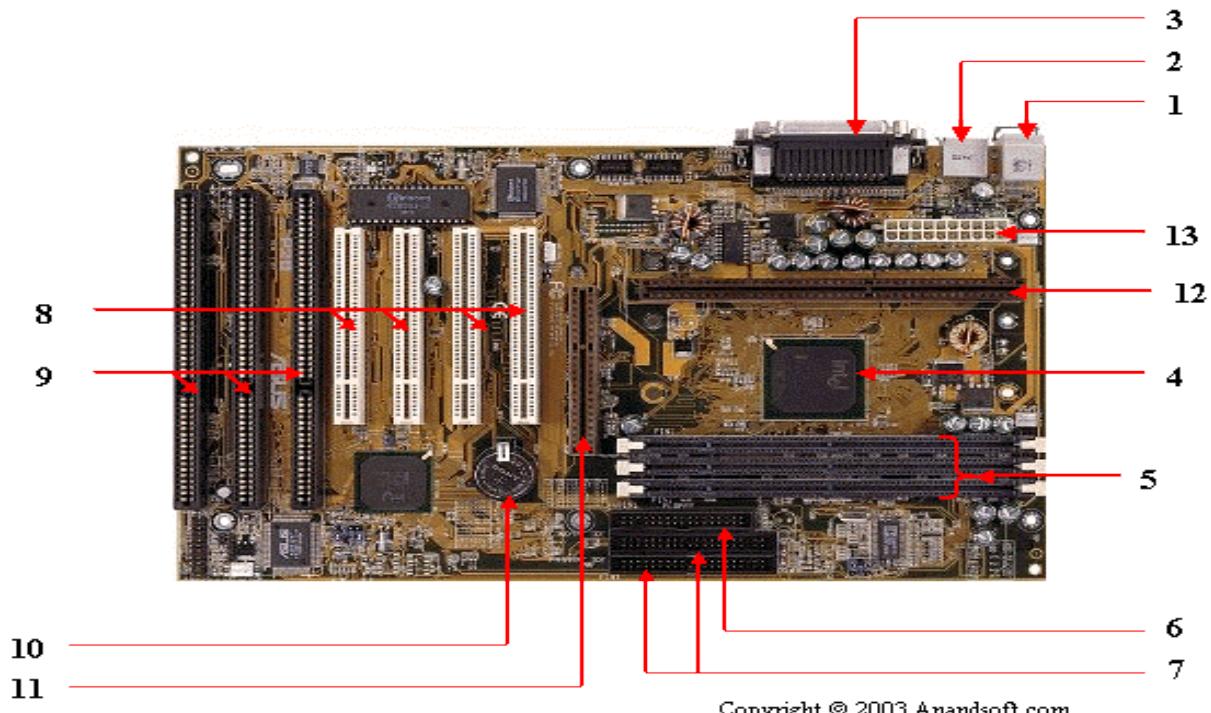


FAMILIARISATION OF COMPUTER HARDWARE**AIM**

Familiarisation of Computer Hardware.



1. Mouse & keyboard
2. USB
3. Parallel port
4. CPU Chip
5. RAM slots
6. Floppy controller
7. IDE controller
8. PCI slot
9. ISA slot
10. CMOS Battery
11. AGP slot
12. CPU slot
13. Power supply plug in

1. Mouse & keyboard: Keyboard Connectors are two types basically. All PCs have a Key board port connected directly to the motherboard. The oldest, but still quite common type, is a special DIN, and most PCs until recently retained this style connector. The AT-style keyboard connector is quickly disappearing, being replaced by the smaller mini DIN PS/2-style keyboard connector.

You can use an AT-style keyboard with a PS/2-style socket (or the other way around) by using a converter. Although the AT connector is unique in PCs, the PS/2-style mini-DIN is also used in more modern PCs for the mouse. Fortunately , most PCs that use the mini-DIN for both the keyboard and mouse clearly mark each mini-DIN socket as to its correct use. Some keyboards have a USB connection, but these are fairly rare compared to the PS/2 connection keyboards.

2. USB (Universal serial bus): USB is the General-purpose connection for PC. You can find USB versions of many different devices, such as mice, keyboards, scanners, cameras, and even printers. a USB connector's distinctive rectangular shape makes it easily recognizable.

USB has a number of features that makes it particularly popular on PCs. First, USB devices are hot swappable. You can insert or remove them without restarting your system.

3. Parallel port: Most printers use a special connector called a parallel port. Parallel port carry data on more than one wire, as opposed to the serial port, which uses only one wire. Parallel ports use a 25-pin female DB connector. Parallel ports are directly supported by the motherboard through a direct connection or through a dangle.

4. CPU Chip : The central processing unit, also called the microprocessor performs all the calculations that take place inside a pc. CPUs come in Variety of shapes and sizes.

Modern CPUs generate a lot of heat and thus require a cooling fan or heat sink. The cooling device (such as a cooling fan) is removable, although some CPU manufacturers sell the CPU with a fan permanently attached.

5. RAM slots: Random-Access Memory (RAM) stores programs and data currently being used by the CPU. RAM is measured in units called bytes. RAM has been packaged in many different ways. The most current package is called a 168-pin DIMM (Dual Inline Memory module).

6. Floppy controller: The floppy drive connects to the computer via a 34-pin ribbon cable, which in turn connects to the motherboard. A floppy controller is one that is used to control the floppy drive.

7. IDE controller: Industry standards define two common types of hard drives: EIDE and SCSI. Majority of the PCs use EIDE drives. SCSI drives show up in high end PCs such as network servers or graphical workstations. The EIDE drive connects to the hard drive via a 2-inch-wide, 40-pin ribbon cable, which in turn connects to the motherboard. IDE controller is responsible for controlling the hard drive

8. PCI slot: Intel introduced the Peripheral component interconnect bus protocol. The PCI bus is used to connect I/O devices (such as NIC or RAID controllers) to the main logic of the computer. PCI bus has replaced the ISA bus.

9. ISA slot: (Industry Standard Architecture) It is the standard architecture of the Expansion bus. Motherboard may contain some slots to connect ISA compatible cards.

10. CMOS Battery: To provide CMOS with the power when the computer is turned off all motherboards comes with a battery. These batteries mount on the motherboard in one of three ways: the obsolete external battery, the most common onboard battery, and built-in battery.

11. AGP slot: If you have a modern motherboard, you will almost certainly notice a single connector that looks like a PCI slot, but is slightly shorter and usually brown. You also probably have a video card inserted into this slot. This is an Advanced Graphics Port (AGP) slot.

12. CPU slot: To install the CPU, just slide it straight down into the slot. Special notches in the slot make it impossible to install them incorrectly. So remember if it does not go easily, it is probably not correct. Be sure to plug in the CPU fan's power.

13. Power supply plug in: The Power supply, as its name implies, provides the necessary electrical power to make the pc operate. the power supply takes standard 110-V AC power and converts into 12-Volt, 5-Volt, and 3.3-Volt DC power.

RESULT

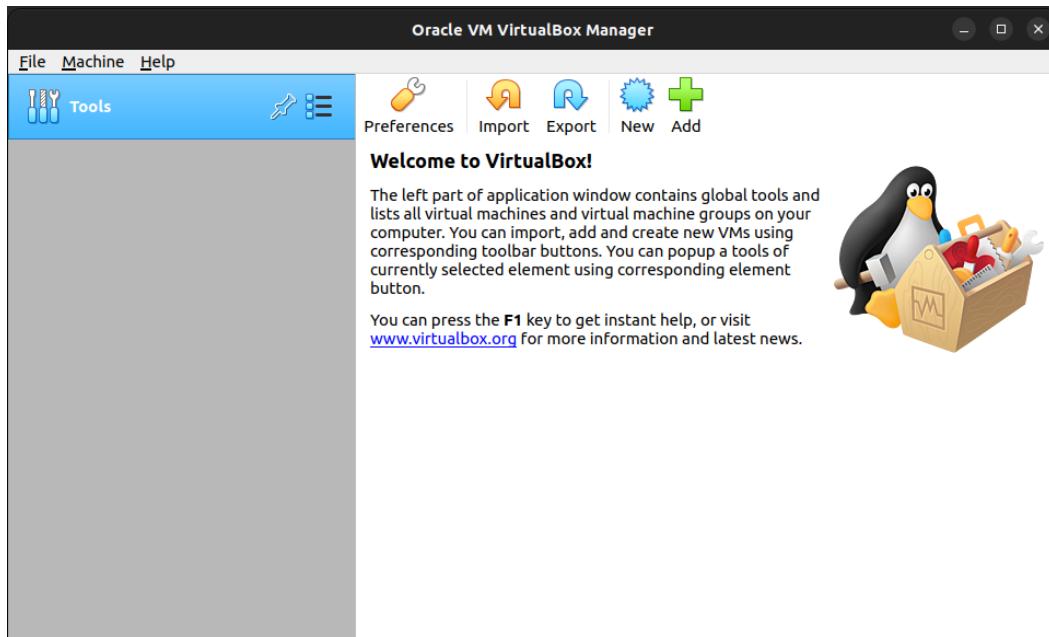
Understood the components of Computer Hard wares

INSTALLATION OF VIRTUAL MACHINE

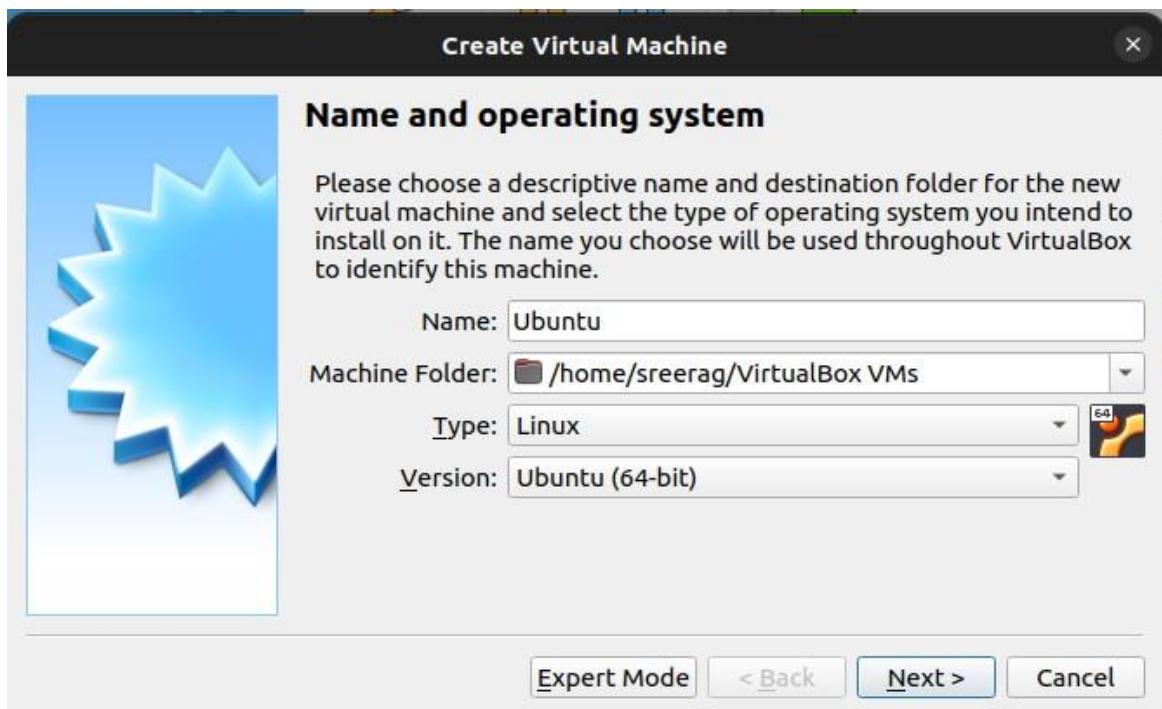
AIM

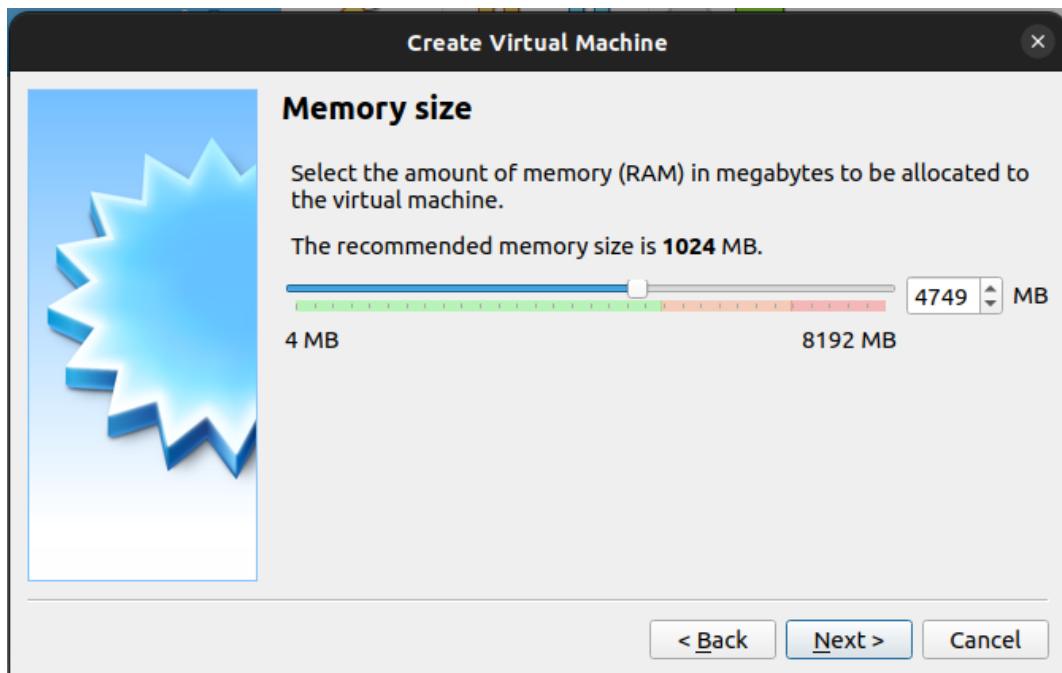
Installation steps of virtual machine in the system

1. Start the virtualBox application



2. Click on new and fill the necessary details and click next

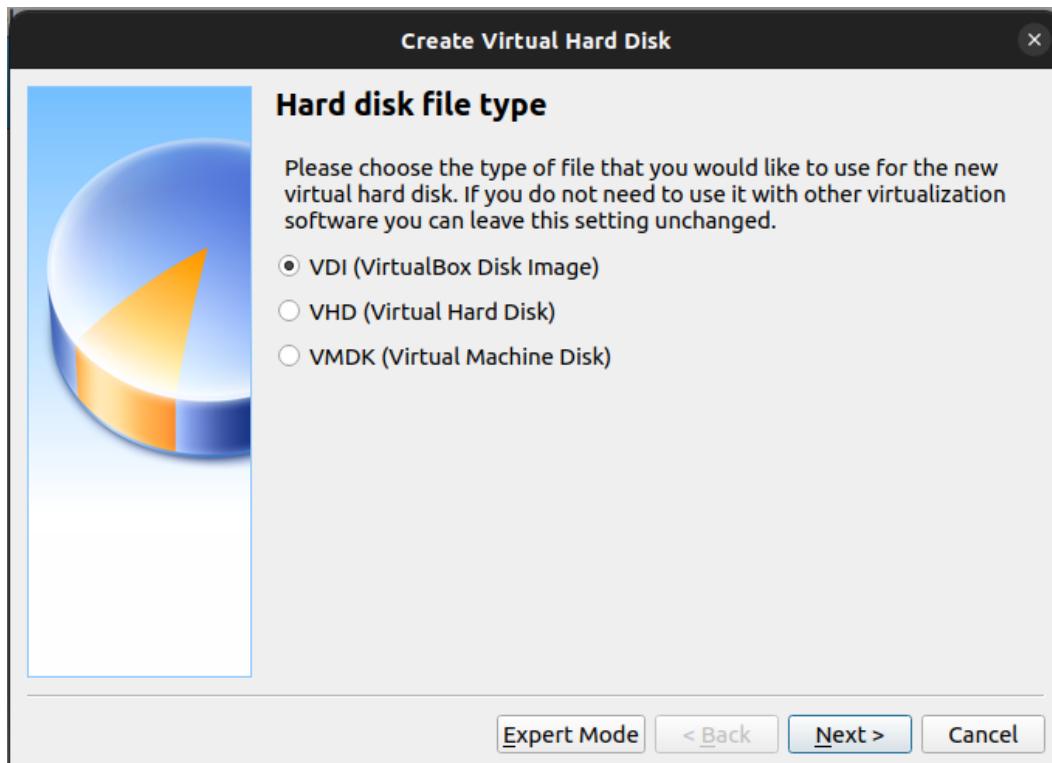




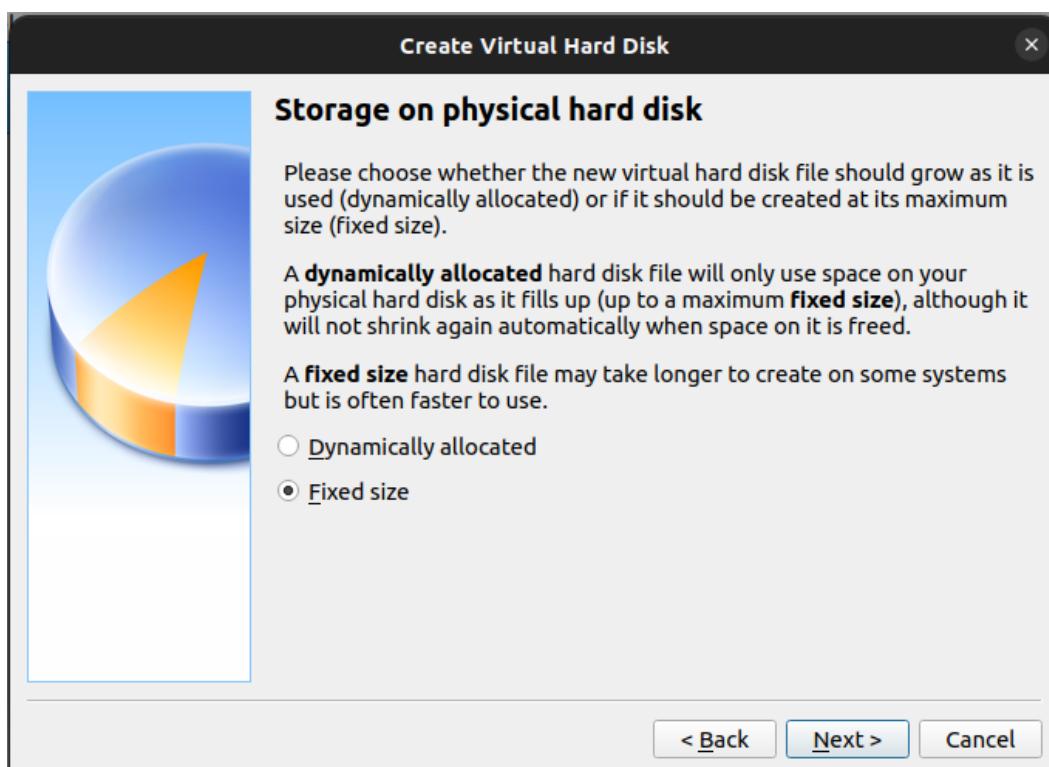
4. Select the option create a virtual hard disk now and click create



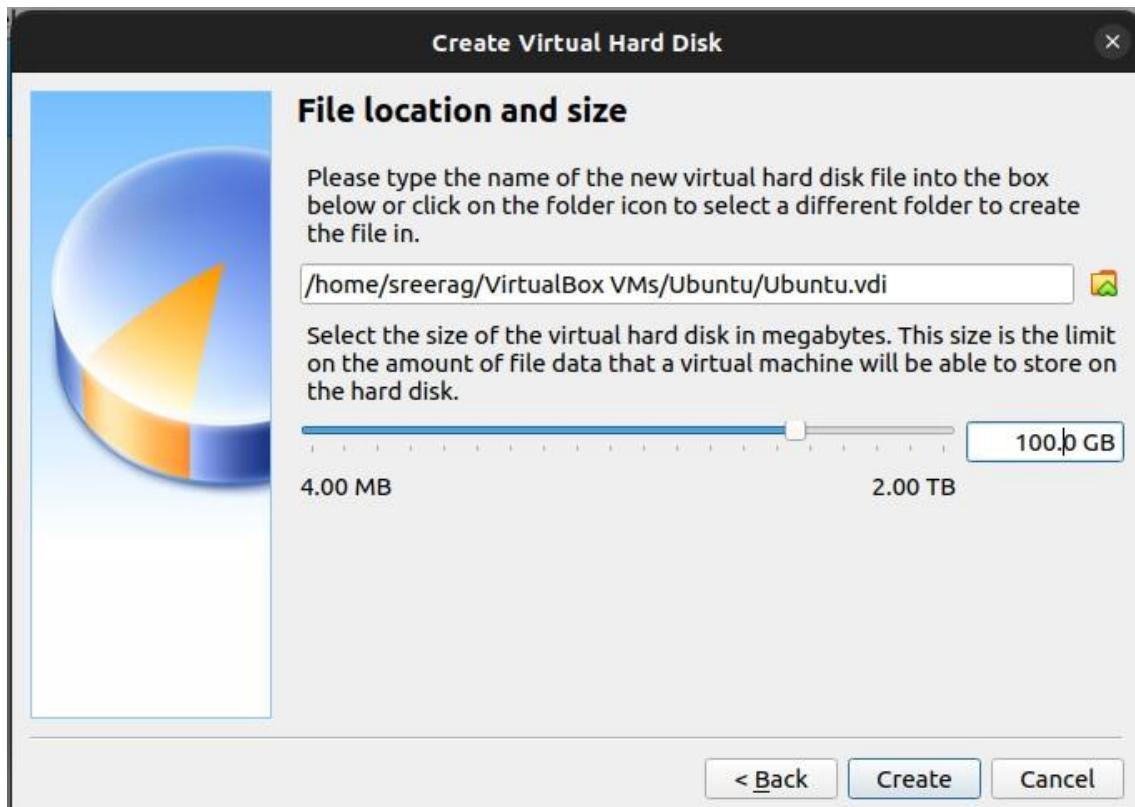
5. Select the option VDI and click next



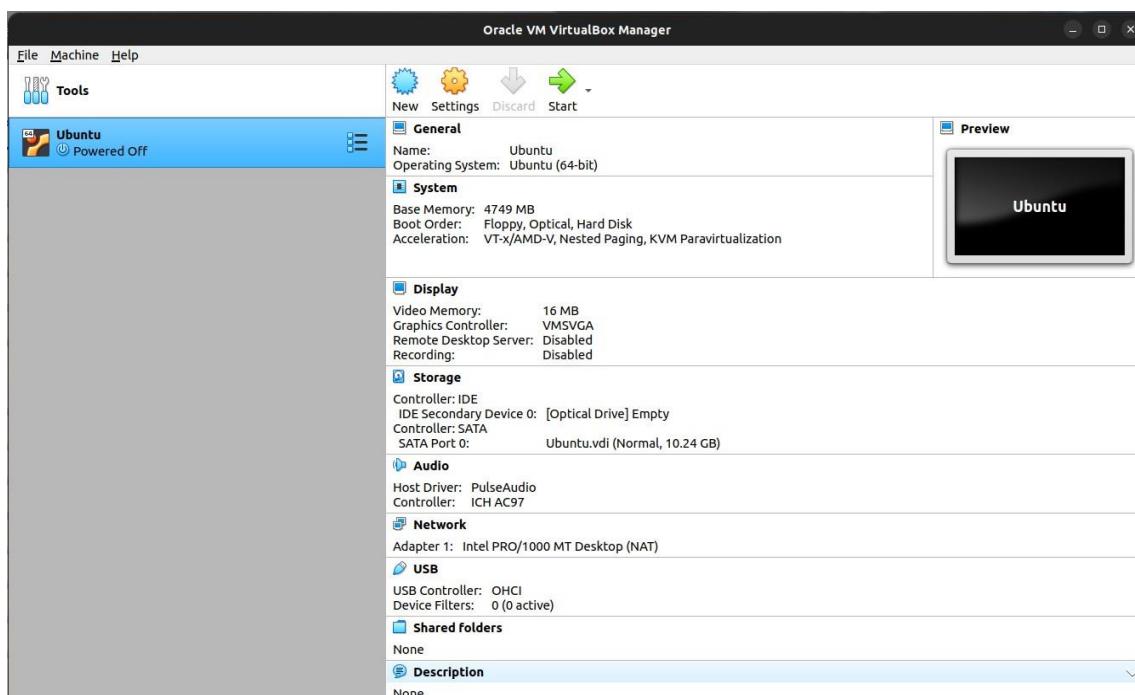
6. Select the option Fixed size and click next

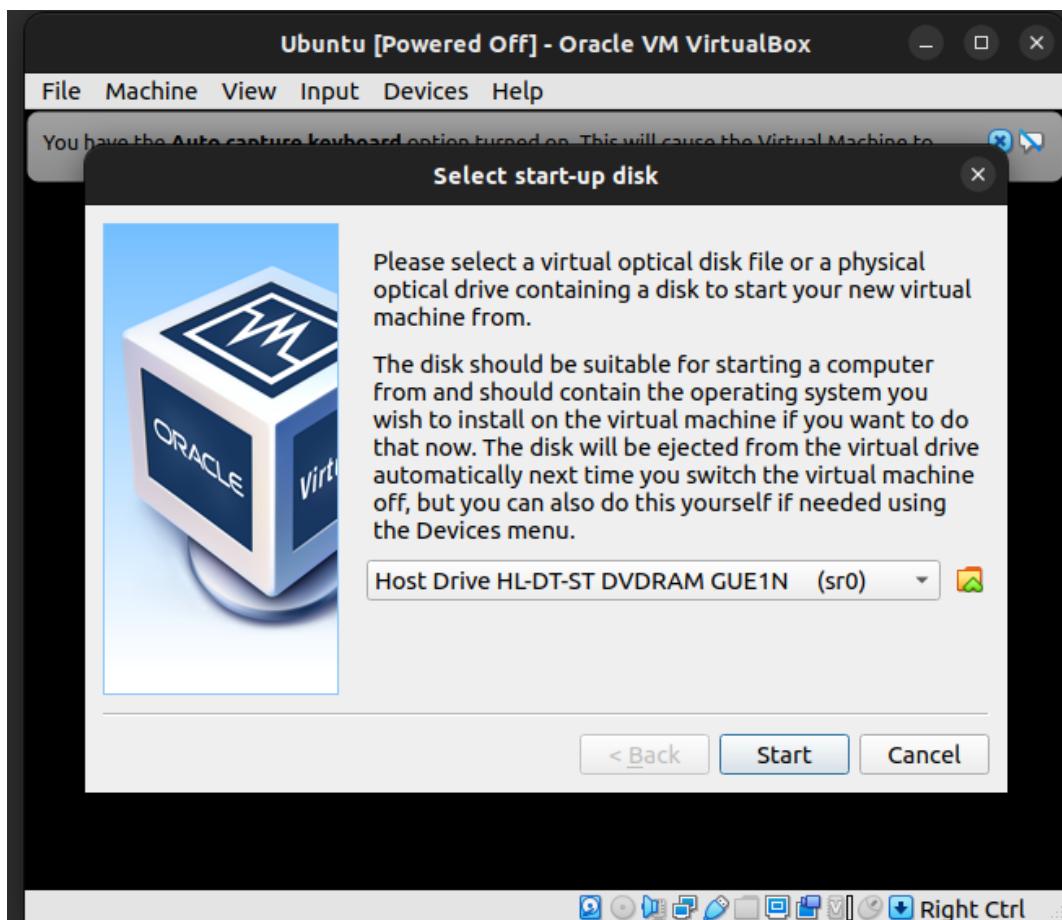


7. Select the memory size as 100 GB for the OS partition and click create

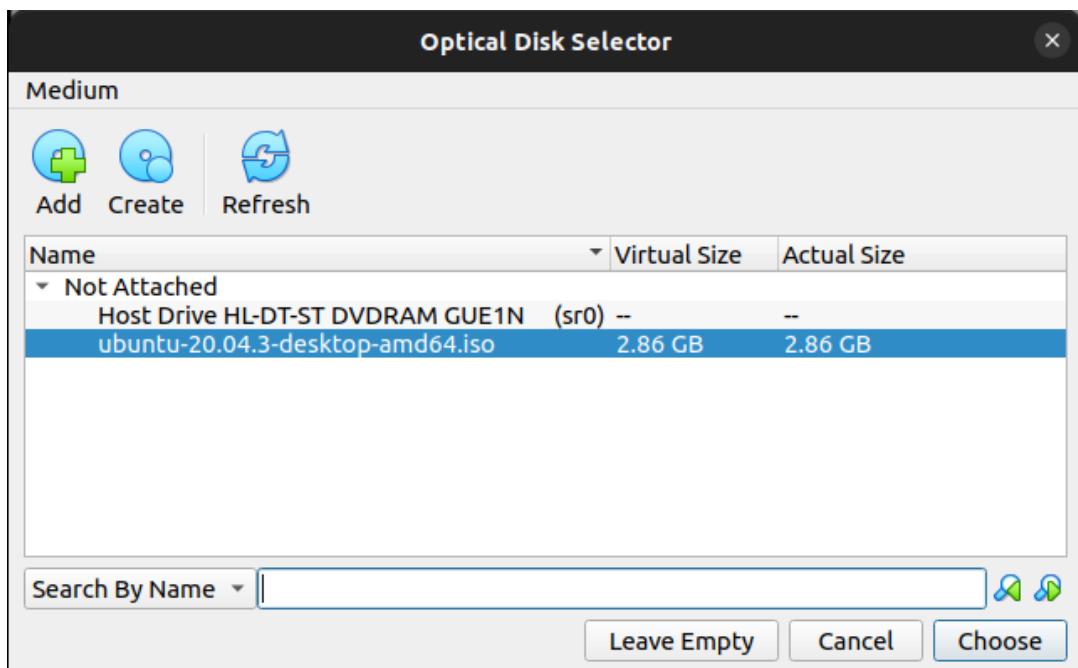


8. Click on start by selecting the OS from left side.

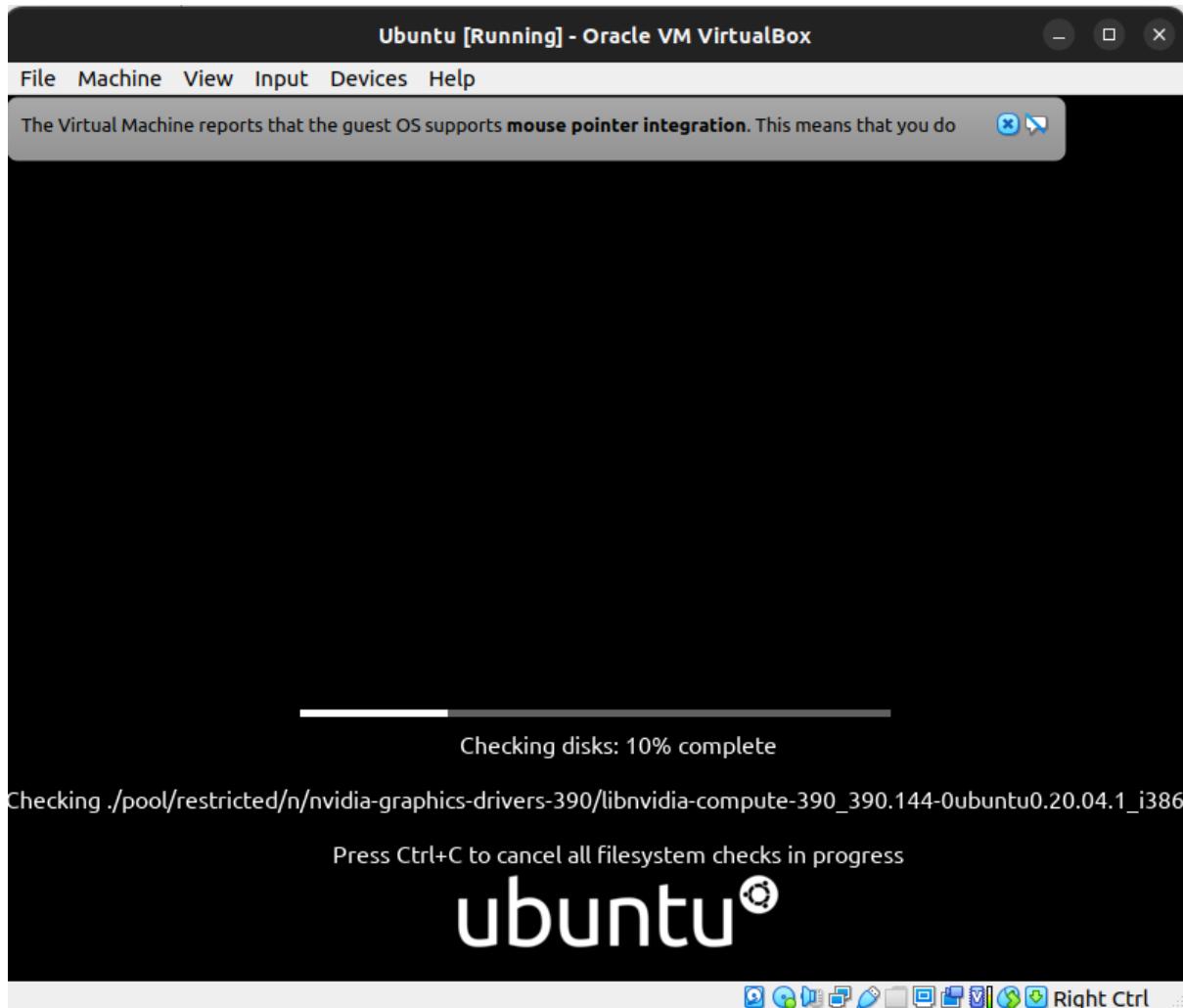




10. Select the ISO file by clicking add. And click on choose



11. Start with installation of selected OS and use it once the installation is done.



RESULT

Understood the steps for installing the virtual box.

LINUX FILE SYSTEM HIERARCHY

AIM

Understand the different Linux File System and its Hierarchy.

The Linux File System Hierarchy Standard (FSH) defines the organization and structure of files and directories in a Linux-based operating system. It provides a consistent and standardized way to organize system files, configuration files, libraries, binaries, and other resources. The following is a brief overview of the main directories in the Linux File System Hierarchy:

/ (root):

The root directory is the top-level directory in the file system hierarchy. It contains all other directories and files.

/bin:

This directory holds essential executable files that are required for the system to boot and run, accessible to all users.

/boot:

The /boot directory contains files necessary for booting the system, such as the Linux kernel, bootloader configuration, and initial RAM disk (initramfs).

/dev:

This directory contains device files that represent hardware devices or pseudo-devices. It allows direct access to various hardware devices.

/etc:

The /etc directory contains system-wide configuration files, such as network settings, user account information, and startup scripts.

/home:

Each user on the system has a dedicated subdirectory under /home, which serves as their home directory. Users can store personal files and configuration here.

/lib and /lib64:

These directories contain libraries that are essential for the system and various programs to run. The /lib directory is used for 32-bit libraries, while /lib64 is for 64-bit libraries.

/media: The /media directory is used as a mount point for removable media devices, such as USB drives or optical discs.

/mnt:

The /mnt directory is a generic mount point for temporarily mounting file systems or other devices.

/opt:

The /opt directory is used for installing optional software or third-party applications. Each application typically has its own subdirectory here.

/proc:

The /proc directory provides a virtual file system that contains information about running processes and system configuration. It allows access to system information through special files.

/root:

The /root directory is the home directory for the system's root user (superuser). It is different from /home/root, which would be a separate directory for a user named "root."

/sbin:

Similar to /bin, the /sbin directory contains essential system binaries. However, the binaries in /sbin are typically meant for system administration tasks and require administrative privileges.

/srv:

The /srv directory is used for storing data files related to services provided by the system. It may include files for web servers, FTP servers, or other network services.

/tmp:

The /tmp directory is used for temporary files created by various programs and users. The files in this directory are usually deleted upon system reboot.

/usr:

The /usr directory contains most of the user-readable, non-system-critical files. It includes subdirectories like /usr/bin (user binaries), /usr/lib (user libraries), /usr/share (shared data), and more.

/var:

The /var directory holds variable files that frequently change during system operation, such as log files, spool files (for printing), and temporary files generated by system processes.

RESULT

Understood the Linux File System Hierarchy.

FAMILIARISATION OF LINUX COMMANDS

AIM

Execute the basic Linux commands.

1. Display following messages on your terminal (including quotes and newline).

“God bless us!

We are starting with basic linux command”

```
shreyas@shreyas-VirtualBox:~$ echo -e "\"God bless us ! \n We are Starting with the basic linux Command\""
"God bless us !
We are Starting with the basic linux Command"
```

2. Read your name from the keyboard and display it.

```
shreyas@shreyas-VirtualBox:~/NSAL$ read -p "Enter your name :" name
Enter your name :Shreyas
shreyas@shreyas-VirtualBox:~/NSAL$ echo "Your name is : $name"
Your name is : Shreyas
```

3. Create the directory structure dir1/dir4 and dir1/dir2/dir3 with a single command and then change directory to dir3.

```
shreyas@shreyas-VirtualBox:~/NSAL$ mkdir -p dir1/dir4 dir1/dir2/dir3
shreyas@shreyas-VirtualBox:~/NSAL$ cd dir1/dir2/dir3
shreyas@shreyas-VirtualBox:~/NSAL/dir1/dir2/dir3$ pwd
/home/shreyas/NSAL/dir1/dir2/dir3
```

4. Create a file file1 using nano.

Display the file

- (i) starting with the first 10 lines and

(ii) starting with the 10th line with provision for

- a) Scrolling Up
- b) Scrolling Up and Down

```
shreyas@shreyas-VirtualBox:~/NSAL$ head -n 10 testfile1.txt
abcde
fghi
jk
lm
no
pq
rs
tu
vw
x
shreyas@shreyas-VirtualBox:~/NSAL$ less +10 testfile1.txt
shreyas@shreyas-VirtualBox:~/NSAL$ less testfile1.txt
fgf
h
s
s
h
h
tr
rthts
x
g
h
sh
ghs
fghfgj
dhgrt
dtyty
etrtgg
dfgrgr
dfgdfh
xvvb
nnny
t
t
file1
```

5. Get the manual page of 'ls' command. Search for the word "alphabetic". Find the next occurrence and then find the previous occurrence.

```
shreyas@shreyas-VirtualBox:~/dir1/dir2/dir3$ man ls | less +/alphabetic
[7]+  Stopped                  man ls | less +/alphabetic
```

```
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

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

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

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

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

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

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

-B, --ignore-backups
```

6. Create 2 files testfile2 and testfile3 using nano.

```
shreyas@shreyas-VirtualBox:~$ nano testfile2
shreyas@shreyas-VirtualBox:~$ nano testfile3
```

(a) Modify the permissions of testfile2 using symbolic mode

i. Add read permission to others

```
shreyas@shreyas-VirtualBox:~$ chmod o+r testfile2
```

ii. revoke write from owner

```
shreyas@shreyas-VirtualBox:~$ chmod u-w testfile2
```

```
[ Error writing testfile2: Permission denied ]
```

iii. set only execute to Group.

```
shreyas@shreyas-VirtualBox:~$ chmod g=x testfile2
```

iv. add write to owner, revoke read from others and set read only to group.

```
shreyas@shreyas-VirtualBox:~$ chmod u+w,o-r,g=r testfile2
```

v. set read and write to all

```
shreyas@shreyas-VirtualBox:~$ chmod u=rw,g=rw,o=rw testfile2
```

(b) Modify the permissions of testfile3 using numeric mode

i. Set read and write to all

```
shreyas@shreyas-VirtualBox:~$ chmod 666 testfile3
```

ii. set read, write and execute to owner, read and execute to group and read only to others

```
shreyas@shreyas-VirtualBox:~$ chmod 754 testfile3
```

(c) Set the permissions of testfile2 the same as that of testfile3

```
shreyas@shreyas-VirtualBox:~$ chmod --reference=testfile3 testfile2
```

(d) Set the permissions of the tree (the directory, its children , grandchildren, etc.) rooted at dir1 (Qn. 3) directory to 664

```
shreyas@shreyas-VirtualBox:~/NSAL$ find dir1 -type f -exec chmod 664 {} \;
```

7. Change the owner and group of the directory tree from dir2 to student.

```
shreyas@shreyas-VirtualBox:~/NSAL/dir1$ sudo chown -R shreyas:shreyas dir2
shreyas@shreyas-VirtualBox:~/NSAL/dir1$ ls -lR dir2
dir2:
total 4
drwxrwxr-x 2 shreyas shreyas 4096 Feb 20 15:32 dir3

dir2/dir3:
total 0
```

8. Display the current directory

```
shreyas@shreyas-VirtualBox:~/NSAL/dir1$ pwd
/home/shreyas/NSAL/dir1
```

9. Listing Files and folders

- (a) List the contents of dir1 (Qn. 3) and all its descendants

```
shreyas@shreyas-VirtualBox:~/NSAL$ ls -R dir1
dir1:
dir2  dir4

dir1/dir2:
dir3  network

dir1/dir2/dir3:
System

dir1/dir4:
administration
```

- (b) List the contents of dir3 (Qn. 3) in

- Alphabetical Order

```
shreyas@shreyas-VirtualBox:~/NSAL$ ls dir1/dir2/dir3
Admin.txt  nsal.txt  Python.py  sum.c  System
```

- Sorted on Time of modification, newest first

```
shreyas@shreyas-VirtualBox:~/NSAL$ ls -lt dir1/dir2/dir3
total 36
drwxrwxr-x 2 shreyas shreyas 4096 Feb 27 15:14 lab
-rw-rw-r-- 1 shreyas shreyas   46 Feb 27 15:11 Admin.txt
-rw-rw-r-- 1 shreyas shreyas   36 Feb 27 15:11 nsal.txt
-rw-rw-r-- 1 shreyas shreyas   18 Feb 27 15:10 sum.c
-rw-rw-r-- 1 shreyas shreyas   15 Feb 27 15:09 Python.py
-rw-rw-r-- 1 shreyas shreyas   38 Feb 27 15:04 System
-rw-rw-r-- 1 shreyas shreyas  486 Feb 20 15:39 testfile1.txt
-rwxrwxr-x 1 shreyas shreyas   64 Feb 20 15:22 program2.sh
-rw-rw-r-- 1 shreyas shreyas   33 Feb 20 14:06 commands.txt
```

- Sorted on Size

```
shreyas@shreyas-VirtualBox:~/NSAL$ ls -ls dir1/dir2/dir3
total 36
drwxrwxr-x 2 shreyas shreyas 4096 Feb 27 15:14 lab
-rw-rw-r-- 1 shreyas shreyas  486 Feb 20 15:39 testfile1.txt
-rwxrwxr-x 1 shreyas shreyas   64 Feb 20 15:22 program2.sh
-rw-rw-r-- 1 shreyas shreyas   46 Feb 27 15:11 Admin.txt
-rw-rw-r-- 1 shreyas shreyas   38 Feb 27 15:04 System
-rw-rw-r-- 1 shreyas shreyas   36 Feb 27 15:11 nsal.txt
-rw-rw-r-- 1 shreyas shreyas   33 Feb 20 14:06 commands.txt
```

iv. Reverse of all above

```
shreyas@shreyas-VirtualBox:~/NSAL$ ls -r dir1/dir2/dir3
testfile1.txt System sum.c Python.py program2.sh nsal.txt lab commands.txt Admin.txt
```

v. Long listing of files Sorted on Size with smallest first and size

```
shreyas@shreyas-VirtualBox:~/NSAL$ ls -ls dir1/dir2/dir3 | tac
-rw-rw-r-- 1 shreyas shreyas 15 Feb 27 15:09 Python.py
-rw-rw-r-- 1 shreyas shreyas 18 Feb 27 15:10 sum.c
-rw-rw-r-- 1 shreyas shreyas 33 Feb 20 14:06 commands.txt
-rw-rw-r-- 1 shreyas shreyas 36 Feb 27 15:11 nsal.txt
-rw-rw-r-- 1 shreyas shreyas 38 Feb 27 15:04 System
-rw-rw-r-- 1 shreyas shreyas 46 Feb 27 15:11 Admin.txt
-rwxrwxr-x 1 shreyas shreyas 64 Feb 20 15:22 program2.sh
-rw-rw-r-- 1 shreyas shreyas 486 Feb 20 15:39 testfile1.txt
drwxrwxr-x 2 shreyas shreyas 4096 Feb 27 15:14 lab
total 36
```

vi. Displayed in human readable form

```
shreyas@shreyas-VirtualBox:~/NSAL$ ls -lh dir1/dir2/dir3
total 36K
-rw-rw-r-- 1 shreyas shreyas 46 Feb 27 15:11 Admin.txt
-rw-rw-r-- 1 shreyas shreyas 33 Feb 20 14:06 commands.txt
drwxrwxr-x 2 shreyas shreyas 4.0K Feb 27 15:14 lab
-rw-rw-r-- 1 shreyas shreyas 36 Feb 27 15:11 nsal.txt
-rwxrwxr-x 1 shreyas shreyas 64 Feb 20 15:22 program2.sh
-rw-rw-r-- 1 shreyas shreyas 15 Feb 27 15:09 Python.py
-rw-rw-r-- 1 shreyas shreyas 18 Feb 27 15:10 sum.c
-rw-rw-r-- 1 shreyas shreyas 38 Feb 27 15:04 System
```

10. (a) Execute ls and store the output to a file lsoutput

```
shreyas@shreyas-VirtualBox:~/NSAL$ ls > lsoutput
shreyas@shreyas-VirtualBox:~/NSAL$ cat lsoutput
commands.txt
dir1
lsoutput
program2.sh
student
testfile1.txt
```

(b) Execute ls -l and add the output to lsoutput, at the end.

```
shreyas@shreyas-VirtualBox:~/NSAL$ ls -l >> lsoutput
shreyas@shreyas-VirtualBox:~/NSAL$ cat lsoutput
commands.txt
dir1
lsoutput
program2.sh
student
testfile1.txt
total 24
-rw-rw-r-- 1 shreyas shreyas 33 Feb 20 14:06 commands.txt
drwxrwxr-x 4 shreyas shreyas 4096 Feb 20 15:32 dir1
-rw-rw-r-- 1 shreyas shreyas 61 Feb 27 15:34 lsoutput
-rwxrwxr-x 1 shreyas shreyas 64 Feb 20 15:22 program2.sh
```

11. Execute ls -l and feed the result to less command, to scroll through the directory listing.

```
shreyas@shreyas-VirtualBox:~/NSAL$ ls -l | less
total 24
-rw-rw-r-- 1 shreyas shreyas 33 Feb 20 14:06 commands.txt
drwxrwxr-x 4 shreyas shreyas 4096 Feb 20 15:32 dir1
-rw-rw-r-- 1 shreyas shreyas 413 Feb 27 15:36 lsoutput
-rwxrwxr-x 1 shreyas shreyas 64 Feb 20 15:22 program2.sh
drwxrwxr-x 2 shreyas shreyas 4096 Feb 27 14:23 student
-rw-rw-r-- 1 shreyas shreyas 486 Feb 20 15:39 testfile1.txt
(END)
```

12. (a) Create a file file1 containing the word "Hello," using cat and output redirection

```
shreyas@shreyas-VirtualBox:~/NSAL$ cat > file1
Hello !
shreyas@shreyas-VirtualBox:~/NSAL$ cat file1
Hello !
```

(b) Create another file file2 containing the word ", Greetings!!"

```
shreyas@shreyas-VirtualBox:~/NSAL$ cat > file2
, Greetings !!
shreyas@shreyas-VirtualBox:~/NSAL$ cat file2
, Greetings !!
```

(c) Display the sentence,

Hello,

Your name, Greetings!!

using cat, by concatenating file1, Standard Input and file2

```
shreyas@shreyas-VirtualBox:~/NSAL$ cat file1 - file2
Hello !
Shreyas Shreyas , Greetings !!
```

13.. Copy the file file1 to newfile.

(a) If newfile already exists, it should be replaced.

```
shreyas@shreyas-VirtualBox:~/NSAL$ cp file1 newfile
shreyas@shreyas-VirtualBox:~/NSAL$ cat newfile
Hello !
```

(b) If newfile already exists, it should not be replaced.

```
shreyas@shreyas-VirtualBox:~/NSAL$ cp -n file1 newfile
```

(c) If newfile already exists, it should be replaced, but only with the consent of the user.

```
shreyas@shreyas-VirtualBox:~/NSAL$ cp -i file1 newfile
cp: overwrite 'newfile'? y
```

(d) If newfile already exists, it should be replaced only if its contents is older than that of newfile.

```
shreyas@shreyas-VirtualBox:~/NSAL$ cp -u file1 newfile
```

(e) Even if newfile is read only.

```
shreyas@shreyas-VirtualBox:~/NSAL$ cp -f file1 newfile
```

(f) Create a link instead of copying.

```
shreyas@shreyas-VirtualBox:~/NSAL$ ln file1 newfile
ln: failed to create hard link 'newfile': File exists
shreyas@shreyas-VirtualBox:~/NSAL$ ln -s file1 newfile
ln: failed to create symbolic link 'newfile': File exists
```

(g) Copy the entire directory tree from dir1 of Cycle 1 to a new directory dir5

```
shreyas@shreyas-VirtualBox:~/NSAL$ cp -r dir1 dir5
```

14. Create a new directory, dir6 inside dir1

```
shreyas@shreyas-VirtualBox:~/NSAL$ mkdir -p dir1/dir6
```

(a) Move all files in dir5 into it.

```
shreyas@shreyas-VirtualBox:~/NSAL$ mv dir5/* dir1/dir6/
```

(b) Rename the file newfile in Qn.4 to oldfile

```
shreyas@shreyas-VirtualBox:~/NSAL$ mv newfile oldfile
```

(c) Move the file file1 in Qn.4 to dir6 with the name file3

```
shreyas@shreyas-VirtualBox:~/NSAL$ mv file1 dir1/dir6/file3
```

(d) Delete all files where name starts with a vowel character, upper or lower case.

```
shreyas@shreyas-VirtualBox:~/NSAL$ rm -f [AaEeIiOoUu]*
```

(e) Delete all files where the name is at least 3 characters long.

```
shreyas@shreyas-VirtualBox:~/NSAL$ rm -f ???*
```

(f) Delete all hidden folders, and files.

```
shreyas@shreyas-VirtualBox:~/NSAL$ rm -rf .[^.]*
```

15. Using cut filter

(a) Display the filenames from ls -l assuming filenames start at column 50.

```
shreyas@shreyas-VirtualBox:~/NSAL$ ls -l | cut -c50-  
r1  
r5  
r7  
1  
dfile
```

(b) Display user Id and user name of all users from /etc/passwd. (fields 1 and 3)

```
shreyas@shreyas-VirtualBox:~/NSAL$ cut -d: -f1,3 /etc/passwd  
root:0  
daemon:1  
bin:2  
sys:3  
sync:4  
games:5  
man:6  
lp:7  
mail:8  
news:9  
uucp:10  
proxy:13  
www-data:33  
backup:34  
list:38  
irc:39
```

16. Create 3 files containing name, age and marks of 5 students respectively and paste them into a single csv (comma separated values) file.

```
shreyas@shreyas-VirtualBox:~/NSAL$ echo -e "Alice\nBob\nCharlie\nDavid\nEve" > names.txt
shreyas@shreyas-VirtualBox:~/NSAL$ echo -e "22\n23\n21\n20\n22" > ages.txt
shreyas@shreyas-VirtualBox:~/NSAL$ echo -e "85\n90\n78\n88\n95" > marks.txt
shreyas@shreyas-VirtualBox:~/NSAL$ paste -d ',' names.txt ages.txt marks.txt > students.csv
shreyas@shreyas-VirtualBox:~/NSAL$ cat students.csv
Alice,22,85
Bob,23,90
Charlie,21,78
David,20,88
Eve,22,95
```

17. Using find

- (a) piped with wc, display the number of files in a directory that starts with the letter a

```
shreyas@shreyas-VirtualBox:~/NSAL$ find . -type f -name "a*" | wc -l
3
```

- (b) Delete all .c files in the parent directory

```
shreyas@shreyas-VirtualBox:~/NSAL$ find .. -maxdepth 1 -type f -name "*.c" -delete
```

- (c) Copy all files that starts with a to dir2

```
shreyas@shreyas-VirtualBox:~/NSAL$ find . -type f -name "a*" -exec cp {} dir2/ \;
```

- (d) Display files in the current directory that were modified in the last 30 minutes.

```
shreyas@shreyas-VirtualBox:~/NSAL$ find . -type f -mmin -30
./hl.c
./students.csv
./dir2/abc.txt
./dir2/abcd.txt
./dir2/ages.txt
./names.txt
./marks.txt
./abc.txt
./ages.txt
```

18. Use head and tail piped with cat /etc/passwd to display the details of

(a) The first 12 users in the system.

```
shreyas@shreyas-VirtualBox:~/NSAL$ head -n 12 /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
```

(b) The last 7 users in the system.

```
shreyas@shreyas-VirtualBox:~/NSAL$ tail -n 7 /etc/passwd
saned:x:120:129::/var/lib/saned:/usr/sbin/nologin
colord:x:121:130:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
pulse:x:122:131:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
speech-dispatcher:x:123:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
nm-openvpn:x:124:133:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
hplip:x:125:7:HPLIP system user,,,:/run/hplip:/bin/false
shreyas:x:1000:1000:Shreyas,,,:/home/shreyas:/bin/bash
```

(c) All but the first 3.

```
shreyas@shreyas-VirtualBox:~/NSAL$ tail -n +4 /etc/passwd
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
```

(d) All but the last 5.

```
shreyas@shreyas-VirtualBox:~/NSAL$ head -n -5 /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
```

(e) Only the 9 th .

```
shreyas@shreyas-VirtualBox:~/NSAL$ head -n 9 /etc/passwd | tail -n 1
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
```

19. Use grep to

(a) Display all lines in a file that contains the string abc

```
shreyas@shreyas-VirtualBox:~/NSAL$ grep "abc" alph.txt
abc
abcdefg
```

(b) Display all lines in a file that does not contain the string abc

```
shreyas@shreyas-VirtualBox:~/NSAL$ grep -v "abc" sample.txt
Another line without it.
Something else.
```

(c) List names of all .c files that contains a printf

```
shreyas@shreyas-VirtualBox:~/NSAL$ grep -l "printf" *.c
hello.c
multi_printf.c
test.c
```

(d) List names of all .c files that does not contain a printf

```
shreyas@shreyas-VirtualBox:~/NSAL$ grep -L "printf" *.c
nl.c
no_printf.c
```

(e) Display the number of #include statements in each .c file.

```
shreyas@shreyas-VirtualBox:~/NSAL$ grep -c "#include" *.c
hello.c:1
hl.c:1
multi_printf.c:1
no_printf.c:1
test.c:1
```

(f) Display the Line numbers of printf in a .c file.

```
shreyas@shreyas-VirtualBox:~/NSAL$ grep -n "printf" test.c
2:void test() { printf("Testing printf!\n"); }
```

(g) List names of all files in the directory tree that contain a printf.

```
shreyas@shreyas-VirtualBox:~/NSAL$ grep -rl "printf" .
./multi_printf.c
./hello.c
./test.c
```

(h) Display the context of every printf in a .c file. i.e., n lines before and after every printf.

```
jithu@jithu-VirtualBox:~/dir1/dir2/dir3$ grep -C 1 'printf' program1.c
void main(){
    printf("Hello World");
    printf();
//dsahdihs
    printf();_
}
```

(i) ls -l starts with d for directories. Use ls -l piped with grep & cut to display the names of all directories in the current directory.

```
shreyas@shreyas-VirtualBox:~/NSAL$ ls -l | grep "^\d" | awk '{print $9}'  
dir1  
dir2  
dir5  
dir7  
dr1  
oldfile
```

20. Using expr

(a) Read two integers X and Y . Display the sum, difference, product, quotient and remainder of these variables.

```
shreyas@shreyas-VirtualBox:~/NSAL$ read X Y  
12 4  
shreyas@shreyas-VirtualBox:~/NSAL$ echo "Sum: $(expr $x + $y)"  
Sum: 16  
shreyas@shreyas-VirtualBox:~/NSAL$ echo "Difference: $(expr $X - $Y)"  
Difference: 8  
shreyas@shreyas-VirtualBox:~/NSAL$ echo "Product: $(expr $X \* $Y)"  
Product: 48  
shreyas@shreyas-VirtualBox:~/NSAL$ echo "Quotient: $(expr $X / $Y)"  
Quotient: 3  
shreyas@shreyas-VirtualBox:~/NSAL$ echo "Remainder: $(expr $X % $Y)"  
Remainder: 0
```

(b) Read a string, S, a position, p and a length l. Display the substring of length l starting at position p from the string S

```
shreyas@shreyas-VirtualBox:~/NSAL$ read string  
welcome to NSAL  
shreyas@shreyas-VirtualBox:~/NSAL$ read pos len  
3 8  
shreyas@shreyas-VirtualBox:~/NSAL$ echo $string | cut -c ${pos}-$((pos + $len -1))  
lcome to
```

RESULT

Basic Linux commands has been executed and the output is verified.

SHELL SCRIPTING**AIM**

Execute the basic shell scripting programs.

1. Write a shell program to read a string and display it.

```
echo "Enter a string :"  
read str  
echo "You entered: $str "
```

OUTPUT

```
Enter a string:  
Hello  
You entered: Hello
```

2. Write a shell program to read 2 numbers and find sum, difference, product, quotient.

```
echo "Enter two numbers:"  
read a b  
sum=$((a + b))  
diff=$((a - b))  
prod=$((a * b))  
if [ $b -ne 0 ]; then  
    quot=$((a / b))  
else  
    quot="undefined (division by zero)"  
fi  
echo "Sum: $sum"  
echo "Difference: $diff"  
echo "Product: $prod"  
echo "Quotient: $quot"
```

OUTPUT

```
Enter two numbers:  
10 5  
Sum: 15  
Difference: 5  
Product: 50  
Quotient: 2
```

3. Write a shell program to check if a number is odd or even.

```
echo "Enter a number:"  
read num  
if [ $((num % 2)) -eq 0 ]; then  
    echo "$num is even"  
else  
    echo "$num is odd"  
fi
```

OUTPUT

```
Enter a number:  
7  
7 is odd
```

4. Write a shell program to read 3 numbers and find the largest of them.

```
echo "Enter three numbers:"  
read a b c  
if [ $a -ge $b ] && [ $a -ge $c ]; then  
    echo "Largest number is: $a"  
elif [ $b -ge $c ]; then  
    echo "Largest number is: $b"  
else  
    echo "Largest number is: $c"  
fi
```

OUTPUT

```
Enter three numbers:  
5 7 3  
Largest number is: 7
```

5. Read 3 marks of a student and find the average. Display the grade of the student based on the average.

- S >= 90%
- A < 90%, but >= 80%
- B < 80%, but >= 60%
- P < 80%, but >= 40%
- F < 40%

```
echo "Enter three marks:"
```

```
read m1 m2 m3
```

```
sum=$((m1 + m2 + m3))
```

```
avg=$((sum / 3))
```

```
echo "Average: $avg"
```

```
if [ $avg -ge 90 ]; then
```

```
    echo "Grade: S"
```

```
elif [ $avg -ge 80 ]; then
```

```
    echo "Grade: A"
```

```
elif [ $avg -ge 60 ]; then
```

```
    echo "Grade: B"
```

```
elif [ $avg -ge 40 ]; then
```

```
    echo "Grade: P"
```

```
else
```

```
    echo "Grade: F"
```

```
fi
```

OUTPUT

```
Enter three marks:  
78 86 88  
Average: 84  
Grade: A
```

6. Write a shell program to read a filename as command line argument and check whether it exists or not.

```
if [ -e "$1" ]; then
    echo "File '$1' exists."
else
    echo "File '$1' does not exist."
fi
```

OUTPUT

```
$ ./check_file.sh test.txt
File 'test.txt' exists.
```

7. Write a shell program to display the multiplication table of a number n?

```
echo "Enter a number:"
read num
for i in {1..10}; do
    echo "$num x $i = $((num * i))"
done
```

OUTPUT

```
Enter a number:
5
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
```

8. Write a shell program to display the contents in your current folder.

```
echo "Contents of the current folder:"
ls -l
```

OUTPUT

```
~/NSAL $ ./contents.sh
Contents of the current folder:
total 36
-rwx----- . 1 u0_a360 u0_a360 247 Mar 15 12:11 argumentFile.sh
-rwx----- . 1 u0_a360 u0_a360 85 Mar 15 12:20 contents.sh
-rw----- . 1 u0_a360 u0_a360 36 Mar 15 12:06 file2.txt
```

9. Write a shell program to find the sum of squares of first n numbers? (use while)

```
echo -n "Enter a number: "
read n
sum=0
i=1
while [ $i -le $n ]
do
    sum=$((sum + (i * i)))
    i=$((i + 1))
done
echo "Sum of squares of first $n numbers: $sum"
```

OUTPUT

```
Enter the value of n:
6
The sum of squares of first 6 numbers is: 91
```

10. Write a menu driven shell program to find the sum, difference, product, quotient of 2 numbers?

```
while true
do
    echo "MENU:"
    echo "1. Sum"
    echo "2. Difference"
    echo "3. Product"
    echo "4. Quotient"
    echo "5. Exit"
    echo -n "Enter your choice: "
    read choice
    if [ $choice -eq 5 ]; then
```

```

echo "Exiting..."
exit
fi
echo -n "Enter first number: "
read num1
echo -n "Enter second number: "
read num2
case $choice in
    1) echo "Sum: $((num1 + num2))" ;;
    2) echo "Difference: $((num1 - num2))" ;;
    3) echo "Product: $((num1 * num2))" ;;
    4) if [ $num2 -eq 0 ]; then
            echo "Division by zero is not allowed!"
        else
            echo "Quotient: $((num1 / num2))"
        fi
        ;;
    *) echo "Invalid choice!" ;;
esac
done

```

OUTPUT

```

Enter first number:
12
Enter second number:
5

Choose an operation:
1. Sum
2. Difference
3. Product
4. Quotient
5. Exit
1
Sum = 17

```

11. Write a menu driven shell program to find the month if a number gives. (repeat the menu infinitely)

```

while true
do
    echo "Enter a number (1-12) to get the corresponding month, or 0 to exit:"
    read num

```

```

if [ $num -eq 0 ]; then
    echo "Exiting..."
    exit
fi
case $num in
    1) echo "January" ;;
    2) echo "February" ;;
    3) echo "March" ;;
    4) echo "April" ;;
    5) echo "May" ;;
    6) echo "June" ;;
    7) echo "July" ;;
    8) echo "August" ;;
    9) echo "September" ;;
    10) echo "October" ;;
    11) echo "November" ;;
    12) echo "December" ;;
    *) echo "Invalid number! Enter a value between 1 and 12." ;;
esac
done

```

OUTPUT

```

Enter a number (1-12) to find the month:
Enter 0 to Exit.
3
March

Enter a number (1-12) to find the month:
Enter 0 to Exit.
0
Exiting program. Goodbye!

```

12. Write a shell program to find the factorial of a number? (Use function)

```

factorial() {
    num=$1
    fact=1
    while [ $num -gt 1 ]
        do

```

```
fact=$((fact * num))
num=$((num - 1))
done
echo "Factorial: $fact"
}
echo -n "Enter a number: "
read n
factorial $n
```

OUTPUT

```
Enter a number:
7
Factorial of 7 is: 5040
```

13. Write a shell program to print the Fibonacci numbers upto N?

```
echo -n "Enter N: "
read n
a=0
b=1
echo "Fibonacci series up to $n:"
echo -n "$a $b "
while [ $((a + b)) -le $n ]
do
    fib=$((a + b))
    echo -n "$fib "
    a=$b
    b=$fib
done
echo
```

OUTPUT

```
Enter how many Fibonacci numbers to print:
8
Fibonacci series up to 8 terms:
0 1 1 2 3 5 8 13
```

14. Read a Decimal number. Convert it to Binary and display the result. (Use while)

```
echo -n "Enter a decimal number: "
read num
binary=""
while [ $num -gt 0 ]
do
    remainder=$((num % 2))
    binary="$remainder$binary"
    num=$((num / 2))
done
echo "Binary: $binary"
```

OUTPUT

```
Enter a decimal number:
13
Binary: 1101
```

RESULT

Basic Shell scripting programs has been executed and the output is verified.

COMMAND LINE TOOLS FOR NETWORKING

AIM

Study important options and uses of following command line tools for networking.
ping, traceroute, netstat, tcpdump, ip, nslookup, route, ifconfig, ifup,, ifquery, curl, wget.

1. ping:

- **Options:** Common options include **-c** to specify the number of packets to send, **-s** to set the packet size, and **-i** to set the interval between packets.
- **Use Cases:** Ping is used to test network connectivity and measure the round-trip time between your computer and a target host.

Eg:

```
shreyas@shreyas-VirtualBox:~$ ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=63 time=20.5 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=63 time=88.7 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=63 time=9.30 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=63 time=3.73 ms
64 bytes from 192.168.1.1: icmp_seq=5 ttl=63 time=8.07 ms
64 bytes from 192.168.1.1: icmp_seq=6 ttl=63 time=4.66 ms
64 bytes from 192.168.1.1: icmp_seq=7 ttl=63 time=3.45 ms
64 bytes from 192.168.1.1: icmp_seq=8 ttl=63 time=46.4 ms
64 bytes from 192.168.1.1: icmp_seq=9 ttl=63 time=33.9 ms
64 bytes from 192.168.1.1: icmp_seq=10 ttl=63 time=13.7 ms
^C
--- 192.168.1.1 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9014ms
rtt min/avg/max/mdev = 3.446/23.242/88.690/25.645 ms
```

2. traceroute:

- **Options:** Common options include **-n** to disable DNS resolution, and **-I** or **-T** to use ICMP or TCP packets for tracing.
- **Use Cases:** Traceroute helps determine the path that packets take to reach a target host and identifies potential network issues.

Eg:

```
sreerag@sreerag-GL553VE:~/NSA/nsa$ traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1 _gateway (192.168.1.1)  1.476 ms  1.570 ms  2.042 ms
 2 * * *
 3 keralavisionisp-dynamic-5.160.199.103.keralavisionisp.com (103.199.160.5)  7.015 ms  7.118 ms  7.260 ms
 4 10.1.1.33 (10.1.1.33)  22.184 ms  22.346 ms  10.1.1.17 (10.1.1.17)  22.083 ms
 5 72.14.205.178 (72.14.205.178)  24.105 ms  23.340 ms  33.115 ms
 6 * * *
 7 dns.google (8.8.8.8)  19.436 ms  19.572 ms  19.364 ms
```

3. netstat:

- Options:** Common options include **-t** for TCP connections, **-u** for UDP connections, and **-a** to show all connections (listening and established).
- Use Cases:** Netstat provides information about active network connections, ports, routing tables, and network interface statistics.

Eg:

```
sreerag@sreerag-GL553VE:~/NSA/nsa$ netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 sreerag-GL553VE:43392   104.192.137.12:https ESTABLISHED
tcp      0      0 sreerag-GL553VE:45116   sf-in-f188.1e100.n:5228 ESTABLISHED
tcp      0      0 sreerag-GL553VE:34832   162.247.241.14:https ESTABLISHED
tcp      0      0 sreerag-GL553VE:45846   162.247.241.14:https ESTABLISHED
tcp      0      0 sreerag-GL553VE:60772   whatsapp-cdn-shv-:https ESTABLISHED
tcp      0      0 sreerag-GL553VE:59130   sd-in-f188.1e100.n:5228 ESTABLISHED
```

4. tcpdump:

- Options:** Numerous options are available to specify filters, capture interfaces, and display formats.
- Use Cases:** Tcpdump captures and displays network traffic in real-time, aiding in network troubleshooting, packet analysis, and security monitoring.

Eg:

```
shreyas@shreyas-VirtualBox:~$ sudo tcpdump -n
[sudo] password for shreyas:
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
14:03:51.527135 IP 91.189.91.48.80 > 10.0.2.15.47748: Flags [S.], seq 25088001, ack 1688368035, win 65535, option
14:03:51.527151 IP 10.0.2.15.47748 > 91.189.91.48.80: Flags [.], ack 1, win 64240, length 0
14:03:51.527214 IP 10.0.2.15.47748 > 91.189.91.48.80: Flags [P.], seq 1:88, ack 1, win 64240, length 87: HTTP: G
14:03:51.527280 IP 91.189.91.48.80 > 10.0.2.15.47748: Flags [.], ack 88, win 65535, length 0
14:03:52.140672 IP 91.189.91.48.80 > 10.0.2.15.47748: Flags [P.], seq 1:190, ack 88, win 65535, length 189: HTTP
14:03:52.140694 IP 10.0.2.15.47748 > 91.189.91.48.80: Flags [.], ack 190, win 64051, length 0
14:03:52.140822 IP 10.0.2.15.47748 > 91.189.91.48.80: Flags [F.], seq 88, ack 190, win 64051, length 0
14:03:52.140888 IP 91.189.91.48.80 > 10.0.2.15.47748: Flags [.], ack 89, win 65535, length 0
14:03:52.140939 IP 91.189.91.48.80 > 10.0.2.15.47748: Flags [F.], seq 190, ack 89, win 65535, length 0
14:03:52.140948 IP 10.0.2.15.47748 > 91.189.91.48.80: Flags [.], ack 191, win 64051, length 0
^C
17 packets captured
17 packets received by filter
0 packets dropped by kernel
```

5. ip:

- **Options:** Extensive options are available for configuring networking aspects, including addresses, routes, tunnels, and more.
- **Use Cases:** Ip is a versatile tool for managing network-related configurations and settings on modern Linux systems.

Eg:

```
shreyas@shreyas-VirtualBox:~$ ip route
default via 10.0.2.2 dev enp0s3 proto dhcp metric 100
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.15 metric 100
169.254.0.0/16 dev enp0s3 scope link metric 1000
```

6. nslookup:

- **Options:** Commonly used for interactive mode, where you can type domain names to obtain DNS information.
- **Use Cases:** Nslookup is used to query DNS records and retrieve information about domain names, IP addresses, and name servers.

Eg:

```
shreyas@shreyas-VirtualBox:~$ nslookup google.com
Server:          127.0.0.53
Address:         127.0.0.53#53

Non-authoritative answer:
Name:    google.com
Address: 142.250.195.110
Name:    google.com
Address: 2404:6800:4007:829::200e
```

7. route:

- **Options:** Options include **-n** to show numeric addresses, **-add** to add a new route, and **-delete** to remove a route.
- **Use Cases:** Route helps manage routing tables and configure network routes on the system.

Eg:

```
shreyas@shreyas-VirtualBox:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref  Use Iface
0.0.0.0         10.0.2.2      0.0.0.0       UG    100    0      0 enp0s3
10.0.2.0        0.0.0.0       255.255.255.0 U     100    0      0 enp0s3
169.254.0.0     0.0.0.0       255.255.0.0   U     1000   0      0 enp0s3
```

8. ifconfig:

- **Options:** Options include specifying the interface with **-a**, **-s** for a short output, and various flags to configure interface properties.
- **Use Cases:** Ifconfig displays and configures network interfaces, IP addresses, netmasks, and related settings.

Eg:

```
shreyas@shreyas-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
        inet6 fe80::3b7c:9962:7323:4a67 prefixlen 64 scopeid 0x20<link>
          ether 08:00:27:58:c3:11 txqueuelen 1000 (Ethernet)
            RX packets 17777 bytes 26233429 (26.2 MB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 10238 bytes 640557 (640.5 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 208 bytes 19217 (19.2 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 208 bytes 19217 (19.2 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

9. ifup, ifquery:

- **Options:** Generally used with interface names to bring up, or query status respectively.
- **Use Cases:** These tools are used to control network interfaces, enabling you to start or stop network connectivity for specific interfaces.

Eg:

```
shreyas@shreyas-VirtualBox:~$ ifquery -l
lo
shreyas@shreyas-VirtualBox:~$ sudo ifup lo
[sudo] password for shreyas:
ifup: interface lo already configured
```

10. curl:

- **Options:** Curl supports a wide range of options including **-o** to specify output file, **-X** for HTTP method, and **-H** to set headers.
- **Use Cases:** Curl is used for sending and receiving data with URLs, making it versatile for testing APIs, downloading files, and more.

Eg:

```
shreyas@shreyas-VirtualBox:~$ curl -o outputtxt.html https://ubuntu.com/download/desktop.html
% Total    % Received % Xferd  Average Speed   Time   Time     Current
          Dload  Upload Total   Spent    Left  Speed
100  221  100  221    0     0  214      0  0:00:01  0:00:01 --:--:--  214
```

11. wget:

- **Options:** Options include **-O** to specify output file, **-r** for recursive downloads, and **-nc** to skip existing files.
- **Use Cases:** Wget is used for non-interactive downloads from the web, supporting recursive downloads, mirroring, and resuming.

Eg:

```
shreyas@shreyas-VirtualBox:~$ wget http://ftp.gnu.org/gnu/wget/wget2-2.0.0.tar.gz
--2025-05-07 14:36:22--  http://ftp.gnu.org/gnu/wget/wget2-2.0.0.tar.gz
Resolving ftp.gnu.org (ftp.gnu.org)... 209.51.188.20, 2001:470:142:3::b
Connecting to ftp.gnu.org (ftp.gnu.org)|209.51.188.20|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3565643 (3.4M) [application/x-gzip]
Saving to: 'wget2-2.0.0.tar.gz'

wget2-2.0.0.tar.gz           100%[=====] 3.40M
2025-05-07 14:36:57 (66.3 MB/s) - 'wget2-2.0.0.tar.gz' saved [3565643/3565643]
```

RESULT

Command line tools for networking has been executed and the output is verified.

GNU MAKE TOOLS

AIM

Execute the gnu make tool operations.

- Create file named hellomake.c with the below code

```
#include <hellomake.h>

int main() {

    myPrintHelloMake();

    return(0);

}
```

- Create file named hellofunc.c with the below code

```
#include <stdio.h>
#include <hellomake.h>
void myPrintHelloMake(void) {
    printf("Hellomakefiles!\n");
    return;
}
```

- Create file named hellomake.c with the below code

```
void myPrintHelloMake(void);
```

- Create the make file of the 3 files with the following command
gcc -o hellomake hellomake.c hellofunc.c-I.

This compiles the two .c files and names the executable hellomake. The -I. is included so that gcc will look in the current directory (.) for the include file hellomake.h. Without a makefile, the typical approach to the test/modify/debug cycle is to use the up arrow in a terminal to go back to your last compile command so you don't have to type it each time, especially once you've added a few more .c files to the mix.

- Run the makefile created with the following command
sudo ./hellomake

OUTPUT

```
sreerag@sreerag-GL553VE:~/NSA$ nano hellomake.c
sreerag@sreerag-GL553VE:~/NSA$ nano hellofunc.c
sreerag@sreerag-GL553VE:~/NSA$ nano hellomake.h
sreerag@sreerag-GL553VE:~/NSA$ ls
hellofunc.c hellomake.c hellomake.h nsa
sreerag@sreerag-GL553VE:~/NSA$ gcc -o hellomake hellomake.c hellofunc.c -I.
sreerag@sreerag-GL553VE:~/NSA$ ls
hellofunc.c hellomake hellomake.c hellomake.h nsa
sreerag@sreerag-GL553VE:~/NSA$ sudo ./hellomake
[sudo] password for sreerag:
Hello makefiles!
```

RESULT

GNU make tool has been executed and the output is verified.

SETTING UP WEB SERVER AND WORDPRESS

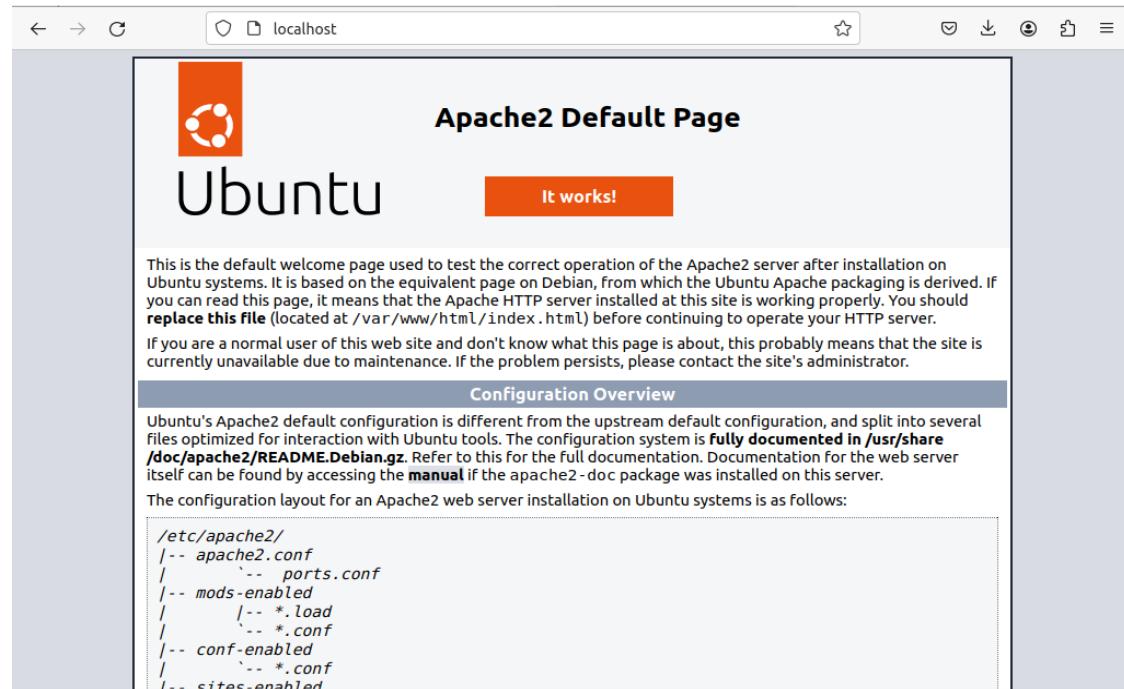
AIM

Setting up webserver and WordPress .

Install the Apache2

```
rahul@rahul-VirtualBox:~$ sudo apt install apache2
[sudo] password for rahul:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-db-sqlite3 libaprutil1-ldap
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-db-sqlite3 libaprutil1-ldap
0 upgraded, 8 newly installed, 0 to remove and 527 not upgraded.
Need to get 1,716 kB of archives.
After this operation, 7,511 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://archive.ubuntu.com/ubuntu focal/main amd64 libapr1 amd64 1:6.5-1ubuntu1 [91.4 kB]
Get:2 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libaprutil1 amd64 1:6.1-4ubuntu2.1 [84.9 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libaprutil1-db-sqlite3 amd64 1:6.1-4ubuntu2.1 [10.6 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libaprutil1-ldap amd64 1:6.1-4ubuntu2.1 [8.756 kB]
Get:5 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 apache2-bin amd64 2.4.41-4ubuntu3.14 [1.182 kB]
Get:6 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 apache2-data all 2.4.41-4ubuntu3.14 [158 kB]
Get:7 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 apache2-utils amd64 2.4.41-4ubuntu3.14 [84.4 kB]
Get:8 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 apache2 amd64 2.4.41-4ubuntu3.14 [95.6 kB]
Fetched 1,716 kB in 11s (157 kB/s)
Selecting previously unselected package libapr1:amd64.
(Reading database ... 275584 files and directories currently installed.)
Preparing to unpack .../0-libapr1_1.6.5-1ubuntu1_amd64.deb ...
Unpacking libapr1:amd64 (1:6.5-1ubuntu1) ...
Selecting previously unselected package libaprutil1:amd64.
Preparing to unpack .../1-libaprutil1_1.6.1-4ubuntu2.1_amd64.deb ...
Unpacking libaprutil1:amd64 (1:6.1-4ubuntu2.1) ...
Selecting previously unselected package libaprutil1-db-sqlite3:amd64.
Preparing to unpack .../2-libaprutil1-db-sqlite3_1.6.1-4ubuntu2.1_amd64.deb ...
Unpacking libaprutil1-db-sqlite3:amd64 (1:6.1-4ubuntu2.1) ...
Selecting previously unselected package libaprutil1-ldap:amd64.
Preparing to unpack .../3-libaprutil1-ldap_1:6.1-4ubuntu2.1_amd64.deb ...
Unpacking libaprutil1-ldap:amd64 (1:6.1-4ubuntu2.1) ...
Selecting previously unselected package apache2-bin.
Preparing to unpack .../4-apache2-bin_2.4.41-4ubuntu3.14_amd64.deb ...
Unpacking apache2-bin (2.4.41-4ubuntu3.14) ...
Selecting previously unselected package apache2-data.
Preparing to unpack .../5-apache2-data_2.4.41-4ubuntu3.14_all.deb ...
Unpacking apache2-data (2.4.41-4ubuntu3.14) ...
Selecting previously unselected package apache2-utils.
Preparing to unpack .../6-apache2-utils_2.4.41-4ubuntu3.14_amd64.deb ...
Unpacking apache2-utils (2.4.41-4ubuntu3.14) ...
Selecting previously unselected package apache2.
```

Verify that the localhost page is available



Install Mysql and mysql secure installation

```
mint@mint:~$ sudo mysql_secure_installation
Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: y

There are three levels of password validation policy:

LOW    Length >= 8
MEDIUM Length >= 8, numeric, mixed case, and special characters
STRONG Length >= 8, numeric, mixed case, special characters and dictionary      file

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 0

Skipping password set for root as authentication with auth_socket is used by default.
If you would like to use password authentication instead, this can be done with the "ALTER_USER" command.
See https://dev.mysql.com/doc/refman/8.0/en/alter-user.html#alter-user-password-management for more information.

By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : |
```

Install php,php-mysql and libapache2-mod-php

```
devika@devika-VirtualBox:~$ sudo apt install php libapache2-mod-php php-mysql
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libapache2-mod-php7.4 php-common php7.4-cli php7.4-common php7.4-json php7.4-mysql php7.4-opcache php7.4-readline
Suggested packages:
  php-pear
The following NEW packages will be installed:
  libapache2-mod-php libapache2-mod-php7.4 php php-common php-mysql php7.4 php7.4-cli php7.4-common php7.4-json php7.4-mysql php7.4-opcache php7.4-readline
0 upgraded, 12 newly installed, 0 to remove and 532 not upgraded.
Need to get 4,157 kB of archives.
After this operation, 18.5 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu focal/main amd64 php-common all 2:7.5 [11.9 kB]
Get:2 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 php7.4-common amd64 7.4.3-4ubuntu2.19 [983 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 php7.4-json amd64 7.4.3-4ubuntu2.19 [19.2 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 php7.4-opcache amd64 7.4.3-4ubuntu2.19 [198 kB]
Get:5 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 php7.4-readline amd64 7.4.3-4ubuntu2.19 [12.6 kB]
32% [Waiting for headers] 201 kB/s 14s |
```

Check the status apache2 server after performing the restart

```
devika@devika-VirtualBox:~$ sudo systemctl restart apache2
devika@devika-VirtualBox:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2023-07-25 02:29:04 IST; 14s ago
       Docs: https://httpd.apache.org/docs/2.4/
     Process: 19322 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
    Main PID: 19342 (apache2)
      Tasks: 6 (limit: 4613)
     Memory: 9.8M
    CGroup: /system.slice/apache2.service
            └─19342 /usr/sbin/apache2 -k start
              ├─19344 /usr/sbin/apache2 -k start
              ├─19345 /usr/sbin/apache2 -k start
              ├─19346 /usr/sbin/apache2 -k start
              ├─19347 /usr/sbin/apache2 -k start
              └─19348 /usr/sbin/apache2 -k start

Jul 25 02:29:04 devika-VirtualBox systemd[1]: apache2.service: Succeeded.
Jul 25 02:29:04 devika-VirtualBox systemd[1]: Stopped The Apache HTTP Server.
Jul 25 02:29:04 devika-VirtualBox systemd[1]: Starting The Apache HTTP Server...
Jul 25 02:29:04 devika-VirtualBox apachectl[19333]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' or 'ServerAlias' directive in the configuration file.
Jul 25 02:29:04 devika-VirtualBox systemd[1]: Started The Apache HTTP Server.
lines 1-21/21 (END)
```

Install php-cli

```
devika@devika-VirtualBox:~$ sudo apt install php-cli
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  php-cli
0 upgraded, 1 newly installed, 0 to remove and 534 not upgraded.
Need to get 2,792 B of archives.
After this operation, 13.3 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu focal/main amd64 php-cli all 2:7.4+75 [2,792 B]
Fetched 2,792 B in 0s (7,010 B/s)
Selecting previously unselected package php-cli.
(Reading database ... 276669 files and directories currently installed.)
Preparing to unpack .../php-cli_2%3a7.4+75_all.deb ...
Unpacking php-cli (2:7.4+75) ...
Setting up php-cli (2:7.4+75) ...
```

Create a php file in the server folder to check the webserver and run in browser.

```
rahul@rahul-VirtualBox:/var$ cd www
rahul@rahul-VirtualBox:/var/www$ cd html
rahul@rahul-VirtualBox:/var/www/html$ nano firsttt.php
rahul@rahul-VirtualBox:/var/www/html$ nano first.php
```



A screenshot of a terminal window titled "Terminal - mint@mint: /var/www/html". The window shows the nano text editor with the following content:

```
GNU nano 6.2
<?php
phpinfo();
?>
```



Download the WordPress file from internet and run the installation file through browser. Fill the required details and click on Install at last.

```
mint@mint:~/Downloads$ sudo mv wordpress /var/www/html  
mint@mint:~/Downloads$
```

```
mysql> create database mydb;  
Query OK, 1 row affected (0.00 sec)  
  
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mydb |  
| mysql |  
| performance_schema |  
| sys |  
+-----+  
5 rows in set (0.00 sec)
```

```
mysql> alter user 'root'@'localhost' identified with mysql_native_password by 'password';  
Query OK, 0 rows affected (0.00 sec)  
  
mysql>
```

The screenshot shows a Linux desktop environment with several windows open:

- Terminal Window:** Shows command-line history related to WordPress installation, including moving the WordPress files to the web root and creating a MySQL database named "mydb".
- Mozilla Firefox Browser:** The address bar shows "localhost/wordpress/wp-admin/setup-config.php?step=1". The page displays the "WordPress Setup Configuration File" step, where users are prompted to enter database connection details. The form fields are filled as follows:
 - Database Name:** mydb
 - Username:** root
 - Password:** (Redacted)
 - Database Host:** localhost
 - Table Prefix:** wp_
- Taskbar:** Shows various application icons and the date/time: 08:48 2024-04-24.

WordPress › Setup Configuration File — Mozilla Firefox

networking commands - Go × Download – WordPress.org × WordPress › Setup Configuration File +

localhost/wordpress/wp-admin/setup-config.php?step=2

Unable to write to wp-config.php file.

You can create the wp-config.php file manually and paste the following text into it.

Configuration rules for wp-config.php:

```
/* Add any custom values between this line and the "stop editing" line. */

/* That's all, stop editing! Happy publishing. */

/** Absolute path to the WordPress directory. */
if ( ! defined( 'ABSPATH' ) ) {
    define( 'ABSPATH', __DIR__ . '/' );
}

/** Sets up WordPress vars and included files. */
require_once ABSPATH . 'wp-settings.php';
```

After you've done that, click "Run the installation".

Run the installation



WordPress › Installation — Mozilla Firefox

networking commands - Go × Download – WordPress.org × WordPress › Installation +

localhost/wordpress/wp-admin/install.php?language=en_US

Information needed

Please provide the following information. Do not worry, you can always change these settings later.

Site Title myWeb

Username root
Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

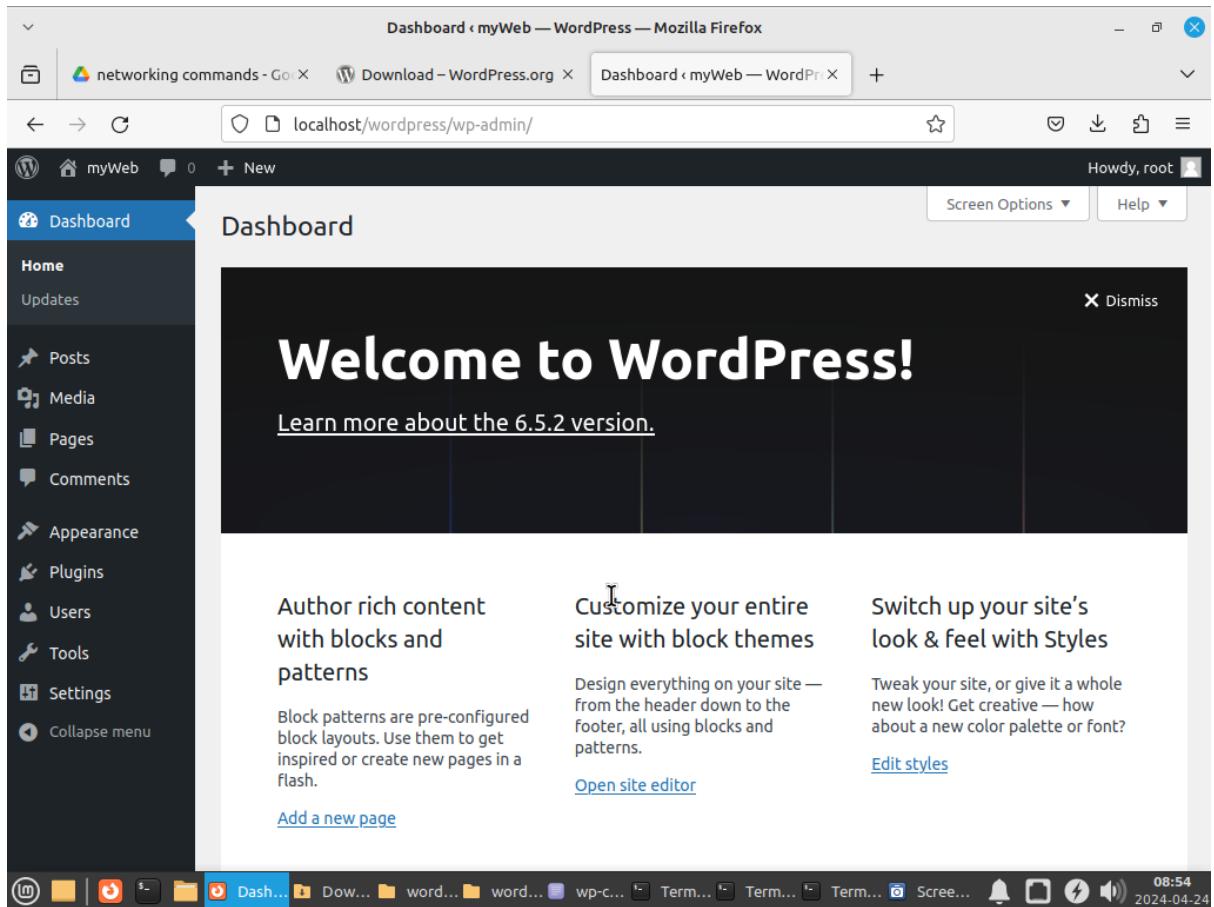
Password eccuh&oc*6enoEiRvc Hide Strong

Your Email
Double-check your email address before continuing.

Search engine visibility Discourage search engines from indexing this site
It is up to search engines to honor this request.

Install WordPress





RESULT

Webserver setup has been done tried the WordPress for creating the websites.

NETWORK PACKET STREAM ANALYSIS USING WIRESHARK

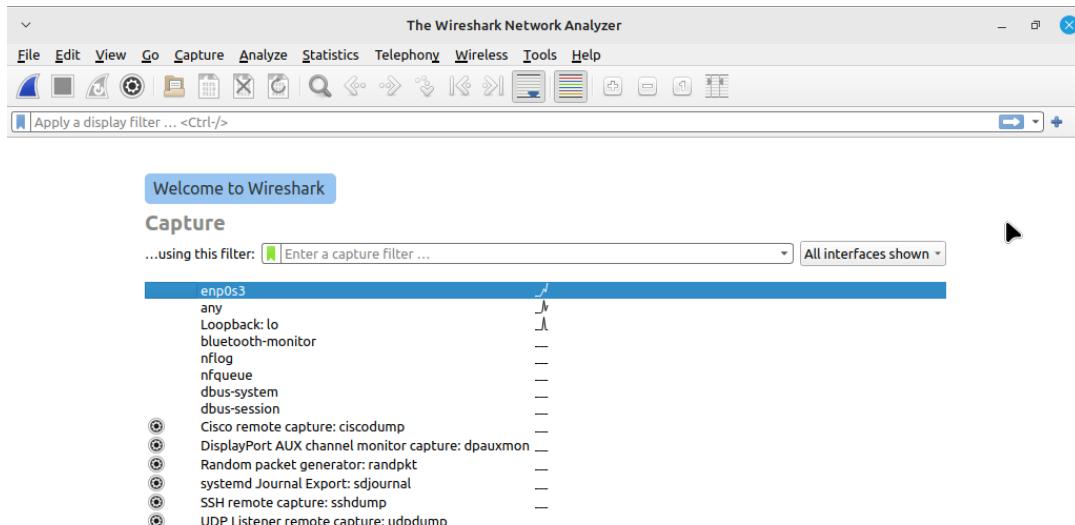
AIM

Network packet stream analysis using Wireshark

Install Wireshark

```
sreerag@sreerag-GL553VE:~$ sudo apt install wireshark
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  app-install-data-partner bsdmainutils crda g++-9 gcc-10-base
  gir1.2-clutter-1.0 gir1.2-clutter-gst-3.0 gir1.2-cogl-1.0
  gir1.2-cogl-pango-1.0 gir1.2-gnomebluetooth-1.0 gir1.2-gtkclutter-1.0
  gnome-getting-started-docs gnome-screenshot ippusrbxd libamtk-5-0
  libamtk-5-common libasn1-8-heimdal libboost-date-time1.71.0
  libboost filesystem1.71.0 libboost iostreams1.71.0 libboost locale1.71.0
  libboost thread1.71.0 libbsslapi0.7 libcamel-1.2-62 libcbor0.6 libcdio18
  libcmis-0.5-5v5 libdns-export1109 libedataserver-1.2-24
  libedataserverui-1.2-2 libextutils-pkgconfig-perl libfuse2
  libgdk-pixbuf-xlib-2.0-0 libgdk-pixbuf2.0-0 libgssapi3-heimdal
  libgupnp-1.2-0 libhandy-0.0-0 libhcrypto4-heimdal libheimbase1-heimdal
  libheimntlm0-heimdal libhogweed5 libhx509-5-heimdal libicu66 libidn11
  libisl22 libjson-c4 libjuh-java libjurt-java libkrb5-26-heimdal
  libldap-2.4-2 lib libreoffice-java libllvm12 libllvm13 libmozjs-68-0
  libmpdec2 libmysqlclient21 libneon27-gnutls libnettle7 libntfs-3g883
  liborcus-0.15-0 libphonenumber7 libpoppler97 libprotobuf17 libpython3.8
  libpython3.8-minimal libpython3.8-stdlib libqpdf26 libraw19
  libreoffice-style-tango libridl-java libroken18-heimdal libsane libssl1.1
  libstdc++-9-dev libtepl-4-0 libtracker-control-2.0-0 libtracker-miner-2.0-0
  libtracker-sparql-2.0-0 libunloader-java libvpx6 libwebp6 libwind0-heimdal
  libwmf0.2-7 ltrace lz4 ncal popularity-contest python3-entrypoints
```

Open Wireshark and you will be able to see the list of interfaces available in your system like wlp2s0, any, loopback etc.. (wlp2s0 is interface for WiFi and enp3s0 is interface for Ethernet.). You can click on the respective interface name. In the below list i have clicked on WiFi interface.

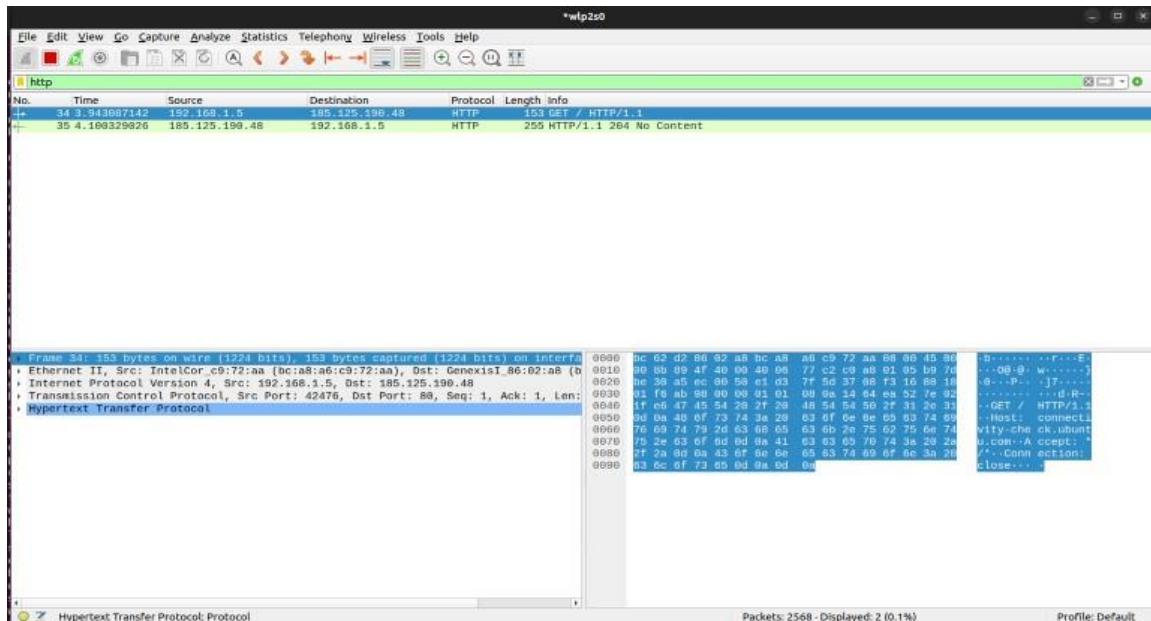


Learn

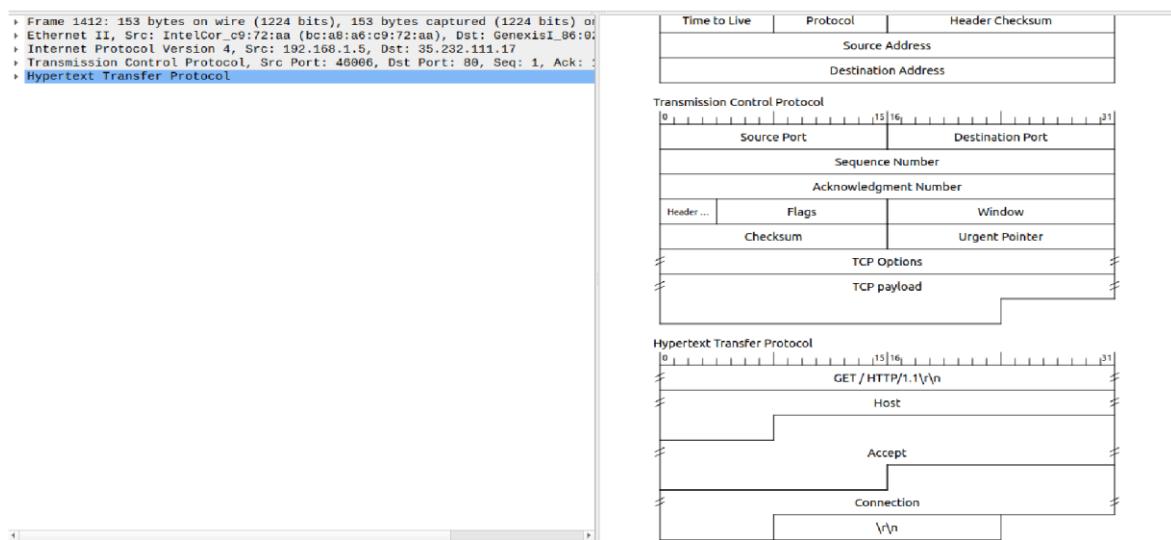
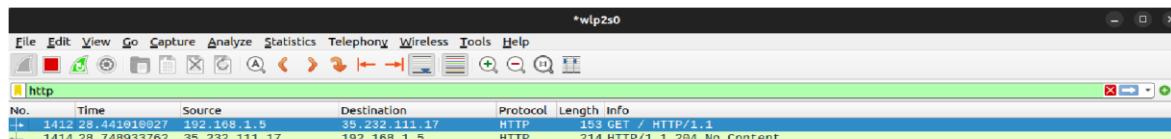
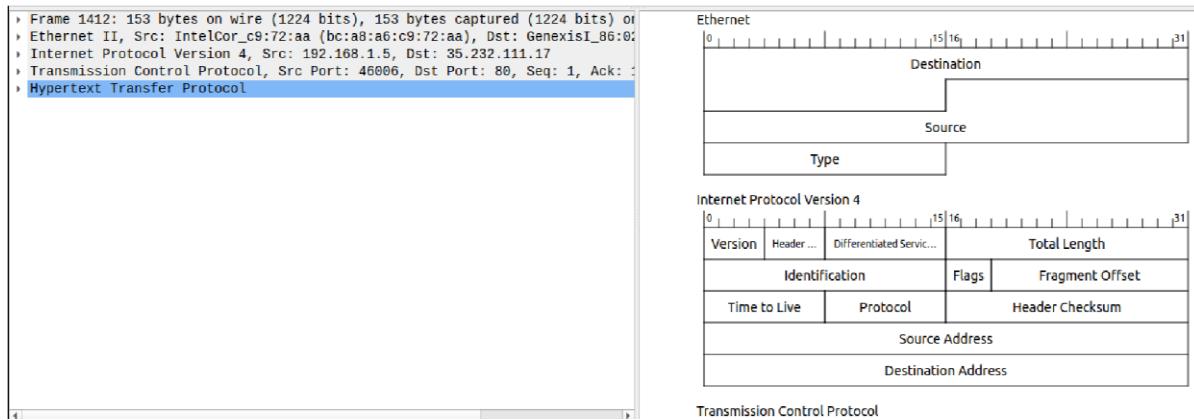
[User's Guide](#) · [Wiki](#) · [Questions and Answers](#) · [Mailing Lists](#)

You are running Wireshark 3.6.2 (Git v3.6.2 packaged as 3.6.2-2).

In this page you can see all the traffic going out and coming to the interface. Also you can use the filter option for seeing only the specific type requests like HTTP



If you check at the bottom you can identify the data's in each layers of TCP/IP



RESULT

Network packet stream analysis using Wireshark.