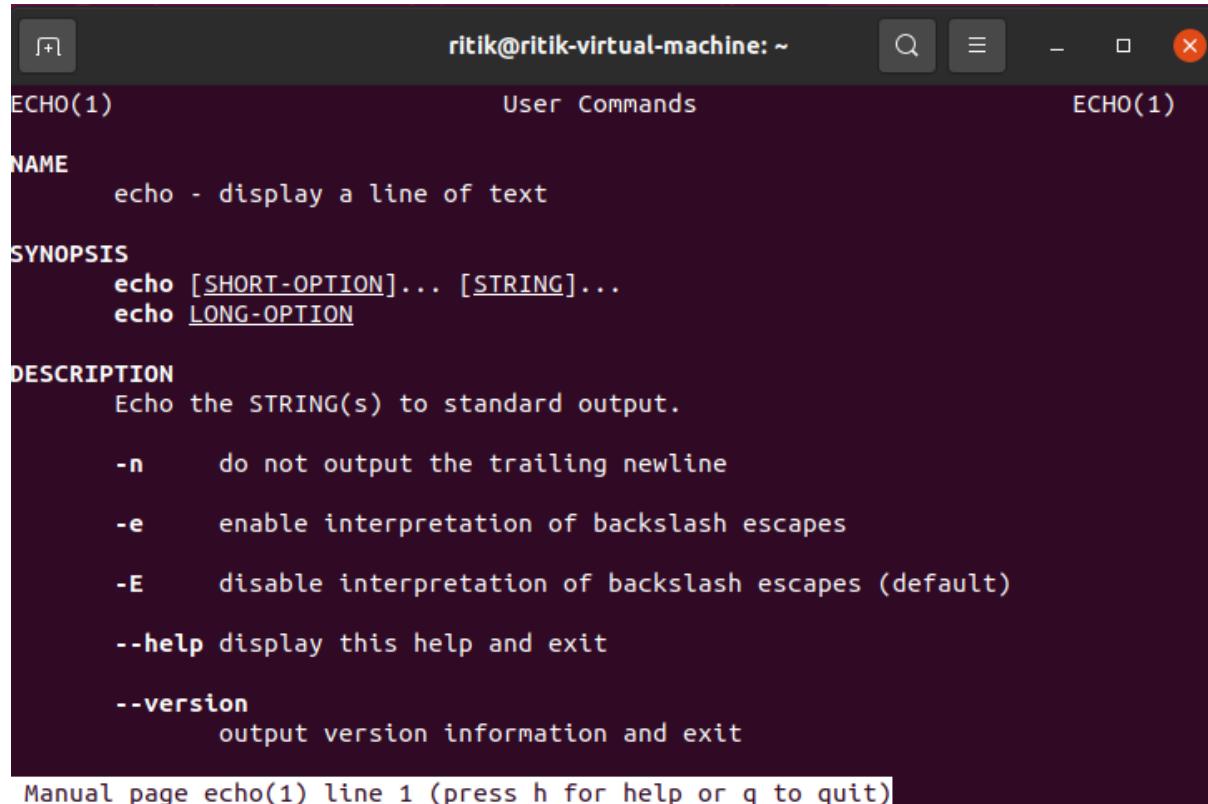
## Experiment-2

**AIM:** - Perform various Linux command.

### **COMMAND 1: - man**

(Whenever you need help with a command type “man” and the command name)

```
ritik@ritik-virtual-machine:~$ man
What manual page do you want?
For example, try 'man man'.
ritik@ritik-virtual-machine:~$ man echo
```



```
ECHO(1) User Commands ECHO(1)

NAME
    echo - display a line of text

SYNOPSIS
    echo [SHORT-OPTION]... [STRING]...
    echo LONG-OPTION

DESCRIPTION
    Echo the STRING(s) to standard output.

    -n      do not output the trailing newline
    -e      enable interpretation of backslash escapes
    -E      disable interpretation of backslash escapes (default)
    --help  display this help and exit
    --version
            output version information and exit

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

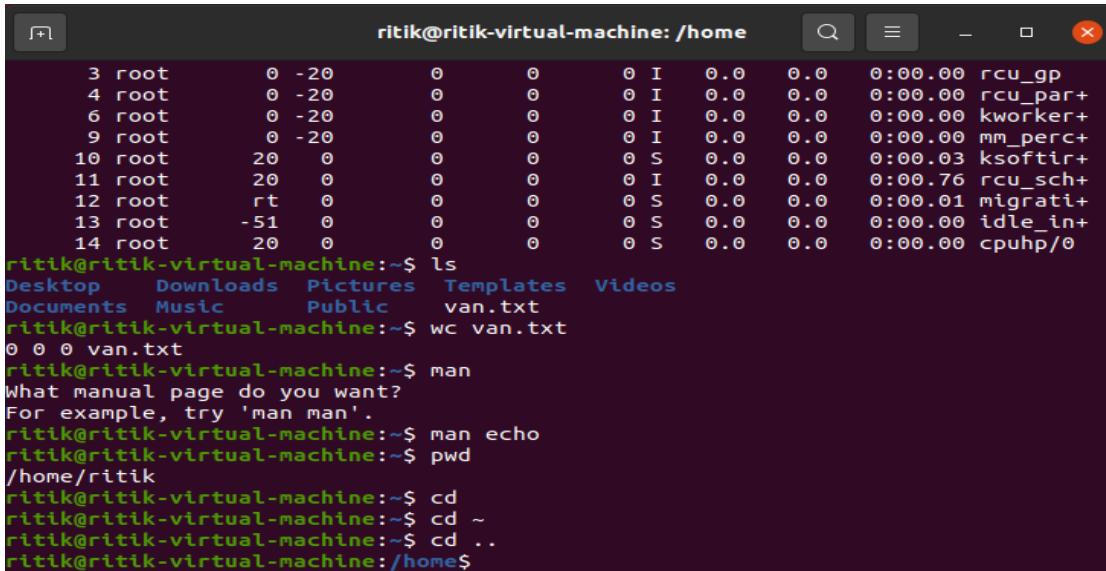
## COMMAND 2 : - pwd

(To find your current path use “pwd”)

```
ritik@ritik-virtual-machine:~$ top
10481 ritik    20   0  20624  3940  3184 R  0.3  0.1  0:00.25 top
  1 root    20   0 169000 13252  8588 S  0.0  0.3  0:03.74 systemd
  2 root    20   0      0      0      0 S  0.0  0.0  0:00.03 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+
  6 root    0 -20      0      0      0 I  0.0  0.0  0:00.00 kworker+
  9 root    0 -20      0      0      0 I  0.0  0.0  0:00.00 mm_perc+
 10 root    20   0      0      0      0 S  0.0  0.0  0:00.03 ksoftir+
 11 root    20   0      0      0      0 I  0.0  0.0  0:00.76 rcu_sch+
 12 root    rt  0      0      0      0 S  0.0  0.0  0:00.01 migrati+
 13 root    -51  0      0      0      0 S  0.0  0.0  0:00.00 idle_in+
 14 root    20   0      0      0      0 S  0.0  0.0  0:00.00 cpuhp/0
ritik@ritik-virtual-machine:~$ ls
Desktop  Downloads  Pictures  Templates  Videos
Documents  Music  Public  van.txt
ritik@ritik-virtual-machine:~$ wc van.txt
0 0 0 van.txt
ritik@ritik-virtual-machine:~$ man
What manual page do you want?
For example, try 'man man'.
ritik@ritik-virtual-machine:~$ man echo
ritik@ritik-virtual-machine:~$ pwd
/home/ritik
ritik@ritik-virtual-machine:~$
```

## COMMAND 3 :- cd

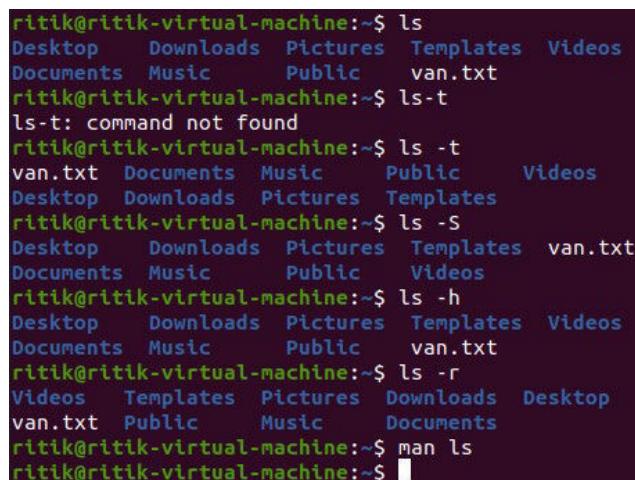
- a) To change to a specific directory use “cd”
- b) “~” is the location of your home directory
- c) “..” is the location of the directory below current one



```
ritik@ritik-virtual-machine: /home
 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+
 6 root      0 -20      0      0      0 I    0.0    0.0    0:00.00 kworker+
 9 root      0 -20      0      0      0 I    0.0    0.0    0:00.00 mm_perc+
10 root     20  0      0      0      0 S    0.0    0.0    0:00.03 ksoftir+
11 root     20  0      0      0      0 I    0.0    0.0    0:00.76 rcu_sch+
12 root     rt  0      0      0      0 S    0.0    0.0    0:00.01 migrati+
13 root    -51  0      0      0      0 S    0.0    0.0    0:00.00 idle_in+
14 root     20  0      0      0      0 S    0.0    0.0    0:00.00 cpuhp/0
ritik@ritik-virtual-machine:~$ ls
Desktop  Downloads  Pictures  Templates  Videos
Documents  Music  Public  van.txt
ritik@ritik-virtual-machine:~$ wc van.txt
0 0 0 van.txt
ritik@ritik-virtual-machine:~$ man
What manual page do you want?
For example, try 'man man'.
ritik@ritik-virtual-machine:~$ man echo
ritik@ritik-virtual-machine:~$ pwd
/home/ritik
ritik@ritik-virtual-machine:~$ cd
ritik@ritik-virtual-machine:~$ cd ~
ritik@ritik-virtual-machine:~$ cd ..
ritik@ritik-virtual-machine:/home$
```

## COMMAND 4 :- ls

- a) To list the files in the current directory use “ls”
- b) -l long list (displays lots of info)
- c) -t sort by modification time
- d) -S sort by size
- e) -h list file sizes in human readable format
- f) -r reverse the order
- g) man ls for more option



```
ritik@ritik-virtual-machine:~$ ls
Desktop  Downloads  Pictures  Templates  Videos
Documents  Music  Public  van.txt
ritik@ritik-virtual-machine:~$ ls -t
ls: command not found
ritik@ritik-virtual-machine:~$ ls -t
van.txt  Documents  Music  Public  Videos
Desktop  Downloads  Pictures  Templates
ritik@ritik-virtual-machine:~$ ls -S
Desktop  Downloads  Pictures  Templates  van.txt
Documents  Music  Public  Videos
ritik@ritik-virtual-machine:~$ ls -h
Desktop  Downloads  Pictures  Templates  Videos
Documents  Music  Public  van.txt
ritik@ritik-virtual-machine:~$ ls -r
Videos  Templates  Pictures  Downloads  Desktop
van.txt  Public  Music  Documents
ritik@ritik-virtual-machine:~$ man ls
ritik@ritik-virtual-machine:~$
```

## **COMMAND 5 : - ls -lt**

(List files by time in reverse order with long listing)

```
ritik@ritik-virtual-machine:~$ ls -lt
total 32
-rw-rw-r-- 1 ritik ritik    0 Jul 27 13:57 van.txt
drwxr-xr-x 2 ritik ritik 4096 Jul 19 13:31 Desktop
drwxr-xr-x 2 ritik ritik 4096 Jul 19 13:31 Documents
drwxr-xr-x 2 ritik ritik 4096 Jul 19 13:31 Downloads
drwxr-xr-x 2 ritik ritik 4096 Jul 19 13:31 Music
drwxr-xr-x 2 ritik ritik 4096 Jul 19 13:31 Pictures
drwxr-xr-x 2 ritik ritik 4096 Jul 19 13:31 Public
drwxr-xr-x 2 ritik ritik 4096 Jul 19 13:31 Templates
drwxr-xr-x 2 ritik ritik 4096 Jul 19 13:31 Videos
ritik@ritik-virtual-machine:~$
```

## **COMMAND 6 : - General Syntax: \***

("\*" can be used as a wildcard in unix/linux)

```
ritik@ritik-virtual-machine:~$ ls *.txt
van.txt
ritik@ritik-virtual-machine:~$
```

## **COMMAND 7 : - mkdir**

(To create a new directory use “mkdir”)

```
ritik@ritik-virtual-machine:~$ mkdir helloworld
ritik@ritik-virtual-machine:~$ ls
Desktop  Downloads  Music      Public      van.txt
Documents  helloworld  Pictures  Templates  Videos
ritik@ritik-virtual-machine:~$
```

## **COMMAND 8 : - rmdir**

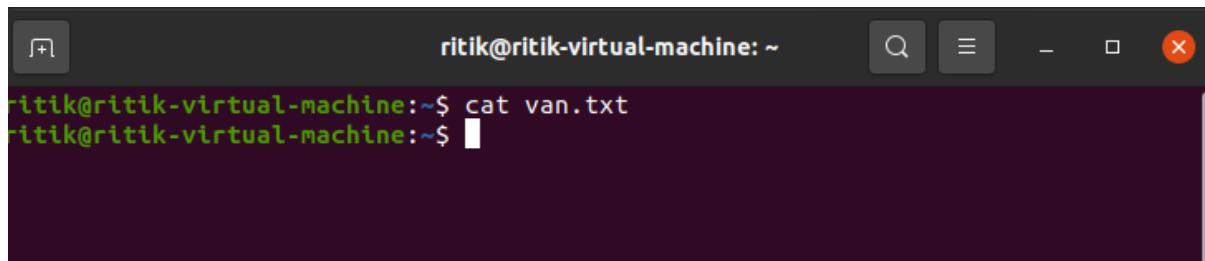
(To remove and empty directory use “rmdir”)

```
ritik@ritik-virtual-machine:~$ rmdir helloworld
ritik@ritik-virtual-machine:~$ ls
Desktop  Downloads  Pictures  Templates  Videos
Documents  Music      Public      van.txt
ritik@ritik-virtual-machine:~$
```

## **COMMAND 9: -**

### **a) Cat**

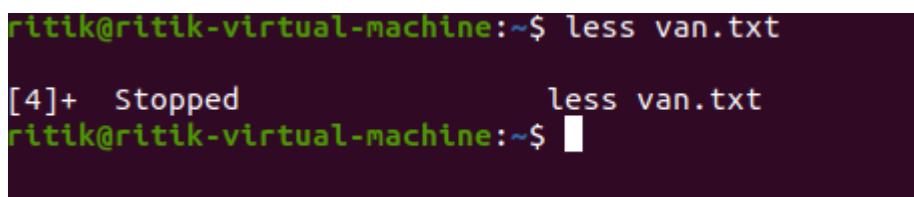
(Dumps an entire file to standard output Good for displaying short, simple files)



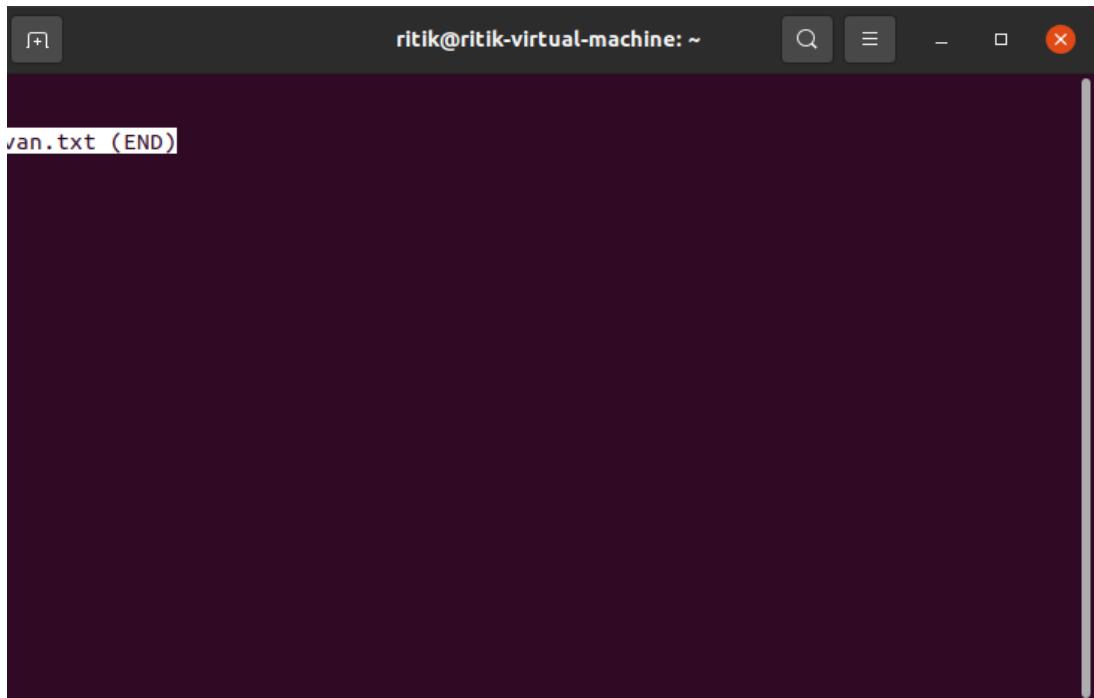
A screenshot of a terminal window titled "ritik@ritik-virtual-machine: ~". The window has a dark background and light-colored text. At the top, there are icons for search, minimize, maximize, and close. In the main area, the command "cat van.txt" is typed in green, followed by a dollar sign. Below the command, there is a large black rectangular redaction box covering the output of the command.

### **b) less**

( “less” displays a file, allowing forward/backward movement within it , – return scrolls forward one line, space one page , – y scrolls back one line, b one page , use “/” to search for a string, Press q to quit)



A screenshot of a terminal window titled "ritik@ritik-virtual-machine: ~\$". The window has a dark background and light-colored text. At the top, there are icons for search, minimize, maximize, and close. In the main area, the command "less van.txt" is typed in green, followed by a dollar sign. Below the command, the text "[4]+ Stopped less van.txt" is displayed in white, indicating the process has stopped. There is also a large black rectangular redaction box covering the output of the command.



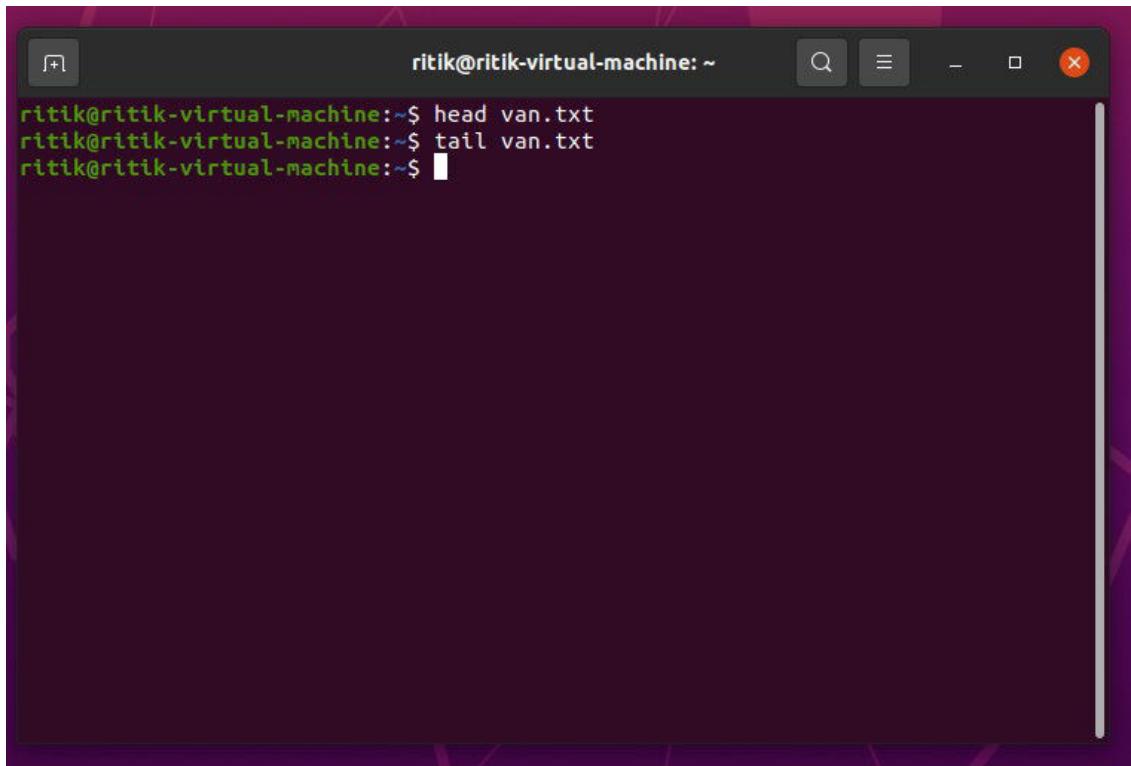
A screenshot of a terminal window titled "ritik@ritik-virtual-machine: ~". The window contains the command "cat van.txt" followed by the text "van.txt (END)". The terminal has a dark background and a light-colored text area.

**c) head**

(“head” displays the top part of a file ,By default it shows the first 10 lines , -n option allows you to change that , “head -n50 file.txt” displays the first 50 lines of file.txt)

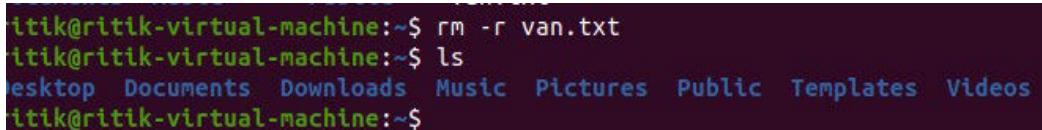
**d) tail**

(Same as head, but shows the last lines)



A screenshot of a terminal window titled "ritik@ritik-virtual-machine: ~". The window contains three lines of text:  
ritik@ritik-virtual-machine:~\$ head van.txt  
ritik@ritik-virtual-machine:~\$ tail van.txt  
ritik@ritik-virtual-machine:~\$

**COMMAND 10: - rm** ( To remove a file use “rm”, To remove a file recursively”: rm –r)



```
ritik@ritik-virtual-machine:~$ rm -r van.txt
ritik@ritik-virtual-machine:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
ritik@ritik-virtual-machine:~$
```

## **COMMAND 11: - ls -l filename**

(to find the permission level of that file)

### **Permission level : -**

“r” means “read only” permission

“w” means “write” permission

“x” means “execute” permission

- In case of directory, “x” grants permission to list directory contents

```
ritik@ritik-virtual-machine:~$ ls -l
total 32
drwxr-xr-x 2 ritik ritik 4096 Jul 19 13:31 Desktop
drwxr-xr-x 2 ritik ritik 4096 Jul 19 13:31 Documents
drwxr-xr-x 2 ritik ritik 4096 Jul 19 13:31 Downloads
drwxr-xr-x 2 ritik ritik 4096 Jul 19 13:31 Music
drwxr-xr-x 2 ritik ritik 4096 Jul 19 13:31 Pictures
drwxr-xr-x 2 ritik ritik 4096 Jul 19 13:31 Public
drwxr-xr-x 2 ritik ritik 4096 Jul 19 13:31 Templates
drwxr-xr-x 2 ritik ritik 4096 Jul 19 13:31 Videos
ritik@ritik-virtual-machine:~$ █
```

## COMMAND 12: - chmod

(If you own the file, you can change it's permissions with “chmod”)

- 7, rwx: read, write, and execute
- 6, rw-: read and write
- 5, r-x: read and execute
- 4, r--: read-only
- 3, -wx: write and execute
- 2, -w-: write only
- 1, --x: execute only
- 0, ---: none

```
ritik@ritik-virtual-machine:~$ ls -l
total 36
drwxr-xr-x 2 ritik ritik 4096 Jul 19 13:31 Desktop
drwxr-xr-x 2 ritik ritik 4096 Jul 19 13:31 Documents
drwxr-xr-x 2 ritik ritik 4096 Jul 19 13:31 Downloads
drwxrwxr-x 2 ritik ritik 4096 Jul 27 14:57 hello
drwxr-xr-x 2 ritik ritik 4096 Jul 19 13:31 Music
drwxr-xr-x 2 ritik ritik 4096 Jul 19 13:31 Pictures
drwxr-xr-x 2 ritik ritik 4096 Jul 19 13:31 Public
drwxr-xr-x 2 ritik ritik 4096 Jul 19 13:31 Templates
drwxr-xr-x 2 ritik ritik 4096 Jul 19 13:31 Videos
ritik@ritik-virtual-machine:~$ chmod 777 hello
ritik@ritik-virtual-machine:~$ ls -l
total 36
drwxr-xr-x 2 ritik ritik 4096 Jul 19 13:31 Desktop
drwxr-xr-x 2 ritik ritik 4096 Jul 19 13:31 Documents
drwxr-xr-x 2 ritik ritik 4096 Jul 19 13:31 Downloads
drwxrwxrwx 2 ritik ritik 4096 Jul 27 14:57 hello
drwxr-xr-x 2 ritik ritik 4096 Jul 19 13:31 Music
drwxr-xr-x 2 ritik ritik 4096 Jul 19 13:31 Pictures
drwxr-xr-x 2 ritik ritik 4096 Jul 19 13:31 Public
drwxr-xr-x 2 ritik ritik 4096 Jul 19 13:31 Templates
drwxr-xr-x 2 ritik ritik 4096 Jul 19 13:31 Videos
ritik@ritik-virtual-machine:~$
```

## COMMAND 13: - ps

(To view the processes that are running)

```
ritik@ritik-virtual-machine:~$ ps
  PID TTY      TIME CMD
 11844 pts/0    00:00:00 bash
 11855 pts/0    00:00:00 ps
ritik@ritik-virtual-machine:~$
```

### a) ps -u root

(To view the processes that you're running)

```
ritik@ritik-virtual-machine:~$ ps -u root
 PID TTY      TIME CMD
  1 ?        00:00:06 systemd
  2 ?        00:00:00 kthreadd
  3 ?        00:00:00 rcu_gp
  4 ?        00:00:00 rcu_par_gp
  6 ?        00:00:00 kworker/0:0H-kblockd
  9 ?        00:00:00 mm_percpu_wq
 10 ?       00:00:00 ksoftirqd/0
 11 ?       00:00:01 rcu_sched
 12 ?       00:00:00 migration/0
 13 ?       00:00:00 idle_inject/0
 14 ?       00:00:00 cpuhp/0
 15 ?       00:00:00 cpuhp/1
 16 ?       00:00:00 idle_inject/1
 17 ?       00:00:00 migration/1
 18 ?       00:00:01 ksoftirqd/1
 20 ?       00:00:00 kworker/1:0H-kblockd
 21 ?       00:00:00 cpuhp/2
 22 ?       00:00:00 idle_inject/2
```

### COMMAND 14: - top

(To view the CPU usage of all processes)

```
top - 20:19:00 up 1:39, 1 user, load average: 0.06, 0.02, 0.00
Tasks: 214 total,  3 running, 180 sleeping,   0 stopped,   0 zombie
%CPU(s):  9.7 us,  0.7 sy,  0.0 nl, 89.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
Mem: 4037260 total, 2096980 free, 1083684 used, 856596 buff/cache
Swap: 1014488 total, 1014488 free,          0 used. 2690360 avail Mem

 PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
1443 ritik     20   0 3037704 329652 115456 S  0.3   8.2  0:55.04 gnome-shell
1259 ritik     20   0 447036 108328 59384 R  4.3   2.7  0:14.28 Xorg
1944 ritik     20   0  808100 40300 28116 S  2.3   1.0  0:08.12 gnome-terminal-
 1 root      20   0 159932  9192 6724 S  0.0   0.2  0:02.91 systemd
 2 root      20   0      0  0      0      0 S  0.0   0.0  0:00.01 kthreadd
 3 root      0 -20  0      0  0      0 I  0.0   0.0  0:00.00 rcu_gp
 4 root      0 -20  0      0  0      0 I  0.0   0.0  0:00.00 rcu_par_gp
 5 root      20   0      0  0      0  0      0 I  0.0   0.0  0:00.00 kworker/0:0-eve
 6 root      0 -20  0      0  0      0 I  0.0   0.0  0:00.00 kworker/0:0H-kb
 8 root      0 -20  0      0  0      0 I  0.0   0.0  0:00.00 mm_percpu_wq
 9 root      20   0      0  0      0  0      0 S  0.0   0.0  0:00.15 ksoftirqd/0
10 root     20   0      0  0      0  0      0 R  0.0   0.0  0:00.82 rcu_sched
11 root     rt  0      0  0      0  0      0 S  0.0   0.0  0:00.01 migration/0
13 root    -51   0      0  0      0  0      0 S  0.0   0.0  0:00.00 idle_inject/0
14 root     20   0      0  0      0  0      0 S  0.0   0.0  0:00.00 cpuhp/0
15 root     20   0      0  0      0  0      0 S  0.0   0.0  0:00.00 kdevtmpfs
16 root    -20  0      0  0      0  0      0 I  0.0   0.0  0:00.00 netns
17 root     20   0      0  0      0  0      0 S  0.0   0.0  0:00.00 rcu_tasks_kthre
18 root     20   0      0  0      0  0      0 S  0.0   0.0  0:00.00 kauditd
19 root     20   0      0  0      0  0      0 S  0.0   0.0  0:00.00 khungtaskd
20 root     20   0      0  0      0  0      0 S  0.0   0.0  0:00.00 oom_reaper
21 root    -20  0      0  0      0  0      0 I  0.0   0.0  0:00.00 writeback
22 root     20   0      0  0      0  0      0 S  0.0   0.0  0:00.00 kcompactd0
23 root     25   5      0  0      0  0      0 S  0.0   0.0  0:00.00 ksmd
24 root     39   19     0  0      0  0      0 S  0.0   0.0  0:00.00 khugepaged
25 root    -20  0      0  0      0  0      0 I  0.0   0.0  0:00.00 crypto
26 root    -20  0      0  0      0  0      0 I  0.0   0.0  0:00.00 kintegrityd
27 root    -20  0      0  0      0  0      0 I  0.0   0.0  0:00.00 kblockd
28 root    -20  0      0  0      0  0      0 I  0.0   0.0  0:00.00 tpm_dev_wq
29 root    -20  0      0  0      0  0      0 I  0.0   0.0  0:00.00 ata_sff
30 root    -20  0      0  0      0  0      0 I  0.0   0.0  0:00.00 md
31 root    -20  0      0  0      0  0      0 I  0.0   0.0  0:00.00 edac-poller
32 root    -20  0      0  0      0  0      0 I  0.0   0.0  0:00.00 devfreq_wq
33 root     rt  0      0  0      0  0      0 S  0.0   0.0  0:00.00 watchdogd
36 root     20   0      0  0      0  0      0 I  0.0   0.0  0:02.21 kworker/0:2-eve
39 root     20   0      0  0      0  0      0 S  0.0   0.0  0:00.00 kswapd0
40 root    -20  0      0  0      0  0      0 I  0.0   0.0  0:00.00 kworker/u3:0
41 root     20   0      0  0      0  0      0 S  0.0   0.0  0:00.00 encryptfs-kthrea
130 root   -20  0      0  0      0  0      0 I  0.0   0.0  0:00.00 kthrotld
131 root   -20  0      0  0      0  0      0 I  0.0   0.0  0:00.00 acpi_thermal_pm
132 root   -20  0      0  0      0  0      0 S  0.0   0.0  0:00.01 scsi_eh_0
133 root   -20  0      0  0      0  0      0 I  0.0   0.0  0:00.00 scsi_tmf_0
```

## **COMMAND 15 :- kill**

(To terminate a process use “kill”)

- a) kill -l (To display all the available signals you can use below command option)**

```
ritik@ritik-virtual-machine:~$ 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
ritik@ritik-virtual-machine:~$
```

- b) kill process**

```
ritik@ritik-virtual-machine:~$ ps
  PID TTY      TIME CMD
 11844 pts/0    00:00:00 bash
 11866 pts/0    00:00:00 ps
ritik@ritik-virtual-machine:~$ kill 11844
ritik@ritik-virtual-machine:~$
```

## **COMMAND 16: - PIPE EXAMPLES**

### **a) ls | more**

(output of ls command is given as input to more command So that output is printed one screen full page at a time ).

```
ritik@ritik-virtual-machine:~$ ls |more
Desktop
Documents
Downloads
hello
Music
Pictures
Public
Templates
Videos
ritik@ritik-virtual-machine:~$
```

### **b) Who | sort**

( output of who command is given as input to sort command so that it will print sorted list of users )

```
ritik@ritik-virtual-machine:~$ who | sort
ritik :0          2021-07-27 13:23 (:0)
ritik@ritik-virtual-machine:~$
```

### **c) Who | wc -l**

(output of who command is given as input to wc command so that So that it will print number of users who log on to the computer)

```
ritik@ritik-virtual-machine:~$ who | wc -l
1
ritik@ritik-virtual-machine:~$
```

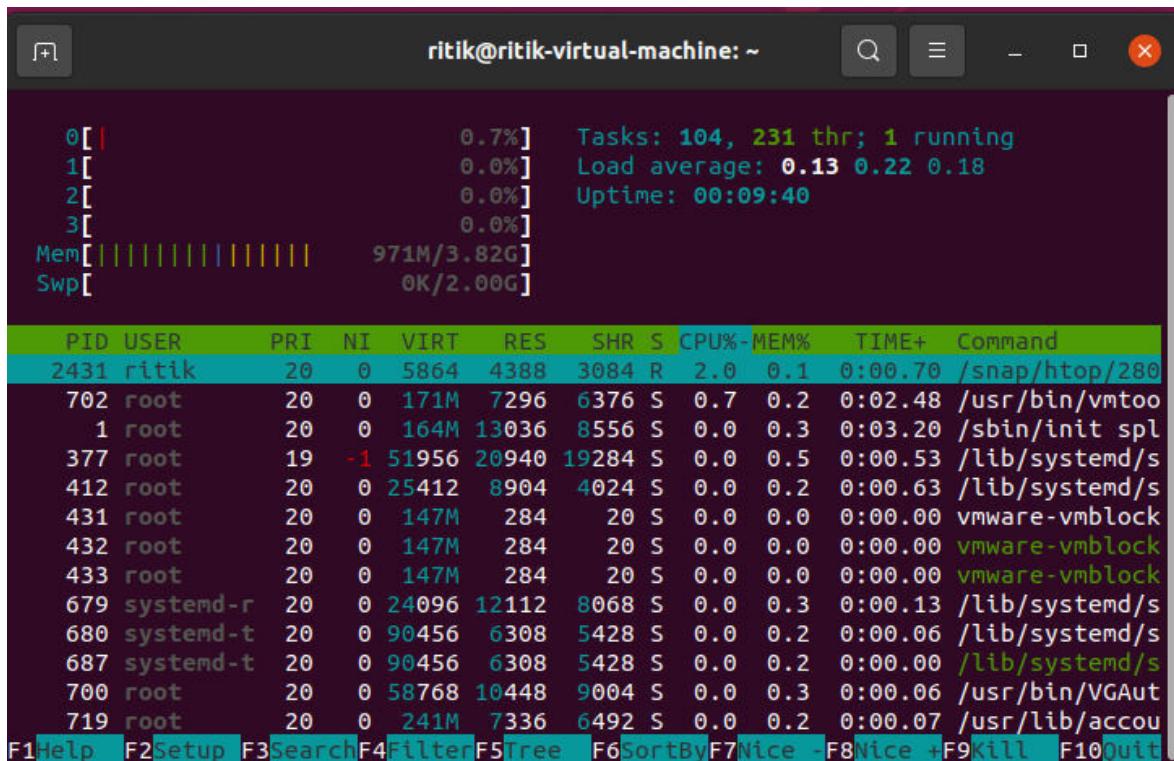
## COMMAND 17: - grep

(To search files in a directory for a specific string use “grep”)

```
ritik@ritik-virtual-machine:~$ grep "unix" sample.txt
unix is great to work. unix is free to use.
ritik@ritik-virtual-machine:~$
```

## COMMAND 18: - htop

(The htop command is an improved top.)



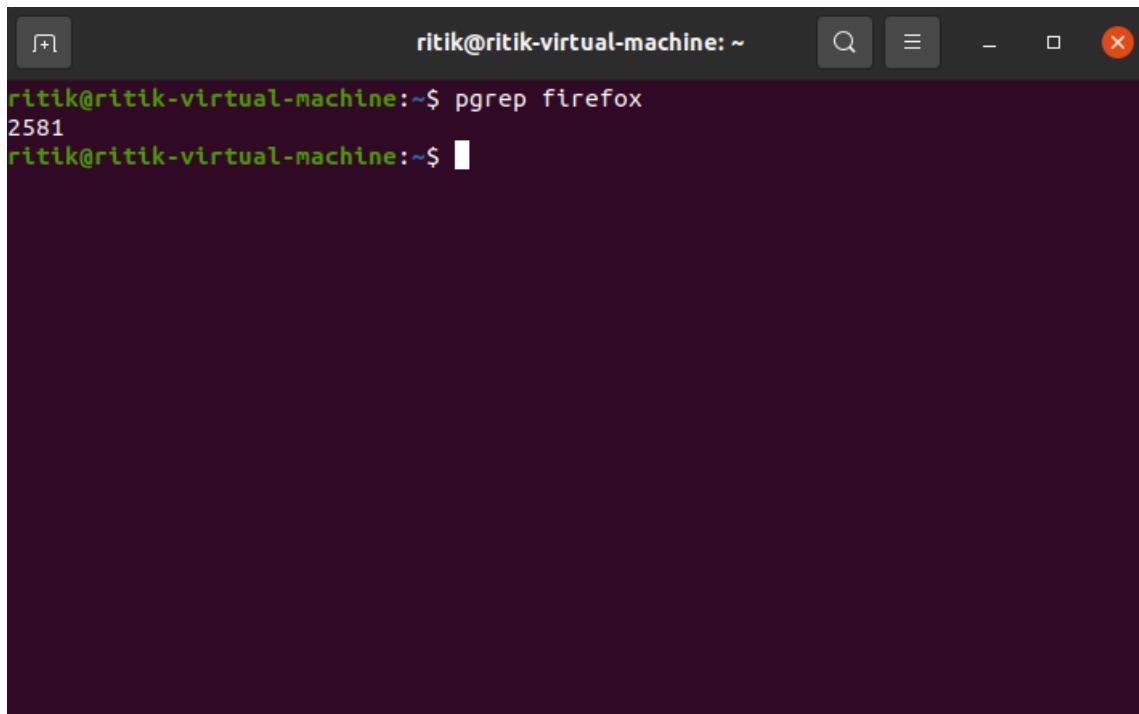
## COMMAND 19: -pstree

(The **pstree** command is another way of visualizing processes. It displays them in tree format.)

```
ritik@ritik-virtual-machine:~$ pstree
systemd—ModemManager—2*[{ModemManager}]
      | NetworkManager—2*[{NetworkManager}]
      | VAuthService
      | accounts-daemon—2*[{accounts-daemon}]
      | acpid
      | avahi-daemon—avahi-daemon
      | colord—2*[{colord}]
      | cron
      | cups-browsed—2*[{cups-browsed}]
      | cupsd
      | dbus-daemon
      | gdm3—gdm-session-wor—gdm-x-session—Xorg—{Xorg}
          |          | gnome-session-b—ssh-agent
          |          | 2*[{gnome+}]
          |          | 2*[{gdm-x-session}]
          |          | 2*[{gdm-session-wor}]
          |          | 2*[{gdm3}]
          |          | gnome-keyring-d—3*[{gnome-keyring-d}]
          |          | irqbalance—{irqbalance}
          |          | 2*[{kerneloops}]
          |          | networkd-dispat
          |          | polkitd—2*[{polkitd}]
          |          | rsyslogd—3*[{rsyslogd}]
          |          | rtkit-daemon—2*[{rtkit-daemon}]
          |          | snapd—15*[{snapd}]
          |          | switcheroo-cont—2*[{switcheroo-cont}]
          |          | systemd—(sd-pam)
          |          |           at-spi-bus-laun—dbus-daemon
          |          |           3*[{at-spi-bus-laun}]
          |          |           at-spi2-registr—2*[{at-spi2-registr}]
          |          |           dbus-daemon
```

## COMMAND 20: - pgrep

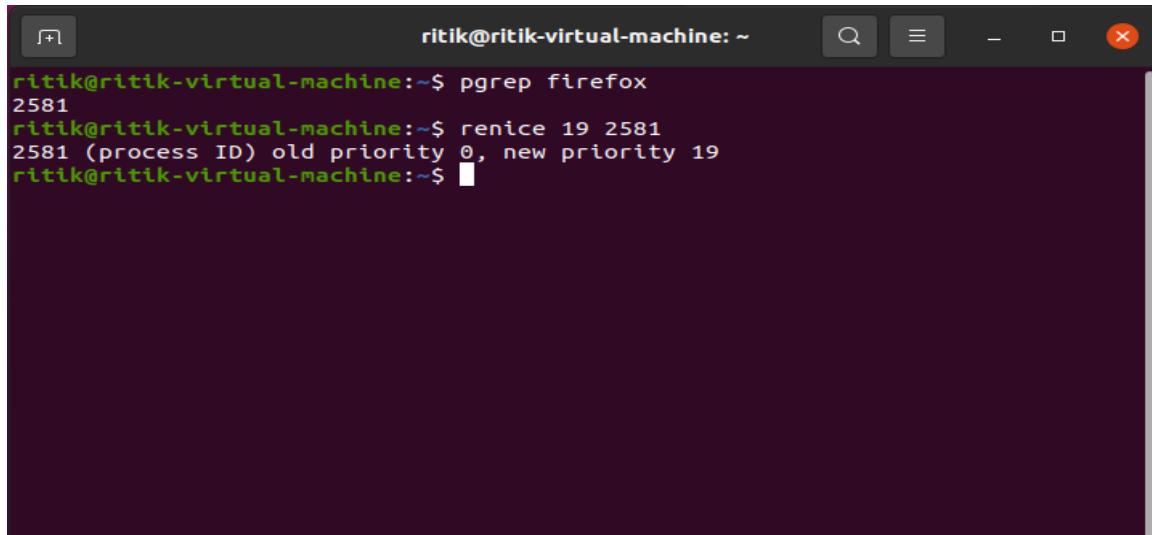
(Given a search term, **pgrep** returns the process IDs that match it)



```
ritik@ritik-virtual-machine:~$ pgrep firefox
2581
ritik@ritik-virtual-machine:~$
```

## COMMAND 21: - renice

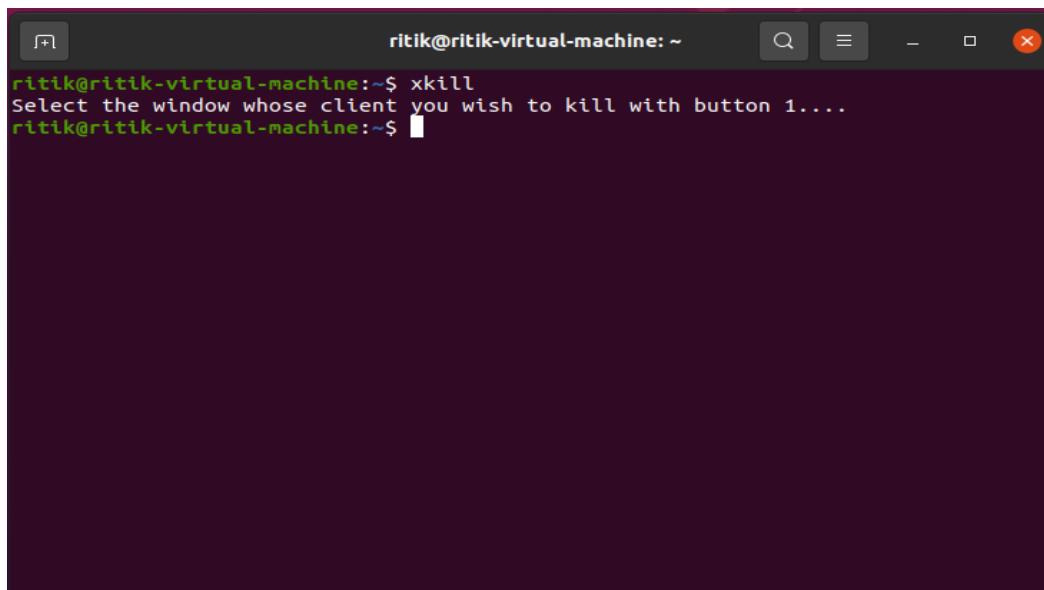
(The **renice** command changes the nice value of an already running process. The nice value determines what priority the process runs with. A value of **-19** is very high priority, while a value of **19** is very low priority. A value of **0** is the default priority.)



```
ritik@ritik-virtual-machine:~$ pgrep firefox
2581
ritik@ritik-virtual-machine:~$ renice 19 2581
2581 (process ID) old priority 0, new priority 19
ritik@ritik-virtual-machine:~$ █
```

## COMMAND 22: - xkill

(The **xkill** command is a way of easily killing graphical programs. Run it and your cursor will turn into an x sign. Click a program's window to kill that program. If you don't want to kill a program, you can back out of xkill by right-clicking instead)



```
ritik@ritik-virtual-machine:~$ xkill
Select the window whose client you wish to kill with button 1....
ritik@ritik-virtual-machine:~$ █
```

## COMMAND 23: - iostat

(The [iostat command](#) lists CPU utilization, device utilization and network file system utilization. This utility displays system's average CPU utilization since the last reboot. Running the following command without any options displays all the three reports.)

```
ritik@ritik-virtual-machine:~$ iostat
Linux 5.8.0-59-generic (ritik-virtual-machine) 28/07/21      _x86_64_      (4 CPU)

avg-cpu: %user   %nice %system %iowait  %steal   %idle
          1.32    1.08   4.80   14.78    0.00   78.02

Device     tps   kB_read/s   kB_wrtn/s   kB_dscd/s   kB_read   kB_w
rtn      kB_dscd
loop0      0.23      1.83        0.00        0.00      347
  0
loop1      0.33      5.63        0.00        0.00     1069
  0
loop10     2.09     63.36        0.00        0.00    12027
  0
loop11     0.23      1.83        0.00        0.00      347
```

## COMMAND 24: - mpstat

(The [mpstat command](#) writes to standard output about activities for each available processor, processor 0 being the first one.)

```
ritik@ritik-virtual-machine:~$ mpstat
Linux 5.8.0-59-generic (ritik-virtual-machine) 28/07/21      _x86_64_      (4 CPU)

11:36:44 AM IST  CPU  %usr   %nice   %sys %iowait  %irq   %soft  %steal  %guest  %gnice
  %idle
11:36:44 AM IST  all   1.05    0.82    3.17   11.21    0.00    0.55    0.00    0.00    0.00
  83.20
ritik@ritik-virtual-machine:~$
```

## COMMAND 25: - sar

(The sar is a performance monitoring tool for collecting, viewing and recording performance data and can list what a system is doing all the time. )

```
ritik@ritik-virtual-machine:~$ sar 10 3
Linux 5.8.0-59-generic (ritik-virtual-machine) 28/07/21      _x86_64_      (4 CPU)

11:37:31 AM IST      CPU      %user      %nice      %system      %iowait      %steal      %idle
11:37:41 AM IST      all      0.35      6.07      2.13      0.18      0.00      91.27
11:37:51 AM IST      all      0.00      0.00      0.05      0.00      0.00      99.95
^C

Average:      all      0.18      3.03      1.09      0.09      0.00      95.61
ritik@ritik-virtual-machine:~$
```

## COMMAND 26: - dstat

(dstat command is a versatile tool for generating system resource statistics and it would also show cpu stats. You need to install it in order to use it)

```
ritik@ritik-virtual-machine:~$ dstat
You did not select any stats, using -cdngy by default.
--total-cpu-usage-- -dsk/total- -net/total- ---paging-- ---system--
usr sys idl wai stl| read writ| recv send| in out| int csw
 2  2 89  7 |1547k 229k| 0  0 | 0  0 | 498 723
 0  1 99  0 | 0  576k| 0  0 | 0  0 | 350 468
 0  1 99  0 | 0  0 | 0  0 | 0  0 | 367 641
 1  1 99  0 | 0  0 | 0  0 | 0  0 | 272 562
 0  1 99  0 | 0  0 | 0  0 | 0  0 | 183 294
```

## COMMAND 27: - who ,whoami

(It displays the username of the current user when this **command** is invoked)

```
ritik@ritik-virtual-machine:~$ who
ritik :0          2021-07-28 11:33 (:0)
ritik@ritik-virtual-machine:~$ whoami
ritik
ritik@ritik-virtual-machine:~$
```

## **COMMAND 28: - finger**

(On Unix-like operating systems, the **finger** command looks up and displays information about system users.)

```
ritik@ritik-virtual-machine:~$ finger
Login      Name      Tty      Idle  Login Time   Office      Office Phone
ritik      RITIK      *:0          Jul 28 11:33 (:0)
ritik@ritik-virtual-machine:~$
```

## **COMMAND 29: - netstat**

(**netstat** (network statistics) is a **command-line** tool that displays network connections (both incoming and outgoing), routing tables, and a number of network interface statistics. )

```
ritik@ritik-virtual-machine:~$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 ritik-virtual-mac:43640 43.255.166.254:http    TIME_WAIT
udp        0      0 ritik-virtual-mac:59351 _gateway:domain        ESTABLISHED
udp        0      0 ritik-virtual-ma:bootpc 192.168.16.254:bootps  ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type            State           I-Node Path
unix    2      [ ]        DGRAM           CONNECTED     25137  /run/systemd/journal/syslog
unix    2      [ ]        DGRAM           CONNECTED     41889  /run/user/1000/systemd/notify
unix   16      [ ]        DGRAM           CONNECTED     25147  /run/systemd/journal/dev-log
unix    8      [ ]        DGRAM           CONNECTED     25151  /run/systemd/journal/socket
unix    3      [ ]        DGRAM           CONNECTED     25123  /run/systemd/notify
unix    3      [ ]        STREAM          CONNECTED     42767
unix    3      [ ]        STREAM          CONNECTED     45218
unix    3      [ ]        STREAM          CONNECTED     51070  @/tmp/dbus-8htyNbd8aR
unix    3      [ ]        STREAM          CONNECTED     47833
unix    3      [ ]        STREAM          CONNECTED     47740  /run/systemd/journal/stdout
unix    3      [ ]        STREAM          CONNECTED     43841
```

## **COMMAND 30: - date;who**

(Will print today's date followed by users who are currently login.)

```
ritik@ritik-virtual-machine:~$ date;who
Wednesday 28 July 2021 11:49:23 AM IST
ritik      :0          2021-07-28 11:33 (:0)
ritik@ritik-virtual-machine:~$
```

## **CONCLUSION-**

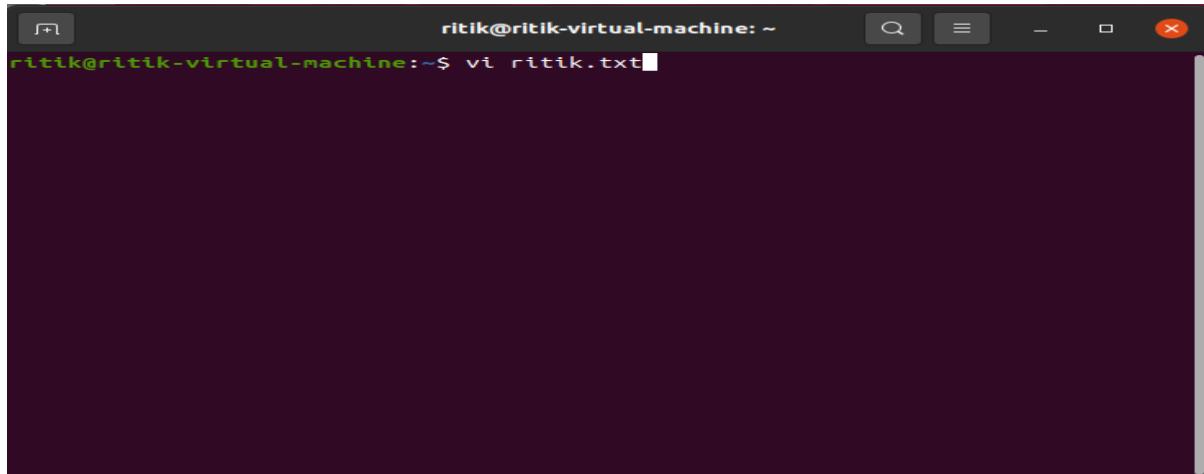
Got know about various linux commands and their functionality.



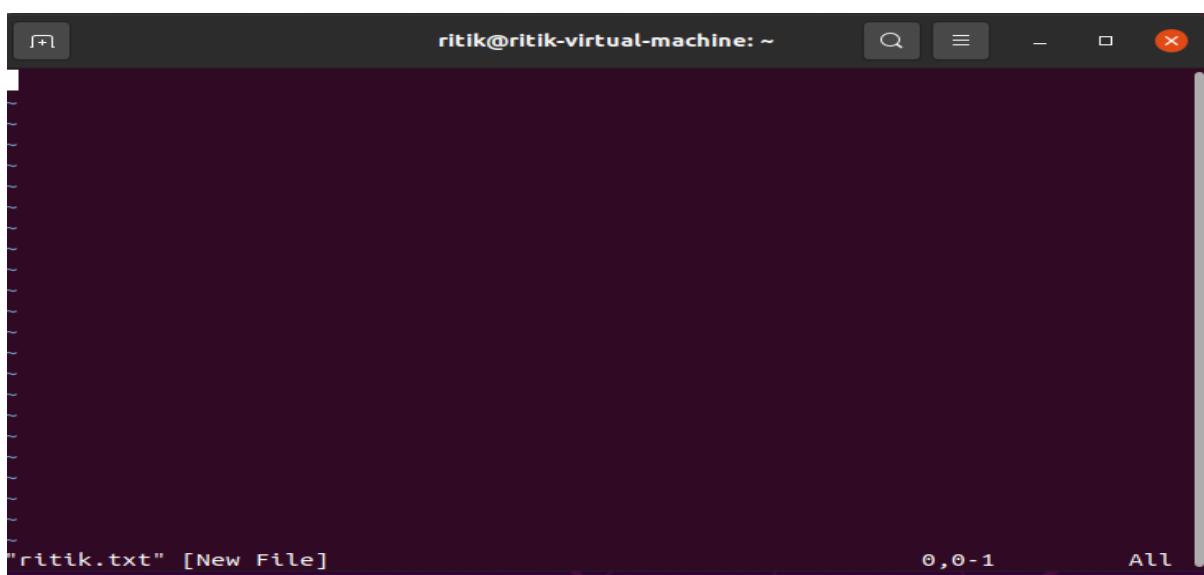
# Experiment 3

**Aim:** To perform various operation on VI Editor

- **vi filename** - Creates a new file if it already does not exist, otherwise opens an existing file.



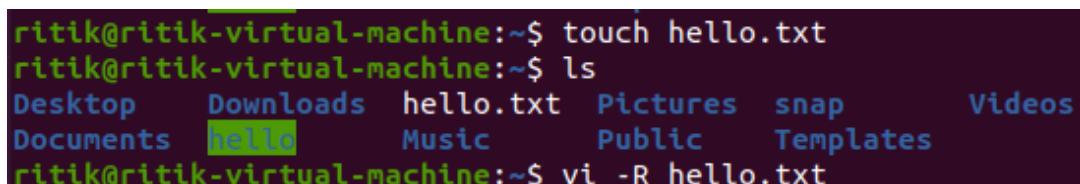
```
ritik@ritik-virtual-machine:~$ vi ritik.txt
```



```
ritik@ritik-virtual-machine:~$
```

"ritik.txt" [New File] 0,0-1 All

- **vi -R** - filename Opens an existing file in the read-only mode.



```
ritik@ritik-virtual-machine:~$ touch hello.txt
ritik@ritik-virtual-machine:~$ ls
Desktop  Downloads  hello.txt  Pictures  snap      Videos
Documents  hello    Music      Public    Templates
ritik@ritik-virtual-machine:~$ vi -R hello.txt
```

```
ritik@ritik-virtual-machine: ~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
"hello.txt" [readonly] 0L, 0C
```

```
0,0-1
```

```
All
```

- **view filename** - Opens an existing file in the read-only mode.

```
ritik@ritik-virtual-machine: ~
```

```
ritik@ritik-virtual-machine:~$ view hello.txt
```

## Deleting Characters

- **x Deletes** - the character under the cursor location

A screenshot of a terminal window titled "ritik@ritik-virtual-machine: ~". The terminal shows the following text:  
my name is ritik verma  
hello world 1234  
I love linux█  
The cursor is at the end of the word "linux". Below the terminal window, the status bar displays "3, 13" and "All".

A screenshot of a terminal window titled "ritik@ritik-virtual-machine: ~". The terminal shows the following text:  
my name is ritik verma  
hello world 1234  
I love linu█  
The cursor is at the end of the word "linu". Below the terminal window, the status bar displays "3, 13" and "All".

- **X -** Deletes the character before the cursor location

A screenshot of a terminal window titled "ritik@ritik-virtual-machine: ~". The terminal shows the following text:  
my name is ritik verma  
hello world 1234  
I love liu█  
The cursor is at the end of the word "liu". Below the terminal window, the status bar displays "3, 13" and "All".

- **dw** - Deletes from the current cursor location to the next word

A screenshot of a terminal window titled "ritik@ritik-virtual-machine: ~". The terminal has a dark background and light-colored text. It displays the following text:  
my name is ritik verma  
hello world 1234  
I love **linux**  
~  
~  
~  
~

A screenshot of a terminal window titled "ritik@ritik-virtual-machine: ~". The terminal has a dark background and light-colored text. It displays the following text:  
my name is ritik verma  
hello world 1234  
I love**l**inux  
~  
~  
~  
~

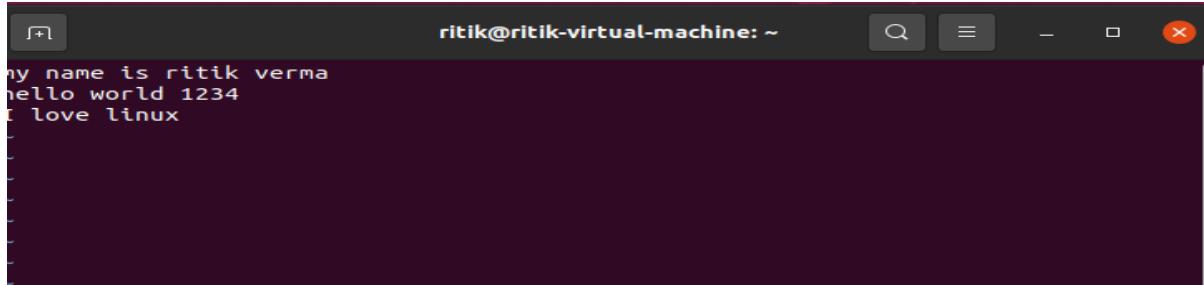
- **d^** - Deletes from the current cursor position to the beginning of the line

A screenshot of a terminal window titled "ritik@ritik-virtual-machine: ~". The terminal has a dark background and light-colored text. It displays the following text:  
my name is ritik verma  
hello world 1234  
**l**inux  
~  
~  
~  
~

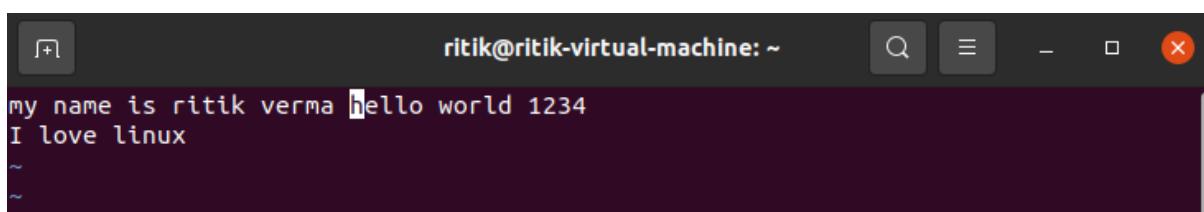
## Advanced Commands

There are some advanced commands that simplify day-to-day editing and allow for more efficient use of vi –

- **J-** Joins the current line with the next one. A count of j commands joins many lines.

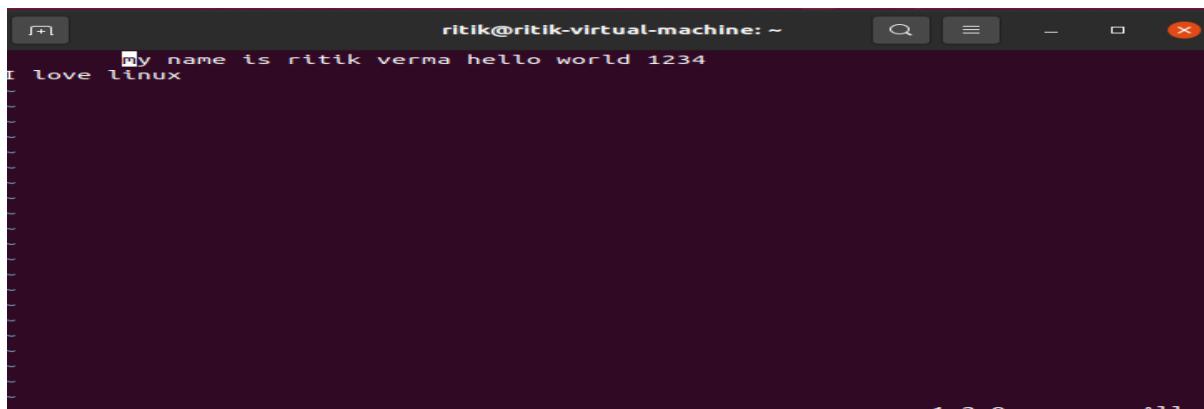


A screenshot of a terminal window titled "ritik@ritik-virtual-machine: ~". The terminal shows the following text:  
my name is ritik verma  
hello world 1234  
I love linux  
~  
~  
The cursor is at the end of the first line. The terminal interface includes a title bar, a search bar, and standard window control buttons.

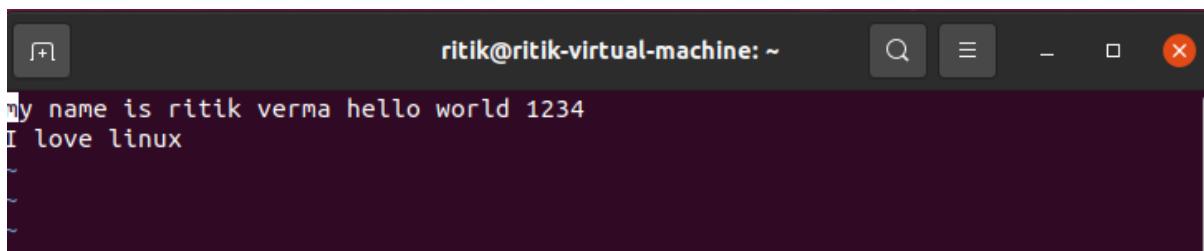


A screenshot of a terminal window titled "ritik@ritik-virtual-machine: ~". The terminal shows the following text:  
my name is ritik verma hello world 1234  
I love linux  
~  
~  
The cursor is at the end of the joined line. The terminal interface includes a title bar, a search bar, and standard window control buttons.

- **<< Shifts** - the current line to the left by one shift width

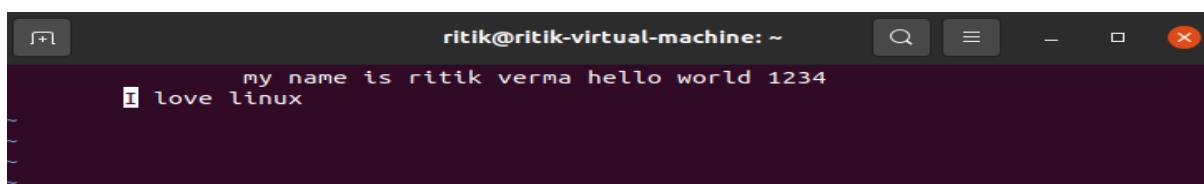


A screenshot of a terminal window titled "ritik@ritik-virtual-machine: ~". The terminal shows the following text:  
my name is ritik verma hello world 1234  
I love linux  
~  
~  
The cursor is at the start of the second line. The terminal interface includes a title bar, a search bar, and standard window control buttons.



A screenshot of a terminal window titled "ritik@ritik-virtual-machine: ~". The terminal shows the following text:  
my name is ritik verma hello world 1234  
I love linux  
~  
~  
The cursor is at the start of the first line. The terminal interface includes a title bar, a search bar, and standard window control buttons.

- **>> Shifts** - the current line to the right by one shift width.



A screenshot of a terminal window titled "ritik@ritik-virtual-machine: ~". The terminal shows the following text:  
my name is ritik verma hello world 1234  
I love linux  
~  
~  
The cursor is at the end of the first line. The terminal interface includes a title bar, a search bar, and standard window control buttons.

- **G**- Press Ctrl and G keys at the same time to show the current filename and the status.

```
ritik@ritik-virtual-machine: ~
```

- **u-** This helps undo the last change that was done in the file. Typing 'u' again will re-do the change.

```
ritik@ritik-virtual-machine: ~
```

- :r -file Reads file and inserts it after the current line.

The image shows two terminal windows side-by-side. Both windows have a dark theme with a light-colored title bar. The title bar of both windows reads "ritik@ritik-virtual-machine: ~". The first window contains the following text:

```
my name is ritik verma
hello world 1234
I love linux
```

The second window contains the same text, followed by a blank line, and then seven tilde (~) characters, indicating the position of the cursor. This demonstrates that the text from the first window was read and inserted after the current line in the second window.

## Word and Character Searching

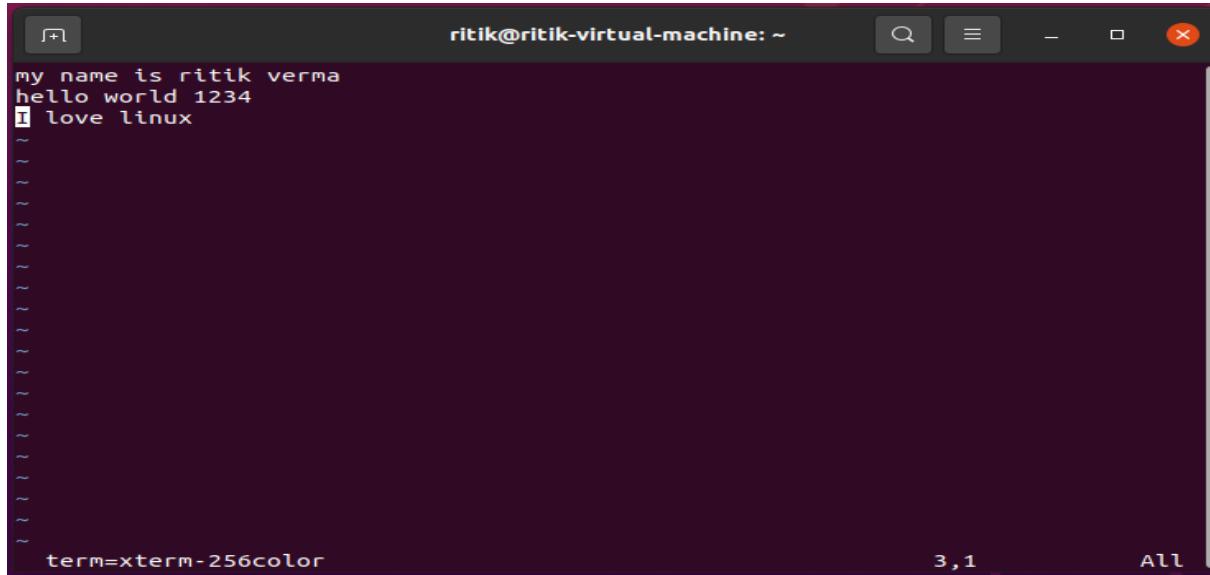
The image shows a single terminal window with a dark theme and a light-colored title bar. The title bar reads "ritik@ritik-virtual-machine: ~". The terminal window displays the following text:

```
my name is ritik verma
hello world 1234
I love linux
~
```

The word "linux" is highlighted in the text. At the bottom of the window, there is a prompt "?linux" followed by a cursor, indicating that the user has entered a search command and is awaiting results.

## Set Commands

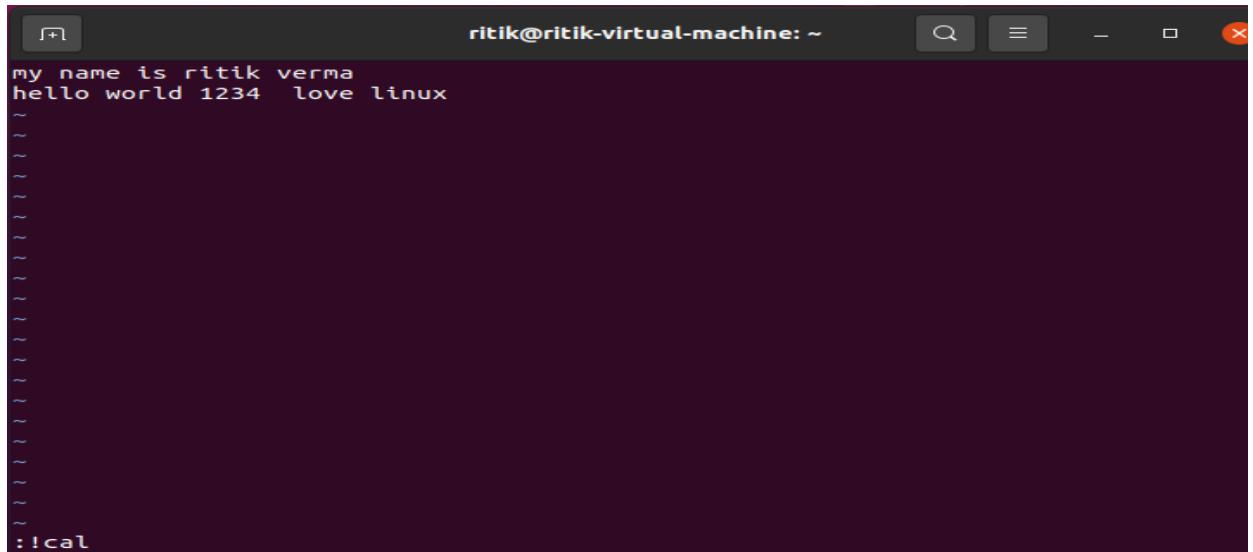
**: set term**- Prints terminal type



A screenshot of a terminal window titled "ritik@ritik-virtual-machine: ~". The window shows the command "term=xterm-256color" at the bottom. The main area displays the text "my name is ritik verma", "hello world 1234", and "I love linux", followed by several blank lines.

## Running Commands

The vi has the capability to run commands from within the editor. To run a command, you only need to go to the command mode and type `:! command`. For example, if you want to check whether a file exists before you try to save your file with that filename, you can type `:! ls` and you will see the output of `ls` on the screen. You can press any key (or the command's escape sequence) to return to your vi session.



A screenshot of a terminal window titled "ritik@ritik-virtual-machine: ~". The main area displays the text "my name is ritik verma", "hello world 1234", "love linux", and several blank lines. At the bottom, the command ": !cal" is visible.

```
ritik@ritik-virtual-machine:~$ vi hello.txt
[No write since last change]
Desktop    hello      Pictures  task.sh    text.sh.save
Documents  hello.txt  Public    Templates  Videos
Downloads  Music      snap     text.sh    welcome.sh

Press ENTER or type command to continue
[No write since last change]
August 2021
Su Mo Tu We Th Fr Sa
1  2  3  4  5  6  7
8  9  10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

Press ENTER or type command to continue
```

# EXPERIMENT NO.4

**Aim:** - To perform shell scripting in linux.

- **Bash**

Bash is a command language interpreter. It is widely available on various operating systems and is a default command interpreter on most GNU/Linux systems. The name is an acronym for the ‘Bourne-Again SHell’.

- **Shell**

Shell is a macro processor which allows for an interactive or non-interactive command execution.

- **Scripting**

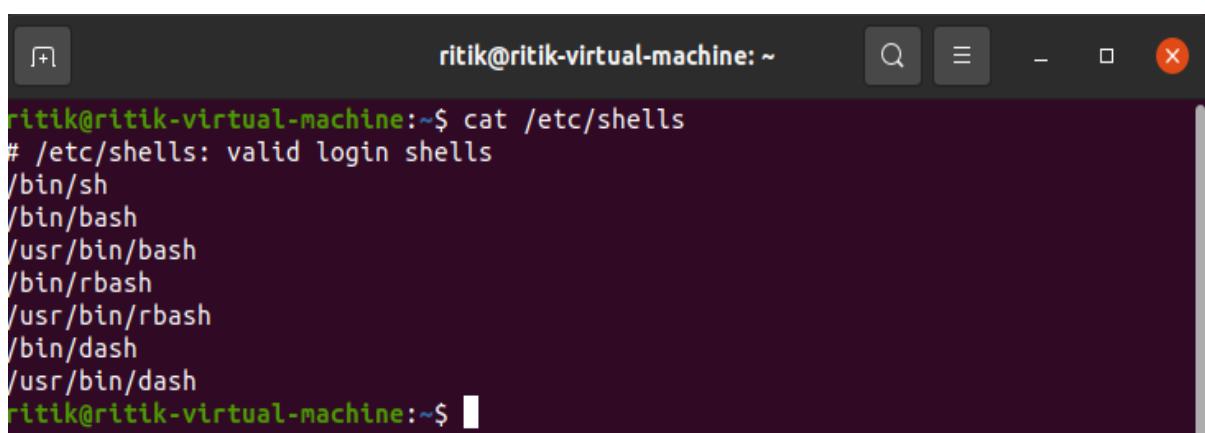
Scripting allows for an automatic commands execution that would otherwise be executed interactively one-by-one.

There are several more shells. Some of them are →

- Bash (Bourne again shell)
- C shell
- Bourne Shell.

The shells that are in our system are along with their location are stored in /etc/shells directory. To view them, we use the following command →

```
$ cat /etc/shells
```



The screenshot shows a terminal window with a dark background and light-colored text. The title bar reads "ritik@ritik-virtual-machine: ~". The command "cat /etc/shells" was run, and its output is displayed. The output lists several shell paths: "/bin/sh", "/bin/bash", "/usr/bin/bash", "/bin/rbash", "/usr/bin/rbash", "/bin/dash", and "/usr/bin/dash". The terminal window has standard Linux-style window controls at the top right.

```
ritik@ritik-virtual-machine:~$ cat /etc/shells
# /etc/shells: valid login shells
/bin/sh
/bin/bash
/usr/bin/bash
/bin/rbash
/usr/bin/rbash
/bin/dash
/usr/bin/dash
ritik@ritik-virtual-machine:~$
```

What you have just done, was that by use of commands and shell you interacted with your computer to retrieve a current date and time (date), looked up a calendar (cal), checked the location of your current working directory (pwd) and retrieved a list of all files and directories located within (ls).

```
ritik@ritik-virtual-machine:~$ cal
      August 2021
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

ritik@ritik-virtual-machine:~$ date
Tuesday 03 August 2021 02:16:55 PM IST
ritik@ritik-virtual-machine:~$ ls
Desktop  Downloads  hello.txt  Pictures  snap      Videos
Documents  hello    Music     Public    Templates
ritik@ritik-virtual-machine:~$ pwd
/home/ritik
ritik@ritik-virtual-machine:~$
```

To see what is meant by scripting, use shell in combination with your favorite text editor eg. vi to create a new file called task.sh containing all the above commands, each on a separate line. Once ready, make your new file executable using chmod command with an option +x. Lastly, execute your new script by prefixing its name with./.

```
ritik@ritik-virtual-machine: ~
date
cal
pwd
ls
~
```

```
ritik@ritik-virtual-machine:~$ vim task.sh
ritik@ritik-virtual-machine:~$ chmod +x task.sh
ritik@ritik-virtual-machine:~$ ./task.sh
Tuesday 03 August 2021 02:22:41 PM IST
    August 2021
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

/home/ritik
Desktop  Downloads  hello.txt  Pictures  snap      Templates
Documents  hello      Music      Public    task.sh  Videos
ritik@ritik-virtual-machine:~$
```

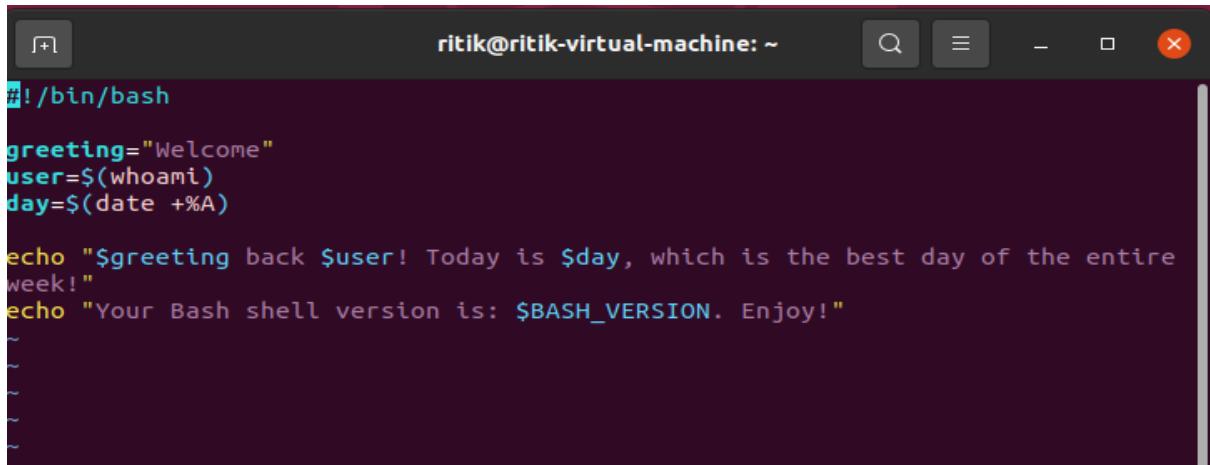
## Hello World Bash Shell Script

```
ritik@ritik-virtual-machine: ~
```

```
ritik@ritik-virtual-machine:~$ vim helloworld.sh
ritik@ritik-virtual-machine:~$ chmod +x helloworld.sh
chmod: cannot access 'helloworld.sh': No such file or directory
ritik@ritik-virtual-machine:~$ chmod +x helloworld.sh
ritik@ritik-virtual-machine:~$ ./helloworld.sh
Hello World
ritik@ritik-virtual-machine:~$
```

## Variables

Variables are the essence of programming. Variables allow a programmer to store data, alter and reuse them throughout the script

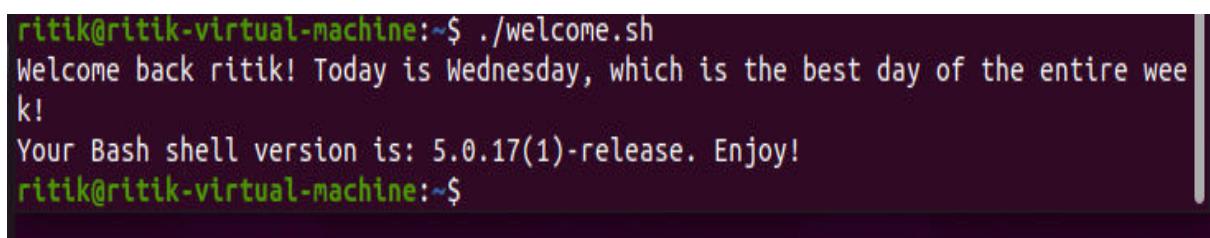


```
#!/bin/bash

greeting="Welcome"
user=$(whoami)
day=$(date +%A)

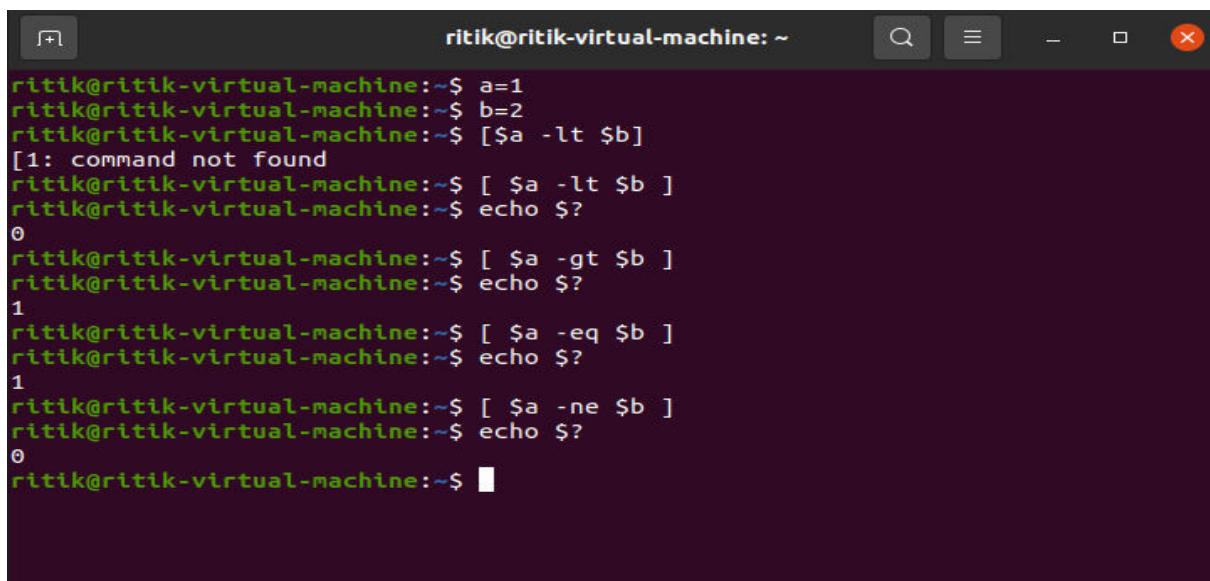
echo "$greeting back $user! Today is $day, which is the best day of the entire week!"
echo "Your Bash shell version is: $BASH_VERSION. Enjoy!"

~
```

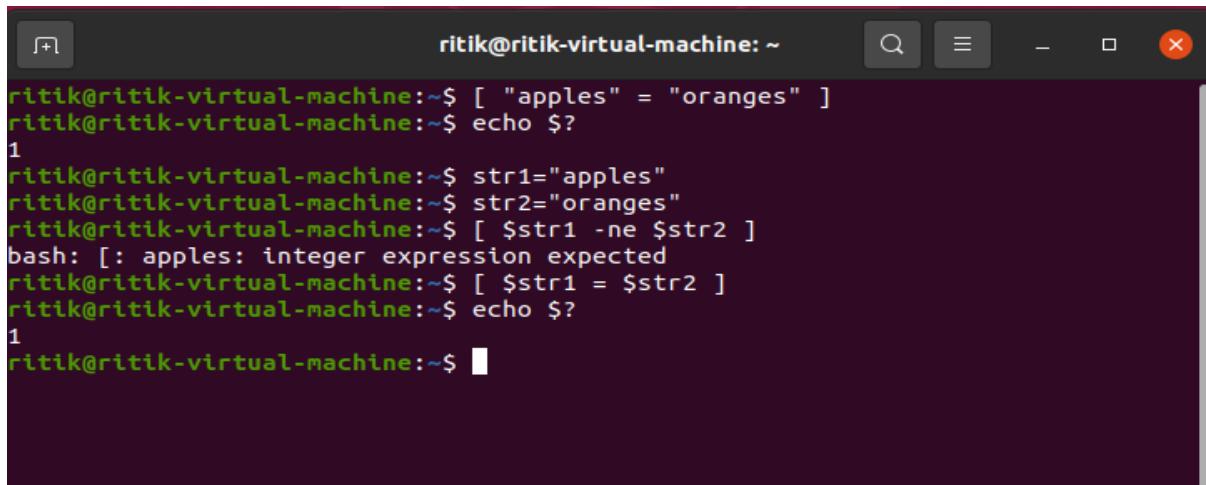


```
ritik@ritik-virtual-machine:~/Documents$ ./welcome.sh
Welcome back ritik! Today is Wednesday, which is the best day of the entire wee
k!
Your Bash shell version is: 5.0.17(1)-release. Enjoy!
ritik@ritik-virtual-machine:~/Documents$
```

## Numeric and String Comparisons



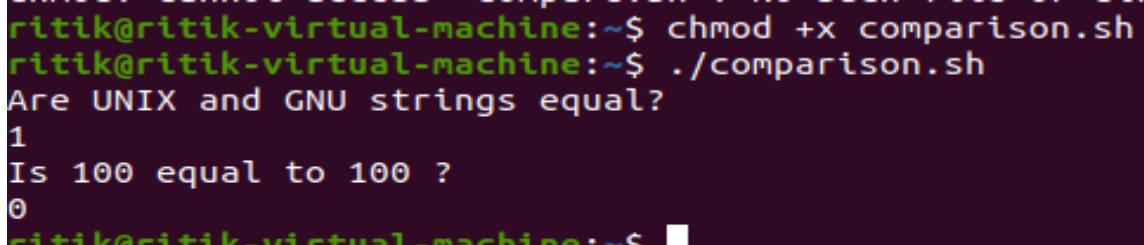
```
ritik@ritik-virtual-machine:~/Documents$ a=1
ritik@ritik-virtual-machine:~/Documents$ b=2
ritik@ritik-virtual-machine:~/Documents$ [ $a -lt $b ]
[1: command not found
ritik@ritik-virtual-machine:~/Documents$ [ $a -lt $b ]
ritik@ritik-virtual-machine:~/Documents$ echo $?
0
ritik@ritik-virtual-machine:~/Documents$ [ $a -gt $b ]
ritik@ritik-virtual-machine:~/Documents$ echo $?
1
ritik@ritik-virtual-machine:~/Documents$ [ $a -eq $b ]
ritik@ritik-virtual-machine:~/Documents$ echo $?
1
ritik@ritik-virtual-machine:~/Documents$ [ $a -ne $b ]
ritik@ritik-virtual-machine:~/Documents$ echo $?
0
ritik@ritik-virtual-machine:~/Documents$
```



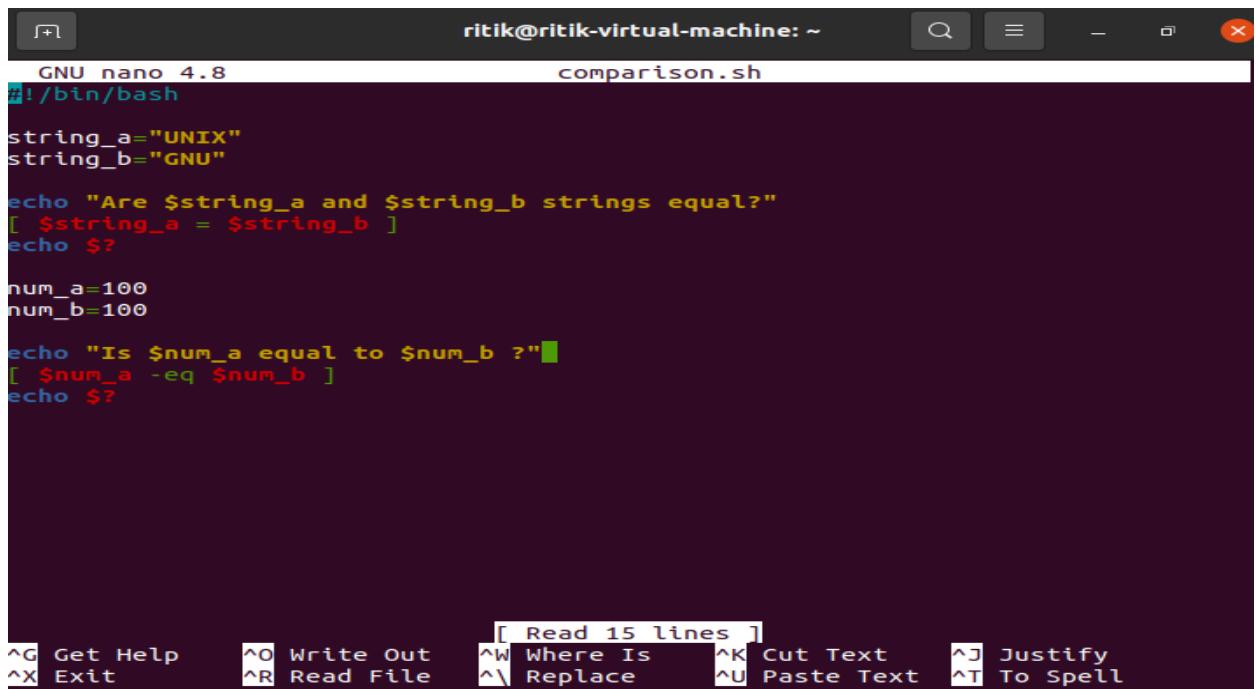
```
ritik@ritik-virtual-machine:~$ [ "apples" = "oranges" ]
ritik@ritik-virtual-machine:~$ echo $?
1
ritik@ritik-virtual-machine:~$ str1="apples"
ritik@ritik-virtual-machine:~$ str2="oranges"
ritik@ritik-virtual-machine:~$ [ $str1 -ne $str2 ]
bash: [: apples: integer expression expected
ritik@ritik-virtual-machine:~$ [ $str1 = $str2 ]
ritik@ritik-virtual-machine:~$ echo $?
1
ritik@ritik-virtual-machine:~$
```

If we were to translate the above knowledge to a simple bash shell script, the script would look as shown below. Using string comparison operator `=` we compare two distinct strings to see whether they are equal.

Similarly, we compare two integers using the numeric comparison operator to determine if they are equal in value. Remember, 0 signals true, while 1 indicates false:



```
ritik@ritik-virtual-machine:~$ chmod +x comparison.sh
ritik@ritik-virtual-machine:~$ ./comparison.sh
Are UNIX and GNU strings equal?
1
Is 100 equal to 100 ?
0
ritik@ritik-virtual-machine:~$
```



```
GNU nano 4.8              ritik@ritik-virtual-machine: ~ comparison.sh

#!/bin/bash

string_a="UNIX"
string_b="GNU"

echo "Are $string_a and $string_b strings equal?"
[ $string_a = $string_b ]
echo $?

num_a=100
num_b=100

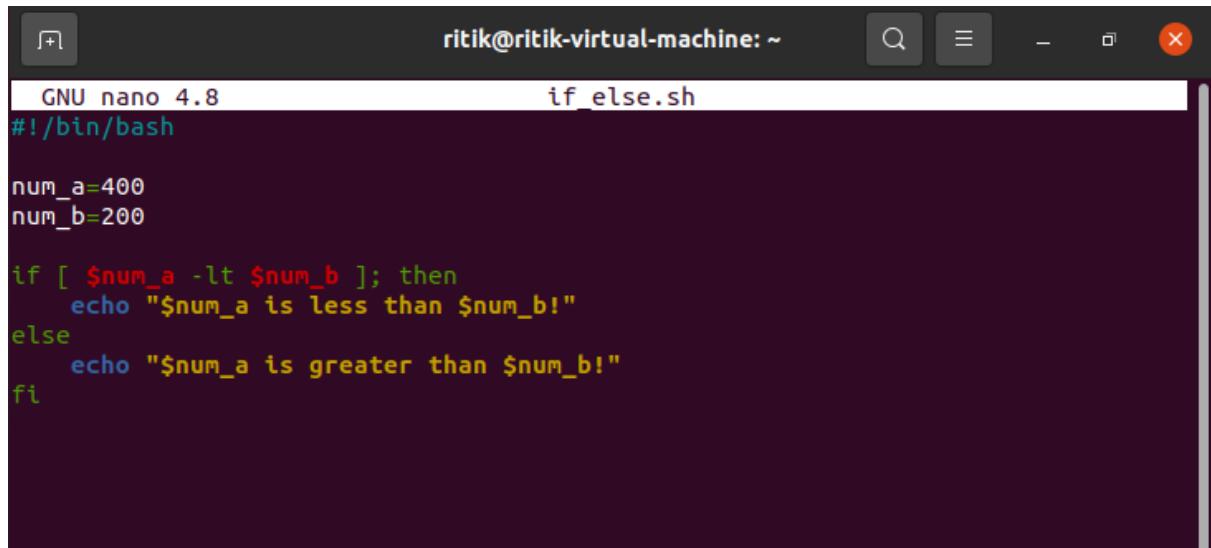
echo "Is $num_a equal to $num_b ?"
[ $num_a -eq $num_b ]
echo $?

[ Read 15 lines ]
```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify  
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell

## Conditional Statements

```
ritik@ritik-virtual-machine:~/Documents$ ./if_else.sh
400 is greater than 200!
ritik@ritik-virtual-machine:~/Documents$
```



The screenshot shows a terminal window with a dark theme. At the top, it says "ritik@ritik-virtual-machine: ~". Below that, the title bar of the nano editor window says "GNU nano 4.8" and "if\_else.sh". The main content of the editor window is a shell script:

```
GNU nano 4.8          if_else.sh
#!/bin/bash

num_a=400
num_b=200

if [ $num_a -lt $num_b ]; then
    echo "$num_a is less than $num_b!"
else
    echo "$num_a is greater than $num_b!"
fi
```

## Loops

### For loop

```
ritik@ritik-virtual-machine:~$ for i in 1 2 3; do echo $i; done
1
2
3
ritik@ritik-virtual-machine:~$
```

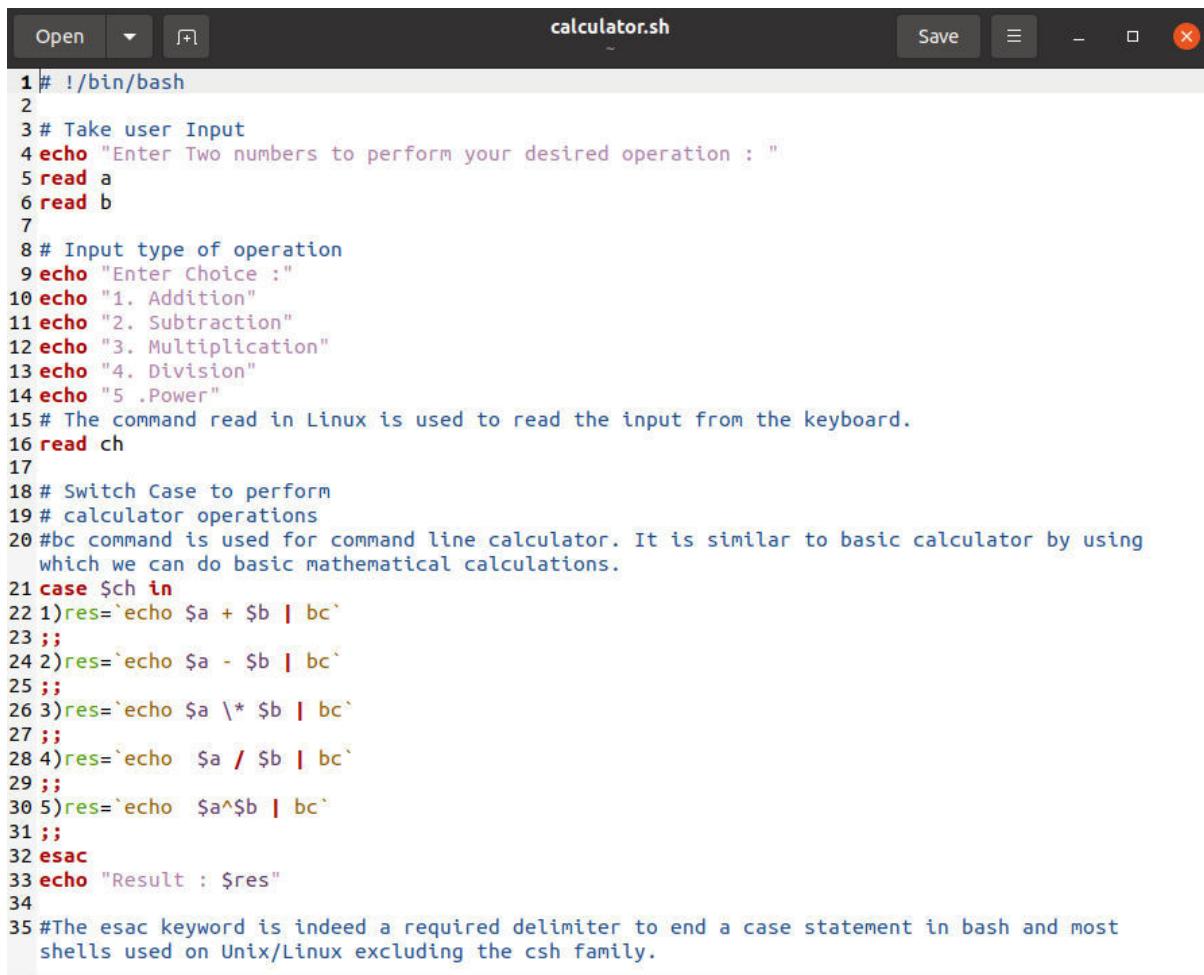
```
ritik@ritik-virtual-machine:~$ vim cat.txt
ritik@ritik-virtual-machine:~$ cat cat.txt
bash
scripting
tutorial

ritik@ritik-virtual-machine:~$ for i in $(cat cat.txt); do echo -n $i | wc -c;
done
4
9
8
ritik@ritik-virtual-machine:~$
```

## EXPERIMENT NO.5

**AIM:** - Implementation of simple calculator

**CODE:**



The screenshot shows a terminal window with the following details:

- File Name: calculator.sh
- File Type: bash script
- Content: The script implements a simple calculator using the bc command-line calculator. It takes two user inputs, performs a choice of arithmetic operations (Addition, Subtraction, Multiplication, Division, or Power), and outputs the result. The script uses a case statement to handle different operation types.

```
1 #!/bin/bash
2
3 # Take user Input
4 echo "Enter Two numbers to perform your desired operation : "
5 read a
6 read b
7
8 # Input type of operation
9 echo "Enter Choice :"
10 echo "1. Addition"
11 echo "2. Subtraction"
12 echo "3. Multiplication"
13 echo "4. Division"
14 echo "5 .Power"
15 # The command read in Linux is used to read the input from the keyboard.
16 read ch
17
18 # Switch Case to perform
19 # calculator operations
20 #bc command is used for command line calculator. It is similar to basic calculator by using
# which we can do basic mathematical calculations.
21 case $ch in
22 1)res=`echo $a + $b | bc`;;
23 ;;
24 2)res=`echo $a - $b | bc`;;
25 ;;
26 3)res=`echo $a \* $b | bc`;;
27 ;;
28 4)res=`echo $a / $b | bc`;;
29 ;;
30 5)res=`echo $a^$b | bc`;;
31 ;;
32 esac
33 echo "Result : $res"
34
35 #The esac keyword is indeed a required delimiter to end a case statement in bash and most
# shells used on Unix/Linux excluding the csh family.
```

## OUTPUT

```
ritik@ritik-virtual-machine:~$ ./calculator.sh
Enter Two numbers to perform your desired operation :
2
3
Enter Choice :
1. Addition
2. Subtraction
3. Multiplication
4. Division
5 .Power
1
Result : 5
```

```
ritik@ritik-virtual-machine:~$ ./calculator.sh
Enter Two numbers to perform your desired operation :
5
10
Enter Choice :
1. Addition
2. Subtraction
3. Multiplication
4. Division
5 .Power
2
Result : -5
```

```
ritik@ritik-virtual-machine:~$ ./calculator.sh
Enter Two numbers to perform your desired operation :
10
15
Enter Choice :
1. Addition
2. Subtraction
3. Multiplication
4. Division
5 .Power
3
Result : 150
ritik@ritik-virtual-machine:~$
```

```
ritik@ritik-virtual-machine:~$ ./calculator.sh
Enter Two numbers to perform your desired operation :
10
2
Enter Choice :
1. Addition
2. Subtraction
3. Multiplication
4. Division
5 .Power
4
Result : 5
```

```
ritik@ritik-virtual-machine:~/Documents$ ./calculator.sh
Enter Two numbers to perform your desired operation :
3
4
Enter Choice :
1. Addition
2. Subtraction
3. Multiplication
4. Division
5 .Power
5
Result : 81
ritik@ritik-virtual-machine:~/Documents$
```

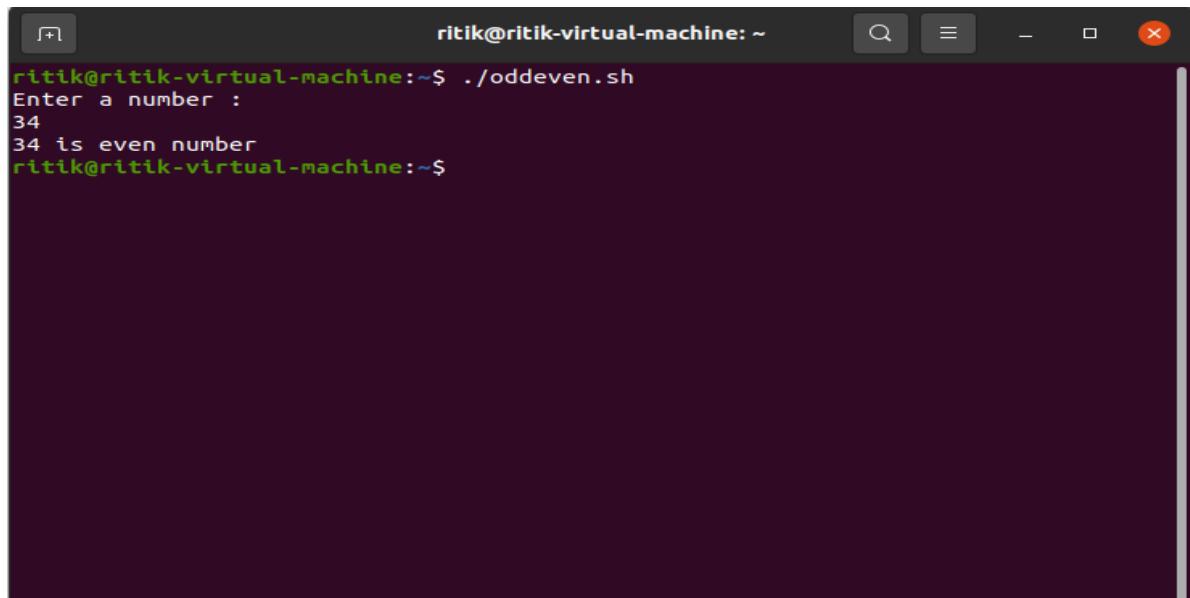
**AIM:** - To identify a given number as even or odd

**CODE:**

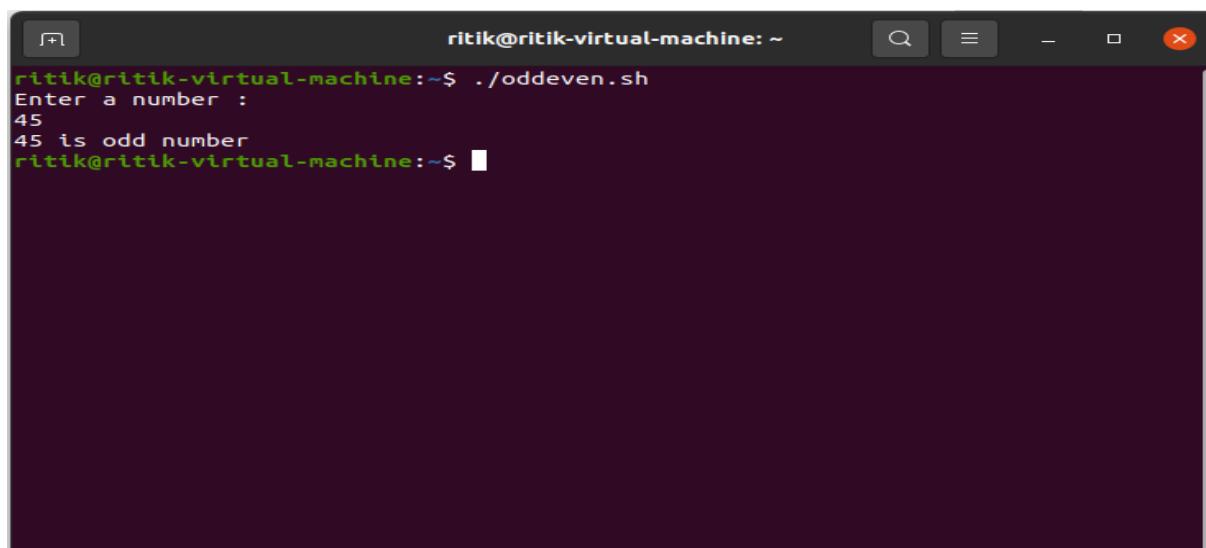


```
Open  oddeven.sh  Save  ~  -  x
1 echo "Enter a number : "
2 read n
3 rem=$(( $n % 2 ))
4 if [ $rem -eq 0 ]
5 then
6 echo "$n is even number"
7 else
8 echo "$n is odd number"
9 fi
```

**OUTPUT**



```
ritik@ritik-virtual-machine:~$ ./oddeven.sh
Enter a number :
34
34 is even number
ritik@ritik-virtual-machine:~$
```



```
ritik@ritik-virtual-machine:~$ ./oddeven.sh
Enter a number :
45
45 is odd number
ritik@ritik-virtual-machine:~$
```

**AIM:** - To find a given number in the list

**CODE:**

The screenshot shows a terminal window with a dark theme. The title bar of the window says "findnumber.sh". The window contains the following code:

```
1 #!/bin/bash
2 # To input array at run
3 # time by using for-loop
4
5 echo "Enter the Total numbers :"
6 read n
7 echo "Enter numbers:"
8 i=0
9
10 # Read upto the size of
11 # given array starting
12 # from index, i=0
13 while [ $i -lt $n ]
14 do
15     # To input from user
16     read a[$i]
17
18     # To increment index
19     # by 1, i=i+1
20     i=`expr $i + 1`
21 done
22
23 # Print the array starting
24 # from index, i=0
25
26
27
28 echo " enter the element to be found"
29 read find
30

for i in "${a[@]}"
do
    # access each element as $i
    if [ $find -eq $i ]
    then
        echo "number found"
        find=1
    fi
done

if [ $find != 1 ]
then echo "not found"
fi
```

The terminal status bar at the bottom shows "sh" with a dropdown arrow, "Tab Width: 8" with a dropdown arrow, "Ln 37, Col 18" with a dropdown arrow, and "INS".

## OUTPUT

```
ritik@ritik-virtual-machine:~/Desktop$ ./findnumber.sh
Enter the Total numbers :
4
Enter numbers:
101
50
79
10
    enter the element to be found
79
number found
ritik@ritik-virtual-machine:~/Desktop$
```

```
ritik@ritik-virtual-machine:~/Desktop$ ./findnumber.sh
Enter the Total numbers :
4
Enter numbers:
101
50
79
10
    enter the element to be found
11
not found
ritik@ritik-virtual-machine:~/Desktop$
```

## EXPERIMENT NO.6

**AIM:** - Install docker and run hello-world docker image.

**CODE:** -

```
● ● ● + ritikverma@pop-os: ~ Q ...  
ritikverma@pop-os:~$ sudo docker run hello-world  
  
Hello from Docker!  
This message shows that your installation appears to be working correctly.  
  
To generate this message, Docker took the following steps:  
1. The Docker client contacted the Docker daemon.  
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.  
(amd64)  
3. The Docker daemon created a new container from that image which runs the  
executable that produces the output you are currently reading.  
4. The Docker daemon streamed that output to the Docker client, which sent i  
t  
to your terminal.  
  
To try something more ambitious, you can run an Ubuntu container with:  
$ docker run -it ubuntu bash  
  
Share images, automate workflows, and more with a free Docker ID:  
https://hub.docker.com/  
  
For more examples and ideas, visit:  
https://docs.docker.com/get-started/
```

**AIM:** - Run docker start command

**CODE:** -

```
ritikverma@pop-os:~$ sudo docker ps -a  
CONTAINER ID IMAGE COMMAND CREATED STATUS  
PORTS NAMES  
9d5771de0f2f hello-world "/hello" 34 minutes ago Exited (0) 34 minute  
s ago musing_thompson  
75a6383d28a0 hello-world "/hello" 40 minutes ago Exited (0) 40 minute  
s ago admiring_easley  
ritikverma@pop-os:~$ sudo docker start musing_thompson  
musing_thompson  
ritikverma@pop-os:~$ sudo docker stop musing_thompson  
musing_thompson
```

**AIM:** - Run docker stop command

## **CODE: -**

```
ritikverma@pop-os:~$ sudo docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS
         PORTS NAMES
9d5771de0f2f hello-world "/hello" 34 minutes ago Exited (0) 34 minute
s ago          musing_thompson
75a6383d28a0 hello-world "/hello" 40 minutes ago Exited (0) 40 minute
s ago          admiring_easley
ritikverma@pop-os:~$ sudo docker start musing_thompson
musing_thompson
ritikverma@pop-os:~$ sudo docker stop musing_thompson
musing_thompson
```

**AIM:** - Run docker push command

## **CODE: -**

```
ritikverma@pop-os:~$ sudo docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't
have a Docker ID, head over to https://hub.docker.com to create one.
Username: ritik25092000
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

```
ritikverma@pop-os:~$ sudo docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
debian          latest   fe3c5de03486  18 hours ago  124MB
hello-world     latest   d1165f221234  5 months ago  13.3kB
docker/whalesay latest   6b362a9f73eb  6 years ago  247MB
ritikverma@pop-os:~$ sudo docker run -it -d debian
72b7abf05f73cd101f0611fe82c7635163e4780b7912d3f0e48a42e26b0f6d86
ritikverma@pop-os:~$ sudo docker ps -a
CONTAINER ID      IMAGE      COMMAND      CREATED      STATUS
PORTS      NAMES
72b7abf05f73      debian      "bash"      19 seconds ago      Up 16 seconds
                  nostalgic_hermann
88ebed6449dc      docker/whalesay      "cowsay boo"      10 hours ago      Exited (0) 10 h
ours ago          optimistic_morse
9d5771de0f2f      hello-world      "/hello"      11 hours ago      Exited (0) 5 mi
nutes ago         musing_thompson
75a6383d28a0      hello-world      "/hello"      11 hours ago      Exited (0) 11 h
ours ago         admiring_easley
```

```
ritikverma@pop-os:~$ sudo docker exec -it 72b7abf05f73 bash
root@72b7abf05f73:/# ls
bin dev home lib64 mnt proc run srv tmp var
boot etc lib media opt root sbin sys usr
root@72b7abf05f73:/# mkdir test-directory
root@72b7abf05f73:/# ls
bin dev home lib64 mnt proc run srv test-directory usr
boot etc lib media opt root sbin sys tmp var
root@72b7abf05f73:/# exit
exit
```

```
ritikverma@pop-os:~$ sudo docker commit 72b7abf05f73 ritik25092000/image-123:2
sha256:686fe200b857969cbf75e6cc688ca9ce18cd3b9c8986ec65d3717bd53cb5d484
ritikverma@pop-os:~$ sudo docker images
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
ritik25092000/image-123   2        686fe200b857    7 seconds ago  124MB
debian              latest    fe3c5de03486    18 hours ago   124MB
hello-world         latest    d1165f221234    5 months ago   13.3kB
docker/whalesay     latest    6b362a9f73eb    6 years ago    247MB
```

```
ritikverma@pop-os:~$ sudo docker push ritik25092000/image-123:2
The push refers to repository [docker.io/ritik25092000/image-123]
9ebc41d1e88f: Pushed
a881cfa23a78: Mounted from library/debian
2: digest: sha256:d38ebb1f01040eed7714a99872320209cb21a4c343197e98699e9cef4f1960
20 size: 736
```

**AIM:** - Run docker pull command

**CODE:** -

```
ritikverma@pop-os:~$ sudo docker pull docker/whalesay
Using default tag: latest
latest: Pulling from docker/whalesay
Image docker.io/docker/whalesay:latest uses outdated schema1 manifest format.
Please upgrade to a schema2 image for better future compatibility. More information at https://docs.docker.com/registry/spec/deprecated-schema-v1/
e190868d63f8: Pull complete
909cd34c6fd7: Pull complete
0b9bfabab7c1: Pull complete
a3ed95caeb02: Pull complete
00bf65475aba: Pull complete
c57b6bcc83e3: Pull complete
8978f6879e2f: Pull complete
8eed3712d2cf: Pull complete
Digest: sha256:178598e51a26abbc958b8a2e48825c90bc22e641de3d31e18aaf55f3258ba9
3b
Status: Downloaded newer image for docker/whalesay:latest
docker.io/docker/whalesay:latest
```

```
ritikverma@pop-os:~$ sudo docker run docker/whalesay cowsay boo
```



**AIM:** - Run docker logs command

**CODE:** -

```
ritikverma@pop-os:~$ sudo docker logs --details --timestamps musing_thompson
2021-08-17T08:44:15.266478916Z
2021-08-17T08:44:15.266744679Z Hello from Docker!
2021-08-17T08:44:15.266761059Z This message shows that your installation appears to be working correctly.
2021-08-17T08:44:15.266766021Z
2021-08-17T08:44:15.266770059Z To generate this message, Docker took the following steps:
2021-08-17T08:44:15.266774040Z 1. The Docker client contacted the Docker daemon.
2021-08-17T08:44:15.266778539Z 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
2021-08-17T08:44:15.266787674Z (amd64)
2021-08-17T08:44:15.266798180Z 3. The Docker daemon created a new container from that image which runs the
2021-08-17T08:44:15.266816599Z executable that produces the output you are currently reading.
2021-08-17T08:44:15.266820322Z 4. The Docker daemon streamed that output to the Docker client, which sent it
2021-08-17T08:44:15.266823585Z to your terminal.
2021-08-17T08:44:15.266830526Z
2021-08-17T08:44:15.266836085Z To try something more ambitious, you can run an Ubuntu container with:
2021-08-17T08:44:15.266840348Z $ docker run -it ubuntu bash
2021-08-17T08:44:15.266844473Z
2021-08-17T08:44:15.266848063Z Share images, automate workflows, and more with a free Docker ID:
2021-08-17T08:44:15.266851358Z https://hub.docker.com/
2021-08-17T08:44:15.266855104Z
2021-08-17T08:44:15.266859962Z For more examples and ideas, visit:
2021-08-17T08:44:15.266863238Z https://docs.docker.com/get-started/
2021-08-17T08:44:15.266866978Z
2021-08-17T09:20:17.658765927Z
2021-08-17T09:20:17.658801967Z Hello from Docker!
2021-08-17T09:20:17.658806557Z This message shows that your installation appears to be working correctly.
2021-08-17T09:20:17.658809983Z
2021-08-17T09:20:17.658813108Z To generate this message, Docker took the following steps:
2021-08-17T09:20:17.658816255Z 1. The Docker client contacted the Docker daemon.
2021-08-17T09:20:17.658819462Z 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
2021-08-17T09:20:17.658822894Z (amd64)
2021-08-17T09:20:17.658826031Z 3. The Docker daemon created a new container from that image which runs the
2021-08-17T09:20:17.658829208Z executable that produces the output you are currently reading.
2021-08-17T09:20:17.658832311Z 4. The Docker daemon streamed that output to the Docker client, which sent it
```

**AIM:** - Run docker ps command

**CODE:** -

```
ritikverma@pop-os:~$ sudo docker ps
CONTAINER ID   IMAGE      COMMAND   CREATED          STATUS          PORTS     NAMES
72b7abf05f73   debian     "bash"    19 minutes ago   Up 19 minutes   nostalgic_hermann
ritikverma@pop-os:~$
```

**AIM:** - Run docker ps -a command

**CODE:** -

```
ritikverma@pop-os:~$ sudo docker ps -a
CONTAINER ID   IMAGE      COMMAND   CREATED          STATUS          PORTS     NAMES
72b7abf05f73   debian     "bash"    21 minutes ago   Up 21 minutes   nostalgic_hermann
88ebbed6449dc   docker/whalesay   "cowsay boo"  10 hours ago   Exited (0) 10 hours ago
9d5771de0f2f   hello-world   "/hello"   11 hours ago   Exited (0) 26 minutes ago
75a6383d28a0   hello-world   "/hello"   11 hours ago   Exited (0) 11 hours ago
ritikverma@pop-os:~$
```

**AIM:** - create docker hub account

**CODE:** -

The screenshot shows the Docker Hub profile page for the user 'ritik25092000'. At the top, there's a navigation bar with links for 'Explore', 'Repositories', 'Organizations', 'Help', and an 'Upgrade' button. On the right, there's a dropdown menu for the user 'ritik25092000'. Below the header, the user's profile picture is displayed, followed by the username 'ritik25092000' and a link to 'Edit profile'. It also shows the user is a 'Community User' who joined on August 17, 2021. Under the profile section, there are tabs for 'Repositories', 'Starred', and 'Contributed', with 'Repositories' being the active tab. A message indicates 'Displaying 1 of 1 repository'. The repository listed is 'ritik25092000/image-123', created by 'ritik25092000' 14 minutes ago. It has 6 downloads and 0 stars. The repository type is 'Container'.

**AIM:** - Run docker compose command

**CODE:** -

```
ritikverma@pop-os:~$ sudo docker-compose -v
docker-compose version 1.25.5, build unknown
```

```
ritikverma@pop-os:~$ ls
calculator.sh  Documents  Javawork  Music      Practice  snap       Videos
Desktop        Downloads  MacOS     Pictures   Public    Templates
ritikverma@pop-os:~$ cd Desktop
ritikverma@pop-os:~/Desktop$ mkdir DockerComposeFile
ritikverma@pop-os:~/Desktop$ ls
DockerComposeFile
ritikverma@pop-os:~/Desktop$ cd DockerComposeFile/
ritikverma@pop-os:~/Desktop/DockerComposeFile$ touch docker-compose.yml
ritikverma@pop-os:~/Desktop/DockerComposeFile$ ls
docker-compose.yml
ritikverma@pop-os:~/Desktop/DockerComposeFile$ nano docker-compose.yml
ritikverma@pop-os:~/Desktop/DockerComposeFile$ cat docker-compose.yml
services:
  web:
    image: nginx
  database:
    image: redis
```

```

ritikverma@pop-os:~/Desktop/DockerComposeFile$ sudo docker-compose config
services:
  database:
    image: redis
  web:
    image: nginx
version: '3.0'

ritikverma@pop-os:~/Desktop/DockerComposeFile$ sudo docker-compose up -d
Creating network "dockercomposefile_default" with the default driver
Pulling web (nginx:)... latest: Pulling from library/nginx
e1acddbe380c: Pull complete
e21006f71c6f: Pull complete
f3341cc17e58: Pull complete
2a53fa598ee2: Pull complete
12455f71a9b5: Pull complete
b86f2ba62d17: Pull complete
Digest: sha256:d4d8e0b81d80202b6b4a5d9fedc52bffb2664e2c72414f131c9d2888ab5af51
Status: Downloaded newer image for nginx:latest
Pulling database (redis:)... latest: Pulling from library/redis
e1acddbe380c: Already exists
a31098369fcc: Pull complete
4a49b0eba86d: Pull complete
fddf1399efac: Pull complete
5c6658b59b72: Pull complete
0b88638a5b77: Pull complete
Digest: sha256:a759ff3de5b2162d8348b96fff11cdc4623b943a175a7b8ae503c9021657383e
Status: Downloaded newer image for redis:latest
Creating dockercomposefile_web_1 ... done
Creating dockercomposefile_database_1 ... done
ritikverma@pop-os:~/Desktop/DockerComposeFile$ █

```

**AIM:** - Run docker build command

**CODE:** -

```

ritikverma@pop-os:~/Desktop/DockerComposeFile$ cd
ritikverma@pop-os:~$ cd Desktop
ritikverma@pop-os:~/Desktop$ mkdir DockerFiles
ritikverma@pop-os:~/Desktop$ ls
DockerComposeFile DockerFiles
ritikverma@pop-os:~/Desktop$ cd DockerFiles
ritikverma@pop-os:~/Desktop/DockerFiles$ touch Dockerfile
ritikverma@pop-os:~/Desktop/DockerFiles$ nano Dockerfile
ritikverma@pop-os:~/Desktop/DockerFiles$ cat Dockerfile
# getting base image
FROM ubuntu

MAINTAINER ritik verma <automation.varmaritik04@gmail.com>

RUN apt-get update

CMD ["echo", "Hello World..! from my first docker image"]

```

```

ritikverma@pop-os:~/Desktop/DockerFiles$ sudo docker build .
Sending build context to Docker daemon 2.048kB
Step 1/4 : FROM ubuntu
latest: Pulling from library/ubuntu
16ec32c2132b: Pull complete
Digest: sha256:82becede498899ec668628e7cb0ad87b6e1c371cb8a1e597d83a47fac21d6af3
Status: Downloaded newer image for ubuntu:latest
--> 1318b700e415
Step 2/4 : MAINTAINER ritik verma <automation.varmaritik04@gmail.com>
--> Running in 57bb6d2689c1
Removing intermediate container 57bb6d2689c1
--> d45f38850ed9
Step 3/4 : RUN apt-get update
--> Running in 6b50d16a8911
Get:1 http://archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:5 http://archive.ubuntu.com/ubuntu focal/universe amd64 Packages [11.3 MB]
Get:6 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [788 kB]
Get:7 http://archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [177 kB]
Get:8 http://archive.ubuntu.com/ubuntu focal/main amd64 Packages [1275 kB]
Get:9 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [1024 kB]
Get:10 http://archive.ubuntu.com/ubuntu focal/restricted amd64 Packages [33.4 kB]
Get:11 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [1058 kB]
Get:12 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1463 kB]
Get:13 http://archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [522 kB]
Get:14 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [30.6 kB]
Get:15 http://archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [33.8 kB]
Get:16 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [472 kB]
Get:17 http://archive.ubuntu.com/ubuntu focal-backports/universe amd64 Packages [6319 B]
Get:18 http://archive.ubuntu.com/ubuntu focal-backports/main amd64 Packages [2668 B]
Fetched 18.8 MB in 4s (4586 kB/s)
Reading package lists...
Removing intermediate container 6b50d16a8911
--> e9fe0468fcf8
Step 4/4 : CMD ["echo", "Hello World..! from my first docker image"]
--> Running in 3e91f0f6a237
Removing intermediate container 3e91f0f6a237
--> 8904017cb349
Successfully built 8904017cb349

```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
<none>	<none>	8904017cb349	42 seconds ago	103MB
ritik25092000/image-123	2	686fe200b857	50 minutes ago	124MB
redis	latest	ddcca4b8a6f0	6 hours ago	105MB
nginx	latest	dd34e67e3371	9 hours ago	133MB
debian	latest	fe3c5de03486	19 hours ago	124MB
ubuntu	latest	1318b700e415	3 weeks ago	72.8MB
hello-world	latest	d1165f221234	5 months ago	13.3kB
docker/whelesay	latest	6b362a9f73eb	6 years ago	247MB

```

ritikverma@pop-os:~/Desktop/DockerFiles$ sudo docker run 8904017cb349
Hello World..! from my first docker image

```

**AIM:** - Run docker run command

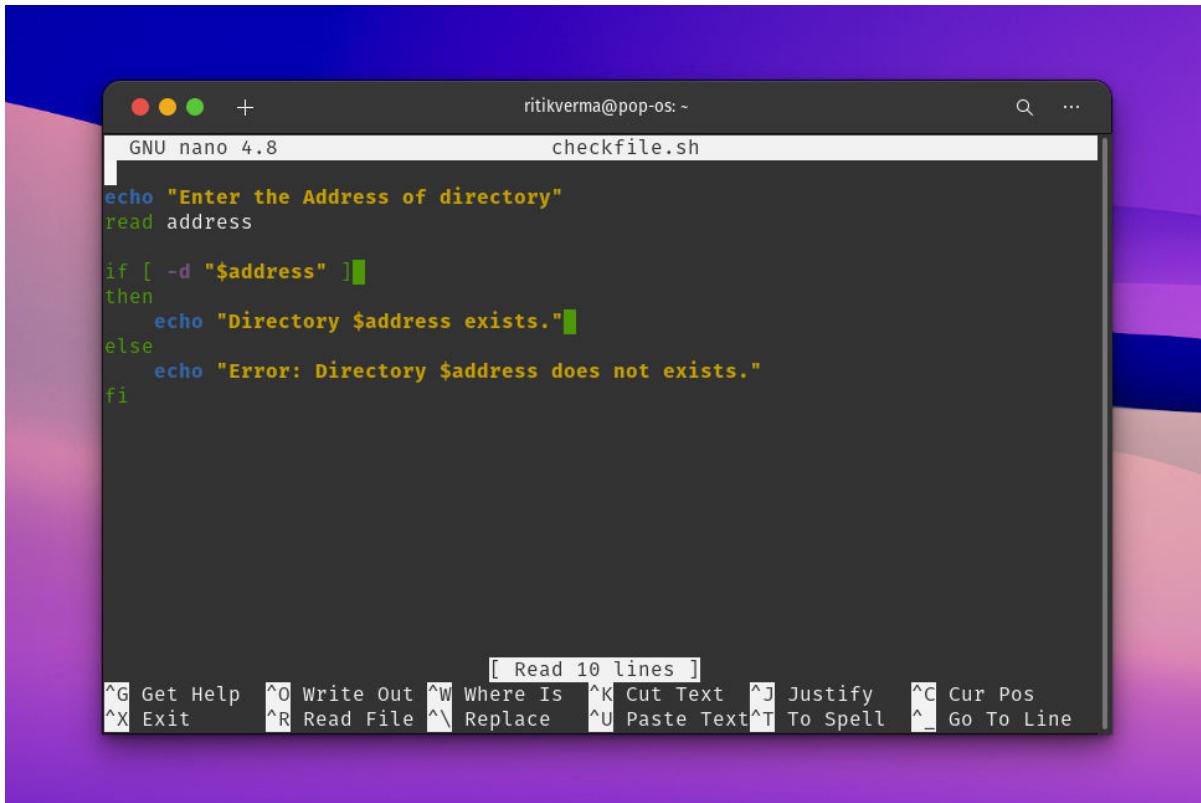
**CODE:** -

```
ritikverma@pop-os:~/Desktop/DockerFiles$ sudo docker run -it ubuntu
root@a463a7915c4f:/# ls
bin  dev  home  lib32  libx32  mnt  proc  run  srv  tmp  var
boot  etc  lib  lib64  media  opt  root  sbin  sys  usr
root@a463a7915c4f:/# █
```

## EXPERIMENT NO.7

AIM: - Shell script program to check whether given file is a directory or not.

CODE: -



The screenshot shows a terminal window titled "checkfile.sh" running on a "GNU nano 4.8" editor. The script contains the following code:

```
echo "Enter the Address of directory"
read address

if [ -d "$address" ]
then
    echo "Directory $address exists."
else
    echo "Error: Directory $address does not exists."
fi
```

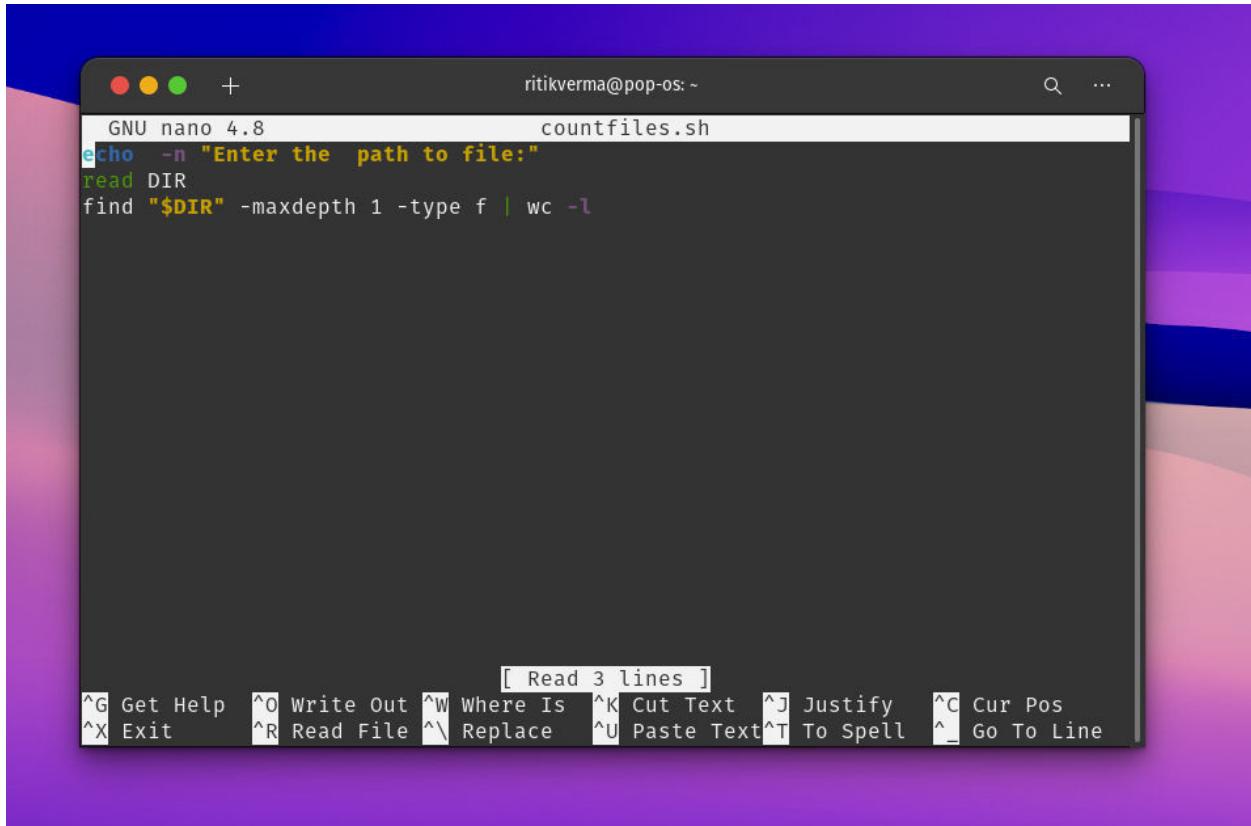
At the bottom of the terminal window, there is a legend for nano editor commands:

[ Read 10 lines ]							
$^G$ Get Help	$^O$ Write Out	$^W$ Where Is	$^K$ Cut Text	$^J$ Justify	$^C$ Cur Pos		
$^X$ Exit	$^R$ Read File	$^{\backslash}$ Replace	$^U$ Paste Text	$^T$ To Spell	$^_$ Go To Line		

```
ritikverma@pop-os:~$ ./checkfile.sh
Enter the Address of directory
/home/ritikverma
Directory /home/ritikverma exists.
ritikverma@pop-os:~$
```

AIM: - Shell script program to count number of files in a Directory.

CODE: -



```
GNU nano 4.8          ritikverma@pop-os: ~
countfiles.sh
echo -n "Enter the path to file:"
read DIR
find "$DIR" -maxdepth 1 -type f | wc -l

[ Read 3 lines ]
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Paste Text^T To Spell  ^_ Go To Line
```

```
ritikverma@pop-os:~$ ./countfiles.sh
Enter the path to file:/home/ritikverma
13
ritikverma@pop-os:~$
```

AIM: - Shell script program to copy contents of one file to another.

CODE: -

```
ritikverma@pop-os: ~
GNU nano 4.8          copy1file2another.sh
echo "enter the filename whose content is to be copied"
read fname

echo "enter the filename to copy the content to"
read nname

cp -v $fname $nname

echo "done"

[ Read 9 lines ]
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Paste Text^T To Spell  ^_ Go To Line
```

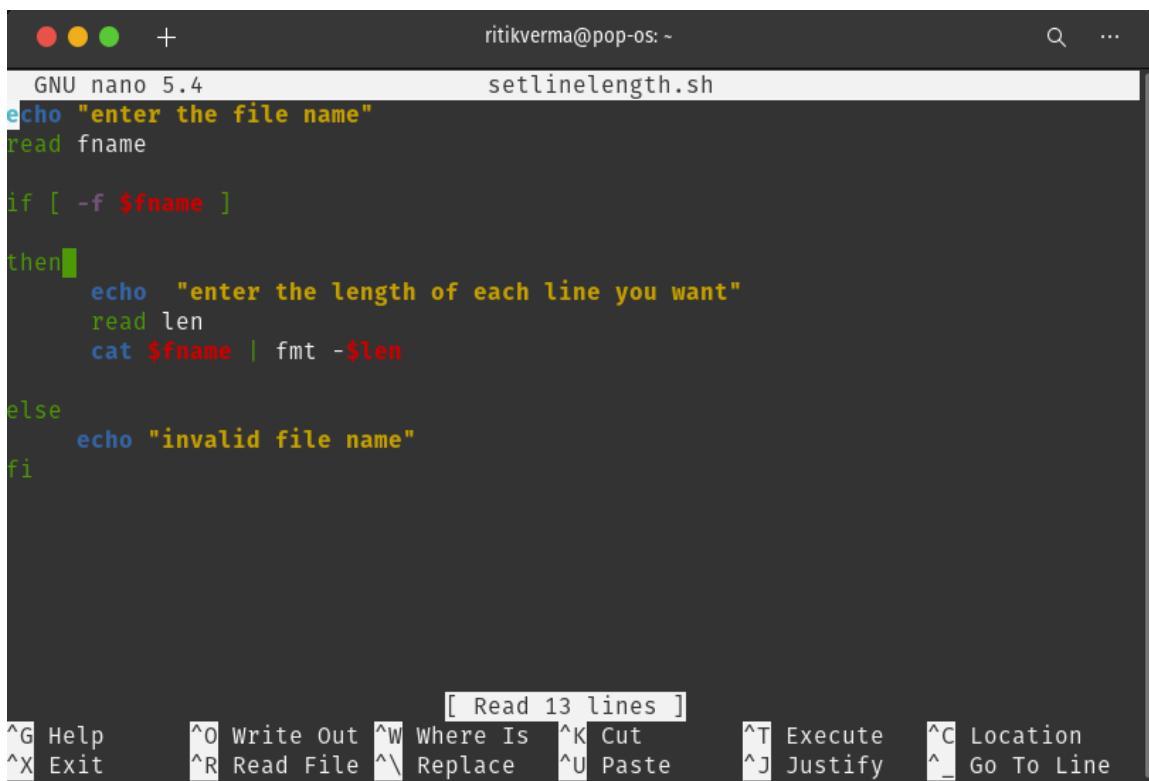
```
ritikverma@pop-os:~$ cat hello.txt
my name is ritik verma
ritikverma@pop-os:~$ cat hello1.txt

ritikverma@pop-os:~$ ./copy1file2another.sh
enter the filename whose content is to be copied
hello.txt
enter the filename to copy the content to
hello1.txt
'hello.txt' -> 'hello1.txt'
done
ritikverma@pop-os:~$ cat hello1.txt
my name is ritik verma
ritikverma@pop-os:~$
```

## EXPERIMENT NO. 8

Aim : - Use a pipeline and command substitution to set the length of a line in file to a variable.

Code : -



```
GNU nano 5.4          ritikverma@pop-os: ~
echo "enter the file name"
read fname

if [ -f $fname ]
then
    echo "enter the length of each line you want"
    read len
    cat $fname | fmt -$len

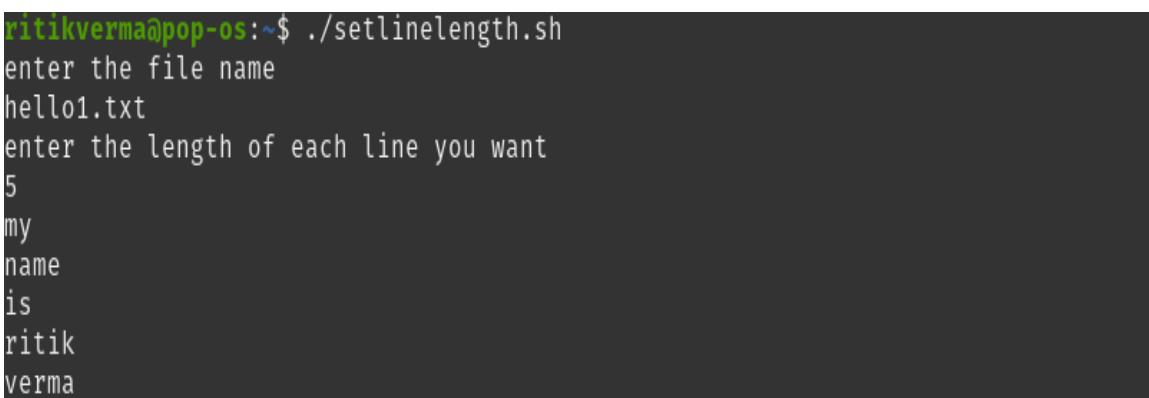
else
    echo "invalid file name"
fi
```

[ Read 13 lines ]

<sup>^G</sup> Help <sup>^O</sup> Write Out <sup>^W</sup> Where Is <sup>^K</sup> Cut <sup>^T</sup> Execute <sup>^C</sup> Location  
<sup>^X</sup> Exit <sup>^R</sup> Read File <sup>^V</sup> Replace <sup>^U</sup> Paste <sup>^J</sup> Justify <sup>^\_</sup> Go To Line



```
ritikverma@pop-os:~$ cat hello1.txt
my name is ritik verma
```



```
ritikverma@pop-os:~$ ./setlinelength.sh
enter the file name
hello1.txt
enter the length of each line you want
5
my
name
is
ritik
verma
```

Aim : - Write a program using sed command to print duplicated lines of Input.

Code : -

The screenshot shows a terminal window titled "removeduplicate.sh" running on a "GNU nano 5.4" editor. The code inside the file is as follows:

```
echo "enter the file name"
read fname
if [ -f $fname ]
then
    sort $fname | sed '$!:N; s/^\\(.*)\\n\\1$/\\1/; t; D'
else
    echo "enter the correct file name"
fi
```

At the bottom of the terminal window, there is a menu bar with various keyboard shortcuts:

- [ Read 13 lines ]
- ^G Help
- ^O Write Out
- ^W Where Is
- ^K Cut
- ^T Execute
- ^C Location
- ^X Exit
- ^R Read File
- ^\\ Replace
- ^U Paste
- ^J Justify
- ^\_ Go To Line

Output : -

```
ritikverma@pop-os:~$ cat test.txt
hi
hello
hi
how
hello
.
```

```
ritikverma@pop-os:~$ ./removeduplicate.sh
enter the file name
test.txt
hello
hi
```

## EXPERIMENT 9

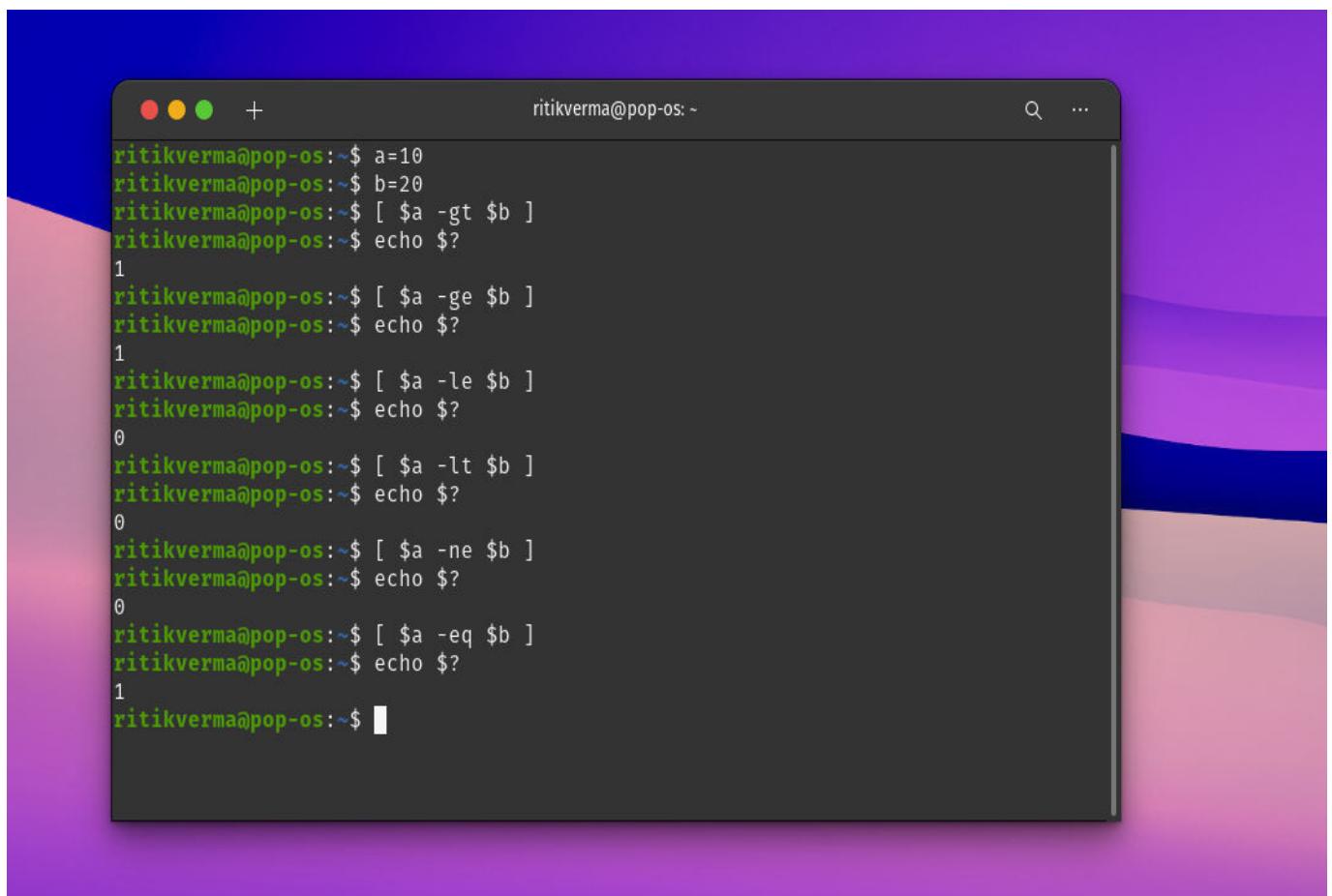
**AIM:** - Write a program to illustrate the implementation of following:

### 1. INTEGER COMPARISON

#### THEORY

1. num1 -eq num2 check if 1<sup>st</sup> number is equal to 2nd number
2. num1 -ge num2 checks if 1<sup>st</sup> number is greater than or equal to 2nd number
3. num1 -gt num2 checks if 1<sup>st</sup> number is greater than 2nd number
4. num1 -le num2 checks if 1<sup>st</sup> number is less than or equal to 2nd number
5. num1 -lt num2 checks if 1<sup>st</sup> number is less than 2nd number
6. num1 -ne num2 checks if 1<sup>st</sup> number is not equal to 2nd number

#### CODE



The screenshot shows a terminal window titled "ritikverma@pop-os:~". The terminal displays the following sequence of commands and their outputs:

```
ritikverma@pop-os:~$ a=10
ritikverma@pop-os:~$ b=20
ritikverma@pop-os:~$ [ $a -gt $b ]
ritikverma@pop-os:~$ echo $?
1
ritikverma@pop-os:~$ [ $a -ge $b ]
ritikverma@pop-os:~$ echo $?
1
ritikverma@pop-os:~$ [ $a -le $b ]
ritikverma@pop-os:~$ echo $?
0
ritikverma@pop-os:~$ [ $a -lt $b ]
ritikverma@pop-os:~$ echo $?
0
ritikverma@pop-os:~$ [ $a -ne $b ]
ritikverma@pop-os:~$ echo $?
0
ritikverma@pop-os:~$ [ $a -eq $b ]
ritikverma@pop-os:~$ echo $?
1
ritikverma@pop-os:~$
```

## 2. STRING COMPARISON

### THEORY

1. var1 = var2 checks if var1 is the same as string var2
2. var1 != var2 checks if var1 is not the same as var2
3. var1 < var2 checks if var1 is less than var2
4. var1 > var2 checks if var1 is greater than var2
5. -n var1 checks if var1 has a length greater than zero
6. -z var1 checks if var1 has a length of zero

### CODE

```
ritikverma@pop-os:~$ s1="hello"
ritikverma@pop-os:~$ s2="hello world"
```

```
ritikverma@pop-os:~$ [ "$s1" = "$s2" ]
ritikverma@pop-os:~$ echo $?
1
ritikverma@pop-os:~$ [ "$s1" != "$s2" ]
ritikverma@pop-os:~$ echo $?
0
```

```
ritikverma@pop-os:~$ [ "$s1" \> "$s2" ]
ritikverma@pop-os:~$ echo $?
1
```

```
ritikverma@pop-os:~$ [ "$s1" \< "$s2" ]
ritikverma@pop-os:~$ echo $?
0
ritikverma@pop-os:~$ [ -n "$s1" ]
ritikverma@pop-os:~$ echo $?
0
ritikverma@pop-os:~$ [ -z "$s1" ]
ritikverma@pop-os:~$ echo $?
1
ritikverma@pop-os:~$ █
```

### 3. LOGICAL OPERATORS

#### THEORY

1. **OR** is used between two or multiple conditions. It returns true if any one of conditions returns as true. First condition is always checked but the second condition is checked only if first condition is returned false
2. **AND** is used between two or multiple conditions. It returns true only if all the conditions returns as true. First condition is always checked but the second condition is checked only if first condition is returned true.

#### CODE

```
ritikverma@pop-os:~$ c=10
ritikverma@pop-os:~$ d=20
ritikverma@pop-os:~$ [ "$c" -le 20 ] && [ "$d" -gt 10 ]
ritikverma@pop-os:~$ echo $?
0
ritikverma@pop-os:~$ 
```

```
ritikverma@pop-os:~$ [ "$c" -le 5 ] || [ "$d" -gt 25 ]
ritikverma@pop-os:~$ echo $?
1
ritikverma@pop-os:~$ 
```

## 4. FILE TESTS

### THEORY

1. -d file checks if the file exists and is it's a directory
2. -e file checks if the file exists on system
3. -w file checks if the file exists on system and if it is writable
4. -r file checks if the file exists on system and it is readable
5. -s file checks if the file exists on system and it is not empty
6. -f file checks if the file exists on system and it is a file
7. -O file checks if the file exists on system and if it's is owned by the current user
8. -G file checks if the file exists and the default group is the same as the current user
9. -x file checks if the file exists on system and is executable
10. file A -nt file B checks if file A is newer than file B
11. file A -ot file B checks if file A is older than file B

### CODE

```
ritikverma@pop-os:~$ dir=/home/ritikverma/Javawork
ritikverma@pop-os:~$ [ -d "$dir" ]
ritikverma@pop-os:~$ echo $?
0
ritikverma@pop-os:~$ [ -e "$dir" ]
ritikverma@pop-os:~$ echo $?
0
ritikverma@pop-os:~$ [ -w "$dir" ]
ritikverma@pop-os:~$ echo $?
0
ritikverma@pop-os:~$ [ -r "$dir" ]
ritikverma@pop-os:~$ echo $?
0
ritikverma@pop-os:~$ [ -s "$dir" ]
ritikverma@pop-os:~$ echo $?
0
ritikverma@pop-os:~$ [ -f "$dir" ]
ritikverma@pop-os:~$ echo $?
1
ritikverma@pop-os:~$ [ -O "$dir" ]
ritikverma@pop-os:~$ echo $?
0
ritikverma@pop-os:~$ [ -G "$dir" ]
ritikverma@pop-os:~$ echo $?
0
ritikverma@pop-os:~$ [ -x "$dir" ]
ritikverma@pop-os:~$ echo $?
0
```

```

ritikverma@pop-os:~$ ls
ai_files           Downloads          removeduplicate.shc
b.sh               hello1.txt        ritikverma
calculator.sh      hello.txt         setlinelength.sh
checkfile.sh       intcomparison.sh setlinelength.sh.save
checkfile.sh.save Javawork          snap
copy1file2another.sh MacOS            Templates
countfiles.sh     Music             test
countfiles.sh.save Pictures          test.txt
Desktop           Practice          test.txt.save
directory.sh      Public            Videos
directory.sh.save PycharmProjects
Documents          removeduplicate.sh

ritikverma@pop-os:~$ [ hello1.txt -nt hello.txt ]
ritikverma@pop-os:~$ echo $?
0
ritikverma@pop-os:~$ [ hello1.txt -ot hello.txt ]
ritikverma@pop-os:~$ echo $?
1
ritikverma@pop-os:~$ 

```

## 5. CONDITIONAL CONTROL STRUCTURE

### THEORY

Condition Control Structures: <i>if, else, elif, case</i>	Function
<b>if</b> <i>command then</i> <i>command</i> <i>fi</i>	<b>if</b> executes an action if its test command is true.
<b>if</b> <i>command then</i> <i>command</i> <b>else</b> <i>command</i> <i>fi</i>	<b>if-else</b> executes an action if the exit status of its test command is true; if false, then the <b>else</b> action is executed.
<b>if</b> <i>command then</i> <i>command</i> <b>elif</b> <i>command then</i> <i>command</i> <b>else</b> <i>command</i> <i>fi</i>	<b>elif</b> allows you to nest <b>if</b> structures, enabling selection among several alternatives; at the first true <b>if</b> structure, its commands are executed and control leaves the entire <b>elif</b> structure.
<b>case</b> <i>string in</i> <b>pattern)</b> <i>command;;</i> <b>esac</b>	<b>case</b> matches the string value to any of several patterns; if a pattern is matched, its associated commands are executed.
<i>command &amp;&amp;</i>	The logical AND condition returns a true 0 value if both commands return a

<i>command</i>	true 0 value; if one returns a non-zero value, then the AND condition is false and also returns a non-zero value.
<i>command</i>    <i>command</i>	The logical OR condition returns a true 0 value if one or the other command returns a true 0 value; if both commands return a non-zero value, then the OR condition is false and also returns a non-zero value.
! <i>command</i>	The logical NOT condition inverts the return value of the command.

## CODE



```
GNU nano 5.4                                     conditionalcontrol.sh
echo Enter s to list file sizes,
echo      otherwise all file information is listed.
echo -n "Please enter option: "
read choice
if [ "$choice" = s ]
then
    ls -s
else
    ls -l
fi
echo Good-bye
```

```
ritikverma@pop-os:~$ nano conditionalcontrol.sh
ritikverma@pop-os:~$ chmod +x conditionalcontrol.sh
ritikverma@pop-os:~$ ./conditionalcontrol.sh
Enter s to list file sizes,
otherwise all file information is listed.
Please enter option: s
total 152
4 ai_files           4 Downloads          4 PycharmProjects
4 b.sh                4 forloop.sh        4 removeduplicate.sh
4 calculator.sh       4 hello1.txt        4 removeduplicate.shc
4 checkfile.sh        4 hello.txt         4 ritikverma
4 checkfile.sh.save   4 intcomparison.sh  4 setlinelength.sh
4 conditionalcontrol.sh 4 intcomparison.sh.save 4 setlinelength.sh.save
4 copy1file2another.sh 4 Javawork          4 snap
4 countfiles.sh       4 MacOS              4 Templates
4 countfiles.sh.save  4 Music              4 test
4 Desktop              4 pattern.sh        4 test.txt
4 directory.sh        4 Pictures           4 test.txt.save
4 directory.sh.save   4 Practice           4 Videos
4 Documents            4 Public
```

Good-bye

## 6. LOOP CONTROL STRUCTURES

### THEORY

Loop Control Structures: <b>while, until, for, for-in, select</b>	Function
<b>while command</b> do command done	<b>while</b> executes an action as long as its test command is true.
<b>until command</b> do command done	<b>until</b> executes an action as long as its test command is false.
<b>for variable in list-values</b> do command done	<b>for-in</b> is designed for use with lists of values; the variable operand is consecutively assigned the values in the list.
<b>for variable</b> do command done	<b>for</b> is designed for reference script arguments; the variable operand is consecutively assigned each argument value.
<b>select string in item-list</b> do command done	<b>select</b> creates a menu based on the items in the <i>item-list</i> ; then it executes the command; the command is usually a <b>case</b> .

### Loop control structures

```
● ● ● +                                         ritikverma@pop-os: ~
GNU nano 5.4                                         pattern.sh
for((i=1 ; i<=5 ; i++))                         pattern.sh
do
    j=$i
    while [ $j -gt 0 ]
    do
        echo -e "$i \c"
        j=$(( $j - 1 ))
    done
    echo -e "\n"
done
```

```
ritikverma@pop-os:~$ nano pattern.sh
ritikverma@pop-os:~$ ./pattern.sh
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

# EXPERIMENT 10

**AIM:** - Write a shell script to print a number in reverse order.

**CODE:** -

```
echo "1. a number in reverse order."
echo "2.find the sum of all numbers in a file in Linux."
echo "3.find out the unique words in a file and also count the occurrence of each of these words."
echo "4.reverse the list of strings and reverse each string further in the list."

echo "Enter your choice"
read ch

if [ $ch -eq 1 ]
then
    echo "enter a number"
    read n
    sd=0
    rev=0
    while [ $n -gt 0 ]
    do
        sd=$(( $n % 10 ))
        rev=$(( $rev * 10 + $sd ))
        n=$(( $n / 10 ))
    done
    echo "Reverse of given number is $rev"
fi
```

## OUTPUT

```
ritikverma@pop-os:~/Documents$ ./reverseNo.sh
1. a number in reverse order.
2.find the sum of all numbers in a file in Linux.
3.find out the unique words in a file and also count the occurrence of each of these words. We can say that the file under consideration contains many lines, and each line has multiple words.
4.reverse the list of strings and reverse each string further in the list.
Enter your choice
1
enter a number
123
Reverse of given number is 321
ritikverma@pop-os:~/Documents$
```

AIM: - Write a shell script to reverse the list of strings and reverse each string further in the list.

CODE: -

```
if [ $ch -eq 4 ]
then
    read -p "Filename: " f
    if [[ -f "$f" ]]; then
        cat $f
        while IFS="\n" read -r line; do
            while IFS=" " read -a arr; do
                for i in "${arr[@]}";do
                    echo -n ${i} " " | rev >> delete
                done
            done <<< $line
            echo >> delete
        done < $f
        echo
        mv delete $f
        cat $f
    else
        echo "No such file: $f"
    fi
fi
```

## OUTPUT

```
ritikverma@pop-os:~$ cat reversfile.txt
Ritik Verma from Amity
```

```
ritikverma@pop-os:~$
```

```
ritikverma@pop-os:~$ ./reverseNo.sh
1. a number in reverse order.
2.find the sum of all numbers in a file in Linux.
3.find out the unique words in a file and also count the occurrence of each of these words. We
can say that the file under consideration contains many lines, and each line has multiple wor
ds.
```

```
4.reverse the list of strings and reverse each string further in the list.
```

```
Enter your choice
```

```
4
```

```
Filename: reversfile.txt
Ritik Verma from Amity
```

```
    kitiT amreV morf ytimA
```

```
ritikverma@pop-os:~$
```

**AIM:** - Write a shell script to find the sum of all numbers in a file in Linux?

**CODE:** -

```
if [ $ch -eq 2 ]
then
    echo "Enter the file name"
    read file
    SUM=0
    while read LINE
    do
        SUM=$(expr $SUM + $LINE)
    done < $file.txt
    echo "Sum of all numbers in file is $SUM"
fi
```

**OUTPUT**

```
ritikverma@pop-os:~$ cat addnumbers.txt
1
2
3
4
5
10
ritikverma@pop-os:~$
```

```
ritikverma@pop-os:~$ ls
addnumbers.txt          directory.sh      pattern.sh       ritikverma
ai_files                directory.sh.save  Pictures         setlinelength.sh
b.sh                     Documents        Practice        setlinelength.sh.save
calculator.sh            Downloads        Public         snap
checkfile.sh             forloop.sh      PycharmProjects Templates
checkfile.sh.save        hello1.txt     removeduplicate.sh test
conditionalcontrol.sh    hello.txt      removeduplicate.shc test.txt
conditionalcontrol.sh.save intcomparison.sh reverseNo.sh   test.txt.save
copy1file2another.sh    intcomparison.sh.save reverseNo.sh.save  Videos
countfiles.sh            Javawork        reverseNo.sh.save.1 wordcountfile.txt
countfiles.sh.save       MacOS          reverseNo.sh.save.2
Desktop                 Music          reversefile.txt
ritikverma@pop-os:~$ ./reverseNo.sh
1. a number in reverse order.
2. find the sum of all numbers in a file in Linux.
3. find out the unique words in a file and also count the occurrence of each of these words. We can say that the file under consideration contains many lines, and each line has multiple words.
4. reverse the list of strings and reverse each string further in the list.
Enter your choice
2
Enter the file name
addnumbers
Sum of all numbers in file is 25
ritikverma@pop-os:~$
```

**AIM:** - Write a shell script to find out the unique words in a file and also count the occurrence of each of these words. We can say that the file under consideration contains many lines, and each line has multiple words.

**CODE:** -

```
if [ $ch -eq 3 ]
then
    echo "Enter the filename"
    read filename
    echo "$( awk '{for(i=1;i<=NF;i++)a[$i]++;}END{for(i in a){print i, a[i];}}' $filename.txt )"
fi
```

**OUTPUT**

```
ritikverma@pop-os:~$ ./reverseNo.sh
1. a number in reverse order.
2.find the sum of all numbers in a file in Linux.
3.find out the unique words in a file and also count the occurrence of each of these words. We
can say that the file under consideration contains many lines, and each line has multiple wor
ds.
4.reverse the list of strings and reverse each string further in the list.
Enter your choice
3
Enter the filename
wordcountfile
this 3
multiple 1
file 6
'file' 1
are 1
occurrences 1
is 2
the 1
there 1
in 3
This 1
to 1
test 3
of 1
a 1
used 1
some 1
count 1
lines 1
word 2
ritikverma@pop-os:~$
```

# EXPERIMENT 11

## CASE STUDY

**AIM:** - Write a case study and shell script with real world usage.

**THEORY:** - current weather of any city using OpenWeathermap API in Python.

The OpenWeatherMap is a service that provides weather data, including current weather data, forecasts, and historical data to the developers of web services and mobile applications.

After writing the code in python executed the python code from shell script created a shell script which goes to the required directory where the python code is present using cd command and after going into the desired directory it executes the python code by running the command “python3 program\_name.py”

**CODE:** -

```
GNU nano 5.4
# Python program to find current
# weather details of any city
# using openweathermap api

# import required modules
import requests, json

# Enter your API key here
api_key = "9115bd1ef05bd48279b4610b37c66ff2"

# base_url variable to store url
base_url = "http://api.openweathermap.org/data/2.5/weather?"

# Give city name
city_name = input("Enter city name : ")

# complete_url variable to store
# complete url address
complete_url = base_url + "q=" + city_name + "&appid=" + api_key

# get method of requests module
# return response object
response = requests.get(complete_url)

# json method of response object
# convert json format data into
# python format data
x = response.json()

# Now x contains list of nested dictionaries
# Check the value of "cod" key is equal to
# "404", means city is found otherwise,
# city is not found
if x["cod"] != "404":

    # store the value of "main"
    # key in variable y
    y = x["main"]

    # store the value corresponding
    # to the "temp" key of y
```

```
# store the value corresponding
# to the "temp" key of y
current_temperature = y["temp"]

# store the value corresponding
# to the "pressure" key of y
current_pressure = y["pressure"]

# store the value corresponding
# to the "humidity" key of y
current_humidity = y["humidity"]

# store the value of "weather"
# key in variable z
z = x["weather"]

# store the value corresponding
# to the "description" key at
# the 0th index of z
weather_description = z[0]["description"]

# print following values
print(" Temperature (in celius unit) = " +
      str(round(current_temperature-273.15,2)) +
      "\n atmospheric pressure (in hPa unit) = " +
      str(current_pressure) +
      "\n humidity (in percentage) = " +
      str(current_humidity) +
      "\n description = " +
      str(weather_description))

else:
    print(" City Not Found ")
```

```
GNU nano 5.4
cd PycharmProjects/CurrentWeather
python3 main.py
```

## OUTPUT

```
ritikverma@pop-os:~$ ./weatherinfo.sh
Enter city name : Delhi
Temperature (in celius unit) = 32.05
atmospheric pressure (in hPa unit) = 1006
humidity (in percentage) = 70
description = haze
ritikverma@pop-os:~$ ./weatherinfo.sh
Enter city name : Mumbai
Temperature (in celius unit) = 25.99
atmospheric pressure (in hPa unit) = 1002
humidity (in percentage) = 100
description = light rain
ritikverma@pop-os:~$ █
```

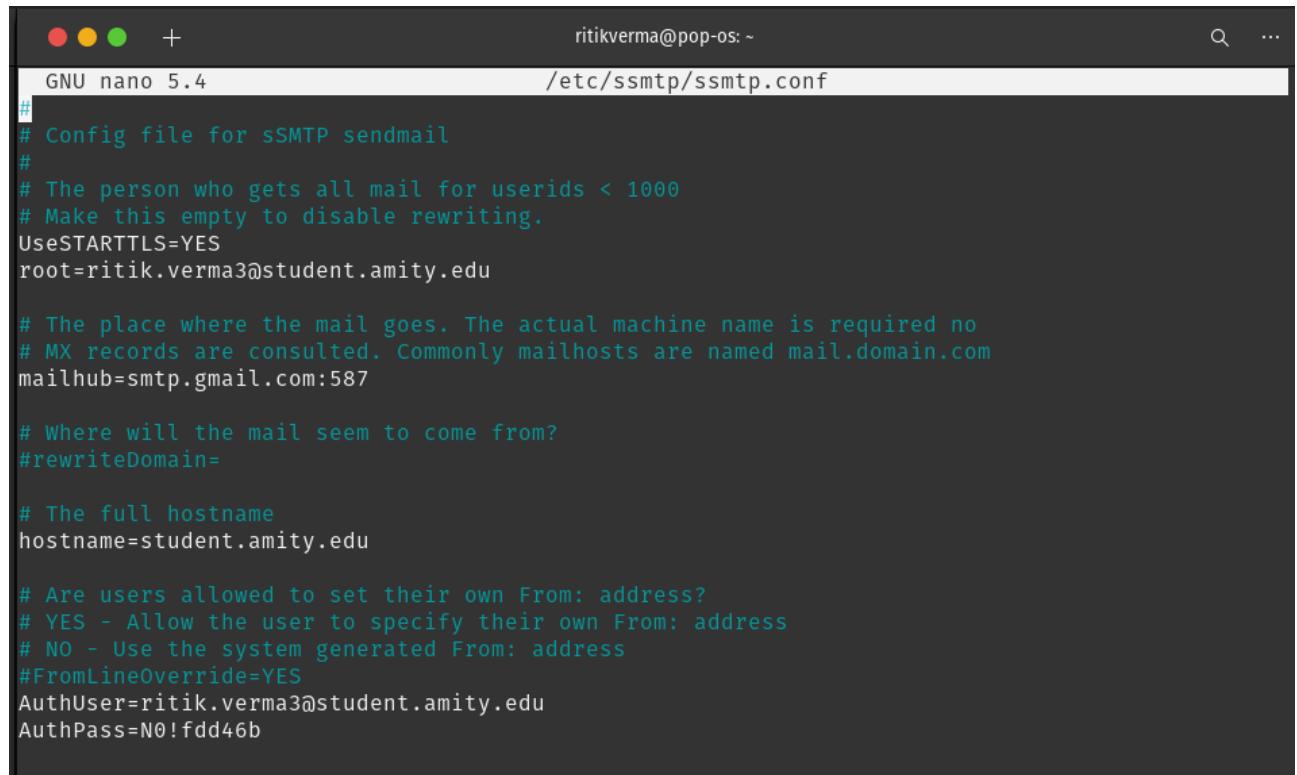


# EXPERIMENT 12

**AIM:** - Design and develop a “Birthday Reminder” that can automatically send birthday wishes with a personalized message via email.

**CODE:** -

## Configure SMTP server for gmail



The screenshot shows a terminal window titled "ritikverma@pop-os: ~". The command "GNU nano 5.4" is running, and the file "/etc/ssmtp/ssmtp.conf" is open. The configuration file contains the following content:

```
# Config file for sSMTP sendmail
#
# The person who gets all mail for userids < 1000
# Make this empty to disable rewriting.
UseSTARTTLS=YES
root=ritik.verma3@student.amity.edu

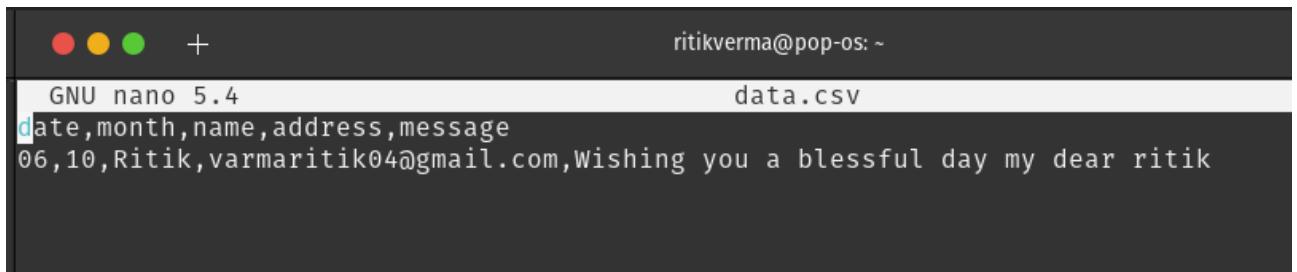
# The place where the mail goes. The actual machine name is required no
# MX records are consulted. Commonly mailhosts are named mail.domain.com
mailhub=smtp.gmail.com:587

# Where will the mail seem to come from?
#rewriteDomain=

# The full hostname
hostname=student.amity.edu

# Are users allowed to set their own From: address?
# YES - Allow the user to specify their own From: address
# NO - Use the system generated From: address
#FromLineOverride=YES
AuthUser=ritik.verma3@student.amity.edu
AuthPass=N0!fdd46b
```

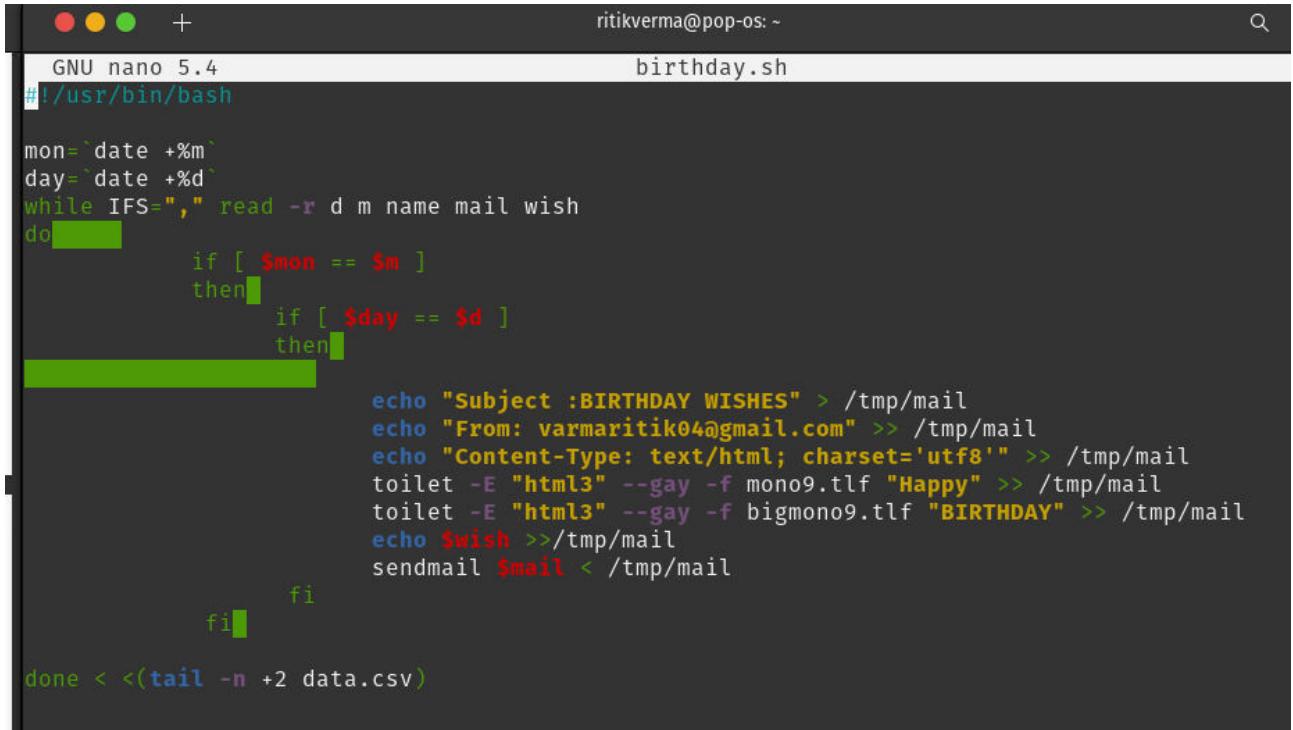
## Wrote a CSV file with birthday entries



The screenshot shows a terminal window titled "ritikverma@pop-os: ~". The command "GNU nano 5.4" is running, and the file "data.csv" is open. The CSV file contains the following data:

date	month	name	address	message
06	10	Ritik	varmaritik04@gmail.com	Wishing you a blessed day my dear ritik

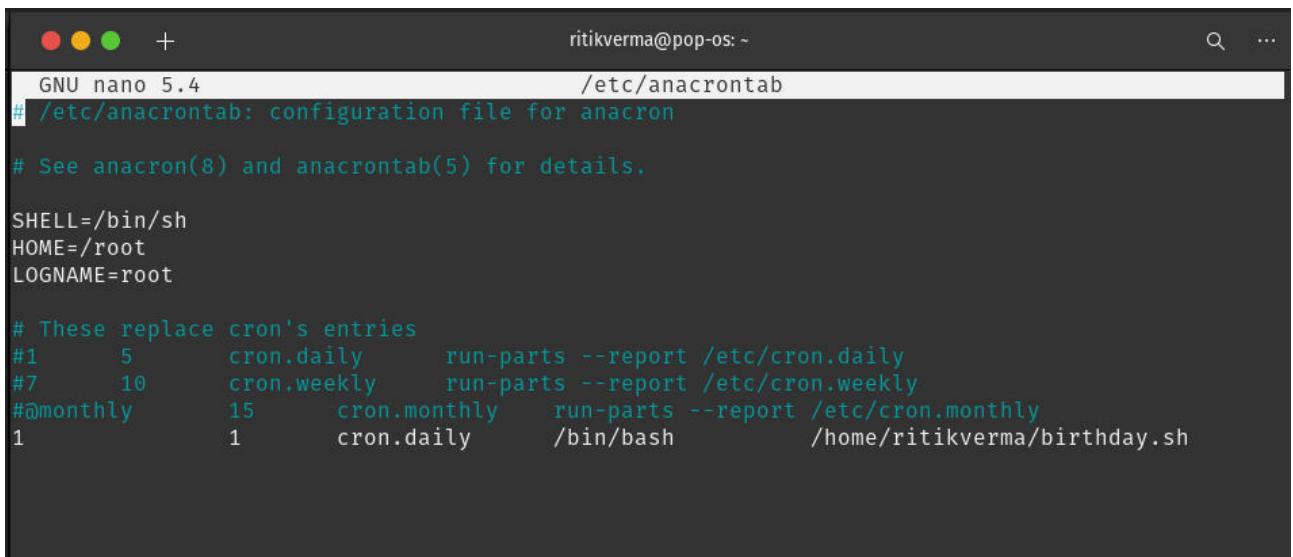
## Shell script to read from CSV file and send birthday wishes



```
ritikverma@pop-os: ~
GNU nano 5.4                                birthday.sh
#!/usr/bin/bash

mon=`date +%m`
day=`date +%d`
while IFS=',' read -r d m name mail wish
do
    if [ $mon == $m ]
    then
        if [ $day == $d ]
        then
            echo "Subject :BIRTHDAY WISHES" > /tmp/mail
            echo "From: varmaritik04@gmail.com" >> /tmp/mail
            echo "Content-Type: text/html; charset='utf8'" >> /tmp/mail
            toilet -E "html3" --gay -f mono9.tlf "Happy" >> /tmp/mail
            toilet -E "html3" --gay -f bigmono9.tlf "BIRTHDAY" >> /tmp/mail
            echo $wish >/tmp/mail
            sendmail $mail < /tmp/mail
        fi
    fi
done < <(tail -n +2 data.csv)
```

## Setup anacron for automatically running daily



```
ritikverma@pop-os: ~
GNU nano 5.4                                /etc/anacrontab
# /etc/anacrontab: configuration file for anacron

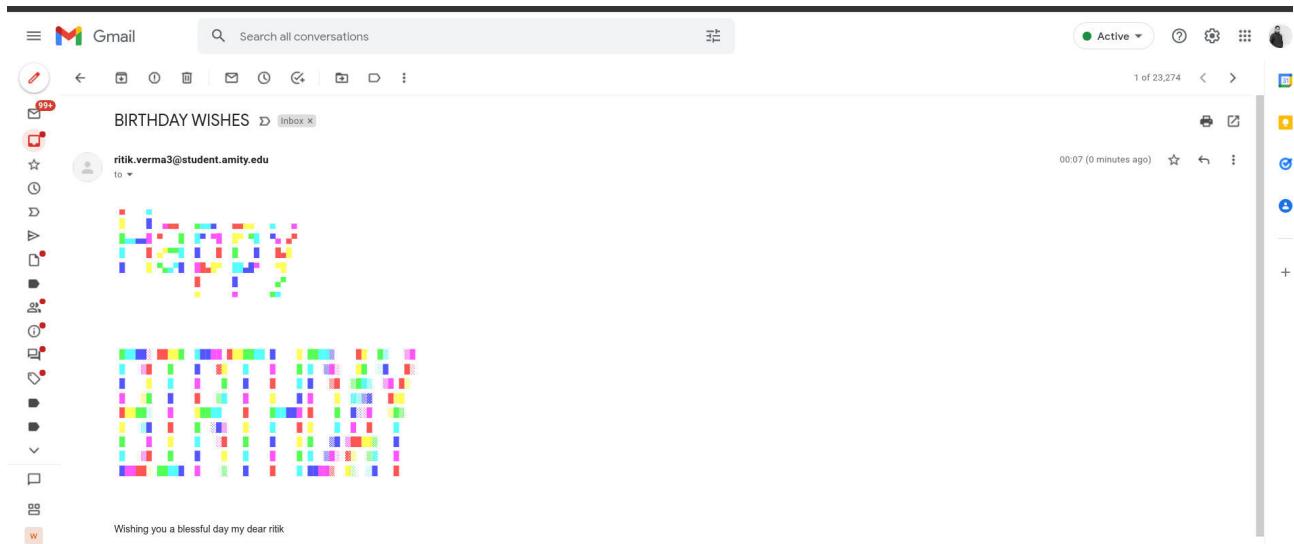
# See anacron(8) and anacrontab(5) for details.

SHELL=/bin/sh
HOME=/root
LOGNAME=root

# These replace cron's entries
#1      5      cron.daily      run-parts --report /etc/cron.daily
#7     10      cron.weekly     run-parts --report /etc/cron.weekly
#@monthly   15      cron.monthly   run-parts --report /etc/cron.monthly
1          1      cron.daily      /bin/bash           /home/ritikverma/birthday.sh
```

## OUTPUT: -

### Received Mail



# EXPERIMENT 13

**AIM:** - Implementation of NIC bonding and Team

## THEORY:

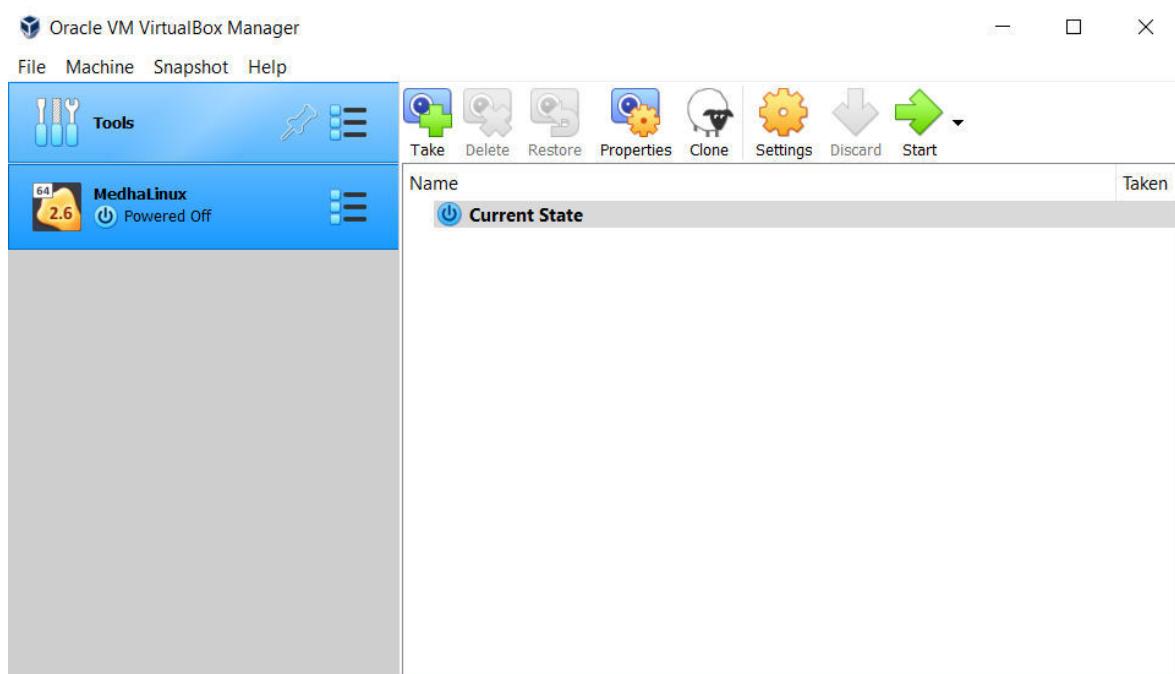
NIC Teaming uses one of two methods, failover, and load-balancing with fail over. With a team you do not get a single 2 GB connection (with two 1 GB NICs). You get two pipes that act as one, but merely are load balancing the traffic over each NIC, and each NIC acts as a failure over to the other.

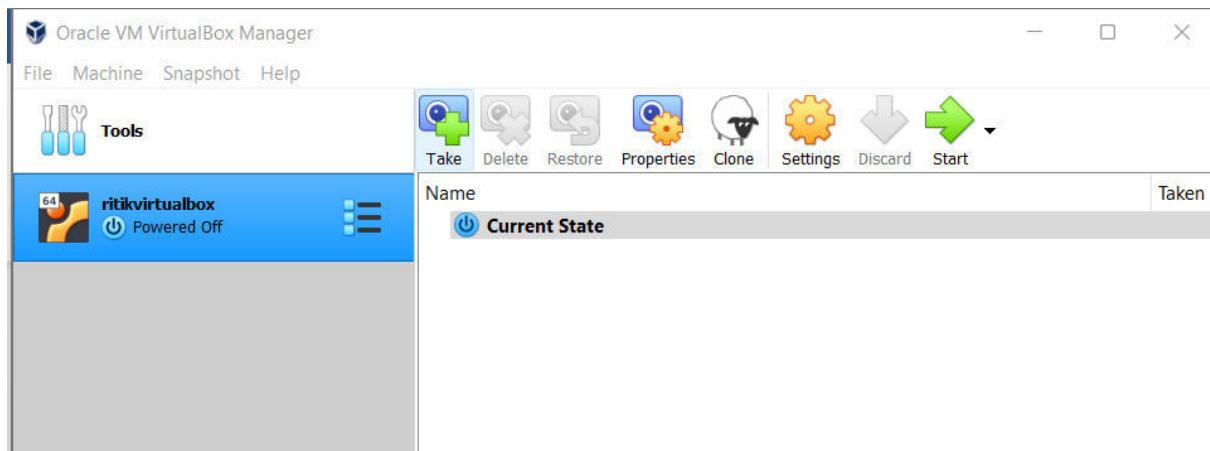
NIC Bonding refers to create a bond or aggregation of two or more links into one in order to provide failover and redundancy to the network.

Nmtui provides a text user interface.

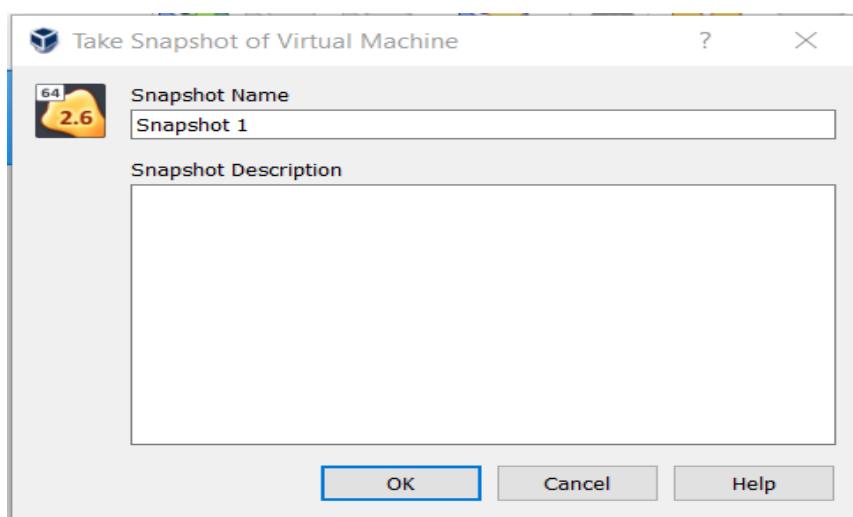
## PROCEDURE:

Step 1: Click on Take Snapshot option.

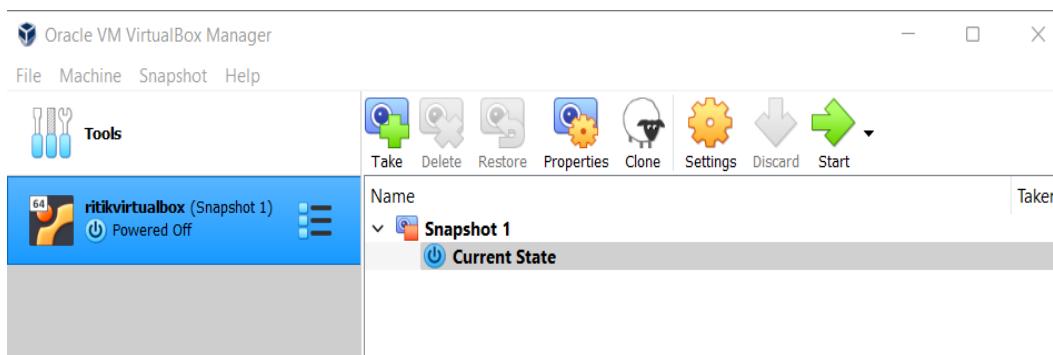
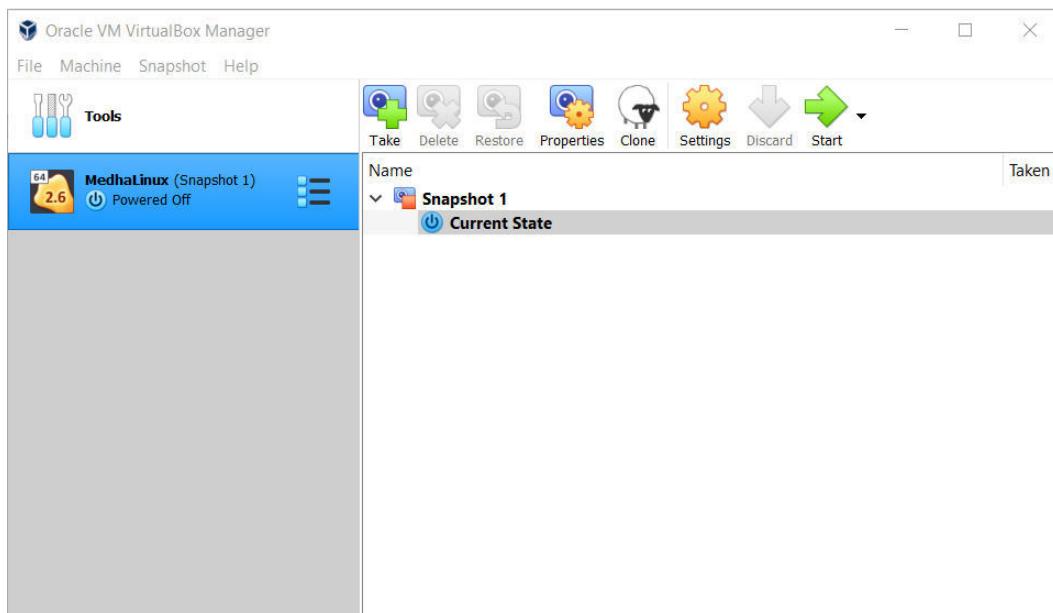




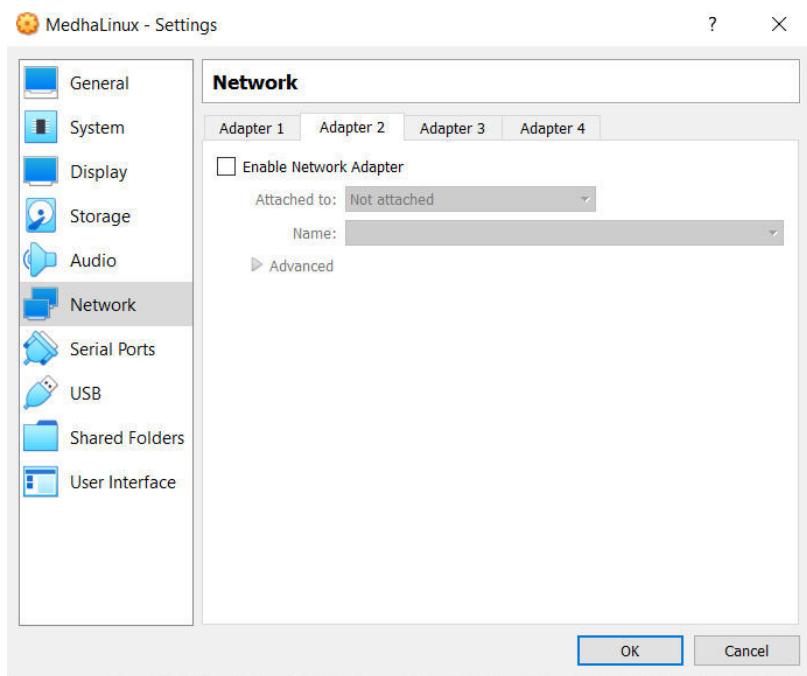
Step 2: Provide the snapshot name and click on OK.



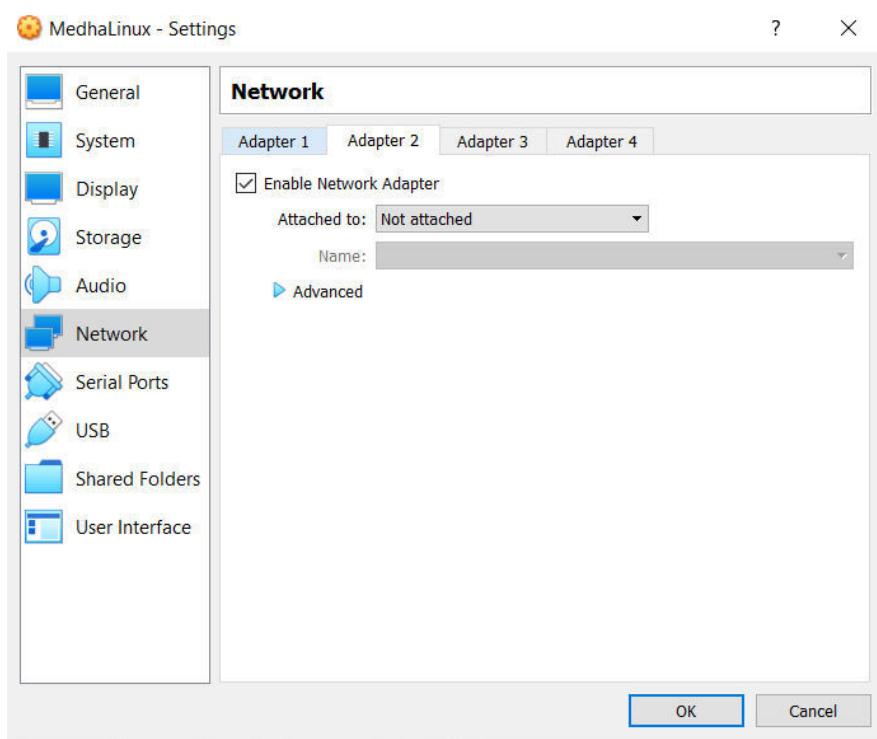
Step 3: The screen would now look like this.

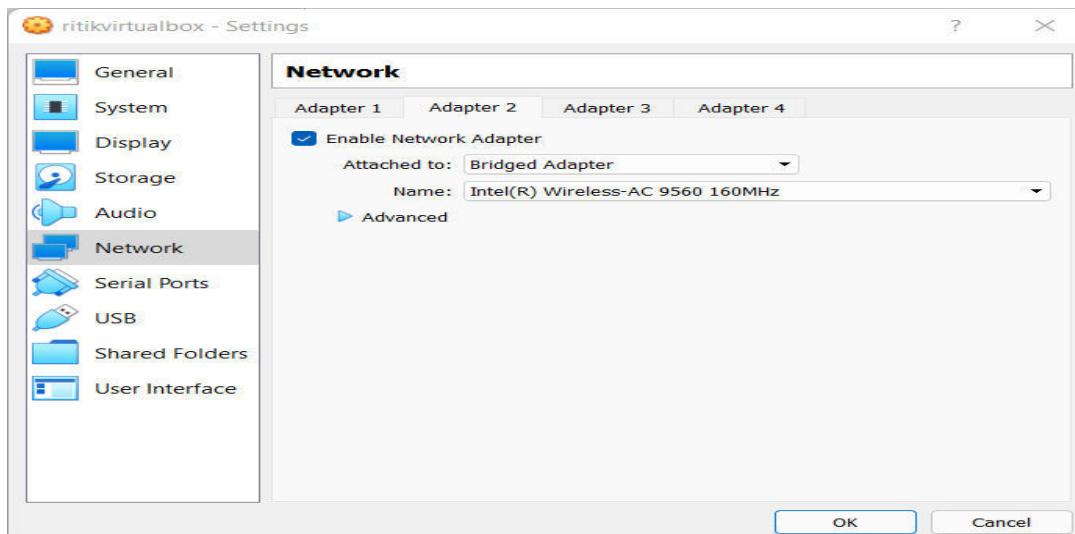


Step 4: Go to Settings > Network > Adapter 2

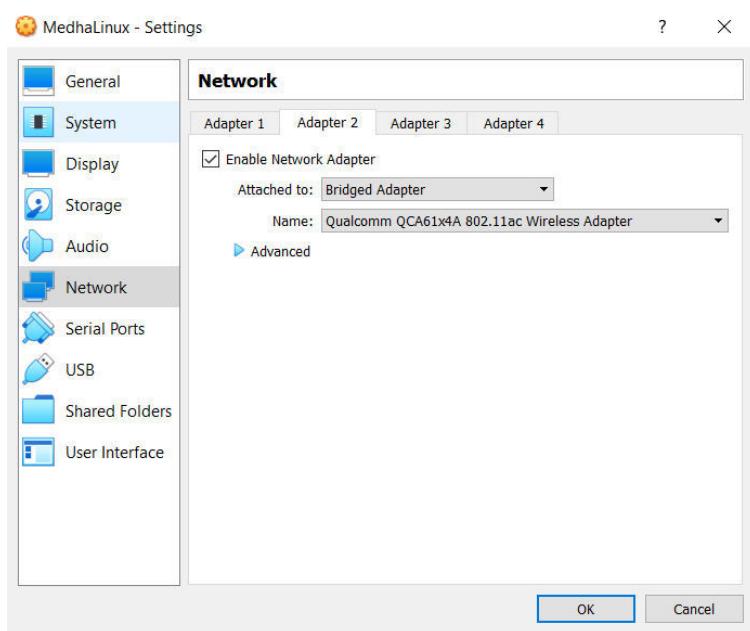


Step 5: Click on Enable Network Adapter.

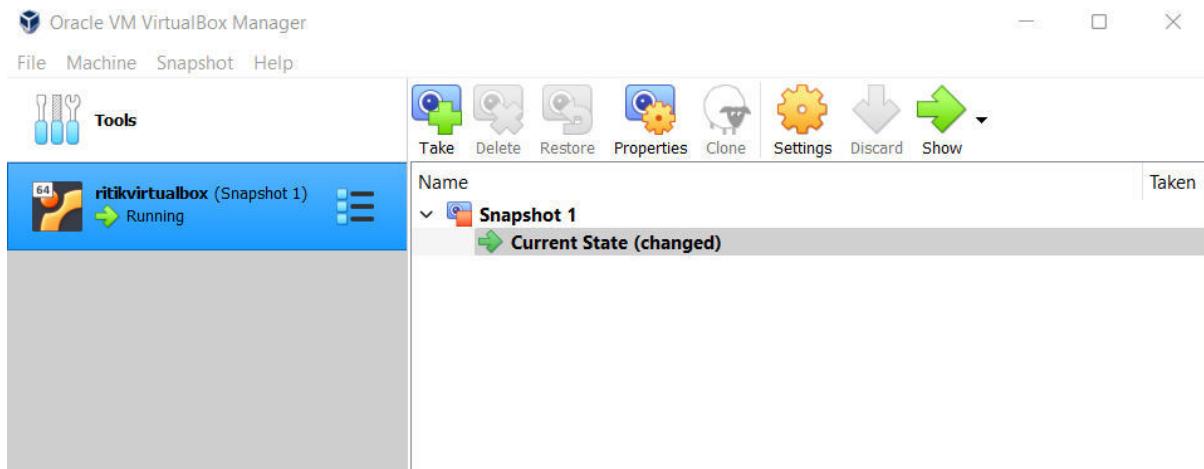




Step 6: Select Bridge Adapter and click on OK.

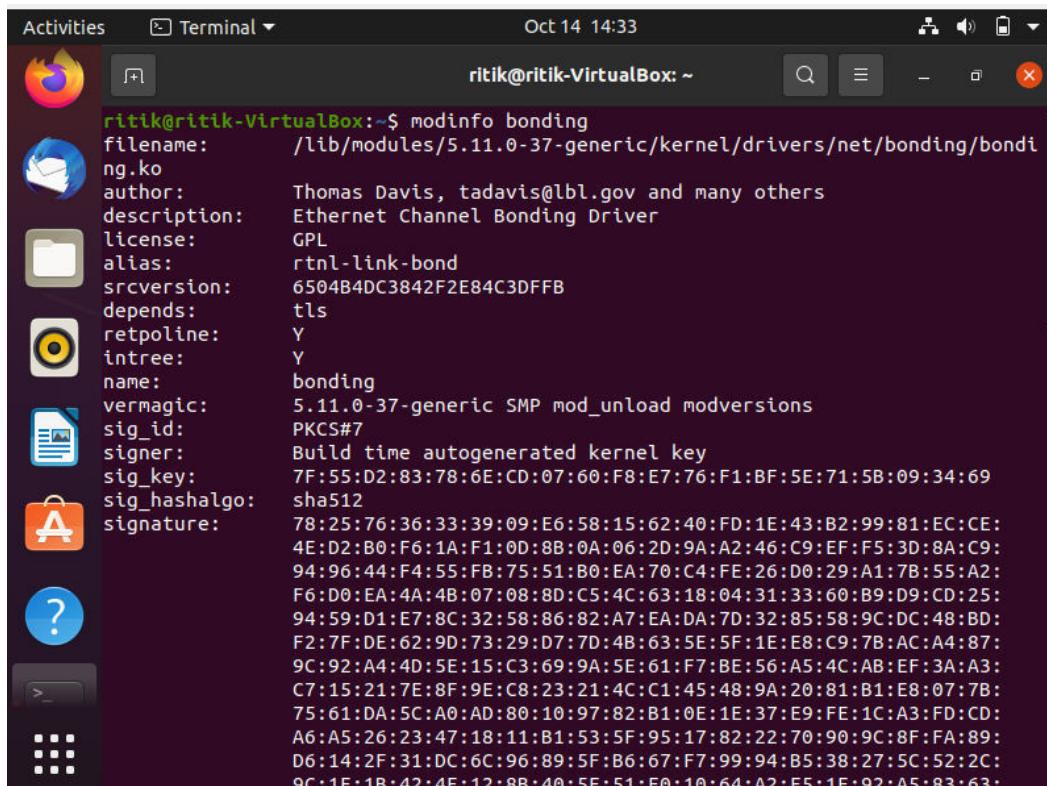


Step 7: The following window appears. Start the system.



Step 8: Type in the command modinfo bonding

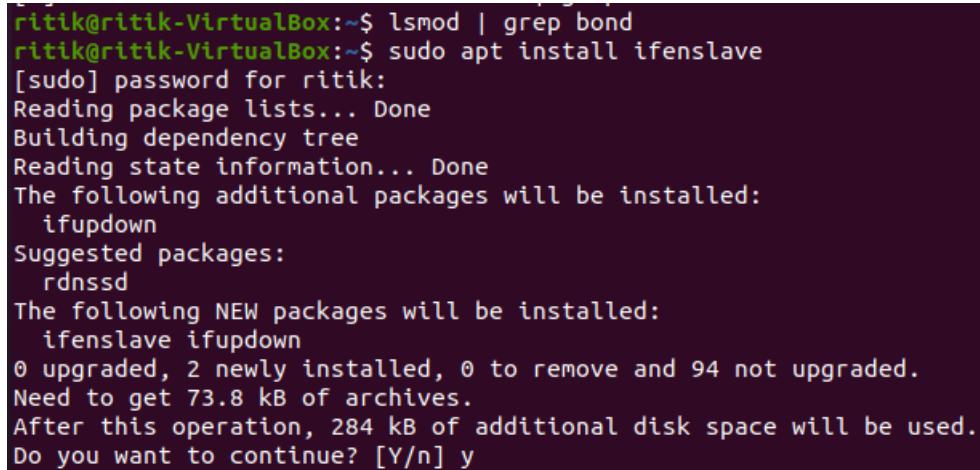
```
medha@medha-VirtualBox:~$ modinfo bonding
filename:      /lib/modules/5.11.0-37-generic/kernel/drivers/net/bonding/bondi
ng.ko
author:        Thomas Davis, tadavis@lbl.gov and many others
description:   Ethernet Channel Bonding Driver
license:       GPL
alias:         rtnl-link-bond
srcversion:    6504B4DC3842F2E84C3DFFB
depends:       tls
retpoline:     Y
intree:        Y
name:          bonding
vermagic:     5.11.0-37-generic SMP mod_unload modversions
sig_id:        PKCS#7
signer:        Build time autogenerated kernel key
sig_key:       7F:55:D2:83:78:6E:CD:07:60:F8:E7:76:F1:BF:5E:71:5B:09:34:69
sig_hashalgo:  sha512
signature:     78:25:76:36:33:39:09:E6:58:15:62:40:FD:1E:43:B2:99:81:EC:CE:
              4E:D2:B0:F6:1A:F1:0D:8B:0A:06:2D:9A:A2:46:C9:EF:F5:3D:8A:C9:
              94:96:44:F4:55:FB:75:51:B0:EA:70:C4:FE:26:D0:29:A1:7B:55:A2:
              F6:D0:EA:4A:4B:07:08:8D:C5:4C:63:18:04:31:33:60:B9:D9:CD:25:
              94:59:D1:E7:8C:32:58:86:82:A7:EA:DA:7D:32:85:58:9C:DC:48:BD:
              F2:7F:DE:62:9D:73:29:D7:7D:4B:63:5E:5F:1E:E8:C9:7B:AC:A4:87:
              9C:92:A4:4D:5E:15:C3:69:9A:5E:61:F7:BE:56:A5:4C:AB:EF:3A:A3:
              C7:15:21:7E:8F:9E:C8:23:21:4C:C1:45:48:9A:20:81:B1:E8:07:7B:
              75:61:DA:5C:A0:AD:80:10:97:82:B1:0E:1E:37:E9:FE:1C:A3:FD:CD:
              A6:A5:26:23:47:18:11:B1:53:5F:95:17:82:22:70:90:9C:8F:FA:89:
              D6:14:2F:31:DC:6C:96:89:5F:B6:67:F7:99:94:B5:38:27:5C:52:2C:
              9C:1E:1B:42:4E:12:8B:40:5F:51:E0:10:64:A2:E5:1E:92:A5:83:63:
```



A screenshot of an Ubuntu desktop environment. On the left, there's a dock with icons for the Dash, Home, Applications, and Help. The main area shows a terminal window titled "Terminal". The terminal window has a dark background and contains the following text:

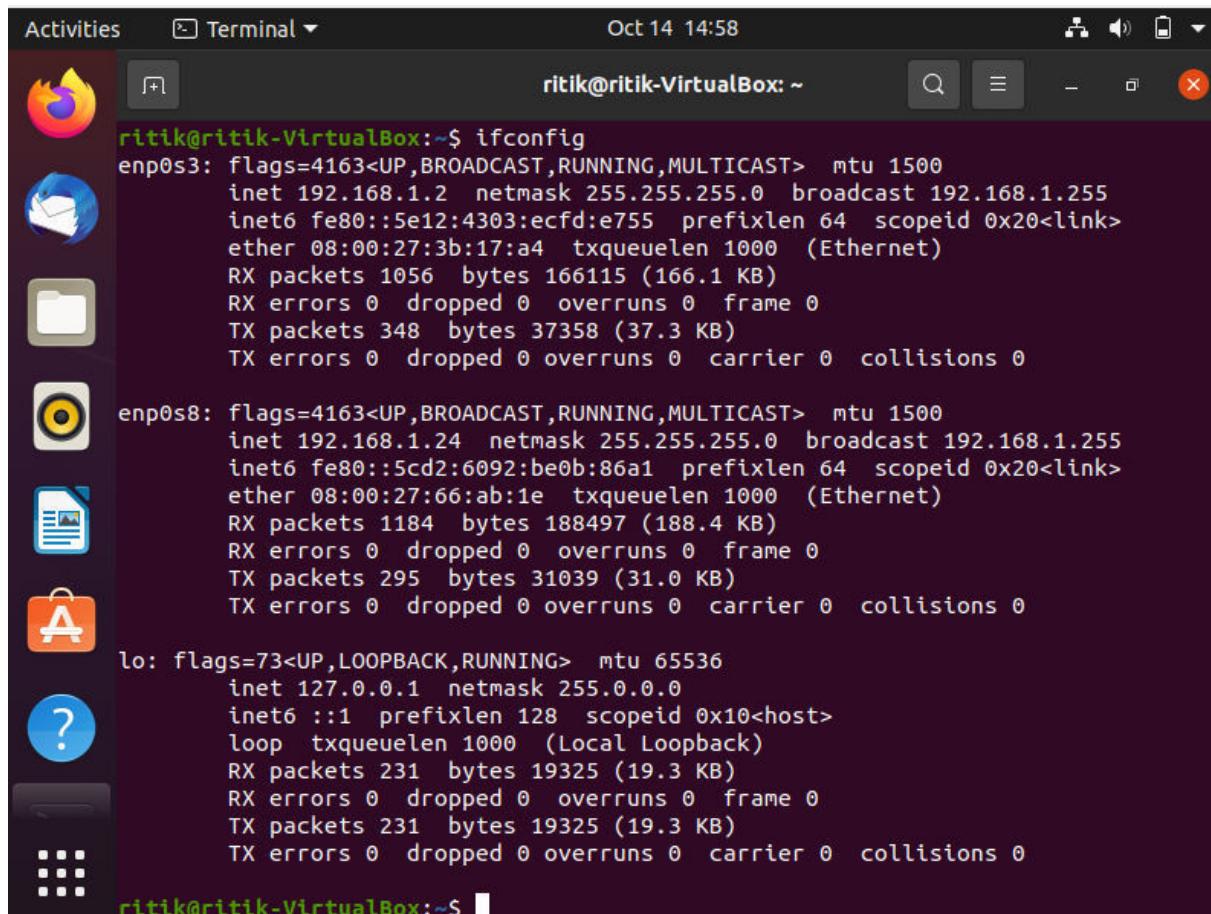
```
ritik@ritik-VirtualBox:~$ modinfo bonding
filename:      /lib/modules/5.11.0-37-generic/kernel/drivers/net/bonding/bondi
ng.ko
author:        Thomas Davis, tadavis@lbl.gov and many others
description:   Ethernet Channel Bonding Driver
license:       GPL
alias:         rtnl-link-bond
srcversion:    6504B4DC3842F2E84C3DFFB
depends:       tls
retpoline:     Y
intree:        Y
name:          bonding
vermagic:     5.11.0-37-generic SMP mod_unload modversions
sig_id:        PKCS#7
signer:        Build time autogenerated kernel key
sig_key:       7F:55:D2:83:78:6E:CD:07:60:F8:E7:76:F1:BF:5E:71:5B:09:34:69
sig_hashalgo: sha512
signature:    78:25:76:36:33:39:09:E6:58:15:62:40:FD:1E:43:B2:99:81:EC:CE:
             4E:D2:B0:F6:1A:F1:0D:8B:0A:06:2D:9A:A2:46:C9:EF:F5:3D:8A:C9:
             94:96:44:F4:55:FB:75:51:B0:EA:70:C4:FE:26:D0:29:A1:7B:55:A2:
             F6:D0:EA:4A:4B:07:08:8D:C5:4C:63:18:04:31:33:60:B9:D9:CD:25:
             94:59:D1:E7:8C:32:58:86:82:A7:EA:DA:7D:32:85:58:9C:DC:48:BD:
             F2:7F:DE:62:9D:73:29:D7:7D:4B:63:5E:5F:1E:E8:C9:7B:AC:A4:87:
             9C:92:A4:4D:5E:15:C3:69:9A:5E:61:F7:BE:56:A5:4C:AB:EF:3A:A3:
             C7:15:21:7E:8F:9E:C8:23:21:4C:C1:45:48:9A:20:81:B1:E8:07:7B:
             75:61:DA:5C:A0:AD:80:10:97:82:B1:0E:1E:37:E9:FE:1C:A3:FD:CD:
             A6:A5:26:23:47:18:11:B1:53:5F:95:17:82:22:70:90:9C:8F:FA:89:
             D6:14:2F:31:DC:6C:96:89:5F:B6:67:F7:99:94:B5:38:27:5C:52:2C:
             9C:1E:1B:42:4E:12:8B:40:5E:51:E0:10:64:A2:E5:1E:92:A5:83:63:
```

Step 9: Type in the command lsmod | grep bond & install ifenslave



```
ritik@ritik-VirtualBox:~$ lsmod | grep bond
ritik@ritik-VirtualBox:~$ sudo apt install ifenslave
[sudo] password for ritik:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ifupdown
Suggested packages:
  rdnssd
The following NEW packages will be installed:
  ifenslave ifupdown
0 upgraded, 2 newly installed, 0 to remove and 94 not upgraded.
Need to get 73.8 kB of archives.
After this operation, 284 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Step 10: Check how many interfaces are available for bonding



A screenshot of a Ubuntu desktop environment. On the left is a dock with icons for the Dash, Home, Applications, and Help. A terminal window is open in the center, titled 'Terminal'. The title bar shows the date and time as 'Oct 14 14:58' and the user as 'ritik@ritik-VirtualBox: ~'. The terminal window contains the output of the 'ifconfig' command:

```
ritik@ritik-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.1.2 netmask 255.255.255.0 broadcast 192.168.1.255
        inet6 fe80::5e12:4303:ecfd:e755 prefixlen 64 scopeid 0x20<link>
          ether 08:00:27:3b:17:a4 txqueuelen 1000 (Ethernet)
            RX packets 1056 bytes 166115 (166.1 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 348 bytes 37358 (37.3 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.1.24 netmask 255.255.255.0 broadcast 192.168.1.255
        inet6 fe80::5cd2:6092:be0b:86a1 prefixlen 64 scopeid 0x20<link>
          ether 08:00:27:66:ab:1e txqueuelen 1000 (Ethernet)
            RX packets 1184 bytes 188497 (188.4 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 295 bytes 31039 (31.0 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 231 bytes 19325 (19.3 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 231 bytes 19325 (19.3 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ritik@ritik-VirtualBox:~$
```

Step 11: Change the state of both the Ethernet interfaces to "down"

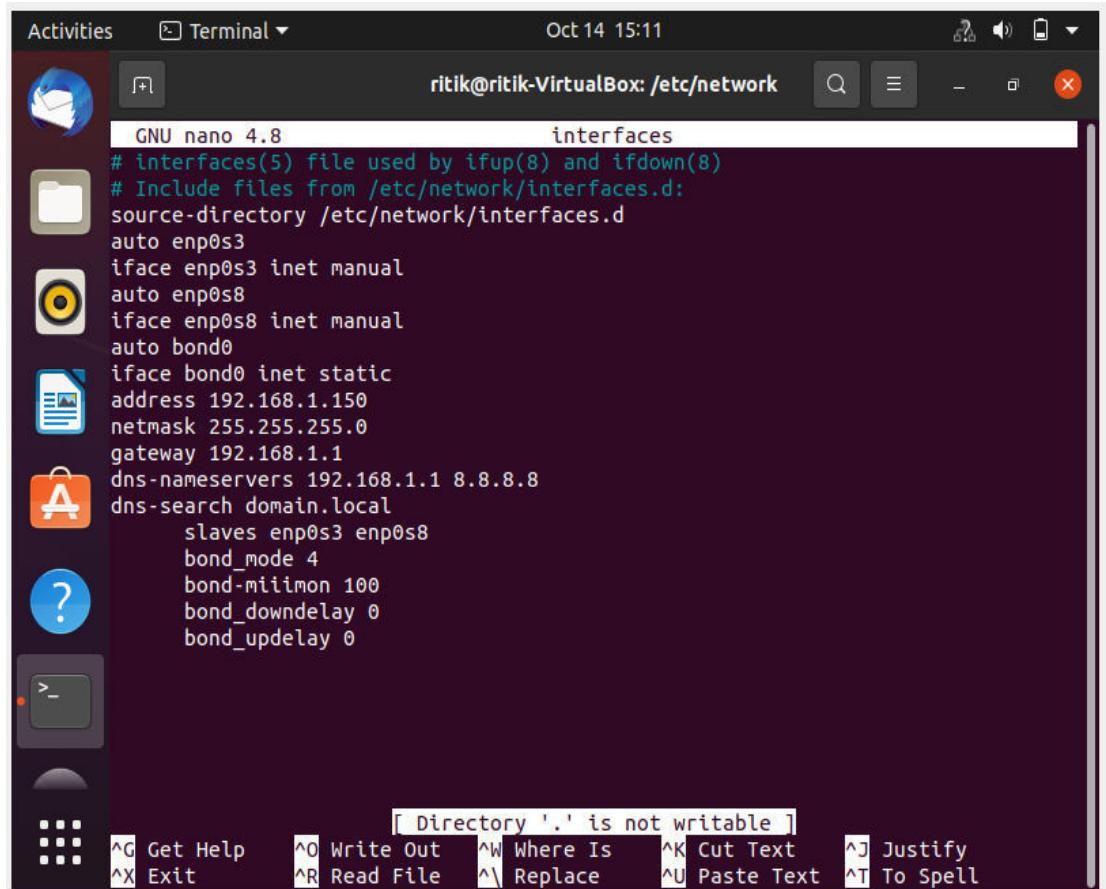
```
ritik@ritik-VirtualBox:/etc/network$ sudo ifconfig enp0s3 down
[sudo] password for ritik:
ritik@ritik-VirtualBox:/etc/network$ sudo ifconfig enp0s8 down
ritik@ritik-VirtualBox:/etc/network$
```

Step 12: make a bond network on master node bond0 via the ip link command, add both the interfaces to the master node and affirm the creation of network bonding

```
ritik@ritik-VirtualBox:/etc/network$ sudo ip link add bond0 type bond mode 802.3ad
ritik@ritik-VirtualBox:/etc/network$ sudo ip link set enp0s3 master bond0
ritik@ritik-VirtualBox:/etc/network$ sudo ip link set enp0s8 master bond0
ritik@ritik-VirtualBox:/etc/network$ sudo ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0s3: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc fq_codel master bond0 state UP mode DEFAULT group default qlen 1000
    link/ether 08:00:27:3b:17:a4 brd ff:ff:ff:ff:ff:ff
3: enp0s8: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc fq_codel master bond0 state UP mode DEFAULT group default qlen 1000
    link/ether 08:00:27:3b:17:a4 brd ff:ff:ff:ff:ff:ff
4: bond0: <BROADCAST,MULTICAST,MASTER> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen 1000
    link/ether 08:00:27:3b:17:a4 brd ff:ff:ff:ff:ff:ff
ritik@ritik-VirtualBox:/etc/network$ █
```

Step 13: Change permissions for /etc/network/interfaces file, update the configuration and save it.

```
ritik@ritik-VirtualBox:/etc/network$ cat interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)
# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d
ritik@ritik-VirtualBox:/etc/network$ ls
if-down.d  if-post-down.d  if-pre-up.d  if-up.d  interfaces  interfaces.d
ritik@ritik-VirtualBox:/etc/network$ ls -ltr
total 24
drwxr-xr-x 2 root root 4096 May  3  2019 interfaces.d
drwxr-xr-x 2 root root 4096 Aug 19 16:10 if-down.d
-rw-r--r-- 1 root root 142 Oct 14 14:35 interfaces
drwxr-xr-x 2 root root 4096 Oct 14 14:35 if-post-down.d
drwxr-xr-x 2 root root 4096 Oct 14 14:35 if-pre-up.d
drwxr-xr-x 2 root root 4096 Oct 14 14:35 if-up.d
ritik@ritik-VirtualBox:/etc/network$ chmod 777 interfaces
chmod: changing permissions of 'interfaces': Operation not permitted
ritik@ritik-VirtualBox:/etc/network$ sudo chmod 777 interfaces
ritik@ritik-VirtualBox:/etc/network$ ls -ltr
total 24
drwxr-xr-x 2 root root 4096 May  3  2019 interfaces.d
drwxr-xr-x 2 root root 4096 Aug 19 16:10 if-down.d
-rwxrwxrwx 1 root root 142 Oct 14 14:35 interfaces
drwxr-xr-x 2 root root 4096 Oct 14 14:35 if-post-down.d
drwxr-xr-x 2 root root 4096 Oct 14 14:35 if-pre-up.d
drwxr-xr-x 2 root root 4096 Oct 14 14:35 if-up.d
ritik@ritik-VirtualBox:/etc/network$
```



Step 14: To enable the network bond, we need to change the states of both slaves interfaces to down and change the state of the master node to up

```
ritik@ritik-VirtualBox:/etc/network$ sudo ifconfig enp0s3 down && sudo ifconfig  
enp0s8 down & sudo ifconfig bond0 up  
[1] 2357  
ritik@ritik-VirtualBox:/etc/network$
```

Step 15: restart the network service

```
ritik@ritik-VirtualBox:/etc/network$ sudo systemctl restart networking.service  
ritik@ritik-VirtualBox:/etc/network$
```

Step 16: Confirm whether the master interface has been “up” or not

```
ritik@ritik-VirtualBox:~$ sudo ifconfig bond0  
bond0: flags=5123<UP,BROADCAST,MASTER,MULTICAST> mtu 1500  
      inet 192.168.1.150 netmask 255.255.255.0 broadcast 192.168.1.255  
        ether 1a:81:7f:a7:0c:f1 txqueuelen 1000 (Ethernet)  
          RX packets 0 bytes 0 (0.0 B)  
          RX errors 0 dropped 0 overruns 0 frame 0  
          TX packets 0 bytes 0 (0.0 B)  
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
ritik@ritik-VirtualBox:~$
```

Step 17: Check out the status of a newly created network bond that has been created successfully by utilizing the below query.

```
ritik@ritik-VirtualBox:~$ sudo cat /proc/net/bonding/bond0  
Ethernet Channel Bonding Driver: v5.11.0-37-generic  
  
Bonding Mode: IEEE 802.3ad Dynamic link aggregation  
Transmit Hash Policy: layer2 (0)  
MII Status: down  
MII Polling Interval (ms): 100  
Up Delay (ms): 0  
Down Delay (ms): 0  
Peer Notification Delay (ms): 0  
  
802.3ad info  
LACP rate: slow  
Min links: 0  
Aggregator selection policy (ad_select): stable  
System priority: 0  
System MAC address: 00:00:00:00:00:00
```

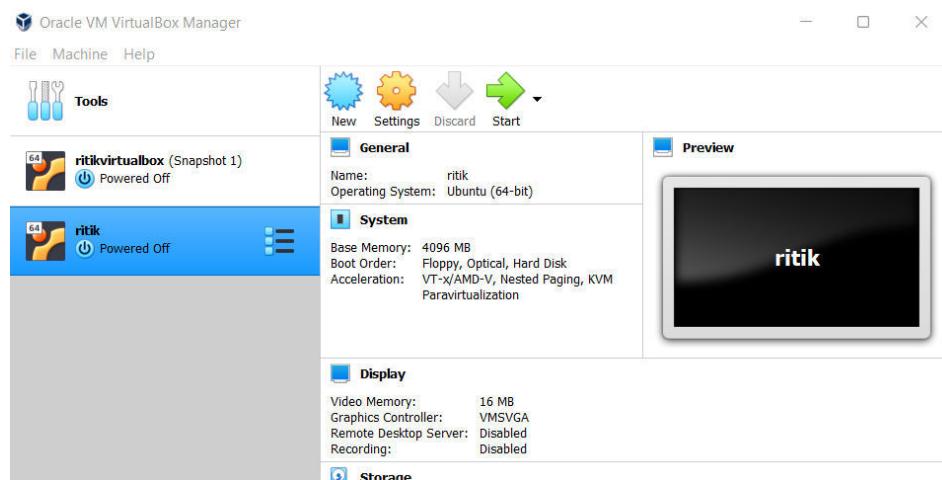
```

Active Aggregator Info:
    Aggregator ID: 2
    Number of ports: 1
    Actor Key: 9
    Partner Key: 1
    Partner Mac Address: 00:00:00:00:00:00

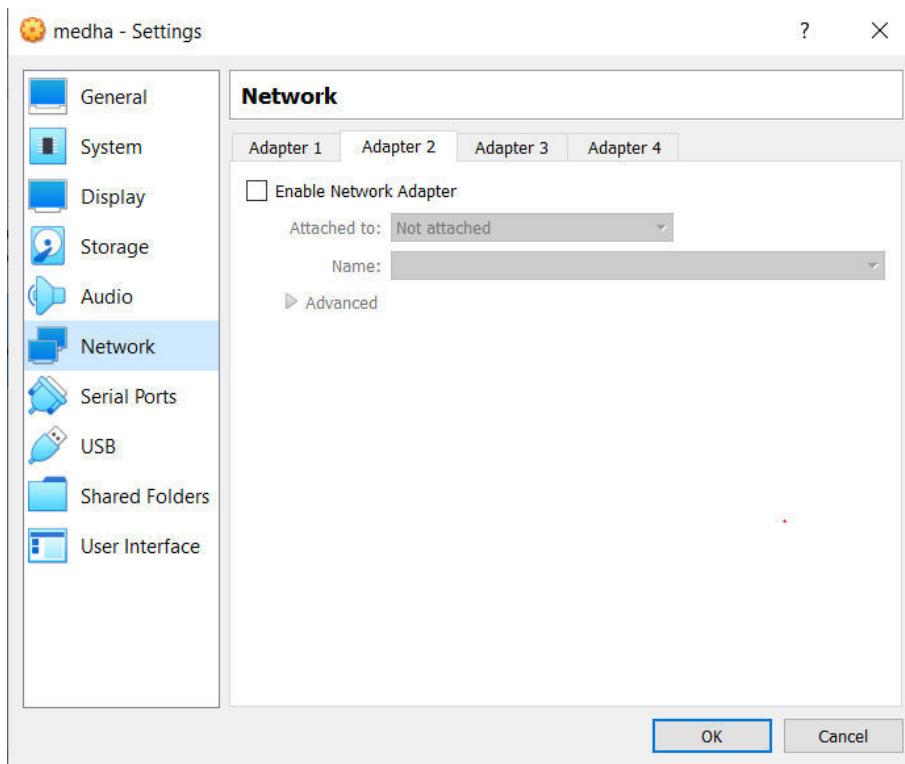
Slave Interface: enp0s8
MII Status: up
Speed: 1000 Mbps
Duplex: full
Link Failure Count: 1
Permanent HW addr: 08:00:27:32:5a:6a
Slave queue ID: 0
Aggregator ID: 2
Actor Churn State: none
Partner Churn State: churned
Actor Churned Count: 1
Partner Churned Count: 2
details actor lacp pdu:
    system priority: 65535
    system mac address: 08:00:27:4d:02:aa
    port key: 9
    port priority: 255
    port number: 2
    port state: 77
details partner lacp pdu:
    system priority: 65535
    system mac address: 00:00:00:00:00:00

```

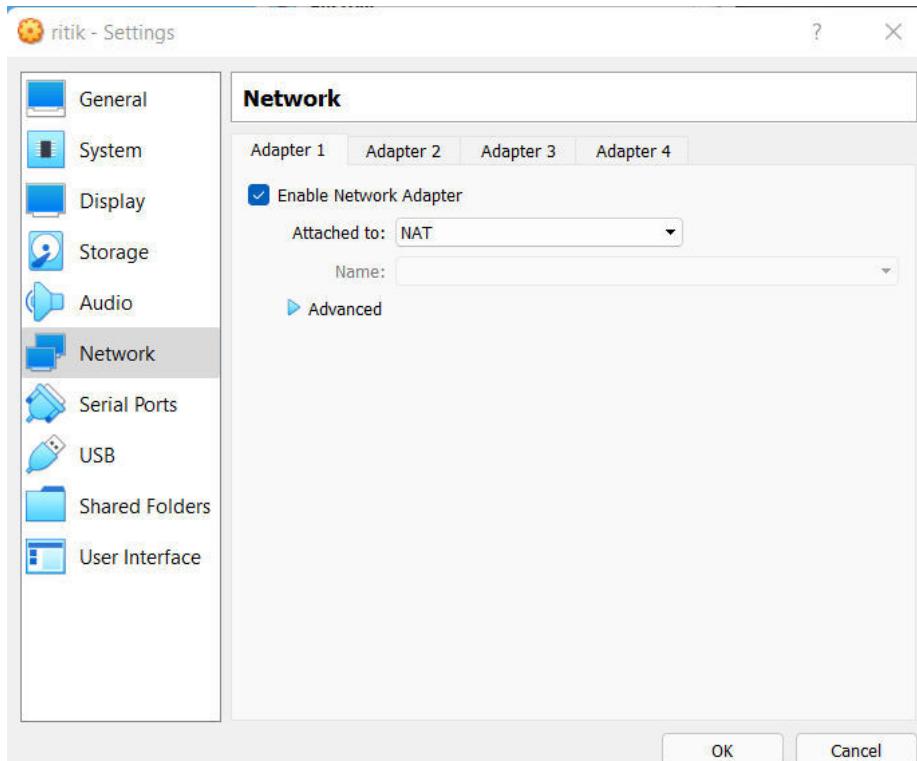
### Step 18: Create a new machine



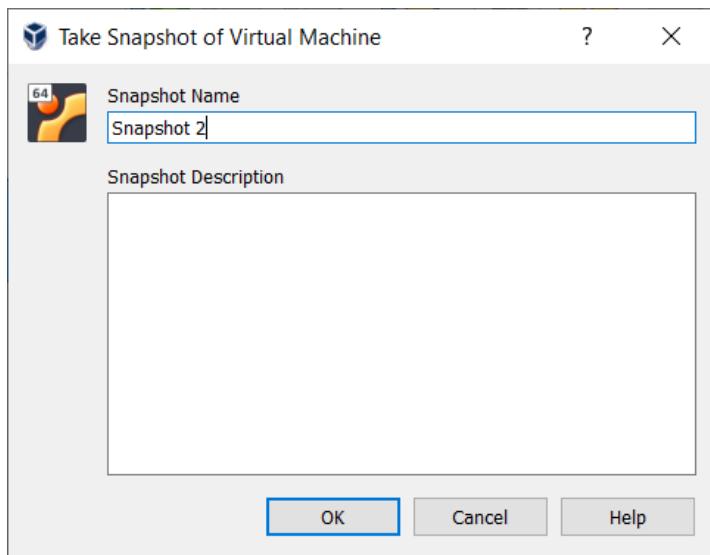
Step 19: Enable Network Adapter 2



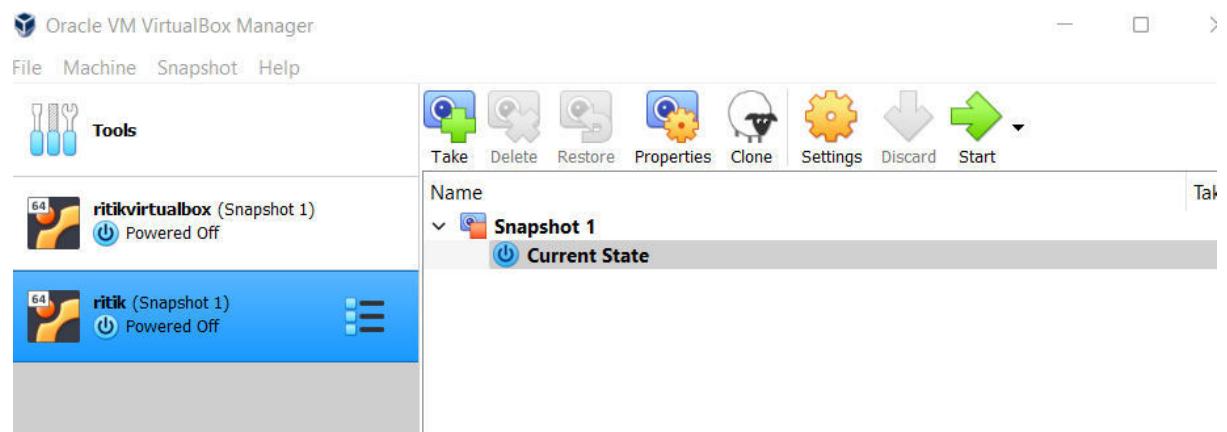
Step 20: Click on Bridged Adapter



Step 21: Take Snapshot of current state



Step 22: The following screen appears



Step 23: Run the command ifconfig in terminal after starting the machine

```
ritik@ritik-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
        inet6 fe80::19c7:928c:cb0d:e9fc prefixlen 64 scopeid 0x20<link>
          ether 08:00:27:af:4e:be txqueuelen 1000 (Ethernet)
            RX packets 488 bytes 462650 (462.6 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 389 bytes 57862 (57.8 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.1.25 netmask 255.255.255.0 broadcast 192.168.1.255
        inet6 fe80::7cf7:d7bf:d052:e7a6 prefixlen 64 scopeid 0x20<link>
          ether 08:00:27:ac:90:99 txqueuelen 1000 (Ethernet)
            RX packets 261 bytes 47114 (47.1 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 265 bytes 31274 (31.2 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

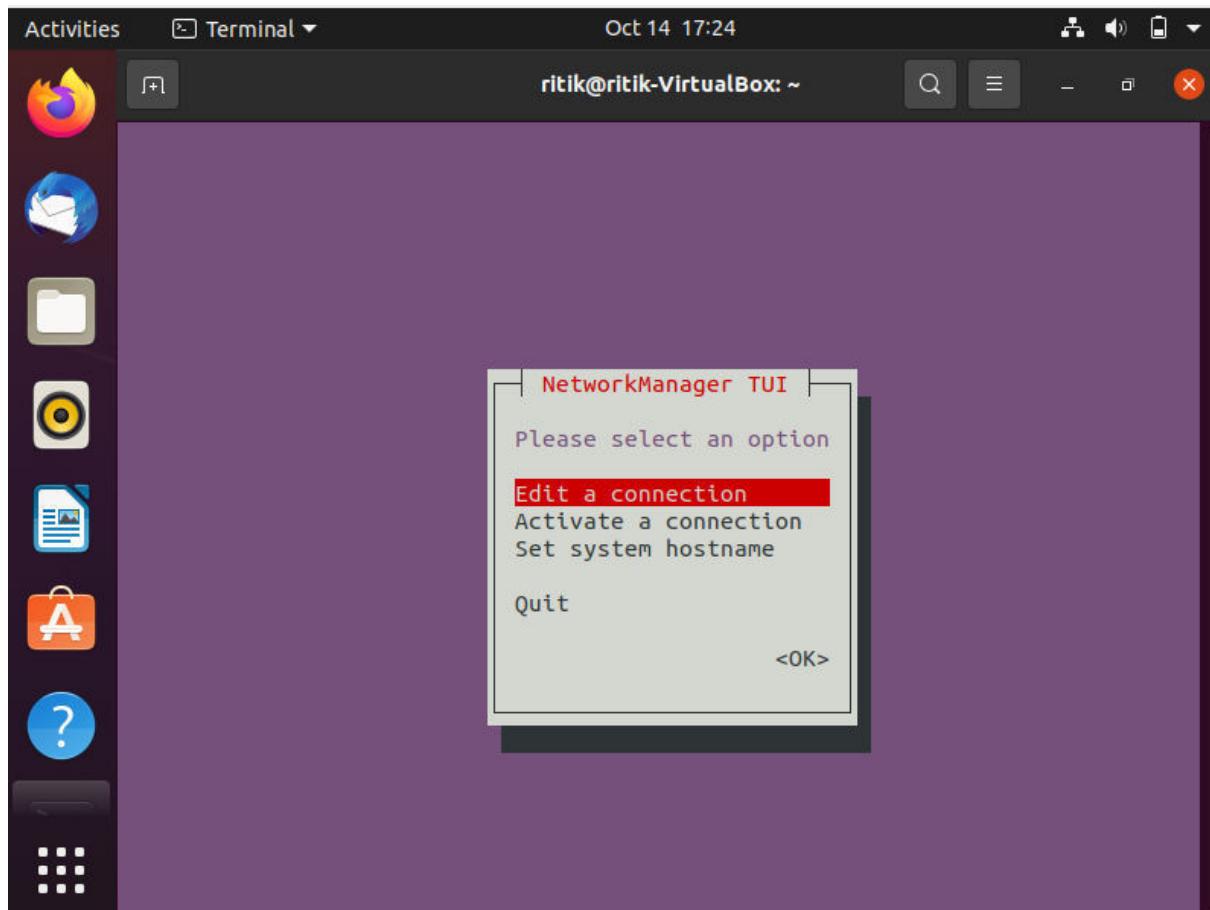
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
          Loop txqueuelen 1000 (Local Loopback)
            RX packets 312 bytes 27413 (27.4 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 312 bytes 27413 (27.4 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ritik@ritik-VirtualBox:~$
```

Step 24: Type in command nmtui.

```
medha@medha-VirtualBox:~$ nmtui
```

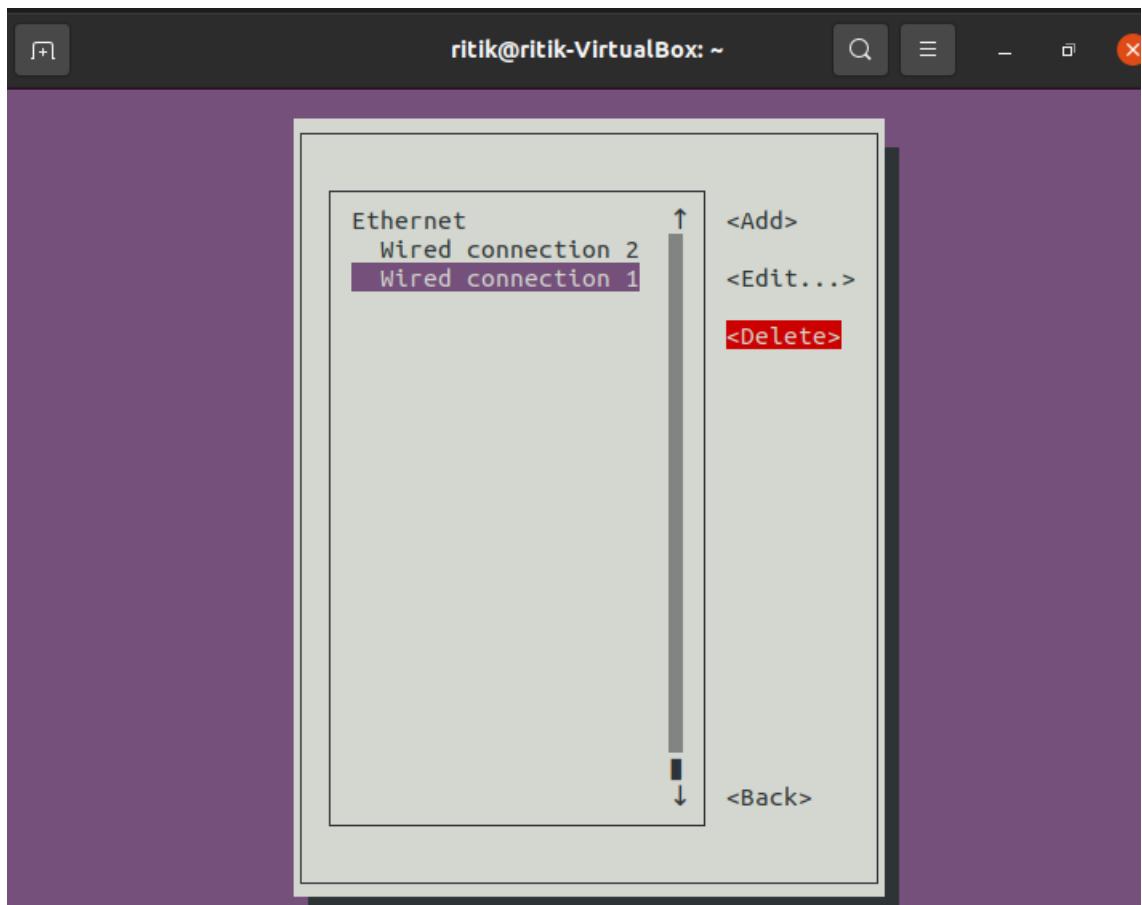
Step 25: Click on Edit a connection



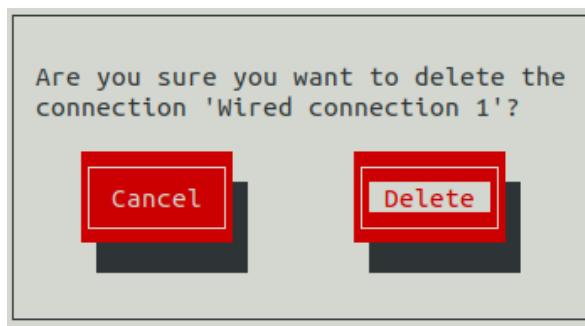
Step 26: Click on Wired connection 1



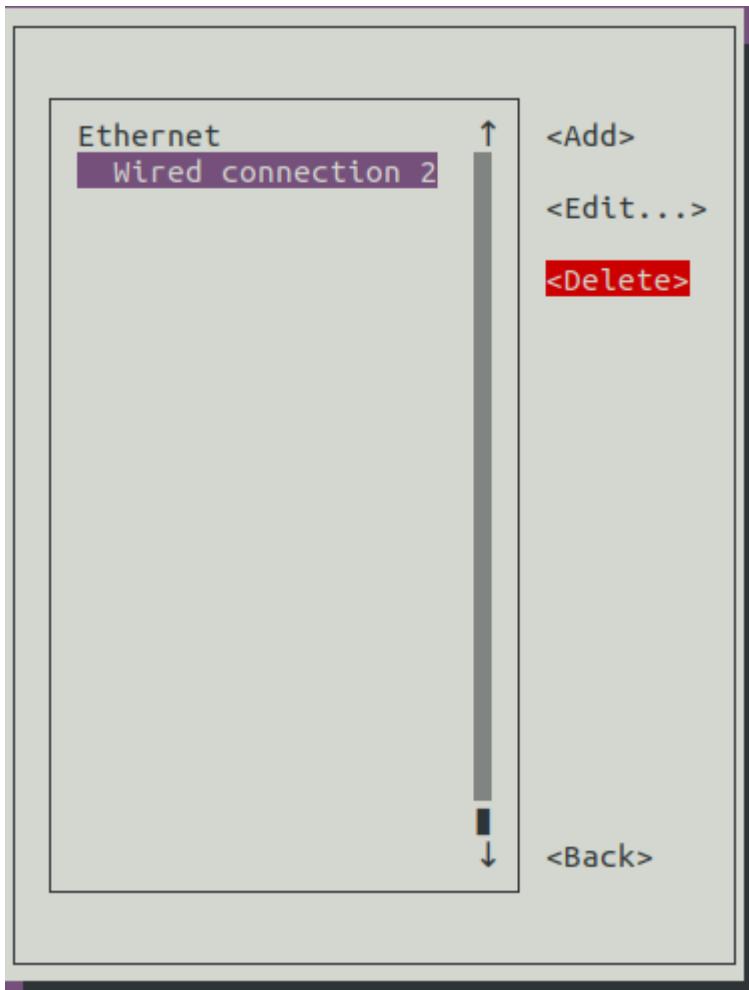
Step 27: Click on Delete.



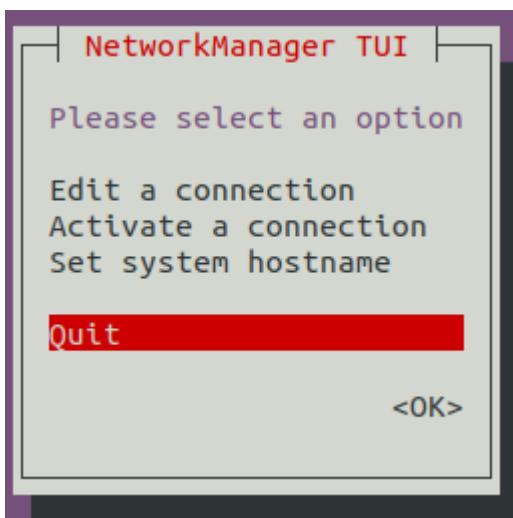
Step 28: Click on Delete.



Step 29: Click on Back option



Step 30: Click on Quit.



Step 31: Type in the following command: ifconfig | more

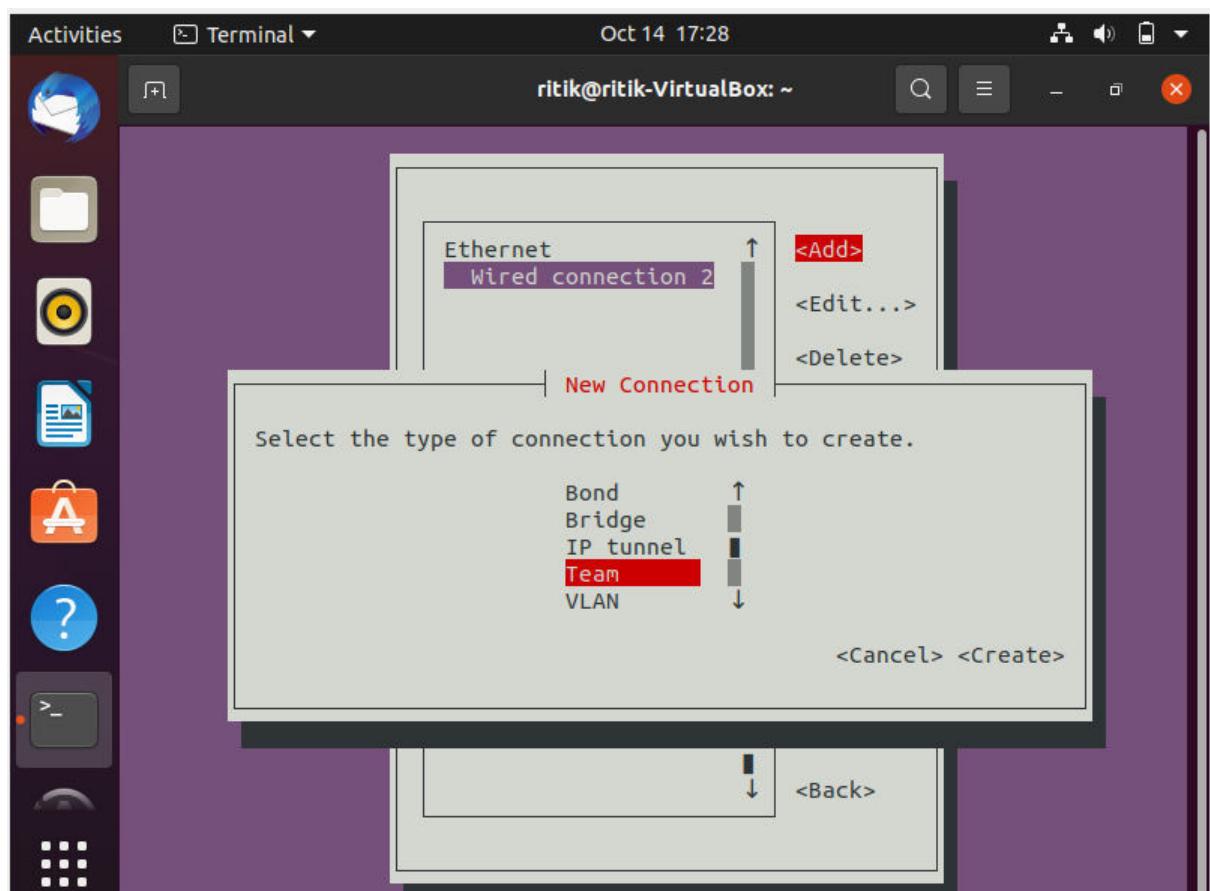
```
ritik@ritik-VirtualBox:~$ ifconfig | more
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        ether 08:00:27:af:4e:be txqueuelen 1000 (Ethernet)
        RX packets 489 bytes 462740 (462.7 KB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 400 bytes 58856 (58.8 KB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.1.25 netmask 255.255.255.0 broadcast 192.168.1.255
        inet6 fe80::7cf7:d7bf:d052:e7a6 prefixlen 64 scopeid 0x20<link>
        ether 08:00:27:ac:90:99 txqueuelen 1000 (Ethernet)
        RX packets 491 bytes 101070 (101.0 KB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 317 bytes 36902 (36.9 KB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

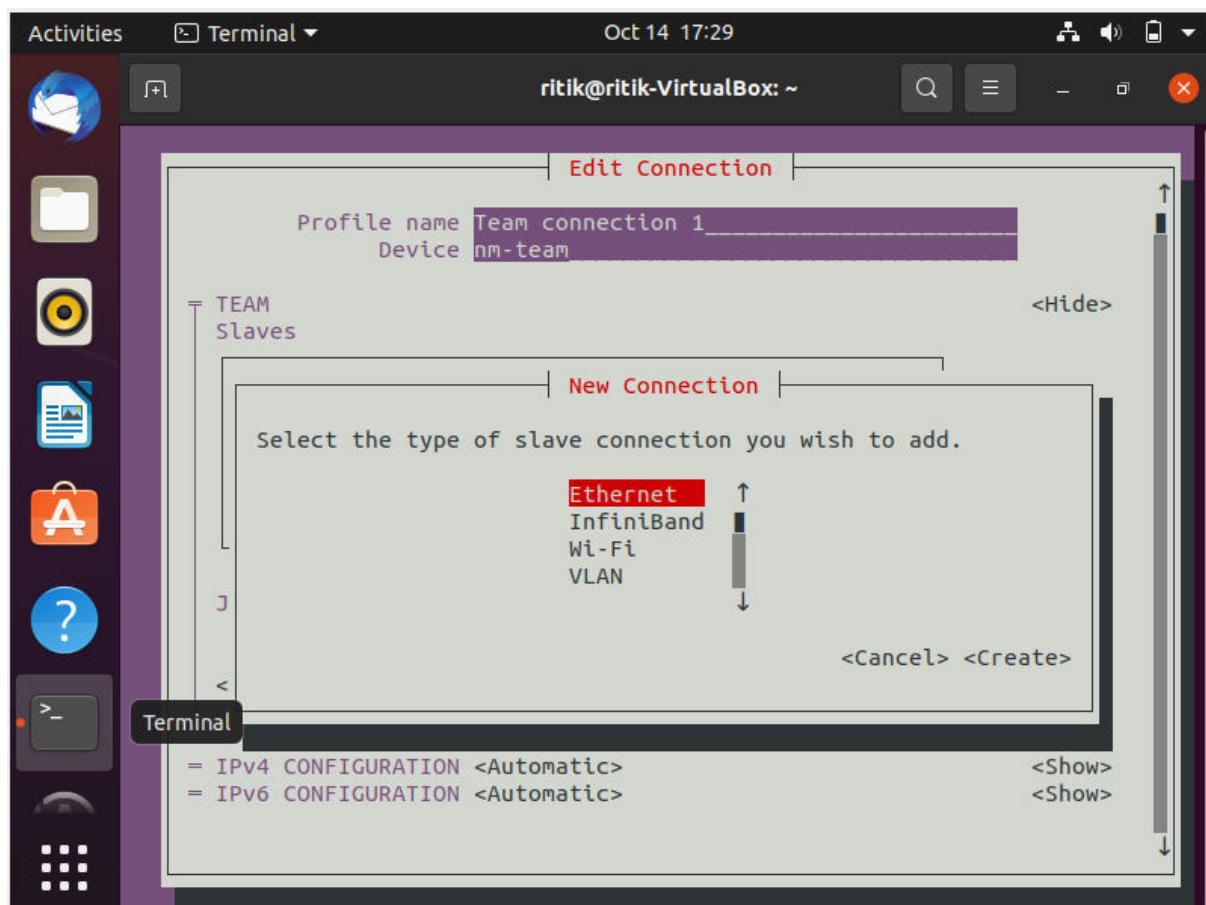
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
        RX packets 341 bytes 30325 (30.3 KB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 341 bytes 30325 (30.3 KB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Step 32: Type in nmtui and then click on add and select team.

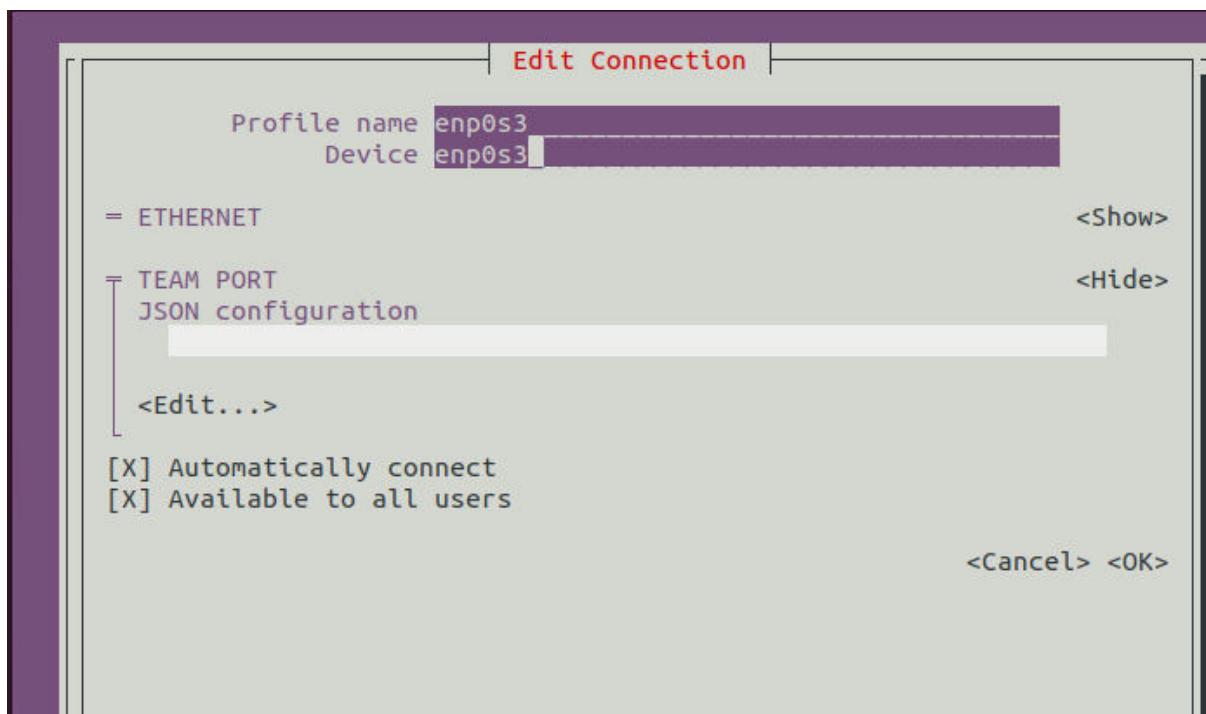
```
ritik@ritik-VirtualBox:~$ nmtui
```

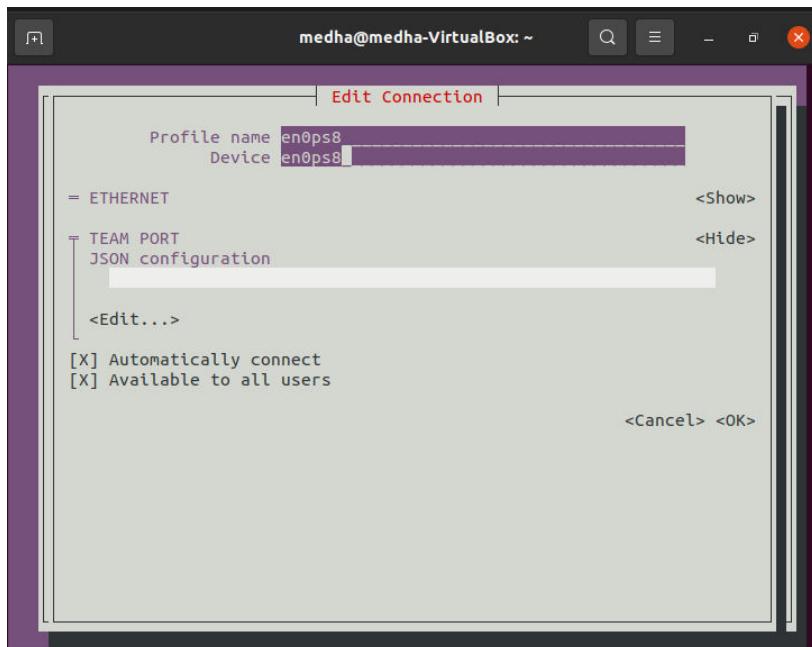
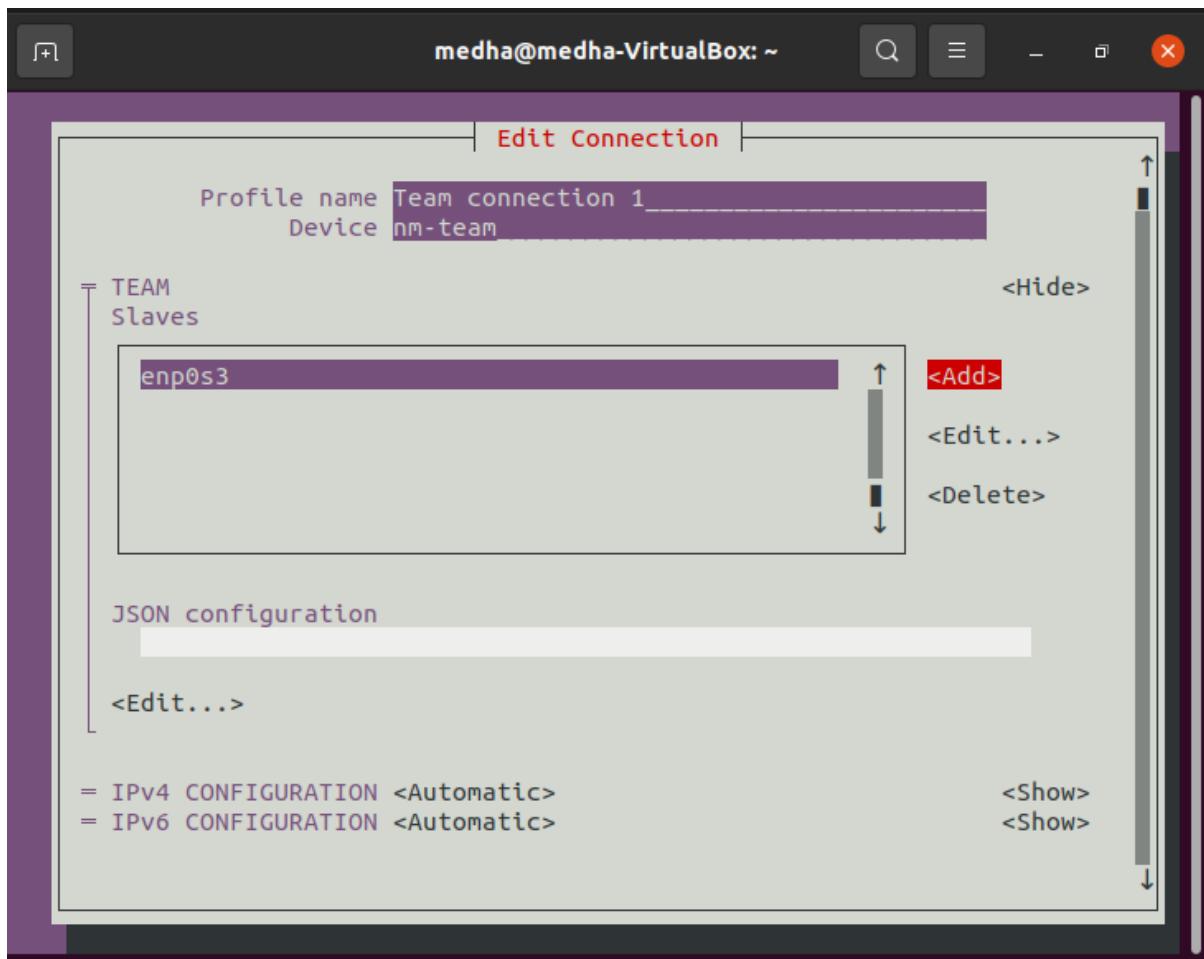


Step 33: Select the slave connection as Ethernet, add and edit the connection names.



0





Step 34: Run the command: ifconfig | more , nmcli con show

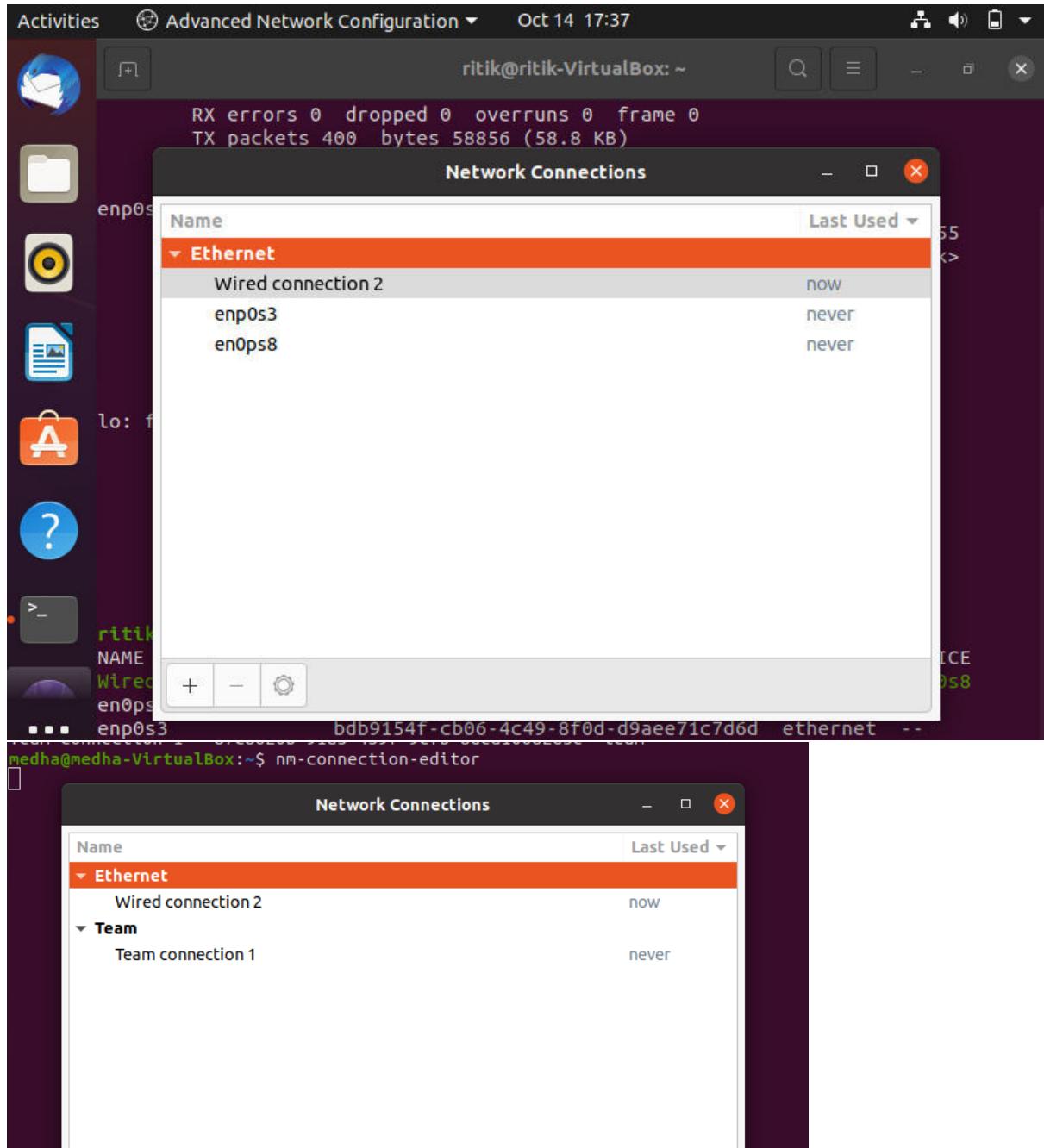
```
ritik@ritik-VirtualBox:~$ ifconfig | more
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      ether 08:00:27:af:4e:be txqueuelen 1000 (Ethernet)
      RX packets 489 bytes 462740 (462.7 KB)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 400 bytes 58856 (58.8 KB)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

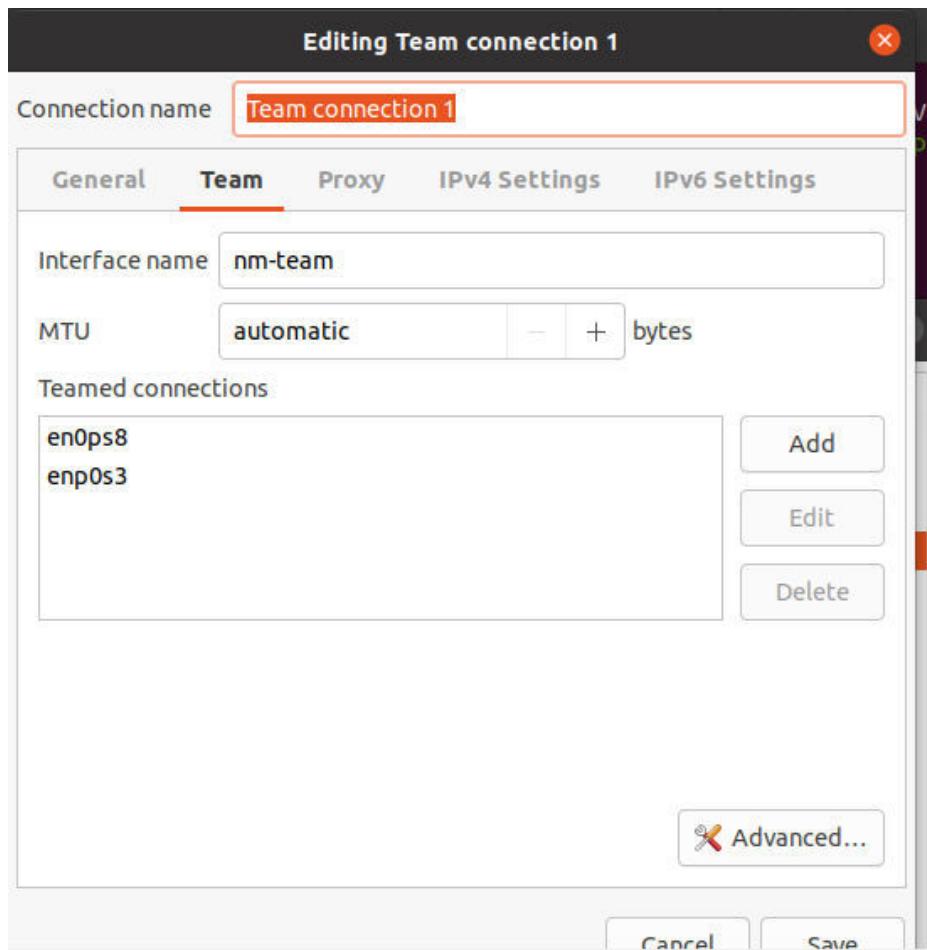
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.1.25 netmask 255.255.255.0 broadcast 192.168.1.255
      inet6 fe80::7cf7:d7bf:d052:e7a6 prefixlen 64 scopeid 0x20<link>
      ether 08:00:27:ac:90:99 txqueuelen 1000 (Ethernet)
      RX packets 964 bytes 189683 (189.6 KB)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 358 bytes 40492 (40.4 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 355 bytes 31354 (31.3 KB)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 355 bytes 31354 (31.3 KB)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
ritik@ritik-VirtualBox:~$ nmcli con show
NAME           UUID                                  TYPE      DEVICE
Wired connection 2  af2275e7-d910-33fc-8f3d-19f806e2b38b  ethernet  enp0s8
en0ps8         c275763e-0613-4fd9-9244-126e73b00204  ethernet  --
enp0s3         bdb9154f-cb06-4c49-8f0d-d9aee71c7d6d  ethernet  --
ritik@ritik-VirtualBox:~$ █
```

Step 35: Run the command nm-connection-editor in terminal





Step 36: Edit the interfaces file configure bond0 and make eth0 and eth1 as slave of bond0

The screenshot shows a terminal window with a dark theme. At the top, there's a header bar with the text "Text Editor" and a timestamp "Oct 12 21:01". Below the header are standard file operations buttons: "Open", "Save", and a "New" icon. The main area of the window displays the contents of the "/etc/network/interfaces" file. The file configuration is as follows:

```
1 auto lo
2 iface lo inet loopback
3
4 auto eth0
5 iface eth0 inet manual
6 bond-master bond0
7 bond-primary eth0
8
9 auto eth1
10 iface eth1 inet manual
11 bond-master bond0
12
13 auto bond0
14 iface bond0 inet dhcp
15 bond-mode 6
16 bond-miimon 100
17 bond-lacp-rate 1
18 bond-slaves eth0 eth1
```

Step 37: Restart NetworkManager.service

```
medha@nedha-VirtualBox:~$ sudo systemctl restart NetworkManager.service
medha@nedha-VirtualBox:~$
```

Step 38: Run ifconfig command

```
ritik@ritik-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      ether 08:00:27:af:4e:be txqueuelen 1000 (Ethernet)
      RX packets 489 bytes 462740 (462.7 KB)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 400 bytes 58856 (58.8 KB)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.1.25 netmask 255.255.255.0 broadcast 192.168.1.255
      inet6 fe80::7cf7:d7bf:d052:e7a6 prefixlen 64 scopeid 0x20<link>
      ether 08:00:27:ac:90:99 txqueuelen 1000 (Ethernet)
      RX packets 4204 bytes 693074 (693.0 KB)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 604 bytes 64388 (64.3 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 434 bytes 38653 (38.6 KB)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 434 bytes 38653 (38.6 KB)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Step 39: Run the following command

```
medha@medha-VirtualBox:~$ sudo cat /proc/net/bonding/bond0
Ethernet Channel Bonding Driver: v5.11.0-37-generic

Bonding Mode: IEEE 802.3ad Dynamic link aggregation
Transmit Hash Policy: layer2 (0)
MII Status: up
MII Polling Interval (ms): 100
Up Delay (ms): 0
Down Delay (ms): 0
Peer Notification Delay (ms): 0

802.3ad info
LACP rate: slow
Min links: 0
Aggregator selection policy (ad_select): stable
System priority: 65535
System MAC address: 08:00:27:4d:02:aa
Active Aggregator Info:
    Aggregator ID: 2
    Number of ports: 1
    Actor Key: 9
    Partner Key: 1
    Partner Mac Address: 00:00:00:00:00:00

Slave Interface: enp0s8
MII Status: up
Speed: 1000 Mbps
Duplex: full
Link Failure Count: 1
```

**RESULTS AND CONCLUSIONS:** NIC Bonding, and Teaming have been implemented.



## **EXPERIMENT 14**

**AIM:** Configure NFS Server and Client in Linux

### **THEORY:**

NFS, which stands for Network File System, is a distributed filesystem protocol that allows for the mounting of remote directories from an NFS server onto a client server. These directories can then be accessed from the client much like local storage, and the mounting process can even be automated for greater convenience. Using NFS mounts is a useful ability in situations where space usage needs to be optimized and multiple clients need to access the same server space, with the main advantage of NFS being its allowance for central management of files. This decreases administrator workload as well as enhancing the sharing possibilities of individual files and repositories.

NFS or Network File System was originally developed only by Sun Microsystems.

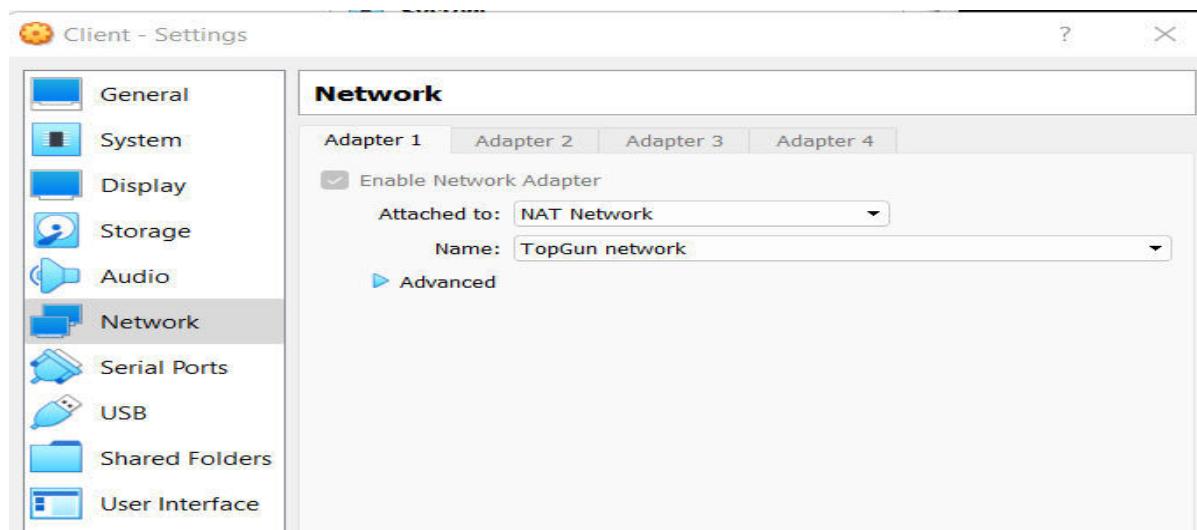
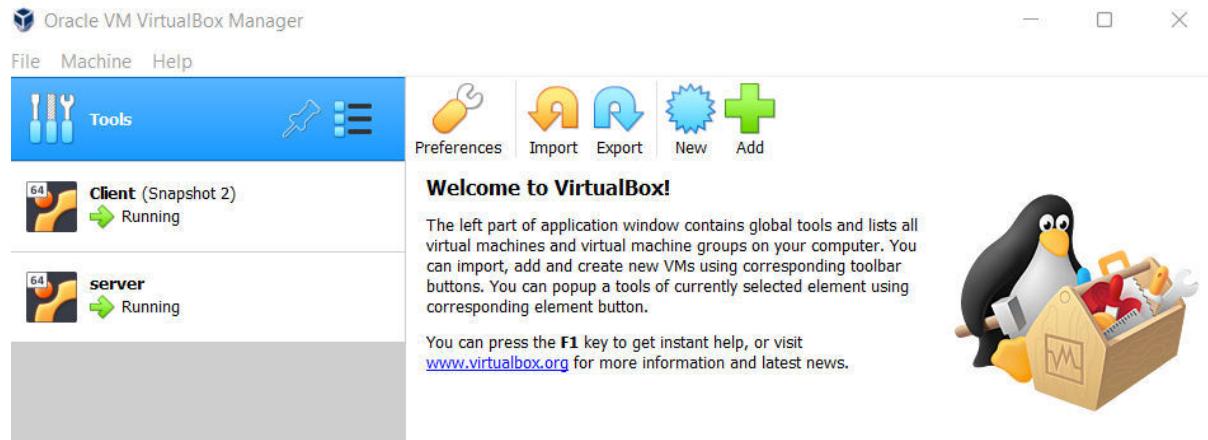
NFS operates in a client-server environment where the server is responsible for managing the authentication, authorization, and administration of clients, as well as for all data shared within a particular file system. After authorization, any number of clients can access the shared data as if it were in their internal storage.

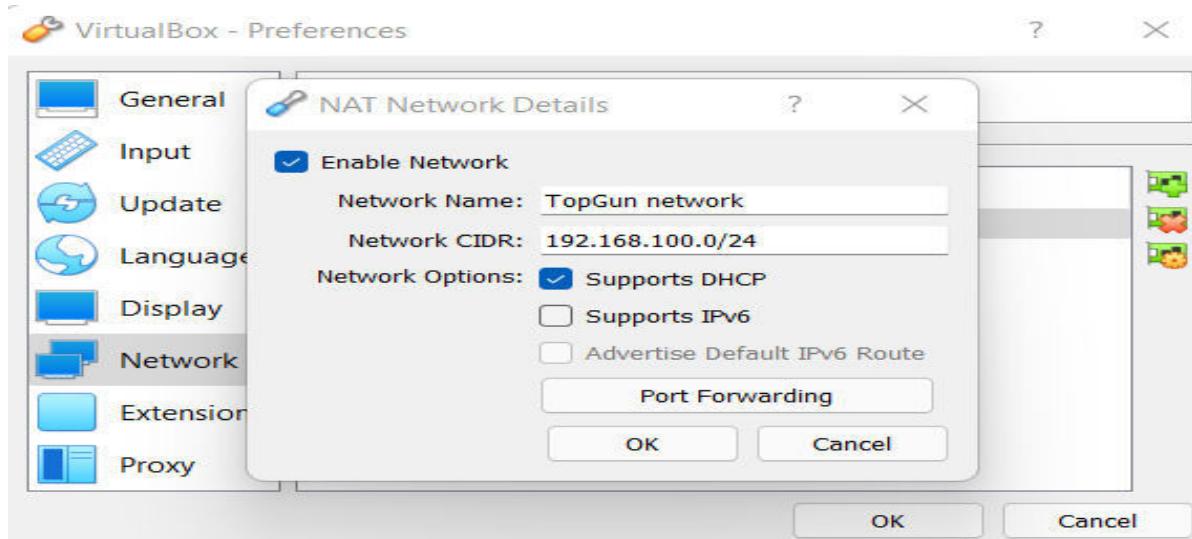
Using NFS is quick and easy for accessing remote network systems, the protocol NFS uses is not encrypted and thus considerations should be taken such as routing over SSH or VPN in order to enhance security.

## PROCEDURE:

### Part 1: Set up a server and client machine

#### Step -1)





**Step – 2)** Start both the machines and type in ‘ip a’ command on both system’s terminal.

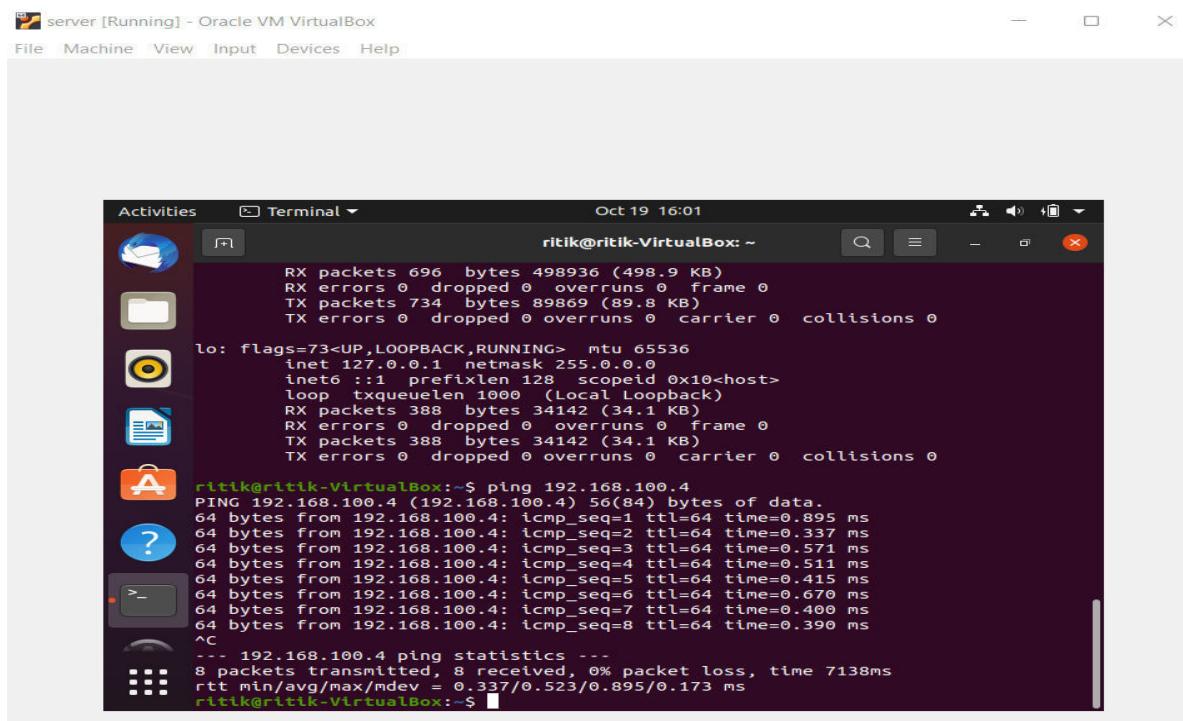
```
ritik@ritik-VirtualBox:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    qlen 1000
        link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:af:4e:be brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.4/24 brd 192.168.100.255 scope global dynamic noprefixroute
        enp0s3
            valid_lft 538sec preferred_lft 538sec
        inet6 fe80::1855:c9b2:5cc0:d0cd/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
ritik@ritik-VirtualBox:~$
```

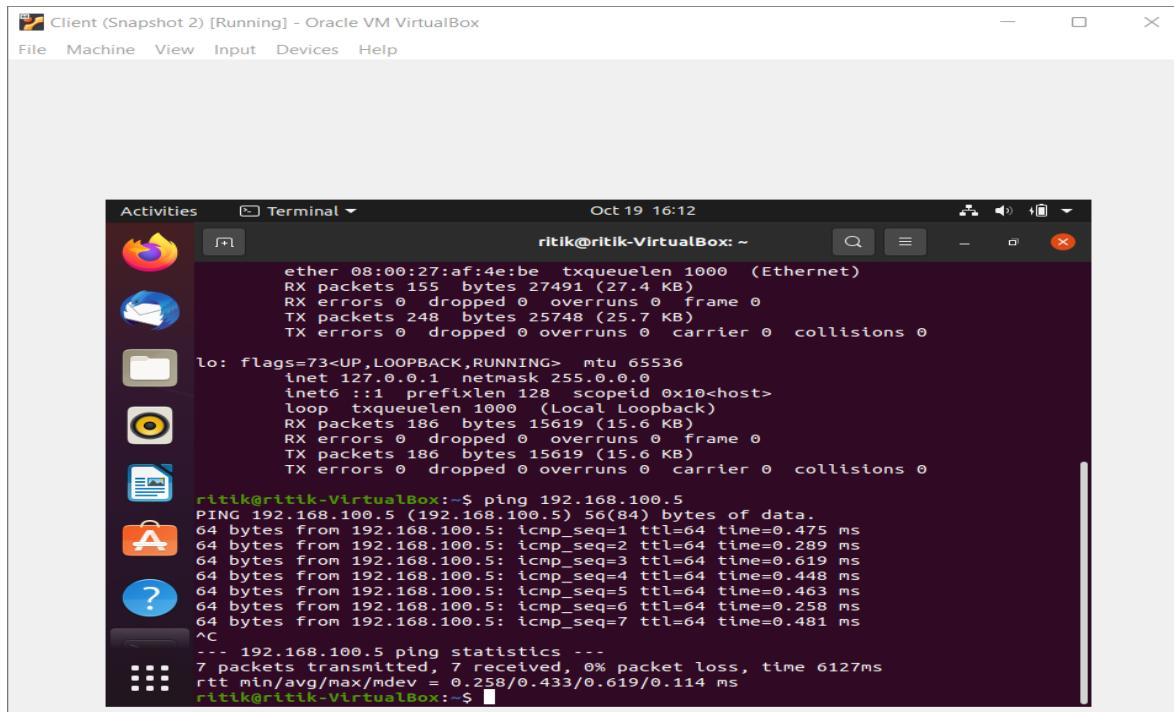
```

ritik@ritik-VirtualBox:/etc$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defau
lt qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP g
roup default qlen 1000
    link/ether 08:00:27:13:a6:8d brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.5/24 brd 192.168.100.255 scope global dynamic noprefixroute
      enp0s3
        valid_lft 592sec preferred_lft 592sec
    inet6 fe80::d18:3abf:eaef:4a20/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
ritik@ritik-VirtualBox:/etc$ 

```

### Step-3) Ping server machine from the client machine.





## **Part 2: Setting up host server**

### **Step 1: Install NFS Kernel Server**

1. Install the latest available version of a software through the Ubuntu repositories.

```
ritik@ritik-VirtualBox:~$ sudo apt-get update
[sudo] password for ritik:
Hit:1 http://in.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1, 284 kB]
Get:5 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu focal-updates/main i386 Packages [549 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [26 8 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metad ata [283 kB]
Get:9 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metada ta [14.4 kB]
Get:10 http://in.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packa ges [516 kB]
Get:11 http://in.archive.ubuntu.com/ubuntu focal-updates/restricted Translation -en [74.0 kB]
```

2. Install the NFS Kernel Server on your system

```
ritik@ritik-VirtualBox:~$ sudo apt install nfs-kernel-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  nfs-kernel-server
0 upgraded, 1 newly installed, 0 to remove and 97 not upgraded.
Need to get 98.9 kB of archives.
After this operation, 420 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 nfs-kernel-s erver amd64 1:1.3.4-2.5ubuntu3.4 [98.9 kB]
Fetched 98.9 kB in 0s (385 kB/s)
Selecting previously unselected package nfs-kernel-server.
(Reading database ... 184110 files and directories currently installed.)
Preparing to unpack .../nfs-kernel-server_1%3a1.3.4-2.5ubuntu3.4_amd64.deb ...
```

## Step 2: Create Export Directory

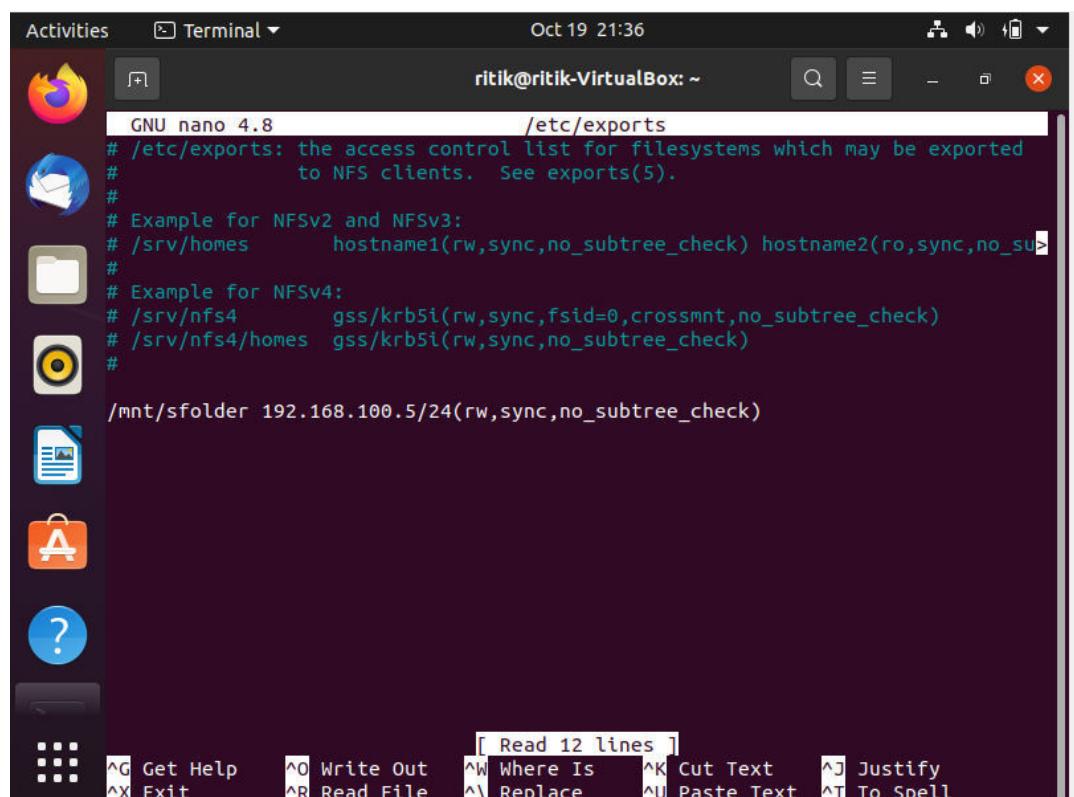
The directory that we want to share with the client system is called an export directory. Make a directory and change its permissions.

```
ritik@ritik-VirtualBox:~$ sudo mkdir -p /mnt/sfolder
ritik@ritik-VirtualBox:~$ sudo chown nobody:nogroup /mnt/sfolder
ritik@ritik-VirtualBox:~$ sudo chmod 777 /mnt/sfolder
```

## Step 3: Assign server access to client(s) through NFS export file

1. Type in sudo nano /etc/exports command in terminal.
2. Multiple clients, by specifying an entire subnet that the clients belong to:

/mnt/sharedfolder subnetIP/24(rw,sync,no\_subtree\_check)



A screenshot of a Linux desktop environment showing a terminal window. The terminal window title is "Terminal" and the date/time is "Oct 19 21:36". The terminal content shows the /etc/exports file being edited in nano. The file contains the following configuration:

```
GNU nano 4.8          /etc/exports
# /etc/exports: the access control list for filesystems which may be exported
#                   to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes      hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_su>
#
# Example for NFSv4:
# /srv/nfs4        gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes  gss/krb5i(rw,sync,no_subtree_check)
#
/mnt/sfolder 192.168.100.5/24(rw,sync,no_subtree_check)
```

The terminal window has a standard nano interface with various keyboard shortcuts at the bottom.

The permissions “rw, sync, no\_subtree\_check” permissions defined in this file mean that the client(s) can perform:

- o rw: read and write operations
- o sync: write any change to the disc before applying it
- o no\_subtree\_check: prevent subtree checking

## Step 4: Export the shared directory

Export the shared directory and restart the NFS Kernel server.

```
ritik@ritik-VirtualBox:~$ sudo exportfs -a  
ritik@ritik-VirtualBox:~$ sudo systemctl restart nfs-kernel-server
```

## Step 5: Open firewall for the client (s)

1. An important step is to verify that the server’s firewall is open to the clients so that they can access the shared content.
2. Check the status of your Ubuntu firewall.

```
ritik@ritik-VirtualBox:~$ sudo ufw allow from 192.168.100.5/24 to any port nfs  
WARN: Rule changed after normalization  
Rule added  
ritik@ritik-VirtualBox:~$ sudo ufw enable  
Firewall is active and enabled on system startup  
ritik@ritik-VirtualBox:~$ sudo ufw status  
Status: active  
  
To                         Action      From  
--                         -----      ---  
2049                       ALLOW       192.168.100.0/24
```

## Part 3: Configuring the Client Machine

### **Step 1: Install NFS Common**

Run the following command in order to install the NFS Common client on your system.

```
ritik@ritik-VirtualBox:~$ sudo apt-get install nfs-common
[sudo] password for ritik:
Reading package lists... Done
Building dependency tree
Reading state information... Done
nfs-common is already the newest version (1:1.3.4-2.5ubuntu3.4).
nfs-common set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 94 not upgraded.
```

### **Step 2: Create a mount point for the NFS host's shared folder and mount the shared directory on the client**

Create a mount folder in the mnt directory of our client's machine and mount the shared folder from the host to a mount folder on the client.

```
ritik@ritik-VirtualBox:~$ sudo mkdir -p /mnt/sfolder_client
ritik@ritik-VirtualBox:~$ sudo mount 192.168.100.4:/mnt/sfolder /mnt/sfolder_client
```

### **Step 3: Test the connection**

1. Please create or save a file in the export folder of the NFS host server

```
ritik@ritik-VirtualBox:/mnt/sfolder$ ls
file1.txt
ritik@ritik-VirtualBox:/mnt/sfolder$ touch ritik.txt
ritik@ritik-VirtualBox:/mnt/sfolder$ ls -l
total 0
-rw-rw-r-- 1 ritik ritik 0 Oct 19 21:49 file1.txt
-rw-rw-r-- 1 ritik ritik 0 Oct 19 21:51 ritik.txt
ritik@ritik-VirtualBox:/mnt/sfolder$
```

2. Open the mount folder on the client machine; you should be able to view the same file shared and accessible in this folder.

```
ritik@ritik-VirtualBox:~$ cd /mnt/sfolder_client
ritik@ritik-VirtualBox:/mnt/sfolder_client$ ls -l
total 0
-rw-rw-r-- 1 ritik ritik 0 Oct 19 21:49 file1.txt
-rw-rw-r-- 1 ritik ritik 0 Oct 19 21:51 ritik.txt
ritik@ritik-VirtualBox:/mnt/sfolder_client$
```

## **RESULTS AND CONCLUSIONS:**

NFS Server and Client have been successfully configured.