

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:

Shefali Bansal
A2305218263
B. Tech (CSE)
7CSE4Y

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

INDEX

Exp. No.	Name of Experiment	Date of Allotment	Date of Evaluation	Sign. of Faculty
1.	Installation of Unix/Linux operating system.	20-07-21	27-07-21	
2.	Study of Unix/Linux general purpose utility command list obtained from (man, who, cat, cd, cp, ps, ls, mv, rm, mkdir, rmdir, echo, more, date, time, kill, history, chmod, chown, finger, pwd, cal, logout, shutdown)	27-07-21	03-08-21	
3.	Study of VI Editor.	03-08-21	10-08-21	
4.	Study of Bash shell, Bourne shell and C shell in Unix/Linux operating system.	10-08-21	12-08-21	
5.	Write a shell script program to display HELLO WORLD. Write a shell script program to create a simple calculator. Write a shell script program to identify a given number as odd or even. Write a shell script program to find a given number in the list.	12-08-21	17-08-21	
6.	Implementation of Docker on Linux operating system.	17-08-21	31-08-21	
7.	<p>A) <u>Implementation of shell script programs.</u></p> <ol style="list-style-type: none"> 1. Check whether the given file is a directory or not. 2. Count the number of files in the directory. 3. Copy contents of one file to another. <p>B) <u>Implementation of shell script programs.</u></p> <ol style="list-style-type: none"> 1. Create a directory, write contents on that and Copy to a suitable location in your home directory. 2. Use a pipeline and command substitution to set the length of a line in file to a variable. 3. Write a program using sed command to print duplicate lines of Input. 	31-08-21	07-09-21	
8.	Write a program to illustrate the implementation of following: <ol style="list-style-type: none"> a. Integer Comparison b. String comparison c. Logical operators d. File tests e. Conditional control structure f. Loop control structures 	07-09-21	21-09-21	

9.	<p>Design an application by using various logical, conditional and arithmetic operators.</p> <ol style="list-style-type: none"> 1. Write a shell script to print a number in reverse order. 2. Find the sum of all numbers in a file in Linux. 3. Write a shell script to delete the lines containing a word <dd> if it appears between the 5th and 7th position. 4. 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. 5. Write a shell script to validate password strength. Here are a few assumptions for the password string. Length – minimum of 8 characters, contain both alphabet and number, Include both the small and capital case letters. 6. Write a shell script to reverse the list of strings and reverse each string further in the list. 	21-09-21	28-09-21	
10.	Design a real life use case of Shell Scripting -- Shell Script to Scrap the Definition of a Word From Wikipedia	28-09-21	05-10-21	
11.	Design and develop a “Birthday Reminder” that can automatically send birthday wishes with a personalized message via email.	05-10-21	12-10-21	
12.	Implement NIC Bonding and teaming.	12-10-21	19-10-21	
13.	Configure NFS in Linux and share a file through it.	19-10-21	02-11-21	

EXPERIMENT 1

AIM: Installation of Oracle VirtualBox and UbuntuOS based on Linux.

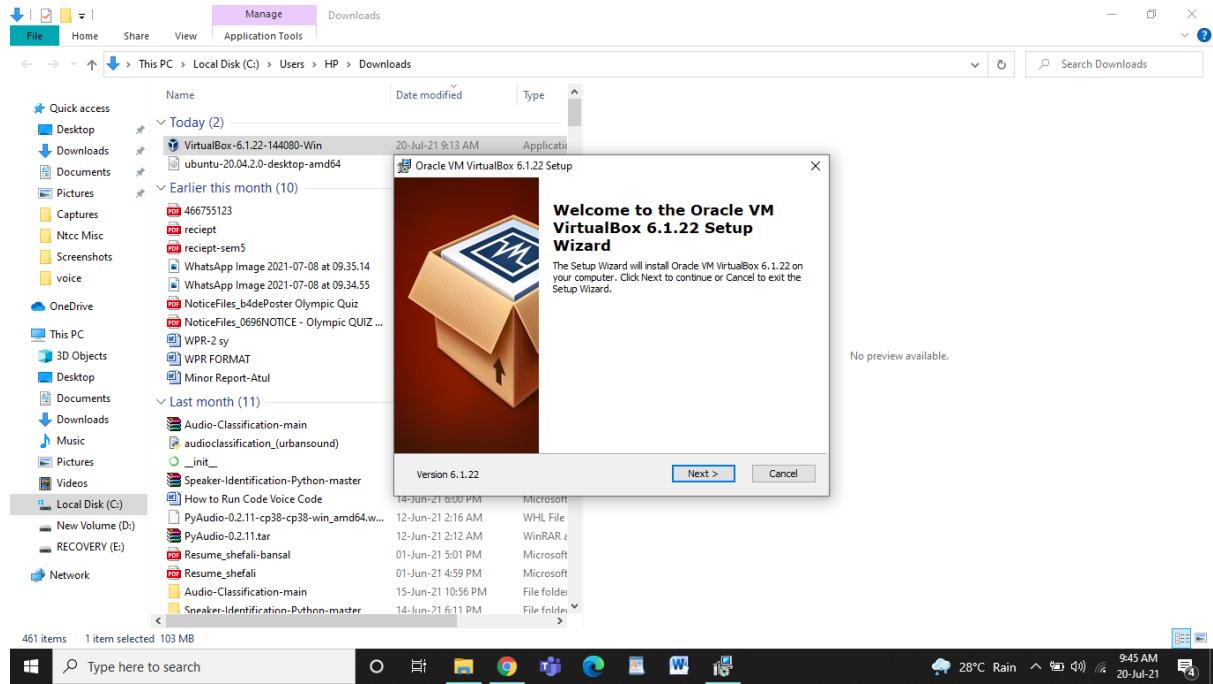
THEORY:

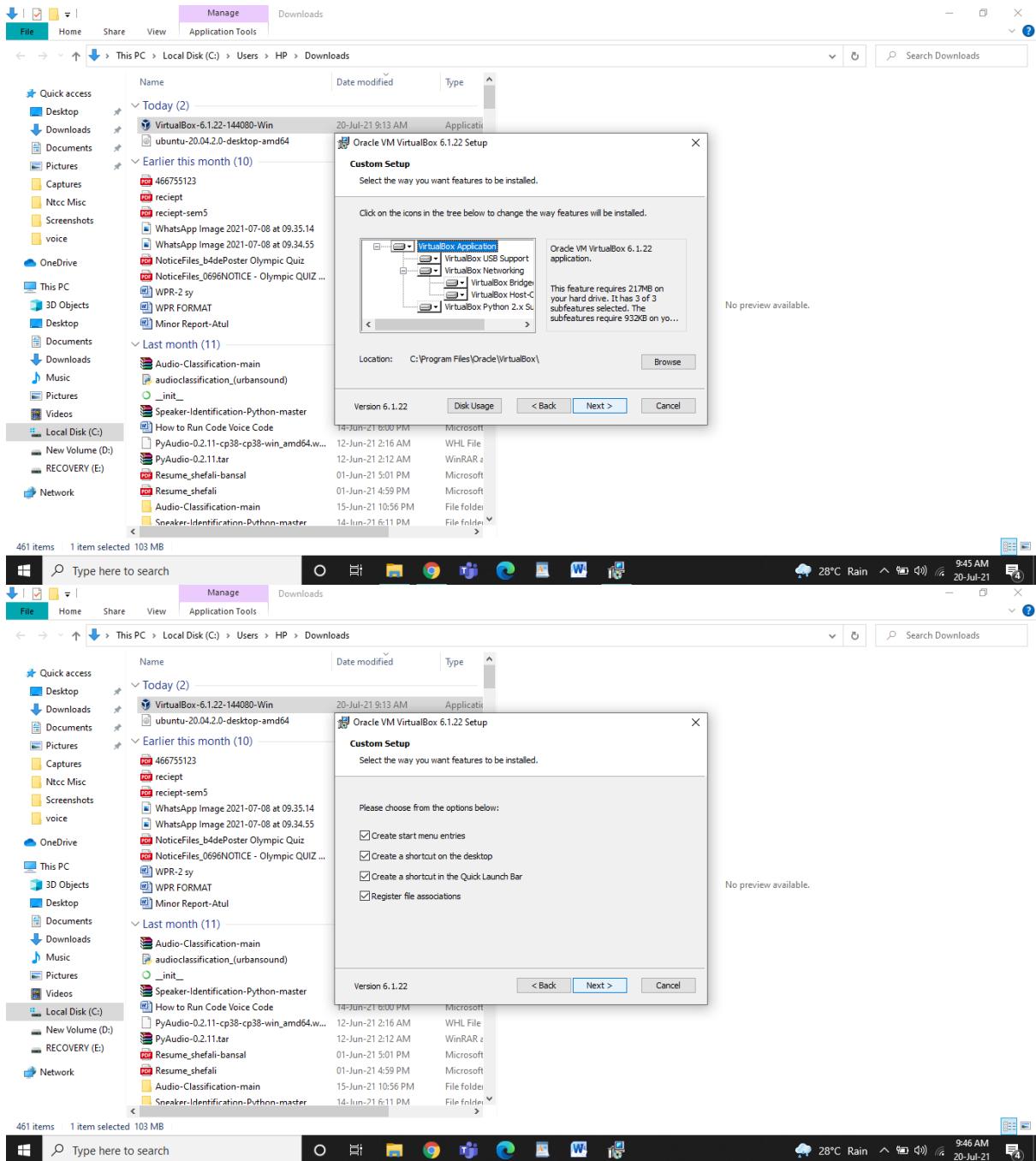
Oracle VM VirtualBox is a free and open-source hosted hypervisor for x86 virtualization, developed by Oracle Corporation. It may be installed on Windows, macOS, Linux, Solaris and OpenSolaris. It supports the creation and management of guest virtual machines, as well as limited virtualization of macOS guests on Apple hardware.

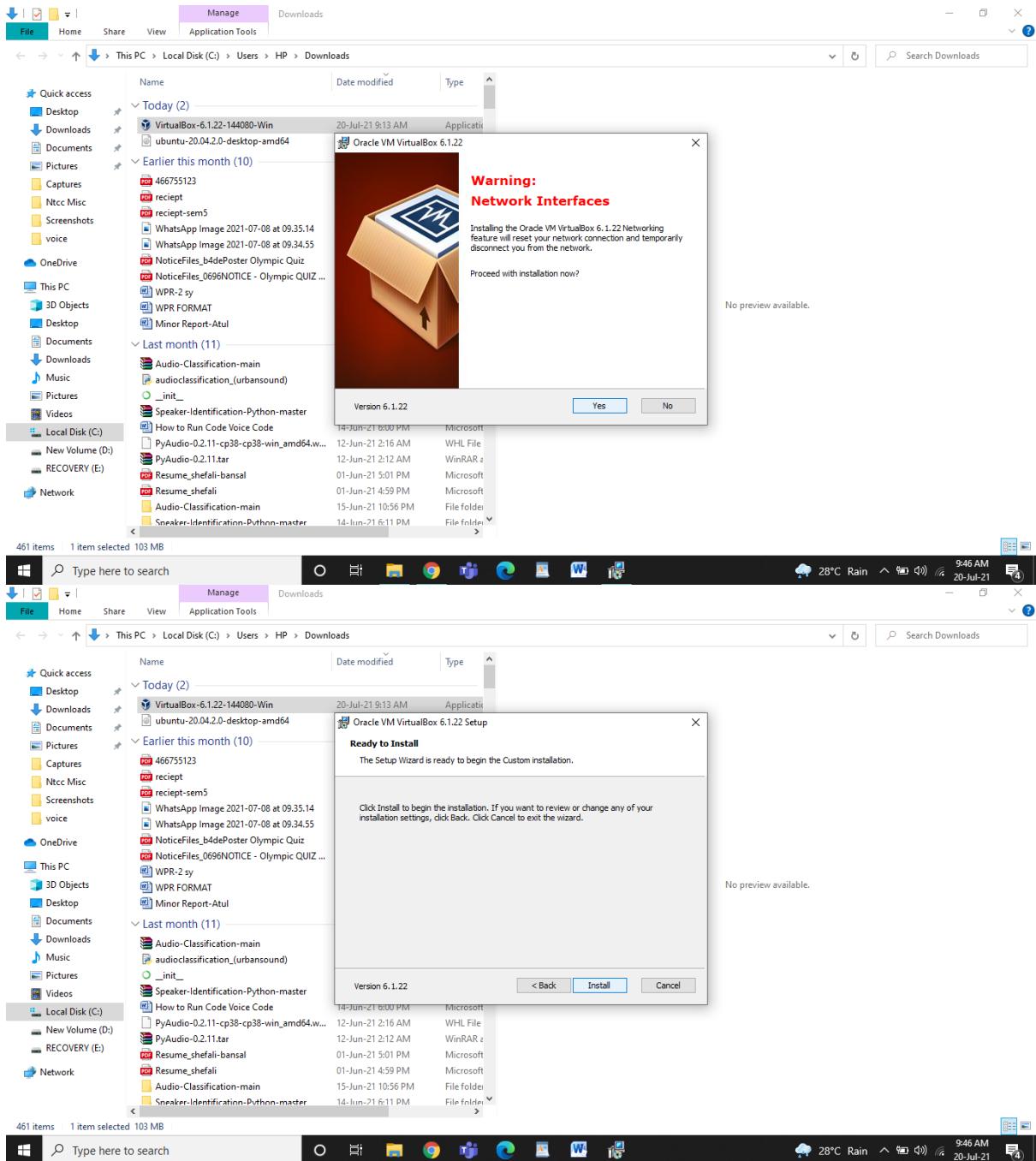
Linux is an open source operating system. Besides the Linux distributions designed for general-purpose use on desktops and servers, distributions may be specialized for different purposes including: computer architecture support, embedded systems, stability, security, localization to a specific region or language, targeting of specific user groups, support for real-time applications, or commitment to a given desktop environment.

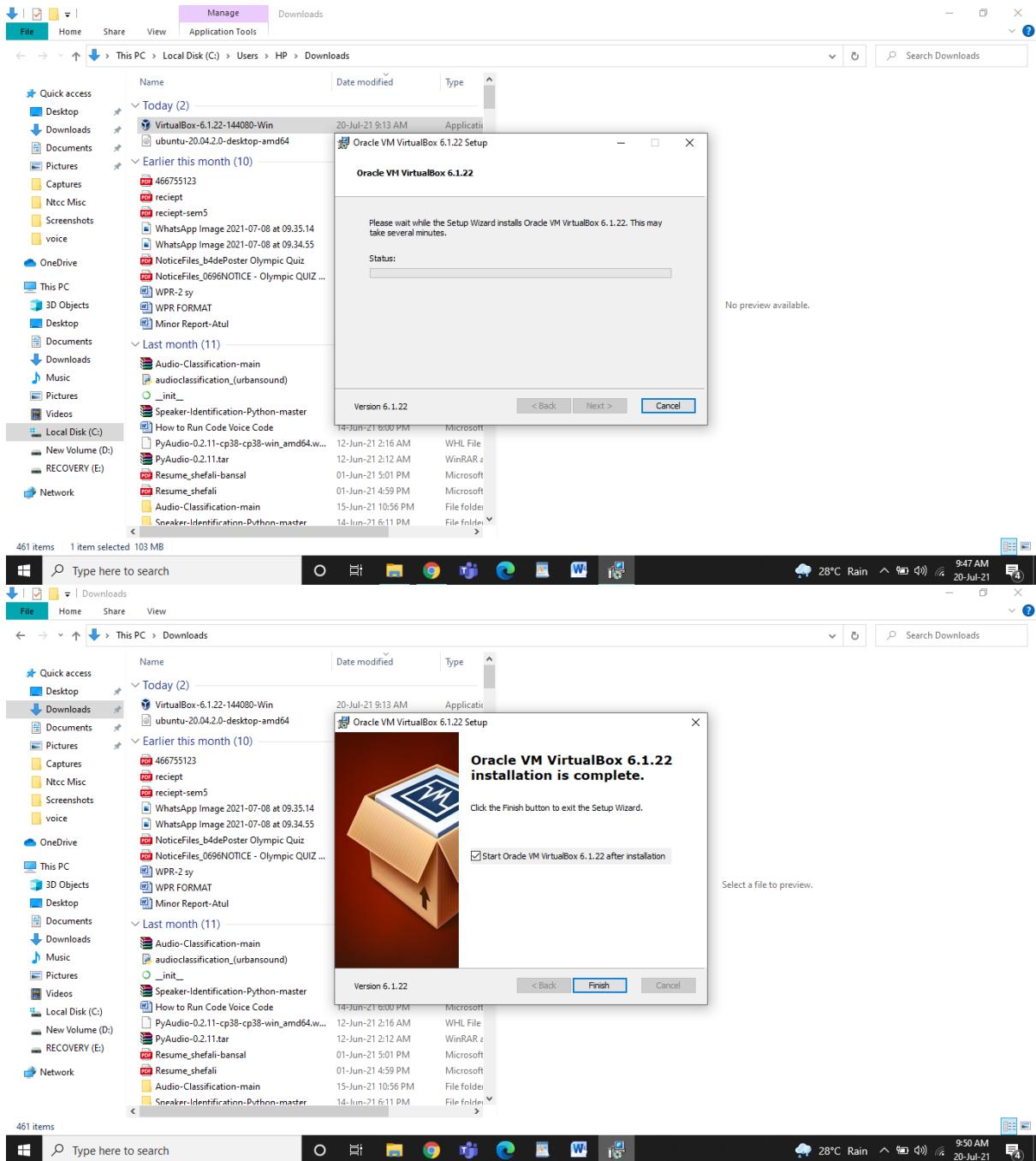
Ubuntu is an open source Debian-based Linux distribution. Sponsored by Canonical Ltd., Ubuntu is considered a good distribution for beginners. The operating system was intended primarily for personal computers but it can also be used on servers.

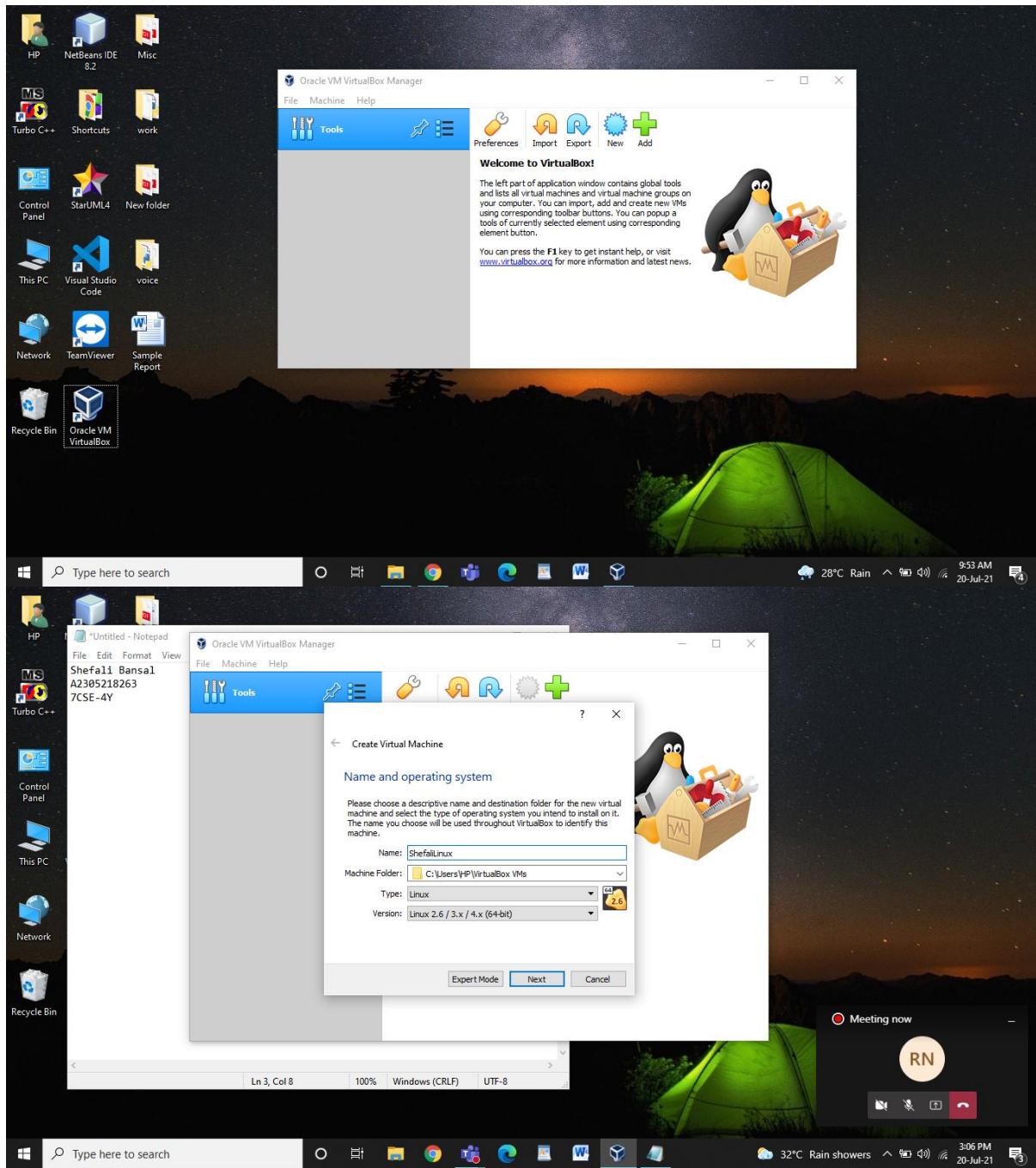
PROCEDURE:

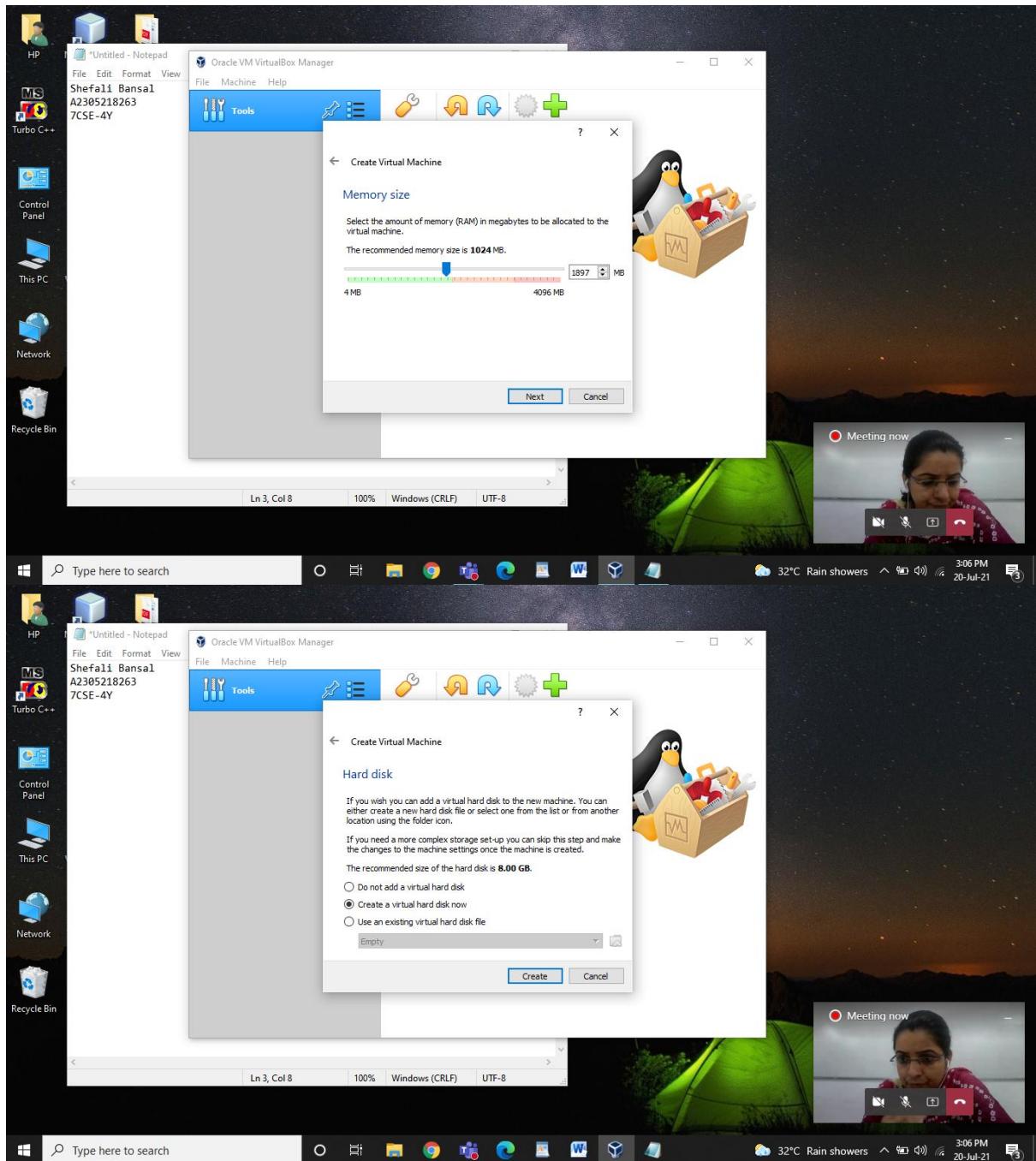


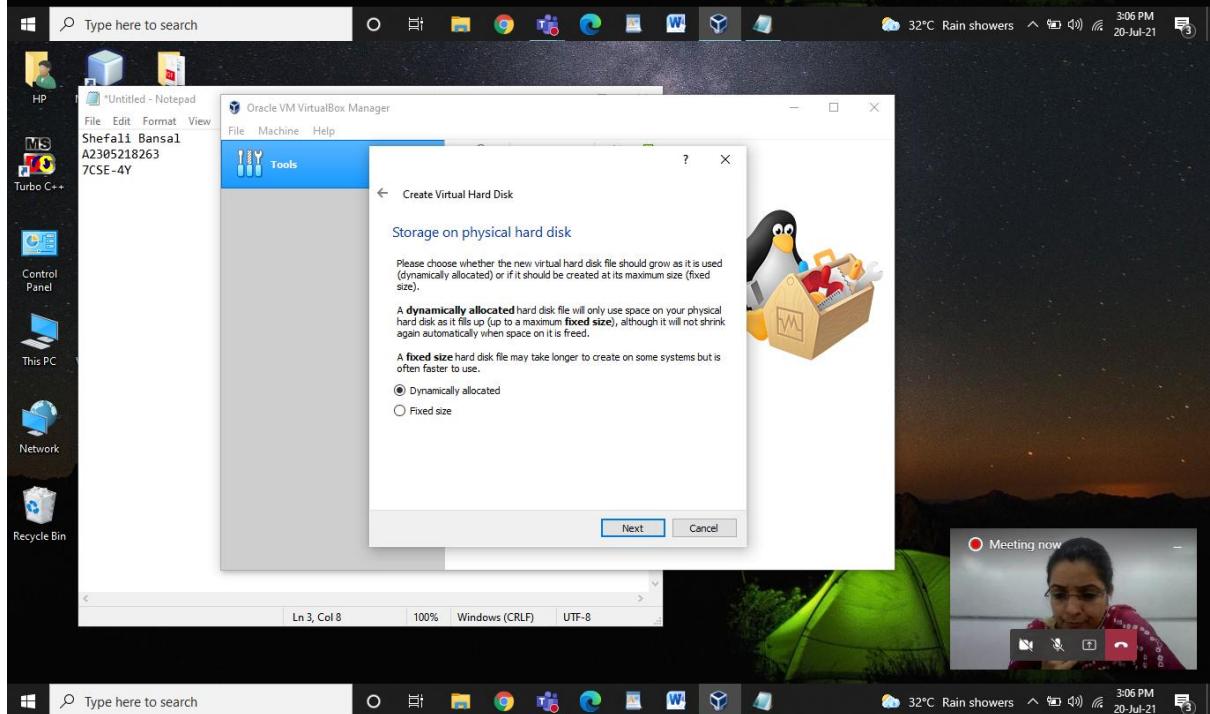
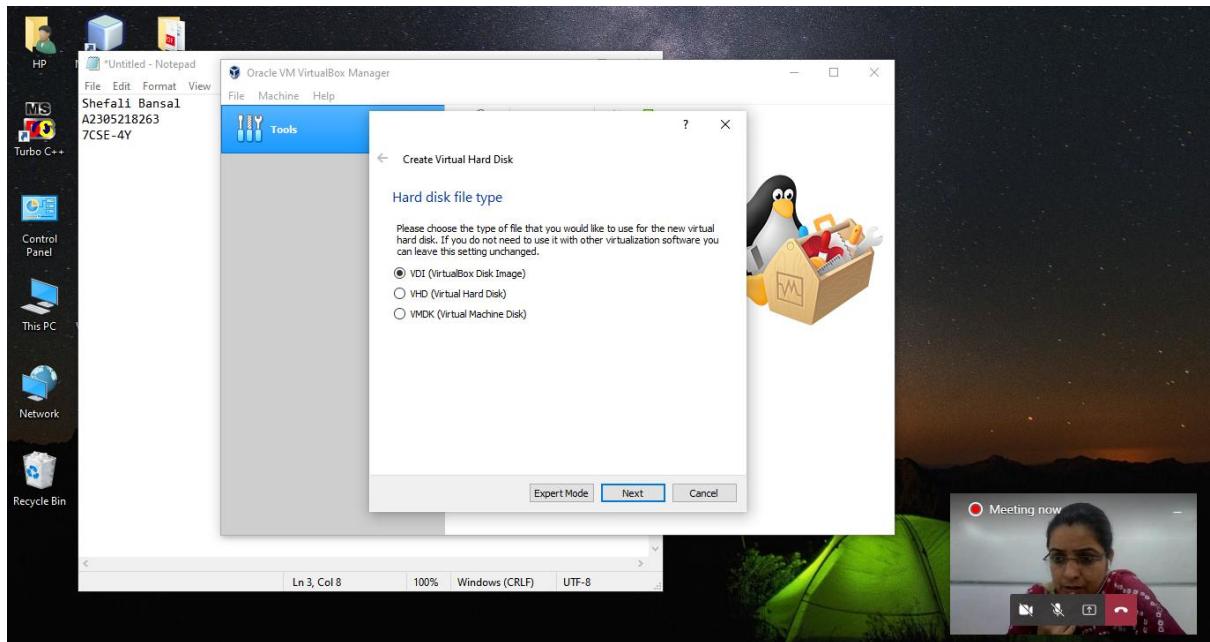


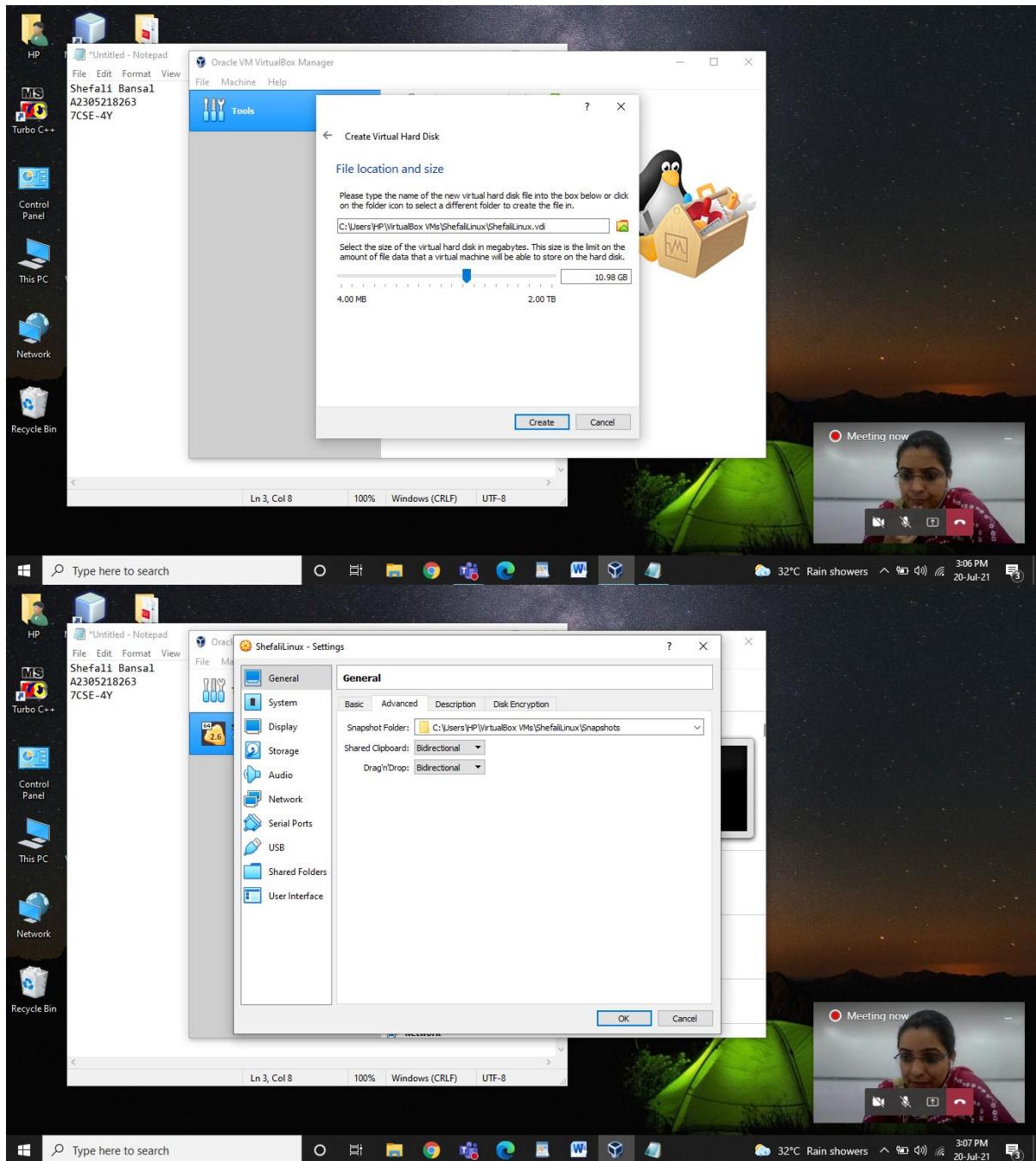


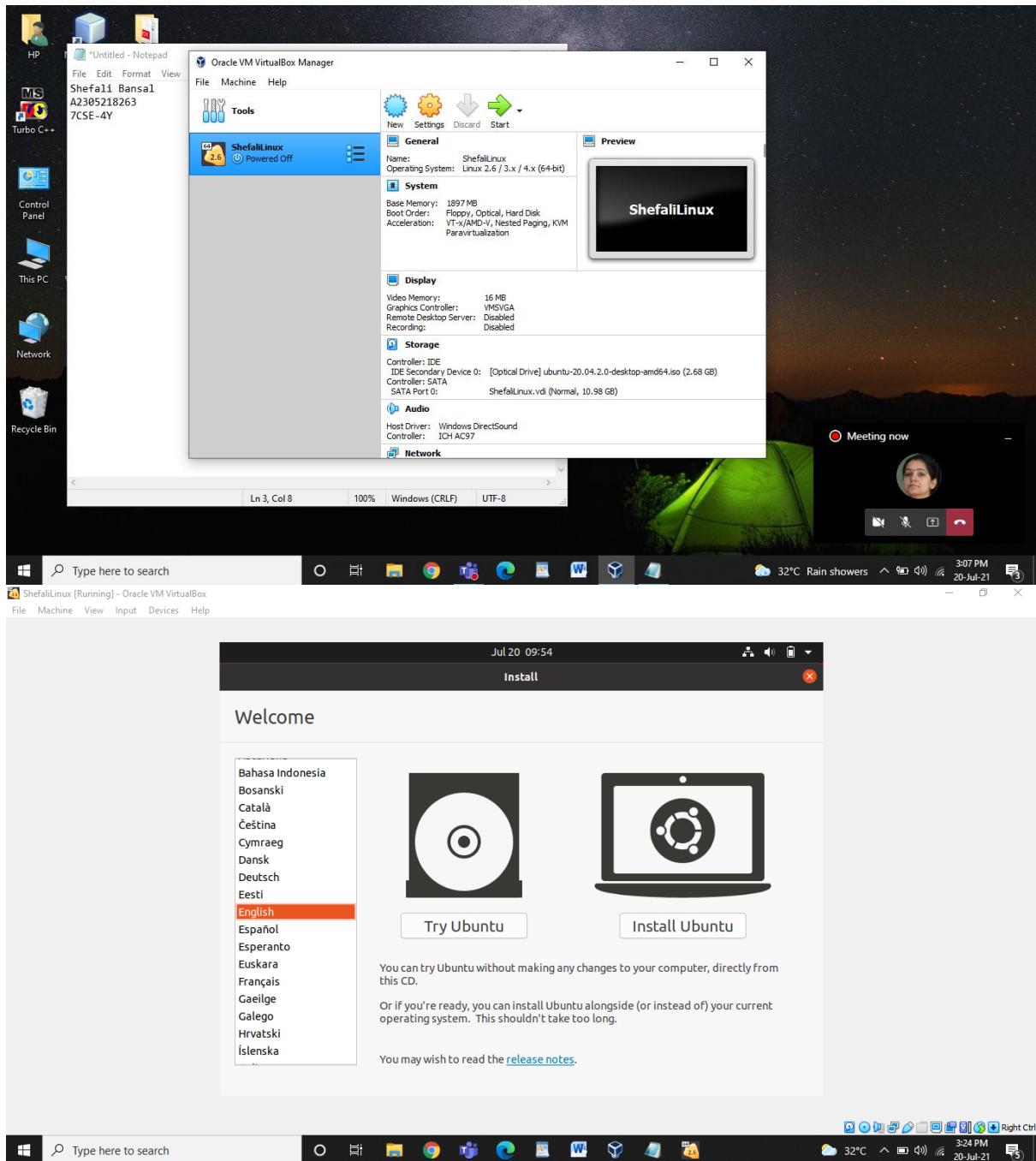


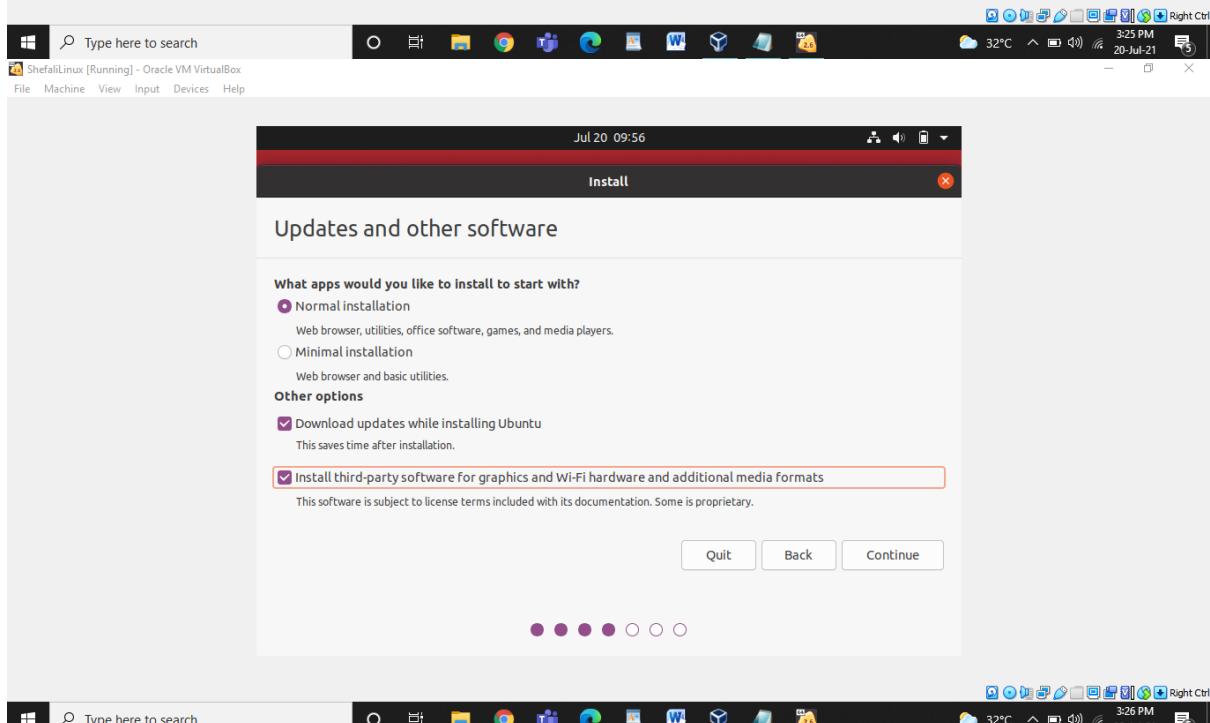
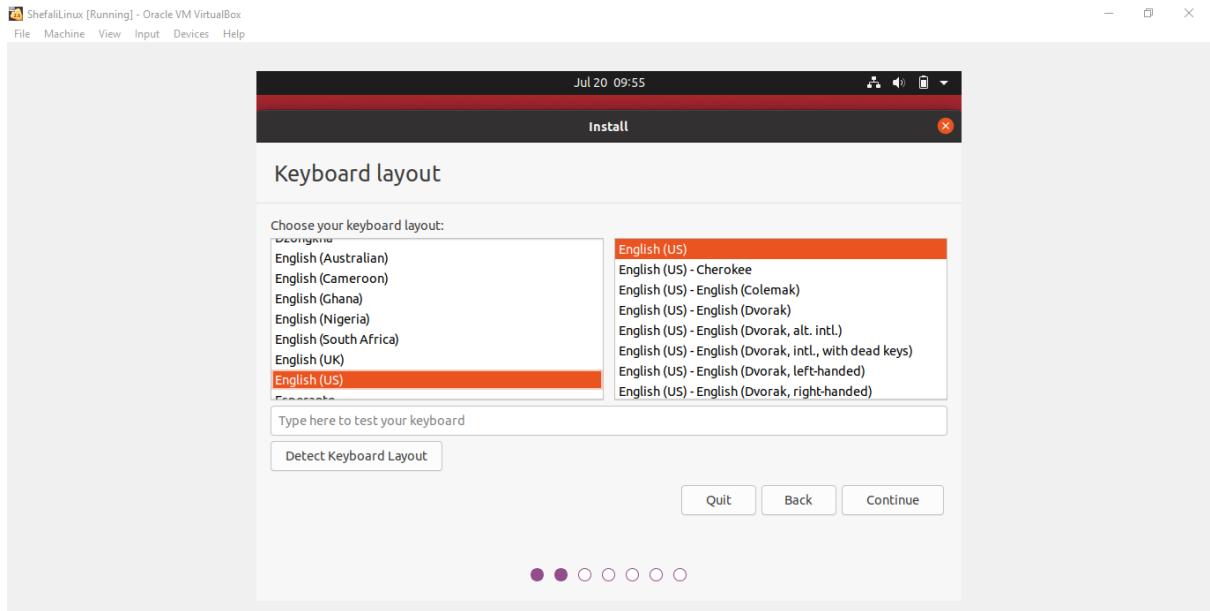


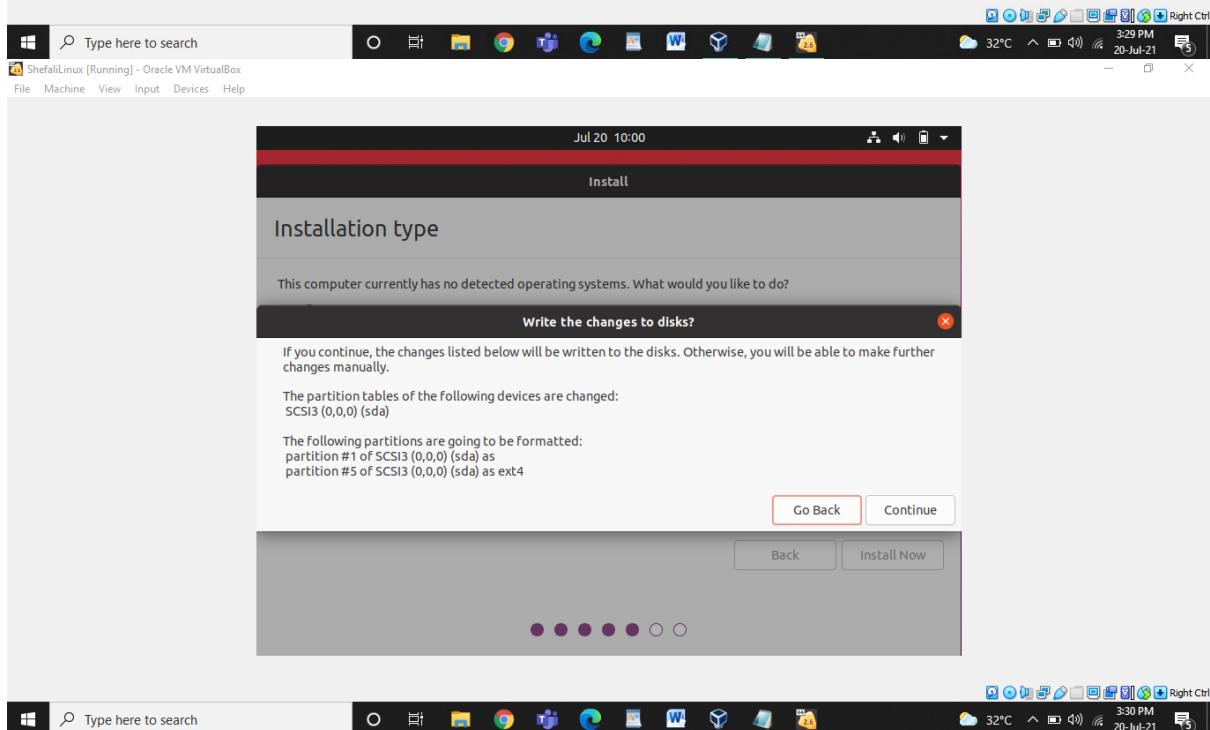
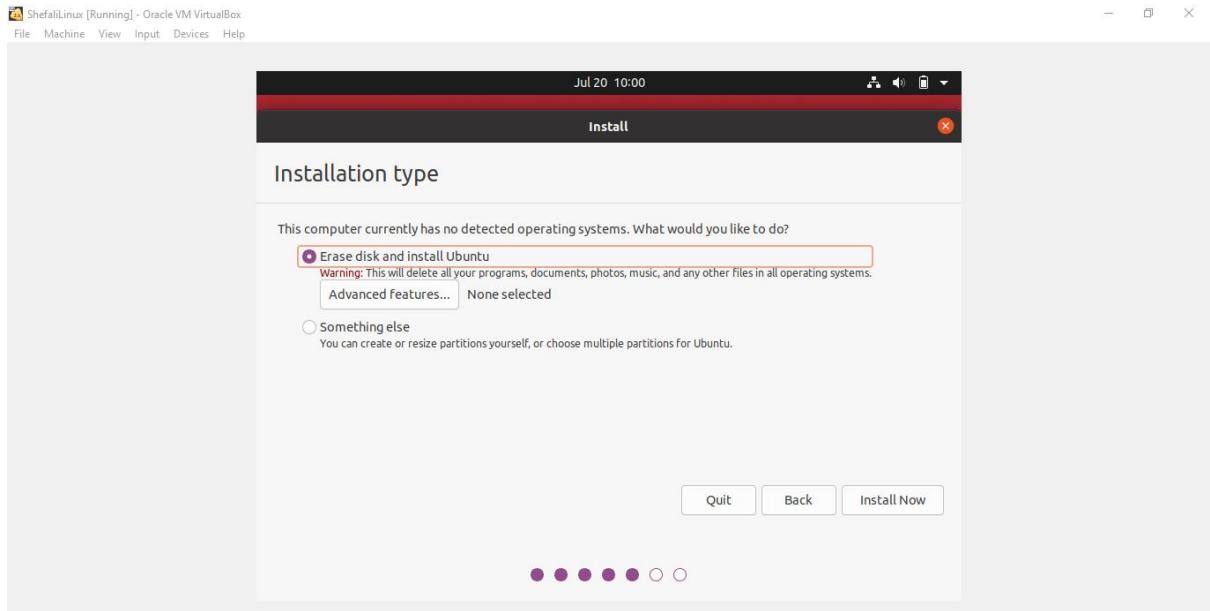


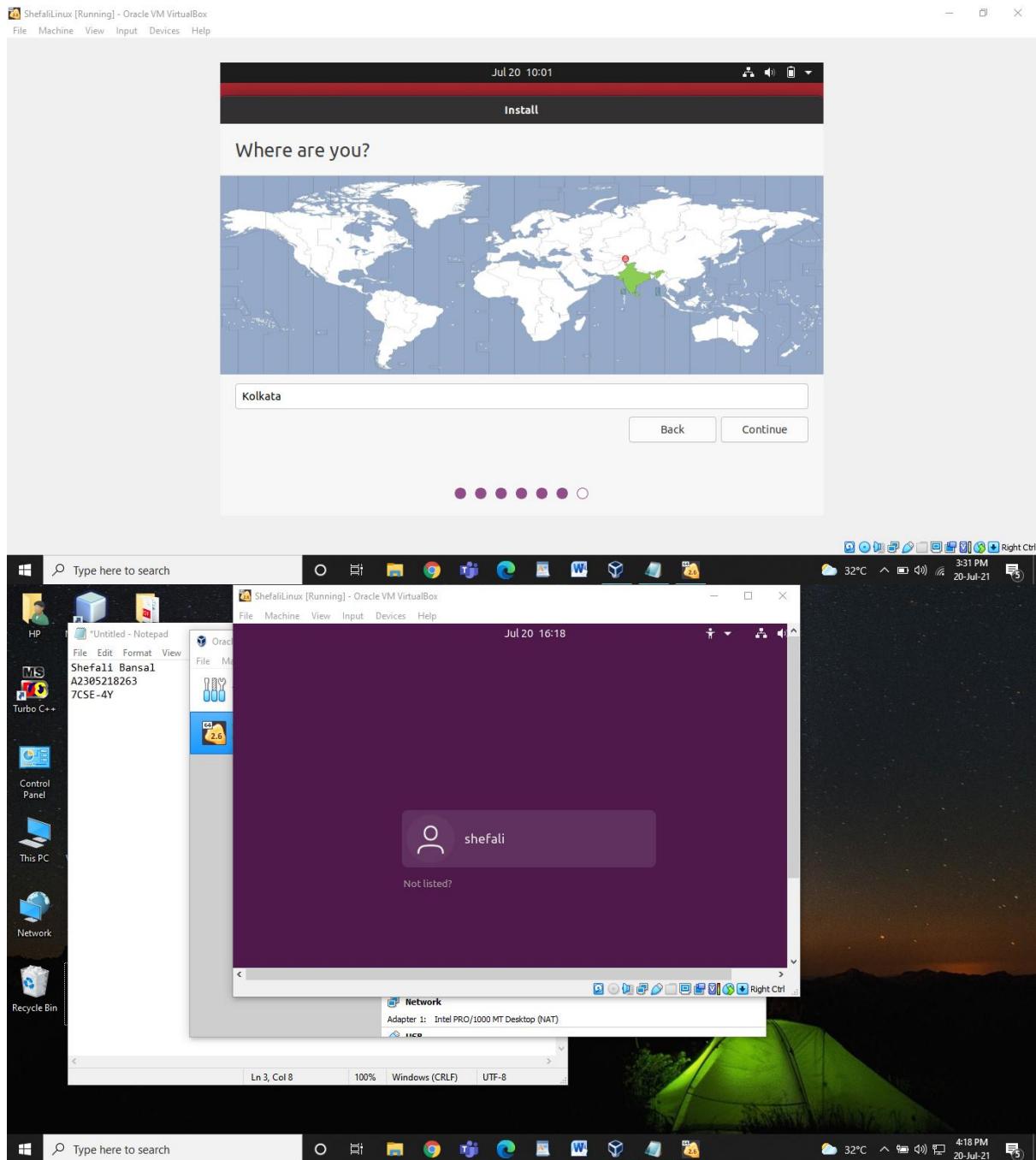


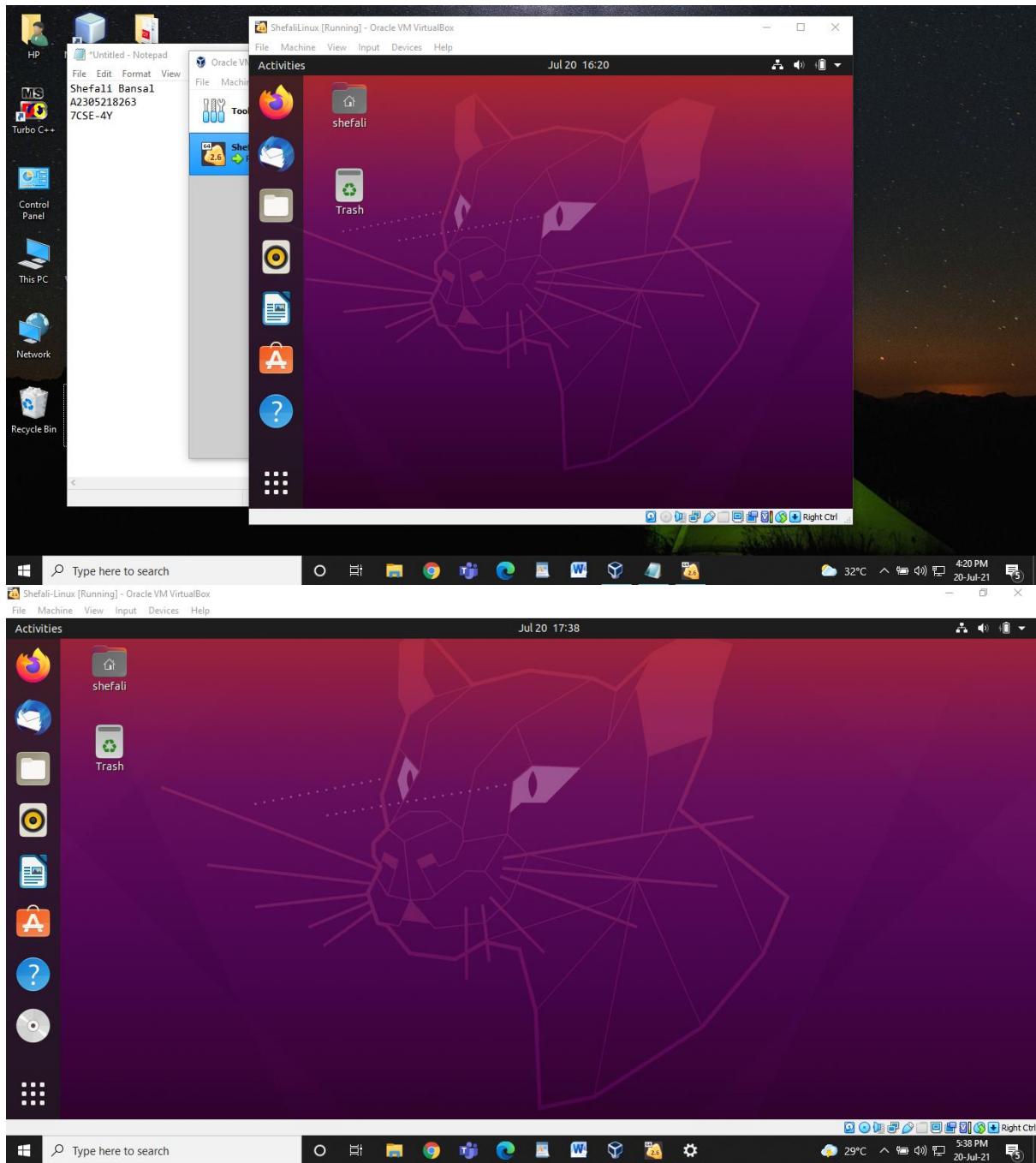












RESULT: Virtual Box and UbuntuOS are successfully installed on the system.

EXPERIMENT 2

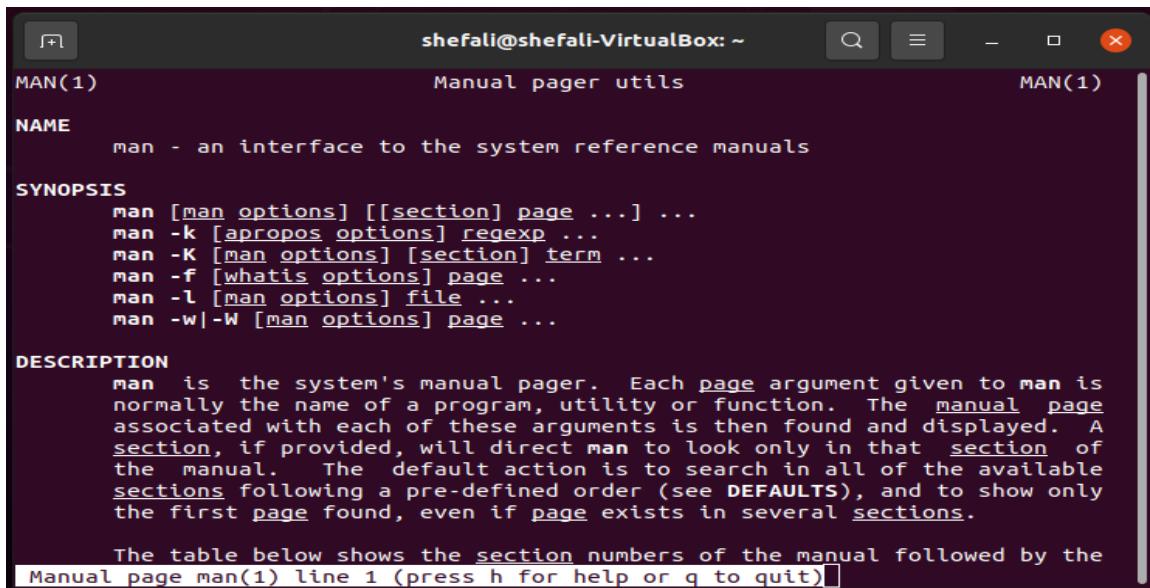
AIM: Study of Unix/Linux general purpose utility command list obtained from (man, who, cat, cd, cp, ps, ls, mv, rm, mkdir, rmdir, echo, more, date, time, kill, history, chmod, chown, finger, pwd, cal, logout, shutdown).

THEORY:

The Linux command is **a utility of the Linux operating system**. All basic and advanced tasks can be done by executing commands. The commands are executed on the Linux terminal. The terminal is a command-line interface to interact with the system, which is similar to the command prompt in the Windows OS.

PROCEDURE:

1. MAN: man command in Linux is used to display the user manual of any command that we can run on the terminal.



The screenshot shows a terminal window with the title bar "shefali@shefali-VirtualBox: ~". The window displays the man(1) manual page for the "man" command. The page is divided into sections: NAME, SYNOPSIS, and DESCRIPTION. The SYNOPSIS section lists various command-line options for "man". The DESCRIPTION section provides a detailed explanation of what "man" does, mentioning it's a manual pager and how it handles multiple sections. At the bottom of the page, there's a note about section numbers and a prompt to press 'h' for help or 'q' to quit.

```
shefali@shefali-VirtualBox: ~
MAN(1)                               Manual pager utils                               MAN(1)

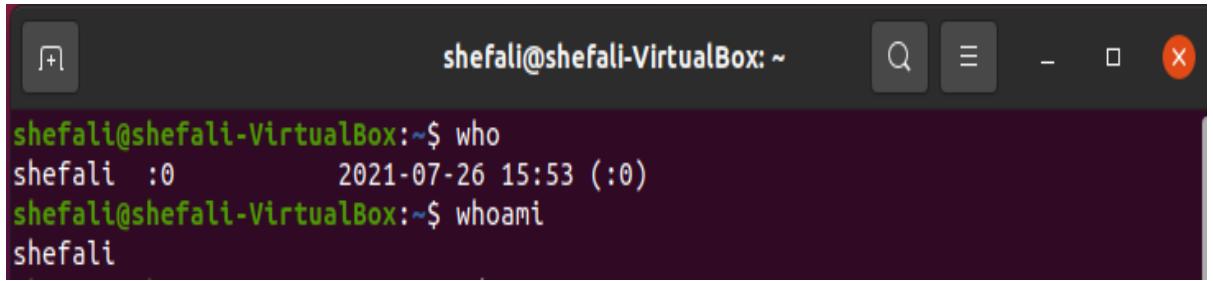
NAME
    man - an interface to the system reference manuals

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

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

    The table below shows the section numbers of the manual followed by the
    Manual page man(1) line 1 (press h for help or q to quit)
```

2. WHO: The who command displays the following information for each user currently logged in to the system if no option is provided :
 - Login name of the users.
 - Terminal line numbers.
 - Login time of the users into the system.
 - Remote host name of the user.
3. WHOAMI:
 - It is basically the concatenation of the strings "who", "am", "i" as whoami.
 - It displays the username of the current user when this command is invoked.
 - It is similar to running the id command with the options -un.



```
shefali@shefali-VirtualBox:~$ who
shefali :0          2021-07-26 15:53 (:0)
shefali@shefali-VirtualBox:~$ whoami
shefali
```

4. LS: It is a Linux shell command that lists directory contents of files and directories.

ls -t : It sorts the file by modification time, showing the last edited file first. head -1 picks up this first file. To open the last edited file in the current directory use the combination of ls and head commands as shown below.

ls -l : To show long listing information about the file/directory.

ls -S : It sorts the file by size

ls -h: It shows the file size in human readable format. For example 10K, 12M, 2G

ls -lt : List files by time in reverse order with long listing

ls -r: It lists the files in current directory and its subdirectories recursively

ls -a: It lists all the hidden files

“*” can be used as a wildcard in unix/linux

```
shefali@shefali-VirtualBox:~$ ls -l
total 32
drwxr-xr-x 2 shefali shefali 4096 Jul 20 16:18 Desktop
drwxr-xr-x 2 shefali shefali 4096 Jul 20 16:18 Documents
drwxr-xr-x 2 shefali shefali 4096 Jul 20 16:18 Downloads
drwxr-xr-x 2 shefali shefali 4096 Jul 20 16:18 Music
drwxr-xr-x 2 shefali shefali 4096 Jul 20 16:18 Pictures
drwxr-xr-x 2 shefali shefali 4096 Jul 20 16:18 Public
drwxr-xr-x 2 shefali shefali 4096 Jul 20 16:18 Templates
drwxr-xr-x 2 shefali shefali 4096 Jul 20 16:18 Videos
shefali@shefali-VirtualBox:~$ ls -t
Desktop Documents Downloads Music Pictures Public Templates Videos
shefali@shefali-VirtualBox:~$ ls -S
Desktop Documents Downloads Music Pictures Public Templates Videos
shefali@shefali-VirtualBox:~$ ls -h
Desktop Documents Downloads Music Pictures Public Templates Videos
shefali@shefali-VirtualBox:~$ ls -h
Desktop Documents Downloads Music Pictures Public Templates Videos
shefali@shefali-VirtualBox:~$ ls -lt
total 32
drwxr-xr-x 2 shefali shefali 4096 Jul 20 16:18 Desktop
drwxr-xr-x 2 shefali shefali 4096 Jul 20 16:18 Documents
drwxr-xr-x 2 shefali shefali 4096 Jul 20 16:18 Downloads
drwxr-xr-x 2 shefali shefali 4096 Jul 20 16:18 Music
drwxr-xr-x 2 shefali shefali 4096 Jul 20 16:18 Pictures
drwxr-xr-x 2 shefali shefali 4096 Jul 20 16:18 Public
drwxr-xr-x 2 shefali shefali 4096 Jul 20 16:18 Templates
drwxr-xr-x 2 shefali shefali 4096 Jul 20 16:18 Videos
shefali@shefali-VirtualBox:~$ ls -r
Videos Templates Public Pictures Music Downloads Documents Desktop
shefali@shefali-VirtualBox:~$ ls -a
. Downloads .sudo_as_admin_successful
.. .gnupg Templates
.bash_history .local .vboxclient-clipboard.pid
.bash_logout .mozilla .vboxclient-display-svga-x11.pid
.bashrc Music .vboxclient-draganddrop.pid
.cache Pictures .vboxclient-seamless.pid
.config .profile Videos
Desktop Public
Documents .ssh
shefali@shefali-VirtualBox:~$
```



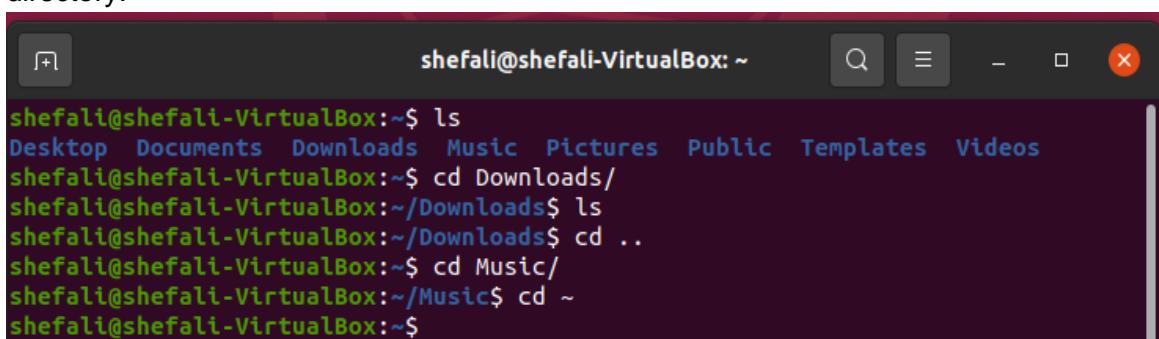
```
shefali@shefali-VirtualBox:~$ ls *
Desktop:
Documents:
Downloads:
Music:
Pictures:
Public:
Templates:
Videos:
shefali@shefali-VirtualBox:~$
```

```
shefali@shefali-VirtualBox:~$ ls -lr
total 32
drwxr-xr-x 2 shefali shefali 4096 Jul 20 16:18 Videos
drwxr-xr-x 2 shefali shefali 4096 Jul 20 16:18 Templates
drwxr-xr-x 2 shefali shefali 4096 Jul 20 16:18 Public
drwxr-xr-x 2 shefali shefali 4096 Jul 20 16:18 Pictures
drwxr-xr-x 2 shefali shefali 4096 Jul 20 16:18 Music
drwxr-xr-x 2 shefali shefali 4096 Jul 20 16:18 Downloads
drwxr-xr-x 2 shefali shefali 4096 Jul 20 16:18 Documents
drwxr-xr-x 2 shefali shefali 4096 Jul 20 16:18 Desktop
shefali@shefali-VirtualBox:~$
```

5. CD: It is known as change directory command. It is used to change the current working directory.

cd ~ : this command is used to change directory to the home directory.

cd .. : this command is used to move to the parent directory of the current directory, or the directory one level up from the current directory. “..” represents the parent directory.



A screenshot of a terminal window titled "shefali@shefali-VirtualBox: ~". The window shows a series of commands and their outputs:

```
shefali@shefali-VirtualBox:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
shefali@shefali-VirtualBox:~$ cd Downloads/
shefali@shefali-VirtualBox:~/Downloads$ ls
shefali@shefali-VirtualBox:~/Downloads$ cd ..
shefali@shefali-VirtualBox:~/Music$ cd /
shefali@shefali-VirtualBox:/$ cd ~
shefali@shefali-VirtualBox:~$
```

6. MKDIR: It allows the user to create directories (also referred to as folders in some operating systems). This command can create multiple directories at once as well as set the permissions for the directories. It is important to note that the user executing this command must have enough permission to create a directory in the parent directory, or he/she may receive a ‘permission denied’ error.
7. RMDIR: It is used to remove empty directories from the file system in Linux. The rmdir command removes each and every directory specified in the command line only if these directories are empty. So if the specified directory has some directories or files in it then this cannot be removed by *rmdir* command.

```
shefali@shefali-VirtualBox:~$ mkdir ShefaliB
shefali@shefali-VirtualBox:~$ ls
Desktop Downloads Pictures ShefaliB Videos
Documents Music Public Templates
shefali@shefali-VirtualBox:~$ rmdir ShefaliB
shefali@shefali-VirtualBox:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
```

8. VI EDITOR: To launch the VI Editor -Open the Terminal (CLI) and type vi <filename_NEW> or <filename_EXISTING>

And if you specify an existing file, then the editor would open it for you to edit. Else, you can create a new file.

- CAT: It is very frequently used in Linux. It reads data from the file and gives their content as output. It helps us to create, view, and concatenate files. So let us see some frequently used cat commands.

```
shefali@shefali-VirtualBox:~$ vi abc.txt
shefali@shefali-VirtualBox:~$ vi abc.txt
shefali@shefali-VirtualBox:~$ cat abc.txt
hello
1
2
3
4
5
6
7
8
8
:wq
```

10. **HEAD**: It prints the top N number of data of the given input. By default, it prints the first 10 lines of the specified files. If more than one file name is provided then data from each file is preceded by its file name.

-n num: Prints the first 'num' lines instead of the first 10 lines. num is mandatory to be specified in command otherwise it displays an error.

```
shefali@shefali-VirtualBox:~$ head abc.txt
hello
1
2
3
4
5
6
7
8
9
```

```
shefali@shefali-VirtualBox:~$ head -n5 abc.txt
hello
1
2
3
4
```

11. TAIL: It prints the last N number of data of the given input. By default it prints the last 10 lines of the specified files. If more than one file name is provided then data from each file is preceded by its file name

```
shefali@shefali-VirtualBox:~$ tail abc.txt
3
4
5
6
7
8
8
:wq

shefali@shefali-VirtualBox:~$ tail -n5 abc.txt
8
8
:wq

shefali@shefali-VirtualBox:~$
```

12. LESS: It is a linux utility which can be used to read the contents of a text file one page(one screen) per time. It has faster access because if a file is large, it doesn't access the complete file, but access it page by page.
Example- if it's a large file and you are reading it using any text editor, then the complete file will be loaded to main memory, but less commands don't load the entire file, but load it part by part, which makes it faster.

```
shefali@shefali-VirtualBox: ~
hello
1
2
3
4
5
6
7
8
:wq

abc.txt (END)
```

13. PWD: It stands for **P**rint **W**orking **D**irectory. It prints the path of the working directory, starting from the root. IT is a shell built-in command(pwd) or an actual binary(/bin/pwd).
\$PWD is an environment variable which stores the path of the current directory.

14. ECHO: Displays a line of text.

```
shefali@shefali-VirtualBox: ~$ echo The weather is nice today
The weather is nice today
shefali@shefali-VirtualBox: ~$ pwd
/home/shefali
```

15. CP: This command is used to copy files or groups of files or directory. It creates an exact image of a file on a disk with a different file name. cp command requires at least two filenames in its arguments.

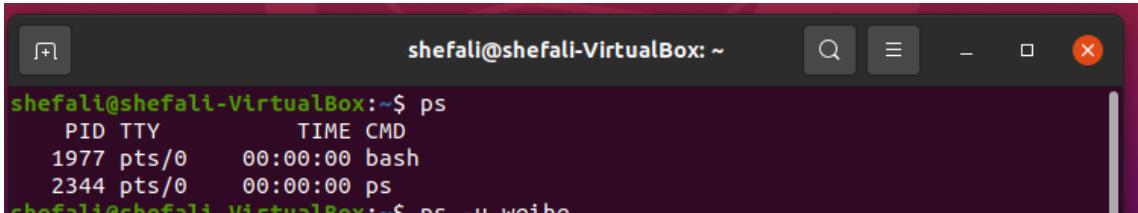
```
shefali@shefali-VirtualBox: ~$ ls -l abc.txt
-rw-rw-r-- 1 shefali shefali 30 Jul 27 14:25 abc.txt
shefali@shefali-VirtualBox: ~$ cp abc.txt new.txt
shefali@shefali-VirtualBox: ~$ cat new.txt
hello
1
2
3
4
5
6
7
8
8
:wq

shefali@shefali-VirtualBox: ~$
```

16. MV: mv is used to move one or more files or directories from one place to another in a file system like UNIX. It has two distinct functions:
- (i) It rename a file or folder.
 - (ii) It moves a group of files to a different directory.
17. RM: rm command is used to remove objects such as files, directories, symbolic links and so on from the file system like UNIX. To be more precise, rm removes references to objects from the file system, where those objects might have had multiple references (for example, a file with two different names).

```
shefali@shefali-VirtualBox:~$ mkdir new_dir
shefali@shefali-VirtualBox:~$ mv new.txt ./new_dir/
shefali@shefali-VirtualBox:~$ cd new_dir/
shefali@shefali-VirtualBox:~/new_dir$ ls
new.txt
shefali@shefali-VirtualBox:~/new_dir$ rm -r
rm: missing operand
Try 'rm --help' for more information.
shefali@shefali-VirtualBox:~/new_dir$ mv new.txt new1.txt
shefali@shefali-VirtualBox:~/new_dir$ ls
new1.txt
shefali@shefali-VirtualBox:~/new_dir$ rm new1.txt
shefali@shefali-VirtualBox:~/new_dir$ ls
shefali@shefali-VirtualBox:~/new_dir$
```

18. PS: Linux provides us a utility called **ps** for viewing information related with the processes on a system which stands as abbreviation for “**Process Status**”. ps command is used to list the currently running processes and their PIDs along with some other information depending on different options. It reads the process information from the virtual files in **/proc** file-system. /proc contains virtual files, this is the reason it's referred to as a virtual file system. ps provides numerous options for manipulating the output according to our need.



The screenshot shows a terminal window with the title bar "shefali@shefali-VirtualBox: ~". The command "ps" is run, and the output is as follows:

```
shefali@shefali-VirtualBox:~$ ps
 PID TTY      TIME CMD
 1977 pts/0    00:00:00 bash
 2344 pts/0    00:00:00 ps
shefali@shefali-VirtualBox:~$ ps -u shefali
```

19. TOP: It is used to show the Linux processes. It provides a dynamic real-time view of the running system. Usually, this command shows the summary information of the system and the list of processes or threads which are currently managed by the Linux Kernel. As soon as you run this command it will open an interactive command mode where the top half portion will contain the statistics of processes and resource usage. And Lower half contains a list of the currently running processes. Pressing **q** will simply exit the command mode.

```

top - 14:34:56 up 18 min, 1 user, load average: 0.12, 0.17, 0.26
Tasks: 189 total, 1 running, 188 sleeping, 0 stopped, 0 zombie
%Cpu(s): 4.8 us, 1.0 sy, 0.0 ni, 93.5 id, 0.7 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 1838.1 total, 314.0 free, 648.7 used, 875.3 buff/cache
MiB Swap: 495.3 total, 495.3 free, 0.0 used. 1024.0 avail Mem

      PID USER      PR  NI    VIRT    RES    SHR   S %CPU %MEM     TIME+ COMMAND
 1585 shefali    20   0 4194608 331428 123076 S  7.9 17.6  0:31.51 gnome-s+
 2350 shefali    20   0 20492   3732   3236 R  1.6  0.2  0:00.12 top
 1353 shefali    20   0 829228 62624  42500 S  1.0  3.3  0:12.62 Xorg
 1968 shefali    20   0 823292 51196  38712 S  1.0  2.7  0:07.44 gnome-t+
 1495 shefali    20   0 163996 2740   2372 S  0.3  0.1  0:03.79 VBoxCli+
 2332 root      20   0      0      0      0 I  0.3  0.0  0:00.22 kworker+
 1 root       20   0 102260 11484  8288 S  0.0  0.6  0:03.89 systemd
 2 root       20   0      0      0      0 S  0.0  0.0  0:00.00 kthreadd
 3 root       0 -20      0      0      0 I  0.0  0.0  0:00.00 rcu_gp
 4 root       0 -20      0      0      0 I  0.0  0.0  0:00.00 rcu_par+
 5 root       20   0      0      0      0 I  0.0  0.0  0:00.34 kworker+
 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  0:00.08 ksoftir+
11 root      20   0      0      0      0 I  0.0  0.0  0:00.73 rcu_sch+
12 root      rt   0      0      0      0 S  0.0  0.0  0:00.01 migrati+

```

20. CLEAR: clear is a computer operating system command which is used to bring the command line on top of the computer terminal.

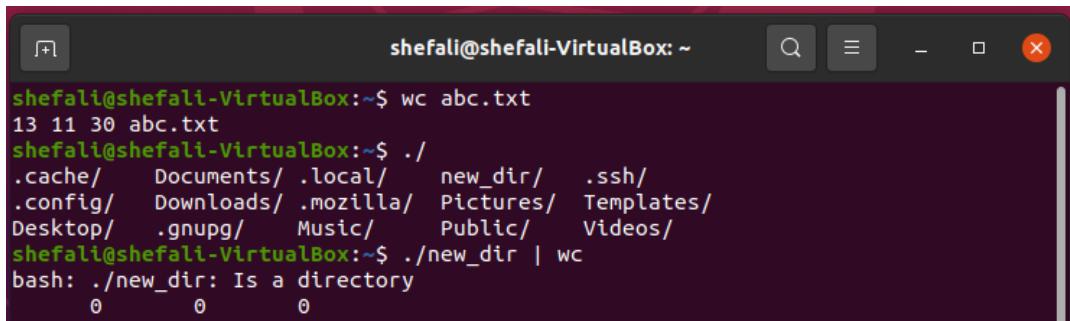
```

Tasks: 186 total, 1 running, 185 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.7 us, 0.2 sy, 0.0 ni, 99.2 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 1838.1 total, 310.8 free, 651.0 used, 876.3 buff/cache
MiB Swap: 495.3 total, 495.3 free, 0.0 used. 1021.1 avail Mem

      PID USER      PR  NI    VIRT    RES    SHR   S %CPU %MEM     TIME+ COMMAND
 1585 shefali    20   0 4194712 331832 123292 S  0.7 17.6  0:40.33 gnome-s+
 895 root      20   0 304056 2676  2308 S  0.3  0.1  0:00.39 VBoxSer+
 1353 shefali    20   0 834360 64244  43084 S  0.3  3.4  0:20.20 Xorg
 1968 shefali    20   0 827992 52884  39504 S  0.3  2.8  0:12.44 gnome-t+
 1 root       20   0 102260 11484  8288 S  0.0  0.6  0:03.89 systemd
 2 root       20   0      0      0      0 S  0.0  0.0  0:00.00 kthreadd
 3 root       0 -20      0      0      0 I  0.0  0.0  0:00.00 rcu_gp
 4 root       0 -20      0      0      0 I  0.0  0.0  0:00.00 rcu_par+
 5 root       20   0      0      0      0 I  0.0  0.0  0:00.40 kworker+
 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  0:00.10 ksoftir+
11 root      20   0      0      0      0 I  0.0  0.0  0:00.80 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 I  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

shefali@shefali-VirtualBox:~$ clear
shefali@shefali-VirtualBox:~$ 
```

21. WC: wc stands for **word count**. As the name implies, it is mainly used for counting purposes. It is used to find out the number of lines, word count, byte and characters count in the files specified in the file arguments. By default it displays four-columnar output. First column shows number of lines present in a file specified, second column shows number of words present in the file, third column shows number of characters present in file and fourth column itself is the file name which are given as argument.



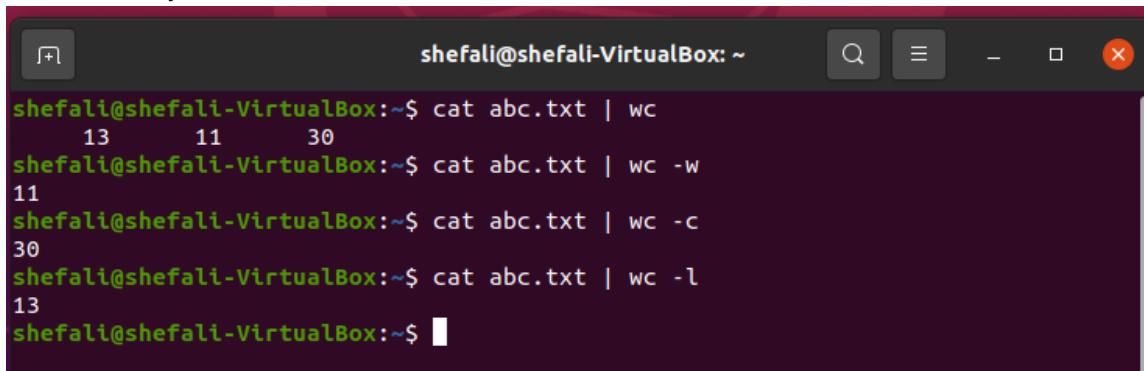
```
shefali@shefali-VirtualBox:~$ wc abc.txt
13 11 30 abc.txt
shefali@shefali-VirtualBox:~$ ./
.cache/    Documents/ .local/    new_dir/   .ssh/
.config/   Downloads/ .mozilla/ Pictures/  Templates/
Desktop/   .gnupg/   Music/     Public/   Videos/
shefali@shefali-VirtualBox:~$ ./new_dir | wc
bash: ./new_dir: Is a directory
      0      0      0
```

22. GREP: Suppose you want to search a particular piece of information, the postal code, from a text file. You may manually skim the content yourself to trace the information. A better option is to use the grep command. It will scan the document for the desired information and present the result in a format you want.



```
^C
shefali@shefali-VirtualBox:~$ grep "hello" *.txt
abc.txt:hello
shefali@shefali-VirtualBox:~$
```

23. PIPE: The Pipe is a command in Linux that lets you use two or more commands such that output of one command serves as input to the next. In short, the output of each process directly as input to the next one is like a pipeline. The symbol '|' denotes a pipe. Pipes help you mash-up two or more commands at the same time and run them consecutively.



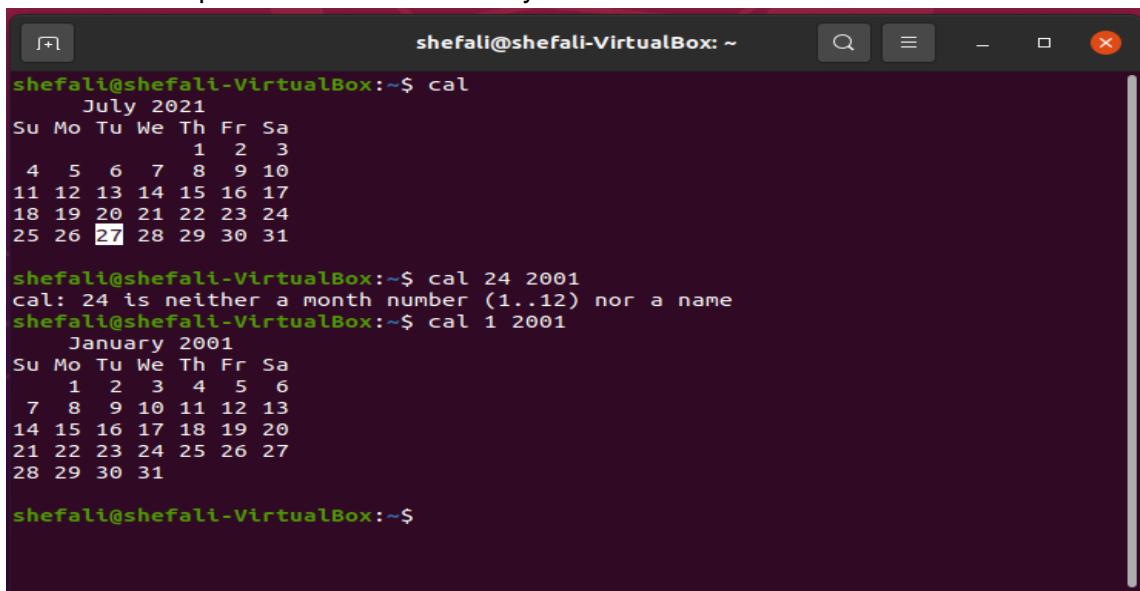
```
shefali@shefali-VirtualBox:~$ cat abc.txt | wc
      13      11      30
shefali@shefali-VirtualBox:~$ cat abc.txt | wc -w
      11
shefali@shefali-VirtualBox:~$ cat abc.txt | wc -c
      30
shefali@shefali-VirtualBox:~$ cat abc.txt | wc -l
      13
shefali@shefali-VirtualBox:~$
```

24. DF: The **df** command (short for disk free), is used to display information related to file systems about total space and available space.

```

shefali@shefali-VirtualBox:~$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
udev             911528       0   911528   0% /dev
tmpfs            188220     1324   186896   1% /run
/dev/sda5      10750520 7862176  2322536  78% /
tmpfs            941080       0   941080   0% /dev/shm
tmpfs              5120       4    5116   1% /run/lock
tmpfs            941080       0   941080   0% /sys/fs/cgroup
/dev/loop0        56832     56832       0 100% /snap/core18/2074
/dev/loop2        224256    224256       0 100% /snap/gnome-3-34-1804/66
/dev/loop1        56832     56832       0 100% /snap/core18/1988
/dev/loop3        224256    224256       0 100% /snap/gnome-3-34-1804/72
/dev/loop4        66432     66432       0 100% /snap/gtk-common-themes/1514
/dev/loop5        66688     66688       0 100% /snap/gtk-common-themes/1515
/dev/loop6        52352     52352       0 100% /snap/snap-store/518
/dev/loop7        52224     52224       0 100% /snap/snap-store/547
/dev/loop9        33152     33152       0 100% /snap/snapd/12398
/dev/loop8        31872     31872       0 100% /snap/snapd/11036
/dev/sda1         523248       4   523244   1% /boot/efi
tmpfs            188216      20   188196   1% /run/user/1000
/dev/sr0          59590     59590       0 100% /media/shefali/VBox_GAs_6.1.22
shefali@shefali-VirtualBox:~$
```

25. CAL: If a user wants a quick view of the calendar in the Linux terminal, cal is the command for you. By default, the cal command shows the current month's calendar as output. **cal** command is a calendar command in Linux which is used to see the calendar of a specific month or a whole year.



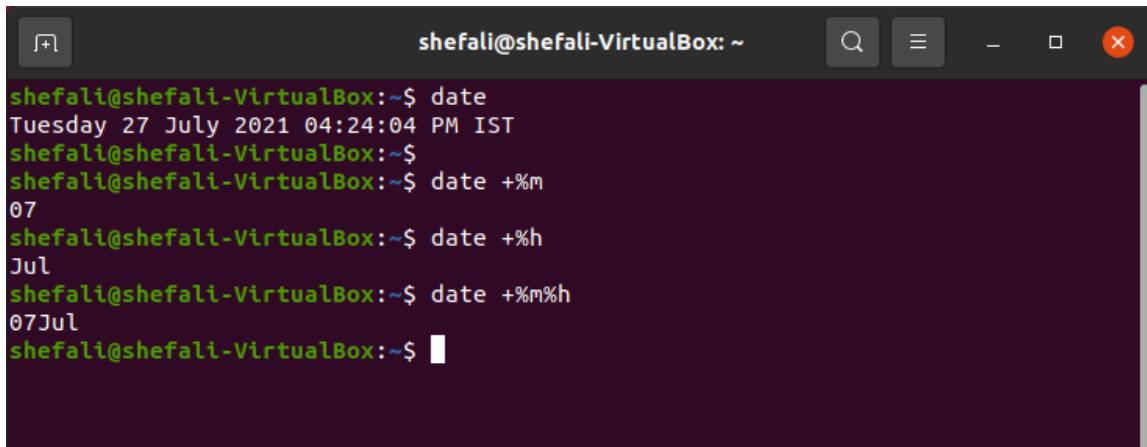
```

shefali@shefali-VirtualBox:~$ cal
July 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

shefali@shefali-VirtualBox:~$ cal 24 2001
cal: 24 is neither a month number (1..12) nor a name
shefali@shefali-VirtualBox:~$ cal 1 2001
January 2001
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

shefali@shefali-VirtualBox:~$
```

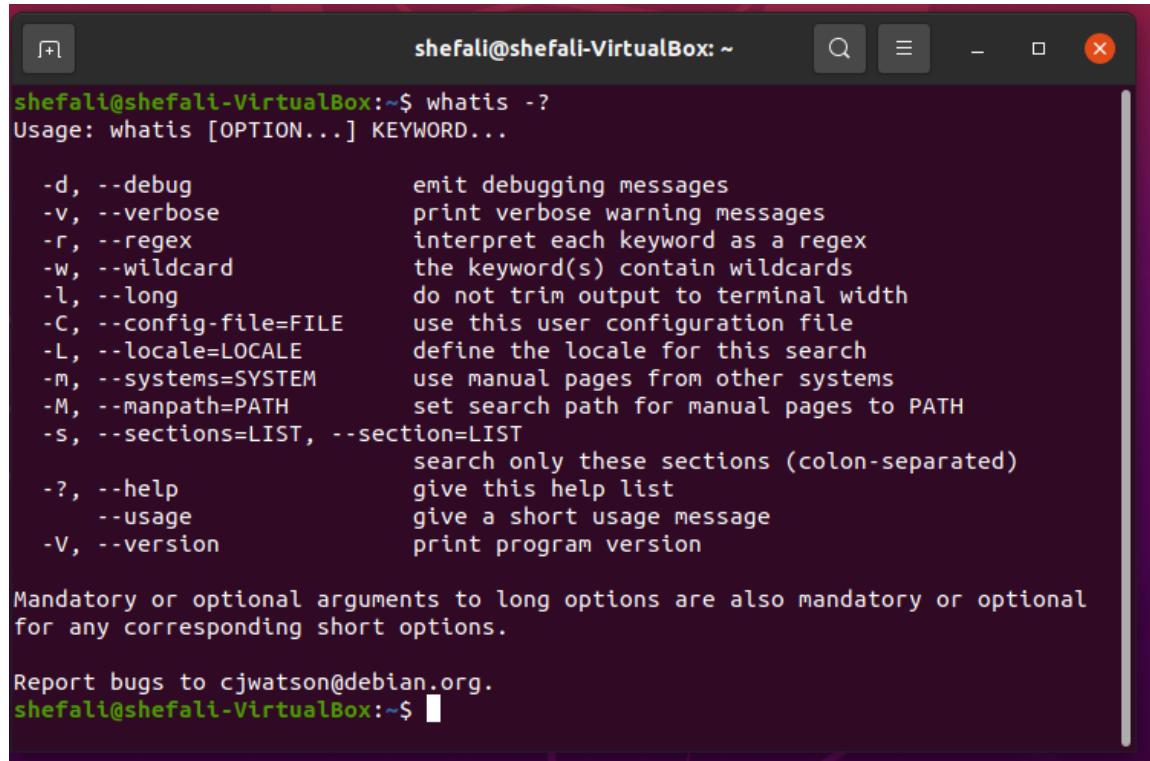
26. DATE: **date** command is used to display the system date and time. date command is also used to set the date and time of the system. By default the date command displays the date in the time zone on which the unix/linux operating system is configured. You must be the super-user (root) to change the date and time. **+%m** is used to display month (01...12). **+%h** is used to display abbreviated month name (eg. Jan).



A screenshot of a terminal window titled "shefali@shefali-VirtualBox: ~". The window shows the following sequence of commands and their outputs:

```
shefali@shefali-VirtualBox:~$ date
Tuesday 27 July 2021 04:24:04 PM IST
shefali@shefali-VirtualBox:~$ 
shefali@shefali-VirtualBox:~$ date +%m
07
shefali@shefali-VirtualBox:~$ date +%h
Jul
shefali@shefali-VirtualBox:~$ date +%m%h
07Jul
shefali@shefali-VirtualBox:~$
```

27. WHATIS: whatis command in Linux is **used to get a one-line manual page description**. In Linux, each manual page has some sort of description within it. So this command searches for the manual pages names and shows the manual page description of the specified filename or argument.



```
shefali@shefali-VirtualBox:~$ whatis -?
Usage: whatis [OPTION...] KEYWORD...

-d, --debug          emit debugging messages
-v, --verbose         print verbose warning messages
-r, --regex           interpret each keyword as a regex
-w, --wildcard        the keyword(s) contain wildcards
-l, --long            do not trim output to terminal width
-C, --config-file=FILE use this user configuration file
-L, --locale=LOCALE   define the locale for this search
-m, --systems=SYSTEM  use manual pages from other systems
-M, --manpath=PATH    set search path for manual pages to PATH
-s, --sections=LIST, --section=LIST
                     search only these sections (colon-separated)
-?, --help             give this help list
--usage               give a short usage message
-V, --version          print program version

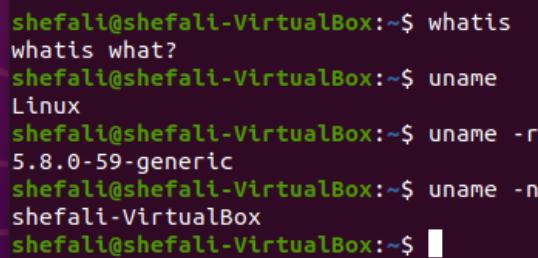
Mandatory or optional arguments to long options are also mandatory or optional
for any corresponding short options.

Report bugs to cjwtson@debian.org.
shefali@shefali-VirtualBox:~$
```

28. UNAME: The command ‘*uname*’ displays the information about the system.

-n option: It prints the hostname of the network node(current computer).

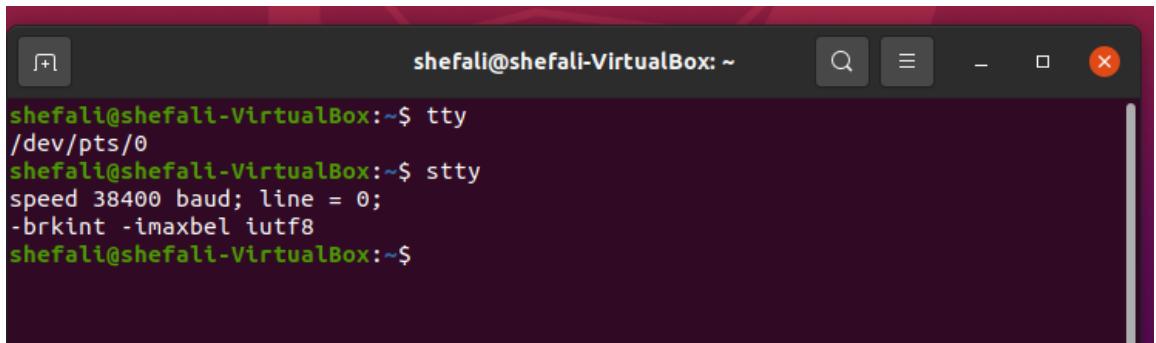
-r option: It prints the kernel release date.



```
shefali@shefali-VirtualBox:~$ whatis what?
whatis what?
shefali@shefali-VirtualBox:~$ uname
Linux
shefali@shefali-VirtualBox:~$ uname -r
5.8.0-59-generic
shefali@shefali-VirtualBox:~$ uname -n
shefali-VirtualBox
shefali@shefali-VirtualBox:~$
```

29. TTY: **tty** displays information related to the terminal. The **tty** command of terminal basically prints the file name of the terminal connected to standard input. **tty** is short for teletype, but popularly known as a terminal it allows you to interact with the system by passing on the data (you input) to the system, and displaying the output produced by the system.

30. STTY: **stty** command in Linux is used to change and print terminal line settings. Basically, this command shows or changes terminal characteristics.



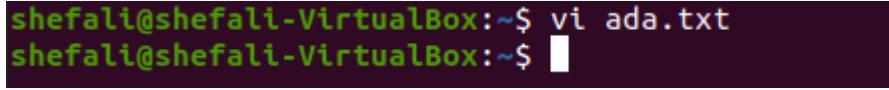
A screenshot of a terminal window titled "shefali@shefali-VirtualBox: ~". The window has a dark purple header bar with standard window controls (minimize, maximize, close) on the right. The main area of the terminal shows the following text:

```
shefali@shefali-VirtualBox:~$ tty  
/dev/pts/0  
shefali@shefali-VirtualBox:~$ stty  
speed 38400 baud; line = 0;  
-brkint -imaxbel iutf8  
shefali@shefali-VirtualBox:~$
```

EXPERIMENT 3

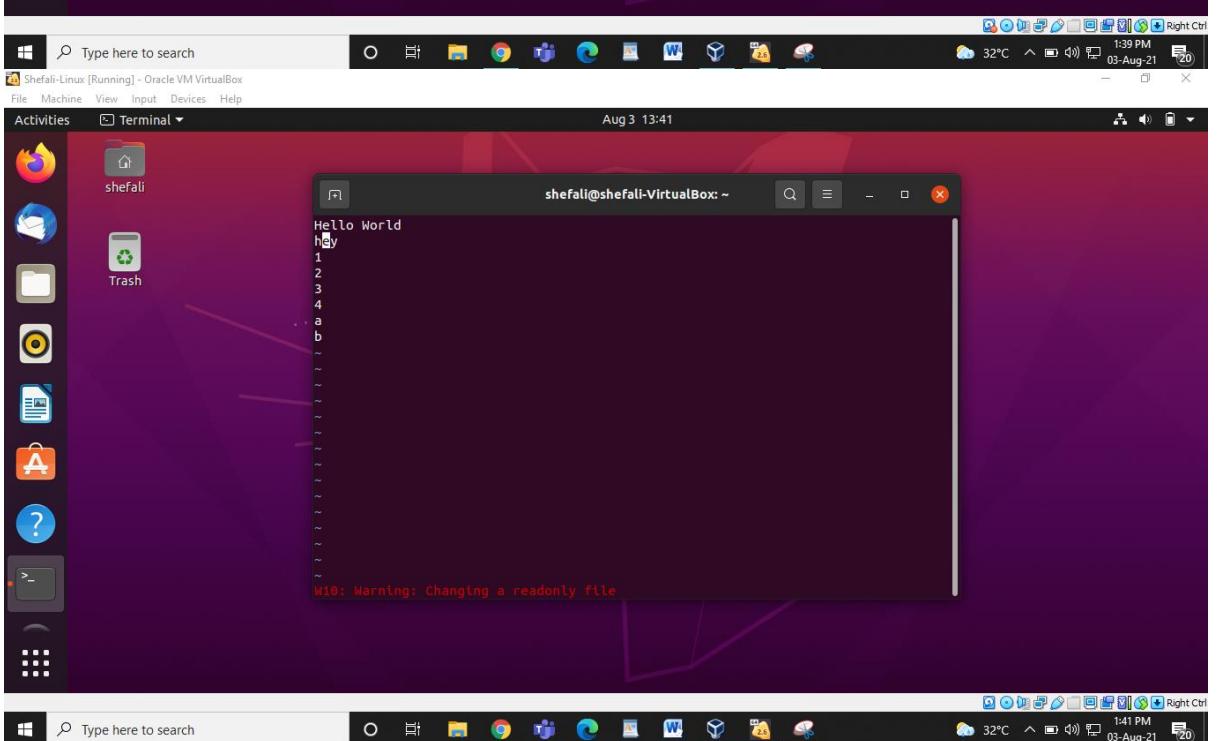
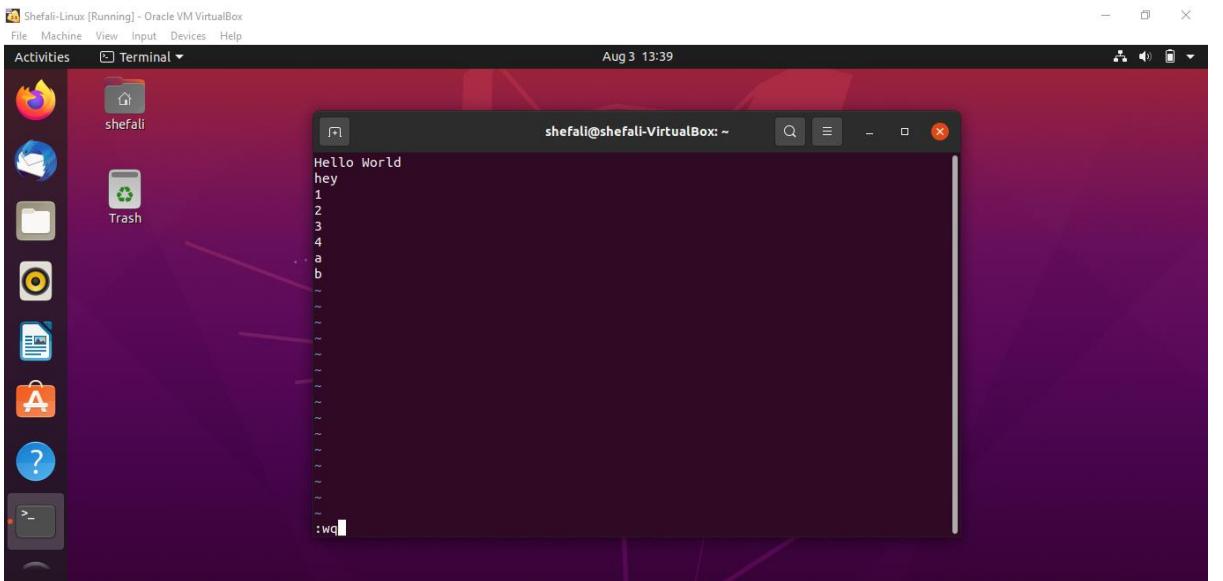
AIM: Study of VI Editor.

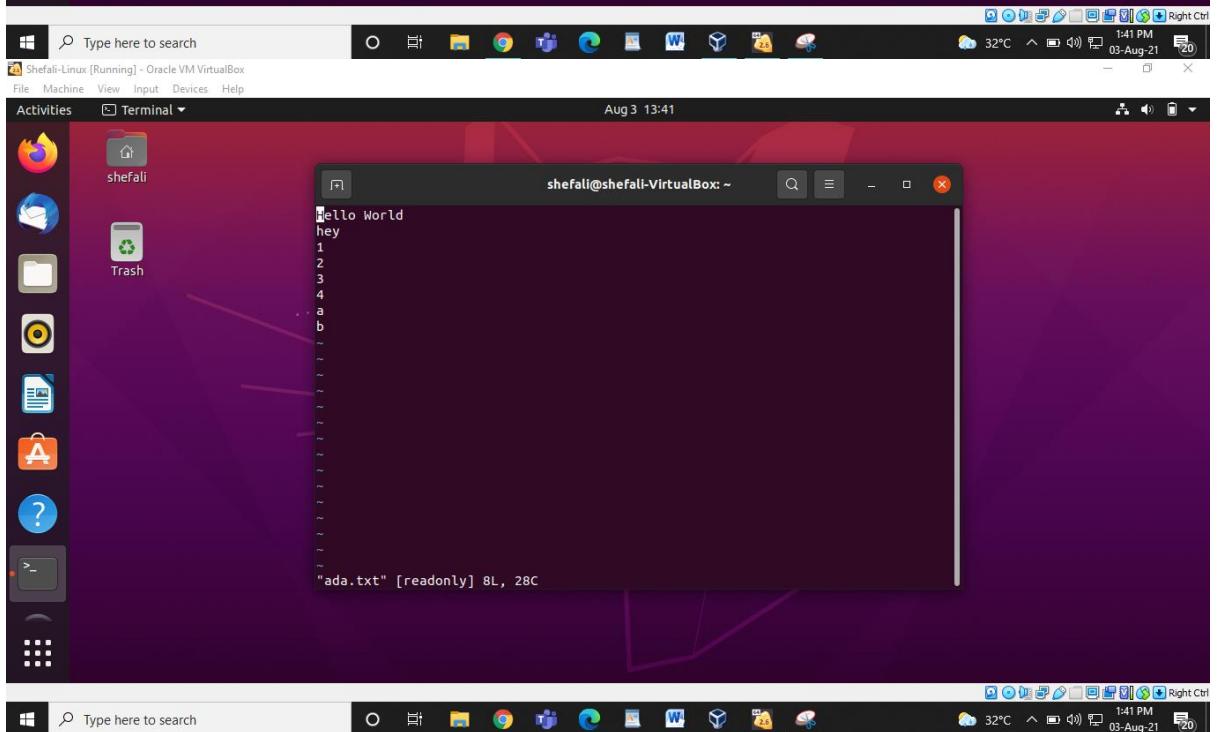
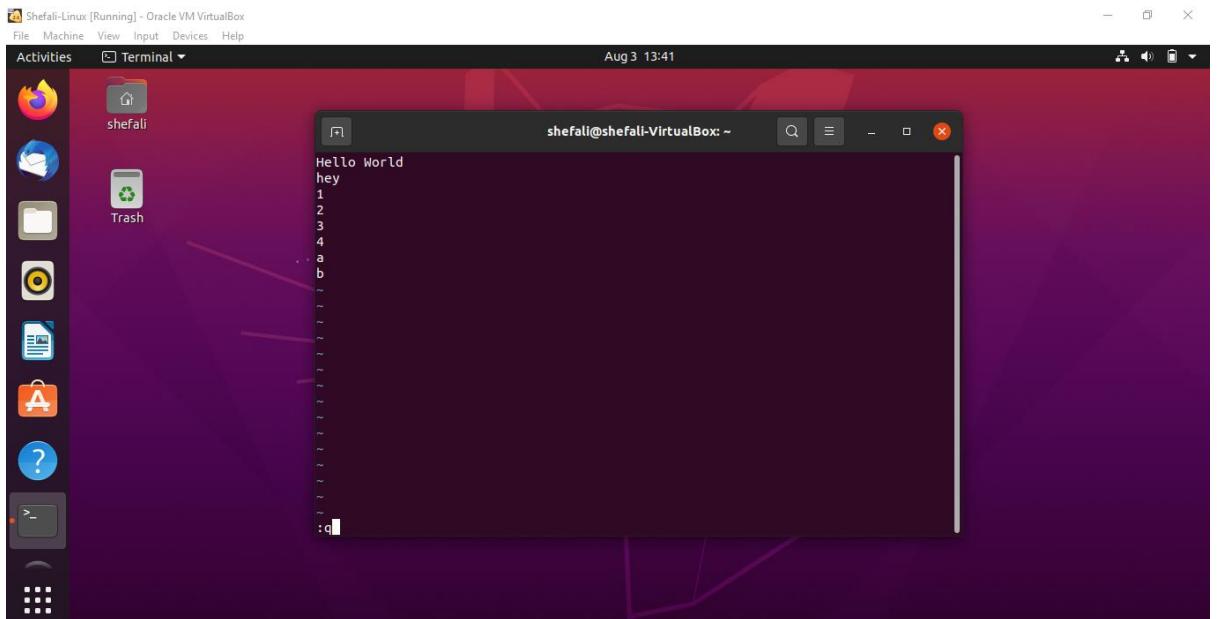
THEORY:

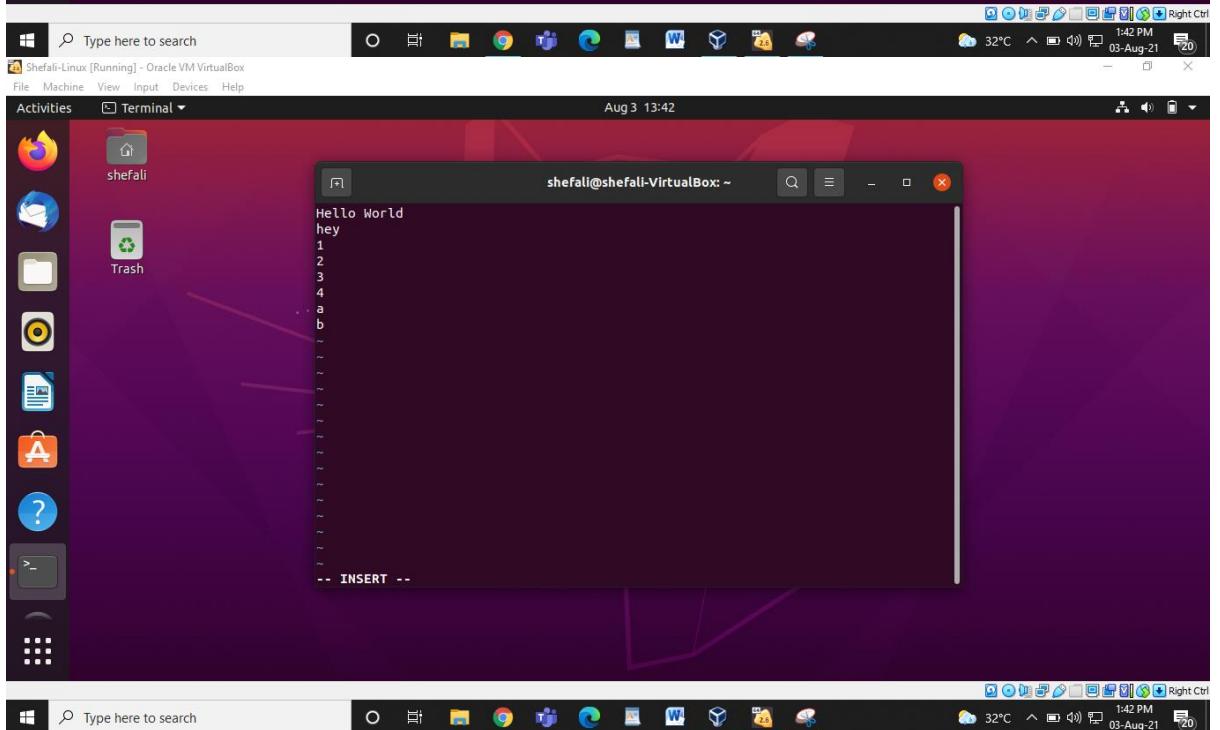
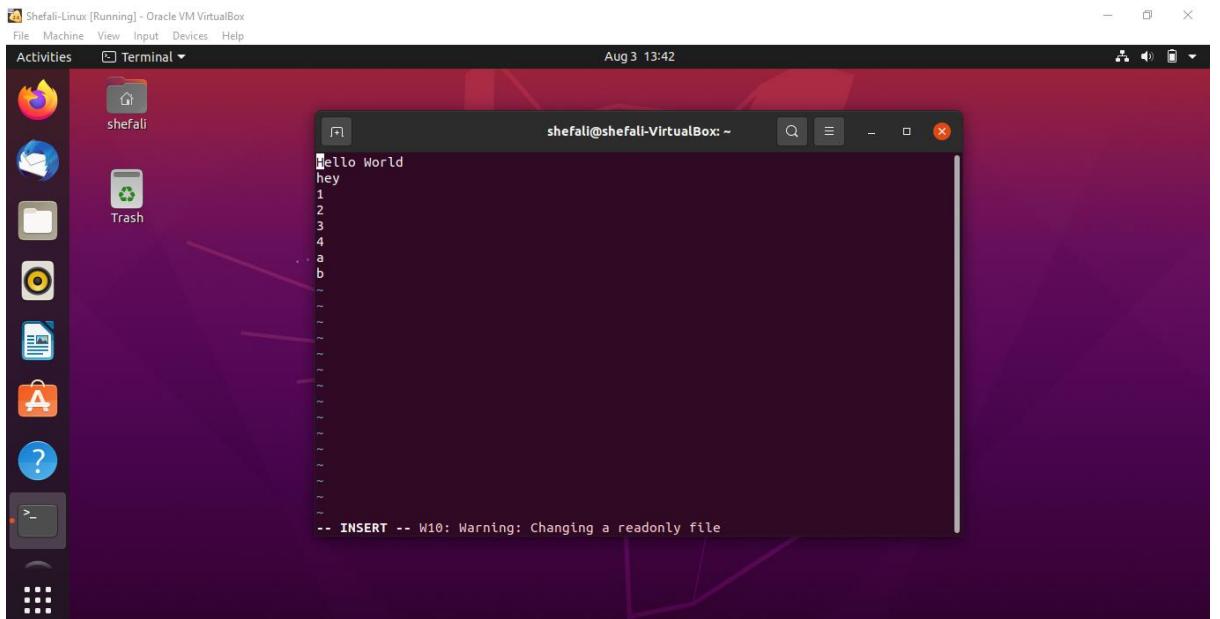
1. vi filename: Creates a new file if it already does not exist, otherwise opens an existing file.


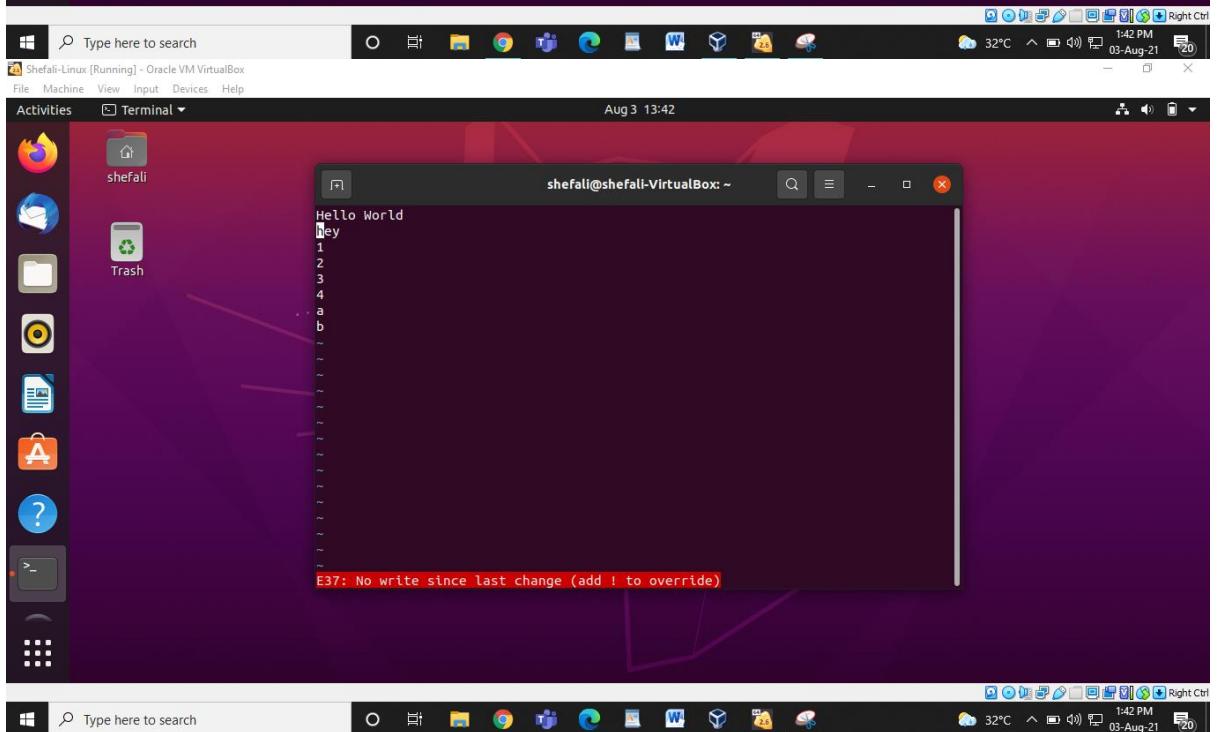
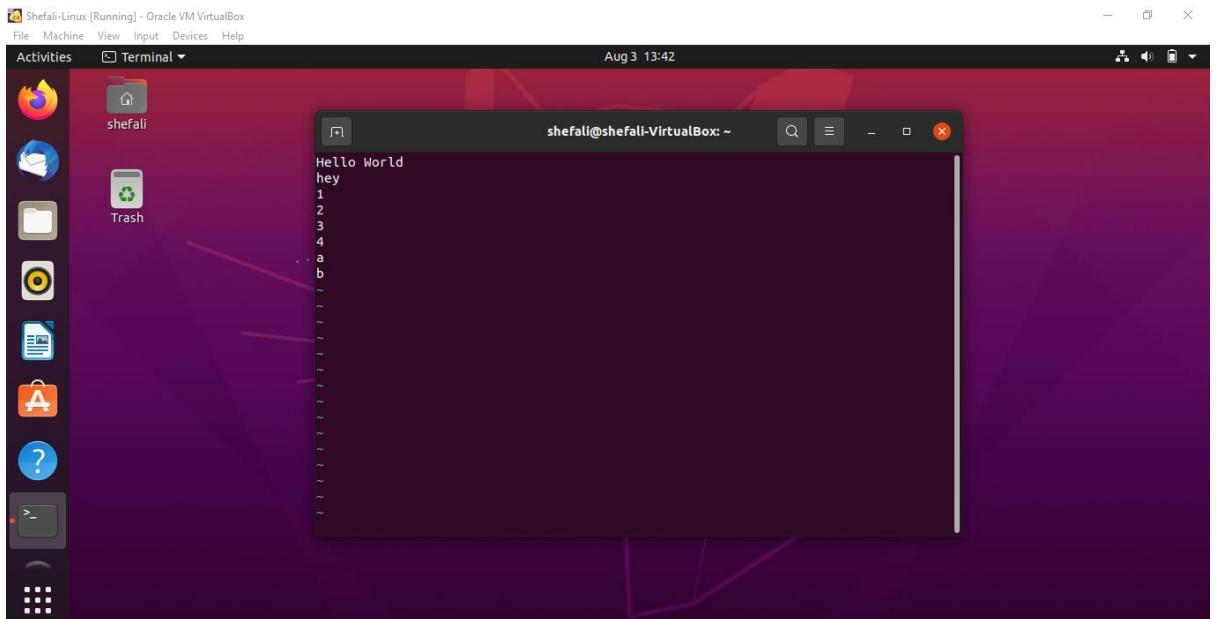
```
shefali@shefali-VirtualBox:~$ vi ada.txt
shefali@shefali-VirtualBox:~$ █
```
2. vi -R filename: Opens an existing file in the read-only mode.
3. view filename Opens an existing file in the read-only mode
4. Command mode: This mode enables you to perform administrative tasks such as saving the files, executing the commands, moving the cursor, cutting (yanking) and pasting the lines or words, as well as finding and replacing. In this mode, whatever you type is interpreted as a command.
5. Insert mode: This mode enables you to insert text into the file. Everything that's typed in this mode is interpreted as input and placed in the file.
To enter text, you must be in the insert mode for which simply type i. To come out of the insert mode, press the Esc key, which will take you back to the command mode
6. Exit from VI:
 - a. The command to quit out of vi is :q.
 - b. The command to quit out of vi without saving is :q!. This lets you exit vi without saving any of the changes.
 - c. The command to save the contents of the editor is :w. You can combine the above command with the quit command, or use :wq and return.
 - d. The easiest way to save your changes and exit vi is with the ZZ command.
 - e. If you want to specify/state any particular name for the file, you can do so by specifying it after the :w. For example, if you wanted to save the file you were working on as another filename called filename2, you would type :w filename2 and return.

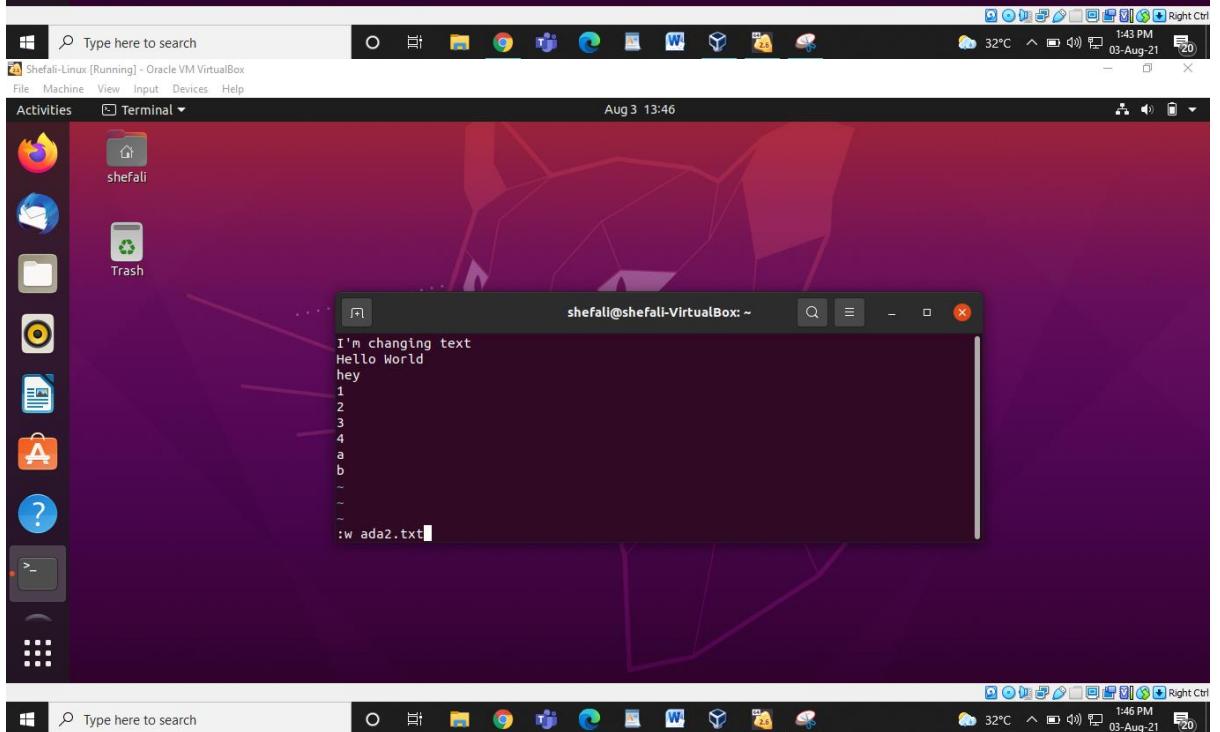
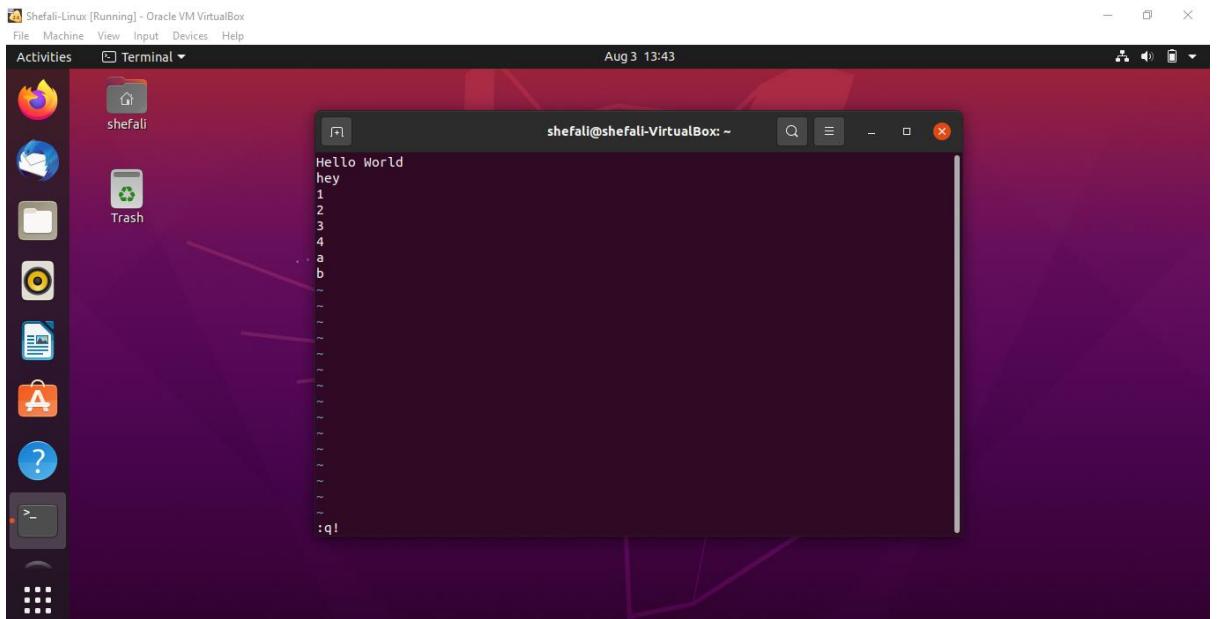
RESULTS:

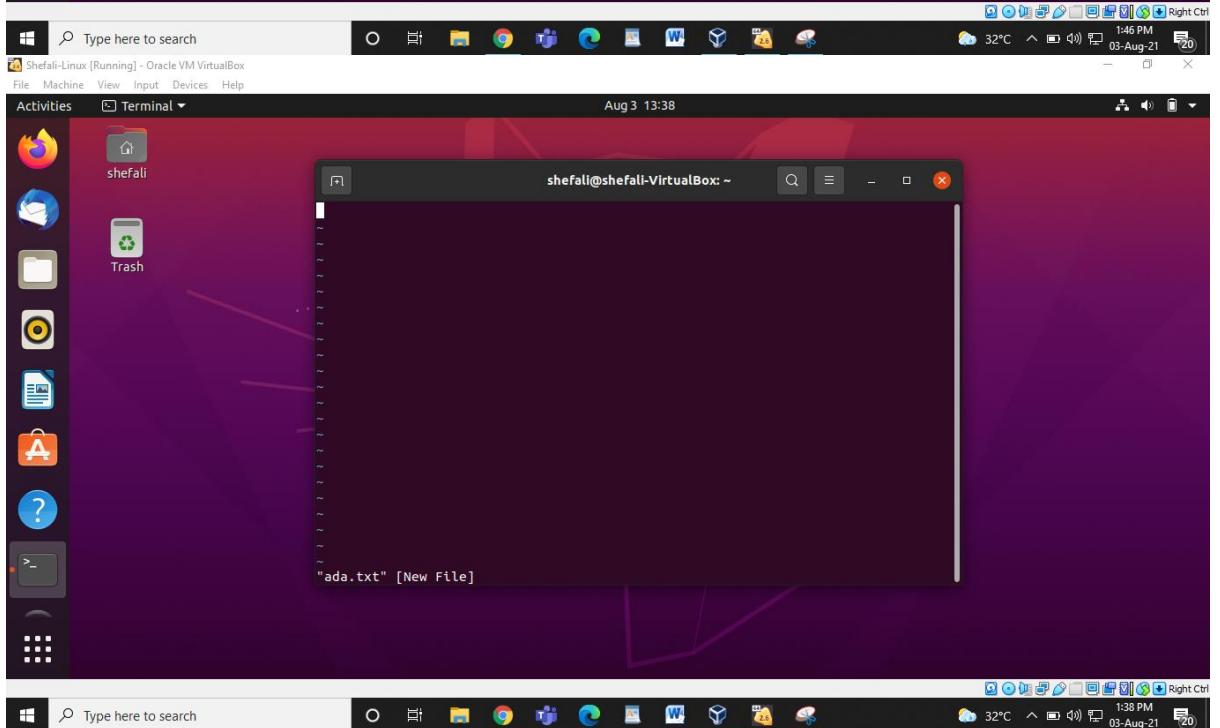
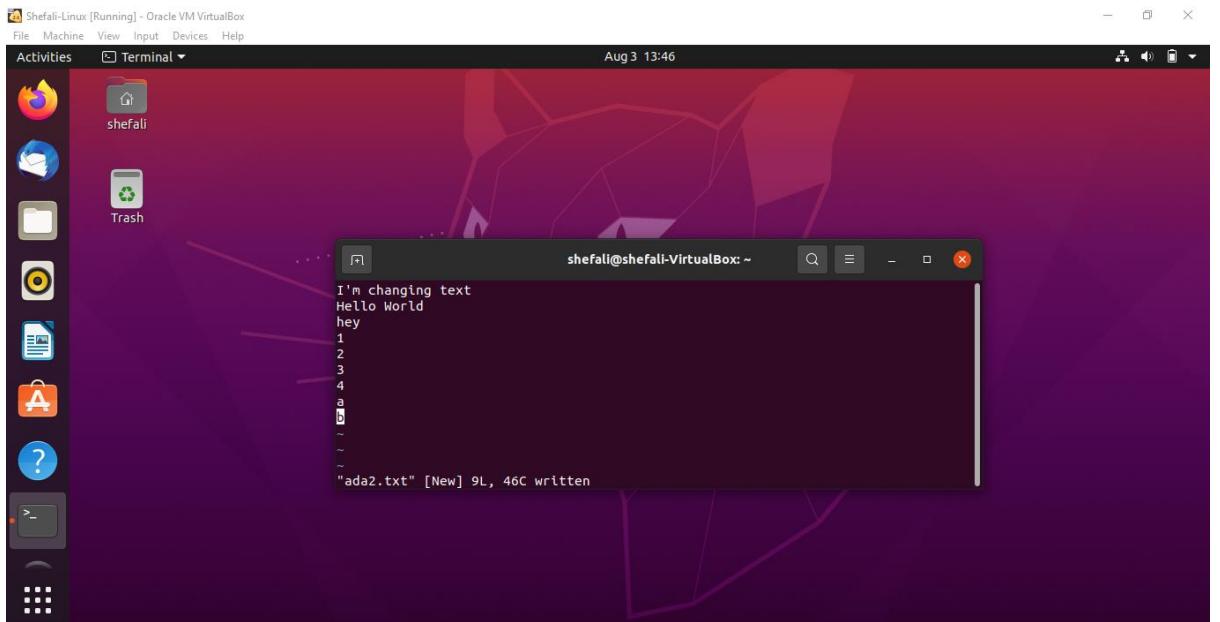












EXPERIMENT 4

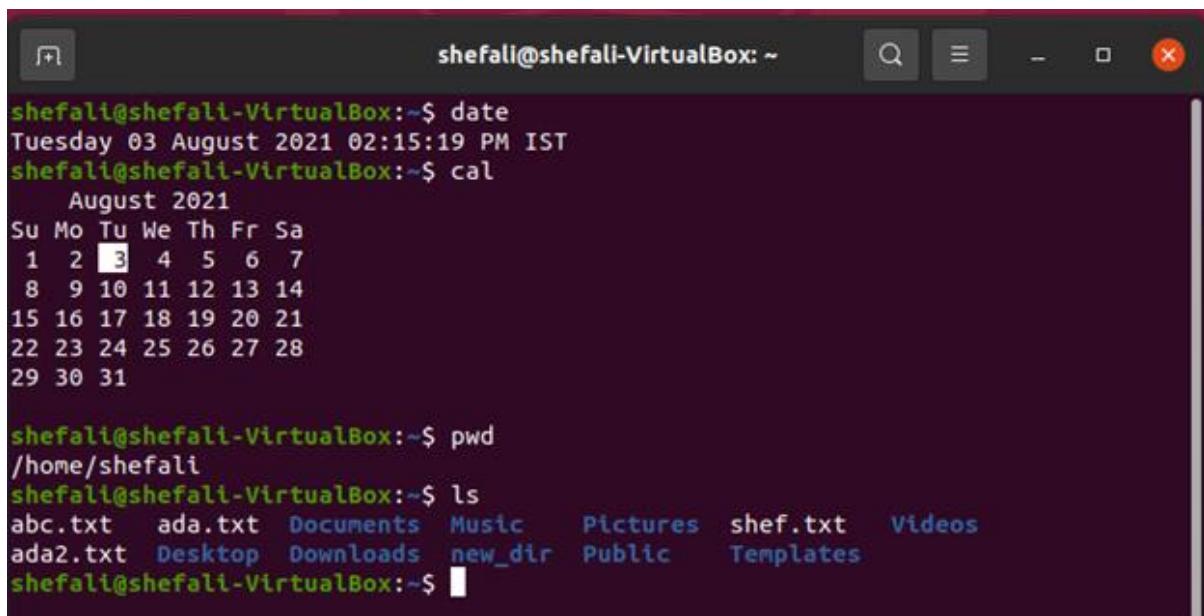
AIM: Study of Bash and Shell scripting in Linux (Ubuntu) OS.

THEORY:

Bash (bash) is one of many available (yet the most commonly used) Unix shells. Bash stands for "Bourne Again SHell", and is a replacement/improvement of the original Bourne shell (sh). Shell scripting is scripting in any shell, whereas Bash scripting is scripting specifically for Bash.

PROCEDURE:

SHELL:



The screenshot shows a terminal window titled 'shefali@shefali-VirtualBox: ~'. The window contains the following command-line session:

```
shefali@shefali-VirtualBox:~$ date
Tuesday 03 August 2021 02:15:19 PM IST
shefali@shefali-VirtualBox:~$ 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

shefali@shefali-VirtualBox:~$ pwd
/home/shefali
shefali@shefali-VirtualBox:~$ ls
abc.txt  ada.txt  Documents  Music    Pictures  shef.txt  Videos
ada2.txt  Desktop  Downloads  new_dir  Public   Templates
shefali@shefali-VirtualBox:~$
```

CREATING A SCRIPT:

```
shefali@shefali-VirtualBox:~$ vi task.sh
shefali@shefali-VirtualBox:~$ view task.sh
shefali@shefali-VirtualBox:~$ view task.sh
shefali@shefali-VirtualBox:~$ chmod +x task.sh
shefali@shefali-VirtualBox:~$ ./task.sh
Tuesday 03 August 2021 02:20:22 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/shefali
abc.txt  ada.txt  Documents  Music      Pictures  shef.txt  Templates
ada2.txt  Desktop   Downloads  new_dir  Public     task.sh  Videos
shefali@shefali-VirtualBox:~$
```

BASH:

View default operator:

```
shefali@shefali-VirtualBox:~$ echo $SHELL
/bin/bash
shefali@shefali-VirtualBox:~$ which bash
/usr/bin/bash
shefali@shefali-VirtualBox:~$ vi task.sh
shefali@shefali-VirtualBox:~$ view task.sh
```

```
#!/bin/bash
date
cal
pwd
ls
~
~
~
~
~
~
~
```

```
shefali@shefali-VirtualBox:~/
```

```
shefali@shefali-VirtualBox:~/
```

```
Tuesday 03 August 2021 02:27:51 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/shefali
```

```
abc.txt ada.txt Documents Music Pictures shef.txt Templates
```

```
ada2.txt Desktop Downloads new_dir Public task.sh Videos
```

```
shefali@shefali-VirtualBox:~$
```

```
shefali@shefali-VirtualBox:~$ file task.sh
```

```
task.sh: Bourne-Again shell script, ASCII text executable
```

```
shefali@shefali-VirtualBox:~$ view task.sh
```

```
shefali@shefali-VirtualBox:~$ file task.sh
```

```
task.sh: ASCII text
```

```
shefali@shefali-VirtualBox:~$
```

```
shefali@shefali-VirtualBox: ~
```

```
date
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
"date.sh" [readonly] 1L, 5C
```

```
shefali@shefali-VirtualBox:~$ echo date > date.sh
```

```
shefali@shefali-VirtualBox:~$ cat date.sh
```

```
date
```

```
shefali@shefali-VirtualBox:~$ ./date.sh
```

```
bash: ./date.sh: Permission denied
```

```
shefali@shefali-VirtualBox:~$ bash date.sh
```

```
Tuesday 03 August 2021 02:36:19 PM IST
```



```
shefali@shefali-VirtualBox:~/dev$ cd /dev/
shefali@shefali-VirtualBox:/dev$ cd ../home/shefali/
shefali@shefali-VirtualBox:~$ pwd
/home/shefali
shefali@shefali-VirtualBox:~$ cd -
/dev
shefali@shefali-VirtualBox:/dev$ pwd
/dev
shefali@shefali-VirtualBox:/dev$ cd -
/home/shefali
shefali@shefali-VirtualBox:~$ pwd
/home/shefali
shefali@shefali-VirtualBox:~$ cd ../../..
shefali@shefali-VirtualBox:$ pwd
/
shefali@shefali-VirtualBox:$ cd
shefali@shefali-VirtualBox:~$ pwd
/home/shefali
shefali@shefali-VirtualBox:~$
```

```
shefali@shefali-VirtualBox:~$ which bash > hello-world.sh
shefali@shefali-VirtualBox:~$ view hello-world.sh
shefali@shefali-VirtualBox:~$ chmod +x hello-world.sh
shefali@shefali-VirtualBox:~$ ./hello-world.sh
Hello World!!
shefali@shefali-VirtualBox:~$
```

```
shefali@shefali-VirtualBox:~$ ls
abc.txt  ada.txt  Desktop  Downloads  Music  Pictures  shef.txt  Templates
ada2.txt  date.sh  Documents  hello-world.sh  new_dir  Public  task.sh  Videos
shefali@shefali-VirtualBox:~$ ls -l
total 64
-rw-rw-r-- 1 shefali shefali 30 Jul 27 14:25 abc.txt
-rw-rw-r-- 1 shefali shefali 46 Aug 3 13:46 ada2.txt
-rw-rw-r-- 1 shefali shefali 28 Aug 3 13:40 ada.txt
-rwxrwxr-x 1 shefali shefali 17 Aug 3 14:39 date.sh
drwxr-xr-x 2 shefali shefali 4096 Jul 20 16:18 Desktop
drwxr-xr-x 2 shefali shefali 4096 Jul 20 16:18 Documents
drwxr-xr-x 2 shefali shefali 4096 Jul 20 16:18 Downloads
-rwxrwxr-x 1 shefali shefali 38 Aug 3 14:49 hello-world.sh
drwxr-xr-x 2 shefali shefali 4096 Jul 20 16:18 Music
drwxrwxr-x 2 shefali shefali 4096 Jul 27 14:32 new_dir
drwxr-xr-x 2 shefali shefali 4096 Jul 20 16:18 Pictures
drwxr-xr-x 2 shefali shefali 4096 Jul 20 16:18 Public
-rw-rw-r-- 1 shefali shefali 1 Jul 27 14:24 shef.txt
-rwxrwxr-x 1 shefali shefali 28 Aug 3 14:33 task.sh
drwxr-xr-x 2 shefali shefali 4096 Jul 20 16:18 Templates
drwxr-xr-x 2 shefali shefali 4096 Jul 20 16:18 Videos
shefali@shefali-VirtualBox:~$ ls -l hello-world.sh
-rwxrwxr-x 1 shefali shefali 38 Aug 3 14:49 hello-world.sh
shefali@shefali-VirtualBox:~$
```

EXPERIMENT 5

AIM:

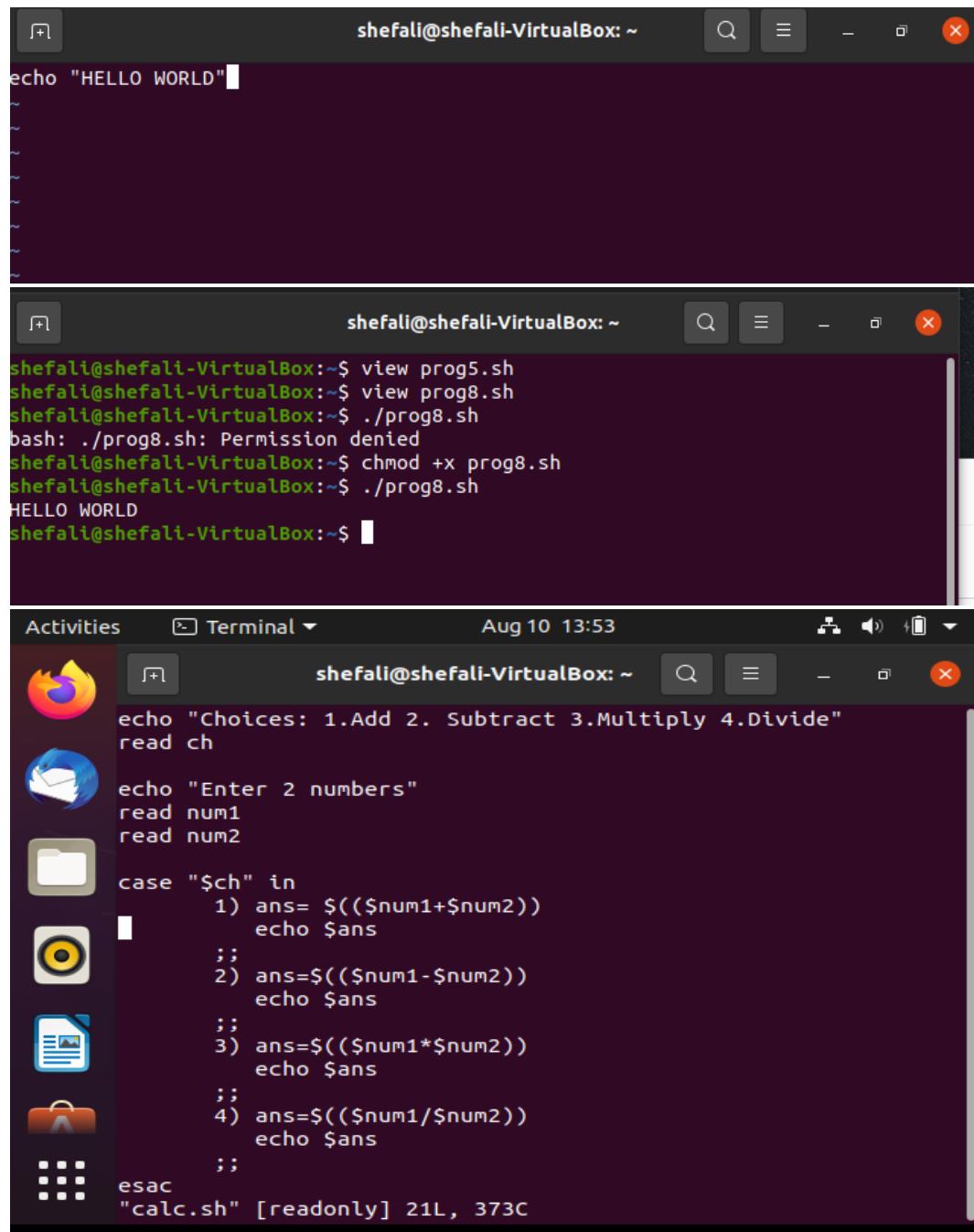
Write a shell script program to display HELLO WORLD.

Write a shell script program to create a simple calculator.

Write a shell script program to identify a given number as odd or even.

Write a shell script program to find a given number in the list.

PROCEDURE:



The image shows three separate terminal windows side-by-side, each with a dark purple background and white text. The top-left window contains a single command: "echo \"HELLO WORLD\"". The top-right window shows a sequence of commands to run a file named "prog8.sh": "view prog8.sh", "./prog8.sh", "chmod +x prog8.sh", and finally "./prog8.sh" which outputs "HELLO WORLD". The bottom window shows a more complex script named "calc.sh" that performs arithmetic operations based on user input. The script reads a choice from the user, calculates the result, and then prints it. It includes cases for addition, subtraction, multiplication, and division. The script ends with an "esac" statement and a note about its size ("21L, 373C").

```
echo "HELLO WORLD"  
~  
~  
~  
~  
~  
~  
~  
  
shefali@shefali-VirtualBox:~$ view prog8.sh  
shefali@shefali-VirtualBox:~$ view prog8.sh  
shefali@shefali-VirtualBox:~$ ./prog8.sh  
bash: ./prog8.sh: Permission denied  
shefali@shefali-VirtualBox:~$ chmod +x prog8.sh  
shefali@shefali-VirtualBox:~$ ./prog8.sh  
HELLO WORLD  
shefali@shefali-VirtualBox:~$  
  
Activities Terminal Aug 10 13:53  
shefali@shefali-VirtualBox:~$ echo "Choices: 1.Add 2. Subtract 3.Multiply 4.Divide"  
read ch  
shefali@shefali-VirtualBox:~$ echo "Enter 2 numbers"  
read num1  
read num2  
shefali@shefali-VirtualBox:~$ case "$ch" in  
    1) ans= $(($num1+$num2))  
       echo $ans  
    ;;  
    2) ans=$(($num1-$num2))  
       echo $ans  
    ;;  
    3) ans=$(($num1*$num2))  
       echo $ans  
    ;;  
    4) ans=$(($num1/$num2))  
       echo $ans  
    ;;  
esac  
"calc.sh" [readonly] 21L, 373C
```

```
shefali@shefali-VirtualBox:~$ ./calc.sh
Choices: 1.Add 2. Subtract 3.Multiply 4.Divide
2
Enter 2 numbers
10
5
5
shefali@shefali-VirtualBox:~$ ./calc.sh
Choices: 1.Add 2. Subtract 3.Multiply 4.Divide
3
Enter 2 numbers
10
5
50
shefali@shefali-VirtualBox:~$ ./calc.sh
Choices: 1.Add 2. Subtract 3.Multiply 4.Divide
4
Enter 2 numbers
10
5
2
shefali@shefali-VirtualBox:~$
```

```
shefali@shefali-VirtualBox:~$ ./prog2.sh
Enter num:
5
5 is odd
shefali@shefali-VirtualBox:~$ ./prog2.sh
Enter num:
2
2 is even
shefali@shefali-VirtualBox:~$
```

The image shows a Linux desktop environment with two terminal windows open. A context menu is displayed over the top terminal window, listing various desktop options.

Terminal Window 1 (Top):

```
echo "Enter num: "
read num

m=$((num%2))

if [ $m -eq 0 ]
then
    echo "$num is even"
else
    echo "$num is odd"
fi
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"prog2.sh" [readonly] 11L, 117C
```

Terminal Window 2 (Bottom):

```
arr=(5 10 0 1 3)
echo "Enter num: "
read num

a=0
length=${#arr[*]}
flag=0
echo "List:"
echo ${arr[*]}
while [ $a -lt $length ]
do
    x=${arr[$a]}
    if [ $x -eq $num ]
    then
        echo "Found at index $a"
        flag=`expr $flag + 1`
        break
    fi
    a=`expr $a + 1`
done

if [ $flag -eq 0 ]
then
    echo "$num not found in array"
fi
~
```

Context Menu (Visible over Terminal 1):

- Toolbars
- Search
- News and interests
- ✓ Show Cortana button
- ✓ Show Task View button
- Show People on the taskbar
- Show Windows Ink Worksp...
- Show touch keyboard button
- Cascade windows
- Show windows stacked
- Show windows side by side
- Show the desktop

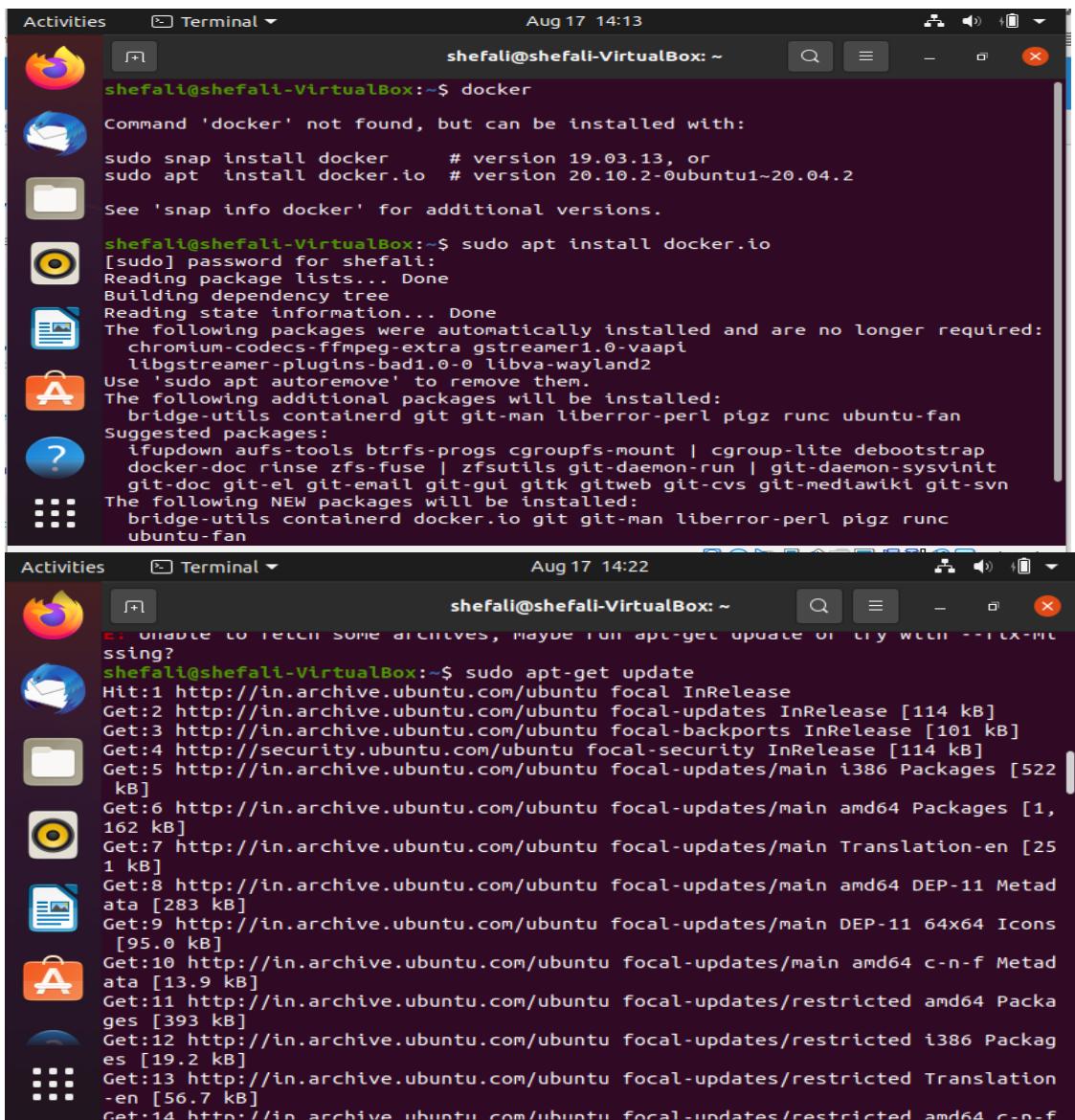
```
+| shefali@shefali-VirtualBox:~$ view prog3.sh
shefali@shefali-VirtualBox:~$ ./prog3.sh
Enter num:
6
List:
5 10 0 1 3
6 not found in array
shefali@shefali-VirtualBox:~$ ./prog3.sh
Enter num:
0
List:
5 10 0 1 3
Found at index 2
shefali@shefali-VirtualBox:~$
```

EXPERIMENT 6

AIM: Implementation of Docker on Linux OS (Ubuntu).

THEORY: Docker is a set of platforms as a service product that use OS-level virtualization to deliver software in packages called containers. Containers are isolated from one another and bundle their own software, libraries and configuration files and they can communicate with each other through well-defined channels.

PROCEDURE:



The image shows two terminal windows side-by-side. Both windows have a dark theme and are titled "Terminal". The top window shows the command "shefali@shefali-VirtualBox:~\$ docker" followed by an error message: "Command 'docker' not found, but can be installed with: sudo snap install docker # version 19.03.13, or sudo apt install docker.io # version 20.10.2-0ubuntu1~20.04.2". It also includes instructions to see "snap info docker" for additional versions. The bottom window shows the command "shefali@shefali-VirtualBox:~\$ sudo apt install docker.io" followed by a long list of package dependencies and suggested packages. The second terminal window then shows the command "shefali@shefali-VirtualBox:~\$ sudo apt-get update" followed by a list of package downloads from the Ubuntu repositories.

```
shefali@shefali-VirtualBox:~$ docker
Command 'docker' not found, but can be installed with:
sudo snap install docker      # version 19.03.13, or
sudo apt install docker.io    # version 20.10.2-0ubuntu1~20.04.2
See 'snap info docker' for additional versions.

shefali@shefali-VirtualBox:~$ sudo apt install docker.io
[sudo] password for shefali:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
  libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  bridge-utils containerd git git-man liberror-perl pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools btrfs-progs cgroupfs-mount | cgroup-lite debootstrap
  docker-doc rinse zfs-fuse | zfsutils git-daemon-run | git-daemon-sysvinit
  git-doc git-el git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  bridge-utils containerd docker.io git git-man liberror-perl pigz runc
  ubuntu-fan

shefali@shefali-VirtualBox:~$ sudo apt-get update
E: Unable to fetch some archives, maybe run apt-get update or try with --fix-missing?
shefali@shefali-VirtualBox:~$ sudo apt-get update
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://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu focal-updates/main i386 Packages [522 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1,162 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [251 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metadata [283 kB]
Get:9 http://in.archive.ubuntu.com/ubuntu focal-updates/main DEP-11 64x64 Icons [95.0 kB]
Get:10 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metadata [13.9 kB]
Get:11 http://in.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [393 kB]
Get:12 http://in.archive.ubuntu.com/ubuntu focal-updates/restricted i386 Packages [19.2 kB]
Get:13 http://in.archive.ubuntu.com/ubuntu focal-updates/restricted Translation-en [56.7 kB]
Get:14 http://in.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 c-n-f
```

Activities Terminal Aug 17 14:23

```
shefali@shefali-VirtualBox:~$ sudo apt-get install \
> apt-transport-https \
> ca-certificates \
> curl \
> gnupg \
> lsb-release
Reading package lists... Done
shefali@shefali-VirtualBox:~$ Building dependency tree
Reading state information... Done
lsb-release is already the newest version (11.1.0ubuntu2).
lsb-release set to manually installed.
ca-certificates is already the newest version (20210119-20.04.1).
ca-certificates set to manually installed.
The following packages were automatically installed and are no longer required:
chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
dirmngr gnupg-l10n gnupg-utils gpg gpg-agent gpg-wks-client gpg-wks-server
gpgconf gpgsm gpgv libcurl4
Suggested packages:
tor parcmimonie xloadimage scdaemon
The following NEW packages will be installed:
```

Activities Terminal Aug 17 14:23

```
shefali@shefali-VirtualBox:~$ After this operation, 397 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 runc amd64 1
.0.0~rc95-0ubuntu1~20.04.2 [4,087 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 containerd a
md64 1.5.2-0ubuntu1~20.04.2 [32.9 MB]
Get:3 http://in.archive.ubuntu.com/ubuntu focal-updates/universe amd64 docker.i
o amd64 20.10.7-0ubuntu1~20.04.1 [36.9 MB]
Fetched 73.9 MB in 11s (6,570 kB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 188838 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.4-1_amd64.deb ...
Unpacking pigz (2.4-1) ...
Selecting previously unselected package bridge-utils.
Preparing to unpack .../1-bridge-utils_1.6-2ubuntu1_amd64.deb ...
Unpacking bridge-utils (1.6-2ubuntu1) ...
Selecting previously unselected package runc.
Preparing to unpack .../2-runc_1.0.0~rc95-0ubuntu1~20.04.2_amd64.deb ...
Unpacking runc (1.0.0~rc95-0ubuntu1~20.04.2) ...
Selecting previously unselected package containerd.
Preparing to unpack .../3-containerd_1.5.2-0ubuntu1~20.04.2_amd64.deb ...
Unpacking containerd (1.5.2-0ubuntu1~20.04.2) ...

Progress: [ 19%] [#####.....]
```

Activities Terminal Aug 17 14:27

```
shefali@shefali-VirtualBox:~$ docker --version
Docker version 20.10.7, build 20.10.7-0ubuntu1~20.04.1
shefali@shefali-VirtualBox:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
b8dfde127a29: Pull complete
Digest: sha256:0fe98d7debd9049c50b597ef1f85b7c1e8cc81f59c8d623fc2250e8bec85b38
Status: Downloaded newer image for hello-world:latest

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 it
    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:
```

Activities Terminal Aug 17 14:30

```
shefali@shefali-VirtualBox:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2021-08-17 14:25:03 IST; 5min ago
     TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
      Main PID: 5360 (dockerd)
        Tasks: 11
       Memory: 58.0M
      CGroup: /system.slice/docker.service
              └─5360 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Aug 17 14:24:58 shefali-VirtualBox dockerd[5360]: time="2021-08-17T14:24:58.612+05:30" level=info msg="Starting Docker Application Container Engine"
Aug 17 14:24:58 shefali-VirtualBox dockerd[5360]: time="2021-08-17T14:24:58.612+05:30" level=info msg="Starting Docker Application Container Engine"
Aug 17 14:24:58 shefali-VirtualBox dockerd[5360]: time="2021-08-17T14:24:58.620+05:30" level=info msg="Starting Docker Application Container Engine"
Aug 17 14:24:59 shefali-VirtualBox dockerd[5360]: time="2021-08-17T14:24:59.483+05:30" level=info msg="Starting Docker Application Container Engine"
Aug 17 14:25:00 shefali-VirtualBox dockerd[5360]: time="2021-08-17T14:25:00.267+05:30" level=info msg="Starting Docker Application Container Engine"
Aug 17 14:25:03 shefali-VirtualBox dockerd[5360]: time="2021-08-17T14:25:03.813+05:30" level=info msg="Starting Docker Application Container Engine"
Aug 17 14:25:03 shefali-VirtualBox dockerd[5360]: time="2021-08-17T14:25:03.813+05:30" level=info msg="Starting Docker Application Container Engine"
Aug 17 14:25:03 shefali-VirtualBox systemd[1]: Started Docker Application Container Engine.
Aug 17 14:25:03 shefali-VirtualBox dockerd[5360]: time="2021-08-17T14:25:03.953+05:30" level=info msg="Starting Docker Application Container Engine.
A Show Applications shefali-VirtualBox dockerd[5360]: time="2021-08-17T14:27:26.262+05:30" level=info msg="Starting Docker Application Container Engine.

Lines 1-21/21 (END)
```

```
shefali@shefali-VirtualBox:~$ sudo docker images
[sudo] password for shefali:
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
hello-world      latest   d1165f221234  5 months ago   13.3kB
shefali@shefali-VirtualBox:~$ 
shefali@shefali-VirtualBox:~$ 
shefali@shefali-VirtualBox:~$ sudo docker search ubuntu
NAME          DESCRIPTION
ubuntu        Ubuntu is a Debian-based Linux operating sys...
ubuntu_12646   [OK]
dorowu/ubuntu-desktop-lxde-vnc  Docker image to provide HTML5 VNC interface ...
560          [OK]
websphere-liberty  WebSphere Liberty multi-architecture images ...
277          [OK]
rastasheep/ubuntu-sshd  Dockerized SSH service, built on top of offic...
255          [OK]
consol/ubuntu-xfce-vnc  Ubuntu container with "headless" VNC session...
241          [OK]
ubuntu-upstart  Upstart is an event-based replacement for th...
113          [OK]
ansible/ubuntu14.04-ansible  Ubuntu 14.04 LTS with ansible
98           [OK]
1and1internet/ubuntu-16-nginx-php-phpmyadmin-mysql-5  ubuntu-16-nginx-php-phpmyadmin-mysql-5
50           [OK]
ubuntu-debootstrap  debootstrap --variant=minbase --components=...
44           [OK]
i386/ubuntu    Ubuntu is a Debian-based Linux operating sys...
```

PULL:

```
shefali@shefali-VirtualBox:~$ sudo docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
16ec32c2132b: Pull complete
Digest: sha256:82becede498899ec668628e7cb0ad87b6e1c371cb8a1e597d83a47fac21d6af3
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
shefali@shefali-VirtualBox:~$
```

PS and PS -A:

```
shefali@shefali-VirtualBox:~$ sudo docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
ubuntu        latest   1318b700e415  3 weeks ago   72.8MB
hello-world      latest   d1165f221234  5 months ago   13.3kB
shefali@shefali-VirtualBox:~$ 
shefali@shefali-VirtualBox:~$ 
shefali@shefali-VirtualBox:~$ sudo docker run -it -d ubuntu
3723fbff56e86ef2f7588af9318866010c35a4a6c791799a163ed0f0447e2764
shefali@shefali-VirtualBox:~$ 
shefali@shefali-VirtualBox:~$ sudo docker ps
CONTAINER ID      IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES
3723fbff56e8      ubuntu      "bash"      17 seconds ago   Up 5 seconds   funny_gates
shefali@shefali-VirtualBox:~$ 
shefali@shefali-VirtualBox:~$ sudo docker ps -a
CONTAINER ID      IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES
3723fbff56e8      ubuntu      "bash"      21 seconds ago   Up 10 seconds   funny_gates
f5089434086f      hello-world  "/hello"    29 hours ago   Exited (0) 29 hours ago
n
shefali@shefali-VirtualBox:~$
```

LOGIN (DockerHub):

```
shefali@shefali-VirtualBox:~$ 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: shefalibansal
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
shefali@shefali-VirtualBox:~$
```

WHALESAY:

```
shefali@shefali-VirtualBox: ~
```

/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: sha256:178598e51a26abbc958b8a2e48825c90bc22e641de3d31e18aaaf55f3258ba93b
Status: Downloaded newer image for docker/whalesay:latest

```
< boo >
-----
\ \
  \
    ##      .
    ## ## ##    ==
    ## ## ## ##   ===
 /"*****"__/_/ ===
~~~ {~~ ~~~~ ~~~ ~~~~ ~~ ~ / ===- ~~~
     \____ o
       \_ \
         \_ \_ \_ /
```

```
shefali@shefali-VirtualBox:~$
```

RUN:

```

shefali@shefali-VirtualBox:~$ sudo docker run nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
e1acddbe380c: Pull complete
e21006f71c6f: Pull complete
f3341cc17e58: Pull complete
2a53fa598ee2: Pull complete
12455f71a9b5: Pull complete
b86f2ba62d17: Pull complete
Digest: sha256:4d4d96ac750af48c6a551d757c1cbfc071692309b491b70b2b8976e102dd3fef
Status: Downloaded newer image for nginx:latest
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2021/08/18 14:01:06 [notice] 1#1: using the "epoll" event method
2021/08/18 14:01:06 [notice] 1#1: nginx/1.21.1
2021/08/18 14:01:06 [notice] 1#1: built by gcc 8.3.0 (Debian 8.3.0-6)

```

STOP:

```

shefali@shefali-VirtualBox:~$ sudo docker ps -a
CONTAINER ID   IMAGE          COMMAND       CREATED      STATUS
              PORTS NAMES
11cde2510ae1   nginx         "/docker-entrypoint..."   About a minute ago   Exited (0) 25 seconds ago
                unruffled_hamilton
398d59c3cae   docker/whalesay  "cowsay boo"    3 minutes ago     Exited (0) 3 minutes ago
                peaceful_chatelet
3723fbff56e8   ubuntu        "bash"        21 minutes ago    Up 21 minutes
                funny_gates
f5089434086f   hello-world   "/hello"       29 hours ago     Exited (0) 29 hours ago
                sweet_bouman
shefali@shefali-VirtualBox:~$ sudo docker stop unruffled_hamilton
unruffled_hamilton
shefali@shefali-VirtualBox:~$ sudo docker stop unruffled_hamilton
unruffled_hamilton

```

RM -

```

shefali@shefali-VirtualBox:~$ sudo docker rm unruffled_hamilton
unruffled_hamilton
shefali@shefali-VirtualBox:~$ sudo docker ps -a
CONTAINER ID   IMAGE          COMMAND       CREATED      STATUS      PORTS     NAMES
398d59c3cae   docker/whalesay  "cowsay boo"    5 minutes ago   Exited (0) 5 minutes ago
                peaceful_chatelet
3723fbff56e8   ubuntu        "bash"        23 minutes ago   Up 23 minutes
                funny_gates
f5089434086f   hello-world   "/hello"       29 hours ago     Exited (0) 29 hours ago
                sweet_bouman

```

RMI -

```
shefali@shefali-VirtualBox:~$ sudo docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
nginx          latest   dd34e67e3371  26 hours ago  133MB
ubuntu          latest   1318b700e415  3 weeks ago  72.8MB
hello-world     latest   d1165f221234  5 months ago  13.3kB
docker/wholesay latest   6b362a9f73eb  6 years ago  247MB
shefali@shefali-VirtualBox:~$ sudo docker rm nginx
Error: No such container: nginx
shefali@shefali-VirtualBox:~$ sudo docker rmi nginx
Untagged: nginx:latest
Untagged: nginx@sha256:4d4d96ac750af48c6a551d757c1cbfc071692309b491b70b2b8976e102dd3fef
Deleted: sha256:dd34e67e3371dc2d1328790c3157ee42dfcae74afffd86b297459ed87a98c0fb
Deleted: sha256:ec6149850eea7af0bfa5f4aa0130d2c3cbae06e4b5da8c748d8b6b1b0cb81d07
Deleted: sha256:2a3d94c7adfe6e94ef038a9b3ea3631168e979f8ddb49a38b203e364627af2d9
Deleted: sha256:2bbff8011bb867605e83fdb8095f94a347307726b8cce81d752886a8af974aea
Deleted: sha256:f151353bef203bd70680578f33abd9667b65434ffadf547f900dc09927cc435
Deleted: sha256:47c01ba78b6d0bdef530c46858d4c83b87452d42dc9faa54b02b3e026107ff27
Deleted: sha256:f68ef921efae588b3dd5cc466a1ca9c94c24785f1fa9420bea15ecc2dedbe781
shefali@shefali-VirtualBox:~$
```

EXPERIMENT 7-A

AIM: Implementation of shell script programs.

1. Check whether the given file is a directory or not.
2. Count the number of files in the directory.
3. Copy contents of one file to another.

PROCEDURE:

```
shefali@shefali-VirtualBox: ~
#!/bin/bash

# if passed argument is directory or not

if [ -d "$1" ];
then
  echo " $1 is a directory"
elif [ -f "$1" ];
then
  echo "$1 is a file"
else
  echo " $1 is not a file or a directory"
fi

~
~
~
~
~
~
~
~

"prog4.sh" [readonly] 14L, 201C
```

```
shefali@shefali-VirtualBox:~$ view prog4.sh
shefali@shefali-VirtualBox:~$ ./prog4.sh ada.txt
ada.txt is a file
shefali@shefali-VirtualBox:~$ ./prog4.sh Downloads
Downloads is a directory
shefali@shefali-VirtualBox:~$ ./prog4.sh random
random is not a file or a directory
shefali@shefali-VirtualBox:~$
```

```
Shefali-Linux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Sep 7 13:46
shefali@shefali-VirtualBox: ~
#!/bin/bash

# count no. of files in given directory

if [ -d "$1" ];
then
  echo "Total no. of files in $1:"
  find $1 -maxdepth 1 -type f | wc -l
else
  echo " $1 is not a directory"
fi
```

```
shefali@shefali-VirtualBox:~/
```

```
shefali@shefali-VirtualBox:~$ ./prog5.sh Videos
```

```
Total no. of files in Videos:
```

```
0
```

```
shefali@shefali-VirtualBox:~$ ./prog5.sh new_dir
```

```
Total no. of files in new_dir:
```

```
0
```

```
shefali@shefali-VirtualBox:~$ ./prog5.sh Desktop
```

```
Total no. of files in Desktop:
```

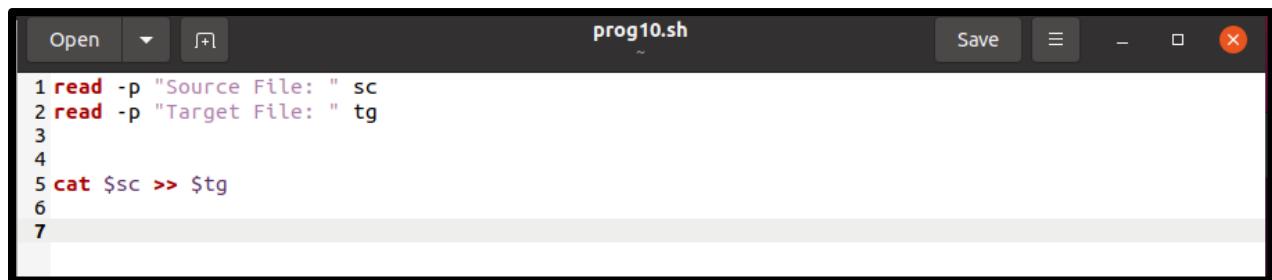
```
0
```

```
shefali@shefali-VirtualBox:~$ ./prog5.sh Documents
```

```
Total no. of files in Documents:
```

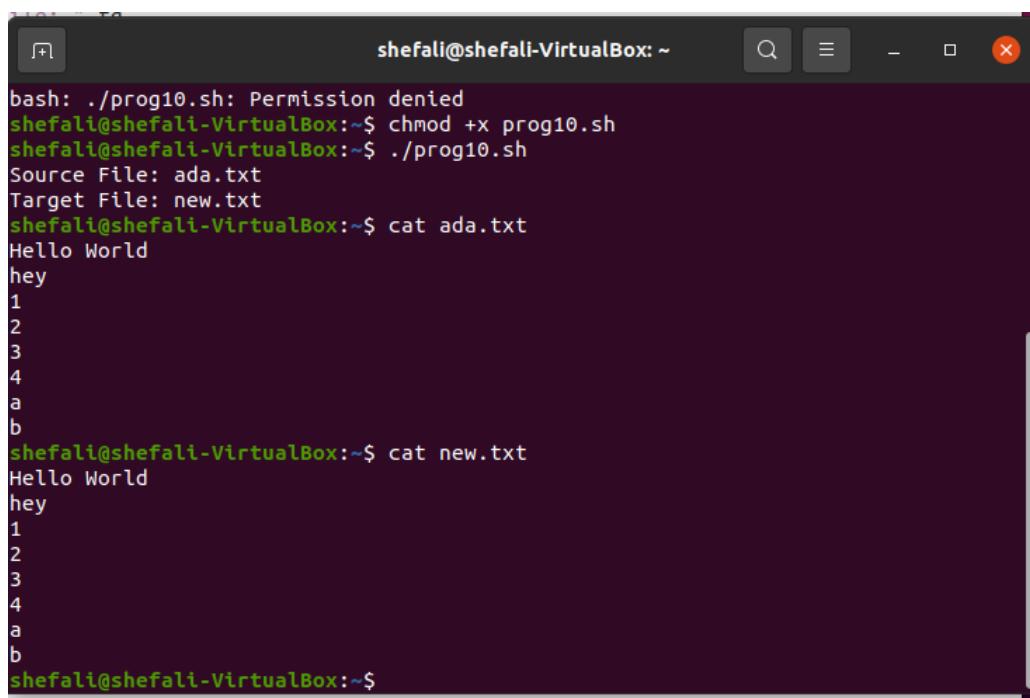
```
0
```

```
shefali@shefali-VirtualBox:~$
```



A screenshot of a terminal window titled "prog10.sh". The window has a dark theme with white text. It contains the following code:

```
1 read -p "Source File: " sc
2 read -p "Target File: " tg
3
4
5 cat $sc >> $tg
6
7
```



A screenshot of a terminal window titled "shefali@shefali-VirtualBox: ~". The window has a dark theme with white text. It shows the following session:

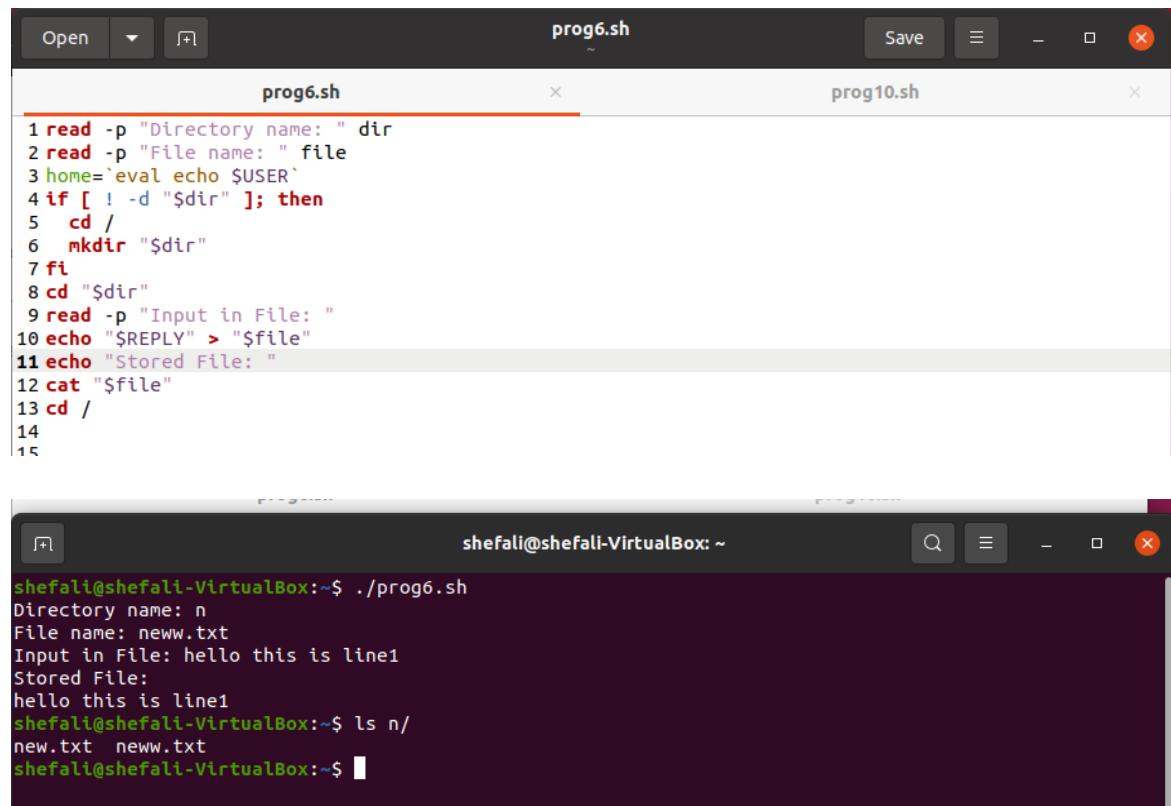
```
bash: ./prog10.sh: Permission denied
shefali@shefali-VirtualBox:~$ chmod +x prog10.sh
shefali@shefali-VirtualBox:~$ ./prog10.sh
Source File: ada.txt
Target File: new.txt
shefali@shefali-VirtualBox:~$ cat ada.txt
Hello World
hey
1
2
3
4
a
b
shefali@shefali-VirtualBox:~$ cat new.txt
Hello World
hey
1
2
3
4
a
b
shefali@shefali-VirtualBox:~$
```

EXPERIMENT 7-B

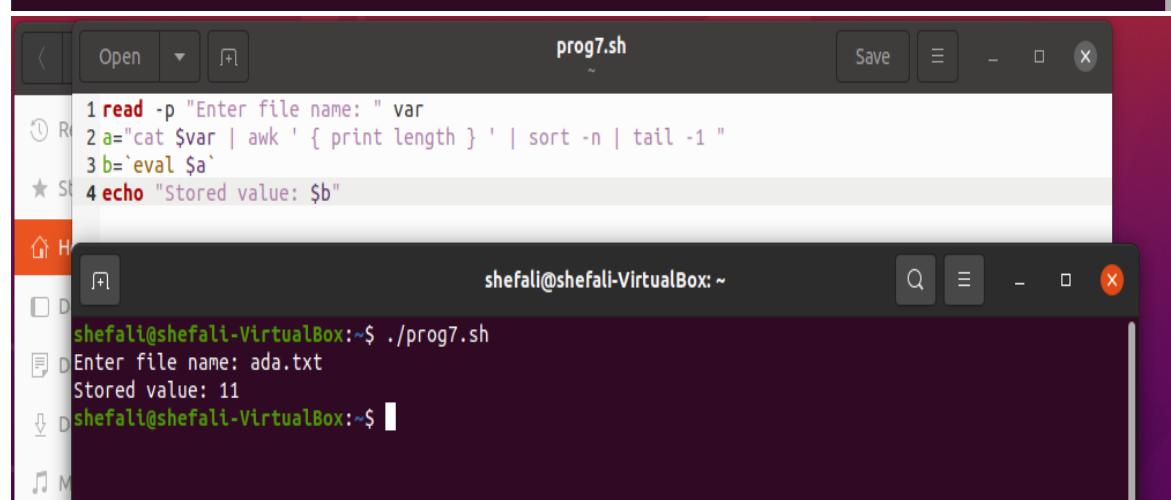
AIM: Implementation of shell script programs.

1. Create a directory, write contents on that and Copy to a suitable location in your home directory.
2. Use a pipeline and command substitution to set the length of a line in file to a variable.
3. Write a program using sed command to print duplicate lines of Input.

PROCEDURE:



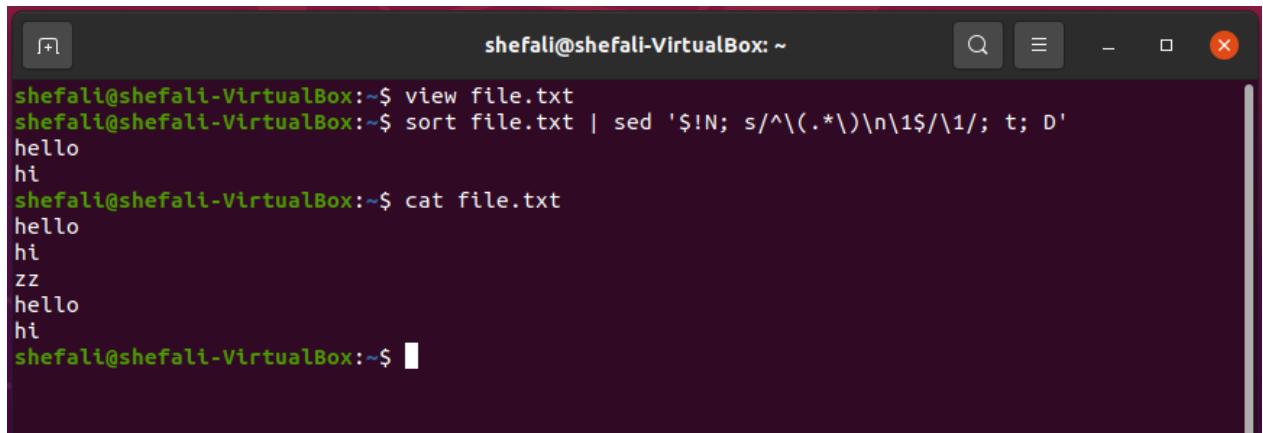
```
shefali@shefali-VirtualBox:~/Desktop$ ./prog6.sh
Directory name: n
File name: neww.txt
Input in File: hello this is line1
Stored File:
hello this is line1
shefali@shefali-VirtualBox:~/Desktop$ ls n/
new.txt  neww.txt
shefali@shefali-VirtualBox:~/Desktop$
```



```
shefali@shefali-VirtualBox:~/Desktop$ ./prog7.sh
Enter file name: ada.txt
Stored value: 11
shefali@shefali-VirtualBox:~/Desktop$
```

```
1 read -p "Directory name: " dir
2 read -p "File name: " file
3 home=`eval echo $USER`
4 if [ ! -d "$dir" ]; then
5   cd /
6   mkdir "$dir"
7 fi
8 cd "$dir"
9 read -p "Input in File: "
10 echo "$REPLY" > "$file"
11 echo "Stored File: "
12 cat "$file"
13 cd /
14
15
```

```
1 read -p "Enter file name: " var
2 a="cat $var | awk '{ print length }' | sort -n | tail -1 "
3 b='eval $a'
4 echo "Stored value: $b"
```



A screenshot of a terminal window titled "shefali@shefali-VirtualBox: ~". The window contains the following command-line session:

```
shefali@shefali-VirtualBox:~$ view file.txt
shefali@shefali-VirtualBox:~$ sort file.txt | sed '$!N; s/^(\.*\)\n\1$/\1/; t; D'
hello
hi
shefali@shefali-VirtualBox:~$ cat file.txt
hello
hi
zz
hello
hi
shefali@shefali-VirtualBox:~$
```

EXPERIMENT 8

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

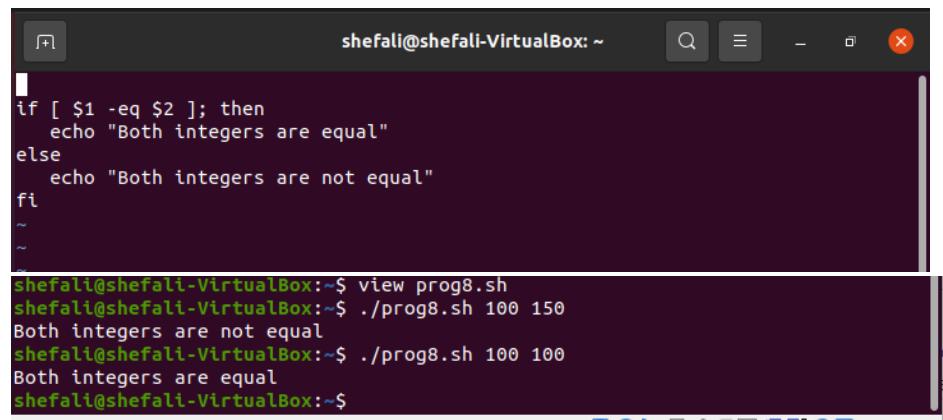
- a. Integer Comparison
- b. String comparison
- c. Logical operators
- d. File tests
- e. Conditional control structure
- f. Loop control structures

THEORY & PROCEDURE:

Integer Comparison

The general form of integer comparisons is *int1 -operator int2*. The following are available:

- eq : Integer equality
- ne : Integer inequality
- lt : Integer less than
- gt : Integer greater than
- le : Integer less than or equal to
- ge : Integer greater than or equal to



The screenshot shows a terminal window titled "shefali@shefali-VirtualBox: ~". The window contains two parts of text. The top part is a shell script named "prog8.sh" with the following content:

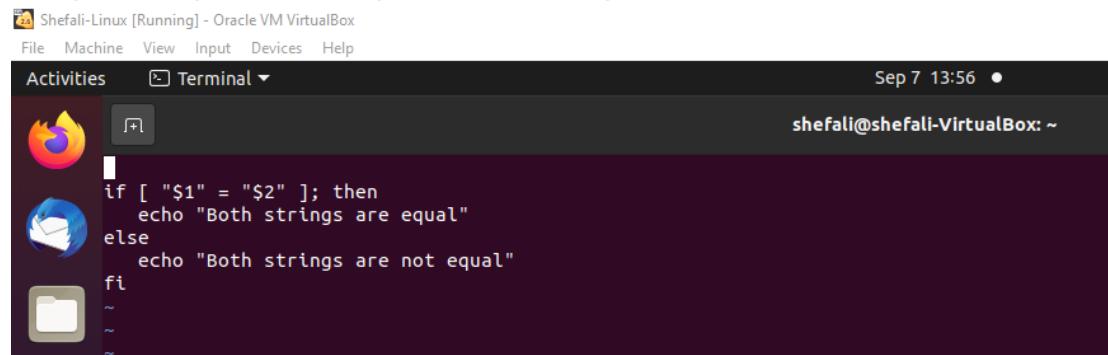
```
if [ $1 -eq $2 ]; then
    echo "Both integers are equal"
else
    echo "Both integers are not equal"
fi
```

The bottom part shows the execution of the script with two different pairs of integers:

```
shefali@shefali-VirtualBox:~$ view prog8.sh
shefali@shefali-VirtualBox:~$ ./prog8.sh 100 150
Both integers are not equal
shefali@shefali-VirtualBox:~$ ./prog8.sh 100 100
Both integers are equal
shefali@shefali-VirtualBox:~$
```

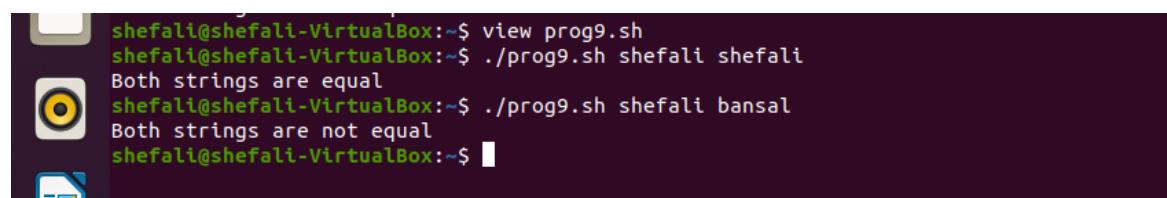
String Comparison:

Comparison operators are operators that compare values and return true or false.



The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "Terminal". The command entered is:

```
if [ "$1" = "$2" ]; then
    echo "Both strings are equal"
else
    echo "Both strings are not equal"
fi
```



The screenshot shows a terminal window with the following history:

```
shefali@shefali-VirtualBox:~$ view prog9.sh
shefali@shefali-VirtualBox:~$ ./prog9.sh shefali shefali
Both strings are equal
shefali@shefali-VirtualBox:~$ ./prog9.sh shefali bansal
Both strings are not equal
shefali@shefali-VirtualBox:~$
```

Logical Operators:

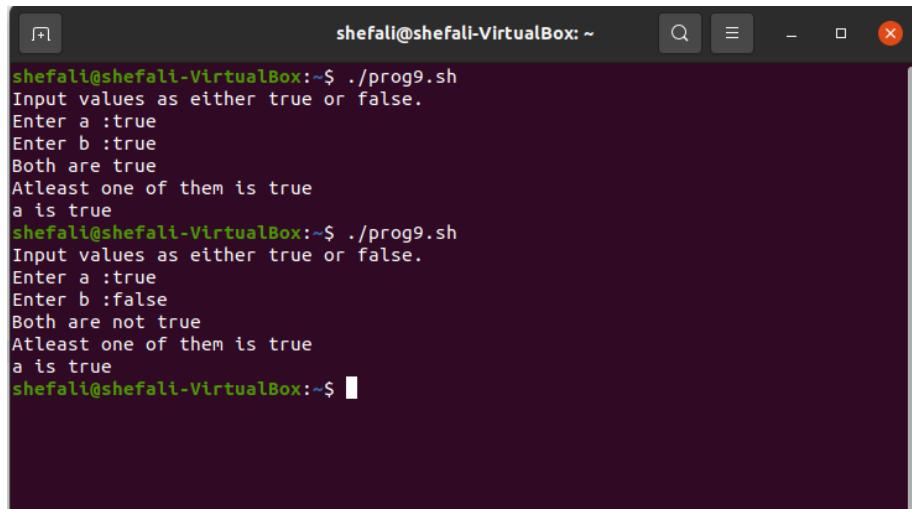
They are also known as boolean operators. These are used to perform logical operations. They are of 3 types:

- Logical AND (&&): This is a binary operator, which returns true if both the operands are true otherwise returns false.
- Logical OR (||): This is a binary operator, which returns true if either of the operands is true or both the operands are true and return false if none of them is false.
- Not Equal to (!): This is a unary operator which returns true if the operand is false and returns false if the operand is true.



The screenshot shows a code editor window with a file named "*prog9.sh". The script contains the following code:

```
1 echo "Input values as either true or false."
2
3 read -p 'Enter a :' a
4 read -p 'Enter b :' b
5
6 if (( $a == "true" && $b == "true" ))
7 then
8     echo "Both are true"
9 else
10    echo "Both are not true"
11 fi
12
13 if (( $a == "true" || $b == "true" ))
14 then
15     echo "Atleast one of them is true"
16 else
17     echo "None of them is true."
18 fi
19
20 if (( ! $a == "true" ))
21 then
22     echo "a is false."
23 else
24     echo "a is true"
25 fi
```



```
shefali@shefali-VirtualBox:~$ ./prog9.sh
Input values as either true or false.
Enter a :true
Enter b :true
Both are true
Atleast one of them is true
a is true
shefali@shefali-VirtualBox:~$ ./prog9.sh
Input values as either true or false.
Enter a :true
Enter b :false
Both are not true
Atleast one of them is true
a is true
shefali@shefali-VirtualBox:~$
```

Tests:

The test command is used to check file types and compare values. Test is used in conditional execution.

```
shefali@shefali-VirtualBox:~$ test -r prog9.sh && echo file has read permission
file has read permission
shefali@shefali-VirtualBox:~$ test -x prog9.sh && echo file has exec permission
file has exec permission
shefali@shefali-VirtualBox:~$ test -x ada.txt && echo file has exec permission || echo file does
not have exec permission
file does not have exec permission
shefali@shefali-VirtualBox:~$
```

Conditional Control Structures:

The control structures have as their test the execution of a Linux command.

if-else statement

If the specified condition is not true in if part then else part will be executed.

```
shefali@shefali-VirtualBox:~$ if [ 10 -gt 20 ]; then echo greater; else echo lesser; fi
lesser
shefali@shefali-VirtualBox:~$ a="true"
shefali@shefali-VirtualBox:~$ if [ '$a' = "true" ]
> then
>   echo yes
> else
>   echo no
> fi
yes
shefali@shefali-VirtualBox:~$
```

Loop Control Structures:

The BASH shell has three loop control structures: while, for, and for-in. The control structures have as their test the execution of a Linux command. All Linux commands return an exit status after they have finished executing.

```
shefali@shefali-VirtualBox:~$ a=0
shefali@shefali-VirtualBox:~$ while [ $a -lt 10 ]
> do
>   echo $a
>   a=`expr $a + 1`
> done
0
1
2
3
4
5
6
7
8
9
shefali@shefali-VirtualBox:~$
```

```
shefali@shefali-VirtualBox:~$ for a in 1 2 3 4 5
> do
>   echo $a
> done
1
2
3
4
5
shefali@shefali-VirtualBox:~$
```

```
shefali@shefali-VirtualBox:~$ a=0
shefali@shefali-VirtualBox:~$ until [ $a -gt 5 ]
> do
>   echo $a
>   a=`expr $a + 1`
> done
0
1
2
3
4
5
shefali@shefali-VirtualBox:~$
```

EXPERIMENT 9

AIM: Design an application by using various logical, conditional and arithmetic operators.

PROCEDURE:

1. Write a shell script to print a number in reverse order.

The screenshot shows a terminal window with three tabs at the top: *prog6.sh*, prog3.sh (which is the active tab), and prog4.sh. The code in prog3.sh is a shell script to reverse a number. The terminal window below shows the script being run and its output.

```
1 echo "Enter num: "
2 read num
3
4 sd=0
5 rev=0
6 while [ $num -gt 0 ]
7 do
8     sd=$(( $num % 10 ))
9     rev=$(( $rev * 10 + $sd ))
10    num=$(( $num / 10 ))
11 done
12 echo "Reverse: $rev"
```

Terminal output:

```
shefali@shefali-VirtualBox:~$ ./prog3.sh
Enter num:
1456723
Reverse: 3276541
shefali@shefali-VirtualBox:~$
```

2. find the sum of all numbers in a file in Linux.

The screenshot shows a terminal window with the following content:

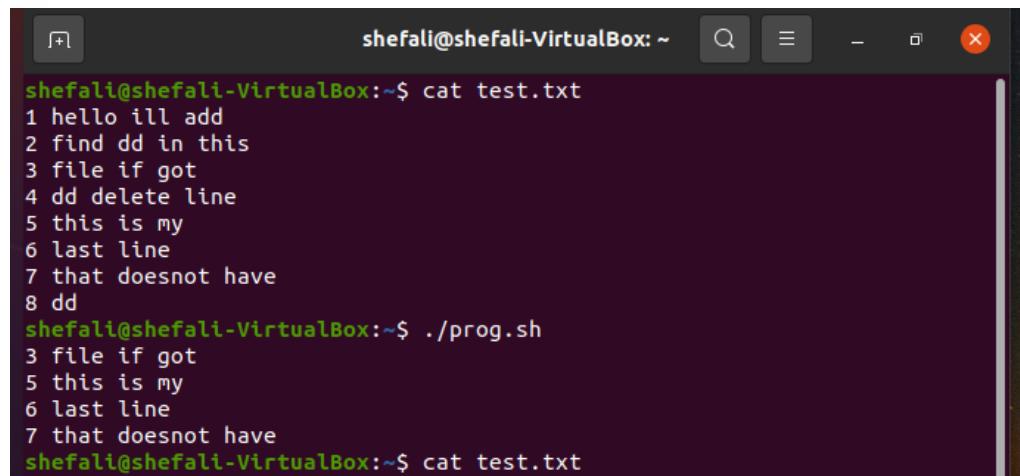
```
shefali@shefali-VirtualBox:~$ cat num.txt
1
2
3
4
5
6
shefali@shefali-VirtualBox:~$ ./prog4.sh
Sum: 21
shefali@shefali-VirtualBox:~$
```

The terminal window is titled "Terminal" and shows the user's session on a VirtualBox machine. The command `cat num.txt` was run to display the contents of the file `num.txt`, which contains the numbers 1 through 6. The command `./prog4.sh` was then run to execute the script `prog4.sh`, which calculates the sum of the numbers in the file and outputs "Sum: 21".

3. Write a shell script to delete the lines containing a word <dd> if it appears between the 5th and 7th position.



```
1 sed '/dd/d' test.txt
2 for i in 'cat text.txt'
3 do
4     echo $i
5 done
```



```
shefali@shefali-VirtualBox:~$ cat test.txt
1 hello ill add
2 find dd in this
3 file if got
4 dd delete line
5 this is my
6 last line
7 that doesnot have
8 dd
shefali@shefali-VirtualBox:~$ ./prog.sh
3 file if got
5 this is my
6 last line
7 that doesnot have
shefali@shefali-VirtualBox:~$ cat test.txt
```

4. 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.

The screenshot shows a terminal window with three tabs at the top: *prog6.sh*, prog10.sh, and *prog.sh*. The current tab, prog8.sh, contains the command:

```
1 tr ' ' '\n' < test.txt | sort | uniq -c
```

The terminal window displays the following session:

```
shefali@shefali-VirtualBox:~$ cat test.txt
hello ill add
find dd in this
file if got
dd delete line
this is my
last line
that doesnot have
dd
shefali@shefali-VirtualBox:~$ ./prog8.sh
this is my
last line
that doesnot have
dd
shefali@shefali-VirtualBox:~$ ./prog8.sh
 1 add
 3 dd
 1 delete
 1 doesnot
 1 file
 1 find
 1 got
 1 have
 1 hello
 1 if
 1 ill
 1 in
 1 is
 1 last
 2 line
 1 my
 1 that
 2 this
shefali@shefali-VirtualBox:~$
```

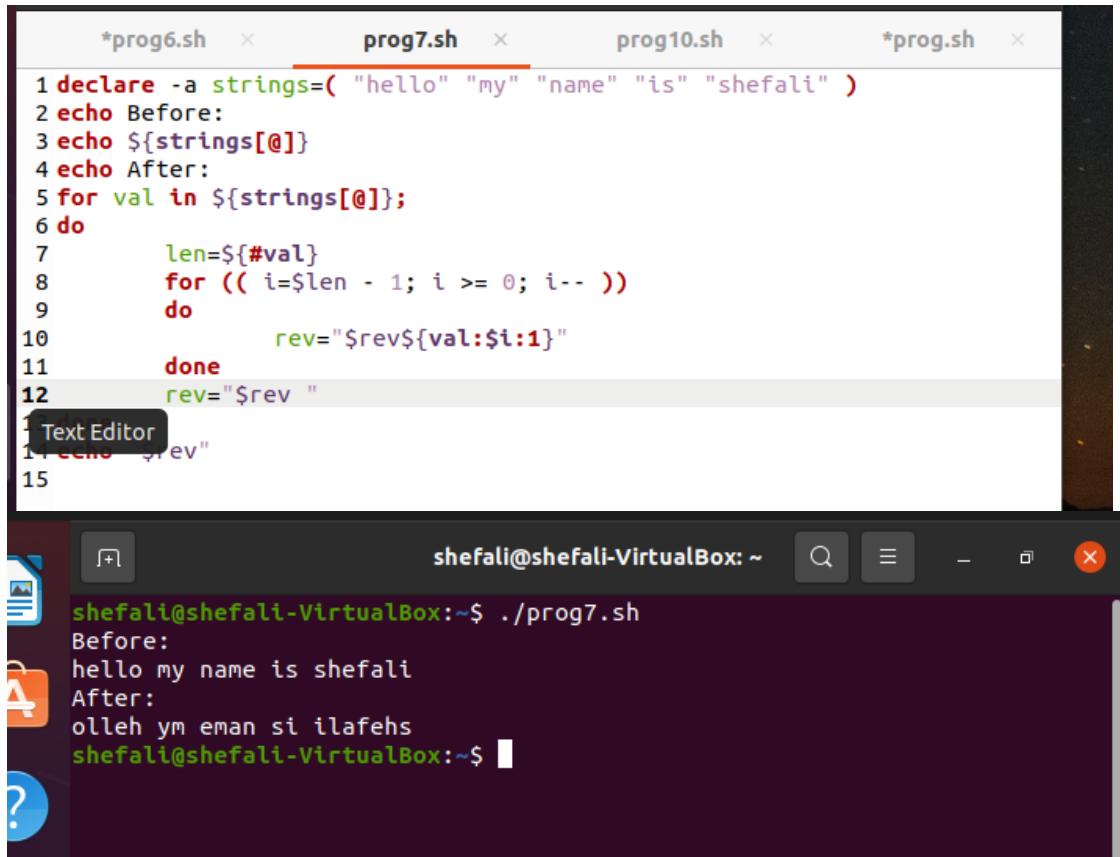
The terminal window has a dark background and light-colored text. It includes standard Linux terminal icons for navigation and file operations at the bottom.

5. Write a shell script to validate password strength. Here are a few assumptions for the password string.Length – minimum of 8 characters,Contain both alphabet and number,Include both the small and capital case letters.

```
prog6.sh          prog7.sh
1 read -p "Enter password: " pwd
2 len="#$pwd"
3 if test $len -ge 8 ; then
4     echo "$pwd" | grep -q "[0-9]"
5     if test $? -eq 0 ; then
6         echo "$pwd" | grep -q "[A-Z]"
7         if test $? -eq 0 ; then
8             echo "$pwd" | grep -q "[a-z]"
9             if test $? -eq 0 ; then
10                echo "Strong Password"
11            else
12                echo "Weak: include small alphabets."
13            fi
14        else
15            echo "Weak: include capital alphabets."
16        fi
17    else
18        echo "Weak: include digits."
19    fi
20 else
21     echo "Weak: minimum length must be 8."
22 fi
J+1
```

```
shefali@shefali-VirtualBox:~$ ./prog6.sh
Enter password: Shefali321
Strong Password
shefali@shefali-VirtualBox:~$ ./prog6.sh
Enter password: shefa
Weak: minimum length must be 8.
shefali@shefali-VirtualBox:~$ ./prog6.sh
Enter password: shefali12
Weak: include capital alphabets.
shefali@shefali-VirtualBox:~$ ./prog6.sh
Enter password: 12345678
Weak: include capital alphabets.
shefali@shefali-VirtualBox:~$ █
```

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



The image shows a terminal window titled "shefali@shefali-VirtualBox:~\$". It displays the execution of a shell script named "prog7.sh". The script's content is visible in the background, showing a loop that prints "Before:" followed by the original list of strings ("hello" "my" "name" "is" "shefali"), then "After:", and finally the reversed list ("olleh" "ym" "eman" "si" "ilafehs").

```
*prog6.sh ×      prog7.sh ×      prog10.sh ×      *prog.sh ×
1 declare -a strings=( "hello" "my" "name" "is" "shefali" )
2 echo Before:
3 echo ${strings[@]}
4 echo After:
5 for val in ${strings[@]};
6 do
7     len=${#val}
8     for (( i=$len - 1; i >= 0; i-- ))
9     do
10         rev="$rev${val:$i:1}"
11     done
12     rev="$rev "
13 done
14 echo $rev
15

Text Editor
shefali@shefali-VirtualBox:~$ ./prog7.sh
Before:
hello my name is shefali
After:
olleh ym eman si ilafehs
shefali@shefali-VirtualBox:~$
```

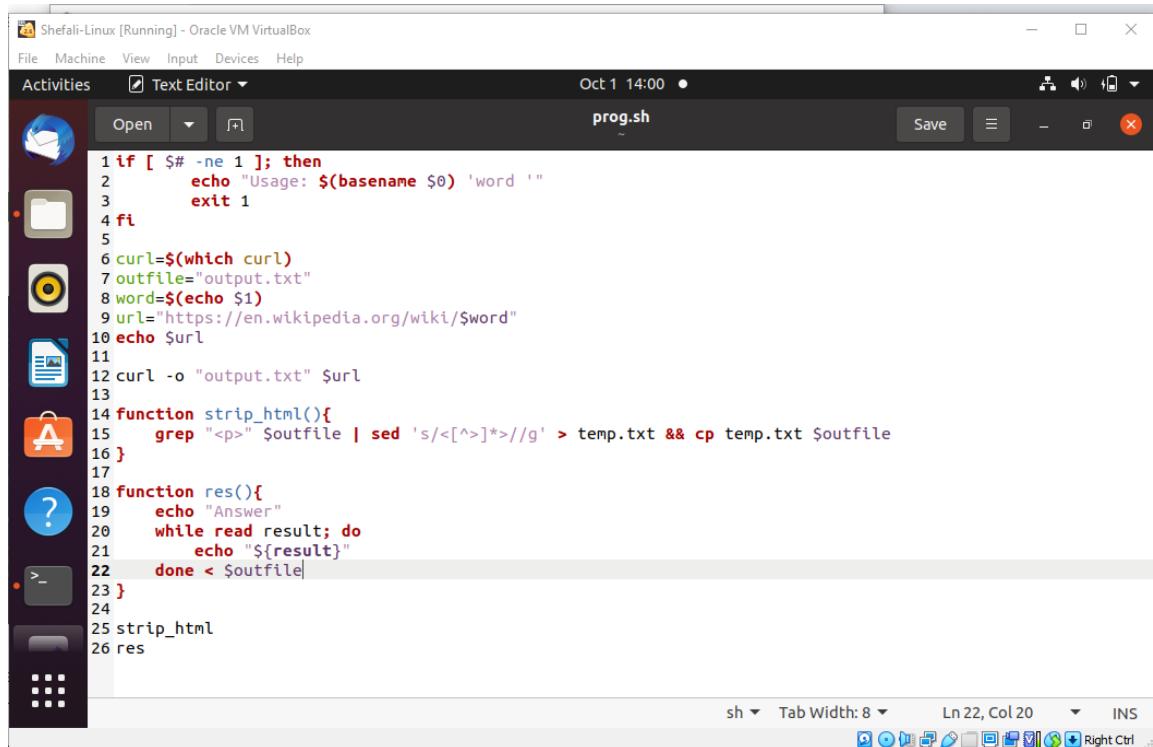
EXPERIMENT 10

AIM: Design a real life use case of Shell Scripting - Shell Script to scrap the definition of a word from Wikipedia

THEORY:

Web Scraping helps in analyzing data and getting some information in various formats. Web Scraping is a process in which a user fetches a website's content using some pattern in those HTML tags and the desired content to be fetched or scraped. The aim is to fetch the meaning of a word entered by the user from the Wikipedia website.

PROCEDURE:



The screenshot shows a Linux desktop environment with a terminal window open. The terminal window is titled 'prog.sh'. The script content is as follows:

```
1 if [ $# -ne 1 ]; then
2     echo "Usage: $(basename $0) 'word'"
3     exit 1
4 fi
5
6 curl=$(which curl)
7 outfile="output.txt"
8 word=${echo $1}
9 url="https://en.wikipedia.org/wiki/$word"
10 echo $url
11
12 curl -o "output.txt" $url
13
14 function strip_html(){
15     grep "<p>" $outfile | sed 's/<[^>]*>/g' > temp.txt && cp temp.txt $outfile
16 }
17
18 function res(){
19     echo "Answer"
20     while read result; do
21         echo "${result}"
22     done < $outfile
23 }
24
25 strip_html
26 res
```

OUTPUT:

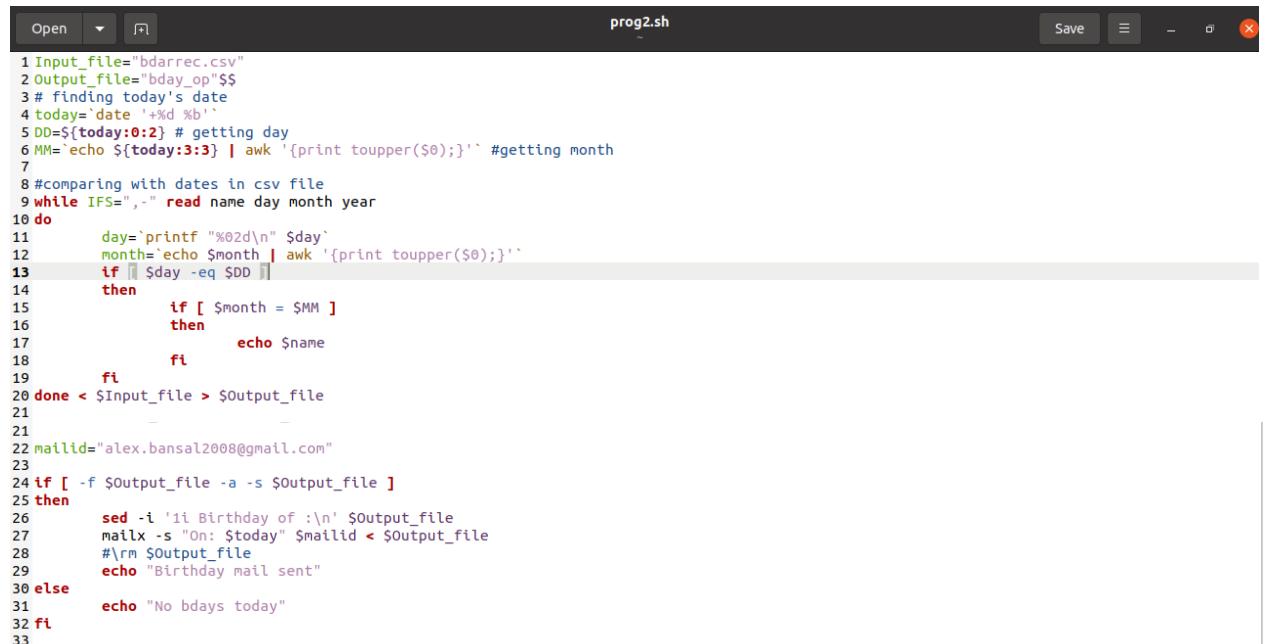
```
shefali@shefali-VirtualBox:~$ ./prog.sh Data
https://en.wikipedia.org/wiki/Data
    % Total      % Received % Xferd  Average Speed   Time     Time      Time  Current
          Dload  Upload Total   Spent   Left Speed
100 145k  100 145k    0      0  281k      0 --::--- --::--- --::--- 280k
Answer
Data (US: /'daɪtə/; UK: /'deɪtə/) are individual facts, statistics, or items of information, often numeric, that are collected through observation.1 In a more technical sense, data are a set of values of qualitative or quantitative variables about one or more persons or objects,1 while a datum (singular of data) is a single value of a single variable.2
Although the terms "data" and "information" are often used interchangeably, these terms have distinct meanings. In some popular publications, data are sometimes said to be transformed into information when they are viewed in context or in post-analysis.3 However, in academic treatments of the subject data are simply units of information. Data are used in scientific research, businesses management (e.g., sales data, revenue, profits, stock price), finance, governance (e.g., crime rates, unemployment rates, literacy rates), and in virtually every other form of human organizational activity (e.g., censuses of the number of homeless people by non-profit organizations).
Data are measured, collected, reported, and analyzed, and used to create data visualizations such as graphs, tables or images. Data as a general concept refers to the fact that some existing information or knowledge is represented or coded in some form suitable for better usage or processing. Raw data ("unprocessed data") is a collection of numbers or characters before it has been "cleaned" and corrected by researchers. R
```

```
shefali@shefali-VirtualBox:~$ ./prog.sh Linux
https://en.wikipedia.org/wiki/Linux
    % Total      % Received % Xferd  Average Speed   Time     Time      Time  Current
          Dload  Upload Total   Spent   Left Speed
100 571k  100 571k    0      0  775k      0 --::--- --::--- --::--- 775k
Answer
Linux (/linʊks/32; (listen) LEEN-uks or /lɪnʊks/ LIN-uks10;) is a family of open-source Unix-like operating systems based on the Linux kernel,11 a n operating system kernel first released on September 17, 1991, by Linus Torvalds.12 Linux is typically packaged in a Linux distribution .
Distributions include the Linux kernel and supporting system software and libraries, many of which are provided by the GNU Project. Many Linux distributions use the word "Linux" in their name, but the Free Software Foundation uses the name "GNU/Linux" to emphasize the importance of GNU software, causing some controversy.15
Popular Linux distributions17 include Debian, Fedora, and Ubuntu. Commercial distributions include Red Hat Enterprise Linux and SUSE Linux Enterprise Server. Desktop Linux distributions include a windowing system such as X11 or Wayland, and a desktop environment such as GNOME or KDE Plasma. Distributions intended for servers may omit graphics altogether, or include a solution stack such as LAMP. Because Linux is freely redistributable, anyone may create a distribution for any purpose.20
Linux was originally developed for personal computers based on the Intel x86 architecture, but has since been ported to more platforms than any other operating system.9
```

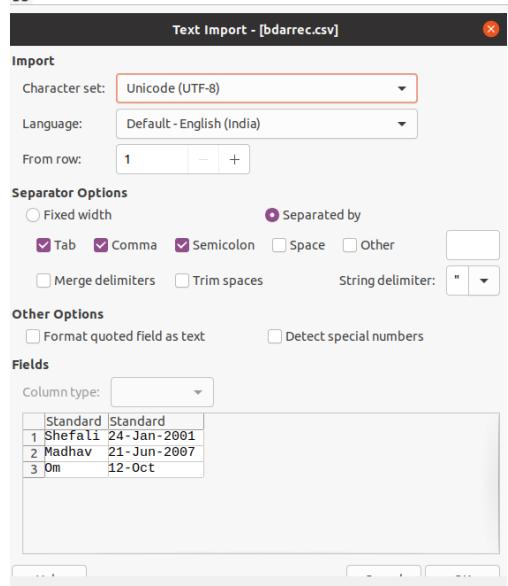
EXPERIMENT 11

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

PROCEDURE:



```
1 Input_file="bdarrec.csv"
2 Output_file="bday_op"$$
3 # finding today's date
4 today=`date +%d %b`
5 DD=${today:0:2} # getting day
6 MM=`echo ${today:3:3} | awk '{print toupper($0)}'` #getting month
7
8 #comparing with dates in csv file
9 while IFS="," read name day month year
10 do
11     day=`printf "%02d\n" $day`
12     month=`echo $month | awk '{print toupper($0)}'`
13     if [ $day -eq $DD ]
14     then
15         if [ $month = $MM ]
16         then
17             echo $name
18         fi
19     fi
20 done < $Input_file > $Output_file
21
22 mailid="alex.bansal2008@gmail.com"
23
24 if [ -f $Output_file -a -s $Output_file ]
25 then
26     sed -i 'i Birthday of :\n' $Output_file
27     mailx -s "On: $today" $mailid < $Output_file
28     #rm $Output_file
29     echo "Birthday mail sent"
30 else
31     echo "No bdays today"
32 fi
33
```



Index	Name	Date
1	Shefali	24-Jan-2001
2	Madhav	21-Jun-2007
3	Om	12-Oct

```
shefali@shefali-VirtualBox:~$ ./prog2.sh
Birthday mail sent
shefali@shefali-VirtualBox:~$
```

```
Return-Path: <shefali@shefali-VirtualBox>
Received: by shefali-VirtualBox (Postfix, from userid 1000)
          id 5F22848301; Tue, 12 Oct 2021 14:52:29 +0530 (IST)
Subject: On: 12 Oct
To: <alex.bansal2008@gmail.com>
X-Mailer: mail (GNU Mailutils 3.7)
Message-Id: <20211012092229.5F22848301@shefali-VirtualBox>
Date: Tue, 12 Oct 2021 14:52:29 +0530 (IST)
From: shefali <shefali@shefali-VirtualBox>

Birthday of :

Om

--5F22848301.1634030813/shefali-VirtualBox--
?
Return-Path: <>
X-Original-To: shefali@shefali-VirtualBox
Delivered-To: shefali@shefali-VirtualBox
Received: by shefali-VirtualBox (Postfix)
          id 596674623F; Tue, 12 Oct 2021 14:56:53 +0530 (IST)
Date: Tue, 12 Oct 2021 14:56:53 +0530 (IST)
From: MAILER-DAEMON@shefali-VirtualBox (Mail Delivery System)
```

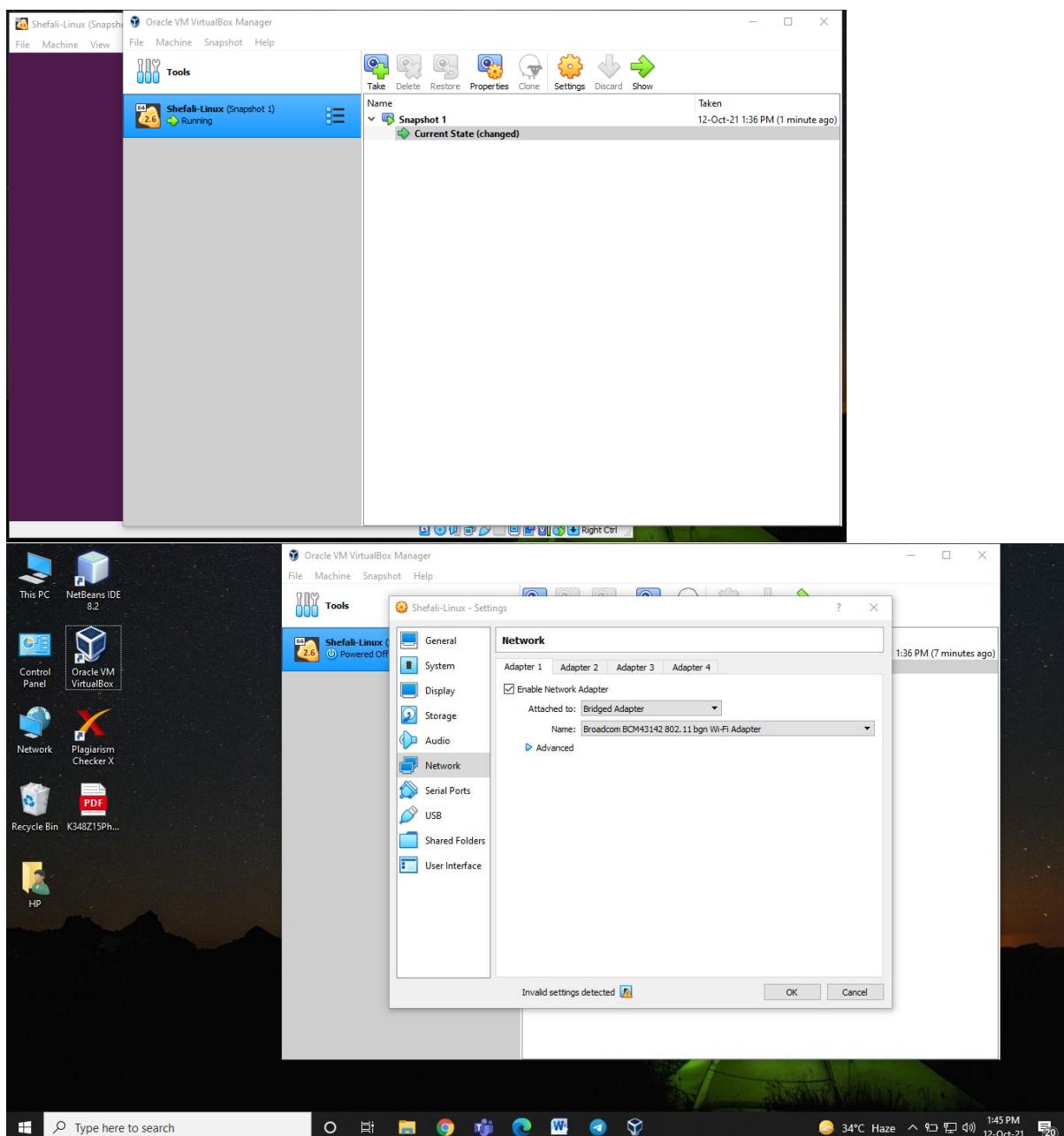
EXPERIMENT 12

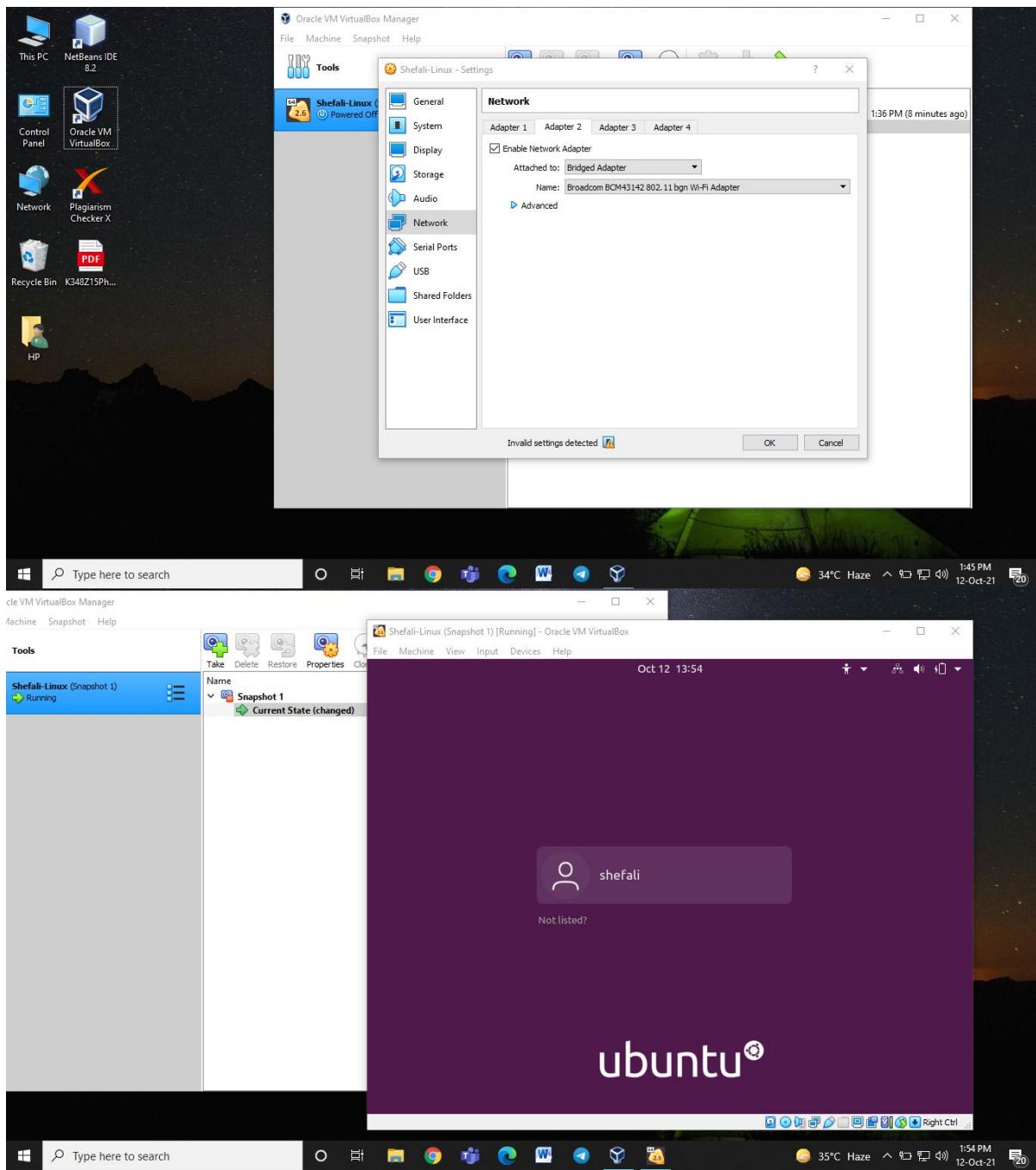
AIM: Implement NIC Bonding and teaming.

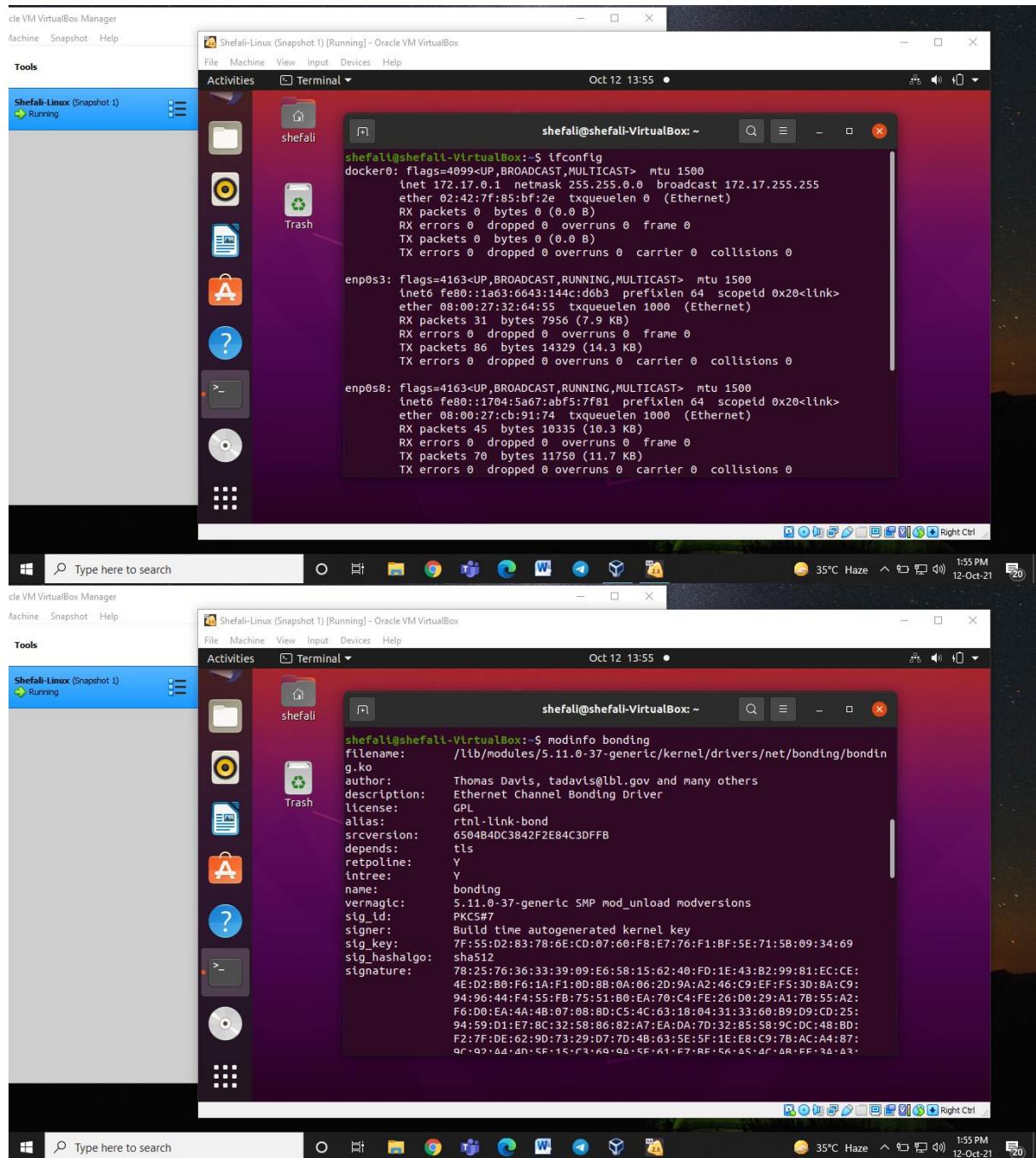
THEORY:

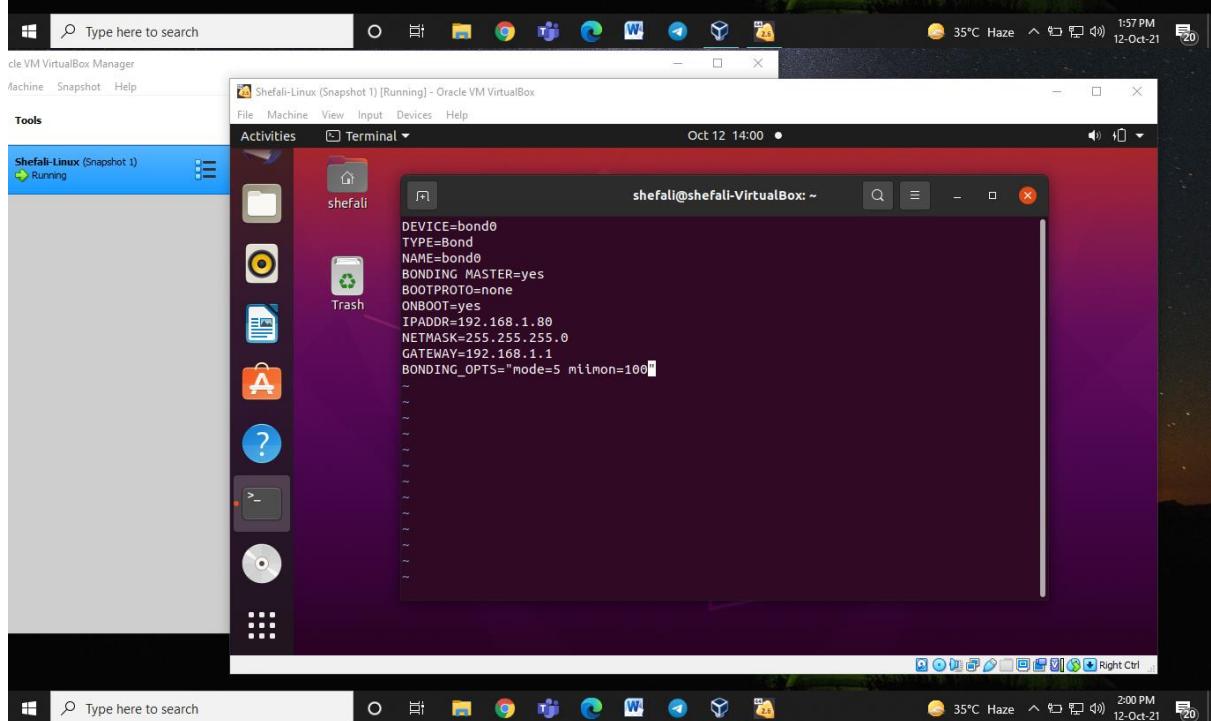
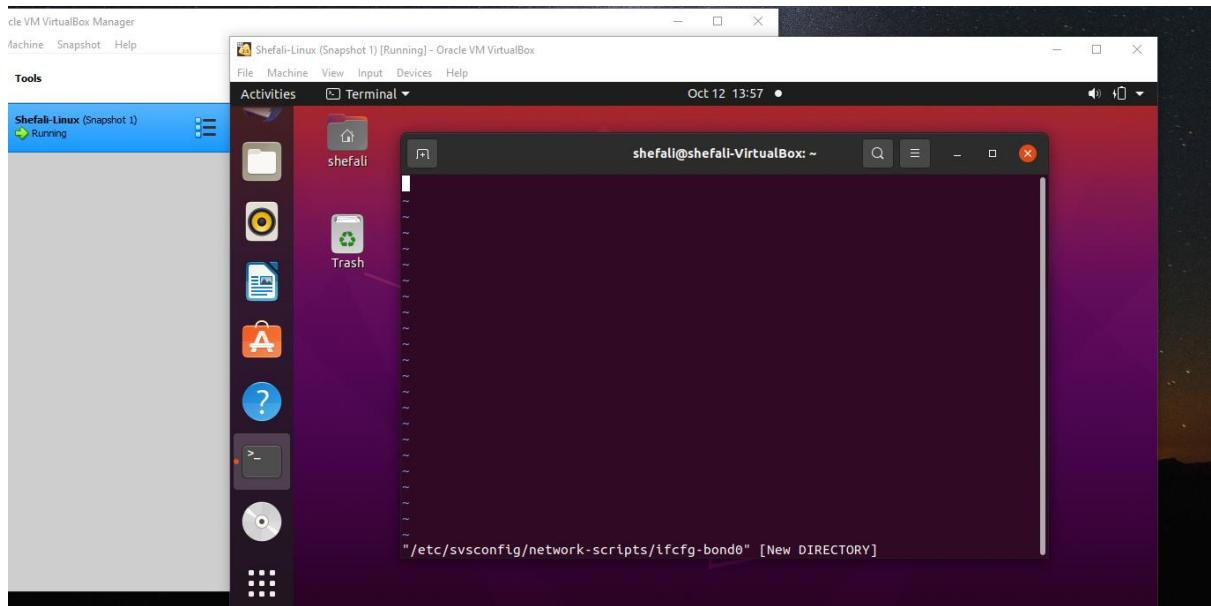
Network Interface Bonding is a mechanism used in Linux servers which consists of binding more physical network interfaces in order to provide more bandwidth than a single interface can provide or provide link redundancy in case of a cable failure. This type of link redundancy has multiple names in Linux, such as **Bonding**, **Teaming** or **Link Aggregation Groups (LAG)**.

PROCEDURE:









Shefali-Linux (Snapshot 1) [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Activities Terminal Oct 12 14:32

```
GNU nano 4.8 /etc/network/interfaces Modified
auto bond0
iface bond0 inet static
    address 192.168.1.150
    netmask 255.255.255.0
    gateway 192.168.1.1
    dns-nameservers 192.168.1.1 8.8.8.8
    dns-search domain.local
        slaves eth0 eth1
        bond_mode 0
        bond-milmon 100
        bond_downdelay 200
        bond_updelay 200
```

Get Help Write Out Where Is Cut Text Justify Cur Pos
 Exit Read File Replace Paste Text To Spell Go To Line

snmp speech-dispatcher ssh ssl sudoers.d systemctl

shefali@shefali-VirtualBox: ~

configuration: autonegotiation=off broadcast=yes driver=bonding driverversion=5.11.0-37-generic firmware=2 ip=192.168.1.150 link=no mas ter=yes multicast=yes

shefali@shefali-VirtualBox: \$ sudo ip link set enp0s3 master bond0

Error: Device can not be enslaved while up.

shefali@shefali-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 08:00:27:32:64:55 brd ff:ff:ff:ff:ff:ff
 inet 192.168.1.33/24 brd 192.168.1.255 scope global dynamic noprefixroute enp0s3
 valid_lft 85825sec preferred_lft 85825sec
 inet6 fe80::1a63:6643:144c:d0b3/64 scope link noprefixroute
 valid_lft forever preferred_lft forever

3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
 link/ether 08:00:27:cb:91:74 brd ff:ff:ff:ff:ff:ff
 inet 192.168.1.32/24 brd 192.168.1.255 scope global dynamic noprefixroute enp0s8
 valid_lft 85824sec preferred_lft 85824sec
 inet6 fe80::1704:5a67:abf5:7f81/64 scope link noprefixroute
 valid_lft forever preferred_lft forever

4: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
 link/ether 02:42:73:4b:b2:ee brd ff:ff:ff:ff:ff:ff
 inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
 valid_lft forever preferred_lft forever

5: bond0: <NO-CARRIER,BROADCAST,MULTICAST,MASTER,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
 link/ether 8a:9a:03:ce:66:a1 brd ff:ff:ff:ff:ff:ff
 inet 192.168.1.150/24 brd 192.168.1.255 scope global bond0
 valid_lft forever preferred_lft forever

shefali@shefali-VirtualBox: \$

The image shows three identical screenshots of a Linux desktop environment, specifically Shefali-Linux, running in Oracle VM VirtualBox. Each screenshot displays a terminal window with the following command history:

```
shefali@shefali-VirtualBox:~$ sudo modprobe bonding
[sudo] password for shefali:
shefali@shefali-VirtualBox:~$ lsmod | grep bond
bonding           172032  0
tls                90112  1 bonding
shefali@shefali-VirtualBox:~$ ifenslave
Not enough arguments
shefali@shefali-VirtualBox:~$ sudo ip link add bond0 type bond mode 802.3ad
RTNETLINK answers: File exists
shefali@shefali-VirtualBox:~$ sudo ip link set eth0 master bond0
Cannot find device "eth0"
shefali@shefali-VirtualBox:~$ sudo ink set eth0 master bond0
Unknown port specified.
Could not get ink level.
shefali@shefali-VirtualBox:~$ sudo ip link set eth0 master bond0
Cannot find device "eth0"
shefali@shefali-VirtualBox:~$ sudo lshw -C network
*-network:0
  description: Ethernet interface
  product: 82540EM Gigabit Ethernet Controller
  vendor: Intel Corporation
  physical id: 3
  bus info: pci@0000:00:03.0
  logical name: enp0s3
  version: 02
  serial: 08:00:27:32:64:55
  size: 1Gbit/s
  capacity: 1Gbit/s
  width: 32 bits
  clock: 66MHz
  capabilities: pm pcix bus_master cap_list ethernet physical tp 10bt 10bt-fd 100bt 100bt-fd 1000bt-fd autonegotiation
  configuration: autonegotiation=on broadcast=yes driver=e1000 driverversion=5.11.0-37-generic duplex=full ip=192.168.1.33 latency=64 link-state=up
  resources: irq:19 memory:f0200000-f021ffff ioport:d020(size=8)
k=yes mngt=z55 Multicast=yes port=twisted pair speed=1Gbit/s
  resources: irq:19 memory:f0200000-f021ffff ioport:d020(size=8)
*-network:1
  description: Ethernet interface
  product: 82540EM Gigabit Ethernet Controller
  vendor: Intel Corporation
  physical id: 8
  bus info: pci@0000:00:08.0
  logical name: enp0s8
  version: 02
  serial: 08:00:27:cb:91:74
  size: 1Gbit/s
  capacity: 1Gbit/s
  width: 32 bits
  clock: 66MHz
  capabilities: pm pcix bus_master cap_list ethernet physical tp 10bt 10bt-fd 100bt 100bt-fd 1000bt-fd autonegotiation
  configuration: autonegotiation=on broadcast=yes driver=e1000 driverversion=5.11.0-37-generic duplex=full link-state=up
  resources: irq:19 memory:f0200000-f021ffff ioport:d020(size=8)
```

The desktop environment includes a top bar with file, machine, view, input, devices, and help options, and a system tray showing the date, time, and battery status.

Shefali-Linux (Snapshot 1) [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Activities Terminal Oct 12 15:14 •

```
shefali@shefali-VirtualBox: ~
```

```
capacity: 1abit/s
width: 32 bits
clock: 66MHz
capabilities: pm pcix bus_master cap_list ethernet physical tp 10bt 10bt-fd 100bt 100bt-fd 1000bt-fd autonegotiation
configuration: autonegotiation=on broadcast=yes driver=e1000 driverversion=5.11.0-37-generic duplex=full ip=192.168.1.33 latency=64 link=yes mingnt=255 multicast=yes port=twisted pair speed=1Gbit/s
resources: irq:19 memory:f0200000-f021ffff ioport:d020(size=8)
*-network:1
    description: Ethernet interface
    product: 82540EM Gigabit Ethernet Controller
    vendor: Intel Corporation
    physical id: 8
    bus info: pci@0000:00:08.0
    logical name: enp0s8
    version: 02
    serial: 08:00:27:cb:91:74
    size: 1abit/s
    capacity: 1abit/s
    width: 32 bits
    clock: 66MHz
    capabilities: pm pcix bus_master cap_list ethernet physical tp 10bt 10bt-fd 100bt 100bt-fd 1000bt-fd autonegotiation
    configuration: autonegotiation=on broadcast=yes driver=e1000 driverversion=5.11.0-37-generic duplex=full ip=192.168.1.32 latency=64 link=yes mingnt=255 multicast=yes port=twisted pair speed=1Gbit/s
    resources: irq:16 memory:f0820000-f083ffff ioport:d240(size=8)
*-network
    description: Ethernet interface
    physical id: 1
    logical name: bond0
    serial: 8a:9a:03:ce:66:ac
    capabilities: ethernet physical
    configuration: autonegotiation=off broadcast=yes driver=bonding driverversion=5.11.0-37-generic firmware=2 ip=192.168.1.150 link=no mas
ter=yes multicast=yes
shefali@shefali-VirtualBox: ~ $ sudo ip link set enp0s8 master bond0
```

Windows Type here to search 35°C 3:14 PM 12-Oct-21

File Machine View Input Devices Help

Activities Terminal Oct 12 15:14 •

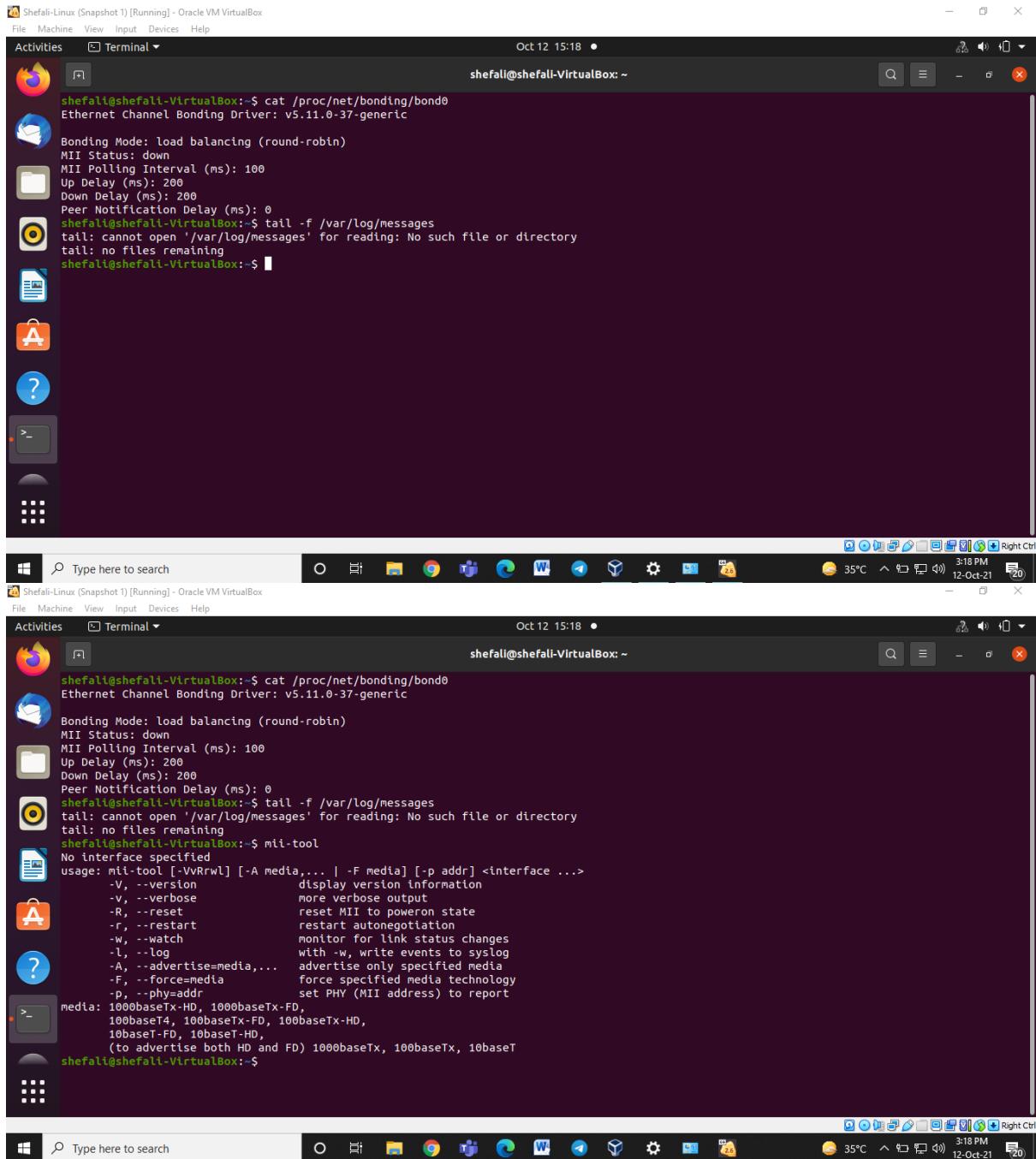
```
shefali@shefali-VirtualBox: ~
```

```
GNU nano 4.8 /etc/network/interfaces
```

```
auto bond0
iface bond0 inet static
    address 192.168.1.150
    netmask 255.255.255.0
    gateway 192.168.1.1
    dns-nameservers 192.168.1.1.8.8.8.8
    dns-search domain.local
    slaves eth0 eth1
    bond_mode 0
    bond_milmon 100
    bond_downdelay 200
    bond_updelay 200
```

Get Help Write Out Where Is Replace Cut Text Paste Text Justify Cur Pos Go To Line Undo Redo Mark Text Copy Text Where Was

Windows Type here to search 35°C 3:14 PM 12-Oct-21



Shefali-Linux (Snapshot 1) [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Activities Terminal Oct 12 15:18 shefali@shefali-VirtualBox: ~

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

Bonding Mode: load balancing (round-robin)
MII Status: down
MII Polling Interval (ms): 100
Up Delay (ms): 200
Down Delay (ms): 200
Peer Notification Delay (ms): 0
shefali@shefali-VirtualBox:~$ tail -f /var/log/messages
tail: cannot open '/var/log/messages' for reading: No such file or directory
tail: no files remaining
shefali@shefali-VirtualBox:~$
```

Type here to search

File Machine View Input Devices Help

Activities Terminal Oct 12 15:18 shefali@shefali-VirtualBox: ~

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

Bonding Mode: load balancing (round-robin)
MII Status: down
MII Polling Interval (ms): 100
Up Delay (ms): 200
Down Delay (ms): 200
Peer Notification Delay (ms): 0
shefali@shefali-VirtualBox:~$ tail -f /var/log/messages
tail: cannot open '/var/log/messages' for reading: No such file or directory
tail: no files remaining
shefali@shefali-VirtualBox:~$ mii-tool
No interface specified
usage: mii-tool [-VVRrw] [-A media,... | -F media] [-p addr] <interface ...>
  -V, --version           display version information
  -v, --verbose            more verbose output
  -R, --reset              reset MII to poweron state
  -r, --restart             restart autonegotiation
  -w, --watch               monitor for link status changes
  -l, --log                 with -w, write events to syslog
  -A, --advertise=media,... advertise only specified media
  -F, --force=media          force specified media technology
  -p, --phy=addr            set PHY (MII address) to report
media: 1000baseTx-HD, 1000baseTx-FD,
       100baseT4, 100baseTx-FD, 100baseTx-HD,
       10baseT-FD, 10baseT-HD,
       (to advertise both HD and FD) 1000baseTx, 100baseTx, 10baseT
shefali@shefali-VirtualBox:~$
```

Type here to search

Shefali-Linux (Snapshot 1) [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Activities Terminal Oct 12 15:20 •

```
shefali@shefali-VirtualBox: ~
```

```
Bonding Mode: load balancing (round-robin)
MII Status: down
MII Polling Interval (ms): 100
Up Delay (ms): 200
Down Delay (ms): 200
Peer Notification Delay (ms): 0
shefali@shefali-VirtualBox: $ tail -f /var/log/messages
tail: cannot open '/var/log/messages' for reading: No such file or directory
tail: no files remaining
shefali@shefali-VirtualBox: $ mii-tool
No interface specified
usage: mii-tool [-VvRrw] [-A media,... | -F media] [-p addr] <interface ...>
    -V, --version           display version information
    -v, --verbose            more verbose output
    -R, --reset              reset MII to poweron state
    -r, --restart             restart autonegotiation
    -w, --watch               monitor for link status changes
    -l, --log                 with -w, write events to syslog
    -A, --advertise=media,... advertise only specified media
    -F, --force=media         force specified media technology
    -p, --phy=addr            set PHY (MII address) to report
media: 100baseTx-HD, 1000baseTx-FD,
100baseT4, 100baseTx-FD, 100baseTx-HD,
10baseT-FD, 10baseT-HD,
(to advertise both HD and FD) 1000baseTx, 100baseTx, 10baseT
shefali@shefali-VirtualBox: $ mii-tool enp0s3
SIOCGMIIPHY on 'enp0s3' failed: Operation not permitted
shefali@shefali-VirtualBox: $ sudo mii-tool enp0s3
[sudo] password for shefali:
enp0s3: no autonegotiation, 1000baseT-FD flow-control, link ok
shefali@shefali-VirtualBox: $
```

Shefali-Linux (Snapshot 1) [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Activities Terminal Oct 12 15:13 •

```
shefali@shefali-VirtualBox: ~
```

```
configuration: autonegotiation=off broadcast=yes driver=bonding driverversion=5.11.0-37-generic firmware=2 ip=192.168.1.150 link=no mas
ter=yes multicast=yes
shefali@shefali-VirtualBox: $ sudo ip link set enp0s3 master bond0
Error: Device can not be enslaved while up.
shefali@shefali-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 08:00:27:32:64:55 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.33/24 brd 192.168.1.255 scope global dynamic noprefixroute enp0s3
        valid_lft 85825sec preferred_lft 85825sec
        inet6 fe80::1a63:6043:144c:dbb3/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:c9:91:74 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.32/24 brd 192.168.1.255 scope global dynamic noprefixroute enp0s8
        valid_lft 85824sec preferred_lft 85824sec
        inet6 fe80::1704:5a67:abf5:7f81/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
4: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:73:4b:b2:eb brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever
5: bond0: <NO-CARRIER,BROADCAST,MULTICAST,MASTER,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether 8a:9a:03:ce:66:a1 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.150/24 brd 192.168.1.255 scope global bond0
        valid_lft forever preferred_lft forever
shefali@shefali-VirtualBox: $
```

EXPERIMENT 13

AIM: Configure NFS in Linux and share a file through it.

THEORY:

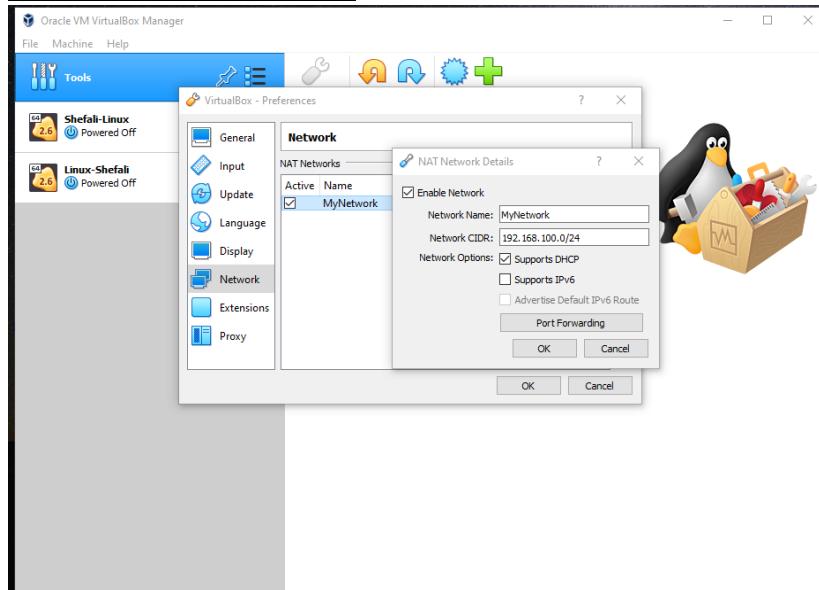
NFS or Network File System is a distributed file system protocol originally developed only by Sun Microsystems. Through NFS, you can allow a system to share directories and files with others over a network. By sharing NFS files, users and even programs can access information on remote systems almost as if they were on a local machine.

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. Setting up an NFS server on your Ubuntu system is very easy. You only need to do some necessary installations and configurations, both on the server and on the client machines, and you are ready to go.

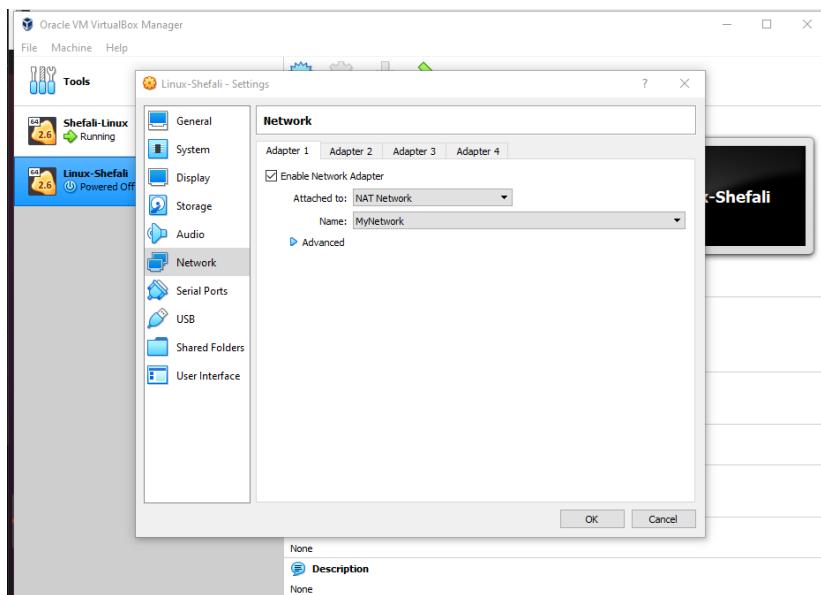
PROCEDURE:

SERVER MACHINE SETUP:

Make a new NATNetwork



Assign the New NATNetwork to Server.



Server IP

```
shefali@shefali-VirtualBox:~$ ifconfig
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 9e:a6:6a:7a:27:9b  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

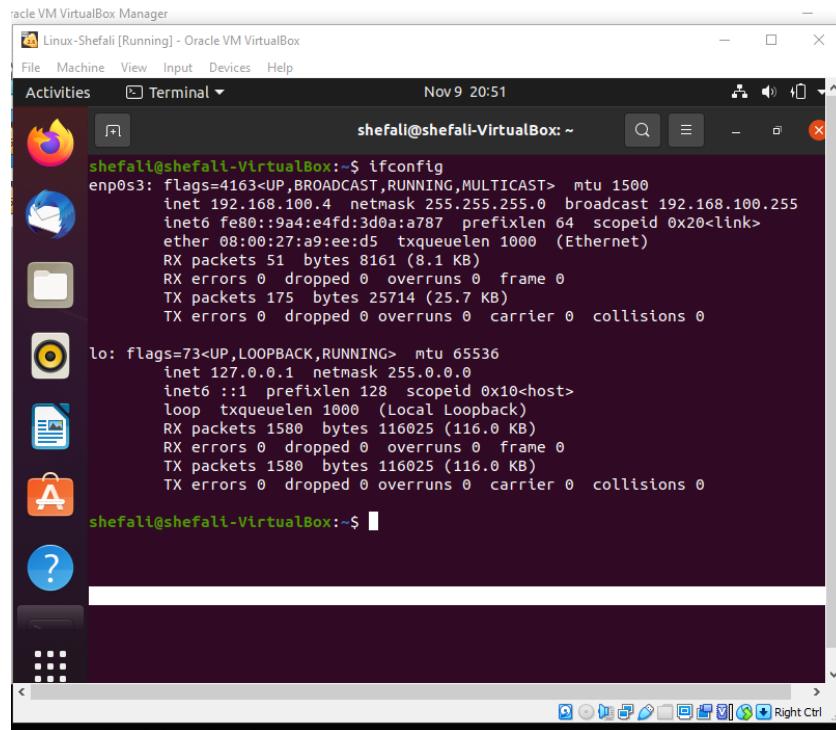
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
                ether 02:42:09:4a:80:14  txqueuelen 0  (Ethernet)
                RX packets 0  bytes 0 (0.0 B)
                RX errors 0  dropped 0  overruns 0  frame 0
                TX packets 0  bytes 0 (0.0 B)
                TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.100.5  netmask 255.255.255.0  broadcast 192.168.100.255
        inet6 fe80::1a63:6643:144c:d6b3  prefixlen 64  scopeid 0x20<link>
                ether 08:00:27:32:64:55  txqueuelen 1000  (Ethernet)
                RX packets 74  bytes 40992 (40.9 KB)
                RX errors 0  dropped 0  overruns 0  frame 0
                TX packets 137  bytes 26832 (26.8 KB)
```

CLIENT MACHINE SETUP:

Assign NATNetwork to client

Client IP:

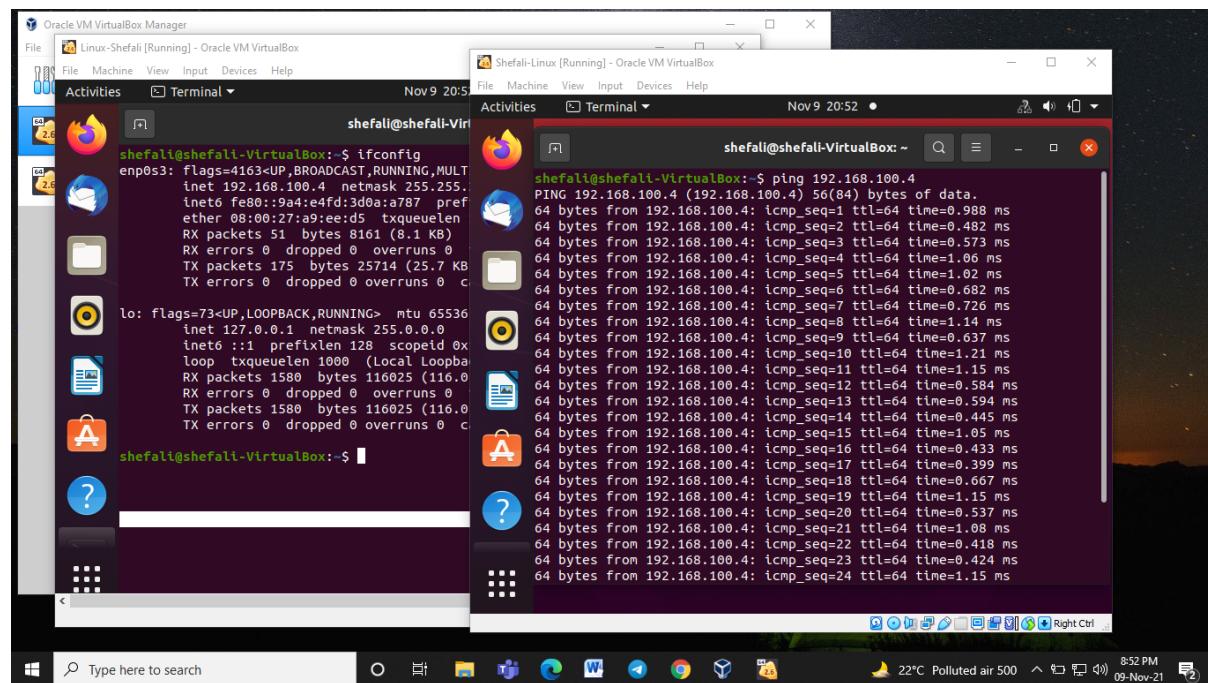


```
shefali@shefali-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.100.4 netmask 255.255.255.0 broadcast 192.168.100.255
                inet6 fe80::9a4:e4fd:3d0a:a787 prefixlen 64 scopeid 0x20<link>
                    ether 08:00:27:a9:ee:d5 txqueuelen 1000 (Ethernet)
                        RX packets 51 bytes 8161 (8.1 KB)
                        RX errors 0 dropped 0 overruns 0 frame 0
                        TX packets 175 bytes 25714 (25.7 KB)
                        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

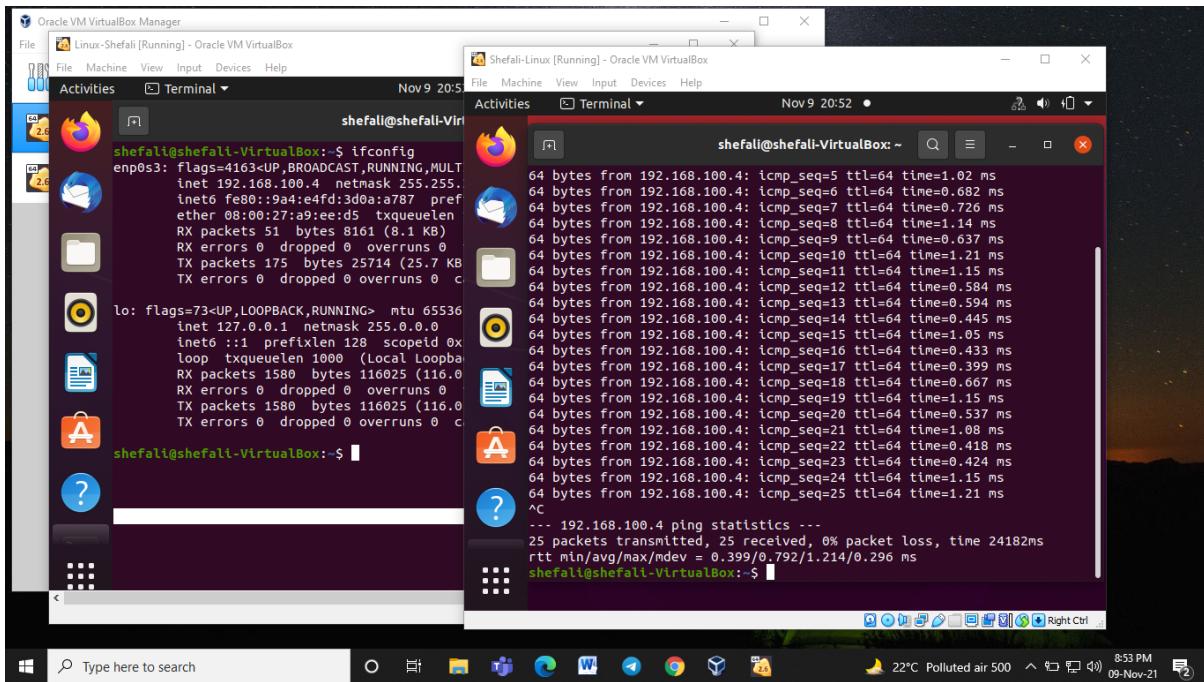
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
                inet6 ::1 prefixlen 128 scopeid 0x10<host>
                    loop txqueuelen 1000 (Local Loopback)
                        RX packets 1580 bytes 116025 (116.0 KB)
                        RX errors 0 dropped 0 overruns 0 frame 0
                        TX packets 1580 bytes 116025 (116.0 KB)
                        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

shefali@shefali-VirtualBox:~$
```

PING FROM SERVER TO CLIENT:



```
shefali@shefali-VirtualBox:~$ ping 192.168.100.4
PING 192.168.100.4 (192.168.100.4) 56(84) bytes of data.
64 bytes from 192.168.100.4: icmp_seq=1 ttl=64 time=0.988 ms
64 bytes from 192.168.100.4: icmp_seq=2 ttl=64 time=0.482 ms
64 bytes from 192.168.100.4: icmp_seq=3 ttl=64 time=0.573 ms
64 bytes from 192.168.100.4: icmp_seq=4 ttl=64 time=1.06 ms
64 bytes from 192.168.100.4: icmp_seq=5 ttl=64 time=1.02 ms
64 bytes from 192.168.100.4: icmp_seq=6 ttl=64 time=0.682 ms
64 bytes from 192.168.100.4: icmp_seq=7 ttl=64 time=0.726 ms
64 bytes from 192.168.100.4: icmp_seq=8 ttl=64 time=1.14 ms
64 bytes from 192.168.100.4: icmp_seq=9 ttl=64 time=0.637 ms
64 bytes from 192.168.100.4: icmp_seq=10 ttl=64 time=1.21 ms
64 bytes from 192.168.100.4: icmp_seq=11 ttl=64 time=1.15 ms
64 bytes from 192.168.100.4: icmp_seq=12 ttl=64 time=0.584 ms
64 bytes from 192.168.100.4: icmp_seq=13 ttl=64 time=0.594 ms
64 bytes from 192.168.100.4: icmp_seq=14 ttl=64 time=0.445 ms
64 bytes from 192.168.100.4: icmp_seq=15 ttl=64 time=1.05 ms
64 bytes from 192.168.100.4: icmp_seq=16 ttl=64 time=0.433 ms
64 bytes from 192.168.100.4: icmp_seq=17 ttl=64 time=0.399 ms
64 bytes from 192.168.100.4: icmp_seq=18 ttl=64 time=0.667 ms
64 bytes from 192.168.100.4: icmp_seq=19 ttl=64 time=1.15 ms
64 bytes from 192.168.100.4: icmp_seq=20 ttl=64 time=0.537 ms
64 bytes from 192.168.100.4: icmp_seq=21 ttl=64 time=1.08 ms
64 bytes from 192.168.100.4: icmp_seq=22 ttl=64 time=0.418 ms
64 bytes from 192.168.100.4: icmp_seq=23 ttl=64 time=0.424 ms
64 bytes from 192.168.100.4: icmp_seq=24 ttl=64 time=1.15 ms
```



PING FROM CLIENT TO SERVER:

shefali@shefali-VirtualBox: ~

```
TX packets 175 bytes 25714 (25.7 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 1580 bytes 116025 (116.0 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 1580 bytes 116025 (116.0 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

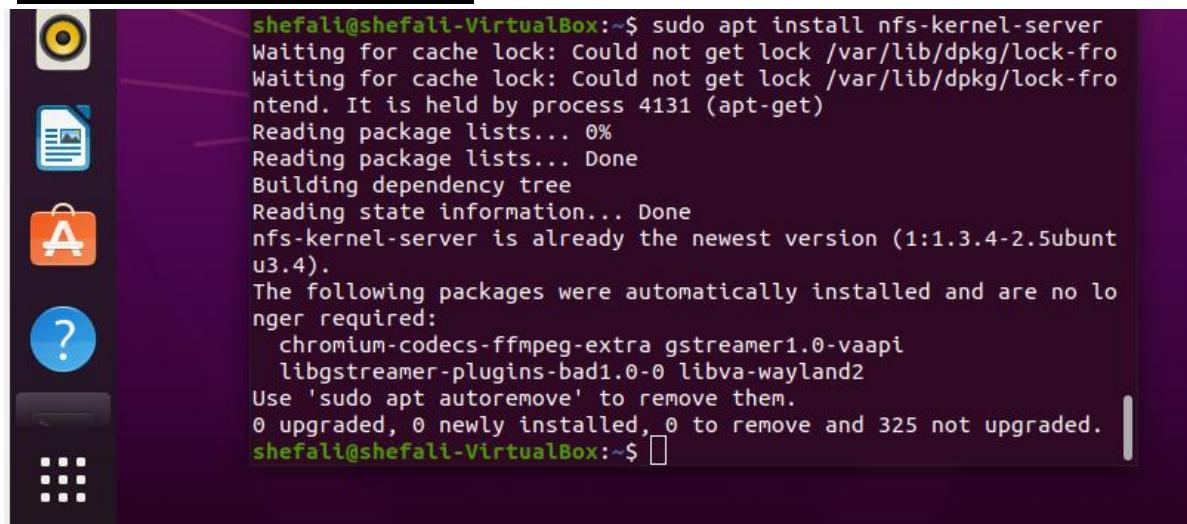
shefali@shefali-VirtualBox: \$ ping 192.168.100.5

```
PING 192.168.100.5 (192.168.100.5) 56(84) bytes of data.
64 bytes from 192.168.100.5: icmp_seq=1 ttl=64 time=0.510 ms
64 bytes from 192.168.100.5: icmp_seq=2 ttl=64 time=0.420 ms
64 bytes from 192.168.100.5: icmp_seq=3 ttl=64 time=0.438 ms
64 bytes from 192.168.100.5: icmp_seq=4 ttl=64 time=0.628 ms
64 bytes from 192.168.100.5: icmp_seq=5 ttl=64 time=0.485 ms
64 bytes from 192.168.100.5: icmp_seq=6 ttl=64 time=0.455 ms
64 bytes from 192.168.100.5: icmp_seq=7 ttl=64 time=0.470 ms
64 bytes from 192.168.100.5: icmp_seq=8 ttl=64 time=0.490 ms
64 bytes from 192.168.100.5: icmp_seq=9 ttl=64 time=0.484 ms
64 bytes from 192.168.100.5: icmp_seq=10 ttl=64 time=0.445 ms
^C
--- 192.168.100.5 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9218ms
rtt min/avg/max/mdev = 0.420/0.482/0.628/0.054 ms
shefali@shefali-VirtualBox: ~
```

Configure NFS(Network File System)

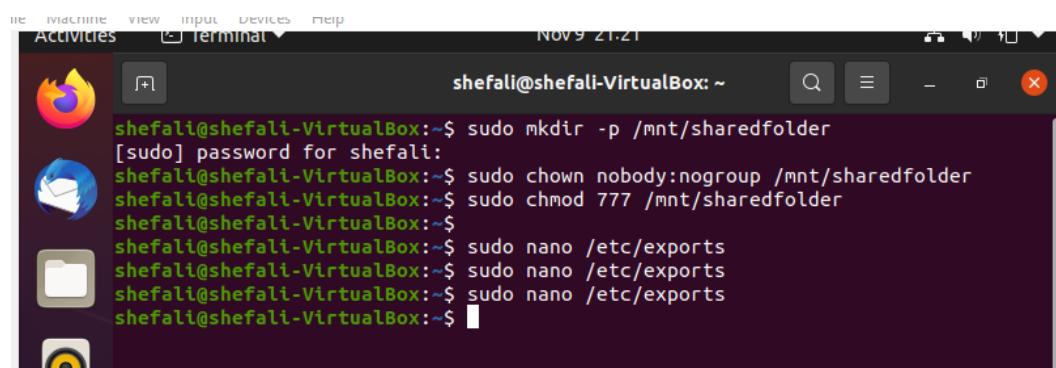
Setting up the host server

Step 1: Install NFS Kernel Server



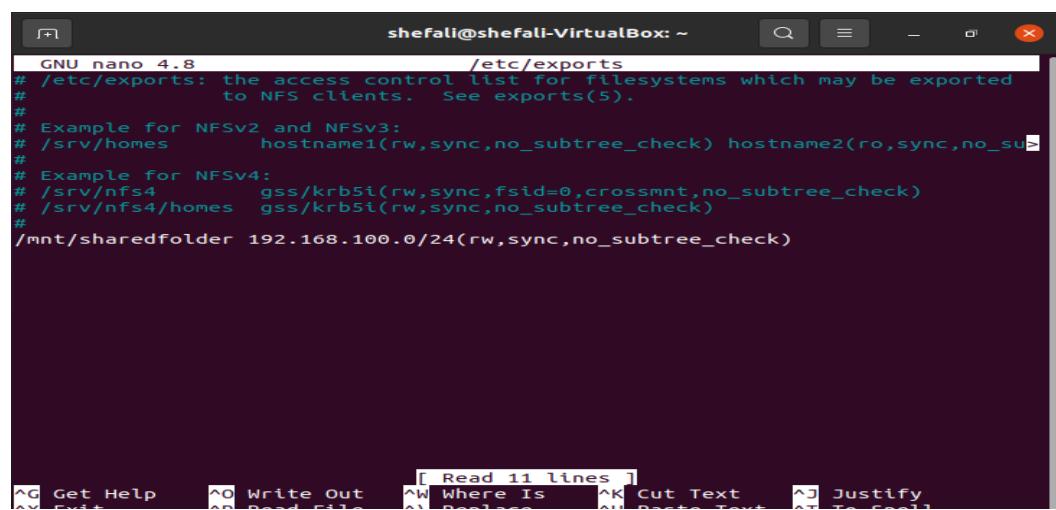
```
shefali@shefali-VirtualBox:~$ sudo apt install nfs-kernel-server
Waiting for cache lock: Could not get lock /var/lib/dpkg/lock-fro
Waiting for cache lock: Could not get lock /var/lib/dpkg/lock-fro
ntend. It is held by process 4131 (apt-get)
Reading package lists... 0%
Reading package lists... Done
Building dependency tree
Reading state information... Done
nfs-kernel-server is already the newest version (1:1.3.4-2.5ubuntu
3.4).
The following packages were automatically installed and are no lo
nger required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
  libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 325 not upgraded.
shefali@shefali-VirtualBox:~$ 
```

Step 2: Create the Export Directory



```
shefali@shefali-VirtualBox:~$ sudo mkdir -p /mnt/sharedfolder
[sudo] password for shefali:
shefali@shefali-VirtualBox:~$ sudo chown nobody:nogroup /mnt/sharedfolder
shefali@shefali-VirtualBox:~$ sudo chmod 777 /mnt/sharedfolder
shefali@shefali-VirtualBox:~$ 
shefali@shefali-VirtualBox:~$ sudo nano /etc/exports
shefali@shefali-VirtualBox:~$ sudo nano /etc/exports
shefali@shefali-VirtualBox:~$ sudo nano /etc/exports
shefali@shefali-VirtualBox:~$ 
```

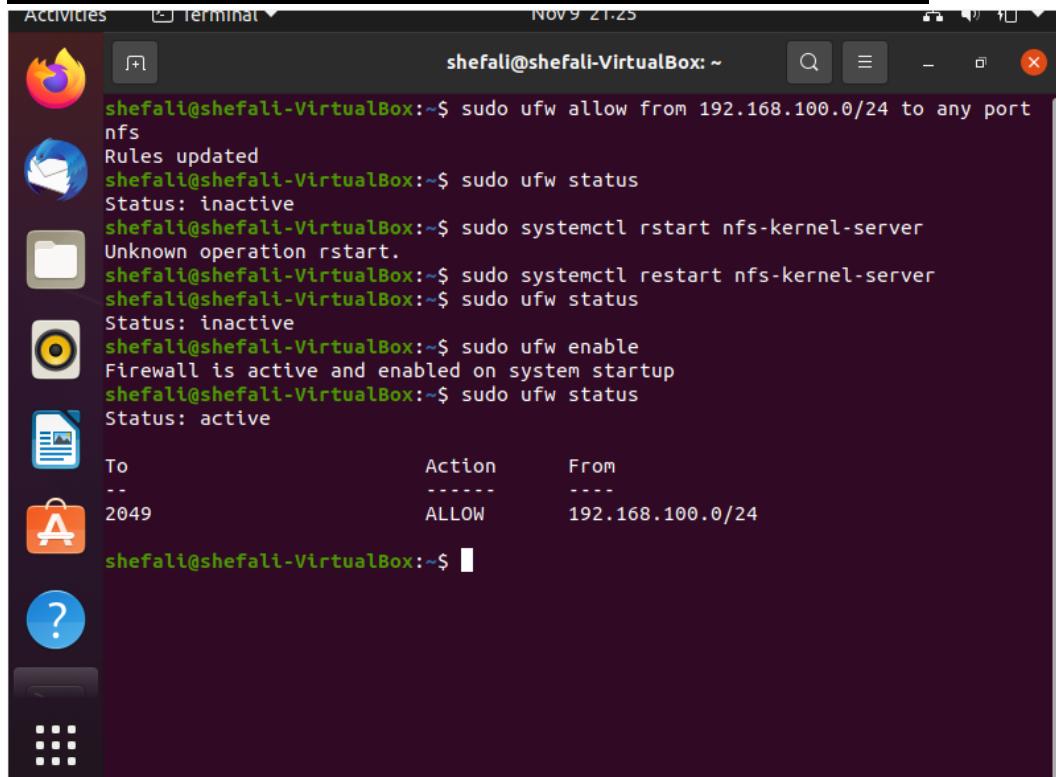
Step 3: Assign server access to clients through NFS



```
shefali@shefali-VirtualBox:~$ 
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/sharedfolder 192.168.100.0/24(rw,sync,no_subtree_check)

^G Get Help   ^O Write Out  [ Read 11 lines ] ^W Where Is  ^K Cut Text  ^J Justify
^X Exit       ^D Read File  ^A Replace   ^L Paste Text  ^T To Spell
```

Step 4: Export the shared directory and open firewall for the client

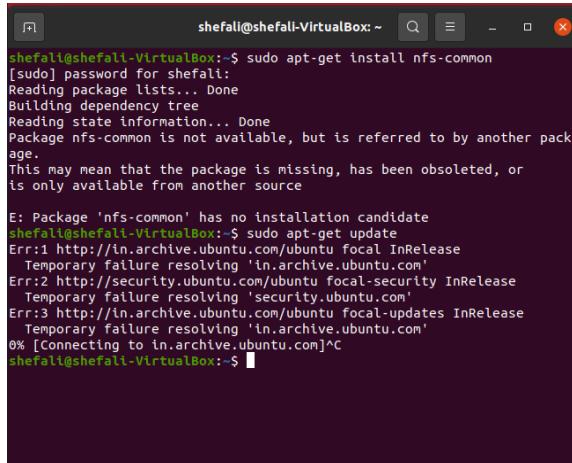


The screenshot shows a terminal window titled "shefali@shefali-VirtualBox:~". The terminal displays the following commands and their outputs:

```
shefali@shefali-VirtualBox:~$ sudo ufw allow from 192.168.100.0/24 to any port  
nfs  
Rules updated  
shefali@shefali-VirtualBox:~$ sudo ufw status  
Status: inactive  
shefali@shefali-VirtualBox:~$ sudo systemctl restart nfs-kernel-server  
Unknown operation restart.  
shefali@shefali-VirtualBox:~$ sudo systemctl restart nfs-kernel-server  
shefali@shefali-VirtualBox:~$ sudo ufw status  
Status: inactive  
shefali@shefali-VirtualBox:~$ sudo ufw enable  
Firewall is active and enabled on system startup  
shefali@shefali-VirtualBox:~$ sudo ufw status  
Status: active  
To                         Action      From  
--                         ----      --  
2049                       ALLOW      192.168.100.0/24  
shefali@shefali-VirtualBox:~$
```

Configuring the Client Machine

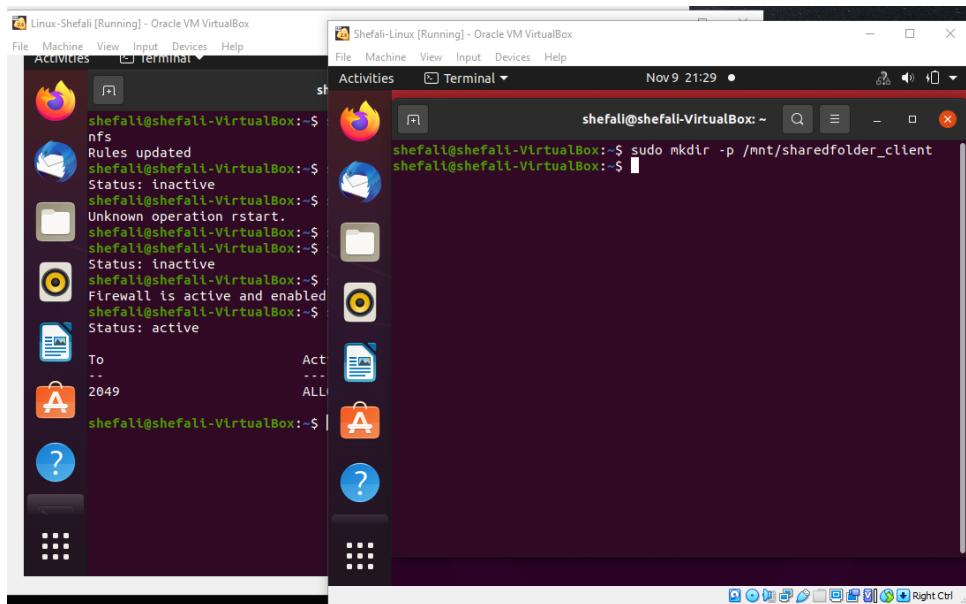
Step 1: Install NFS Common



```
shefali@shefali-VirtualBox:~$ sudo apt-get install nfs-common
[sudo] password for shefali:
Reading package lists... Done
Building dependency tree
Reading state information... Done
Package nfs-common is not available, but is referred to by another pack-
age.
This may mean that the package is missing, has been obsoleted, or
is only available from another source

E: Package 'nfs-common' has no installation candidate
shefali@shefali-VirtualBox:~$ sudo apt-get update
Err:1 http://in.archive.ubuntu.com/ubuntu focal InRelease
  Temporary failure resolving 'in.archive.ubuntu.com'
Err:2 http://security.ubuntu.com/ubuntu focal-security InRelease
  Temporary failure resolving 'security.ubuntu.com'
Err:3 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease
  Temporary failure resolving 'in.archive.ubuntu.com'
0% [Connecting to in.archive.ubuntu.com]^C
shefali@shefali-VirtualBox:~$
```

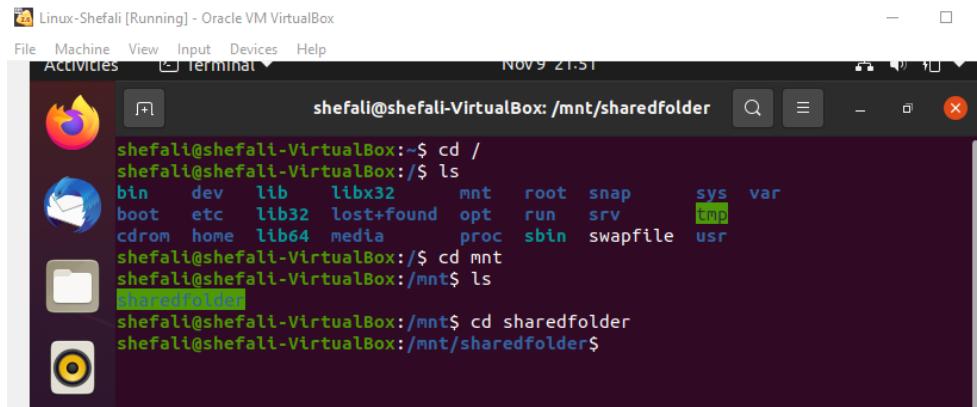
Step 2: Create a mount point for the NFS host's shared folder and mount the shared directory on the client



Performing the transfer

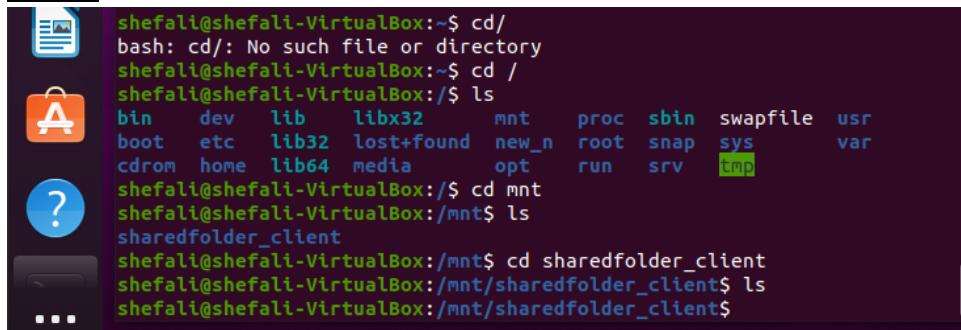
Before Transferring:

Server:



```
shefali@shefali-VirtualBox:~$ cd /
shefali@shefali-VirtualBox:/$ ls
bin dev lib libx32 mnt root snap sys var
boot etc lib32 lost+found opt run srv tmp
cdrom home lib64 media proc sbin swapfile usr
shefali@shefali-VirtualBox:/$ cd mnt
shefali@shefali-VirtualBox:/mnt$ ls
sharedFolder
shefali@shefali-VirtualBox:/mnt$ cd sharedfolder
shefali@shefali-VirtualBox:/mnt/sharedfolder$
```

Client:



```
shefali@shefali-VirtualBox:~$ cd/
bash: cd/: No such file or directory
shefali@shefali-VirtualBox:/$ ls
bin dev lib libx32 mnt proc sbin swapfile usr
boot etc lib32 lost+found new_n root snap sys var
cdrom home lib64 media opt run srv tmp
shefali@shefali-VirtualBox:/$ cd mnt
shefali@shefali-VirtualBox:/mnt$ ls
sharedfolder_client
shefali@shefali-VirtualBox:/mnt$ cd sharedfolder_client
shefali@shefali-VirtualBox:/mnt/sharedfolder_client$ ls
shefali@shefali-VirtualBox:/mnt/sharedfolder_client$
```

After Transferring:

Server:

```
shefali@shefali-VirtualBox:/home$ sudo touch file1.txt
[sudo] password for shefali:
shefali@shefali-VirtualBox:/home$ ls
file1.txt  shefali
shefali@shefali-VirtualBox:/home$ mv file1.txt /mnt/sharedfolder
mv: cannot move 'file1.txt' to '/mnt/sharedfolder/file1.txt': Permission denied
shefali@shefali-VirtualBox:/home$ sudo mv file1.txt /mnt/sharedfolder
shefali@shefali-VirtualBox:/home$ cd mnt
bash: cd: mnt: No such file or directory
shefali@shefali-VirtualBox:/home$ cd ..
shefali@shefali-VirtualBox:$ cd mnt
shefali@shefali-VirtualBox:/mnt$ ls
sharedfolder
shefali@shefali-VirtualBox:/mnt$ cd sharedfolder
shefali@shefali-VirtualBox:/mnt/sharedfolder$ ls
file1.txt
shefali@shefali-VirtualBox:/mnt/sharedfolder$
```

Client:

```
shefali@shefali-VirtualBox:/mnt/sharedfolder_client$ ls
file1.txt
shefali@shefali-VirtualBox:/mnt/sharedfolder_client$
```

RESULT: Configuring NFS (Network File System) in Linux and Transfer Files through the configured Network is done successfully.