

MASTER OF COMPUTER APPLICATIONS

PRACTICAL RECORD WORK

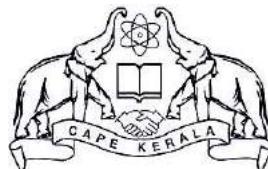
ON

20MCA136 Networking & System Administration Lab

Submitted

By

**GOPIKA P
(Reg. No. : VDA20MCA-2034)**



**DEPARTMENT OF COMPUTER APPLICATIONS
COLLEGE OF ENGINEERING VADAKARA
(CAPE - GOVT. OF KERALA)**

APRIL - 2021

**DEPARTMENT OF COMPUTER APPLICATIONS
COLLEGE OF ENGINEERING VADAKARA
(CAPE - GOVT. OF KERALA)**



CERTIFICATE

Certified that this is a bona fide record of the practical work on the course
20MCA136 NETWORKING & SYSTEM ADMINISTRATION LAB
done by Ms. GOPIKA P(Reg.No.: **VDA20MCA-2034**) Second Semester MCA
student of Department of Computer Applications at College of Engineering
Vadakara in the partial fulfilment for the award of the degree of Master of
Computer Applications (MCA) of APJ Abdul Kalam Technological University
(KTU)

(Ms. Nidhina)

(Ms. Anagha)

FACULTY-IN-CHARGE

HEAD OF THE DEPARTMENT

EXAMINERS:

| SL.NO | EXPERIMENTS | REMARKS |
|-------|--|---------|
| 1 | INTRODUCTION TO COMPUTER HARDWARE | |
| 2 | STUDY OF TERMINAL BASED TEXT EDITOR SUCH AS VIM | |
| 3 | FILE SYSTEM HIERARCHY IN A COMMON LINUX DISTRIBUTION | |
| 4 | SHELL SCRIPTING | |
| 5 | INSTALLATION AND CONFIGURATION OF LAMP STACK | |
| 6 | INSTALLATION AND CONFIGURATION OF COMMON SOFTWARE FRAMEWORKS SUCH AS LARAVEL | |
| 7 | BUILD AND INSTALL SOFTWARE FROM SOURCE CODE | |
| 8 | INTRODUCTION TO COMMAND LINE TOOLS FOR NETWORKING | |
| 9 | ANALYZING PACKET STREAM USING TCPDUMP AND WIRESHARK | |
| 10 | INTRODUCTION TO HYPERVISORS,VMS ,DOCKERS | |
| 11 | AUTOMATION USING ANSIBLE | |
| 12 | SHELL SCRIPTING PROGRAMS | |

EXPERIMENT NO: 1

AIM:- Introduction to computer hardware, physical identification of major components of a computer system such as mother board, RAM modules, daughter cards , bus slots , SMPS ,internal storage devices, interfacing ports.

Computer hardware includes the physical parts of a computer, such as the case, central processing unit (CPU), monitor, mouse, keyboard, computer data storage, graphics card, sound card, speakers and motherboard.

Hardware is typically directed by the software to execute any command or instruction. A combination of hardware and software forms a usable computing system, although other systems exist with only hardware.

MOTHER BOARD

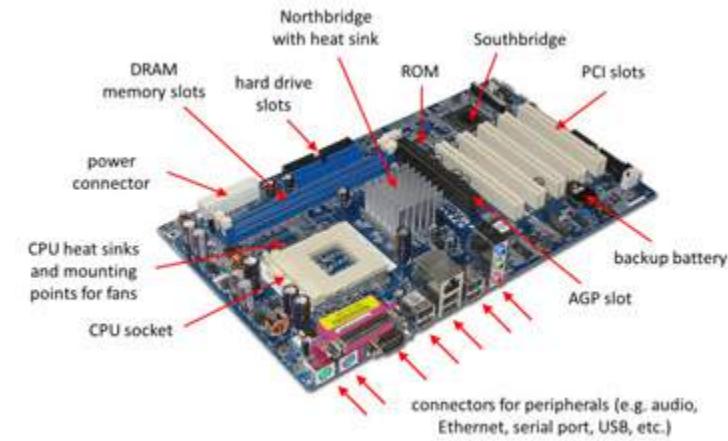
A **motherboard** (also called **main board** , **main circuit board**, **system board**, **baseboard**, **planar board**, **logic board**, or **mobo**) is the main printed circuit board (PCB) in general-purpose computers and other expandable systems. It holds and allows communication between many of the crucial electronic components of a system, such as the central processing unit (CPU) and memory, and provides connectors for other peripherals. Unlike a backplane, a motherboard usually contains significant sub-systems, such as the central processor, the chipset's input/output and memory controllers, interface connectors, and other components integrated for general use.

Location of mother board

In Desktop PC: In a desktop PC, there is a big rectangular computer case. Once the case is opened to expose inside the machine, there is a green/blue/brown/red large square printed circuit plate. This plate is the motherboard of the PC.

In laptop: While opening the bottom cover of the laptop, we will get exposed to the large PCB board which is the motherboard.

In smartphone: While opening the back cover of the smartphone, and screws up some pins then we will find the motherboard.



PARTS OF MOTHER BOARD

1) RAM chip and RAM Slot

RAM stands for Random Access Memory. It is also called the **main memory**. RAM is a **temporary data storage** device in computers and other devices. Data stored in RAM will get erased as soon as power is turned off.

RAM has **bidirectional data transfer** capacity from CPU to memory during a write operation and from RAM to CPU during the reading operation. It acts as a mediator for data transfer from CPU to other devices like HDD, cdrom, PEN drives.

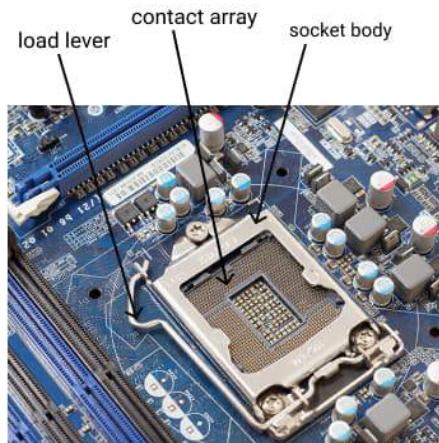
It is called **Random-access memory** because any memory address of RAM can be accessed directly from any location. If row number and column number are known then data in any memory location can be accessed.

Various types of RAM are available in the market some of them are DRAM, SDRAM, DDR, SRAM, CMOS RAM, VRAM etc. Generally available RAM in the PC market is from 2 GB to 16 GB.

2) CPU Chip and Socket



CPU Chip



CPU Socket

CPU stands for Central Processing Unit. Considered as the **brain of the computer** and other electronic devices because all the decision making tasks of the computer is performed by the CPU. It is a large printed circuit board where all

the components and peripherals are directly or indirectly connected. The main function of the CPU is to execute basics arithmetical, logical, and input/output operations.

CPU consists of 3 main typical components. ALU, CU

ALU: Arithmetical Logical Unit (ALU) is a digital circuit(gates) of CPU which is used for performing all arithmetical and logical operations. Some normal arithmetical operations performed by ALU are addition, subtraction, multiplication, and division. Some logical operations performed by ALU is comparisons between numbers and letters. A single CPU may also contain more than one ALU.

CU: Control Unit (CU) is a digital circuit of CPU which controls all the operations within the CPU. It allows and teaches various logical units, I/O devices, the memory of computer how to respond to a program's instructions of the various components as well as the user.

Memory Unit(MU)

3) PCI Slots and PCI Chip



PCI Chip



PCI Slots

Peripheral Component Interconnected(PCI) is an attached hardware component of motherboard for connecting various hardware components like modems, disk controller, NIC cards, Sound Card, graphics cards, SSD add-on cards, RAID cards, extra USB and serial port required so PCI slots help increasing motherboard capabilities without adding or replacing the motherboard.

If there is only limited ports and slots on the motherboard to connects various types of hardware devices like saying graphics card port(AGP port) then you can use PCI slots to connects Graphics cards and enjoy the same features. Same way if you have limited USB port in your computer system and want more than you can use a USB expansion card and get more USB port in your system.

It was introduced in 1992, before that **ISA**, **EISA** was used for the same purpose. Later in 2004 **PCIe slot** is developed and it is replacing **PCI slots**, **AGP slots**, and **ISA slots**.

4) AGP Slot and Chip

Accelerated Graphics Port Slot(AGP Slot) is a kind of expansion slot like a PCI slot but mainly designed for graphics cards. It was first introduced by Intel in 1996. We can easily locate this expansion slot because it is usually presented in **brown color**.

5) North Bridge

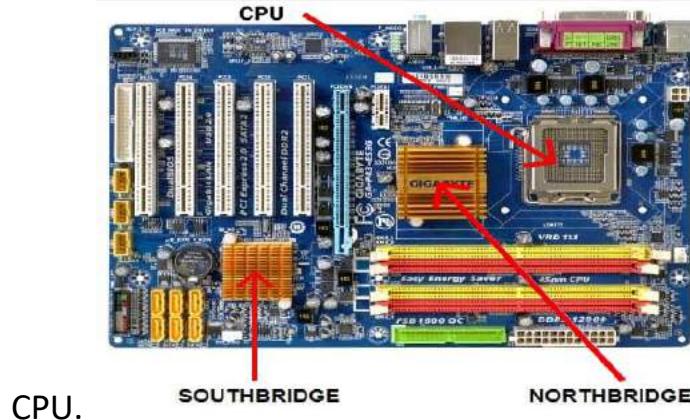
North Bridge is also called Host bridge or Memory Controller Hub. It acts as the primary controller in the motherboard which directs traffics to and from the CPU. So, the performance of the computer also depends on the northbridge chip. It does lots of processing so it generally comes with a heatsink.

Characteristics of North Bridge:

- It connects southbridge to the CPU.
- It handles and communicates faster components on the motherboard like Main Memory, AGP, PCIe, ROM, and CPU.
- It acts as a controller in bus speed on the motherboard.

- Generally, it does lots of work with the CPU, so it is located near to the CPU generally with the heatsink.
- It is a core component and is directly connected to the CPU.

In some processors of Intel, all the functioning of northbridge is performed by



6) South Bridge

Southbridge is an IC chip that generally handles and controls IO functioning in the motherboard. Unlike Northbridge, it does not have direct connection with CPU. It generally handles low-speed devices because its communication speed is lower. Instruction from CPU reaches northbridge then from northbridge to southbridge. It is connected to the PCI bus, ISA buses, IDE buses, audio, serial devices like mouse, keyboard, USB ports, etc, and SATA hard disk connector.

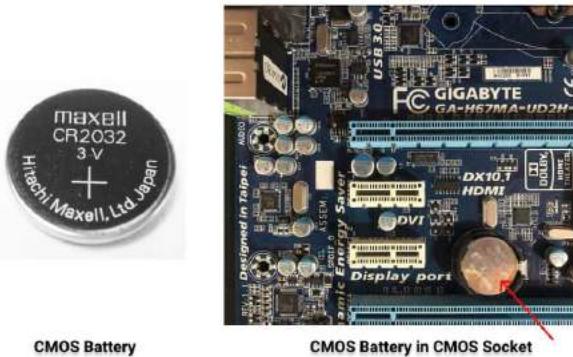
In size, it is smaller than the northbridge. And in some southbridge, we can find heatsink attached to it.

7) CMOS Backup Battery

CMOS stands for "**Complementary Metal Oxide Semiconductor**" and found in both laptop and desktop PC as a small circular coin shape. CMOS stores a wide range of system information like current system clock, date, time, pulses, mostly used hardware settings, BIOS configuration settings, BOOT sequences, BIOS master/admin password, GPU and virtualization settings, power management, etc. They can save those set for a longer time around 2 to 10 years. CMOS works continuously even if you shut down your system because it is continually holding

all those setting mention above. CMOS is also called as **CMOS RAM, COS-MOS, NVRAM**(Non-Volatile RAM) in the market.

It is also called RTC(Real Time Clock) of the computer system because even computer is shut down it is able to store all the required information that the system required to boot the system next time.



8)Power Supply Plug

The main work of the Power Supply port in the Motherboard is to provide power to Motherboard and its attached components and peripherals.

i) 24 (20 + 4) ATX power supply

In modern PCs, ATX power supply is provided which is 24 Pin(20 + 4) Main Power Supply Connector (Older PCs only have 20 Pin)

ii) 4 Pin or 8 Pin Connector

This port in the motherboard is to provide dedicated power to the CPU. Older PCs may not have this Plugin motherboard but modern computers can do lots of works like overclocking so, a dedicated cable is provided to the CPU.

8Pin connector can be split into two and each split part can be used as 4 pin connector.

iii) PCI-Express 6-Pin or 8-Pin Connector

This is required to power the PCI-E port. PCI-E slot required 75W power to operates.

The older PC does not have this.

iv) Molex

Molex pin is 4 power pin which is required to supply power to older CDROM and hard drives. Molex is nowadays used for Case Fan. (some have some do not have)

Molex connector comes with Mini Molex connectors, which is used for floppy disk drives in much older PCs.

v) SATA power supply

Modern hard drives and CDROM uses SATA cable for power. In motherboard, it is L-shape port and so its cable is connected to SATA port in one way only. In motherboard, it has 15 pins. It provides features of hot-swappable hard drives ie. plug and play hard drive features.

9) SATA and PATA Port and Connector

PATA stands for Parallel Advanced Technology Attachment. It is 40 pins long and wide ribbon cable used for connecting mass storage devices like hard disks(HDD or SSD), optical drives to the computer. It was launched in 1986 by Western Digital and Compaq. Every cable of PATA has two or three connectors, of which one is attached to the adapter interfacing and the remaining are plugged into secondary storage devices.

In modern computers, it is not used. It is outdated technology and is replaced by SATA Technology

SATA stands for Serial Advanced Technology Attachment. It is 7 pin cable which is shorter and powerful than the PATA connector and its function is the same as the PATA connector.

10) Parallel Port

A parallel port is used to transfer in a parallel manner through multiple communication channels. Used for printers, scanner, Zip Drive, external HDD, tape backup devices, external CD ROM, etc.

11) Serial Port

With a serial port, only one bit of data gets transfer at a time. It is found in an older PC to connect older keyboards, PDAs, external modems.

RAM MODULES

A memory module is another name for a RAM chip. It is often used as a general term used to describe SIMM, DIMM, and SO-DIMM memory. While there are several different types of memory modules available, they all serve the same purpose, which is to store temporary data while the computer is running.

Memory modules come in different sizes and have several different pin configurations. For example, the original SIMMs had 30 pins (which are metal contacts that connect to the motherboard). However, newer SIMM chips have 72 pins. DIMMs commonly come in 168-pin configurations, but some DIMMs have as many as 240 pins. SO-DIMMs have a smaller form factor than standard DIMM chips, and come in 72-pin, 144-pin, and 200-pin configurations.

While "memory module" is the technical term used to describe computer memory, the terms "RAM," "memory," and "RAM chip" are just as acceptable. But remember, while memory terms may be interchangeable, the memory itself is not. This is because most computers only accept one type of memory. Therefore, if you decide to upgrade your computer's RAM, make sure the memory modules you buy are compatible with your machine.



DAUGHTER BOARD / DAUGHTER CARD

The daughter board is a computer hardware. It is also known as the piggyback board, riser card, daughter board, daughter card or daughter card. A daughter board is a printed circuit board which is connected to the motherboard or expansion card. As compared to the motherboard, it is smaller in size. A daughter board does not act as an expansion card. An expansion card adds extra new functions to the computer. But a daughter board that is connected to the motherboard adds or supports the main functions of the motherboard.

Daughter boards are directly connected to the motherboards. We know that expansion cards are connected to the motherboard by using the bus and other serial interfaces. But daughter board is directly connected to the board by soldering. As an update of the motherboard or expansion card, daughter boards are released to extend the features and services of the motherboard or expansion cards.



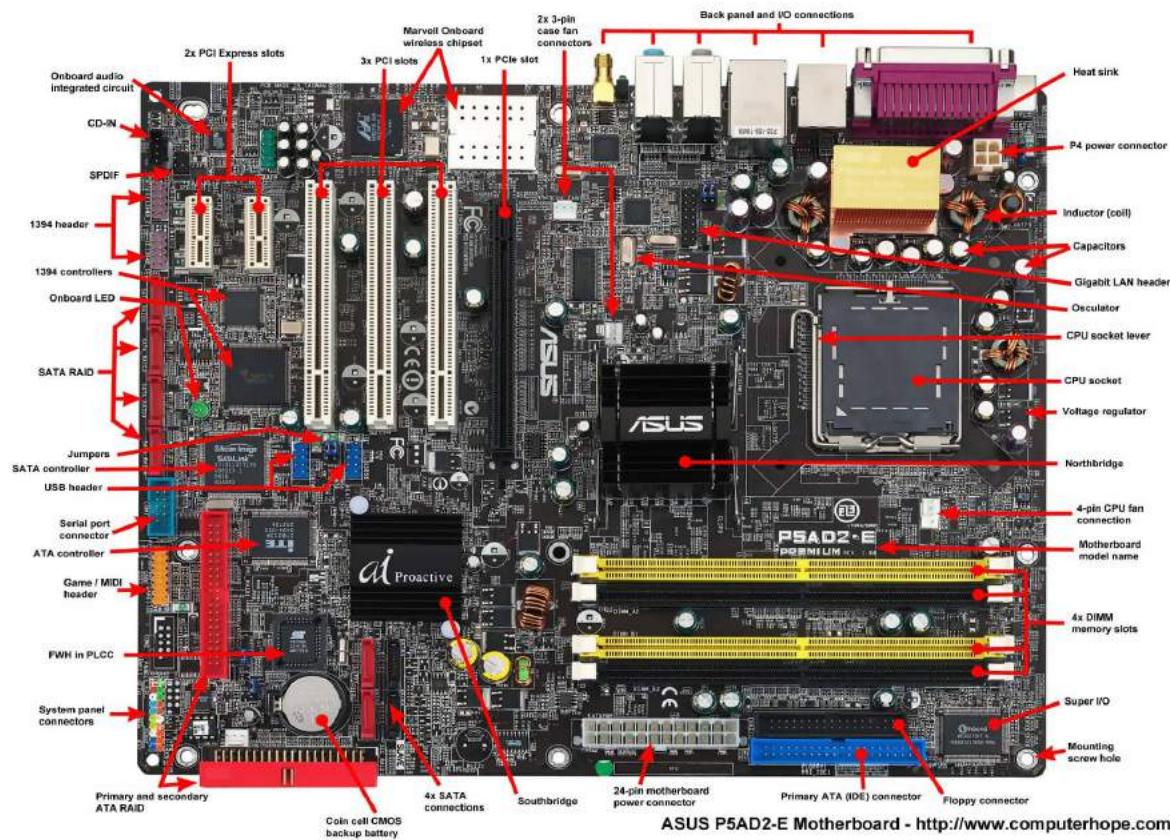
BUS SLOT/EXPANSION SLOT

A **bus slot** or **expansion port**, an **expansion slot** is a connection or port inside a computer on the motherboard or riser card. It provides an installation point for a hardware expansion card to be connected. For example, if we wanted to install a new video card in the computer, we'd purchase a video expansion card and install that card into the compatible expansion slot.

EXPANSION/ BUS SLOTS AND DEVICES CONNECTED

- **AGP** - Video card.
- **AMR** - Modem, sound card.
- **CNR** - Modem, network card, sound card.
- **EISA** - SCSI, network card, video card.
- **ISA** - Network card, sound card, video card.
- **PCI** - Network card, SCSI, sound card, video card.
- **PCI Express** - Video card, modem, sound card, network card.
- **VESA**- Video card.

Many of the expansion card slots above are obsolete. We're most likely only going to encounter AGP, PCI, and PCI Express when working with computers today.



SMPS (SWITCHED MODE POWER SUPPLY)

The SMPS is **Switched Mode Power Supply** also known as **Switching Mode Power Supply**. SMPS is an electronic power supply system that makes use of a switching regulator to transfer electrical power effectively. It is a PSU (power supply unit) and is usually used in computers to change the voltage to the appropriate range for the computer.

There are four main types of SMPS such as

- DC to DC Converter
- AC to DC Converter

- Fly back Converter
- Forward Converter

The AC to DC conversion part in the input section makes the difference between AC to DC converter and DC to DC converter. The Fly back converter is used for Low power applications. Also there are Buck Converter and Boost converter in the SMPS types which decrease or increase the output voltage depending upon the requirements. The other type of SMPS include Self-oscillating fly-back converter, Buck-boost converter, Cuk, Sepic, etc.



Working principles of SMPS

In the SMPS device, the switching regulators are used which switches on and off the load current to maintain and regulate the voltage output. Suitable power generation for a system is the mean voltage between off and on. Unlike the linear power supply, the SMPS carry transistor switches among low dissipation, full-on

and full-off phase, and spend much less time in high dissipation cycles, which decreases depleted strength.

Benefits of SMPS

- The switch-mode power source is small in scale.
- The SMPS is very lightweight.
- SMPS power consumption is typically 60 to 70 per cent, which is ideal for use.
- SMPS is strongly anti-interference.
- The SMPS production range is large.

Limitations of SMPS

- The complexity of SMPS is very large.
- The production reflection is high and its control is weak in the case of SMPS.
- Use of SMPS can only be a step-down regulator.
- In SMPS, the voltage output is just one.

INTERNAL STORAGE DEVICES

Some storage devices are classed as 'internal' which means they are inside the computer case.

Most computers have some form of internal storage. The most common type of internal storage is the hard disk.

HARD DISK

At the most basic level, internal storage is needed to hold the operating system so that the computer is able to access the input and output devices.

It will also be used to store the applications software that you use and more than likely, the original copies of your data files.

Internal storage allows the data and applications to be loaded very rapidly into memory, ready for use. The data can be accessed much faster than data which is stored on an external storage device. This is because internal storage devices are connected directly to the motherboard and its data bus whereas external devices are connected through a hardware interface such as USB, which means they are considerably slower to access.

Internal storage also means that if the computer is moved around, it will still retain its most commonly used data.

The main disadvantage of internal storage is that when the hard disk fails (and it will), all the data and applications may be lost.

This can be avoided to some extent by using more than one hard disk within the machine. Each hard disk has a copy of all the data, so if one fails the other can carry on. This is called a RAID array. An alternative is to use external drives for backup



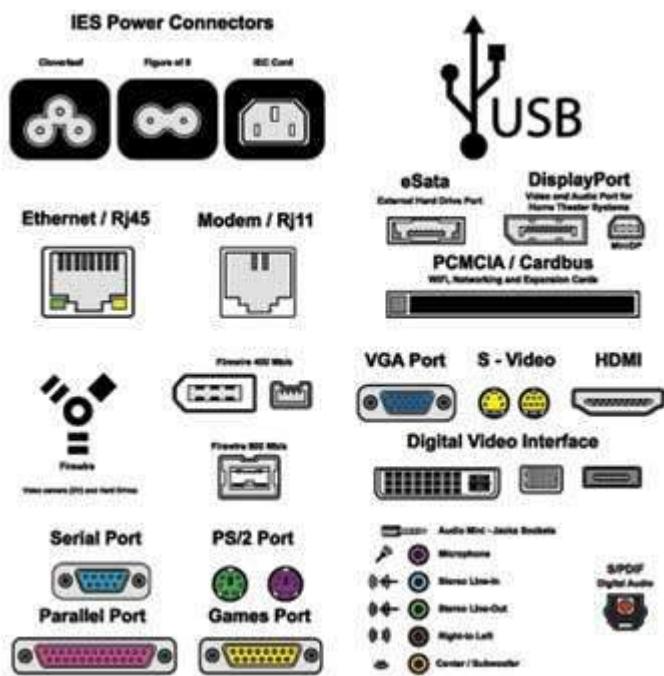
INTERFACINGPORTS

A port is a physical docking point using which an external device can be connected to the computer. It can also be programmatic docking point through which information flows from a program to the computer or over the Internet.

Characteristics of Ports

A port has the following characteristics –

- External devices are connected to a computer using cables and ports.
- Ports are slots on the motherboard into which a cable of external device is plugged in.
- Examples of external devices attached via ports are the mouse, keyboard, monitor, microphone, speakers, etc.



Types of ports –

Serial Port

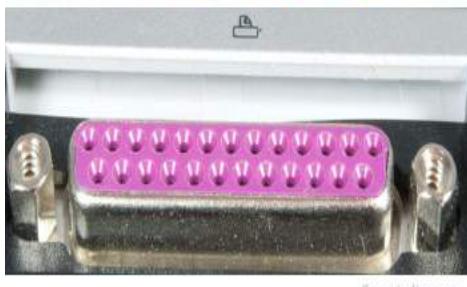
- Used for external modems and older computer mouse
- Two versions: 9 pin, 25 pin model
- Data travels at 115 kilobits per second



Parallel Port

- Used for scanners and printers
- Also called printer port
- 25 pin model
- IEEE 1284-compliant Centronics port

Parallel Port (DB25)



PS/2 Port

- Used for old computer keyboard and mouse
- Also called mouse port
- Most of the old computers provide two PS/2 port, each for the mouse and keyboard
- IEEE 1284-compliant Centronics port

Computer PS/2 Ports



Universal Serial Bus (or USB) Port

- It can connect all kinds of external USB devices such as external hard disk, printer, scanner, mouse, keyboard, etc.
- It was introduced in 1997.
- Most of the computers provide two USB ports as minimum.
- Data travels at 12 megabits per seconds.
- USB compliant devices can get power from a USB port



VGA Port

- Connects monitor to a computer's video card.
- It has 15 holes.
- Similar to the serial port connector. However, serial port connector has pins, VGA port has holes.

- Power Connector
- Three-pronged plug.
- Connects to the computer's power cable that plugs into a power bar or wall socket.



Firewire Port

- Transfers large amount of data at very fast speed.
- Connects camcorders and video equipment to the computer.
- Data travels at 400 to 800 megabits per seconds.
- Invented by Apple.
- It has three variants: 4-Pin FireWire 400 connector, 6-Pin FireWire 400 connector, and 9-Pin FireWire 800 connector.



Modem Port

- Connects a PC's modem to the telephone network.
- Ethernet Port
- Connects to a network and high speed Internet.
- Connects the network cable to a computer.
- This port resides on an Ethernet Card.
- Data travels at 10 megabits to 1000 megabits per seconds depending upon the network bandwidth.



Game Port

- Connect a joystick to a PC

- Now replaced by USB
- Digital Video Interface, DVI port
- Connects Flat panel LCD monitor to the computer's high-end video graphic cards.
- Very popular among video card manufacturers.



Digital Video Interface, DVI port

- Connects Flat panel LCD monitor to the computer's high-end video graphic cards.
- Very popular among video card manufacturers.



TEXT EDITOR :- VIM

A text editor is a type of computer program that edits plain text.

Vim is a highly configurable text editor built to enable efficient text editing. It is an improved version of the vi editor distributed with most UNIX systems.

Vim is often called a "programmer's editor," and so useful for programming that many consider it an entire IDE. It's not just for programmers, though. Vim is perfect for all kinds of text editing, from composing email to editing configuration files.

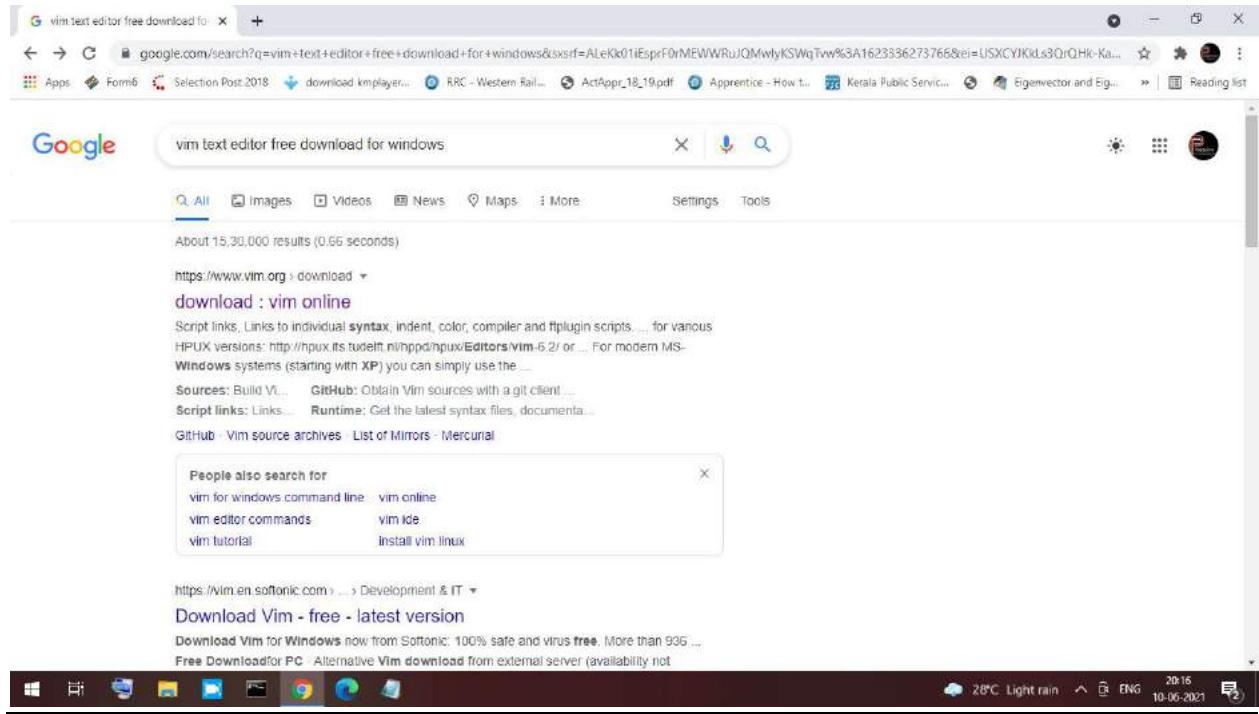
Despite what the above comic suggests, Vim can be configured to work in a very simple (Notepad-like) way, called evim or Easy Vim.

Vim is rock stable and is continuously being developed to become even better. Among its features are:

- persistent, multi-level undo tree
- extensive plugin system
- support for hundreds of programming languages and file formats
- powerful search and replace
- integrates with many tools

STEPS TO INSTALL VIM

1. Search vim text editor download for windows on google.



2. Click on the first link available , i.e,

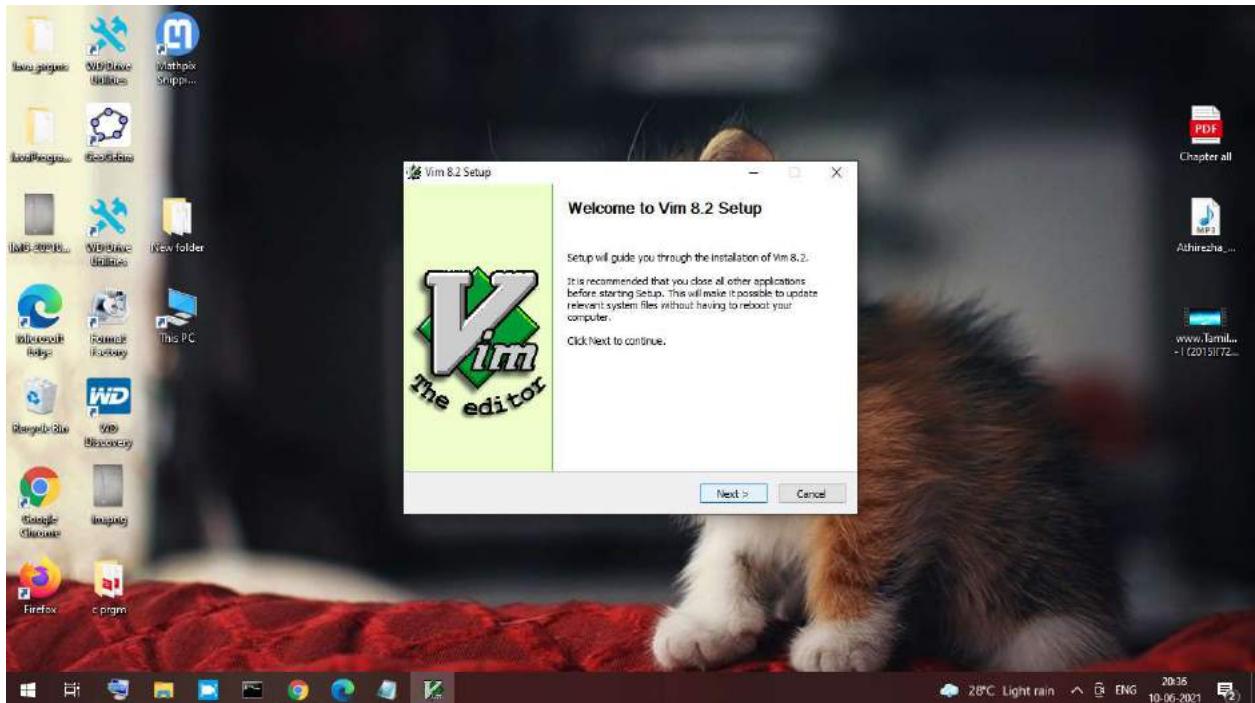
<https://www.vim.org/download.php>

not logged in ([login](#))
ENHANCED BY Google
Search |

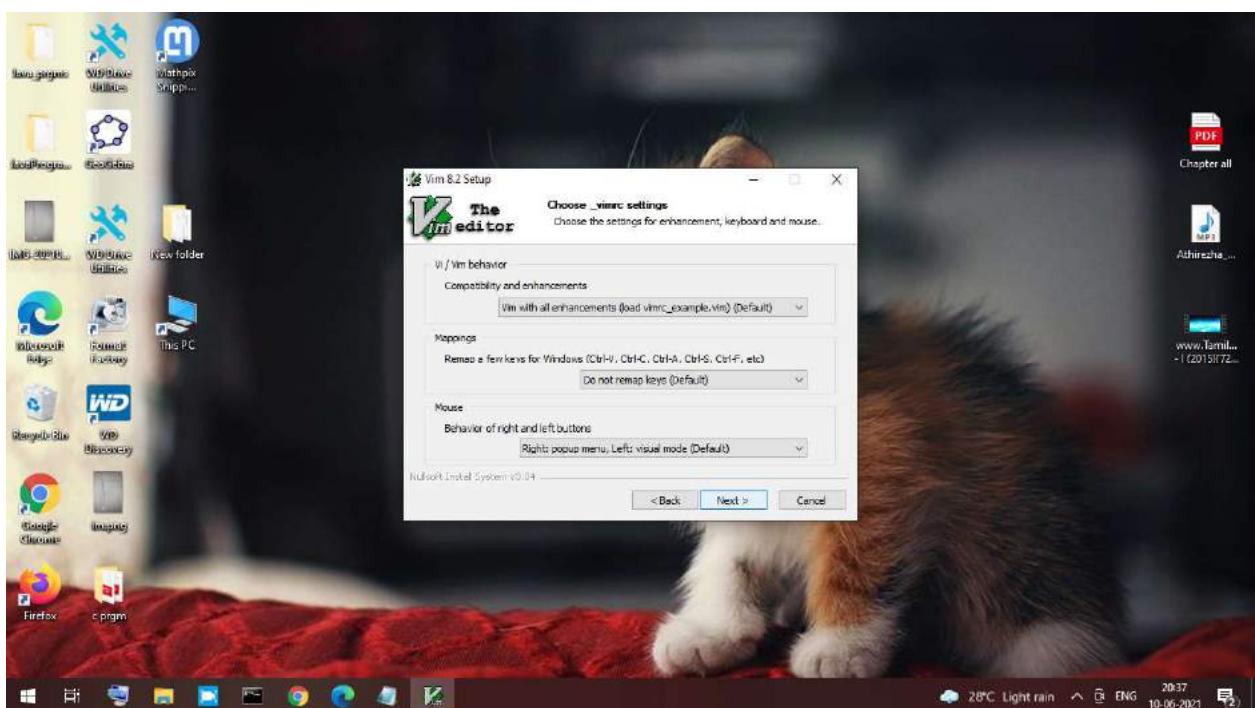
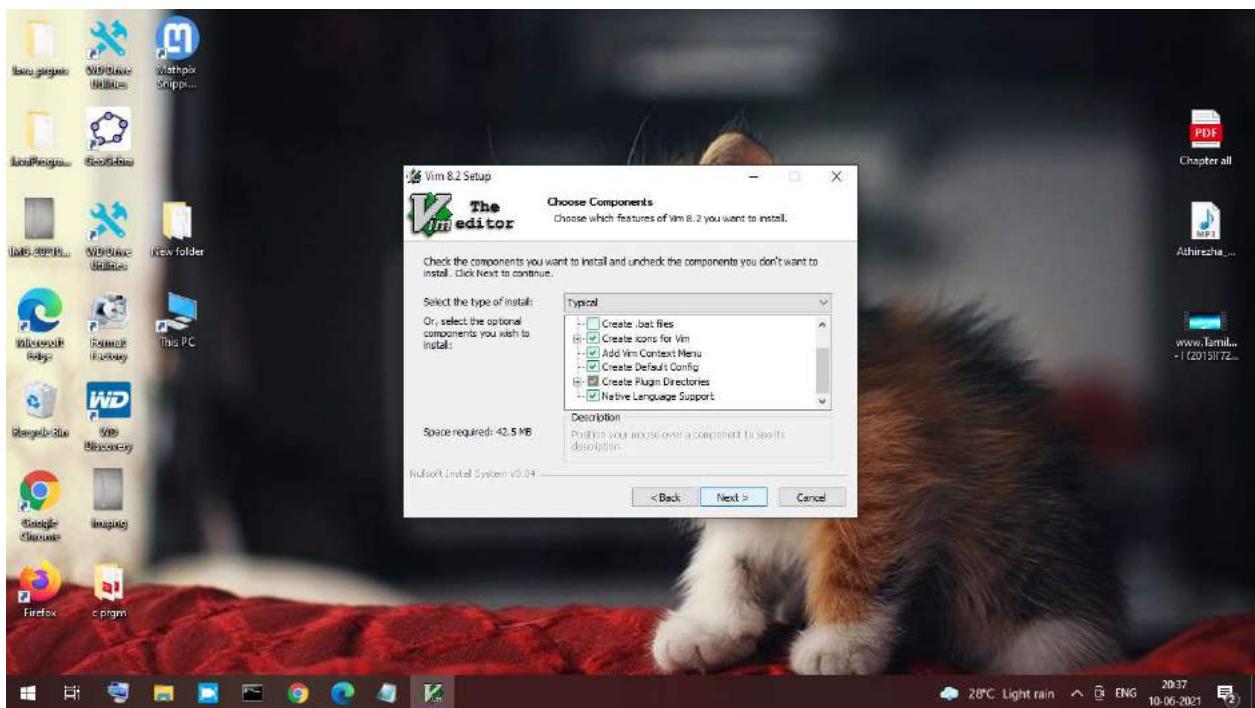
Home Advanced search
About Vim Community News Sponsoring Trivia Documentation
Download
Vim from GitHub Vim from Mercurial List of Mirrors Sources Patches Development Runtime files Script links Translations Old stuff
Scripts Tips My Account
Site Help

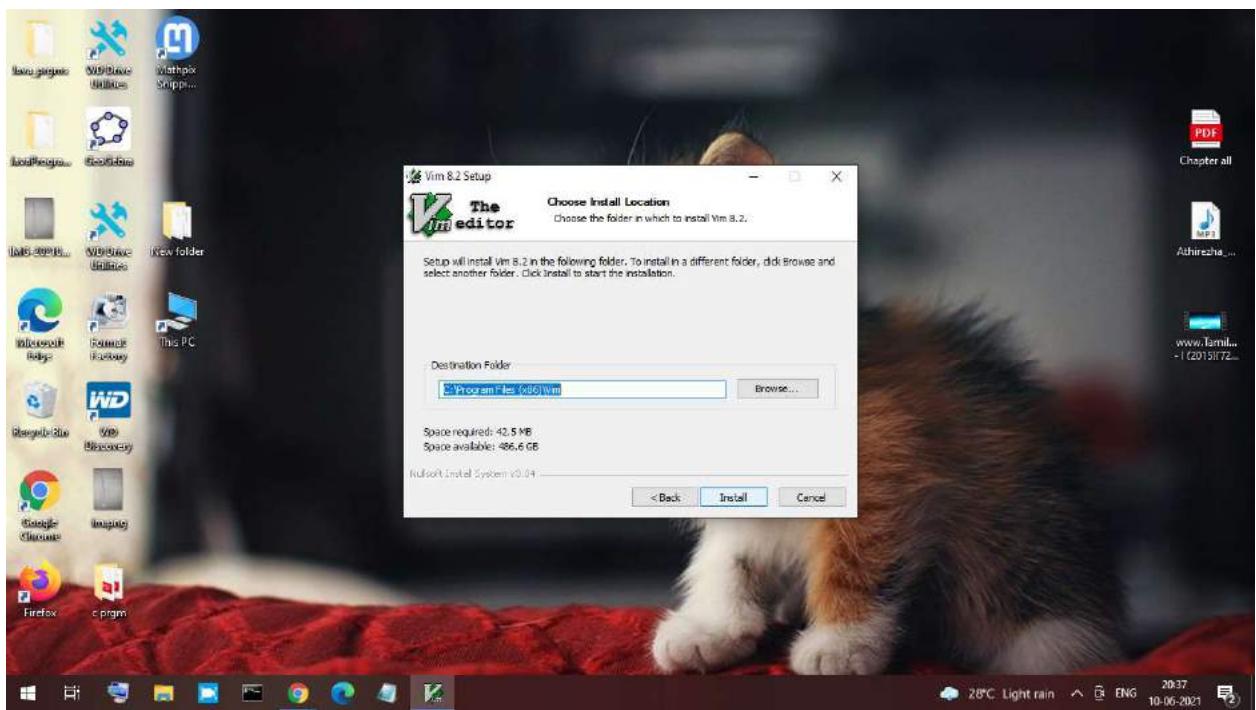
https://github.com/vim/vim-win32-installer/releases/download/v8.2.2825/gvim_8.2.2825_x86_signed.exe
28°C Light rain ENG 10-06-2021 20:30

3. Click on [gvim 8.2.2825.exe](#) and an executable file will get downloaded and follow the steps as in the images given below.

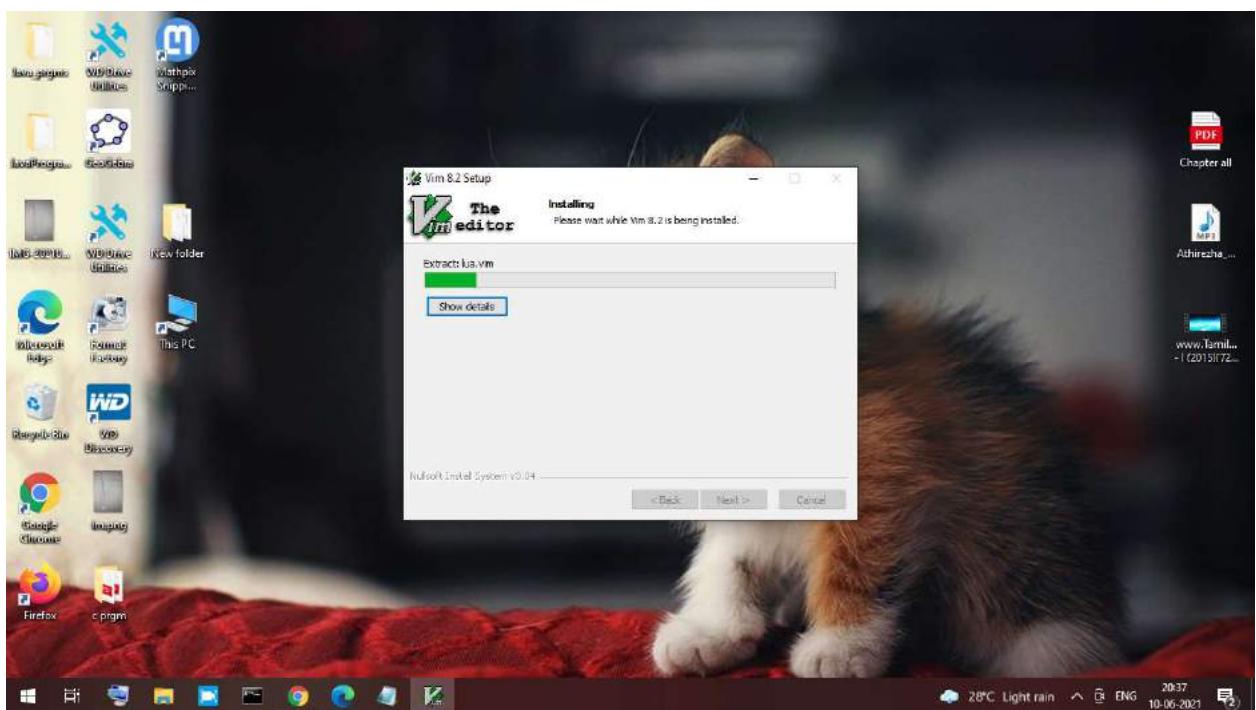


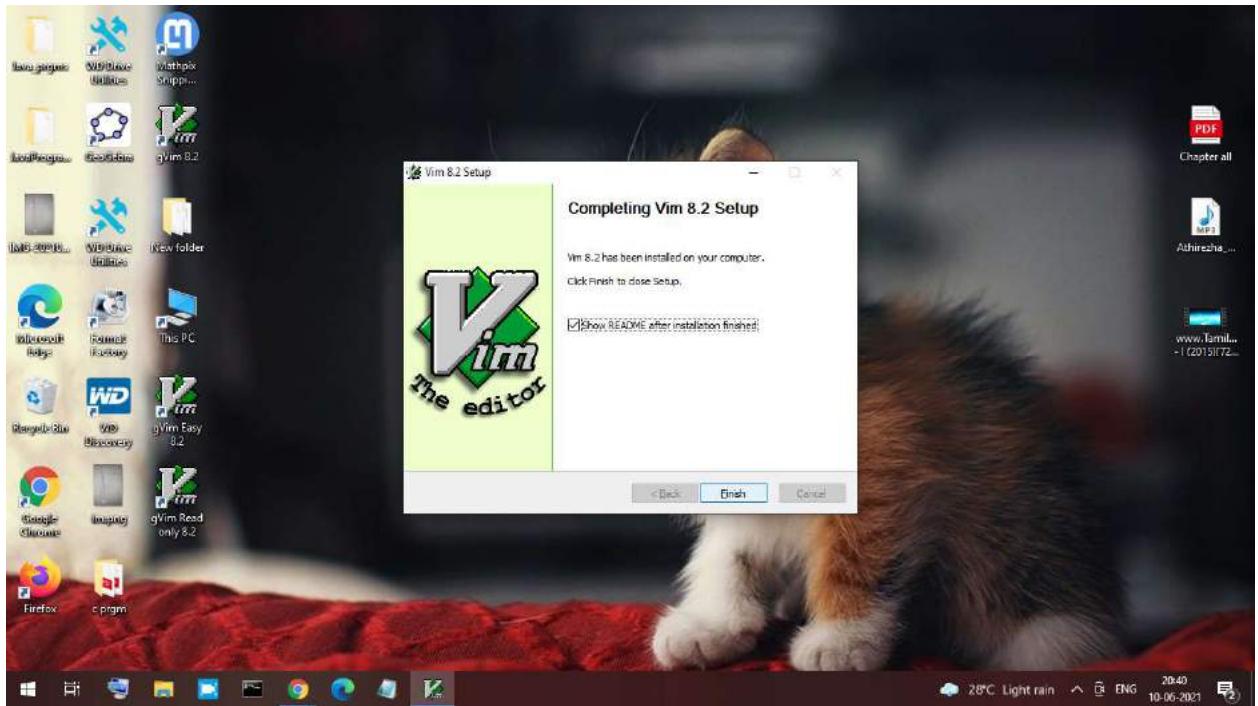
Click on next





Click on install.





Now the installation of vim text editor on the pc is completed and set the path variable by using edit the environment variable option on the pc.

MODES IN VIM

There are three modes in vim. They are command mode ,insert mode and last line mode.

In the command mode, user can move around the file, delete text, etc.

In the insert mode, user can insert text.

In the last line mode, we can move back to the previous stages.

- **Command Mode:** When vi starts up, it is in Command Mode. This mode is where vi interprets any characters we type as commands and thus does not display them in the window. This mode allows us to move through a file, and to delete, copy, or paste a piece of text.

To enter into Command Mode from any other mode, it requires pressing

the **[Esc]** key. If we press [Esc] when we are already in Command Mode, then vi will beep or flash the screen.

Commands in command mode:

Moving within a file

- k - Move cursor up
- j - Move cursor down
- h - Move cursor left
- l - Move cursor right
- **Insert mode:** This mode enables you to insert text into the file. Everything that's typed in this mode is interpreted as input and finally, it is put in the file. The vi always starts in command mode. To enter text, you must be in insert mode. To come in insert mode you simply type i. To get out of insert mode, press the Esc key, which will put you back into command mode.

Commands in insert/input mode:

- i - Insert at cursor (goes into insert mode)
- a - Write after cursor (goes into insert mode)
- A - Write at the end of line (goes into insert mode)
- ESC - Terminate insert mode
- u - Undo last change
- U - Undo all changes to the entire line
- o - Open a new line (goes into insert mode)
- dd - Delete line
- 3dd - Delete 3 lines.

- D - Delete contents of line after the cursor
- C - Delete contents of a line after the cursor and insert new text. Press ESC key to end insertion.
- dw - Delete word
- 4dw - Delete 4 words
- cw - Change word
- x - Delete character at the cursor
- r - Replace character
- R - Overwrite characters from cursor onward
- s - Substitute one character under cursor continue to insert
- S - Substitute entire line and begin to insert at the beginning of the line
- ~ - Change case of individual character

- **Last Line Mode(Escape Mode):** Line Mode is invoked by typing a colon [:], while vi is in Command Mode. The cursor will jump to the last line of the screen and vi will wait for a command. This mode enables you to perform tasks such as saving files, executing commands.

Saving and Closing the file

- Shift+zz - Save the file and quit
- :w - Save the file but keep it open
- :q !- Quit without saving
- :wq - Save the file and quit

- w!-write read-only file
- q - quit

EXPERIMENT NO : 2.1

AIM : LINUX COMMANDS

The Linux commands is a utility of the Linux operating system. All basic and advanced tasks can be done by executing commands in the command line interface, which is the Linux terminal.

open terminal: CTRL+ALT+T

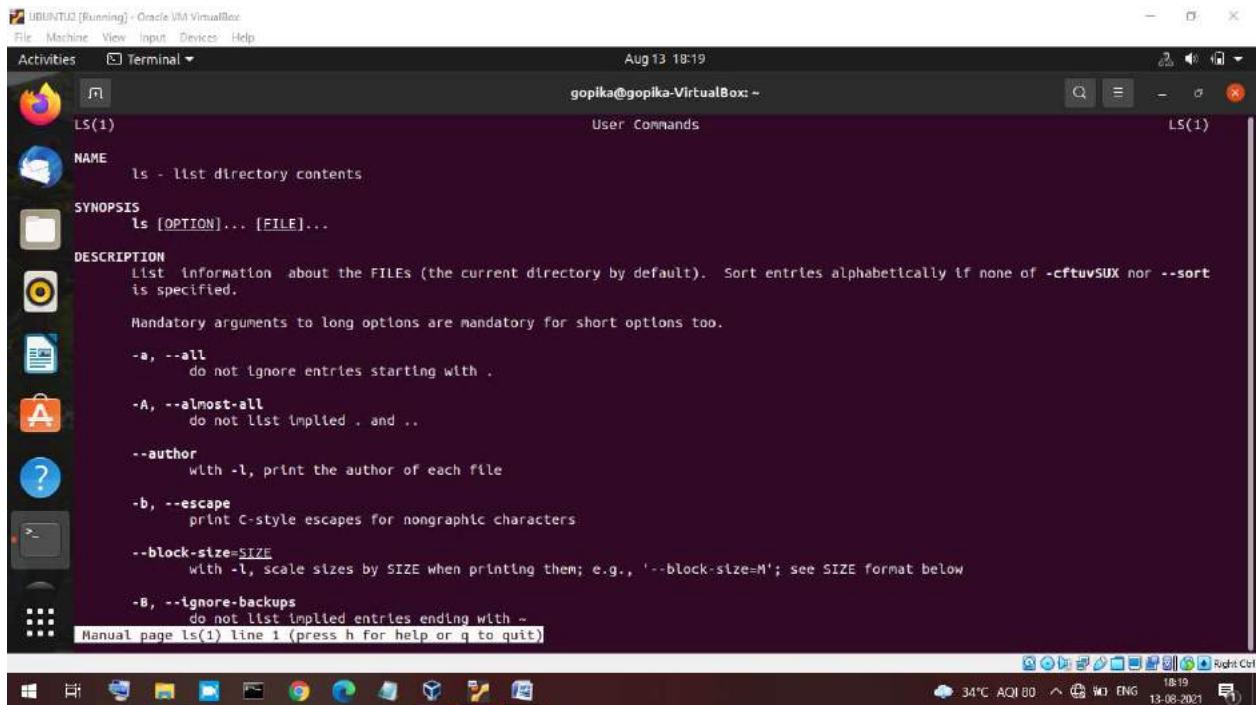
1. man

man command in Linux is used to display the user manual of any command that we can run on the terminal. It provides a detailed view of the command which includes NAME, SYNOPSIS, DESCRIPTION, OPTIONS, EXIT STATUS, RETURN VALUES, ERRORS, FILES, VERSIONS, EXAMPLES, AUTHORS and SEE ALSO.

Every manual is divided into the following sections:

- Executable programs or shell commands
- System calls (functions provided by the kernel)
- Library calls (functions within program libraries)
- Games
- Special files (usually found in /dev)
- File formats and conventions eg /etc/passwd
- Miscellaneous (including macro packages and conventions), e.g. groff(7)
- System administration commands (usually only for root)
- Kernel routines [Non standard]

```
gopika@gopika-VirtualBox:~$ man ls
gopika@gopika-VirtualBox:~$
```



2.cd ,mkdir, pwd, find, rmdir

- **pwd** –to know the present working directory

```
gopika@gopika-VirtualBox:~/Desktop$ pwd
/home/gopika/Desktop
gopika@gopika-VirtualBox:~/Desktop$
```

- **mkdir** –to create a new directory

```
gopika@gopika-VirtualBox:~/Desktop$ mkdir file1
gopika@gopika-VirtualBox:~/Desktop$ ls
file1
gopika@gopika-VirtualBox:~/Desktop$
```

- **rmdir** –used to delete a directory

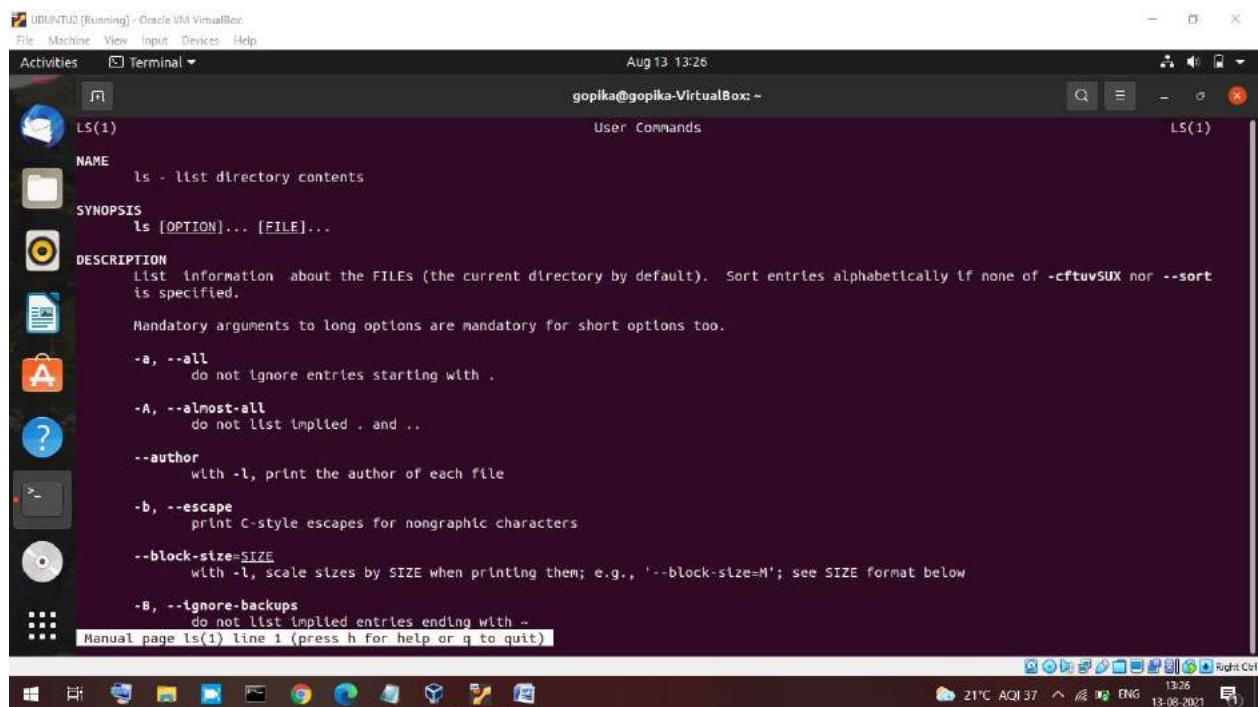
```
gopika@gopika-VirtualBox:~/Desktop$ mkdir file1
gopika@gopika-VirtualBox:~/Desktop$ ls
file1
gopika@gopika-VirtualBox:~/Desktop$ rmdir file1
gopika@gopika-VirtualBox:~/Desktop$ ls
gopika@gopika-VirtualBox:~/Desktop$
```

- **find**- find command can be used in a variety of conditions like you can find files by permissions, users, groups, file types, date, size, and other possible criteria.
- **cd** –used to change the current working directory

```
gopika@gopika-VirtualBox:~$ pwd
/home/gopika
gopika@gopika-VirtualBox:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
gopika@gopika-VirtualBox:~$ cd Desktop
gopika@gopika-VirtualBox:~/Desktop$
```

3.ls ,echo ,read

- **ls** – used to display the list of contents in a directory



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

- **echo** - echo command in linux is used to display line of text/string that are passed as an argument .
- **read** - read command in Linux system is used to read from a file descriptor

```
gopika@gopika-VirtualBox:~$ echo "what is your name?";read name; echo welcome $name!
what is your name?
Gopika
welcome Gopika!
gopika@gopika-VirtualBox:~$
```

4.touch ,cat, more ,less

- **touch** –used to create empty file ,it can also be used to create multiple empty files in a directory

```
gopika@gopika-VirtualBox:~/Desktop$ touch file1
gopika@gopika-VirtualBox:~/Desktop$ ls
file1
gopika@gopika-VirtualBox:~/Desktop$ touch file2 file3
gopika@gopika-VirtualBox:~/Desktop$ ls
file1  file2  file3
gopika@gopika-VirtualBox:~/Desktop$
```

- **cat** – it is a multipurpose utility command which can be used for create a new file, display contents in a file ,copy the content of one file to another etc.

The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "gopika@gopika-VirtualBox: ~/Desktop". The terminal content demonstrates the use of the cat, vim, and more/less commands:

```
gopika@gopika-VirtualBox:~/Desktop$ cat >nsd
^C
gopika@gopika-VirtualBox:~/Desktop$ ls
new nsd
gopika@gopika-VirtualBox:~/Desktop$ vim nsd
gopika@gopika-VirtualBox:~/Desktop$ cat nsd
line 1
line 2
line 3
line 4
hey
hello
hi

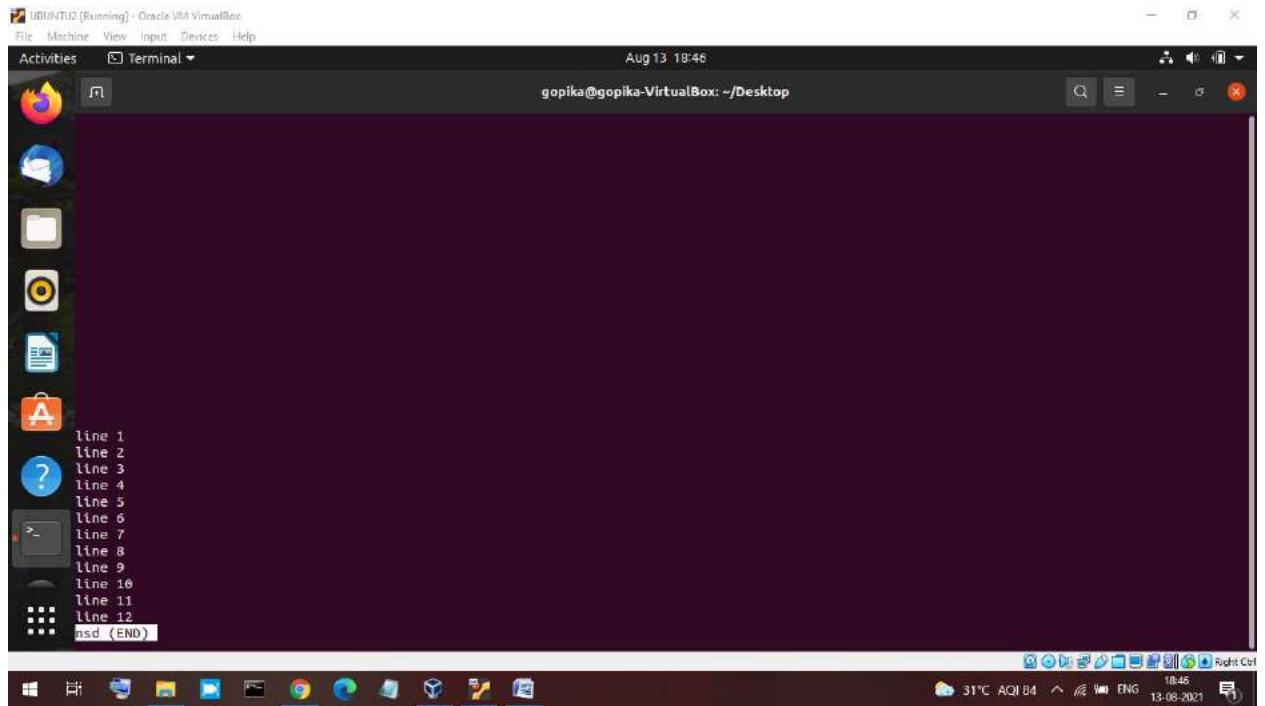
gopika@gopika-VirtualBox:~/Desktop$ vim new
gopika@gopika-VirtualBox:~/Desktop$ cat new nsd
hey i am gopika

gopika@gopika-VirtualBox:~/Desktop$ cat new nsd
hey i am gopika

gopika@gopika-VirtualBox:~/Desktop$ cat new
gopika@gopika-VirtualBox:~/Desktop$ cat nsd
```

- **more** - more command is used to view the text files in the command prompt, displaying one screen at a time in case the file is large (For example log files).
- **less**-Less command is a Linux utility that can be used to read the contents of a text file one page(one screen) at a time

```
gopika@gopika-VirtualBox:~/Desktop$ more nsd
line 1
line 2
line 3
line 4
line 5
line 6
line 7
line 8
line 9
line 10
line 11
line 12
gopika@gopika-VirtualBox:~/Desktop$ less nsd
gopika@gopika-VirtualBox:~/Desktop$ less nsd
```

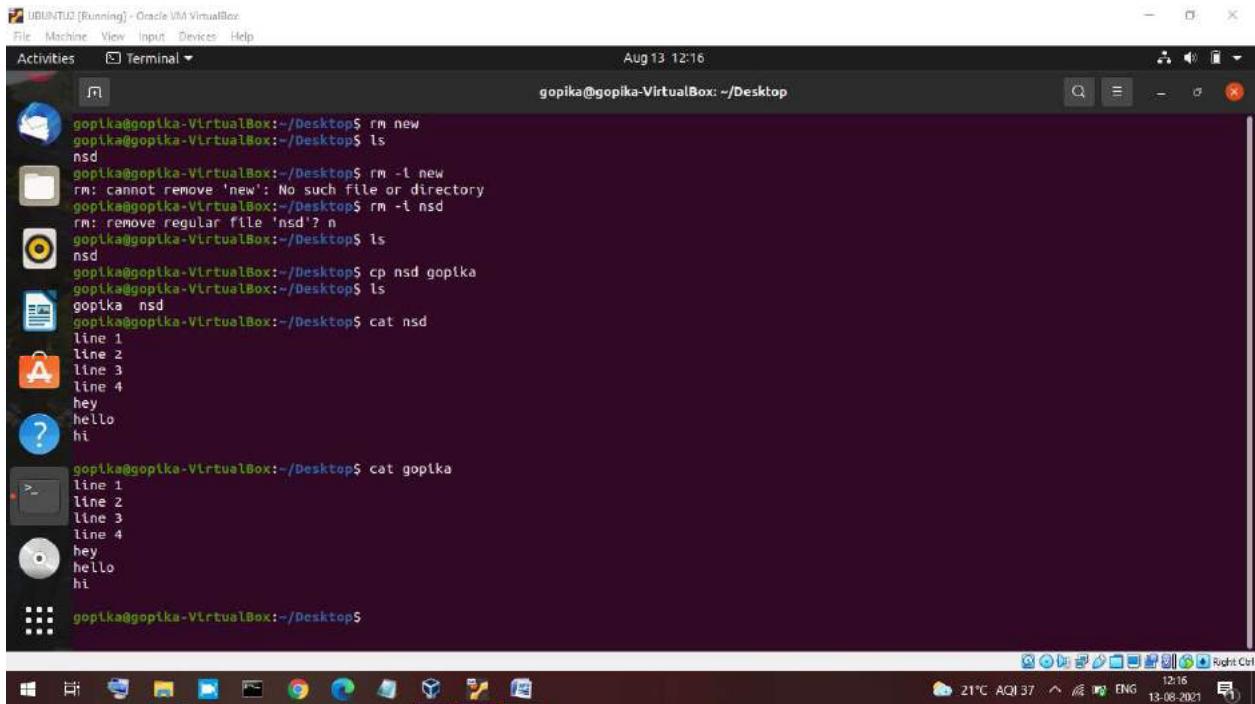


5. rm, cp, mv, tar

- **rm** –used to remove a file

```
gopika@gopika-VirtualBox:~/Desktop$ rm new
gopika@gopika-VirtualBox:~/Desktop$ ls
nsd
gopika@gopika-VirtualBox:~/Desktop$ rm -i new
rm: cannot remove 'new': No such file or directory
gopika@gopika-VirtualBox:~/Desktop$ rm -i nsd
rm: remove regular file 'nsd'? n
gopika@gopika-VirtualBox:~/Desktop$ ls
nsd
gopika@gopika-VirtualBox:~/Desktop$ █
```

- **cp** –used to copy a file or directory



- **mv** –used to move a file or a directory from one location to another

```

gopika@gopika-VirtualBox:~/Desktop$ ls
gopika
gopika@gopika-VirtualBox:~/Desktop$ cd ..
gopika@gopika-VirtualBox:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
gopika@gopika-VirtualBox:~$ cd Downloads
gopika@gopika-VirtualBox:~/Downloads$ ls
nsd
gopika@gopika-VirtualBox:~/Downloads$ mv nsd /home/gopika/Desktop
gopika@gopika-VirtualBox:~/Downloads$ ls
gopika@gopika-VirtualBox:~/Downloads$ cd ..
gopika@gopika-VirtualBox:~$ cd Desktop
gopika@gopika-VirtualBox:~/Desktop$ ls
gopika nsd
gopika@gopika-VirtualBox:~/Desktop$
```

- **rename**- used to rename files

rename <file name>

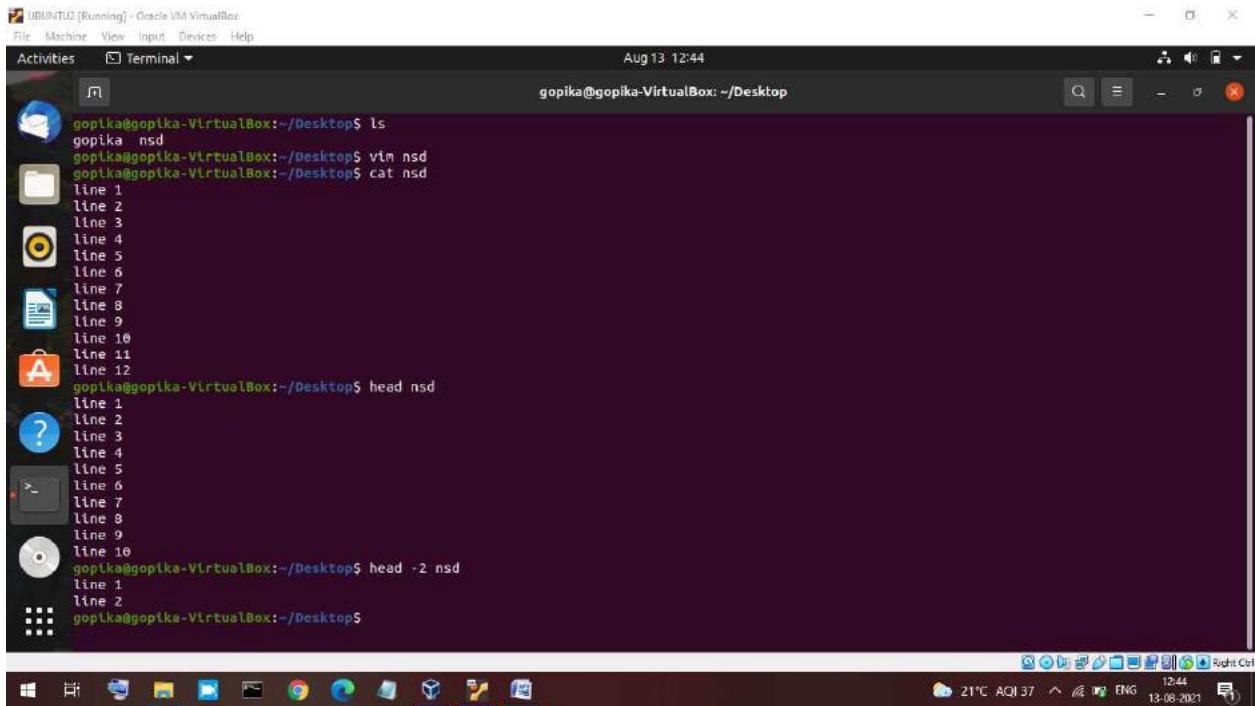
- **tar** – used to create compressed archive files and that can be moved from one disk to another

syntax:- tar {A|c|d|r|t|u|x}[GnSkUWOmpsMBiajJzZhPIRvwo] [ARG...]

6. head, tail, tac , grep

head – used to display the content of a file. By default it display the first 10 lines of the file , if we needed to display particular number of lines use

```
head -<no.of lines> <filename>
```



The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "Ubuntu [Running] - Oracle VM VirtualBox". The terminal content shows the following commands and their outputs:

```
gopiika@gopiika-VirtualBox:~/Desktop$ ls
gopiika  nsd
gopiika@gopiika-VirtualBox:~/Desktop$ vim nsd
gopiika@gopiika-VirtualBox:~/Desktop$ cat nsd
line 1
line 2
line 3
line 4
line 5
line 6
line 7
line 8
line 9
line 10
line 11
line 12
gopiika@gopiika-VirtualBox:~/Desktop$ head nsd
line 1
line 2
line 3
line 4
line 5
line 6
line 7
line 8
line 9
line 10
gopiika@gopiika-VirtualBox:~/Desktop$ head -2 nsd
line 1
line 2
gopiika@gopiika-VirtualBox:~/Desktop$
```

tail – used to display the last ten lines by default. To display specific number of lines

```
tail -<no of lines> <file name>
```

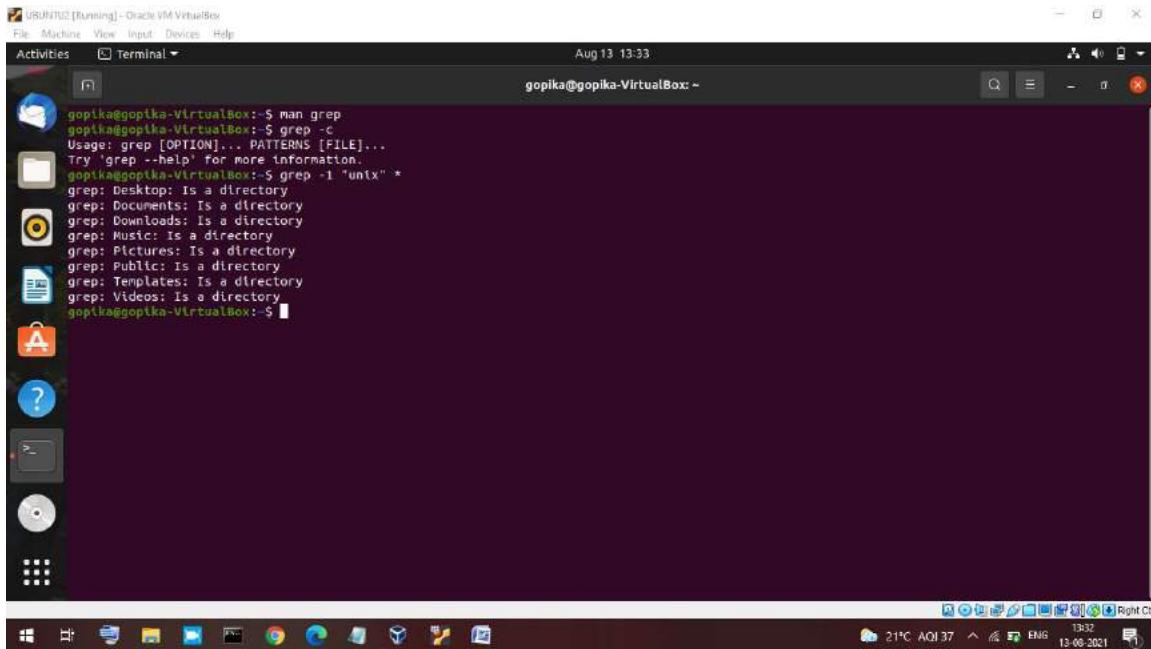
```
gopika@gopika-VirtualBox:~/Desktop$ ls
gopika  nsd
gopika@gopika-VirtualBox:~/Desktop$ tail nsd
line 3
line 4
line 5
line 6
line 7
line 8
line 9
line 10
line 11
line 12
gopika@gopika-VirtualBox:~/Desktop$ tail -4 nsd
line 9
line 10
line 11
line 12
gopika@gopika-VirtualBox:~/Desktop$
```

tac – reverse of cat command. It is used to display the file content in reverse order.

```
gopika@gopika-VirtualBox:~/Desktop$ ls
gopika  nsd
gopika@gopika-VirtualBox:~/Desktop$ tac nsd
line 12
line 11
line 10
line 9
line 8
line 7
line 6
line 5
line 4
line 3
line 2
line 1
```

grep - grep searches for PATTERNS in each FILE. PATTERNS is one or more patterns separated by newline characters, and grep prints each line that matches a pattern. Typically PATTERNS should be quoted when grep is used in a shell command.

Syntax:- grep [OPTION...] PATTERNS [FILE...]



A screenshot of a Linux desktop environment, likely Ubuntu, running in a virtual machine. The desktop has a dark theme with icons for various applications like Dash, Home, and System. A terminal window is open in the top panel, showing the output of a 'grep' command. The terminal window title is 'gopika@gopika-VirtualBox: ~'. The command run was 'grep -c "/unix" *' which lists all files containing the string '/unix' and their counts. The desktop taskbar at the bottom shows various application icons.

```
gopika@gopika-VirtualBox:~$ man grep
gopika@gopika-VirtualBox:~$ grep -c
Usage: grep [OPTION]... PATTERN [FILE]...
Try 'grep --help' for more information.
gopika@gopika-VirtualBox:~$ grep -c "/unix" *
grep: Desktop: Is a directory
grep: Documents: Is a directory
grep: Downloads: Is a directory
grep: Music: Is a directory
grep: Pictures: Is a directory
grep: Public: Is a directory
grep: Templates: Is a directory
grep: Videos: Is a directory
gopika@gopika-VirtualBox:~$
```

7. whatis , whereis

whereis – used to find the location of source/binary file of a command and manuals sections for a specified file.

```
gopika@gopika-VirtualBox:~/Desktop$ cd ..
gopika@gopika-VirtualBox:~$ whereis print
print: /usr/bin/print /usr/share/man/man1/print.1.gz
gopika@gopika-VirtualBox:~$
```

whatis – used to get a one line manual page descriptions.

```
gopika@gopika-VirtualBox:~$ whatis print
print (1)          - execute programs via entries in the mailcap file
gopika@gopika-VirtualBox:~$
```

8.wc,cut ,paste

wc- The wc (word count) command in Unix/Linux operating systems is used to find out number of newline count, word count, byte and characters count in a files specified by the file arguments.

```
gopika@gopika-VirtualBox:~/Desktop$ wc -w nsd
24 nsd
gopika@gopika-VirtualBox:~/Desktop$
```

Cut- cut command is useful for selecting a specific column of a file. It is used to cut a specific sections by byte position, character, and field and writes them to standard output. It cuts a line and extracts the text data.

Paste - cut is a command-line utility that allows you to cut parts of lines from specified files or piped data and print the result to standard output.

9.chmod ,chown

chmod - The chmod command allows to change the permissions on a file using either a symbolic or numeric mode or a reference file.

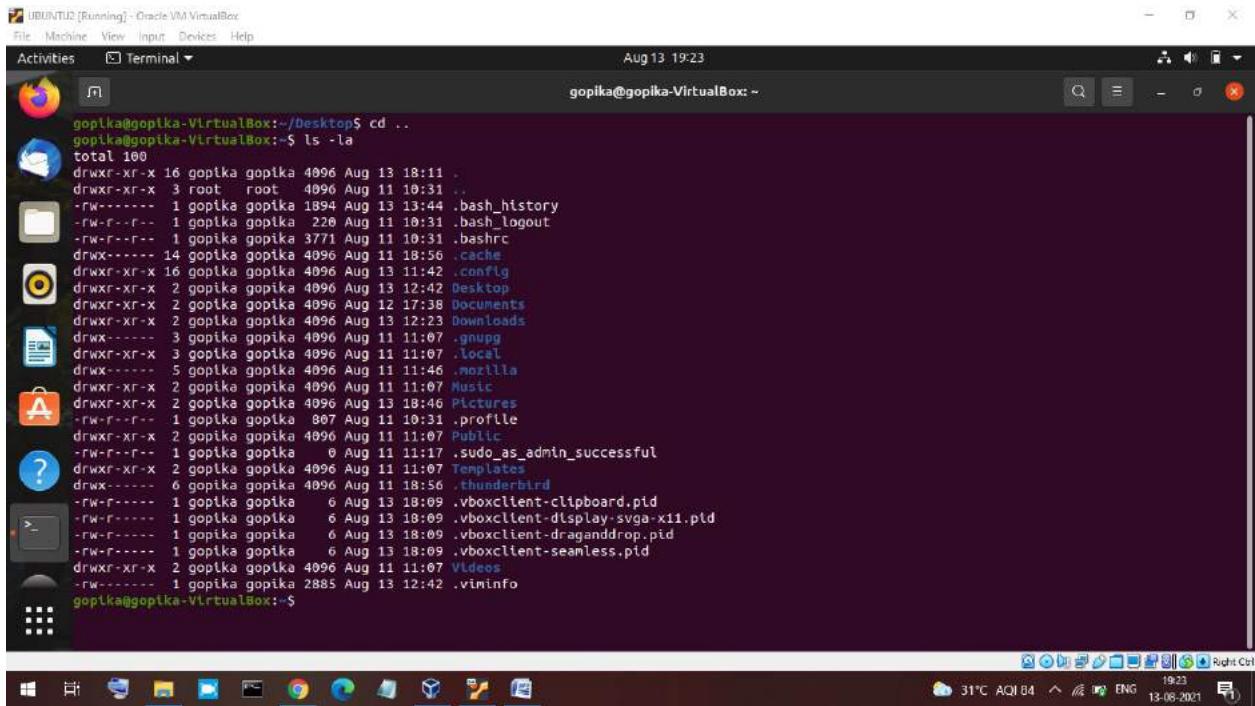
chown - Linux chown command is used to change a file's ownership, directory, or symbolic link for a user or group.

10. redirections and piping

Redirection is a feature in Linux such that when executing a command, you can change the standard input/output devices. The basic workflow of any Linux command is that it takes an input and give an output.

- The standard input (stdin) device is the keyboard.
- The standard output (stdout) device is the screen.

Piping-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 like a pipeline. The symbol '|' denotes a pipe.



```
UBUNTU2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Aug 13 19:23
gopika@gopika-VirtualBox:~$ cd ..
gopika@gopika-VirtualBox:~$ ls -la
total 100
drwxr-xr-x 16 gopika gopika 4096 Aug 13 18:11 .
drwxr-xr-x  3 root  root  4096 Aug 11 10:31 ..
-rw-r--r--  1 gopika gopika 1894 Aug 13 13:44 .bash_history
-rw-r--r--  1 gopika gopika 220 Aug 11 10:31 .bash_logout
-rw-r--r--  1 gopika gopika 3771 Aug 11 10:31 .bashrc
drwxr----- 14 gopika gopika 4096 Aug 11 18:56 .cache
drwxr-xr-x 16 gopika gopika 4096 Aug 13 11:42 .config
drwxr-xr-x  2 gopika gopika 4096 Aug 13 12:42 Desktop
drwxr-xr-x  2 gopika gopika 4096 Aug 12 17:38 Documents
drwxr-xr-x  2 gopika gopika 4096 Aug 13 12:23 Downloads
drwxr-----  3 gopika gopika 4096 Aug 11 11:07 .gnupg
drwxr-xr-x  3 gopika gopika 4096 Aug 11 11:07 .local
drwxr-----  5 gopika gopika 4096 Aug 11 11:46 .mozilla
drwxr-xr-x  2 gopika gopika 4096 Aug 11 11:07 Music
drwxr-xr-x  2 gopika gopika 4096 Aug 13 18:46 Pictures
-rw-r--r--  1 gopika gopika 807 Aug 11 10:31 .profile
drwxr-xr-x  2 gopika gopika 4096 Aug 11 11:07 Public
-rw-r--r--  1 gopika gopika  0 Aug 11 11:17 .sudo_as_admin_successful
drwxr-xr-x  2 gopika gopika 4096 Aug 11 11:07 Templates
drwxr-----  6 gopika gopika 4096 Aug 11 18:56 .thunderbird
-rw-r-----  1 gopika gopika  6 Aug 13 18:09 .vboxclient-clipboard.pid
-rw-r-----  1 gopika gopika  6 Aug 13 18:09 .vboxclient-display-svga-x11.pid
-rw-r-----  1 gopika gopika  6 Aug 13 18:09 .vboxclient-draganddrop.pid
-rw-r-----  1 gopika gopika  6 Aug 13 18:09 .vboxclient-seamless.pid
drwxr-xr-x  2 gopika gopika 4096 Aug 11 11:07 Videos
-rw-r-----  1 gopika gopika 2885 Aug 13 12:42 .viminfo
gopika@gopika-VirtualBox:~$
```

11.useradd ,usermod ,userdel ,passwd

- **useradd** command is used to create new accounts in Linux
- **usermod** command used to modify the existing accounts in linux
- **userdel** command is used to delete local account in linux
- **passwd** command used assign password to local accounts or users.

12. df,top ,ps

Df - This utility reports the free disk space(Hard Disk) on all the file systems.

```

gopika@gopika-VirtualBox:~$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
udev              537368       0   537368  0% /dev
tmpfs             113288   1328   111960  2% /run
/dev/sda5        9736500  8815984  406212  96% /
tmpfs              566424       0   566424  0% /dev/shm
tmpfs               5120       4    5116  1% /run/lock
tmpfs              566424       0   566424  0% /sys/fs/cgroup
/dev/Loop1         224256       0   224256  0 100% /snap/gnome-3-34-1804/66
/dev/Loop0         56832       0   56832  0 100% /snap/core18/1988
/dev/Loop2         56832       0   56832  0 100% /snap/core18/2128
/dev/Loop3         224256       0   224256  0 100% /snap/gnome-3-34-1804/72
/dev/Loop5         66688       0   66688  0 100% /snap/gtk-common-themes/1515
/dev/Loop6         52352       0   52352  0 100% /snap/snap-store/518
/dev/Loop7         31872       0   31872  0 100% /snap/snapd/11836
/dev/Loop4         66432       0   66432  0 100% /snap/gtk-common-themes/1514
/dev/loop9          33152       0   33152  0 100% /snap/snapd/12704
/dev/Loop8          52224       0   52224  0 100% /snap/snap-store/547
/dev/sda1         523248       4   523244  1% /boot/efi
New_folder     439742588 70124156 369618432 10% /media/sf_New_folder
VM            439742588 70124156 369618432 10% /media/sf_VM
tmpfs             113284      20   113264  1% /run/user/1000
/dev/sr0            59642   59642       0 100% /media/gopika/VBox_GAs_6.1.26
gopika@gopika-VirtualBox:~$
```

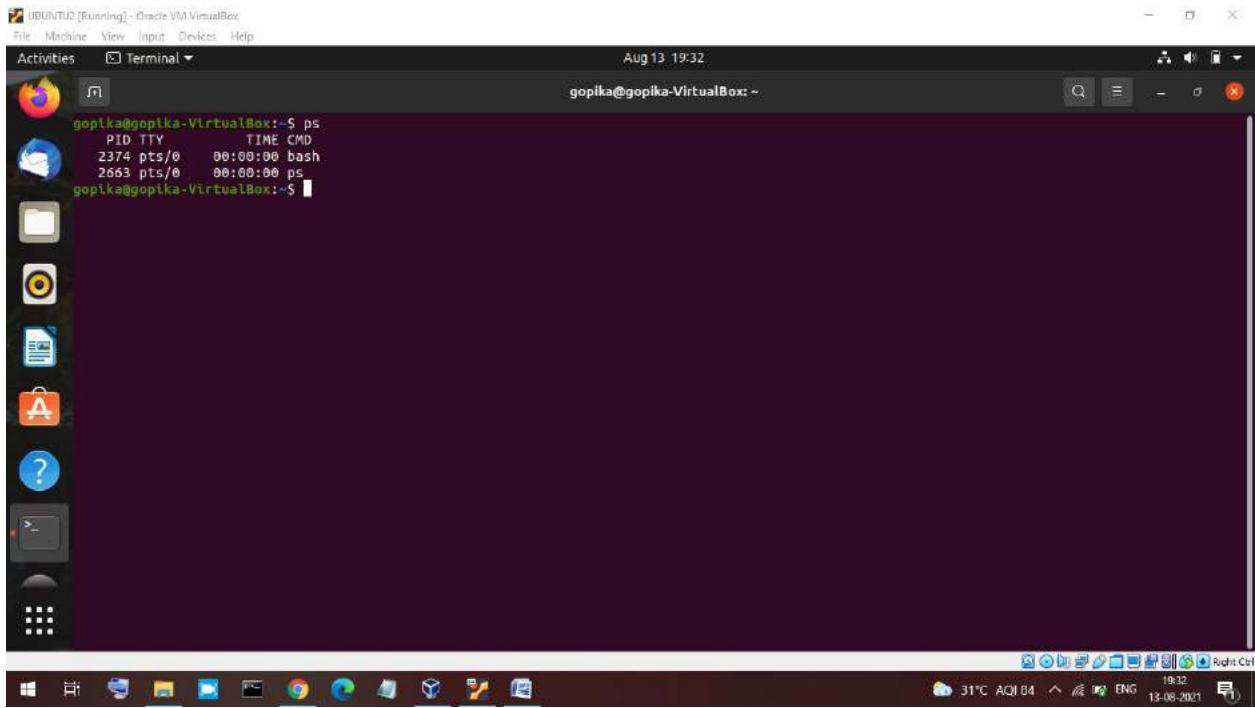
top Details on all Active Processes

```

top - 19:31:32 up 1:25, 1 user, load average: 0.31, 0.11, 0.04
Tasks: 191 total, 1 running, 189 sleeping, 0 stopped, 1 zombie
%Cpu(s): 3.3 us, 2.0 sy, 0.0 nl, 94.7 ld, 0.0 wa, 0.0 hi, 0.0 sl, 0.0 st
MiB Mem : 1106.3 total, 73.0 free, 705.6 used, 327.7 buff/cache
MiB Swap : 448.5 total, 212.3 free, 236.2 used. 259.2 avail Mem

PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM TIME+ COMMAND
2364 gopika  20   0  823240 50976 37552 S  1.6  4.4  0:09.95 gnome-terminal-
2659 gopika  20   0  20488 3900  3256 R  1.6  0.3  0:00.26 top
1478 gopika  20   0 267864 33064 12920 S  1.0  2.9  0:14.07 Xorg
1715 gopika  20   0 3606464 197260 67472 S  0.6 17.4  0:49.73 gnome-shell
235 root     20   0      0      0      0 I  0.3  0.0  0:07.97 kworker/0:3-events
796 mysql    20   0 1301584 190584 3704 S  0.3 16.8  0:25.82 mysqld
1645 gopika  20   0 163996 2424 2188 S  0.3  0.2  0:21.33 VBoxClient
1 root      20   0 315344 9252 6720 S  0.0  0.8  0:05.56 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_gp
6 root      0 -20      0      0      0 I  0.0  0.0  0:00.00 kworker/0:0H-events_highpri
7 root      20   0      0      0      0 I  0.0  0.0  0:00.02 kworker/0:1-events
9 root      0 -20      0      0      0 I  0.0  0.0  0:00.00 mm_percpu_wq
10 root     20   0      0      0      0 S  0.0  0.0  0:00.00 rcu_tasks_rude_
11 root     20   0      0      0      0 S  0.0  0.0  0:00.00 rcu_tasks_trace
12 root     20   0      0      0      0 S  0.0  0.0  0:00.39 ksoftirqd/0
13 root     20   0      0      0      0 I  0.0  0.0  0:02.88 rcu_sched
14 root     rt  0      0      0      0 S  0.0  0.0  0:00.07 migration/0
15 root     -51  0      0      0      0 S  0.0  0.0  0:00.00 idle_inject/0
16 root     20   0      0      0      0 S  0.0  0.0  0:00.00 cpuhp/0
17 root     20   0      0      0      0 S  0.0  0.0  0:00.00 kdevtmpfs
18 root     0 -20      0      0      0 I  0.0  0.0  0:00.00 netns
19 root     0 -20      0      0      0 I  0.0  0.0  0:00.00 inet_frag_wq
20 root     20   0      0      0      0 S  0.0  0.0  0:00.00 kauditd
```

ps Give the status of processes running for a user



13.ssh ,scp , ssh-keygen , ssh-copy-id

ssh-The ssh command provides a secure encrypted connection between two hosts over an insecure network. This connection can also be used for terminal access, file transfers, and for tunneling other applications.

scp -scp (secure copy) command in Linux system is used to copy file(s) between servers in a secure way. The SCP command or secure copy allows secure transferring of files in between the local host and the remote host or between two remote hosts

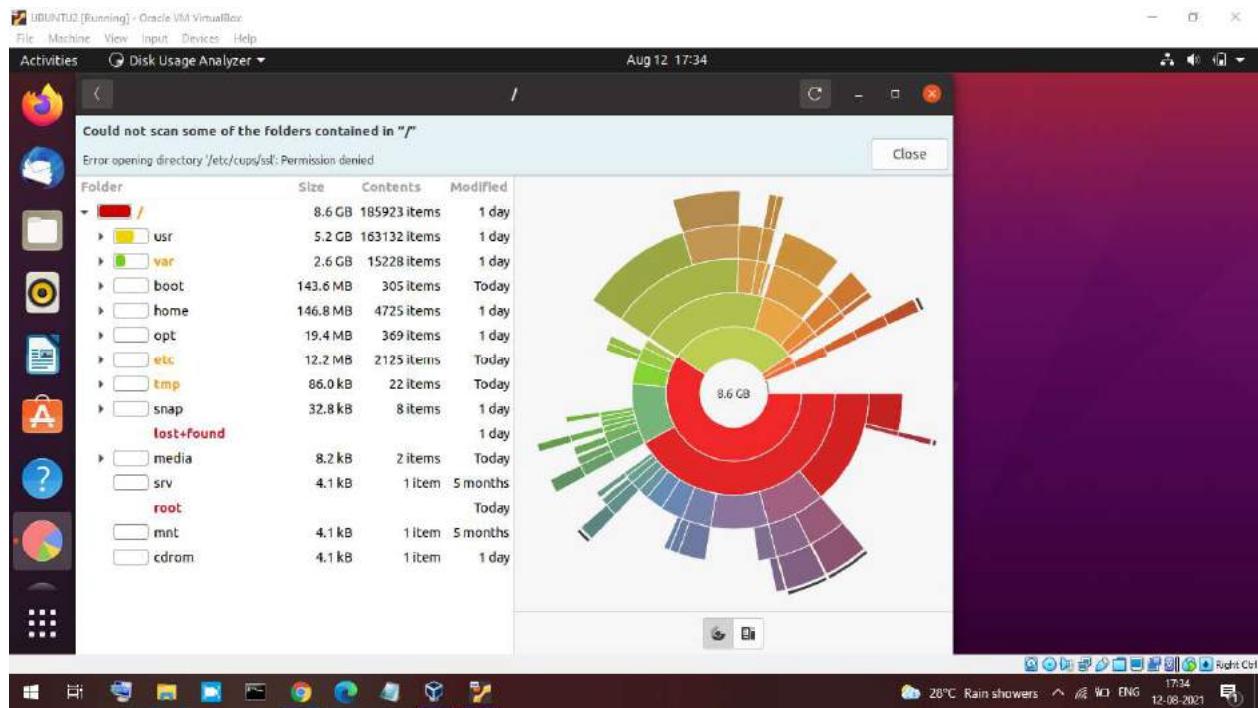
Ssh-keygen is a tool for creating new authentication key pairs for SSH. Such key pairs are used for automating logins, single sign-on, and for authenticating hosts.

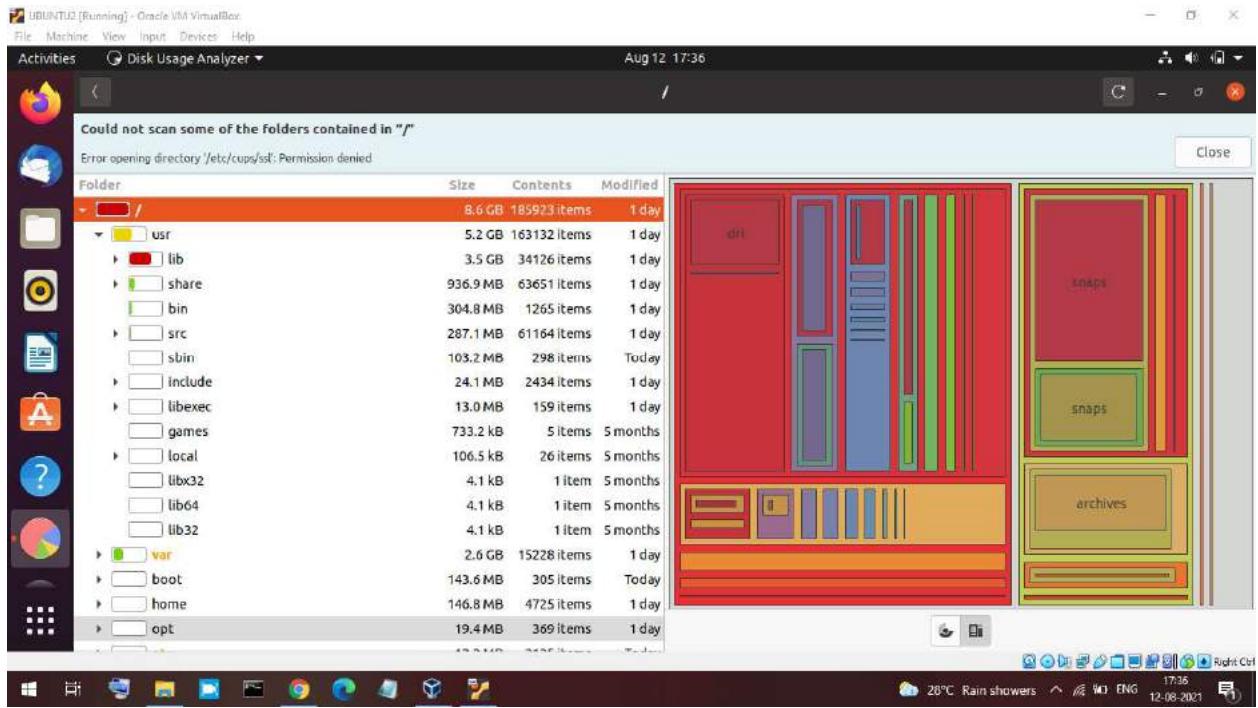
ssh-copy-id copies the local-host's public key to the remote-host's authorized_keys file. ssh-copy-id also assigns proper permission to the remote-host's home, ~/.ssh, and ~/.ssh/authorized_keys.

EXPERIMENT NO :-3

AIM:- FILE SYSTEM HEIRARCHY & FILE PERMISSIONS IN LINUX

LINUX DIRECTORY STRUCTURE (File System Structure)





- **/bin** : All the executable binary programs (file) required during booting, repairing, files required to run into single-user-mode, and other important, basic commands viz., cat, du, df, tar, rpm, wc, history, etc.
- **/boot** : Holds important files during boot-up process, including Linux Kernel.
- **/dev** : Contains device files for all the hardware devices on the machine e.g., cdrom, cpu, etc
- **/etc** : Contains Application's configuration files, startup, shutdown, start, stop script for every individual program.
- **/home** : Home directory of the users. Every time a new user is created, a directory in the name of user is created within home directory which contains other directories like Desktop, Downloads, Documents, etc.
- **/lib** : The Lib directory contains kernel modules and shared library images required to boot the system and run commands in root file system.
- **/lost+found** : This Directory is installed during installation of Linux, useful for recovering files which may be broken due to unexpected shut-down.

- **/media** : Temporary mount directory is created for removable devices viz., media/cdrom.
- **/mnt** : Temporary mount directory for mounting file system.
- **/opt** : Optional is abbreviated as opt. Contains third party application software. Viz., Java, etc.
- **/proc** : A virtual and pseudo file-system which contains information about running process with a particular Process-id aka pid.
- **/root** : This is the home directory of root user and should never be confused with ‘/’
- **/run** : This directory is the only clean solution for early-runtime-dir problem.
- **/sbin** : Contains binary executable programs, required by System Administrator, for Maintenance. Viz., iptables, fdisk, ifconfig, swapon, reboot, etc.
- **/srv** : Service is abbreviated as ‘srv’. This directory contains server specific and service related files.
- **/sys** : Modern Linux distributions include a /sys directory as a virtual filesystem, which stores and allows modification of the devices connected to the system.
- **/tmp** : System’s Temporary Directory, Accessible by users and root. Stores temporary files for user and system, till next boot.
- **/usr** : Contains executable binaries, documentation, source code, libraries for second level program.
- **/var** : Stands for variable. The contents of this file is expected to grow. This directory contains log, lock, spool, mail and temp files.

FILE PERMISSIONS IN LINUX

Linux file permissions, attributes, and ownership control the access level that the system processes and users have to files. This ensures that only authorized users and processes can access specific files and directories.

The basic Linux permissions model works by associating each system file with an owner and a group and assigning permission access rights for three different classes of users:

- The file owner.
- The group members.
- Others (everybody else)

User/Owner

A user is the owner of the file. By default, the person who created a file becomes its owner. Hence, a user is also sometimes called an owner.

Group

A user-group can contain multiple users. All users belonging to a group will have the same Linux group permissions access to the file. Suppose you have a project where a number of people require access to a file. Instead of manually assigning permissions to each user, you could add all users to a group, and assign group permission to file such that only this group members and no one else can read or modify the files.

Other

Any other user who has access to a file. This person has neither created the file, nor he belongs to a usergroup who could own the file. Practically, it means everybody else. Hence, when you set the permission for others, it is also referred as set permissions for the world.

Permissions

Every file and directory in your UNIX/Linux system has following 3 permissions defined for all the 3 owners discussed above.

- **Read:** This permission give you the authority to open and read a file. Read permission on a directory gives you the ability to lists its content.

- **Write:** The write permission gives you the authority to modify the contents of a file. The write permission on a directory gives you the authority to add, remove and rename files stored in the directory. Consider a scenario where you have write permission on a file but do not have write permission on the directory where the file is stored. You will be able to modify the file contents. But you will not be able to rename, move or remove the file from the directory.
- **Execute:** In Windows, an executable program usually has an extension ".exe" and which you can easily run. In Unix/Linux, you cannot run a program unless the execute permission is set. If the execute permission is not set, you might still be able to see/modify the program code(provided read & write permissions are set), but not run it.

Changing file/directory permissions with 'chmod' command

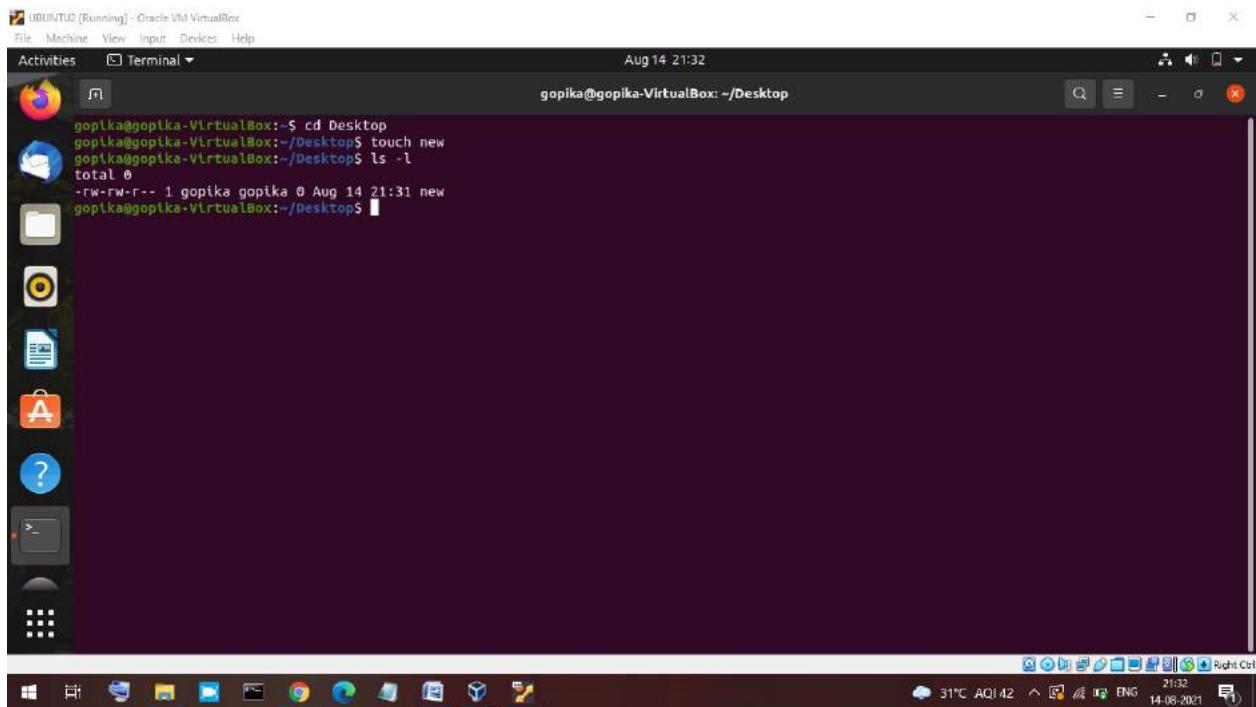
Symbolic Mode

In the Absolute mode, you change permissions for all 3 owners. In the symbolic mode, you can modify permissions of a specific owner. It makes use of mathematical symbols to modify the Unix file permissions.

| OPERATOR | DESCRIPTION |
|----------|--|
| + | Adds a permission to a file or directory |
| - | Removes the permission |
| = | Sets the permission and overrides the permission set earlier |

The various owners are represented as –

| User Denotations | |
|------------------|-------------|
| u | User /owner |
| g | group |
| o | others |
| a | all |



A screenshot of a Linux desktop environment, specifically Ubuntu 22.04 LTS, running in Oracle VM VirtualBox. The desktop has a dark theme. On the left is a vertical dock with icons for the Dash, Home, Activities, Terminal, Dash, Home, Activities, Terminal, and Help. The main window is a terminal window titled 'Terminal' with the command line 'gopika@gopika-VirtualBox: ~/Desktop'. The terminal shows the following session:

```
gopika@gopika-VirtualBox:~$ cd Desktop
gopika@gopika-VirtualBox:~/Desktop$ touch new
gopika@gopika-VirtualBox:~/Desktop$ ls -l
total 0
-rw-rw-r-- 1 gopika gopika 0 Aug 14 21:31 new
gopika@gopika-VirtualBox:~/Desktop$
```

The desktop bar at the bottom shows various application icons and system status indicators like battery level, temperature (31°C), and network.

The permissions are broken into groups of threes, and each position in the group denotes a specific permission, in this order: read (r), write (w), execute (x) –

- The first three characters (2-4) represent the permissions for the file's owner. For example, **-rw-rw-r--** represents that the owner has read (r) and write (w) permission, but no execute permission.

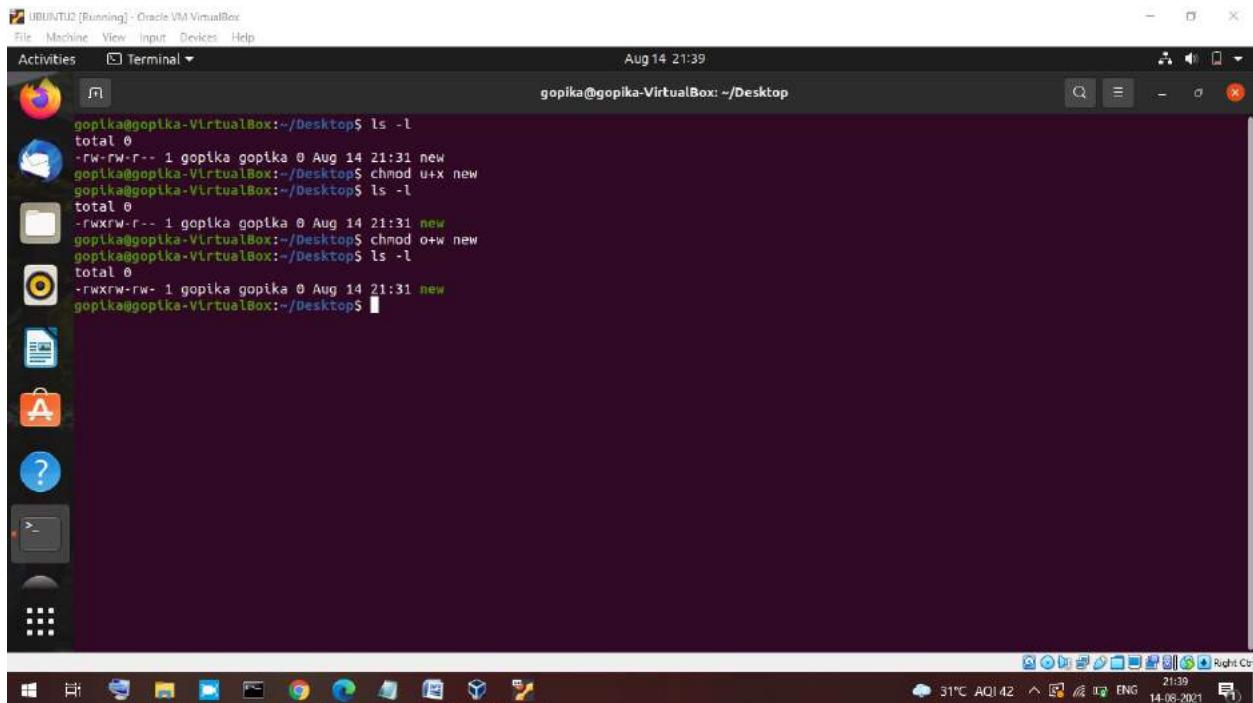
- The second group of three characters (5-7) consists of the permissions for the group to which the file belongs. For example, **-rw-rw-r--** represents that the group has read (r) and write(w) permission but no execute (x) permission.
- The last group of three characters (8-10) represents the permissions for everyone else. For example, **-rw-rw-r--** represents that there is **read (r)** only permission.

To change directory permissions in Linux, use the following:

- **chmod + rwx filename** to add permissions.
- **chmod - rwx directoryname** to remove permissions.
- **chmod + x filename** to allow executable permissions.
- **chmod - wx filename** to take out write and executable permissions.

Here “r” is for read, “w” is for write, and “x” is for execute.

This only changes the permissions for the owner of the file.



The screenshot shows a terminal window on an Ubuntu 12.04 LTS desktop. The terminal output is as follows:

```
gopika@gopika-VirtualBox:~/Desktop$ ls -l
total 0
-rw-rw-r-- 1 gopika gopika 0 Aug 14 21:31 new
gopika@gopika-VirtualBox:~/Desktop$ chmod u+x new
gopika@gopika-VirtualBox:~/Desktop$ ls -l
total 0
-rwxrwxr-- 1 gopika gopika 0 Aug 14 21:31 new
gopika@gopika-VirtualBox:~/Desktop$ chmod o-w new
gopika@gopika-VirtualBox:~/Desktop$ ls -l
total 0
-rwxrwxrw- 1 gopika gopika 0 Aug 14 21:31 new
gopika@gopika-VirtualBox:~/Desktop$
```

The desktop environment includes a dock with icons for Dash, Home, Applications, and others, and a system tray at the bottom showing network, battery, and system status.

-

rwx

"-" indicates a file
"d" indicates directory
"l" indicates a link

Read, write, and execute permissions for the owner of the file

r--

Read, write, and execute permissions for members of the group owning the file

r--

Read, write, and execute permissions for other users

Numeric Mode

Permission numbers are:

- **0 = ---** no permission
- **1 = --x** execute permission only
- **2 = -w-** permission to write only
- **3 = -wx** permission for write and execute
- **4 = r-** permission for read
- **5 = r-x** permission for read and execute
- **6 = rw-** permission for read and write
- **7 = rwx** permission for read ,write and execute

```
UBUNTU2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Aug 15 15:06
gopika@gopika-VirtualBox: ~/Desktop$ ls -l
total 0
--w-rwxrwx 1 gopika gopika 0 Aug 14 22:07 fileper
-rwxr-wr-- 1 gopika gopika 0 Aug 14 22:05 new
gopika@gopika-VirtualBox: ~/Desktop$ chmod 777 new
gopika@gopika-VirtualBox: ~/Desktop$ ls -l
total 0
--w-rwxrwx 1 gopika gopika 0 Aug 14 22:07 fileper
-rwxrwxrwx 1 gopika gopika 0 Aug 14 22:05 new
gopika@gopika-VirtualBox: ~/Desktop$ chmod 327 fileper
gopika@gopika-VirtualBox: ~/Desktop$ ls -l
total 0
--wx-W-rwx 1 gopika gopika 0 Aug 14 22:07 fileper
-rwxrwxrwx 1 gopika gopika 0 Aug 14 22:05 new
gopika@gopika-VirtualBox: ~/Desktop$
```

CHANGING FILE OWNERSHIPS

The chown command allows us to change the user and/or group ownership of a given file, directory, or symbolic link.

In Linux, all files are associated with an owner and a group and assigned with permission access rights for the file owner, the group members, and others.

These commands will give ownership to someone, but all sub files and directories still belong to the original owner.

Chown <name> <file name>

Or

chown [OPTIONS] USER[:GROUP] FILE(s)

USER is the user name or the user ID (UID) of the new owner. GROUP is the name of the new group or the group ID (GID). FILE(s) is the name of one or more files, directories or links. Numeric IDs should be prefixed with the + symbol.

- USER - If only the user is specified, the specified user will become the owner of the given files, the group ownership is not changed.
- USER: - When the username is followed by a colon :, and the group name is not given, the user will become the owner of the files, and the files group ownership is changed to user's login group.
- USER:GROUP - If both the user and the group are specified (with no space between them), the user ownership of the files is changed to the given user and the group ownership is changed to the given group.
- :GROUP - If the User is omitted and the group is prefixed with a colon :, only the group ownership of the files is changed to the given group.
- : If only a colon : is given, without specifying the user and the group, no change is made.

By default, on success, chown doesn't produce any output and returns zero.

Superuser permissions are necessary to execute the chown command.

EXPERIMENT NO:4

AIM :: SHELL SCRIPTING

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

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

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

SHELL SCRIPTING

A shell script is a text file that contains a sequence of commands for a UNIX-based operating system. It is called a shell script because it combines a sequence of commands , that would otherwise have to be typed into the keyboard one at a time, into a single script. The shell is the operating system's command-line interface (CLI) and interpreter for the set of commands that are used to communicate with the system.

A shell script is usually created for command sequences in which a user has a need to use repeatedly in order to save time. Like other programs, the shell script can contain parameters, comments and subcommands that the shell must follow. Users initiate the sequence of commands in the shell script by simply entering the file name on a command line.

In the DOS operating system, a shell script is called a batch file. In IBM's mainframe VM operating systems, it's called an EXEC.

BASH (Bourne Again SHell), which is the most widely used shell and also the default in most Linux systems, which most servers use as their operating system. Shell scripting is particularly useful for DevOps and backend development.

There are two major types of shells –

- **Bourne shell** – If you are using a Bourne-type shell, the \$ character is the default prompt.
- **C shell** – If you are using a C-type shell, the % character is the default prompt.

The basic steps involved with shell scripting are writing the script, making the script accessible to the shell and giving the shell execute permission.

Shell scripts contain ASCII text and are written using a text editor, word processor or graphical user interface (GUI). The content of the script is a series of commands in a language that can be interpreted by the shell. Functions that shell scripts support include loops, variables, if/then/else statements, arrays and shortcuts. Once complete, the file is saved typically with a .txt or .sh extension and in a location that the shell can access.

ADVANTAGES

- Easy to use.

- Quick start, and interactive debugging.
- Time Saving.
- Sys Admin task automation.(system admin)
- Shell scripts can execute without any additional effort on nearly any modern UNIX / Linux / BSD / Mac OS X operating system as they are written an interpreted language.

DISADVANTAGES

- Compatibility problems between different platforms.
- Slow execution speed.
- A new process launched for almost every shell command executed.

Shell Environment

A shell maintains an environment that includes a set of variables defined by the login program, the system initialization file, and the user initialization files. In addition, some variables are defined by default.

A shell can have two types of variables:

- **Environment variables** – Variables that are exported to all processes spawned by the shell. Their settings can be seen with the env command. A subset of environment variables, such as PATH, affects the behavior of the shell itself.
- **Shell (local) variables** – Variables that affect only the current shell. In the C shell, a set of these shell variables have a special

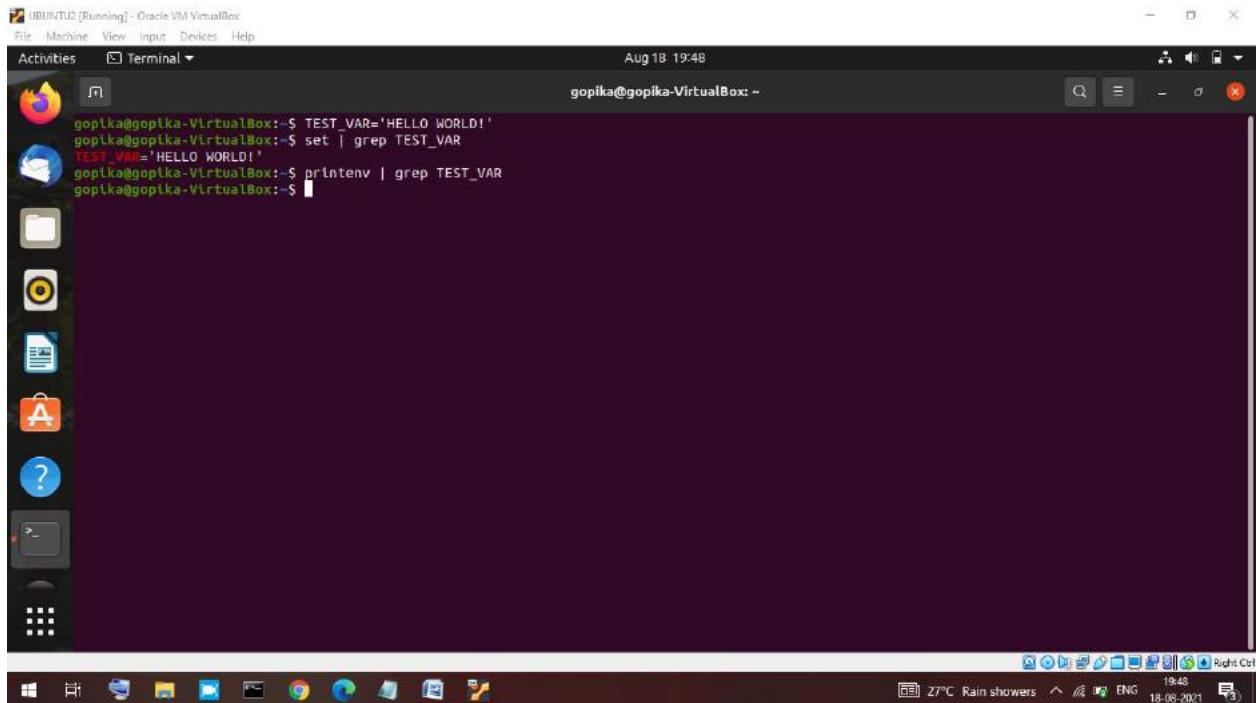
relationship to a corresponding set of environment variables. These shell variables are user, term, home, and path. The value of the environment variable counterpart is initially used to set the shell variable.

Common Environmental and Shell Variables

- SHELL: This describes the shell that will be interpreting any commands you type in. In most cases, this will be bash by default, but other values can be set if you prefer other options.
- TERM: This specifies the type of terminal to emulate when running the shell. Different hardware terminals can be emulated for different operating requirements. You usually won't need to worry about this though.
- USER: The current logged in user.
- PWD: The current working directory.
- OLDPWD: The previous working directory. This is kept by the shell in order to switch back to your previous directory by running cd -.
- LS_COLORS: This defines color codes that are used to optionally add colored output to the ls command. This is used to distinguish different file types and provide more info to the user at a glance.
- MAIL: The path to the current user's mailbox.
- PATH: A list of directories that the system will check when looking for commands. When a user types in a command, the system will check directories in this order for the executable.
- LANG: The current language and localization settings, including character encoding.

- HOME: The current user's home directory.
- _: The most recent previously executed command

CREATING SHELL AND ENVIRONMENTAL VARIABLES

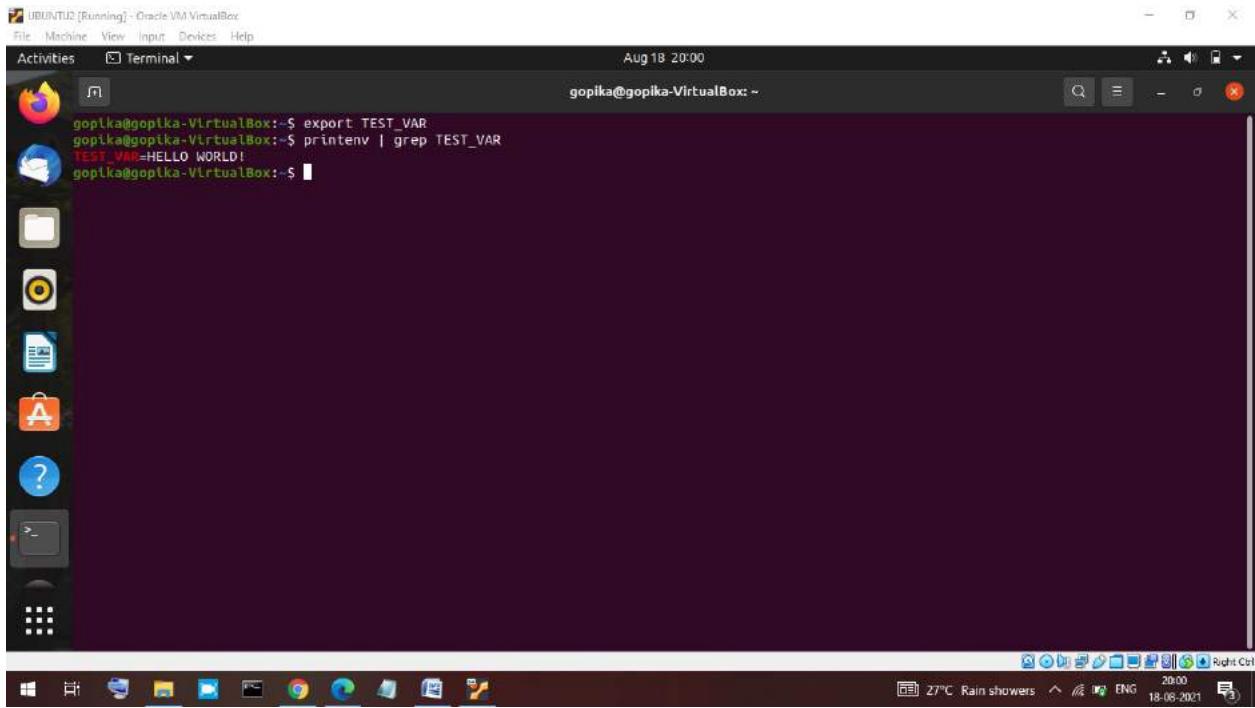


The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "gopika@gopika-VirtualBox: ~". The terminal content shows the following commands and output:

```
gopika@gopika-VirtualBox:~$ TEST_VAR='HELLO WORLD'
gopika@gopika-VirtualBox:~$ set | grep TEST_VAR
TEST_VAR='HELLO WORLD'
gopika@gopika-VirtualBox:~$ printenv | grep TEST_VAR
gopika@gopika-VirtualBox:~$
```

The desktop interface includes a dock with various icons (File Manager, Terminal, etc.), a system tray at the bottom, and a top panel with system status indicators.

Since we are not setted the environment variable so the “printenv” statement willnot return any output .



VARIABLES

Shell variables provide us the ability to store and manipulate information within a shell program. Variables are used to change the flow of the program and to maintain the state.

Types of Variables

1) Environment variables:

These are the variables that are visible to the child processes of a shell program. These include special environment variables that are set by the shell and are required for the shell to function properly.

Example:

- \$PATH – The set of paths to search for commands.

- \$HOME – The path to the current user's home folder.

2) Predefined variables:

When running commands, the shell expands wildcards, and then assigns the arguments to these predefined variables or ‘positional variables’.

Example:

- \$0 – The name of the command being executed.
- \$1 ... \$9 – The first to ninth arguments.

3) User-defined variables:

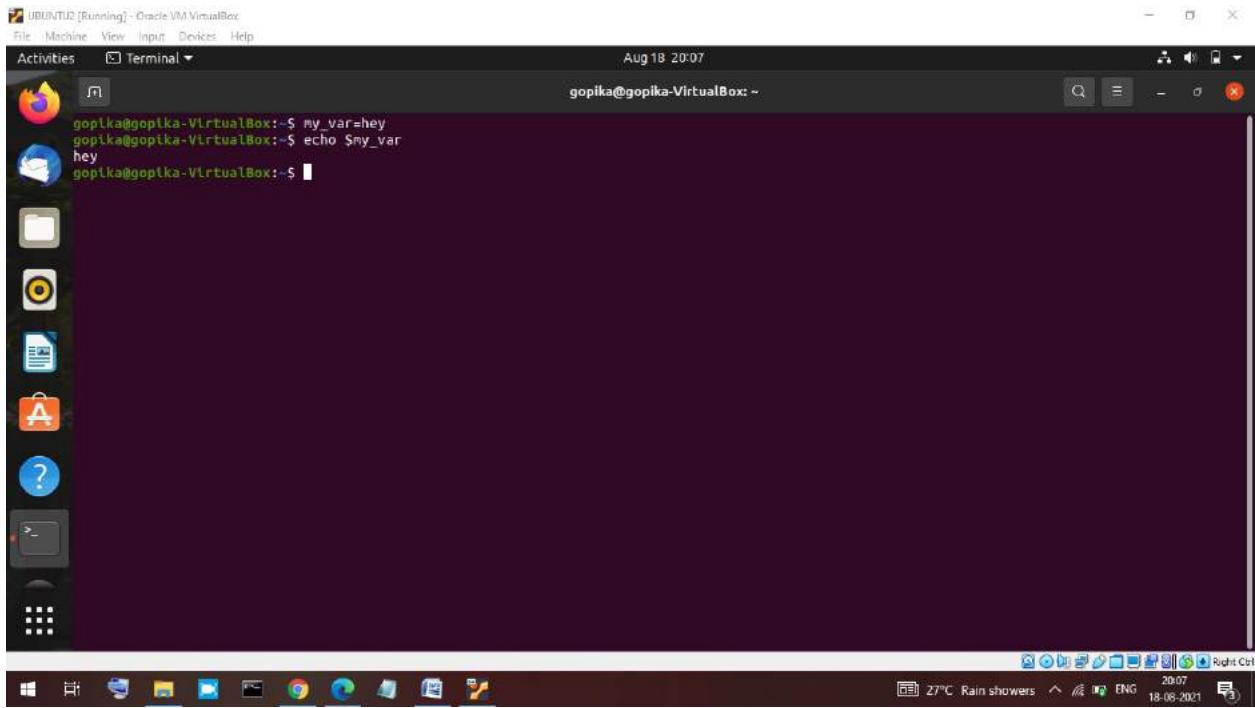
These are the variables that are visible to the current instance of the shell. The ‘export’ command is used to expose local variables to the environment variables.

DEFINING VARIABLE

A variable is defined by simply assigning a value to a name using the ‘=’ operator. A variable name is a series of alphanumeric characters starting with a letter or ‘_’. Variables are all treated as text strings unless the context requires them to be treated as a numeric value.

ACCESSING A VARIABLE

A variable name is de-referenced by simply adding a ‘\$’ prefix to it. The ‘echo’ command is often used to print the value of a variable.



CONTROL STRUCTURES

The shell processes the commands in a script sequentially, one after another in the order they are written in the file. Often, however, we will need to change the way that commands are processed. You may want to choose to run one command or another, depending on the circumstances; or you may want to run a command more than once. To alter the normal sequential execution of commands, the shell offers a variety of control structure.

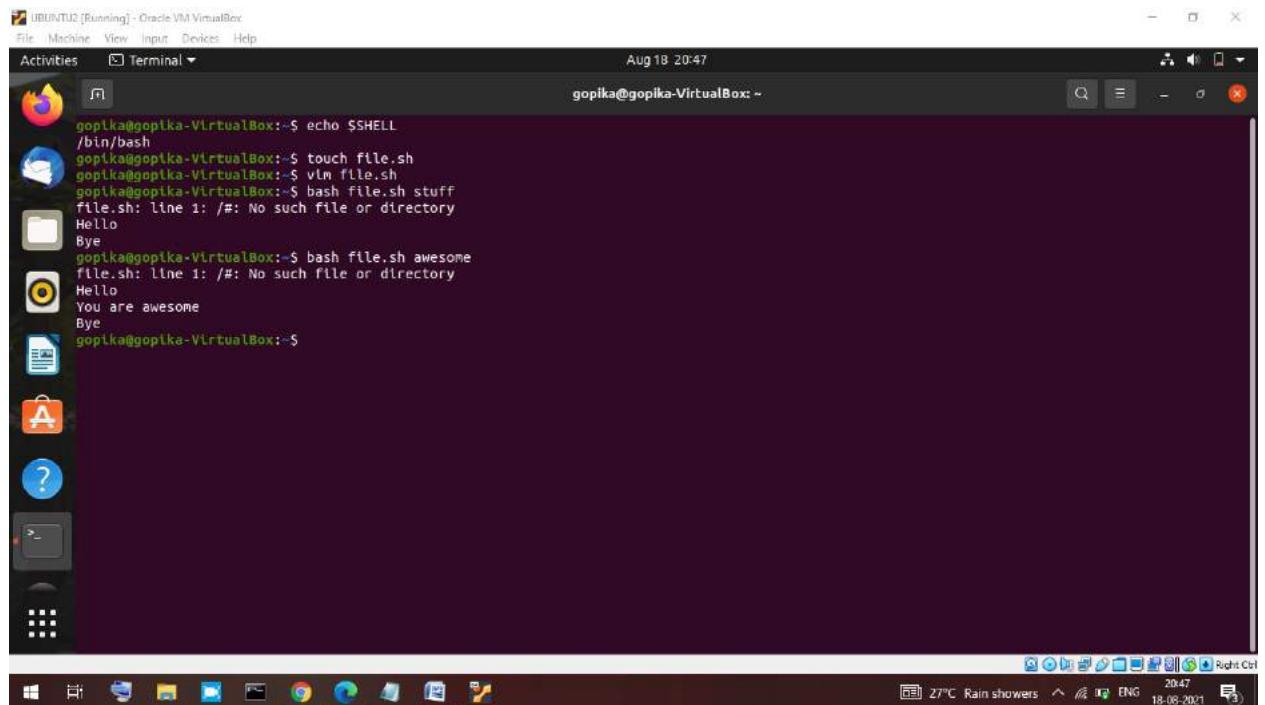
If statement

- The if statement lets us choose whether to run a particular command (or group of commands), depending on some condition.

- The simplest version of this structure has the general form

if conditional expression then command(s) fi

- When the shell encounters a structure such as this, it first checks to see whether the conditional expression is true.
- If so, the shell runs any commands that it finds between the then and the fi (which is just if spelled backwards).
- If the conditional expression is not true, the shell skips the commands between then and fi

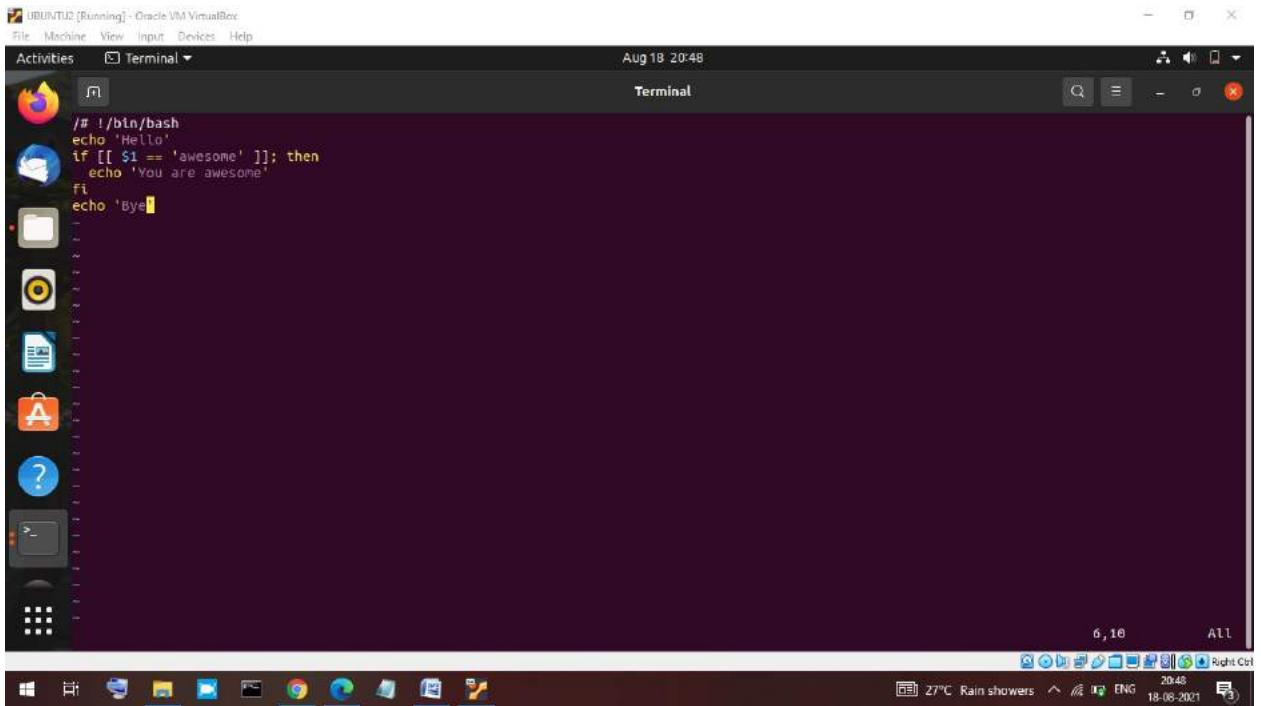


The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "gopika@gopika-VirtualBox: ~". The terminal content is as follows:

```

gopika@gopika-VirtualBox:~$ echo $SHELL
/bin/bash
gopika@gopika-VirtualBox:~$ touch file.sh
gopika@gopika-VirtualBox:~$ vlm file.sh
gopika@gopika-VirtualBox:~$ bash file.sh stuff
file.sh: line 1: /#: No such file or directory
Hello
Bye
gopika@gopika-VirtualBox:~$ bash file.sh awesome
file.sh: line 1: /#: No such file or directory
Hello
You are awesome
Bye
gopika@gopika-VirtualBox:~$
```

The desktop environment includes a dock with various icons and a system tray at the bottom showing weather information (27°C Rain showers), system status, and the date/time (18-08-2021 20:47).



- **for loop**

Sometimes we want to run a command (or group of commands) over and over. This is called iteration, repetition, or looping. The most commonly used shell repetition structure is the for loop, which has the general for:

Syntax:-

```
for var in <list>
do
<commands>
done
```

A screenshot of a Linux desktop environment, likely Ubuntu, running in a VirtualBox window. The desktop has a dark theme with a dock at the bottom containing icons for various applications like File Explorer, Mail, and Browser. A terminal window is open, showing a bash script with a while loop:

```
#!/bin/bash
for value in {1..5}
do
echo $value
done
echo All done
```

A screenshot of a Linux desktop environment, likely Ubuntu, running in a VirtualBox window. The desktop has a dark theme with a dock at the bottom containing icons for various applications like File Explorer, Mail, and Browser. A terminal window is open, showing the execution of a script named vm.sh:

```
gopika@gopika-VirtualBox:~$ touch vm.sh
gopika@gopika-VirtualBox:~$ vim vm.sh
gopika@gopika-VirtualBox:~$ bash vm.sh
1
2
3
4
5
All done
gopika@gopika-VirtualBox:~$
```

While loop

The general form of the while loop is:

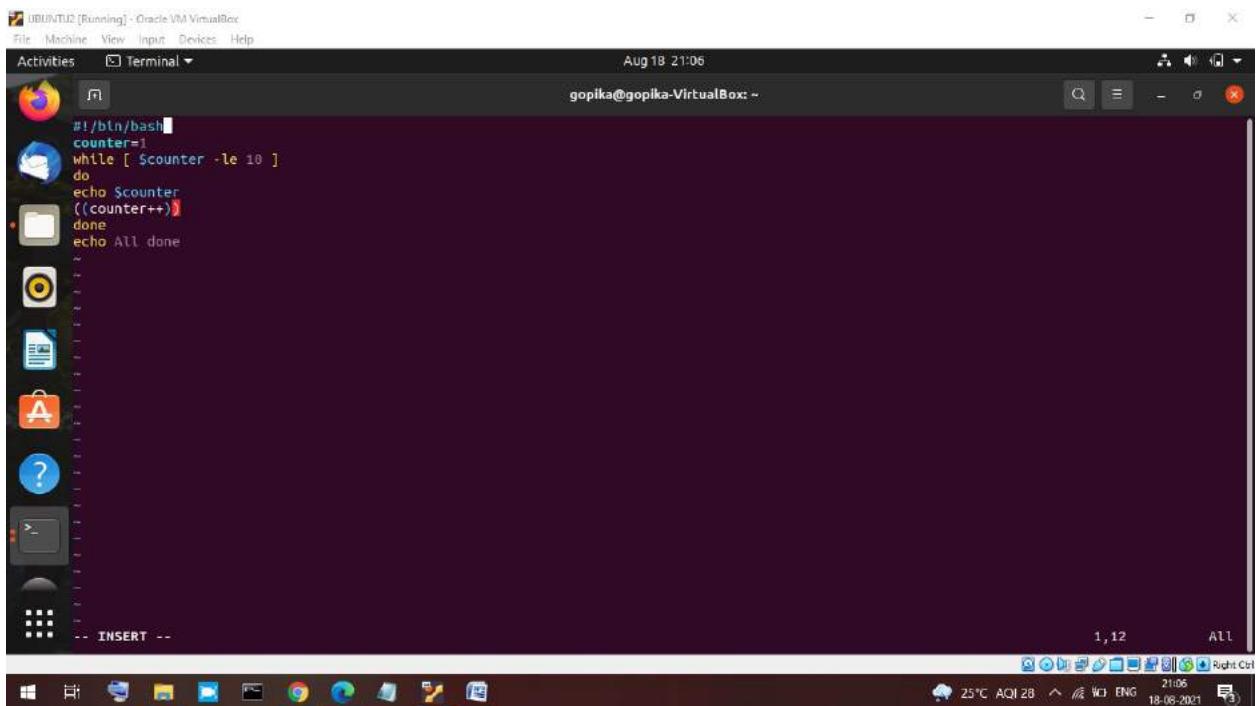
while [<condition>]

do

<command(s)>

done

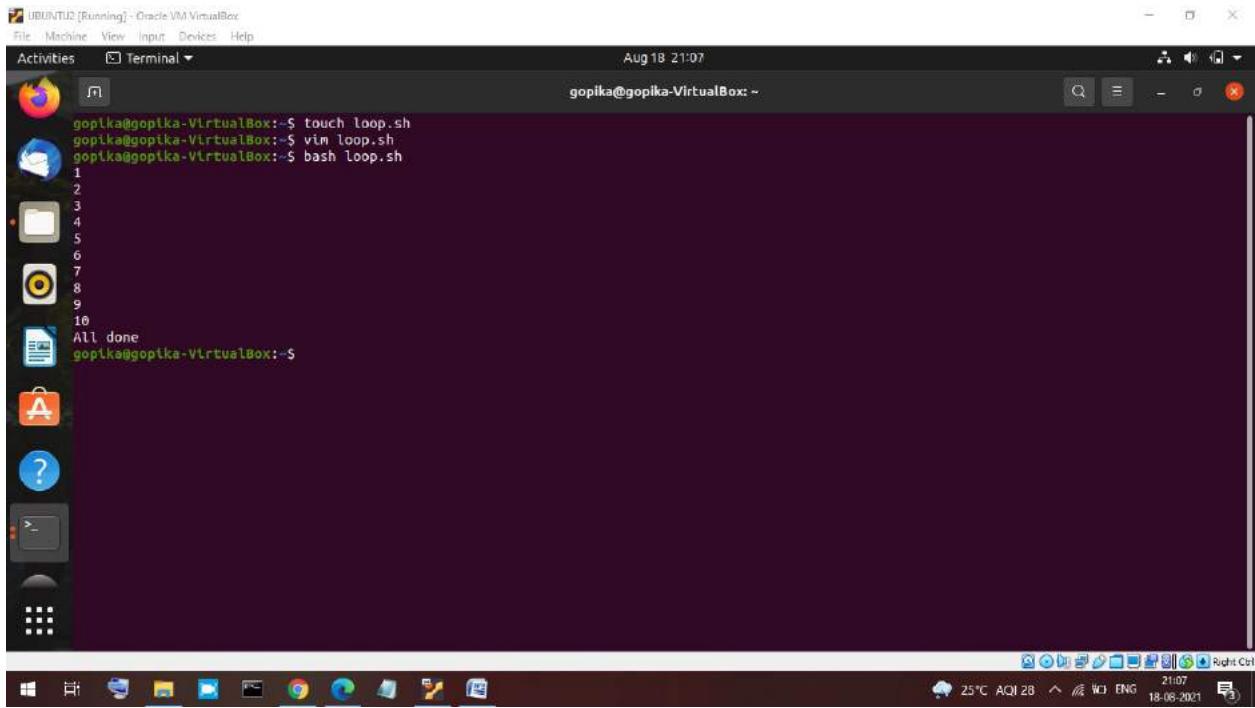
As long as the condition is true, the commands between the do and the done are executed.



The screenshot shows a Linux desktop environment with a dark theme. A terminal window is open, displaying the following bash script:

```
#!/bin/bash
counter=1
while [ $counter -le 10 ]
do
echo $counter
((counter++))
done
echo All done
```

The terminal window has a title bar "UBUNTU2 [Running] - Oracle VM VirtualBox". The status bar at the bottom right shows the date "Aug 18 21:06" and the user "gopika@gopika-VirtualBox: ~". The desktop background is visible, showing various icons in the dock and a weather widget in the bottom right corner.



'alias' in shell scripting

A shell alias is a shortcut to reference a command. It can be used to avoid typing long commands or as a means to correct incorrect input. For common patterns it can reduce keystrokes and improve efficiency. A simple example is setting default options on commands to avoid having to type them each time a command is run.

Syntax:

```
alias [alias_name]=[command]
```

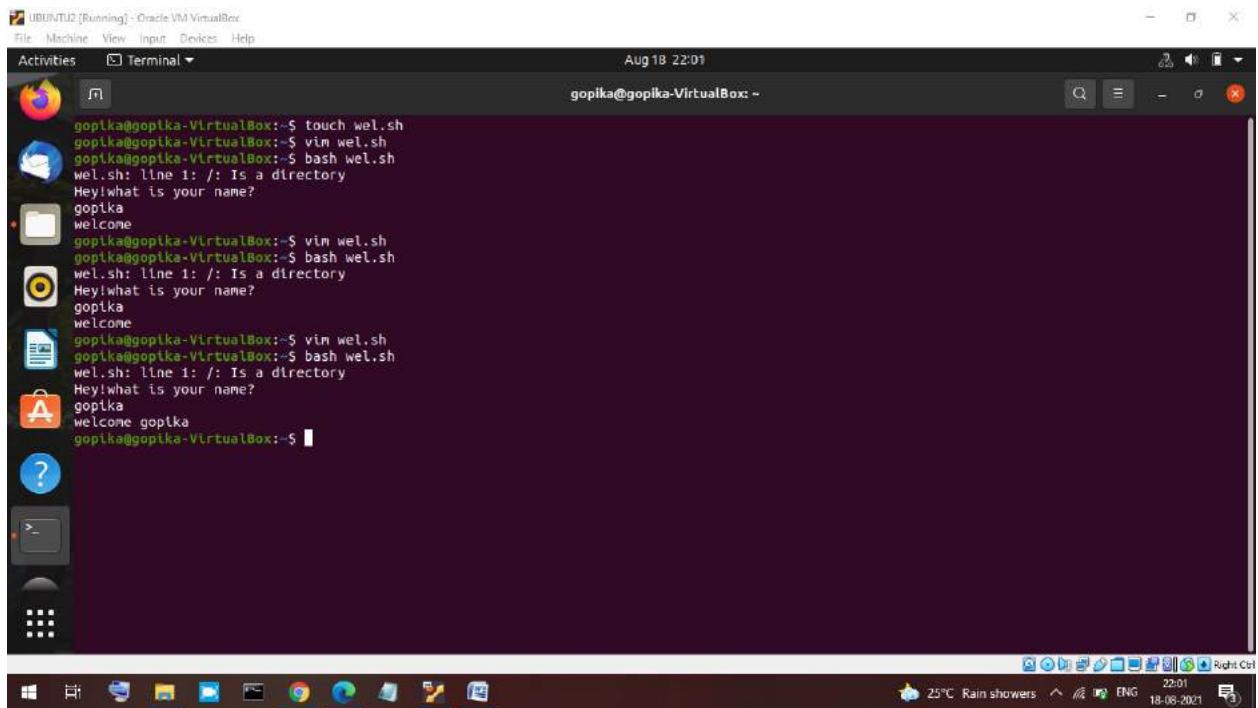
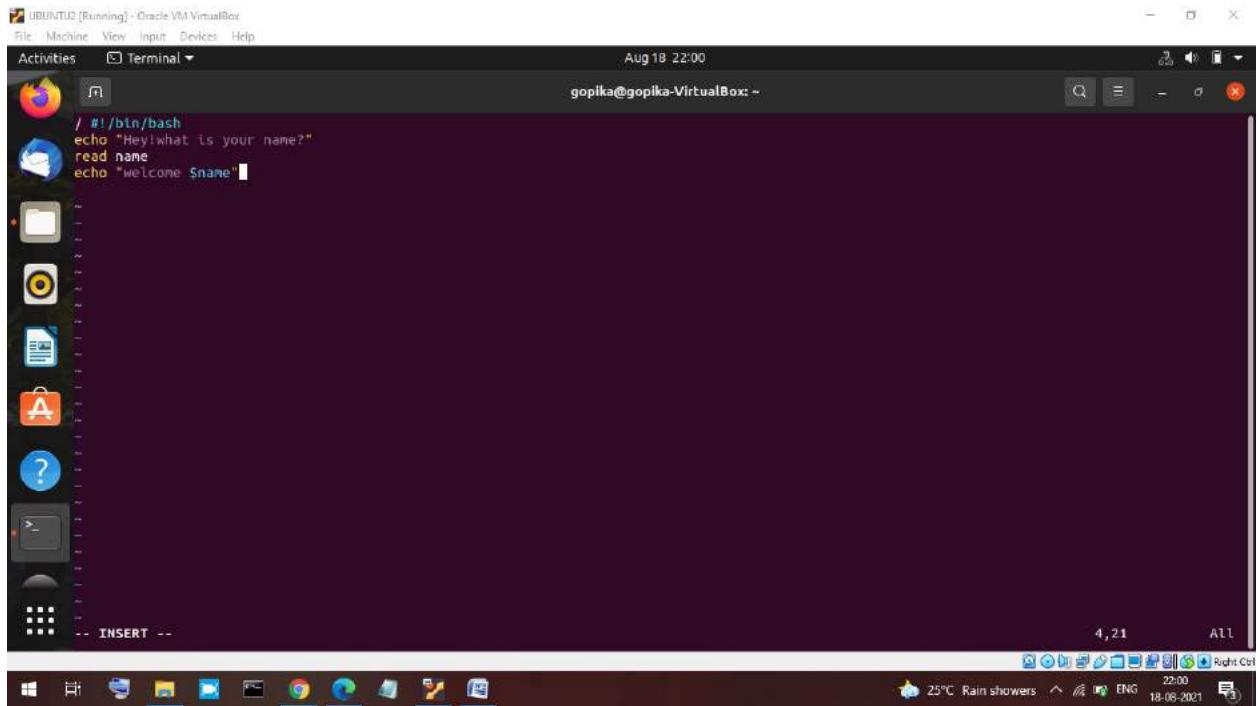
Shell Scripting Example:-

Adding two numbers

A screenshot of a Linux desktop environment, likely Ubuntu, running in Oracle VM VirtualBox. The desktop has a dark theme. On the left is a docked Activities overview panel showing various application icons like a browser, file manager, terminal, and system settings. A terminal window is open at the top right, titled 'gopika@gopika-VirtualBox: ~/Desktop'. It contains a bash script with a function 'Add' that prompts for two numbers and prints their sum. The status bar at the bottom shows system information including temperature, battery level, and network connection.

```
UBUNTU2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Aug 18 21:41
gopika@gopika-VirtualBox: ~/Desktop
gopika@gopika-VirtualBox:~/Desktop$ vim ex.sh
gopika@gopika-VirtualBox:~/Desktop$ bash exsh
bash: exsh: No such file or directory
gopika@gopika-VirtualBox:~/Desktop$ bash ex.sh
Enter the first number: 12
Enter the second number: 8
12+8= 20
gopika@gopika-VirtualBox:~/Desktop$
```

Example 2:



EXPERIMENT NO: 5

AIM:- WORDPRESS INSTALLATION ON LAMP STACK

WORDPRESS

WordPress is the simplest, most popular way to create our own website or blog. It is an open-source content management system licensed under GPLv2, which means that anyone can use or modify the WordPress software for free.

A content management system(CMS) is basically a tool that makes it easy to manage important aspects of the website like content without needing to know anything about programming.

BENEFITS OF WordPress

- WordPress is free
- Simple and easy to use
- Stable and innovative
- Attractive design options
- Flexibility
- Sharing ability
- Easy content management
- Open source

LAMP

LAMP is a software stack designed to host web content, like websites and web apps. It's called LAMP because it's built on Linux (L), Apache web server (A), a MySQL database (M) or MariaDB, and the PHP scripting language (P).

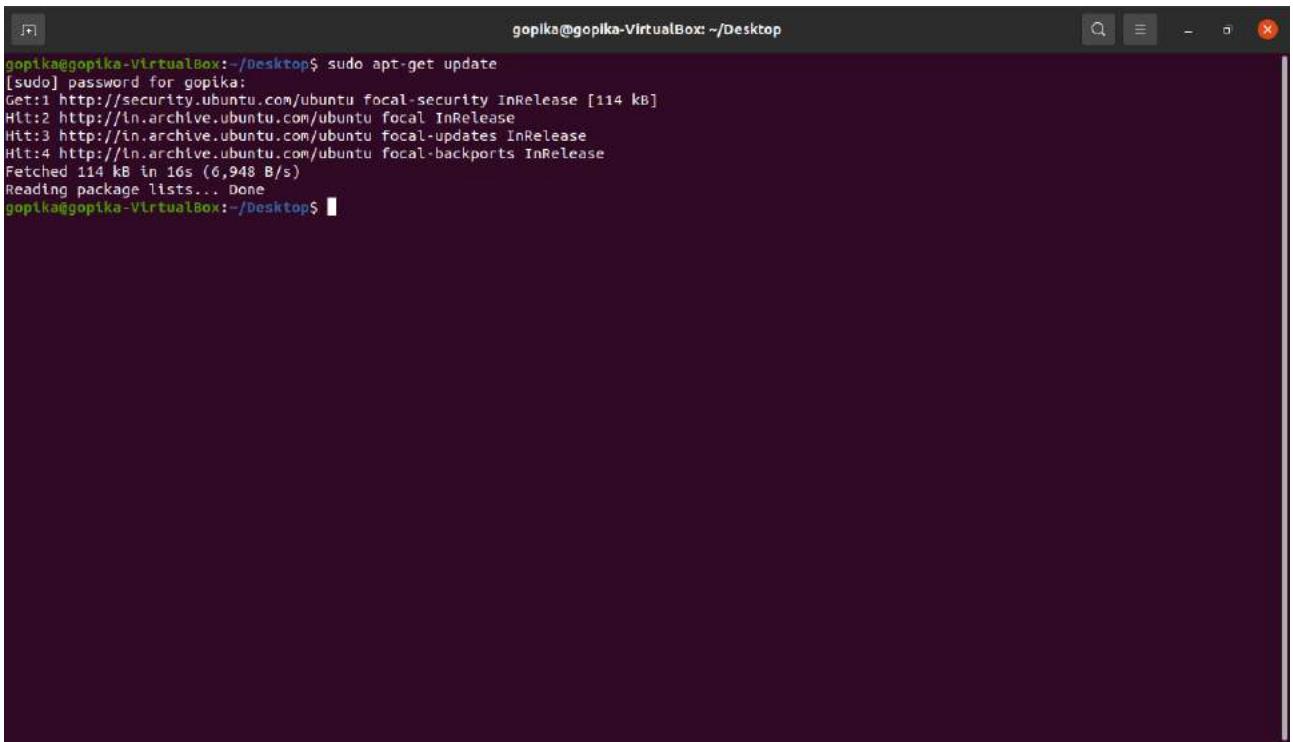
LAMP is referred to as a software stack, because each of them built on top of the other. Linux is the operating system, which creates the base layer. Apache is on the top of the linux because it needs Linux to operate. MySQL is along with Apache to store all the information it needs. And PHP sits atop the stack to drive the actual display of the website, pull info from the database and create user experience.

INSTALLATION PROCESS

STEPS:-

Install LAMP stack , for that open the terminal and type the following commands. At first check whether there is any updates available with

```
sudo apt-get update
```



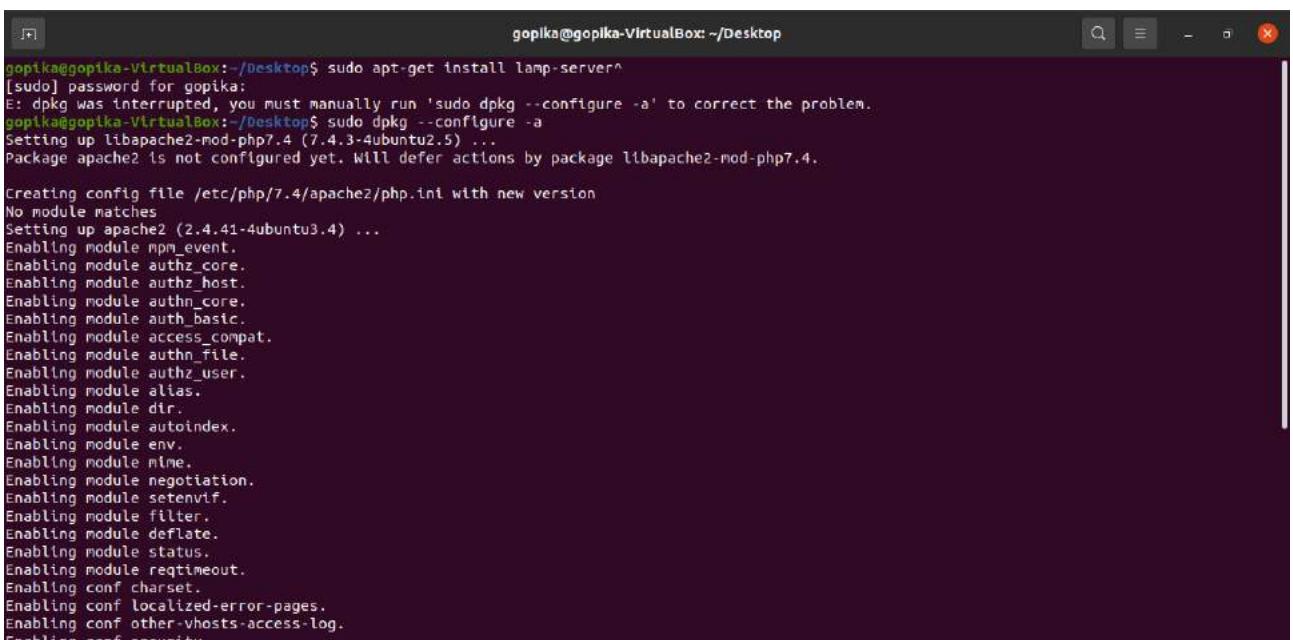
```
gopika@gopika-VirtualBox:~/Desktop$ sudo apt-get update
[sudo] password for gopika:
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Hit:2 http://in.archive.ubuntu.com/ubuntu focal InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:4 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease
Fetched 114 kB in 16s (6,948 B/s)
Reading package lists... Done
gopika@gopika-VirtualBox:~/Desktop$
```

If any updates available upgrade it using

sudo apt-get upgrade

Now type the command in terminal to install LAMP that is

sudo apt-get install lamp-server^



```
gopika@gopika-VirtualBox:~/Desktop$ sudo apt-get install lamp-server^
[sudo] password for gopika:
E: dpkg was interrupted, you must manually run 'sudo dpkg --configure -a' to correct the problem.
gopika@gopika-VirtualBox:~/Desktop$ sudo dpkg --configure -a
Setting up libapache2-mod-php7.4 (7.4.3-4ubuntu2.5) ...
Package apache2 is not configured yet. Will defer actions by package libapache2-mod-php7.4.

Creating config file /etc/php/7.4/apache2/php.ini with new version
No module matches
Setting up apache2 (2.4.41-4ubuntu3.4) ...
Enabling module mpn_event.
Enabling module authz_core.
Enabling module authz_host.
Enabling module authn_core.
Enabling module auth_basic.
Enabling module access_compat.
Enabling module authn_file.
Enabling module authz_user.
Enabling module alias.
Enabling module dir.
Enabling module autoindex.
Enabling module env.
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
```

```
gopika@gopika-VirtualBox: ~/Desktop
Enabling module alias.
Enabling module dir.
Enabling module autoindex.
Enabling module env.
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Info: Switch to mpn prefork for package libapache2-mod-php7.4
Module mpn_event disabled.
Enabling module mpn_prefork.
info: Executing deferred 'a2enmod php7.4' for package libapache2-mod-php7.4
Enabling module php7.4.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib/systemd/system/apache-htcacheclean.service.
Setting up libapache2-mod-php (2:7.4+75) ...
Processing triggers for ufw (0.30-0) ...
Processing triggers for systemd (245.4-4ubuntu3.4) ...
Processing triggers for php7.4-cli (7.4.3-4ubuntu2.5) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libapache2-mod-php7.4 (7.4.3-4ubuntu2.5) ...
gopika@gopika-VirtualBox:~/Desktop$
```

LAMP is successfully installed. Now we want to get into the root directory of Apache ,for that

```
cd var/www/html/
```

and then we need to download the WordPress ,

```
sudo wget https://wordpress.org/latest.tar.gz
```

```
gopika@gopika-VirtualBox:/var/www/html$ sudo wget https://wordpress.org/latest.tar.gz
```

Then we get the zip folder of latest version of WordPress and we need to unzip it using

```
sudo tar xvfz latest.tar.gz
```

After the folder is get unzipped, list the files in it using the command

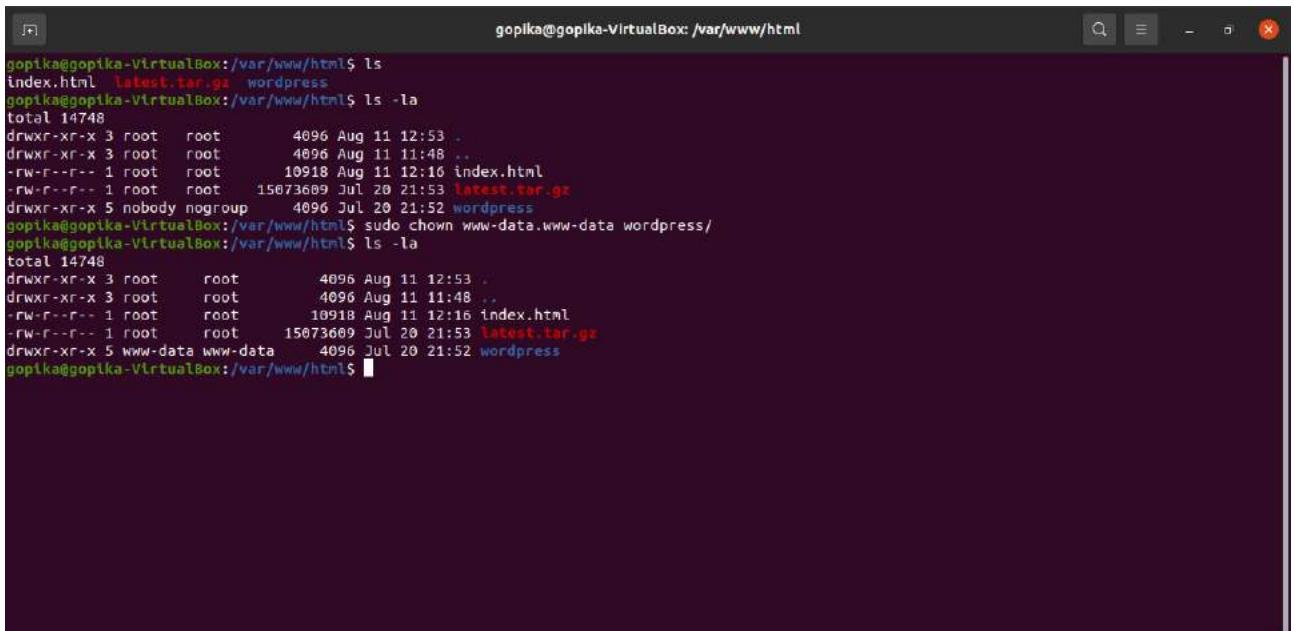
```
ls
```

then check the permission of the wordpress file using

```
ls -la
```

At first the wordpress file seems to be no one can write or change data, so we need to change the file permission using **chown** (change ownership) command

```
sudo chown www-data.www-data wordpress/
```



The screenshot shows a terminal window with a dark background and light-colored text. It displays two sets of directory listings from the '/var/www/html' directory. The first set is before the ownership change, showing files 'index.html', 'latest.tar.gz', and 'wordpress' with various permissions and sizes. The second set is after the 'sudo chown' command, where the 'wordpress' directory now has permissions 'drwxr-xr-x 5 www-data www-data 4096 Jul 20 21:52 wordpress'. The terminal prompt is 'gopika@gopika-VirtualBox: /var/www/html\$'.

Now we need to create a database for wordpress, for that we need to get into mysql

```
sudo mysql
```

along with that we need to create an username and password for MySQL using

```
CREATE USER 'username'@'localhost' IDENTIFIED BY 'password';
```

give username and password as our interest and grant permission to the user to access the databases by

```
GRANT ALL PRIVILEGES ON*.*TO 'GOPIKA'@'localhost';
```

```
gopika@gopika-VirtualBox: /var/www/html
Enter password:
ERROR 1098 (28000): Access denied for user 'root'@'localhost'
gopika@gopika-VirtualBox:/var/www/html$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.26-0ubuntu0.20.04.2 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE USER 'GOPIKA'@'localhost' IDENTIFIED BY 'Gopika@123';
-> CREATE USER 'GOPIKA'@'localhost' IDENTIFIED BY 'Gopika@123';
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax
to use near 'CREATE USER 'GOPIKA'@'localhost' IDENTIFIED BY 'Gopika@123'' at line 2
mysql> mysql --v
-> clear
-> ^C

^C
mysql> CREATE USER 'GOPIKA'@'localhost' IDENTIFIED BY 'Gopika@123';
Query OK, 0 rows affected (4.21 sec)

mysql>
mysql> GRANT ALL PRIVILEGES ON*.* TO 'GOPIKA'@'localhost';
Query OK, 0 rows affected (0.30 sec)

mysql>
```

Check the existing databases using the command
show databases;
and then create database named wordpress

```
create database wordpress;

gopika@gopika-VirtualBox: /var/www/html
mysql> GRANT ALL PRIVILEGES ON*.* TO 'GOPIKA'@'localhost';
Query OK, 0 rows affected (0.30 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.69 sec)

mysql> create database wordpress;
Query OK, 1 row affected (0.21 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| wordpress |
+-----+
5 rows in set (0.00 sec)

mysql> |
```

and then launch **localhost/wordpress** and then connect the database to the wordpress

WordPress › Setup Configuration

localhost/wordpress/wp-admin/setup-config.php?step=1

Below you should enter your database connection details. If you're not sure about these, contact your host.

| | | |
|----------------------|------------|--|
| Database Name | wordpress | The name of the database you want to use with WordPress. |
| Username | GOPINKA | Your database username. |
| Password | Gopika@123 | Your database password. |
| Database Host | localhost | You should be able to get this info from your web host, if localhost doesn't work. |
| Table Prefix | all_ | If you want to run multiple WordPress installations in a single database, change this. |

Submit



click on submit and create user login credentials and click on install wordpress.

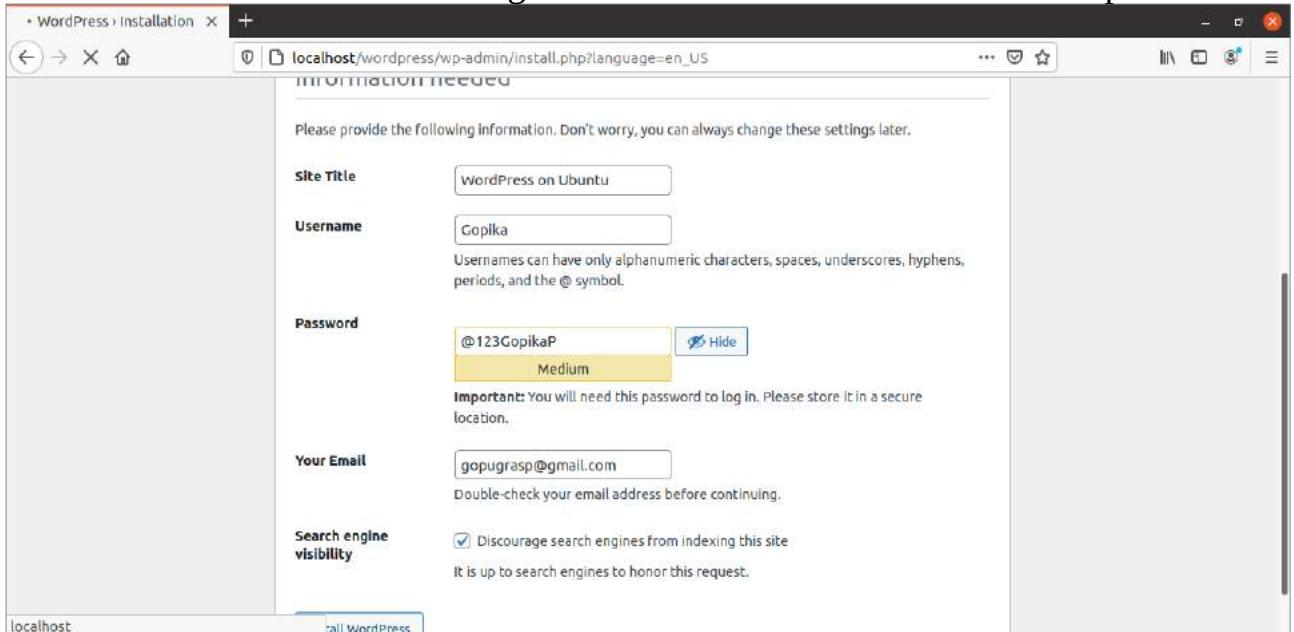
WordPress › Installation

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

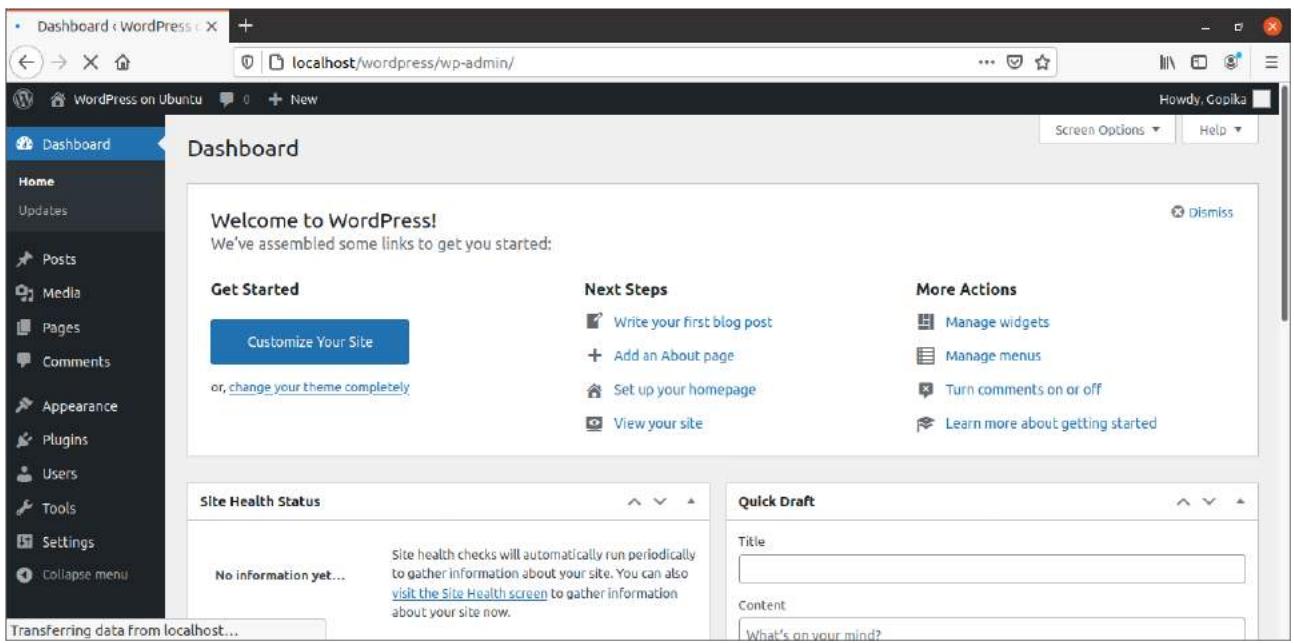
Please provide the following information. Don't worry, you can always change these settings later.

| | |
|--|--|
| Site Title | WordPress on Ubuntu |
| Username | Gopika |
| Password | @123GopikaP Medium |
| Important: You will need this password to log in. Please store it in a secure location. | |
| Your Email | gopugrasp@gmail.com |
| Search engine visibility | <input checked="" type="checkbox"/> Discourage search engines from indexing this site It is up to search engines to honor this request. |

localhost **Install WordPress**



login to the wordpress using the login credentials.



WordPress is successfully installed locally.

EXPERIMENT NO:6

AIM :- INSTALLATION AND CONFIGURATION OF PHP

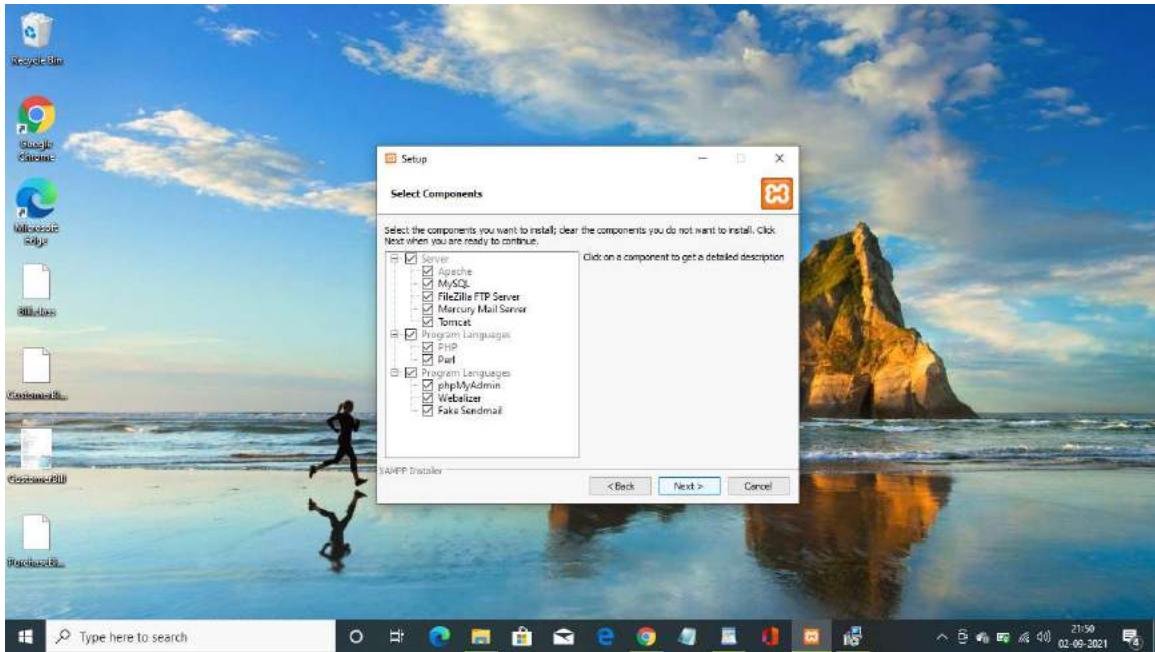
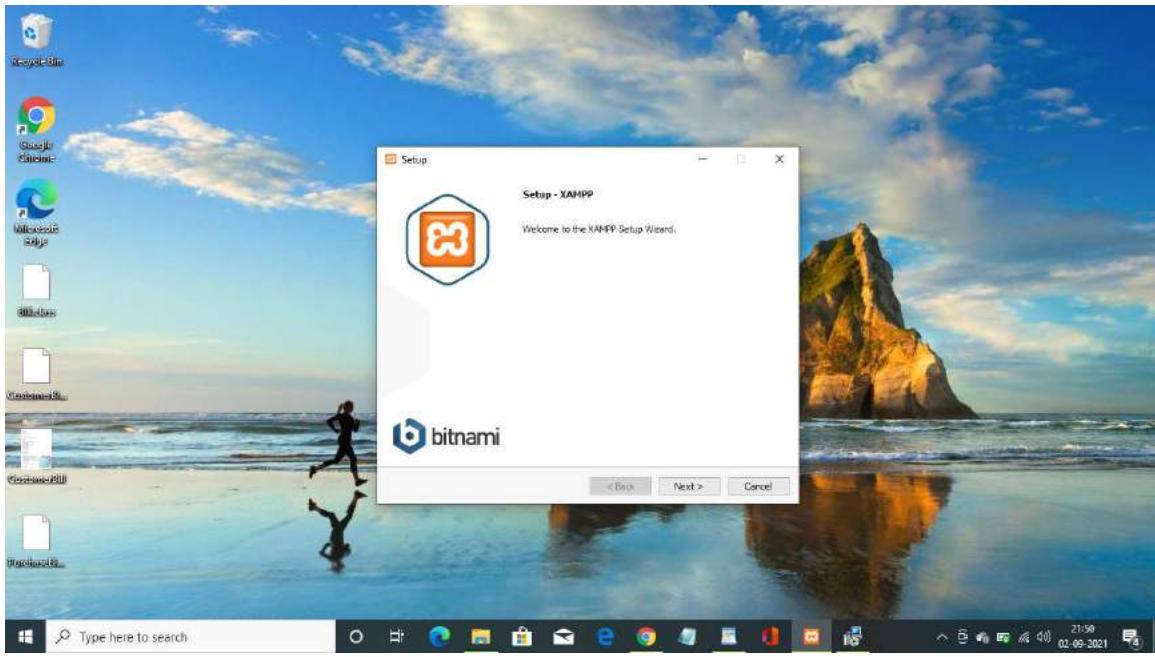
FRAMEWORK LARAVEL

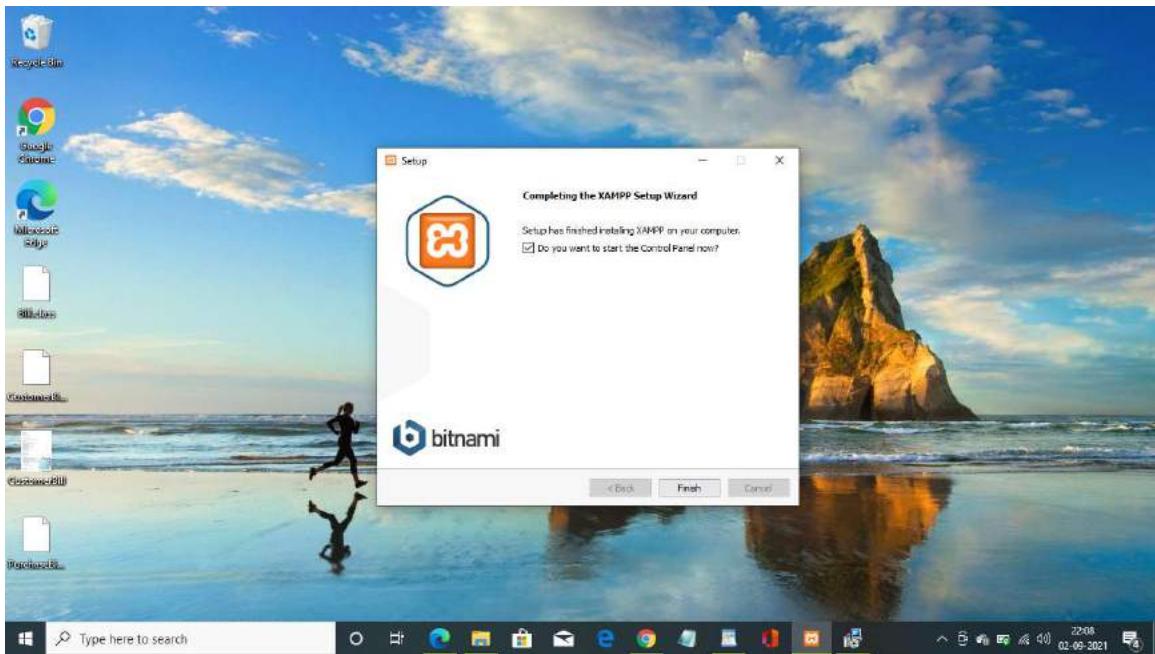
LARAVEL

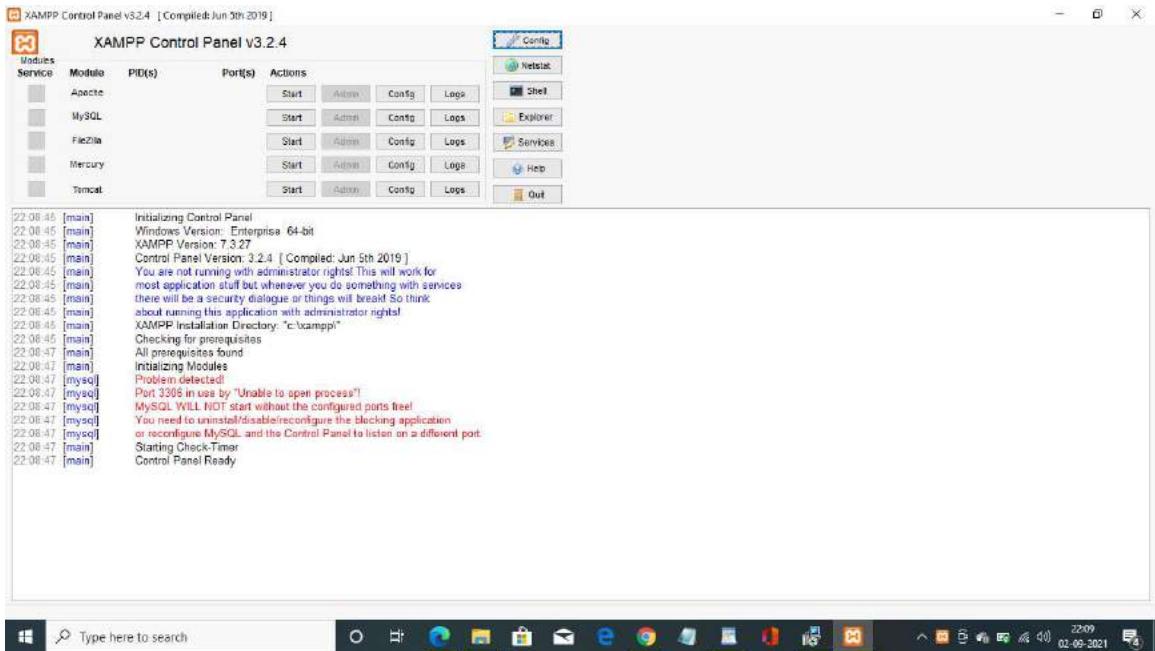
Laravel is an open-source PHP framework, which is robust and easy to understand. It reuses the existing components of different frameworks which helps in creating a web application. The web application thus designed is more structured and sensible. A website built in Laravel is secure and prevents several web attacks.

PROCEDURE

Install xampp



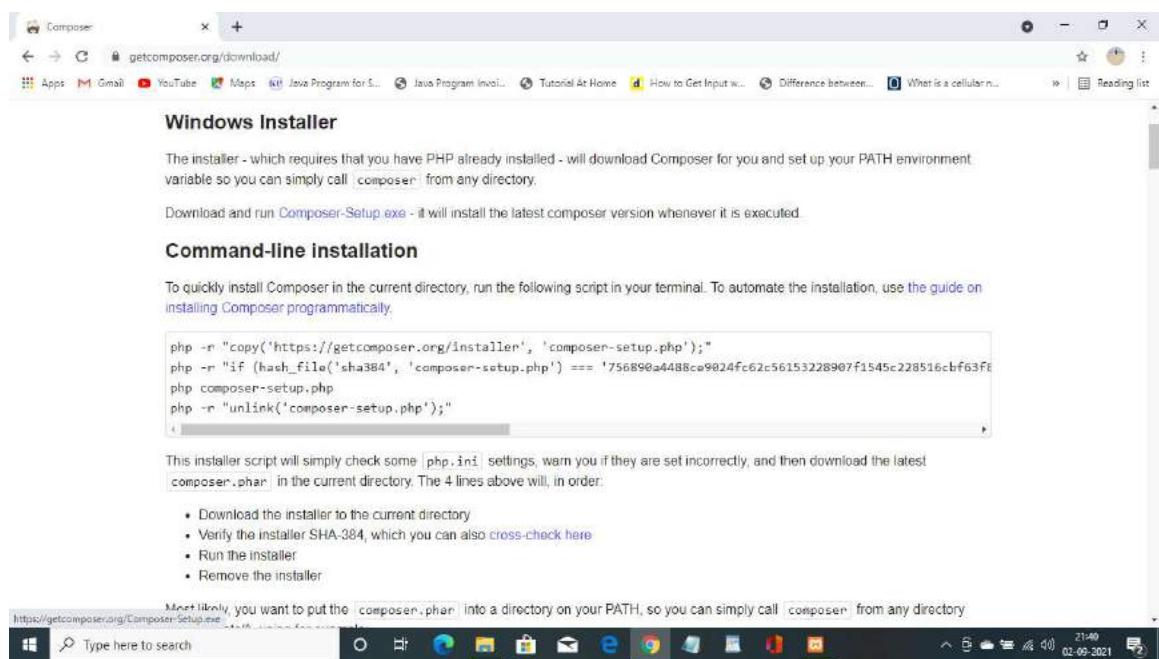
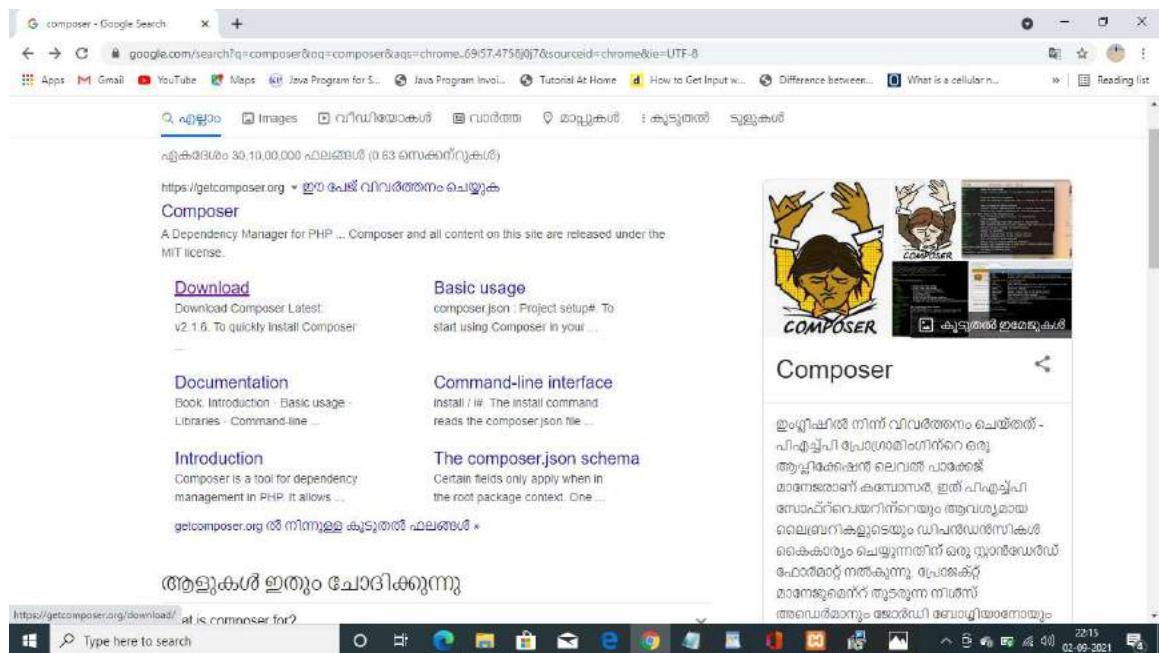




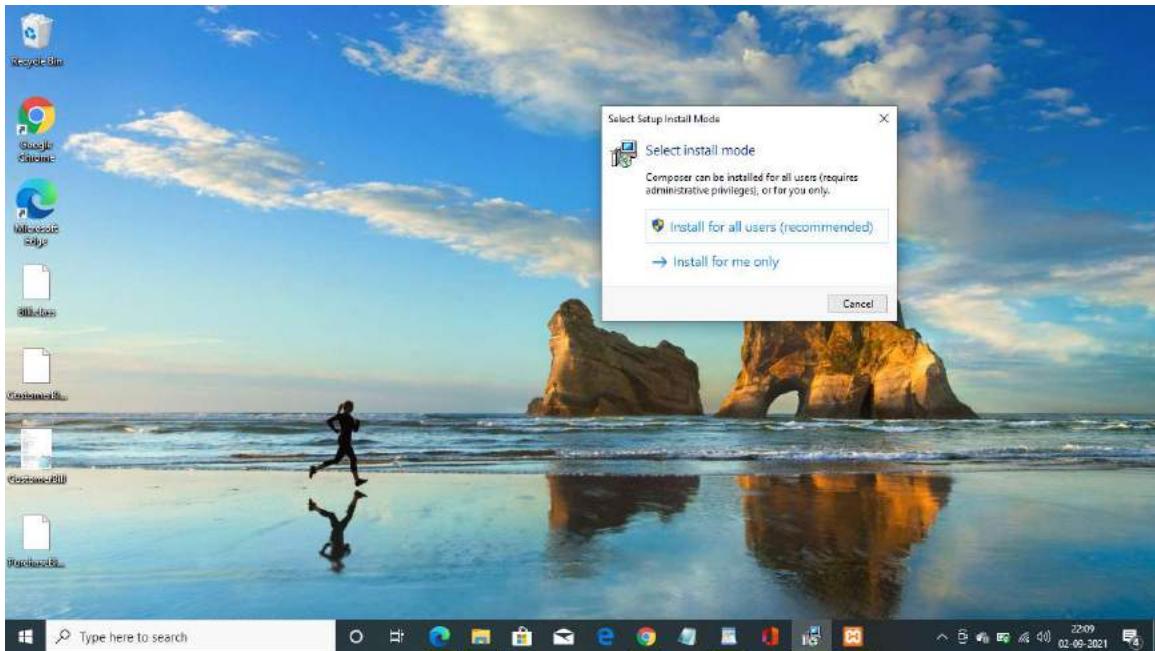
Sucessfully installed XAMPP.

step 2:setup composer

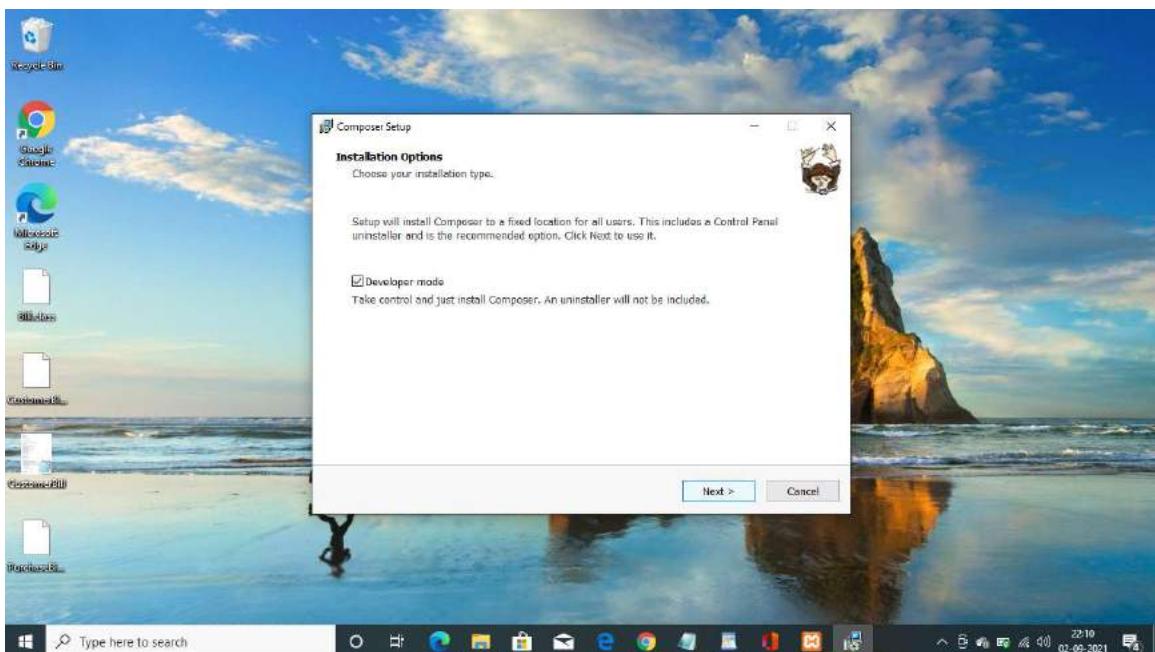
Composer is a tool which includes all the dependencies and libraries. It allows a user to create a project with respect to the mentioned framework (for example, those used in Laravel installation). Third party libraries can be installed easily with help of composer.



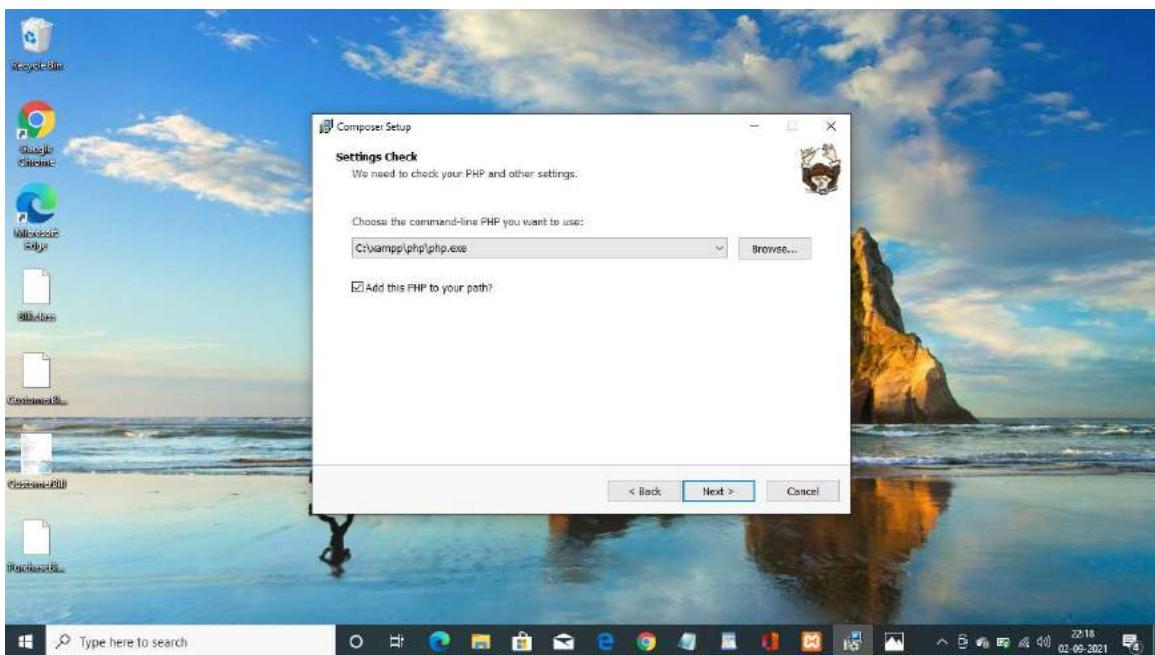
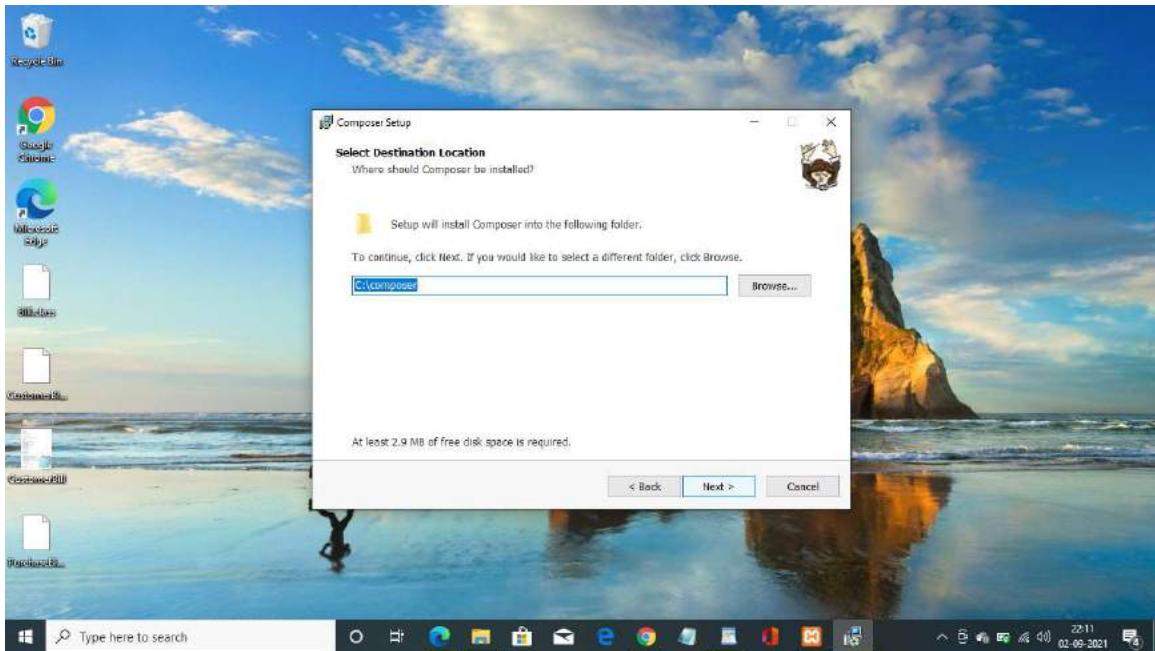
Click on the Composer.setup.exe and the file get downloaded



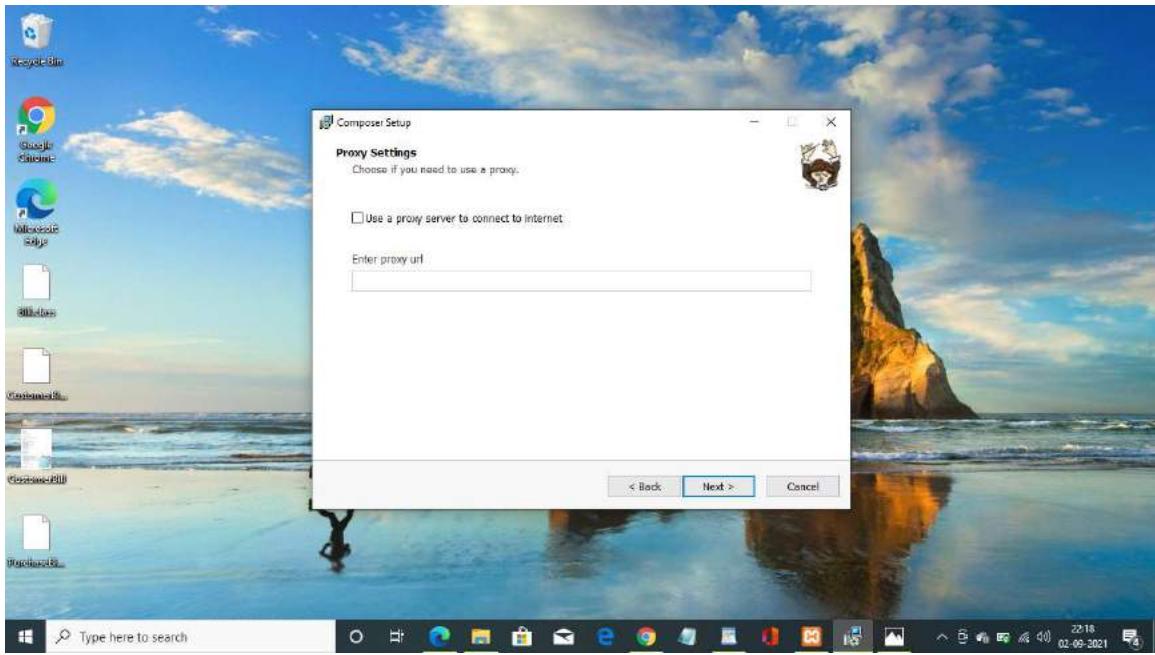
click on install for all users.



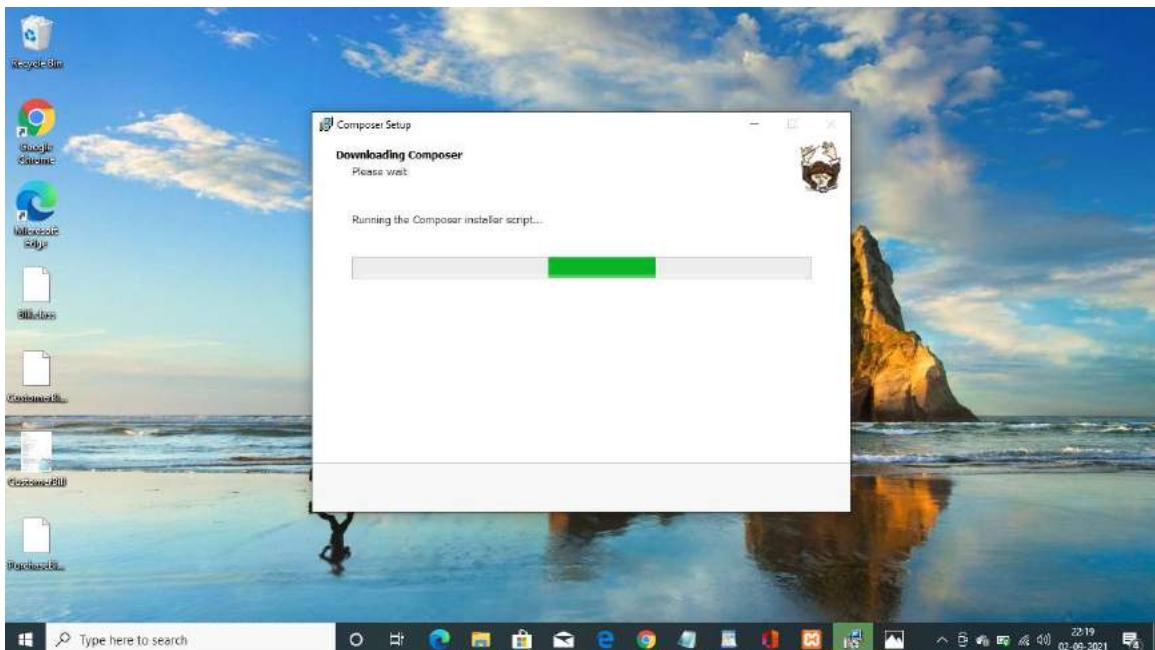
check the developer mode

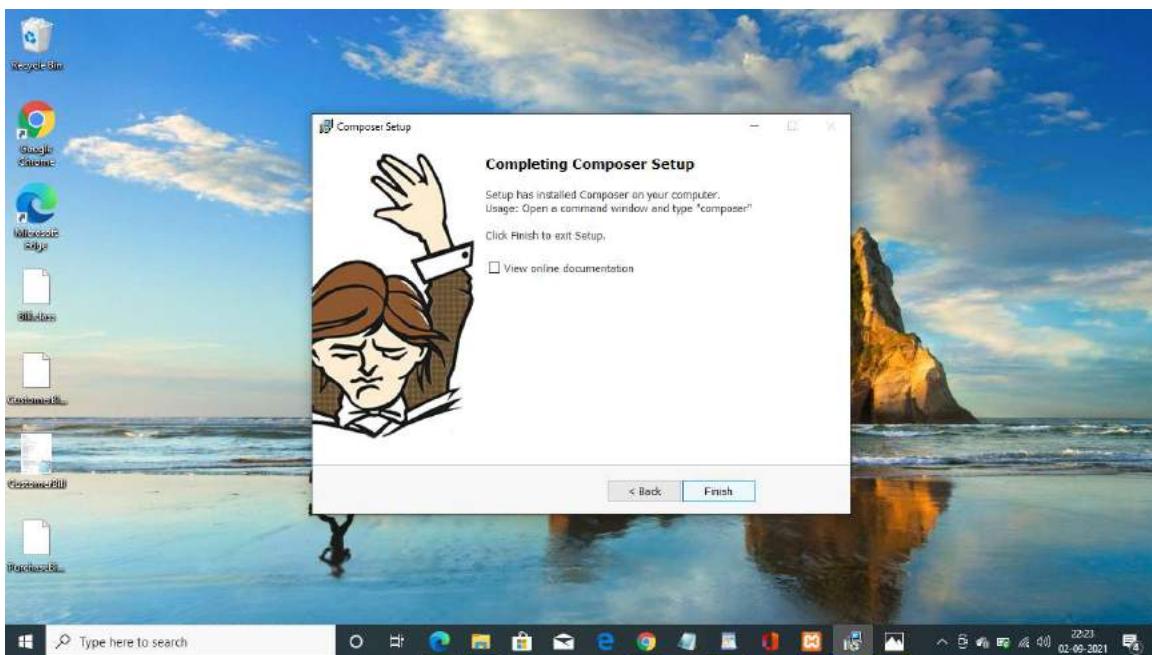
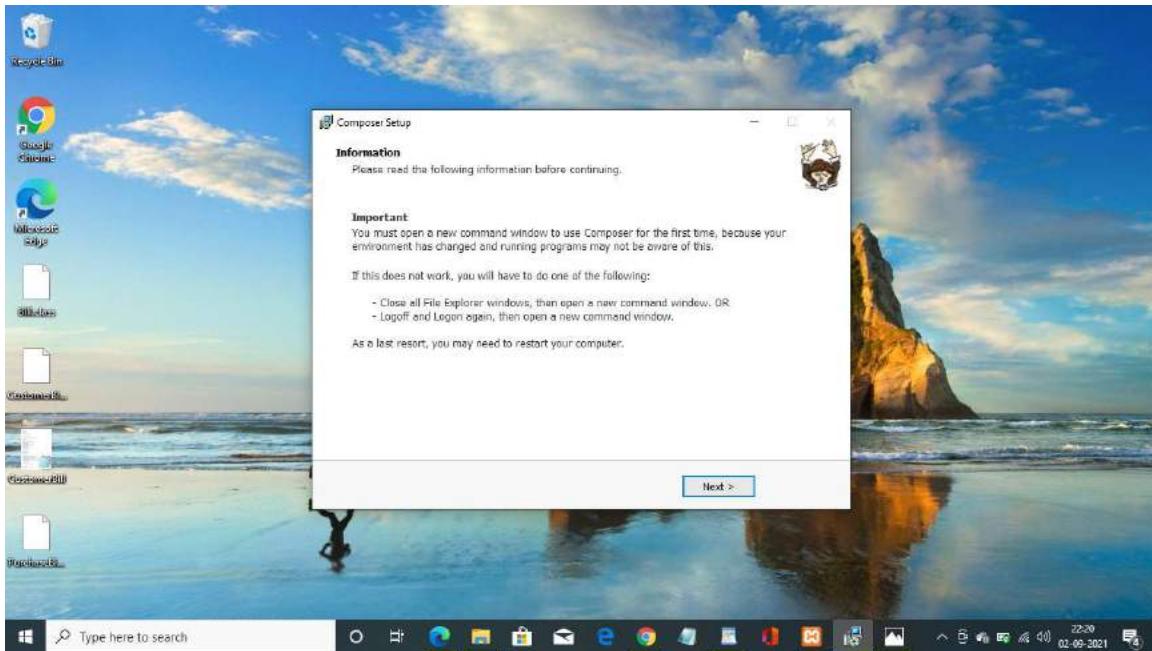


check the box if needed to add path to the system environment variable along with the installation ,otherwise leave as such and later setup the path environment variable.

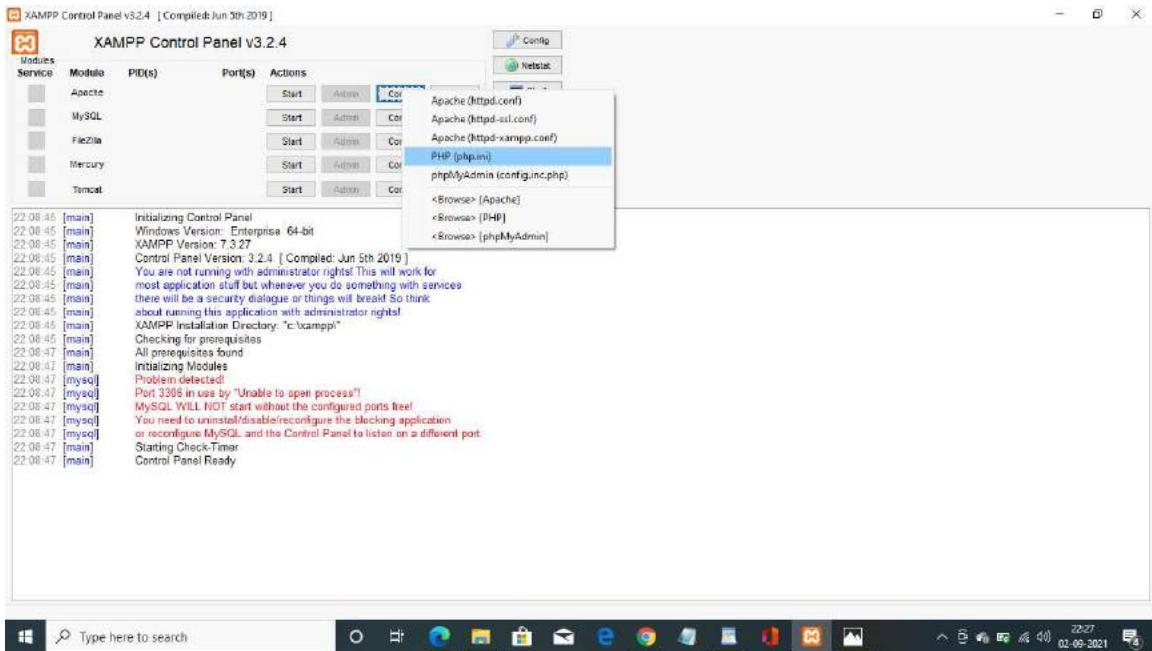


leave as such and click next

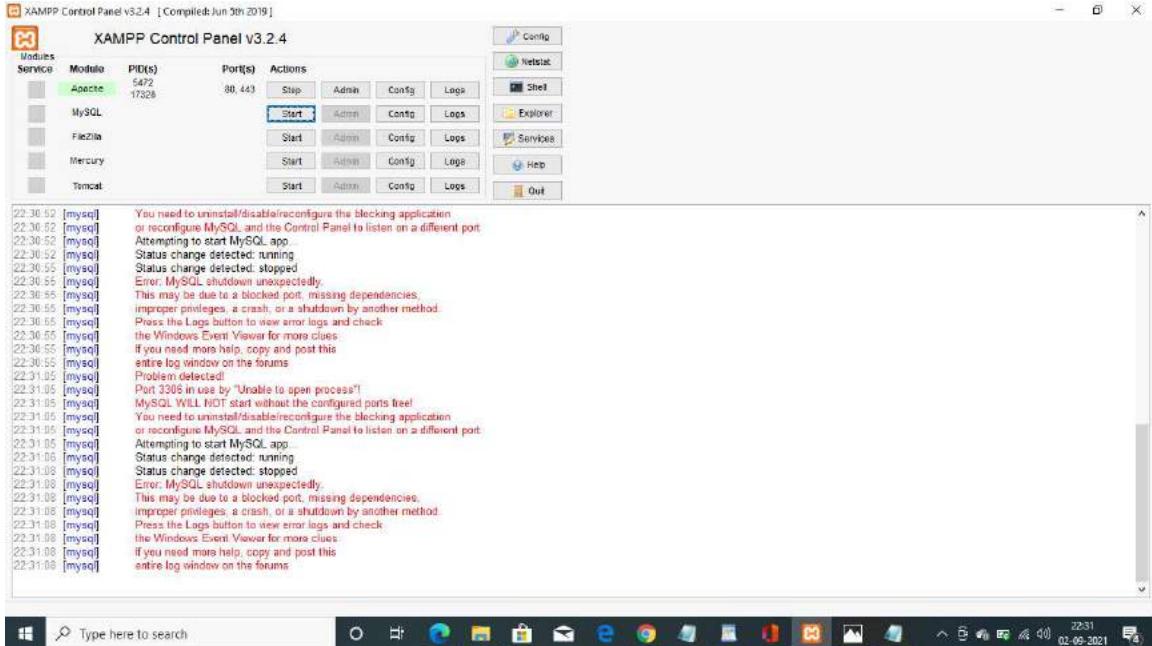




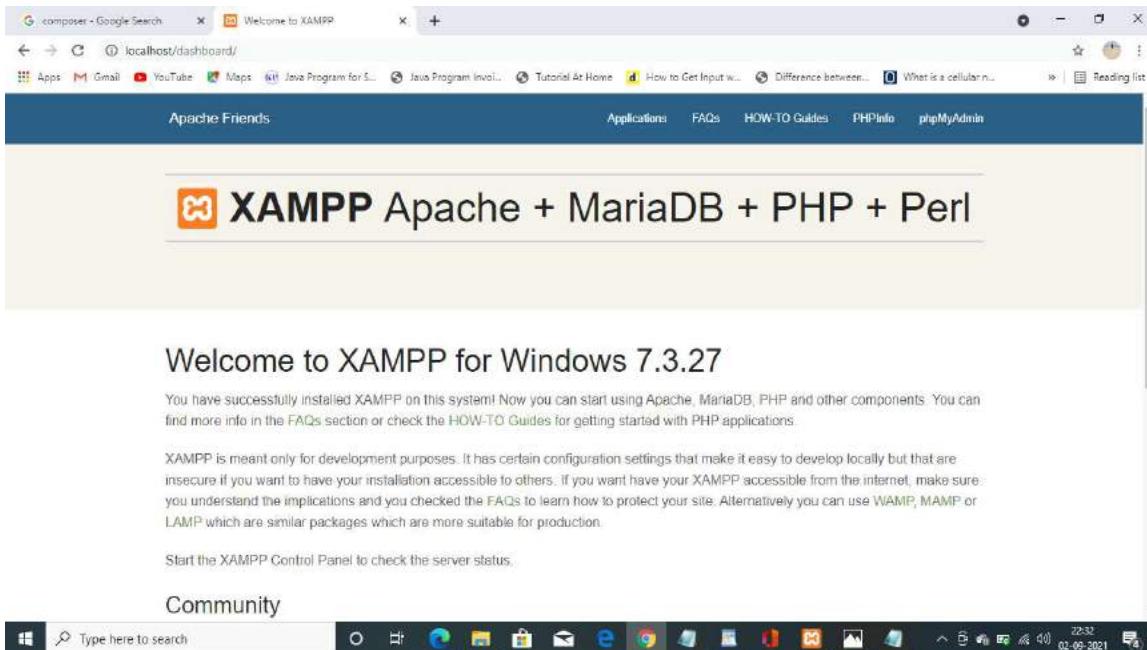
Now the composer get successfully installed



check whether all the extensions required are properly installed by right click on config near apache and select php/php.in and search extension which are installed.



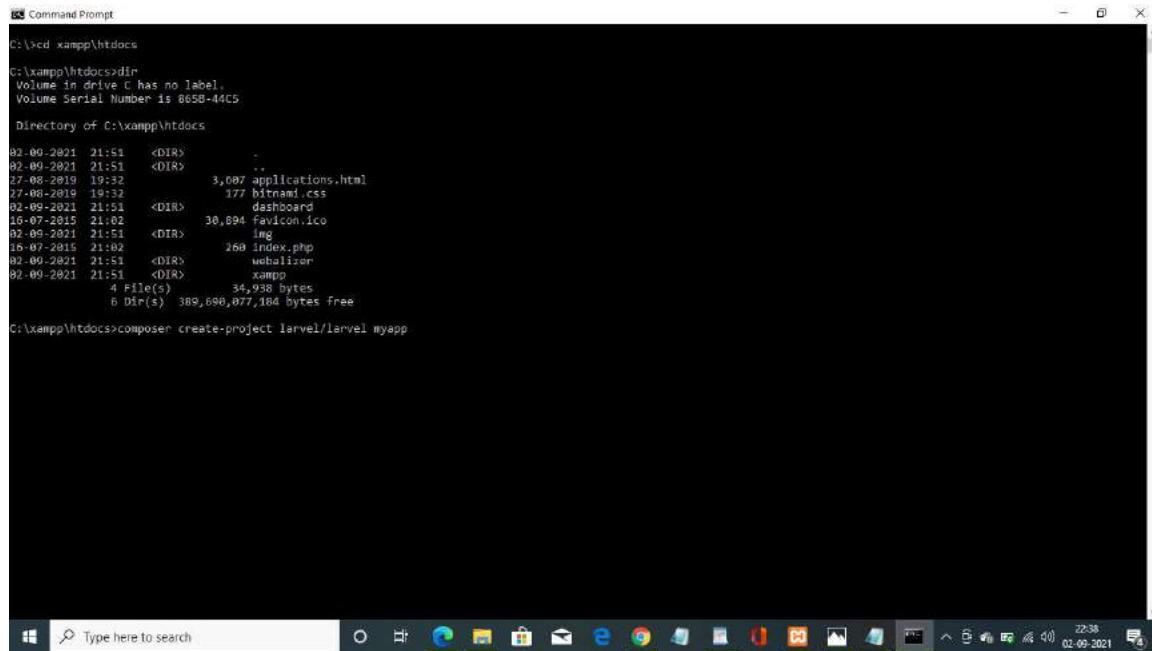
open the browser and type localhost



open command prompt

A screenshot of a Command Prompt window. The command "cd xampp\htdocs>dir" is run, displaying a directory listing of files and subdirectories in the XAMPP htdocs folder. The listing includes "applications.html", "bitnami.css", "dashboard", "favicon.ico", "img", "index.php", "webalizer", and "xampp". The total size of files is 34,938 bytes and the free space is 389,690,077,184 bytes. The Windows taskbar is visible at the bottom.

Enter on htdocs and type the command "composer create-project laravel/laravel myapp" ,here in the place of myapp we can give whatever name we wish to give our application.

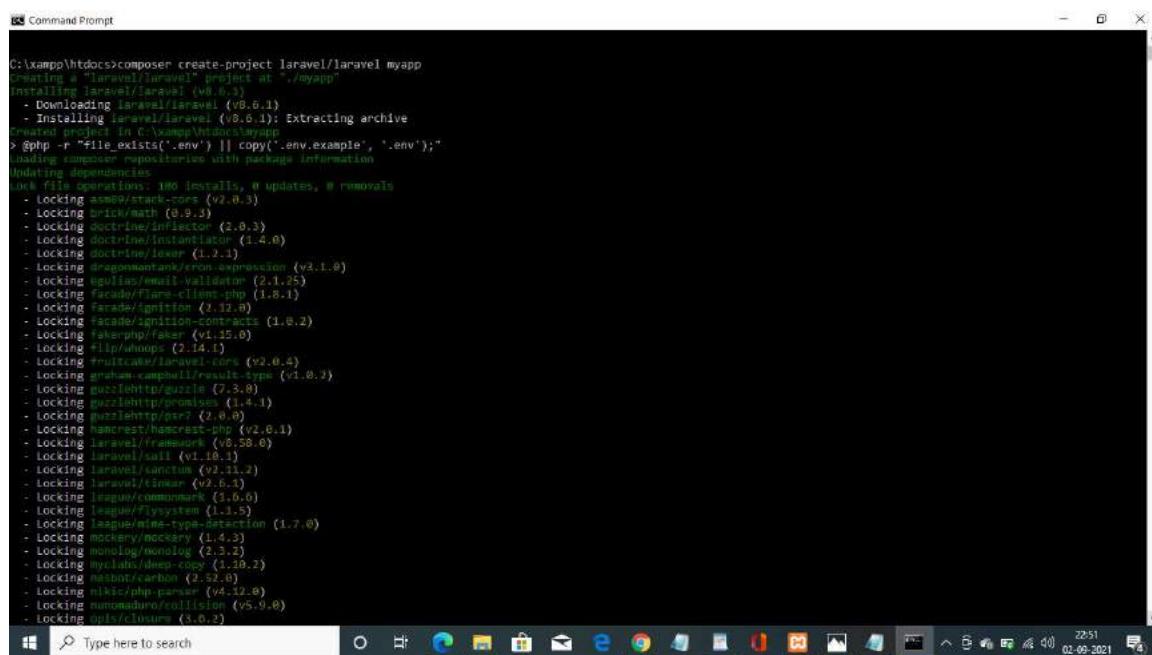


```
C:\ Command Prompt
C:\>cd xampp\htdocs
C:\xampp\htdocs>dir
Volume in drive C has no label.
Volume Serial Number is 8658-44C5

Directory of C:\xampp\htdocs

02-09-2021 21:51    <DIR>      .
02-09-2021 21:51    <DIR>      ..
27-08-2019 19:30        3,007 applications.html
27-08-2019 19:30           177 bitnami.css
02-09-2021 21:51    <DIR>      dashboard
16-07-2015 21:02           30,094 favicon.ico
07-09-2021 21:02    <DIR>      img
02-09-2021 21:51    <DIR>      index.php
02-09-2021 21:51    <DIR>      webalizer
02-09-2021 21:51    <DIR>      xampp
               4 File(s)       34,938 bytes
               6 Dir(s)   389,696,077,184 bytes free

C:\xampp\htdocs>composer create-project laravel/laravel myapp
```



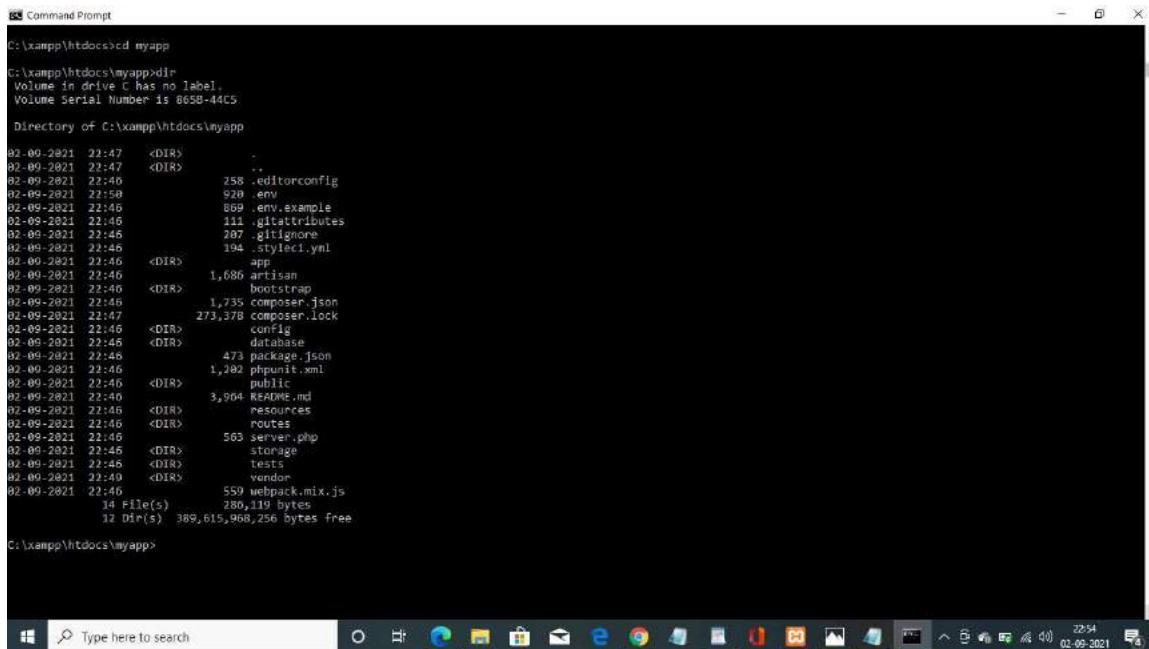
```
C:\xampp\htdocs>composer create-project laravel/laravel myapp
Creating a "laravel/laravel" project at "./myapp"
Installing laravel/laravel (v8.0.1)
- Downloading laravel/laravel (v8.0.1)
- Installing laravel/laravel (v8.0.1): Extracting archive
Created project in C:\xampp\htdocs\myapp
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 100 installs, 0 updates, 0 removals
- Locking aries/stack-concave (v2.0.3)
- Locking arish/math (0.9.3)
- Locking doctrine/inflexion (2.0.3)
- Locking doctrine/inflector (1.4.6)
- Locking doctrine/lexer (1.2.1)
- Locking dragonmantank/cron-expression (v3.1.8)
- Locking egelsias/mailValidator (2.1.5)
- Locking fzaninotto/faker (1.8.1)
- Locking fzaninotto/ionitron (2.12.0)
- Locking fzaninotto/ignition (1.0.2)
- Locking fzaninotto/ignition-contracts (1.0.2)
- Locking fzaninotto/tarantool (v1.15.0)
- Locking fzaninotto/tarantool (2.14.1)
LOCKING fzaninotto/tarantool (v2.0.8)
Locking guzzlehttp/multipart-entity (v1.0.3)
Locking guzzlehttp/psr7 (7.3.0)
Locking guzzlehttp/promises (1.4.1)
Locking guzzlehttp/psr7 (2.0.0)
Locking hamcrest/hamcrest-php (v2.0.1)
Locking laravel/framework (v8.58.0)
Locking laravel/tailwind (v1.10.1)
Locking laravel/sanctum (v3.11.2)
Locking laravel/carbon (v2.6.1)
Locking league/commonmark (1.6.0)
Locking league/flysystem (1.1.5)
Locking league/mime-type-detection (1.7.0)
Locking mockery/mockery (1.4.3)
Locking monolog/monolog (2.3.2)
Locking mylaihs/dmmp-copy (1.10.2)
Locking rabbit/carbon (2.52.0)
Locking r看得见/php-parser (v4.12.0)
Locking numenaduro/collision (v5.9.6)
Locking opsy/closure (5.0.2)
```

```
C:\ Select Command Prompt
- Installing sebastian/object-enumerator (4.0.4): Extracting archive
- Installing sebastian/global-state (0.0.3): Extracting archive
- Installing sebastian/exporter (4.0.0): Extracting archive
- Installing sebastian/environment (5.1.3): Extracting archive
- Installing sebastian/diff (4.0.4): Extracting archive
- Installing sebastian/comparator (4.0.6): Extracting archive
- Installing sebastian/code-unit-iterator (1.0.0): Extracting archive
- Installing sebastian/code-unit-pipeline (1.0.1): Extracting archive
- Installing phpuunit/php-time (5.0.3): Extracting archive
- Installing phpuunit/php-text-template (2.0.4): Extracting archive
- Installing phpuunit/php-invoker (3.1.1): Extracting archive
- Installing phpuunit/php-file-formatter (3.0.5): Extracting archive
- Installing theseer/doctrine-annotations (1.2.1): Extracting archive
- Installing sebastian/metrics (2.0.1): Extracting archive
- Installing sebastian/compiler (3.0.3): Extracting archive
- Installing sebastian/php-metric (7.0.3): Extracting archive
- Installing phpuunit/php-code-coverage (9.2.6): Extracting archive
- Installing doctrine/instantiator (1.4.0): Extracting archive
- Installing phpspec/phpspec (1.13.0): Extracting archive
- Installing phar-io/version (3.1.0): Extracting archive
- Installing phar-io/manifest (2.0.3): Extracting archive
- Installing myclabs/deep-copy (1.10.2): Extracting archive
- Installing phpuunit/phpunit (9.5.9): Extracting archive
78 package suggestions were added by new dependencies, use "composer suggest" to see details.
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
Discovered Package: fideloper/ignition
Discovered Package: fzaninotto/faker
Discovered Package: laravel/sail
Discovered Package: laravel/sanctum
Discovered Package: laravel/tinker
Discovered Package: numenatura/collision
Package manifest generated successfully.
5 packages you are using are looking for funding.
Run `npm audit fix` to find out more.
> @php artisan vendor:publish --tag=laravel-assets --ansi
No published resources for tag [laravel-assets].
Publishing complete.
> @php artisan key:generate --ansi
Optimization key set successfully.

C:\xampp\htdocs>
```

Now the project myapp is successfully created .

```
C:\ Command Prompt
C:\xampp\htdocs>cd myapp
C:\xampp\htdocs\myapp>dir
```



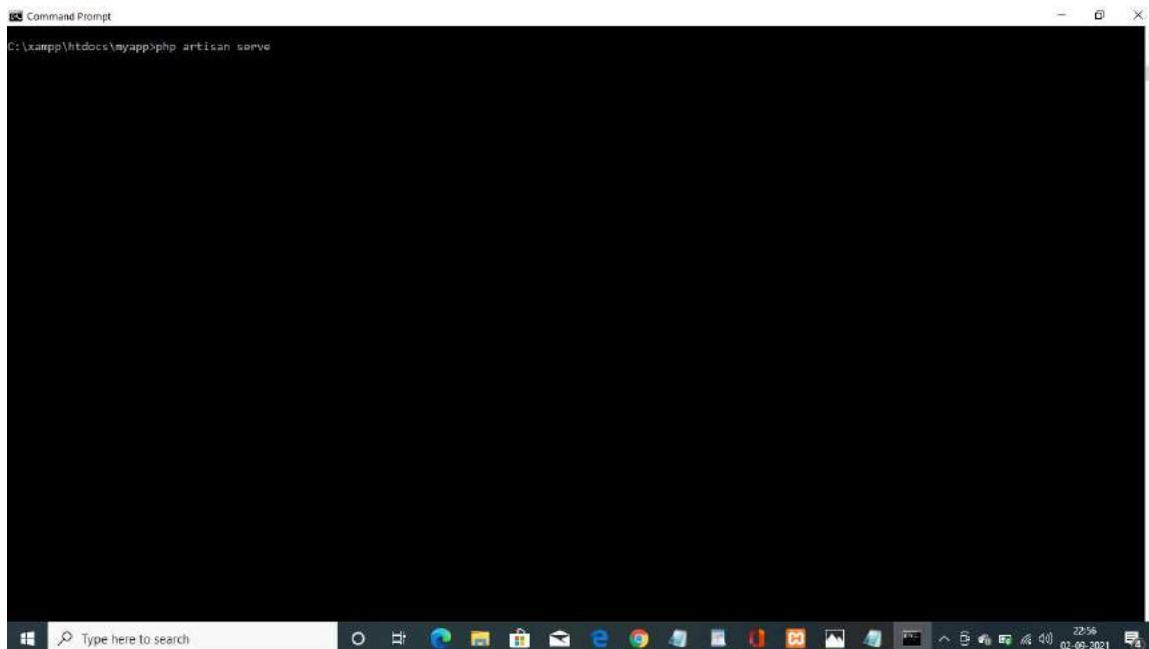
```
C:\xampp\htdocs>cd myapp
C:\xampp\htdocs\myapp>dir
Volume in drive C has no label.
Volume Serial Number is 8658-44C5

Directory of C:\xampp\htdocs\myapp

02-09-2021 22:47 <DIR> .
02-09-2021 22:47 <DIR> ..
02-09-2021 22:46 258 .editorconfig
02-09-2021 22:46 920 .env
02-09-2021 22:46 869 .env.example
02-09-2021 22:46 11 .gitattributes
02-09-2021 22:46 207 .gitignore
02-09-2021 22:46 194 .styleci.yml
02-09-2021 22:46 <DIR> app
02-09-2021 22:46 1,086 artisan
02-09-2021 22:46 <DIR> bootstrap
02-09-2021 22:46 1,735 composer.json
02-09-2021 22:47 273,378 composer.lock
02-09-2021 22:46 <DIR> config
02-09-2021 22:46 <DIR> database
02-09-2021 22:46 473 package.json
02-09-2021 22:46 1,202 phpunit.xml
02-09-2021 22:46 <DIR> public
02-09-2021 22:46 3,904 README.md
02-09-2021 22:46 <DIR> resources
02-09-2021 22:46 <DIR> routes
02-09-2021 22:46 563 server.php
02-09-2021 22:46 <DIR> storage
02-09-2021 22:46 <DIR> tests
02-09-2021 22:46 <DIR> vendor
02-09-2021 22:46 559 webpack.mix.js
14 File(s)      280,119 bytes
12 Dir(s)   389,615,968,256 bytes free

C:\xampp\htdocs\myapp>
```

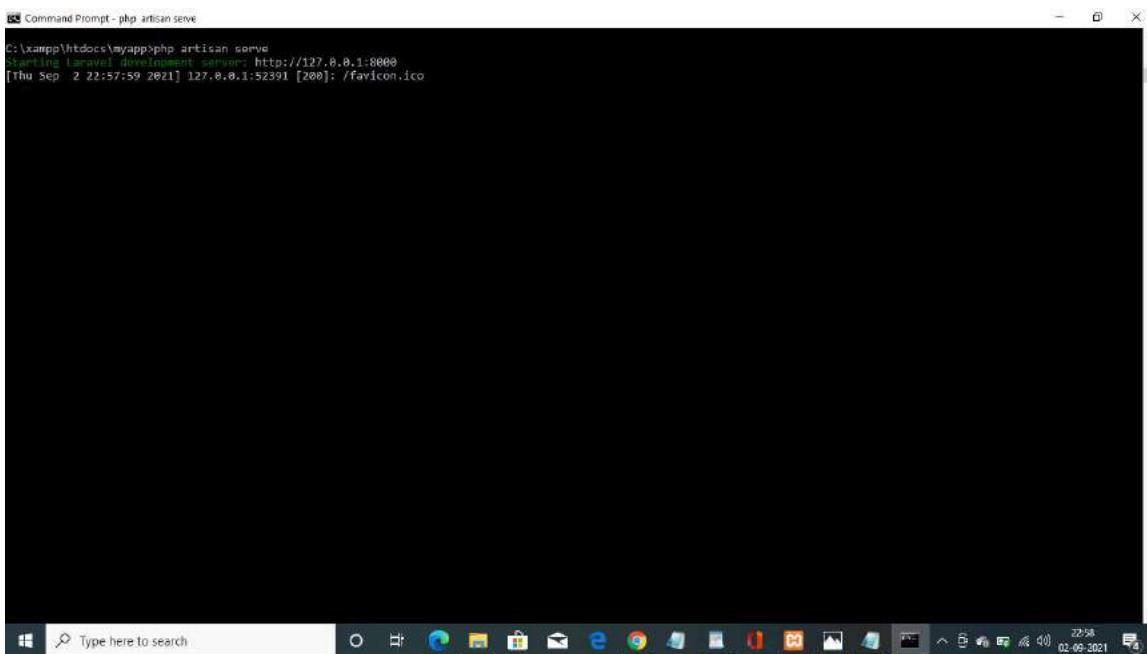
To start the app "myapp" type "php artisan myapp"



```
C:\xampp\htdocs\myapp>php artisan serve
```

To check laravel is successfully installed ,just copy the url shown

and paste it on the browser



The Laravel homepage displayed in a browser. It features the red "Laravel" logo at the top. Below the logo are four main sections: "Documentation", "Laracasts", "Laravel News", and "Vibrant Ecosystem". Each section has a brief description and a link. At the bottom of the page, there are "Shop" and "Sponsor" buttons, and a note about the Laravel version (v8.58.0) and PHP version (v7.3.27).



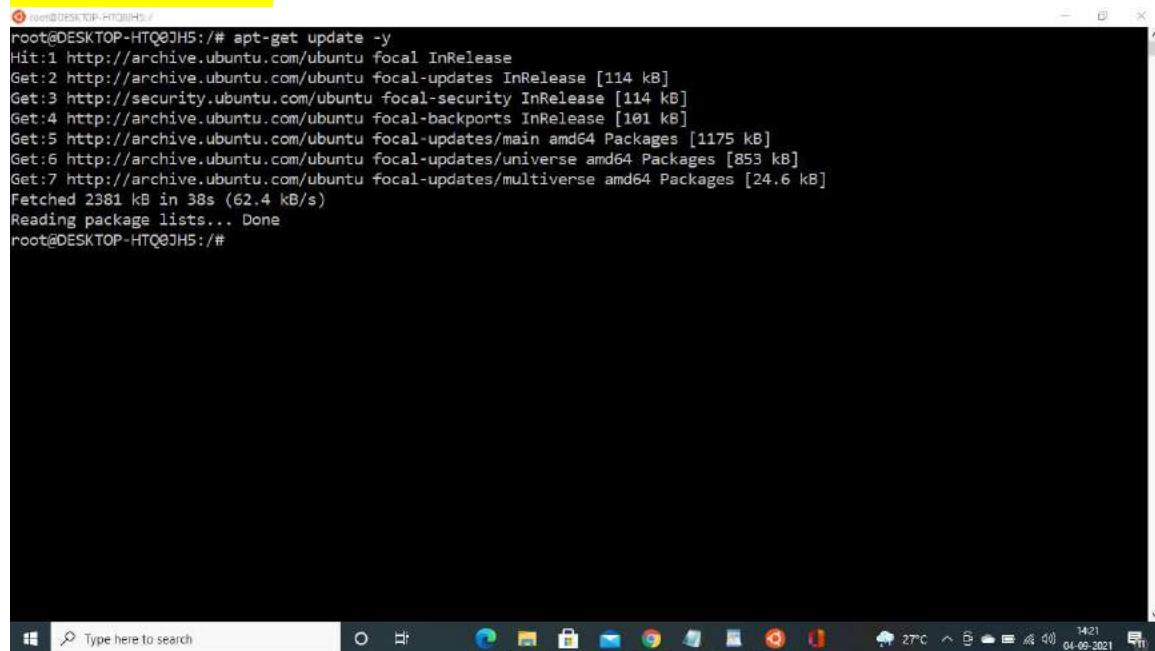
Laravel successfully installed..!!

EXPERIMENT NO: 7

AIM:-INSTALLATION OF SOFTWARE FROM THE SOURCE CODE

Step 1: Get The Server Ready

```
apt-get update -y
```



A screenshot of a Windows desktop environment showing a terminal window. The terminal window has a yellow header bar with the command 'apt-get update -y'. The main body of the window shows the output of the command:

```
root@DESKTOP-HTQ0JH5:/# apt-get update -y
Hit:1 http://archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:5 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1175 kB]
Get:6 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [853 kB]
Get:7 http://archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [24.6 kB]
Fetched 2381 kB in 38s (62.4 kB/s)
Reading package lists... Done
root@DESKTOP-HTQ0JH5:/#
```

The terminal window is titled 'Windows Terminal' and is running as root. The desktop taskbar at the bottom shows various icons for applications like File Explorer, Edge, and Google Chrome. The system tray indicates the date as 04-09-2021, the time as 14:21, and the temperature as 27°C.

make sure a compiler available

```
apt-get install build-essential -y
```

```
root@DESKTOP-HTQ0JH5:/# apt-get install build_essential -y
```

```
root@DESKTOP-HTQ0JH5:/ Setting up libisl122:amd64 (0.22.1-1) ... Setting up libbinutils:amd64 (2.34-6ubuntu1.1) ... Setting up libc-dev-bin (2.31-0ubuntu9.2) ... Setting up libalgorithm-diff-xs-perl (0.04-6) ... Setting up libcc1-0:amd64 (10.3.0-1ubuntu1~20.04) ... Setting up liblsan0:amd64 (10.3.0-1ubuntu1~20.04) ... Setting up libitm1:amd64 (10.3.0-1ubuntu1~20.04) ... Setting up gcc-9-base:amd64 (9.3.0-17ubuntu1~20.04) ... Setting up libalgorithm-merge-perl (0.08-3) ... Setting up libtsan0:amd64 (10.3.0-1ubuntu1~20.04) ... Setting up libctf0:amd64 (2.34-6ubuntu1.1) ... Setting up libasan5:amd64 (9.3.0-17ubuntu1~20.04) ... Setting up cpp-9 (9.3.0-17ubuntu1~20.04) ... Setting up libc6-dev:amd64 (2.31-0ubuntu9.2) ... Setting up binutils-x86-64-linux-gnu (2.34-6ubuntu1.1) ... Setting up binutils (2.34-6ubuntu1.1) ... Setting up dpkg-dev (1.19.7ubuntu3) ... Setting up libgcc-9-dev:amd64 (9.3.0-17ubuntu1~20.04) ... Setting up cpp (4:9.3.0-1ubuntu2) ... Setting up gcc-9 (9.3.0-17ubuntu1~20.04) ... Setting up libstdc++-9-dev:amd64 (9.3.0-17ubuntu1~20.04) ... Setting up gcc (4:9.3.0-1ubuntu2) ... Setting up g++-9 (9.3.0-17ubuntu1~20.04) ... Setting up g++ (4:9.3.0-1ubuntu2) ... update-alternatives: using /usr/bin/g++ to provide /usr/bin/c++ (c++) in auto mode Setting up build-essential (12.8ubuntu1.1) ... Processing triggers for man-db (2.9.1-1) ... Processing triggers for libc-bin (2.31-0ubuntu9.2) ... root@DESKTOP-HTQ0JH5:/#
```

Step 2: Download Dependencies

```
apt install build-essential dh-autoreconf libcurl4-gnutls-dev libexpat1-dev  
gettext libz-dev libssl-dev -y
```

```
root@DESKTOP-HTQ0JH5:/# apt install build-essential dh-autoreconf libcurl4-gnutls-dev libexpat1-dev gettext libz-dev libss^l-dev -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'zlib1g-dev' instead of 'libz-dev'
build-essential is already the newest version (12.8ubuntu1.1).
The following additional packages will be installed:
  autoconf automake autopoint autotools-dev debhelper dh-strip-nondeterminism dwz intltool-debian libarchive-cpio-perl
  libarchive-zip-perl libcroco3 libdebsupport-perl libfile-stripnondeterminism-perl libltdl-dev libmail-sendmail-perl
  libsub-override-perl libsys-hostname-long-perl libtool m4 po-debconf
Suggested packages:
  autoconf-archive gnu-standards autoconf-doc dh-make gettext-doc libasprintf-dev libgettextxpo-dev libcurl4-doc
  libgnutls28-dev libidn11-dev libkrb5-dev libldap2-dev librtmp-dev libssh2-1-dev pkg-config libtool-doc libssl-doc
  gfortran | fortran95-compiler gcj-jdk m4-doc libmail-box-perl
The following NEW packages will be installed:
  autoconf automake autopoint autotools-dev debhelper dh-autoreconf dh-strip-nondeterminism dwz gettext intltool-debian
  libarchive-cpio-perl libarchive-zip-perl libcroco3 libcurl4-gnutls-dev libdebsupport-perl libexpat1-dev
  libfile-stripnondeterminism-perl libltdl-dev libmail-sendmail-perl libssl-dev libsub-override-perl
  libsys-hostname-long-perl libtool m4 po-debconf zlib1g-dev
0 upgraded, 26 newly installed, 0 to remove and 0 not upgraded.
Need to get 6497 kB of archives.
After this operation, 25.4 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu focal/main amd64 m4 amd64 1.4.18-4 [199 kB]
Get:2 http://archive.ubuntu.com/ubuntu focal/main amd64 autoconf all 2.69-11.1 [321 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal/main amd64 autotools-dev all 20180224.1 [39.6 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal/main amd64 automake all 1:1.16.1-4ubuntu6 [522 kB]
Get:5 http://archive.ubuntu.com/ubuntu focal/main amd64 autopoint all 0.19.8.1-10build1 [412 kB]
Get:6 http://archive.ubuntu.com/ubuntu focal/main amd64 libtool all 2.4.6-14 [161 kB]
```

```
Setting up libcurl4-gnutls-dev:amd64 (7.68.0-1ubuntu2.6) ... #####
Setting up autotools-dev (20180224.1) ...#####
Setting up libexpat1-dev:amd64 (2.2.9-1build1) ...#####
Setting up libssl-dev:amd64 (1.1.1f-1ubuntu2.8) ...#####
Setting up autopoint (0.19.8.1-10build1) ...#####
Setting up libcroco3:amd64 (0.6.13-1) ...#####
Setting up autoconf (2.69-11.1) ...#####
Setting up zlib1g-dev:amd64 (1:1.2.11.dfsg-2ubuntu1.2) ...#####
Setting up dwz (0.13-5) ...#####
Setting up libarchive-cpio-perl (0.10-1) ...#####
Setting up libsub-override-perl (0.09-2) ...#####
Setting up libsys-hostname-long-perl (1.5-1) ...#####
Setting up automake (1:1.16.1-4ubuntu6) ...#####
update-alternatives: using /usr/bin/automake-1.16 to provide /usr/bin/automake (automake) in auto mode.
Setting up libfile-stripnondeterminism-perl (1.7.0-1) ...#####
Setting up gettext (0.19.8.1-10build1) ...#####
Setting up libtool (2.4.6-14) ...#####
Setting up intltool-debian (0.35.0+20060710.5) ...#####
Setting up libmail-sendmail-perl (0.80-1) ...#####
Setting up libltdl-dev:amd64 (2.4.6-14) ...#####
Setting up dh-strip-nondeterminism (1.7.0-1) ...#####
Setting up po-debconf (1.0.21) ...#####
Setting up dh-autoreconf (19) ...#####
Setting up debhelper (12.10ubuntu1) ...#####
Processing triggers for man-db (2.9.1-1) ...#####
Processing triggers for install-info (6.7.0.dfsg.2-5) ...
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
root@DESKTOP-HTQ0JH5:/#
```

Step 3: Download The Source Package

wget <https://github.com/git/git/archive/v2.23.0.tar.gz>

```
root@DESKTOP-HTQ0JH5:/# wget https://github.com/git/git/archive/v2.23.0.tar.gz
--2021-09-04 15:00:02-- https://github.com/git/git/archive/v2.23.0.tar.gz
Resolving github.com (github.com)... 13.234.176.102
Connecting to github.com (github.com)|13.234.176.102|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/git/tar.gz/v2.23.0 [following]
--2021-09-04 15:00:04-- https://codeload.github.com/git/tar.gz/v2.23.0
Resolving codeload.github.com (codeload.github.com)... 13.127.152.42
Connecting to codeload.github.com (codeload.github.com)|13.127.152.42|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/x-gzip]
Saving to: 'v2.23.0.tar.gz'

v2.23.0.tar.gz [ <=> ] 8.25M 757KB/s in 17s

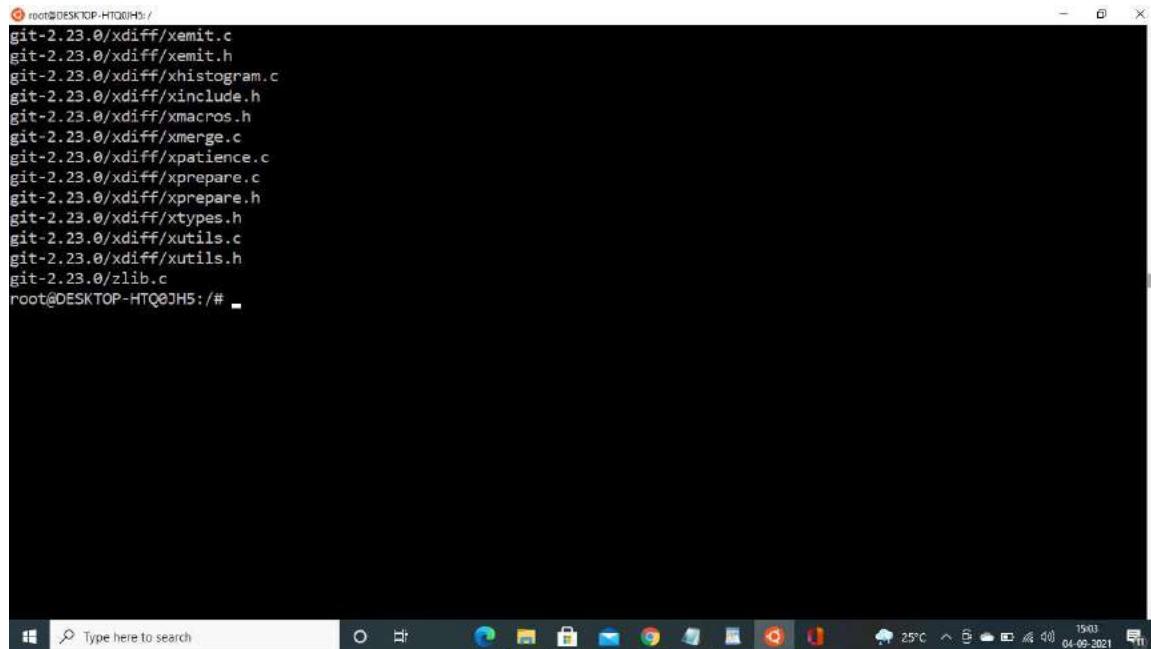
2021-09-04 15:00:21 (504 KB/s) - 'v2.23.0.tar.gz' saved [8647535]

root@DESKTOP-HTQ0JH5:/#
```

extract the archive using

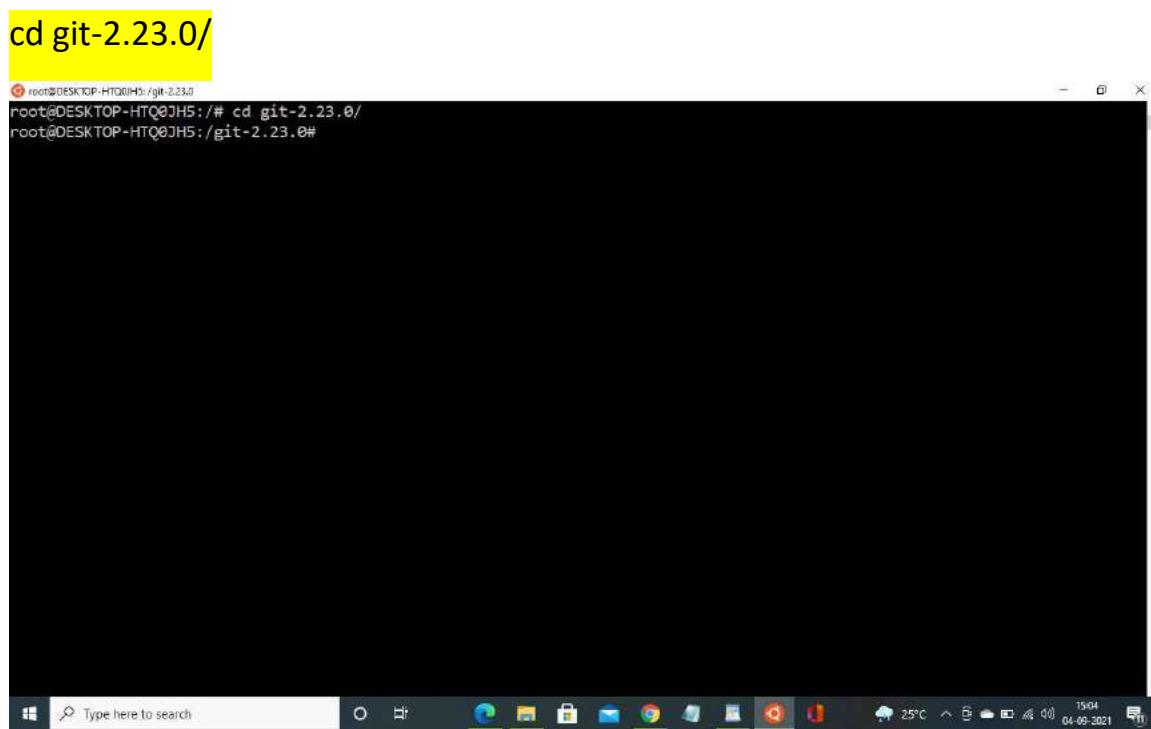
```
tar -xvf v2.23.0.tar.gz
```

```
root@DESKTOP-HTQ0JH5:/# tar -xvf v2.23.0.tar.gz
git-2.23.0/
git-2.23.0/.clang-format
git-2.23.0/.editorconfig
git-2.23.0/.gitattributes
git-2.23.0/.github/
git-2.23.0/.github/CONTRIBUTING.md
git-2.23.0/.github/PULL_REQUEST_TEMPLATE.md
git-2.23.0/.gitignore
git-2.23.0/.gitmodules
git-2.23.0/.mailmap
git-2.23.0/.travis.yml
git-2.23.0/.tsan-suppressions
git-2.23.0/COPYING
git-2.23.0/Documentation/
git-2.23.0/Documentation/.gitattributes
git-2.23.0/Documentation/.gitignore
git-2.23.0/Documentation/CodingGuidelines
git-2.23.0/Documentation/Makefile
git-2.23.0/Documentation/MyFirstContribution.txt
git-2.23.0/Documentation/RelNotes/
git-2.23.0/Documentation/RelNotes/1.5.0.1.txt
git-2.23.0/Documentation/RelNotes/1.5.0.2.txt
git-2.23.0/Documentation/RelNotes/1.5.0.3.txt
git-2.23.0/Documentation/RelNotes/1.5.0.4.txt
git-2.23.0/Documentation/RelNotes/1.5.0.5.txt
git-2.23.0/Documentation/RelNotes/1.5.0.6.txt
git-2.23.0/Documentation/RelNotes/1.5.0.7.txt
git-2.23.0/Documentation/RelNotes/1.5.0.txt
```



```
root@DESKTOP-HTQ0JH5:/  
git-2.23.0/xdiff/xemit.c  
git-2.23.0/xdiff/xemit.h  
git-2.23.0/xdiff/xhistogram.c  
git-2.23.0/xdiff/xinclude.h  
git-2.23.0/xdiff/xmacros.h  
git-2.23.0/xdiff/xmerge.c  
git-2.23.0/xdiff/xpatience.c  
git-2.23.0/xdiff/xprepare.c  
git-2.23.0/xdiff/xprepare.h  
git-2.23.0/xdiff/xtypes.h  
git-2.23.0/xdiff/xutils.c  
git-2.23.0/xdiff/xutils.h  
git-2.23.0/zlib.c  
root@DESKTOP-HTQ0JH5:/#
```

change the current working directory to the folder "git"



```
cd git-2.23.0/  
root@DESKTOP-HTQ0JH5:/git-2.23.0/  
root@DESKTOP-HTQ0JH5:/# cd git-2.23.0/  
root@DESKTOP-HTQ0JH5:/git-2.23.0#
```

Step 4: Install Git

configure the extracted package

make configure

```
root@DESKTOP-HTQ0JHS:/git-2.23.0#
root@DESKTOP-HTQ0JHS:/git-2.23.0# make configure
GIT_VERSION = 2.23.0
    GEN configure
root@DESKTOP-HTQ0JHS:/git-2.23.0#
```

verify that all of the dependencies necessary to build the package are available by running this command:

```
./configure --prefix=/usr
```

```
root@DESKTOP-HTQ0JH5:/git-2.23.0# ./configure --prefix=/usr
configure: Setting lib to 'lib' (the default)
configure: Will try -pthread then -lpthread to enable POSIX Threads.
configure: CHECKS for site configuration
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
checking for ANSI C header files... yes
checking for sys/types.h... yes
checking for sys/stat.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for memory.h... yes
checking for strings.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking for unistd.h... yes
checking for size_t... yes
checking for working alloca.h... yes
checking for alloca... yes
```

```
root@DESKTOP-HTQ0JH5:/git-2.23.0# checking for CLOCK_MONOTONIC... yes
checking for setitimer... yes
checking for library containing setitimer... none required
checking for strcasestr... yes
checking for library containing strcasestr... none required
checking for memmem... yes
checking for library containing memmem... none required
checking for strlcpy... no
checking for uintmax_t... yes
checking for strtoumax... yes
checking for library containing strtoumax... none required
checking for setenv... yes
checking for library containing setenv... none required
checking for unsetenv... yes
checking for library containing unsetenv... none required
checking for mkdtemp... yes
checking for library containing mkdtemp... none required
checking for initgroups... yes
checking for library containing initgroups... none required
checking for getdelim... yes
checking for library containing getdelim... none required
checking for BSD sysctl... no
checking for POSIX Threads with ''... no
checking for POSIX Threads with '-mt'... no
checking for POSIX Threads with '-pthread'... yes
configure: creating ./config.status
config.status: creating config.mak.autogen
config.status: executing config.mak.autogen commands
root@DESKTOP-HTQ0JH5:/git-2.23.0#
```

Build the source code:

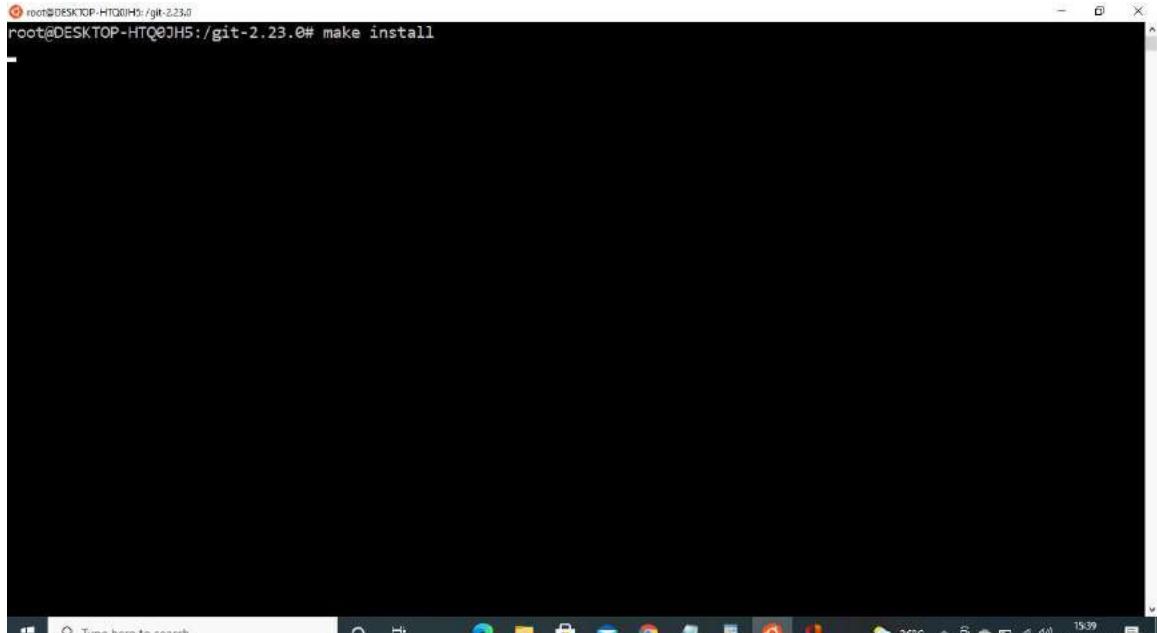
make all

```
root@DESKTOP-HTQ0JH5:/git-2.23.0# make all
 * new build flags
 CC fuzz-commit-graph.o
 CC fuzz-pack-headers.o
 CC fuzz-pack-idx.o
 CC credential-store.o
 * new link flags
 CC common-main.o
 CC abspath.o
 CC advice.o
 CC alias.o
 CC alloc.o
 CC apply.o
 CC archive.o
 CC archive-tar.o
 CC archive-zip.o
 CC argv-array.o
 * new prefix flags
 CC attr.o
 CC base85.o
 CC bisect.o
 CC blame.o
 CC blob.o
 CC branch.o
 CC bulk-checkin.o
 CC bundle.o
 CC cache-tree.o
 CC chdir-notify.o
 CC checkout.o
```

```
root@DESKTOP-HTQ0JH5:/git-2.23.0#
CC t/helper/test-scrap-cache-tree.o
CC t/helper/test-serve-v2.o
CC t/helper/test-sha1.o
CC t/helper/test-sha1-array.o
CC t/helper/test-sha256.o
CC t/helper/test-sigchain.o
CC t/helper/test-strcmp-offset.o
CC t/helper/test-string-list.o
CC t/helper/test-submodule-config.o
CC t/helper/test-submodule-nested-repo-config.o
CC t/helper/test-subprocess.o
CC t/helper/test-trace2.o
CC t/helper/test-urlmatch-normalization.o
CC t/helper/test-xml-encode.o
CC t/helper/test-wildmatch.o
CC t/helper/test-windows-named-pipe.o
CC t/helper/test-write-cache.o
LINK t/helper/test-tool
GEN bin-wrappers/git
GEN bin-wrappers/git-upload-pack
GEN bin-wrappers/git-receive-pack
GEN bin-wrappers/git-upload-archive
GEN bin-wrappers/git-shell
GEN bin-wrappers/git-cvsserver
GEN bin-wrappers/test-fake-ssh
GEN bin-wrappers/test-line-buffer
GEN bin-wrappers/test-svn-fe
GEN bin-wrappers/test-tool
```

Now that the binaries are all built, its time to install git:

make install



```

root@DESKTOP-HTQ0JH5:/git-2.23.0#
root@DESKTOP-HTQ0JH5:/git-2.23.0# make install

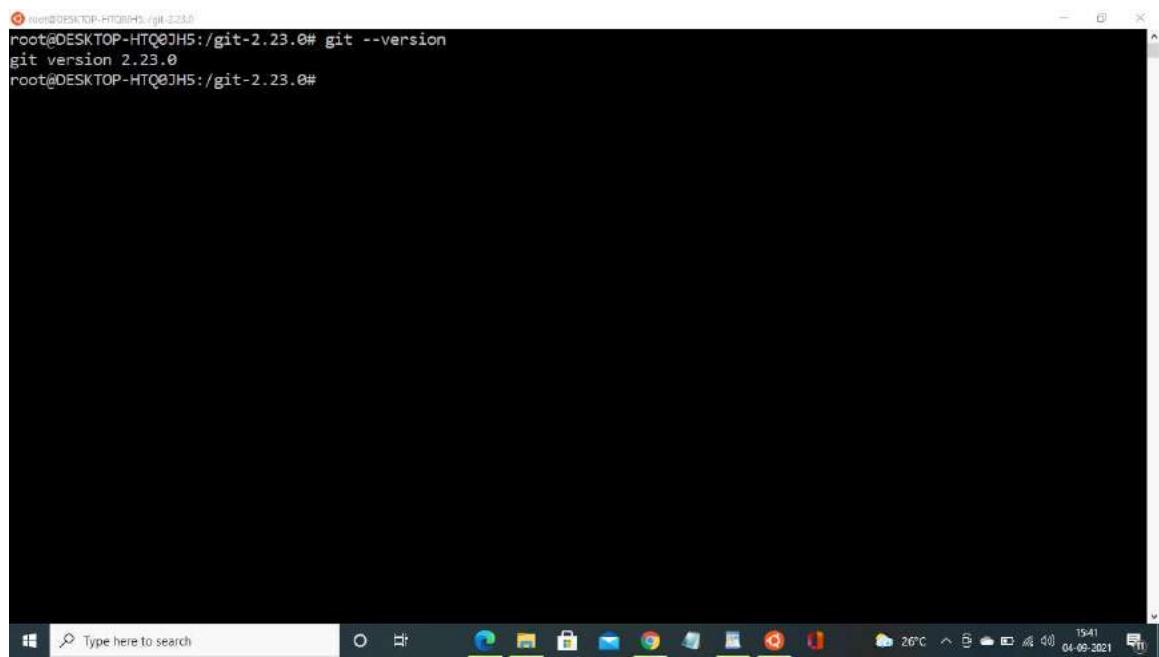
Select root@DESKTOP-HTQ0JH5:/git-2.23.0
remote_curl_aliases="git-remote-https git-remote-ftp git-remote-ftps" && \
for p in $remote_curl_aliases; do \
    rm -f "$execdir/$p" && \
    test -n "" && \
    ln -s "$destdir_from_execdir_SQ/bin/git" "$execdir/$p" || \
    { test -z "" && \
      ln "$execdir/git" "$execdir/$p" 2>/dev/null || \
      ln -s "git" "$execdir/$p" 2>/dev/null || \
      cp "$execdir/git" "$execdir/$p" || exit; } \
done && \
remote_curl_aliases="git-remote-https git-remote-ftp git-remote-ftps" && \
for p in $remote_curl_aliases; do \
    rm -f "$execdir/$p" && \
    test -n "" && \
    ln -s "git-remote-http" "$execdir/$p" || \
    { test -z "" && \
      ln "$execdir/git-remote-http" "$execdir/$p" 2>/dev/null || \
      ln -s "git-remote-http" "$execdir/$p" 2>/dev/null || \
      cp "$execdir/git-remote-http" "$execdir/$p" || exit; } \
done && \
./check_bindir "z$bindir" "z$execdir" "$bindir/git-add"
root@DESKTOP-HTQ0JH5:/git-2.23.0#
root@DESKTOP-HTQ0JH5:/git-2.23.0#

```



Verify that git is working

git --version



```
root@DESKTOP-HTQ0JHS:/git-2.23.0# git --version
git version 2.23.0
root@DESKTOP-HTQ0JHS:/git-2.23.0#
```

Successfully installed the git software from its source code!!!

EXPERIMENT NO : 8

AIM: Introduction to command line tools for networking IPv4 networking, network commands: ping route traceroute, nslookup, ip. Setting up static and dynamic IP addresses. Concept of Subnets, CIDR address schemes, Subnet masks, iptables, setting up a firewall for LAN, Application layer (L7) proxies.

PROCEDURE :

IPv4 networking

IP stands for Internet Protocol and v4 stands for Version Four (IPv4). IPv4 was the primary version brought into action for production within the ARPANET in 1983.

IP version four addresses are 32-bit integers which will be expressed in decimal notation.

Example- 192.0.2.126 could be an IPv4 address.

Parts of IPv4

1. Network part:

The network part indicates the distinctive variety that's appointed to the network. The network part conjointly identifies the category of the network that's assigned.

1. Host Part:

The host part uniquely identifies the machine on your network. This part of the IPv4 address is assigned to every host. For each host on the network, the network part is the same, however, the host half must vary.

1. Subnet number:

This is the nonobligatory part of IPv4. Local networks that have massive numbers of hosts are divided into subnets and subnet numbers are appointed to that.

Characteristics of IPv4

- i. IPv4 could be a 32-Bit IP Address.
- ii. IPv4 could be a numeric address, and its bits are separated by a dot.
- iii. The number of header fields is twelve and the length of the header field is twenty.
- iv. It has Unicast, broadcast, and multicast style of addresses.
- v. IPv4 supports VLSM (Virtual Length Subnet Mask).
- vi. IPv4 uses the Post Address Resolution Protocol to map to the MAC address.
- vii. RIP may be a routing protocol supported by the routed daemon.
- viii. Networks ought to be designed either manually or with DHCP.
- ix. Packet fragmentation permits from routers and causing host.

Advantages of IPv4

- I. IPv4 security permits encryption to keep up privacy and security.
- I. IPV4 network allocation is significant and presently has quite 85000 practical routers.
- I. It becomes easy to attach multiple devices across an outsized network while not NAT.
- I. This is a model of communication so provides quality service also as economical knowledge transfer.
- I. IPV4 addresses are redefined and permit flawless encoding.
- I. Routing is a lot of scalable and economical as a result of addressing is collective more effectively.
- I. Data communication across the network becomes a lot of specific in multicast organizations.
- I. Limits net growth for existing users and hinders the use of the net for brand new users.
- I. Internet Routing is inefficient in IPv4.

- I. IPv4 has high System Management prices and it's labor-intensive, complex, slow & frequent to errors.
- I. Security features are nonobligatory
- .
- I. Difficult to feature support for future desires as a result of adding it on is extremely high overhead since it hinders the flexibility to attach everything over IP.

Ping

The ping command is one of the most well-known tools available. Simply put, ping sends an "are you there?" message to a remote host. If the host is, in fact, there, ping returns a "yup, I'm here" message. It does this using a protocol known as ICMP, or Internet Control Message Protocol. ICMP was designed to be an error reporting protocol and has a wide variety of uses that we won't go into here.

Ping uses two ICMP message types: type 8 (Echo Request) and type 0 (Echo Reply). When you issue a ping command, the source sends an ICMP Echo Request to the destination. If the destination is available and is allowed to respond, then it replies with an ICMP Echo Reply. Once the message returns to the source, the ping command displays a success message as well as the Round Trip Time (RTT). The RTT can be an indicator of the latency between the source and the destination.

```
$ ping google.com
```

```
PING google.com (172.217.10.46): 56 data bytes
```

```
64 bytes from 172.217.10.46: icmp_seq=0 ttl=56 time=15.740 ms
64 bytes from 172.217.10.46: icmp_seq=1 ttl=56 time=14.648 ms
64 bytes from 172.217.10.46: icmp_seq=2 ttl=56 time=11.153 ms
64 bytes from 172.217.10.46: icmp_seq=3 ttl=56 time=12.577 ms
64 bytes from 172.217.10.46: icmp_seq=4 ttl=56 time=22.400 ms
64 bytes from 172.217.10.46: icmp_seq=5 ttl=56 time=12.620 ms
```

--- google.com ping statistics ---

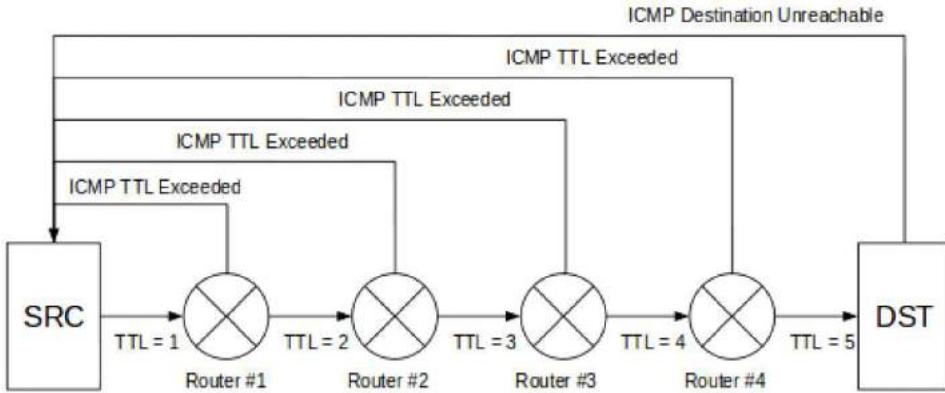
6 packets transmitted, 6 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 11.153/14.856/22.400/3.689 ms
The example above shows a ping session to google.com. From the out

put, you can see the IP address being contacted, the sequence number of each packet sent, and the round-trip time. In this case, six packets were sent with an average RTT of 14ms.

Traceroute

Traceroute is a finicky beast. This tool is meant to identify the path between a source and a destination point. The reality is mostly true, with a couple of caveats. Let's start by explaining how traceroute works:

Image



Source: [StackPath](#)

Think of traceroute as a string of ping commands. At each step along the path, traceroute identifies the hop's IP as well as the latency to that hop. But how is it finding each hop? Turns out, it's using a bit of trickery.

Traceroute uses UDP or ICMP, depending on the OS. On a typical *nix system it uses UDP and sends traffic to port 33434 by default. On a Windows system, traceroute uses ICMP. As with ping, traceroute can be blocked by not responding to the protocol/port being used.

When you invoke traceroute, you identify the destination you're trying to reach. The command begins by sending a packet to the destination, but it sets the packet's time to live (TTL) to one. This behavior is significant because the TTL value determines how many hops a packet is allowed to pass through before an ICMP Time Exceeded message is returned to the source. The trick here is to start the TTL at one and increment it by one after the ICMP message is received:

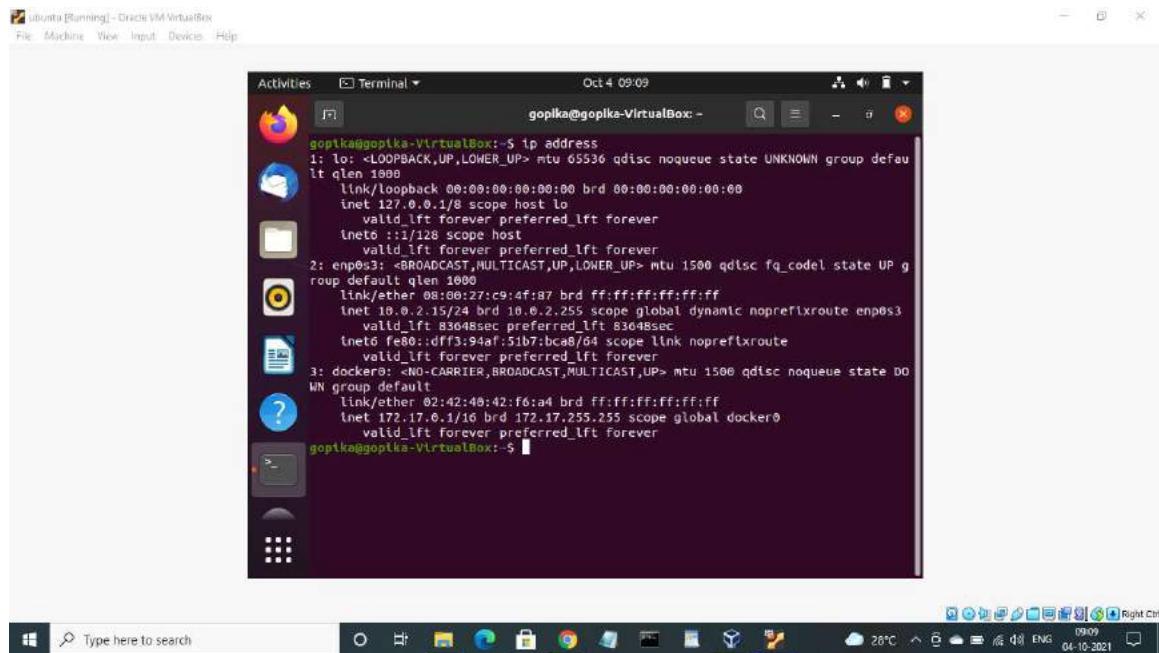
```
$ traceroute google.com
traceroute to google.com (172.217.10.46), 64 hops max, 52 byte packets
1 192.168.1.1 (192.168.1.1) 1747.782 ms 1.812 ms 4.232 ms
2 10.170.2.1 (10.170.2.1) 10.838 ms 12.883 ms 8.510 ms
3 xx.xx.xx.xx (xx.xx.xx.xx) 10.588 ms 10.141 ms 10.652 ms
4 xx.xx.xx.xx (xx.xx.xx.xx) 14.965 ms 16.702 ms 18.275 ms
5 xx.xx.xx.xx (xx.xx.xx.xx) 15.092 ms 16.910 ms 17.127 ms
```

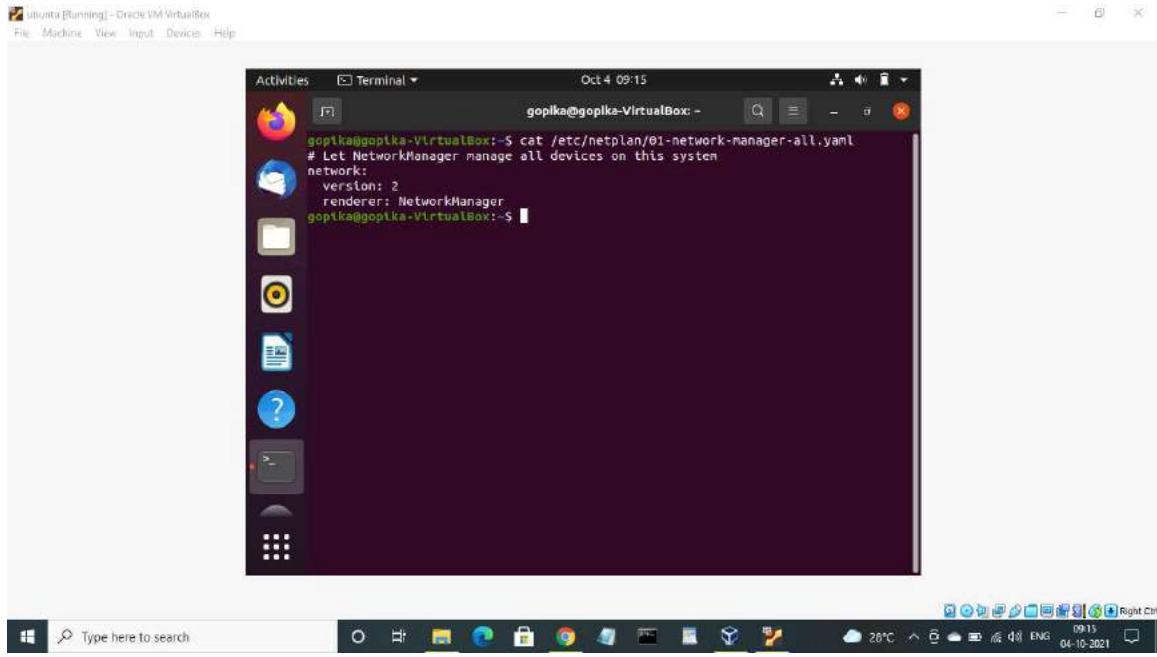
```
6 108.170.248.97 (108.170.248.97) 13.711 ms 14.363 ms 11.698 ms
7 216.239.62.171 (216.239.62.171) 12.802 ms
    216.239.62.169 (216.239.62.169) 12.647 ms 12.963 ms
8 lga34s13-in-f14.1e100.net (172.217.10.46) 11.901 ms 13.666 ms
11.813 ms
```

1. Nslookup

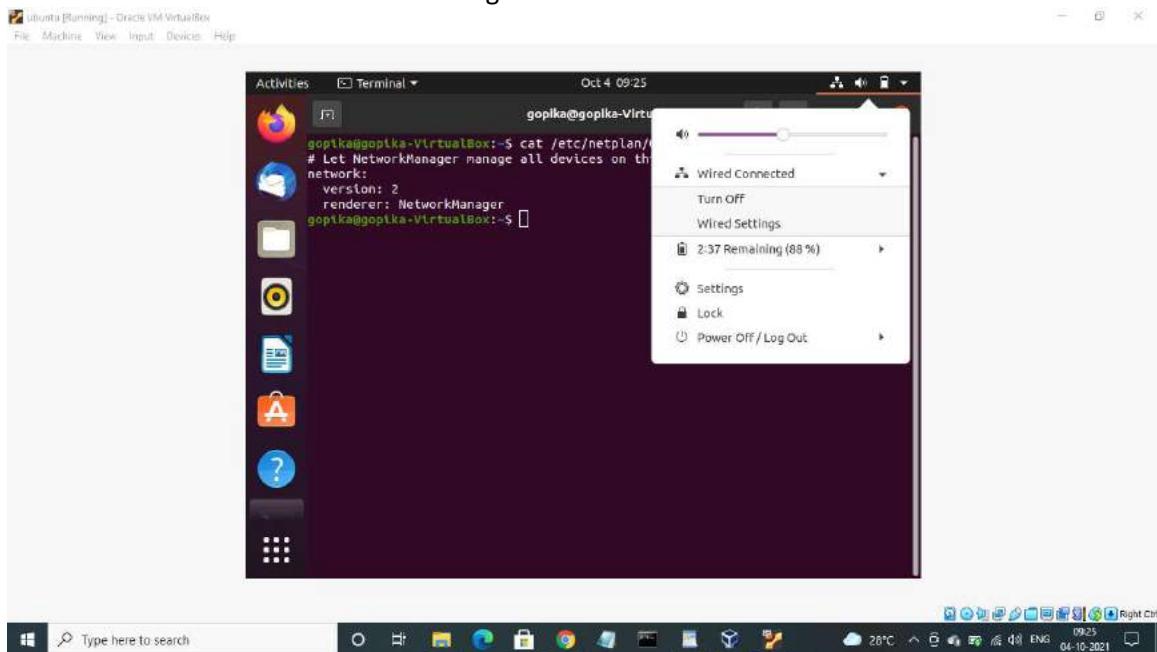
Nslookup identifies where the URL for the server points to. A simple nslookup command checks to determine what IP address the name server is pointing to for your URL. On occasion, the name servers will have conflicts, propagation errors, or specific locations that have trouble finding the server (as opposed to the server being down itself). This test would help identify if that was the case.

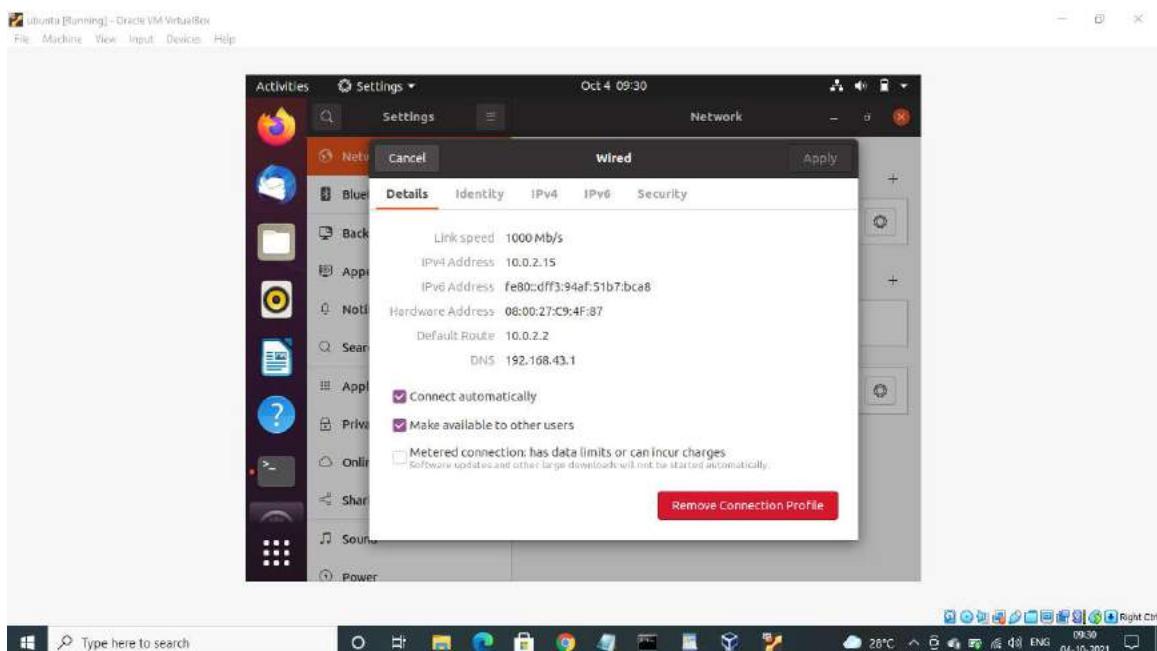
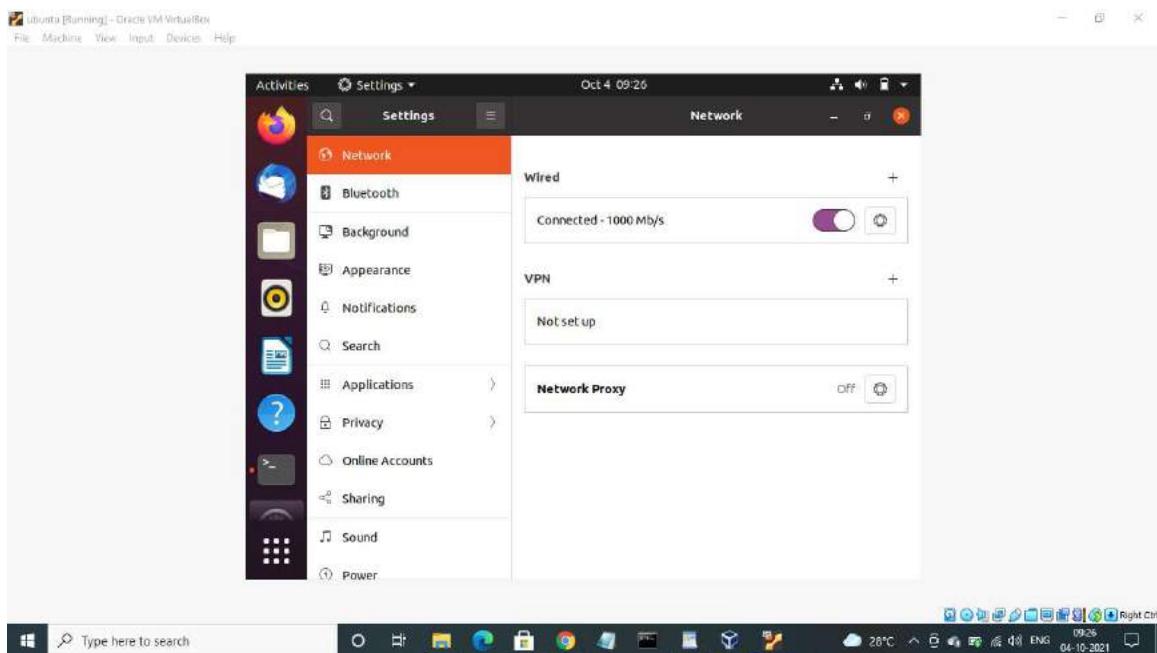
SETTING UP STATIC AND DYNAMIC IP ADDRESSS:



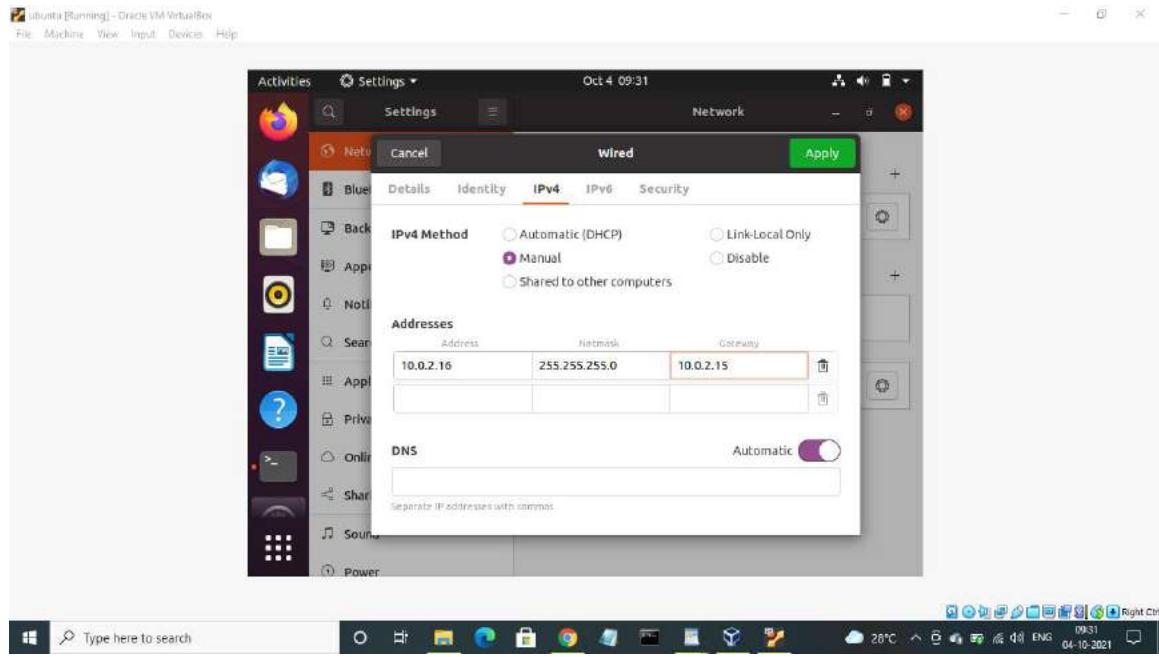


click on wired connected -->wired settings





click on IPv4 tab-->change IPv4 method from automatic to manual and follow the procedure as in the image



```
gopika@gopika-VirtualBox:~$ cat /etc/netplan/01-network-manager-all.yaml
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
gopika@gopika-VirtualBox:~$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    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:c9:4f:87 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.16/24 brd 10.0.2.255 scope global noprefixroute enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::df13:94af:51b7:bca8/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:48:42:f6:a4 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
```

Concept of subnets:

A subnet, or subnetwork, is a segmented piece of a larger network. More specifically, subnets are a logical partition of an IP network into multiple, smaller network segments. The Internet Protocol (IP) is the method for sending data from one computer to another over the internet. Each computer, or host, on the internet has at least one IP address as a unique identifier.

Organizations will use a subnet to subdivide large networks into smaller, more efficient subnetworks. One goal of a subnet is to split a large network into a grouping of smaller, interconnected networks to help minimize traffic. This way, traffic doesn't have to flow through unnecessary routes, increasing network speeds.

Subnetting, the segmentation of a network address space, improves address allocation efficiency. It is described in the formal document, Request for Comments 950, and is tightly linked to IP addresses, subnet masks and Classless Inter-Domain Routing (CIDR) notation.

How subnets work:

Each subnet allows its connected devices to communicate with each other, while routers are used to communicate between subnets. The size of a subnet depends on the connectivity requirements and the network technology employed. A point-to-point subnet allows two devices to connect, while a data center subnet might be designed to connect many more devices.

Each organization is responsible for determining the number and size of the subnets it creates, within the limits of the address space available for its use. Additionally, the details of subnet segmentation within an organization remain local to that organization.

Beneficial uses of subnets

Reallocating IP addresses:

Each class has a limited number of host allocations; for example, networks with more than 254 devices need a Class B allocation. If a network administrator is working with a Class B or C network and needs to allocate 150 hosts for three physical networks located in three different cities, they would need to either request more address blocks for each network -- or divide a network into subnets that enable administrators to use one block of addresses on multiple physical networks.

Relieving network congestion.

If much of an organization's traffic is meant to be shared regularly between the same cluster of computers, placing them on the same subnet can reduce network traffic. Without a subnet, all computers and servers on the network would see data packets from every other computer.

Improving network security.

Subnetting allows network administrators to reduce network-wide threats by quarantining compromised sections of the network and by making it more difficult for trespassers to move around an organization's network.

CIDR address schemes

What Is CIDR?

CIDR is a new addressing scheme for the Internet which allows for more efficient allocation of IP addresses

than the old Class A, B, and C address scheme.

Why Do We Need CIDR?

With a new network being connected to the Internet every 30 minutes the Internet was faced with two critical problems:

Running out of IP addresses

Running out of capacity in the global routing tables

Running Out of IP Addresses

There is a maximum number of networks and hosts that can be assigned unique addresses using the Internet's 32-bit long addresses. Traditionally, the Internet assigned "classes" of addresses: Class A, Class B and Class C were the most common. Each address had two parts: one part to identify a unique network and the second part to identify a unique host in that network. Another way the old Class A, B, and C addresses were identified was by looking at the first 8 bits of the address and converting it to its decimal equivalent.

| Address Class | # Network Bits | # Hosts Bits | Decimal Address Range |
|---------------|----------------|--------------|-----------------------|
| Class A | 8 bits | 24 bits | 1-126 |
| Class B | 16 bits | 16 bits | 128-191 |
| Class C | 24 bits | 8 bits | 192-223 |

Using the old Class A, B, and C addressing scheme the Internet could support the following:

126 Class A networks that could include up to 16,777,214 hosts each

Plus 65,000 Class B networks that could include up to 65,534 hosts each

Plus over 2 million Class C networks that could include up to 254

hosts each (Some addresses are reserved for broadcast messages, etc.). Because Internet addresses were generally only assigned in these three sizes, there was a lot of wasted addresses. For example, if you needed 100 addresses you would be assigned the smallest address (Class C), but that still meant 154 unused addresses. The overall result was that while the Internet was running out of unassigned addresses, only 3% of the assigned addresses were actually being used. CIDR was developed to be a much more efficient method of assigning addresses.

SUBNET MASKS:

The item, which is required for TCP/IP to work, is the subnet mask. The subnet mask is used by the TCP/IP protocol to determine whether a host is on the local subnet or on a remote network.

In TCP/IP, the parts of the IP address that are used as the network and host addresses aren't fixed. Unless you have more information, the network and host addresses above can't be determined. This information is supplied in another 32-bit number called a subnet mask. The subnet mask is 255.255.255.0 in this example. It isn't obvious what this number means unless you know 255 in binary notation equals 11111111. So, the subnet mask is 11111111.11111111.11111111.00000000.

Lining up the IP address and the subnet mask together, the network, and host portions of the address can be separated:

11000000.10101000.01111011.10000100 - IP address
(192.168.123.132)

11111111.11111111.11111111.00000000 - Subnet mask
(255.255.255.0)

The first 24 bits (the number of ones in the subnet

mask) are identified as the network address. The last 8 bits (the number of remaining zeros in the subnet mask) are identified as the host address. It gives you the following addresses:

11000000.10101000.0111011.00000000 - Network address
(192.168.123.0)
00000000.00000000.00000000.10000100 - Host address
(000.000.000.132)

So now you know, for this example using a 255.255.255.0 subnet mask, that the network ID is 192.168.123.0, and the host address is 0.0.0.132. When a packet arrives on the 192.168.123.0 subnet (from the local subnet or a remote network), and it has a destination address of 192.168.123.132, your computer will receive it from the network and process it.

Almost all decimal subnet masks convert to binary numbers that are all ones on the left and all zeros on the right. Some other common subnet masks are:

| Decimal | Binary | |
|-----------------|---------------------------------|-----------------|
| 255.255.255.192 | 1111111.1111111.111111.11000000 | 255.255.255.224 |
| | 1111111.1111111.111111.11100000 | |

Internet RFC 1878 (available from [InterNIC-Public Information Regarding Internet Domain Name Registration Services](#)) describes the valid subnets and subnet masks that can be used on TCP/IP networks.

Iptables:

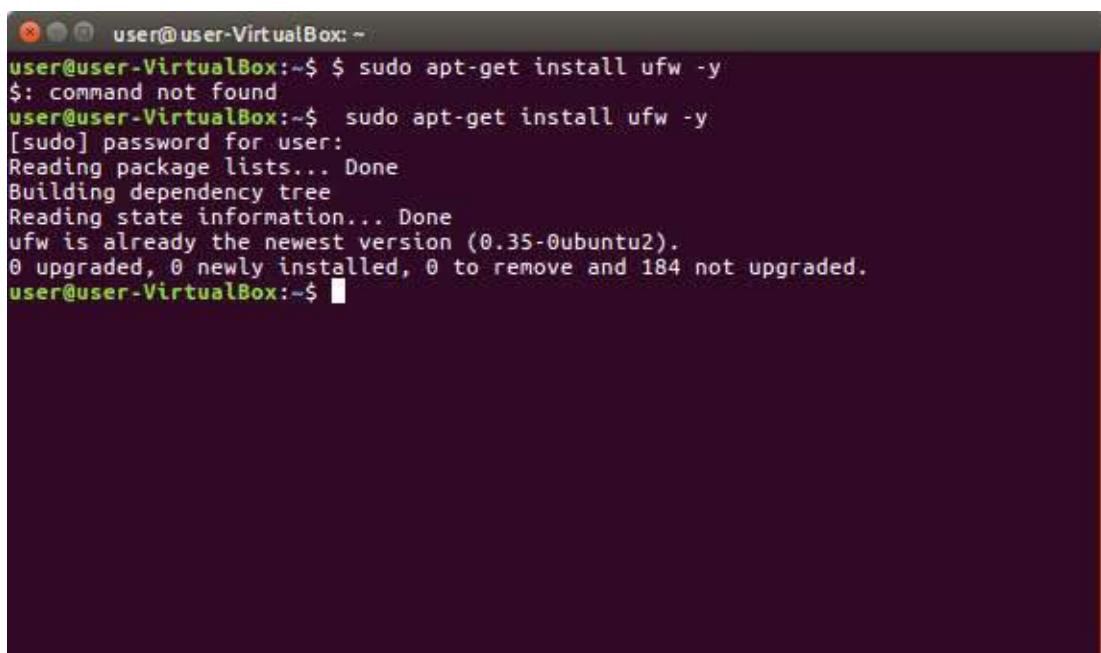
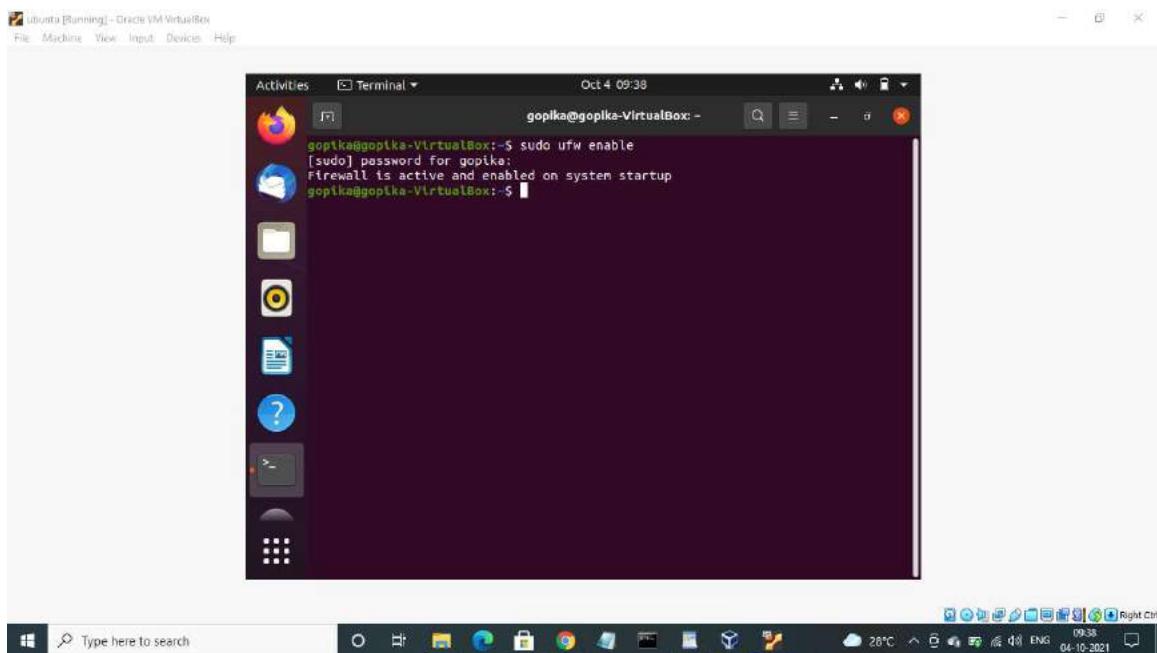
Iptables is a user-space utility program that allows a system administrator to configure the IP packet filter rules of the Linux kernel firewall, implemented as different Netfilter modules. The filters are organized in different tables, which contain chains of rules for how to treat network

traffic packets. Different kernel modules and programs are currently used for different protocols; iptables applies to IPv4, ip6tables to IPv6, arptables to ARP, and ebttables to Ethernet frames.

iptables requires elevated privileges to operate and must be executed by user root, otherwise it fails to function. On most Linux systems, iptables is installed as /usr/sbin/iptables and documented in its man pages, which can be opened using man iptables when installed. It may also be found in /sbin/iptables, but since iptables is more like a service rather than an "essential binary", the preferred location remains /usr/sbin.

The term iptables is also commonly used to inclusively refer to the kernel-level components. x_tables is the name of the kernel module carrying the shared code portion used by all four modules that also provides the API used for extensions; subsequently, Xtables is more or less used to refer to the entire firewall (v4, v6, arp, and eb) architecture

SETTING UP A FIREWALL FOR LAN:



```
user@user-VirtualBox:~$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
user@user-VirtualBox:~$ sudo ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
user@user-VirtualBox:~$
```

```
user@user-VirtualBox:~$ sudo ufw allow from 192.168.100.1 to any port 8080
Rule added
user@user-VirtualBox:~$
```

```
user@user-VirtualBox:~$ sudo ufw allow from 192.168.100.1
Rule added
user@user-VirtualBox:~$
```

```
user@user-VirtualBox:~$ sudo ufw allow 500:800/tcp
Rule added
Rule added (v6)
user@user-VirtualBox:~$
```

```
user@user-VirtualBox:~  
Could not delete non-existent rule (v6)  
user@user-VirtualBox:~$ sudo ufw status numbered  
Status: active  
  
 To           Action    From  
 --          -----  
 [ 1] 22        ALLOW IN  Anywhere  
 [ 2] 1022      ALLOW IN  Anywhere  
 [ 3] 80        DENY IN   Anywhere  
 [ 4] 443       ALLOW IN  Anywhere  
 [ 5] 21/tcp     ALLOW IN  Anywhere  
 [ 6] 500:800/tcp ALLOW IN  Anywhere  
 [ 7] Anywhere   ALLOW IN  192.168.100.1  
 [ 8] 8080      ALLOW IN  192.168.100.1  
 [ 9] Anywhere   ALLOW IN  192.168.0.0/24  
[10] Anywhere   DENY IN   192.168.2.1  
[11] 22 (v6)    ALLOW IN  Anywhere (v6)  
[12] 1022 (v6)  ALLOW IN  Anywhere (v6)  
[13] 80 (v6)    DENY IN   Anywhere (v6)  
[14] 443 (v6)   ALLOW IN  Anywhere (v6)  
[15] 21/tcp (v6) ALLOW IN  Anywhere (v6)  
[16] 500:800/tcp (v6) ALLOW IN  Anywhere (v6)  
  
user@user-VirtualBox:~$
```

```
user@user-VirtualBox:~  
user@user-VirtualBox:~$ sudo ufw deny from 192.168.2.1  
Rule added  
user@user-VirtualBox:~$
```

```
user@user-VirtualBox:~$ sudo ufw delete 14  
Deleting:  
 allow 443  
Proceed with operation (y|n)?
```

EXPERIMENT NO: 9

AIM :Analyzing network packet stream using tcpdump and wireshark. Perform basic network service tests using nc.

PROCEDURE :

Wireshark

Wireshark is a free and open-source packet analyzer. It is used for network troubleshooting, analysis, software and communications protocol development, and education. Originally named Ethereal, the project was renamed Wireshark in May 2006 due to trademark issues.

Wireshark is cross-platform, using the Qt widget toolkit in current releases to implement its user interface, and using pcap to capture packets; it runs on Linux, macOS, BSD, Solaris, some other Unix-like operating systems, and Microsoft Windows. There is also a terminal-based (non-GUI) version called TShark. Wireshark, and the other programs distributed with it such as TShark, are free software, released under the terms of the GNU General Public License version 2 or any later version.

Features:

Wireshark is a data capturing program that "understands" the structure (encapsulation) of different networking protocols. It can parse and display the fields, along with their meanings as specified by different networking protocols. Wireshark uses pcap to capture

packets, so it can only capture packets on the types of networks that pcap supports.

- Data can be captured "from the wire" from a live network connection or read from a file of already-captured packets.
- Live data can be read from different types of networks, including Ethernet, IEEE 802.11, PPP, and loopback.
- Captured network data can be browsed via a GUI, or via the terminal (command line) version of the utility, TShark.
- Captured files can be programmatically edited or converted via command-line switches to the "editcap" program.
- Data display can be refined using a display filter.
- Plug-ins can be created for dissecting new protocols.
- VoIP calls in the captured traffic can be detected. If encoded in a compatible encoding, the media flow can even be played.
- Raw USB traffic can be captured.
- Wireless connections can also be filtered as long as they traverse the monitored Ethernet.

INSTALLATION OF WIRESHARK

1. The first step is to installing the wireshark on to your ubuntu using this commands

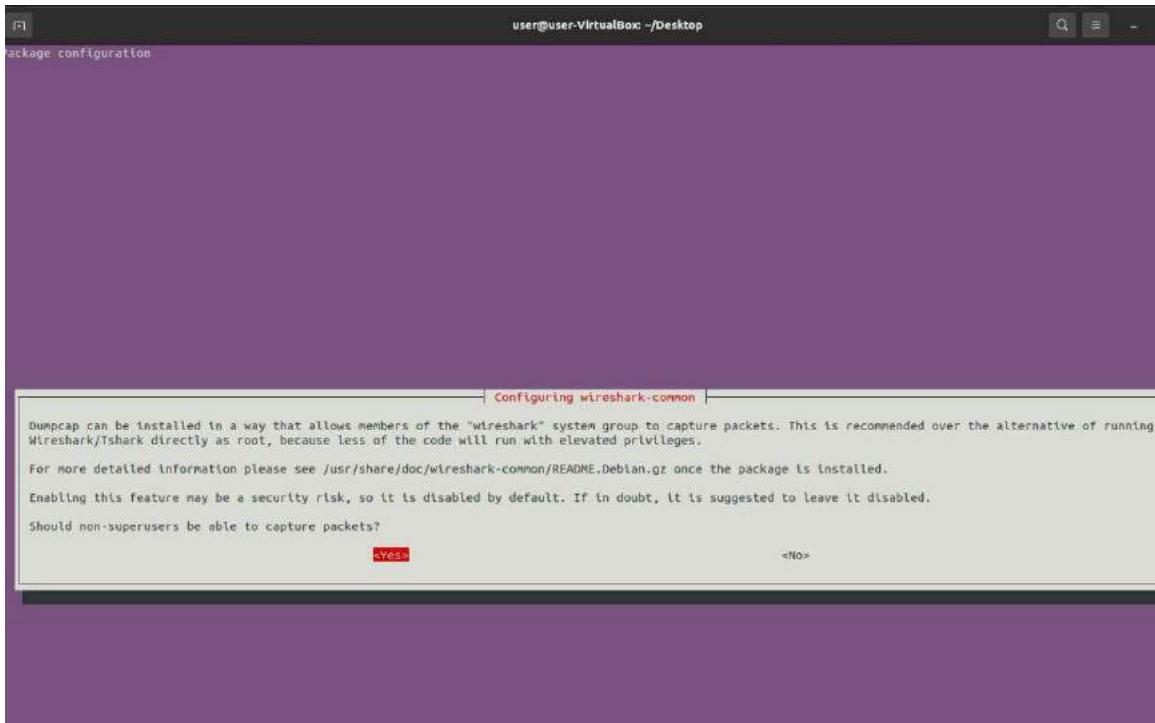
Sudo apt-get install wireshark

```
[user@user-VirtualBox:~/Desktop]$ sudo apt-get install wireshark
[sudo] password for user:
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:
  liblrc-ares2 libldouble-conversion3 liblbusa5.2-0 liblpcr2-16-0 liblqt5core5a liblqt5dbus5 liblqt5guil5 liblqt5multimedia5 liblqt5multimedia5-plugins
  liblqt5multimediasupports5 liblqt5multimedialawidgets5 liblqt5networks liblqt5opengl5 liblqt5printsupports5 liblqt5svg5 liblqt5widgets5 liblsmi2lidl
  liblspandsp2 liblwresarchardata liblwresarchard13 libwiretap10 libwsutil11 libxcb-xinerama0 libxcb-xinput0 qt5-gtk-platformtheme qtreations5-lie
  wireshark-common wireshark-qt
Suggested packages:
  qt5-image-formats-plugins qtwaylands snmp-mibs-downloader geoipupdate geoip-database geoip-database-extra libjs-leaflet
  libjs-leaflet.markercluster wireshark-doc
The following NEW packages will be installed:
  liblrc-ares2 libldouble-conversion3 liblbusa5.2-0 liblpcr2-16-0 liblqt5core5a liblqt5dbus5 liblqt5guil5 liblqt5multimedia5 liblqt5multimedia5-plugins
  liblqt5multimediasupports5 liblqt5multimedialawidgets5 liblqt5networks liblqt5opengl5 liblqt5printsupports5 liblqt5svg5 liblqt5widgets5 liblsmi2lidl
  liblspandsp2 liblwresarchardata liblwresarchard13 libwiretap10 libwsutil11 libxcb-xinerama0 libxcb-xinput0 qt5-gtk-platformtheme qtreations5-lie
  wireshark wireshark-common wireshark-qt
0 upgraded, 29 newly installed, 0 to remove and 60 not upgraded.
Need to get 32.7 MB of archives.
After this operation, 163 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/universe amd64 libldouble-conversion3 amd64 3.1.5-4ubuntu1 [37.9 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 liblpcr2-16-0 amd64 10.34.7 [181 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 liblqt5core5a amd64 5.12.8+dfsg-0ubuntu1 [2,005 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 liblqt5dbus5 amd64 5.12.8+dfsg-0ubuntu1 [208 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 liblqt5networks5 amd64 5.12.8+dfsg-0ubuntu1 [674 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu focal/main amd64 libxcb-xinerama0 amd64 1.14-2 [5,260 B]
Get:7 http://in.archive.ubuntu.com/ubuntu focal/main amd64 libxcb-xinput0 amd64 1.14-2 [29.3 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 liblqt5guil5 amd64 5.12.8+dfsg-0ubuntu1 [2,971 kB]
Get:9 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 liblqt5widgets5 amd64 5.12.8+dfsg-0ubuntu1 [2,293 kB]
Get:10 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 liblqt5svg5 amd64 5.12.8+dfsg-0ubuntu1 [131 kB]
Get:11 http://in.archive.ubuntu.com/ubuntu focal/main amd64 liblbusa5.2-0 amd64 5.2.4-1.1build3 [106 kB]
Get:12 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 liblqt5multimedias5 amd64 5.12.8-0ubuntu1 [263 kB]
Get:13 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 liblqt5opengl5 amd64 5.12.8+dfsg-0ubuntu1 [136 kB]
Get:14 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 liblqt5multimedialawidgets5 amd64 5.12.8-0ubuntu1 [36.8 kB]
Get:15 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 liblqt5multimediasupports5 amd64 5.12.8-0ubuntu1 [104 kB]
Get:16 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 liblqt5multimedias5-plugins amd64 5.12.8-0ubuntu1 [197 kB]
Get:17 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 liblsmi2lidl amd64 0.4.8+dfsg-2 [193 kB]
Get:18 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 liblsmi2lidl amd64 0.4.8+dfsg-2 [100 kB]
Get:19 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 liblspandsp2 amd64 0.8.6+dfsg-2 [272 kB]
Get:20 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 liblwresarchardata all 3.2.3-1 [1,456 kB]
Get:21 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 liblrc-ares2 amd64 1.15.0-1ubuntu0.1 [38.2 kB]
```

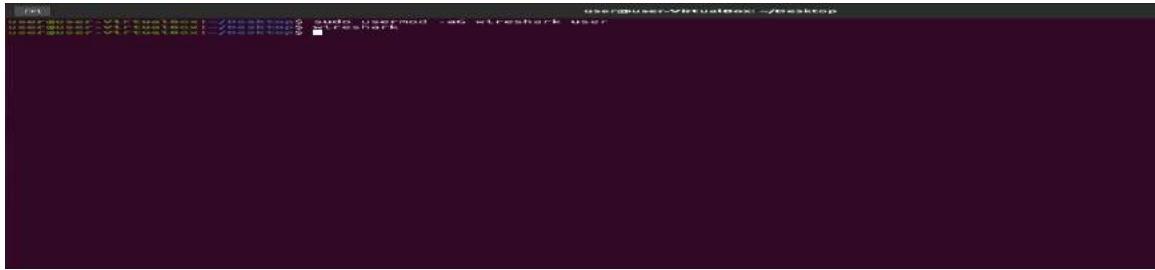
1. After installing the wireshark it reconfigure use the commands

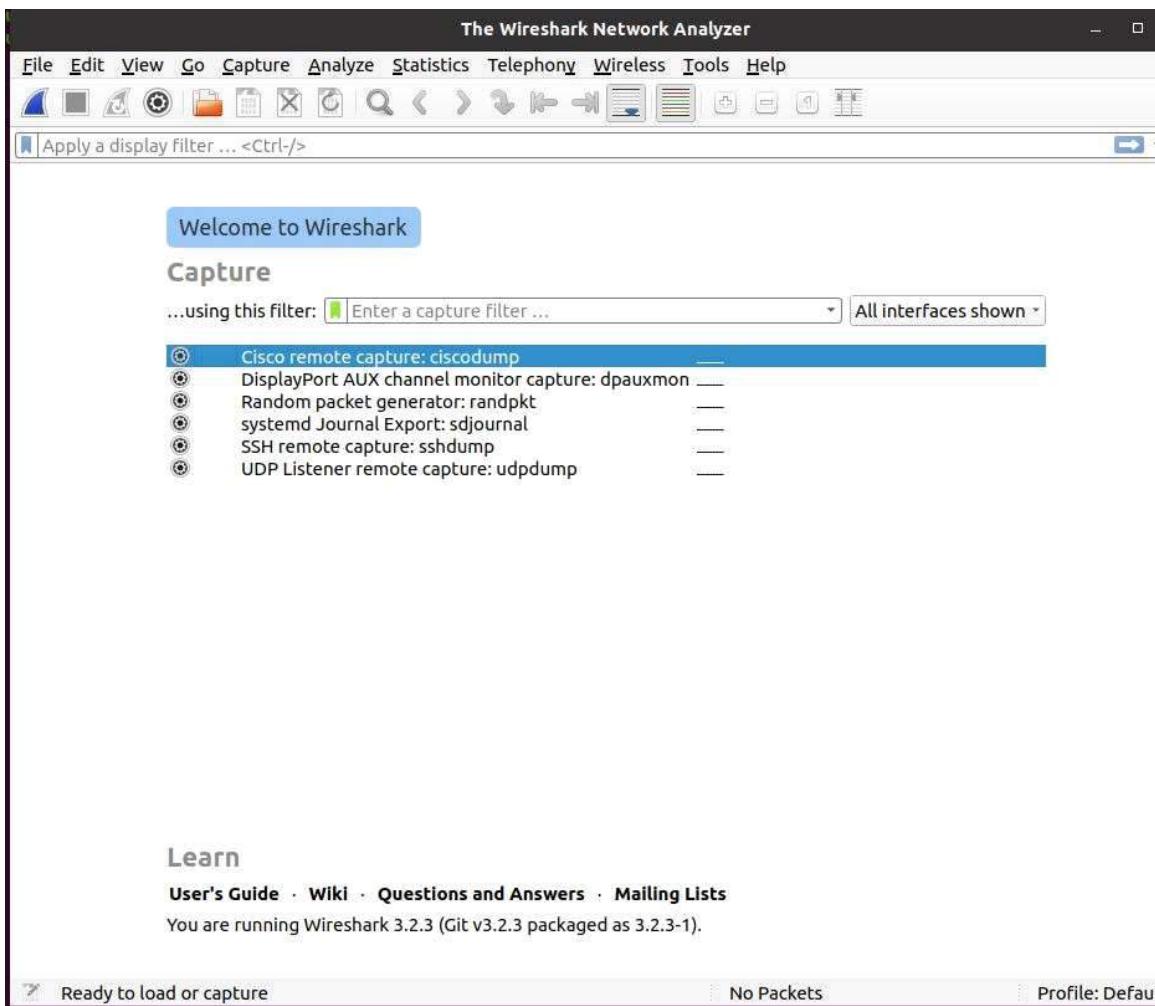
Sudo dpkg --configure -a

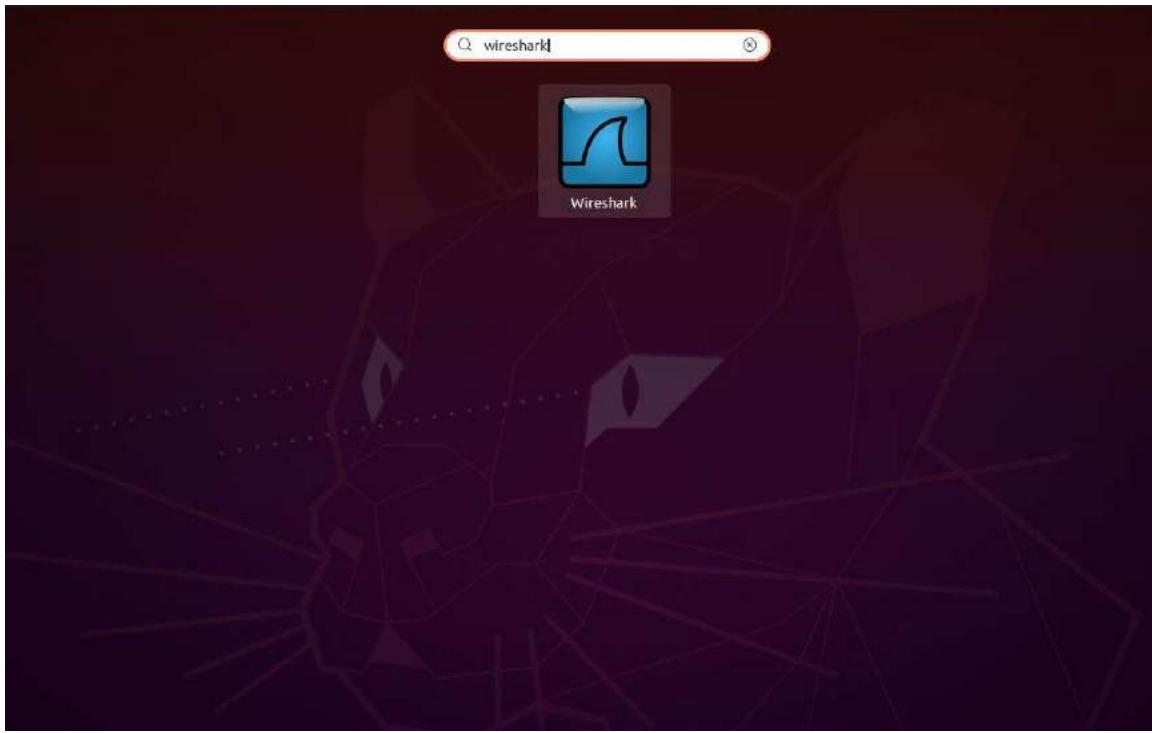
```
user@user-VirtualBox:~$ sudo dpkg --configure -a
Setting up libqt5multimediawidgets5:amd64 (5.5.1-4ubuntu2) ...
Setting up libnghttp2-14:amd64 (1.7.1-1) ...
Setting up libqgsttools-p1:amd64 (5.5.1-4ubuntu2) ...
Setting up libssh-gcrypt-4:amd64 (0.6.3-4.3ubuntu0.6) ...
Setting up libspandsp2:amd64 (0.0.6-2.1) ...
Setting up libnl-route-3-200:amd64 (3.2.27-1ubuntu0.16.04.1) ...
Setting up libsmi2ldbl:amd64 (0.4.8+dfsg2-11) ...
Setting up libqt5multimedia5-plugins:amd64 (5.5.1-4ubuntu2) ...
Setting up libmaxminddb0:amd64 (1.0.4-2.1) ...
Processing triggers for man-db (2.7.5-1) ...
Processing triggers for libc-bin (2.23-0ubuntu11.2) ...
user@user-VirtualBox:~$
```



1. Then add user to the wireshark







TCPDUMP

The packet capture utility used by tcpdump is provided by libpcap, which is a C/C++ library of procedures. The main tcpdump program is the interface for the packet capture process. When run, it will start the libpcap process to capture network packets and then display their contents on the screen. Unless a limit to the number of packets to be captured is specified when the program starts, it will continue to run forever. The processing is then terminated by an interrupt signal (Control-C).

The program is run at the command line and includes several options, which are indicated by flags. These flags alter the behavior of the program to get it to select packets that match a specified pattern, limit its running time, or get it to read stored packets from a file rather than from the network interface.

Tcpdump format

The tcpdump command can be issued by itself or with options, parameters, and/or regular expressions. None of these elements are mandatory and the order is not important.

```
tcpdump <-option_identifier> <option_name> <parameter>
<parameter_value> <regular expressions>
```

Tcpdump options

The command tcpdump is followed by options, which are also known as flags. Each of these is denoted by a hyphen followed by a letter. Below is a list of each of these flags.

- A print packets in ASCII without the link-level headers
- b show the AS number in ASDOT format
- B buffer_size in units of KiB (1024 bytes)
- c count – the limit of packets to capture
- C file_size – the process will create a new file once this file size limit is filled; Size is x 1 million bytes

- d Dump the compiled packet-matching code in ASCII
- dd Dump packet-matching code as a C program fragment
- ddd Dump packet-matching code as decimal numbers preceded with a count
- D List all accessible interfaces
- e Print the link-level header on each dump line
- E spi@ipaddr algo:secret – for decrypting IPsec ESP packets:

EXPERIMENT NO :10

AIM :Introduction to Hypervisors and VMs, Xen or KVM ,Introduction to Containers: Docker, installation and deployment.

PROCEDURE:

HYPERVISORS

A hypervisor, also known as a virtual machine monitor or VMM, is software that creates and runs virtual machines (VMs). A hypervisor allows one host computer to support multiple guest VMs by virtually sharing its resources, such as memory and processing.

Why use a hypervisor?

Hypervisors make it possible to use more of a system's available resources and provide greater IT mobility since the guest VMs are independent of the host hardware. This means they can be easily moved between different servers. Because multiple virtual machines can run off of one physical server with a hypervisor, a hypervisor reduces:

Space
Energy
Maintenance requirements

Types of hypervisors:

There are two main hypervisor types, referred to as “Type 1” (or “bare metal”) and “Type 2” (or “hosted”).

A type 1 :

hypervisor acts like a lightweight operating system and runs directly on the host’s hardware

A type 2 :

hypervisor runs as a software layer on an operating system, like other computer programs.

The most commonly deployed type of hypervisor is the type 1 or bare-metal hypervisor, where virtualization software is installed directly on the hardware where the operating system is normally installed. Because bare-metal hypervisors are isolated from the attack-prone operating system, they are extremely secure. In addition, they generally perform better and more efficiently than hosted hypervisors. For these reasons, most enterprise companies choose bare-metal hypervisors for [data center](#) computing needs.

While bare-metal hypervisors run directly on the computing hardware, hosted hypervisors run on top of the operating system (OS) of the host machine. Although hosted hypervisors run within the OS, additional (and different) operating systems can be installed on top of the hypervisor. The downside of hosted hypervisors is that latency is higher than bare-metal hypervisors. This is because communication between the hardware and the hypervisor must pass through the extra layer of the OS. Hosted hypervisors are sometimes known as client hypervisors

because they are most often used with end users and software testing, where higher latency is less of a concern.

Hardware acceleration technology can create and manage virtual resources faster by boosting processing speed for both bare-metal and hosted hypervisors. A type of hardware accelerator known as a virtual Dedicated Graphics Accelerator (vDGA) takes care of sending and refreshing high-end 3-D graphics. This frees up the main system for other tasks and greatly increases the display speed of images. For industries such as oil and gas exploration, where there is a need to quickly visualize complex data, this technology can be very useful.

INTRODUCTION TO CONTAINERS

DOCKER :

Containerization is an approach to software development in which an application or service, its dependencies, and its configuration (abstracted as deployment manifest files) are packaged together as a container image. The containerized application can be tested as a unit and deployed as a container image instance to the host operating system (OS).

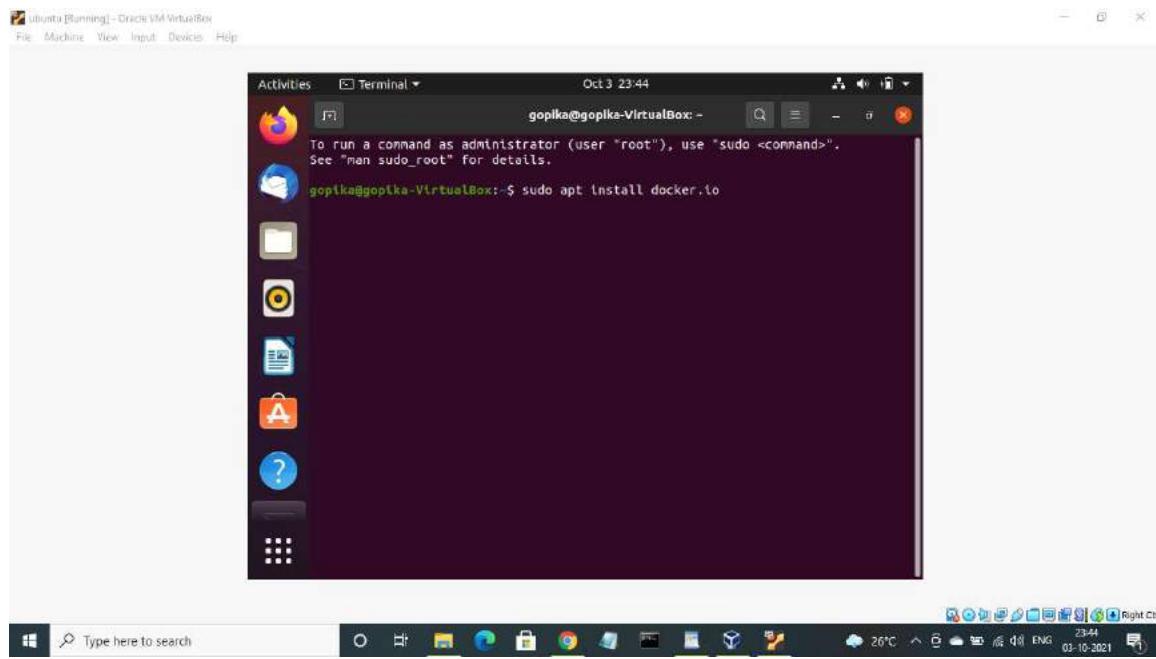
Just as shipping containers allow goods to be transported by ship, train, or truck regardless of the cargo inside, software containers act as a standard unit of software deployment that can contain different code and dependencies. Containerizing software this way enables developers and IT professionals to deploy them across environments with little or no modification.

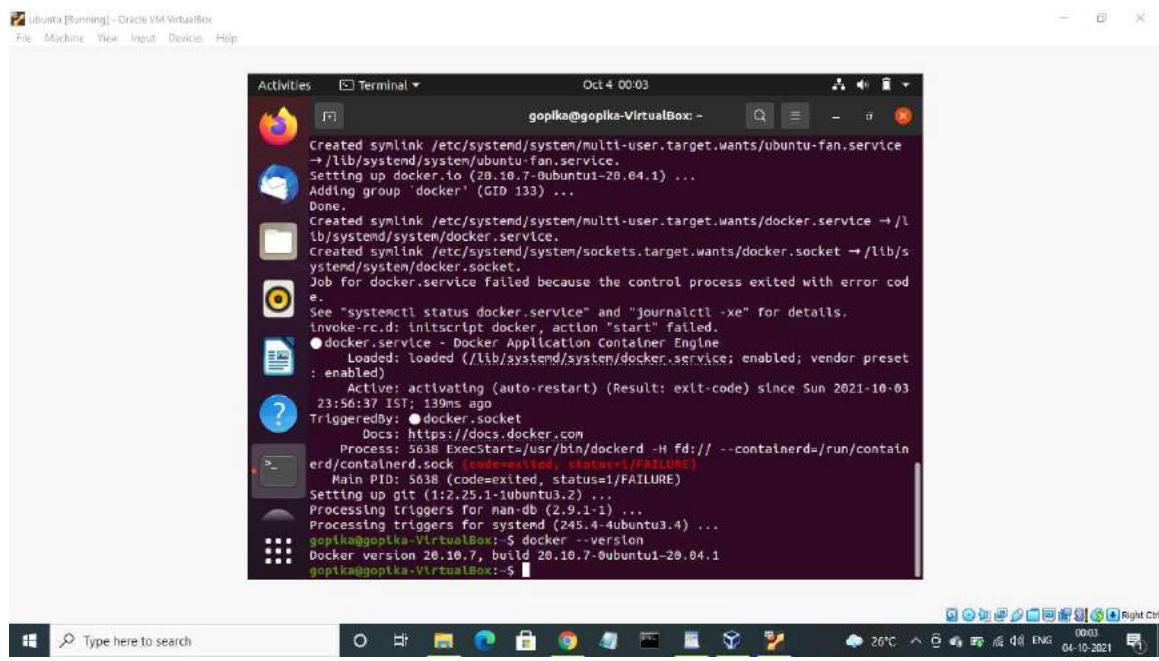
Containers also isolate applications from each other on a shared OS. Containerized applications run on top of a container host that in turn runs on the OS (Linux or Windows). Containers therefore have a significantly smaller footprint than virtual machine (VM) images.

INSTALLATION OF DOKER

1. First we can install the docker into our ubuntu using this command

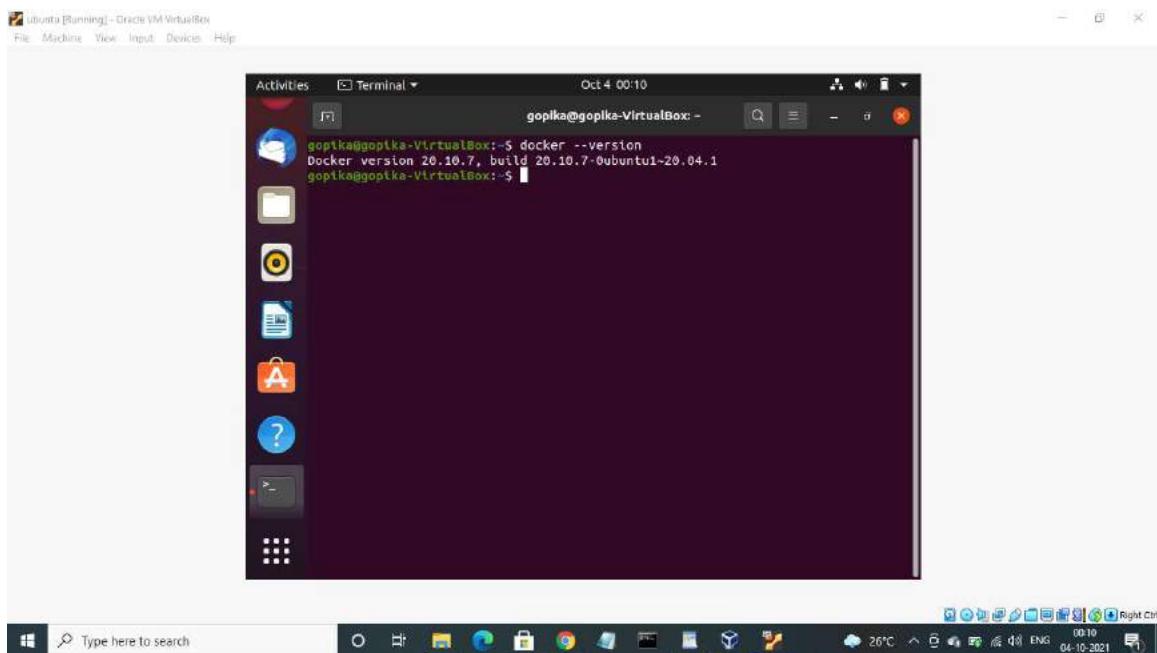
Sudo apt install docker.io





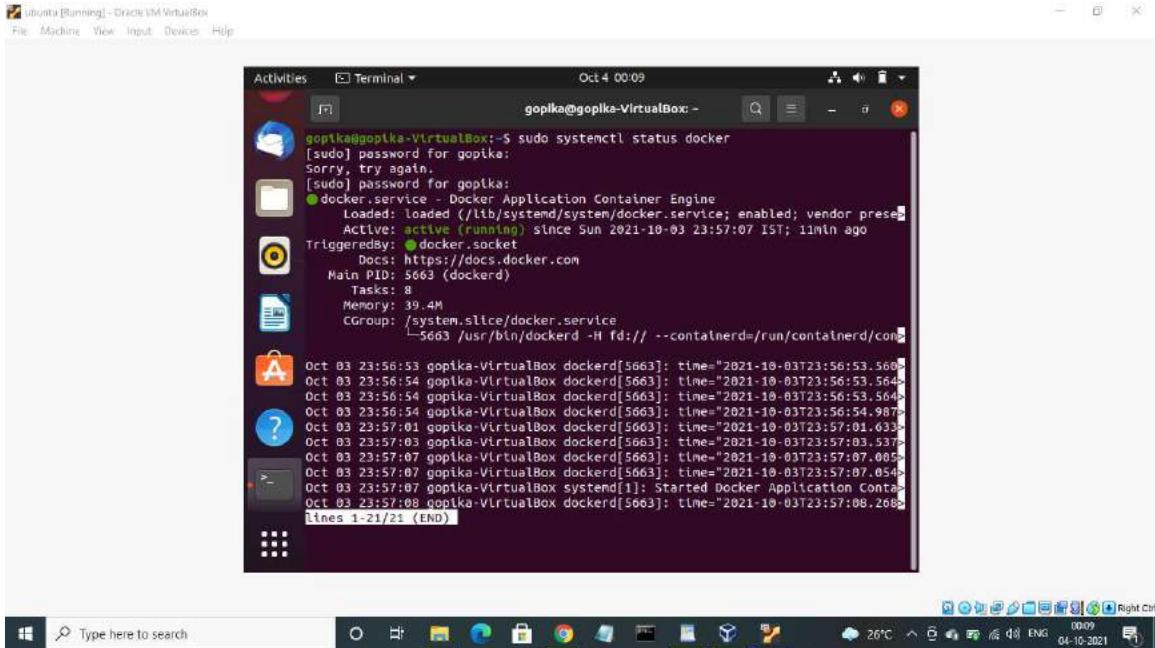
1. After installing docker check the version

Docker --version



1. Then check the status of docker

Sudo systemctl status docker

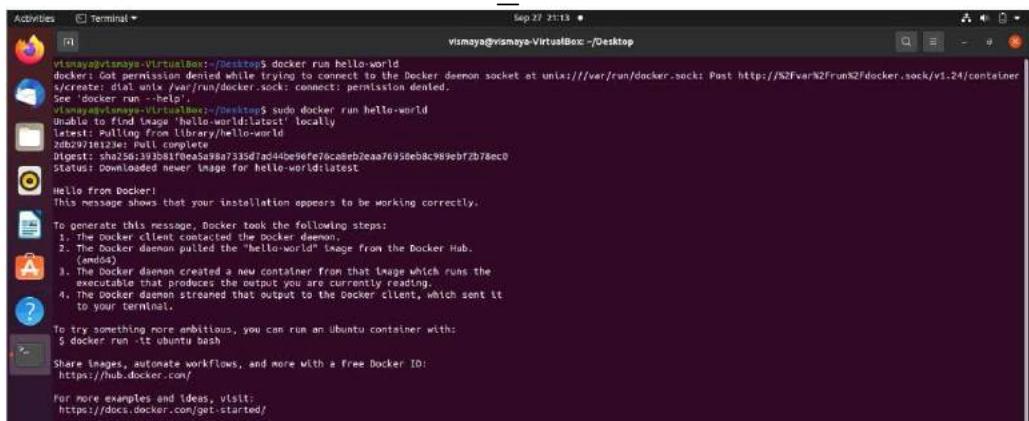


- Then check the docker is running state or not (enable /disable) if it is disable use this command to enable the docker

Sudo systemctl enable --now docker

- Now it is running state then we can type the message

Sudo docker run Hello_world



EXPERIMENT NO : 11

AIM : Automation using Ansible: Spin up a new Linux VM using Ansible playbook

PROCEDURE :

Ansible

Ansible is an open source IT automation engine that automates provisioning, configuration management, application deployment, orchestration, and many other IT processes. Use Ansible automation to install software, automate daily tasks, provision infrastructure, improve security and compliance, patch systems, and share automation across your organization.

How does Ansible work?

Ansible works by connecting to your nodes and pushing out small programs, called modules to them. Modules are used to accomplish automation tasks in Ansible.

These programs are written to be resource models of the desired state of the system. Ansible then executes these modules and removes them when finished.

Without modules, you'd have to rely on ad-hoc commands and scripting to accomplish tasks.

Ansible is agentless, which means the nodes it manages do not require any software to be installed on them.

Ansible reads information about which machines you want to manage from your inventory. Ansible has a default

inventory file, but you can create your own and define which servers you want Ansible to manage.

Ansible uses SSH protocol to connect to servers and run tasks. By default, Ansible uses SSH keys with ssh-agent and connects to remote machines using your current user name. Root logins are not required. You can log in as any user, and then su or sudo to any user.

how SSH establishes secure communication

Once it has connected, Ansible transfers the modules required by your command or playbook to the remote machine(s) for execution.

Ansible uses human-readable YAML templates so users can program repetitive tasks to happen automatically without having to learn an advanced programming language.

Ansible contains built-in modules that you can use to automate tasks, or you can write your own. Ansible modules can be written in any language that can return JSON, such as Ruby, Python, or bash. Windows automation modules are even written in Powershell.

Ansible Playbooks

Ansible Playbooks are used to orchestrate IT processes. A playbook is a YAML file containing 1 or more plays, and is used to define the desired state of a system. This differs from an Ansible module, which is a standalone script that can be used inside an Ansible Playbook.

Understanding YAML for Ansible

Plays consist of an ordered set of tasks to execute

against host selections from your Ansible inventory file. Tasks are the pieces that make up a play, and call Ansible modules. In a play, tasks are executed in the order in which they are written.

When Ansible runs, it is able to keep track of the state of the system. If Ansible scans a system and finds the playbook description of a system and the actual system state don't agree, then Ansible will make whatever changes are necessary for the system to match the playbook.

Ansible includes a "check mode" which allows you to validate playbooks and ad-hoc commands before making any state changes on a system. This shows you what Ansible would do, without actually making any changes.

Handlers in Ansible are used to run a specific task only after a change has been made to the system. They are triggered by tasks and run once, at the end of all of the other plays in the playbook.

Variables are a concept in Ansible that enable you to alter how playbooks run. Variables are used to account for differences between systems, such as package versions or file paths. With Ansible, you can execute playbooks across different systems.

Ansible variables should be defined in relation to what your playbook is actually doing.

Variables follow variable precedence, which defines the order in which variables will override each other. It's important to understand this when including variables in your playbook.

When working with Ansible you will also need to understand collections. Collections are a distribution format for Ansible content that can include playbooks, roles, modules, and plugins.

Ansible roles are a special kind of playbook that is fully self-

contained and portable with the tasks, variables, configuration templates, and other supporting files that are needed to complete a complex orchestration.

Multiple roles can exist inside a collection allowing easy sharing of content via Automation Hub and Ansible Galaxy.

Configuration management with Ansible

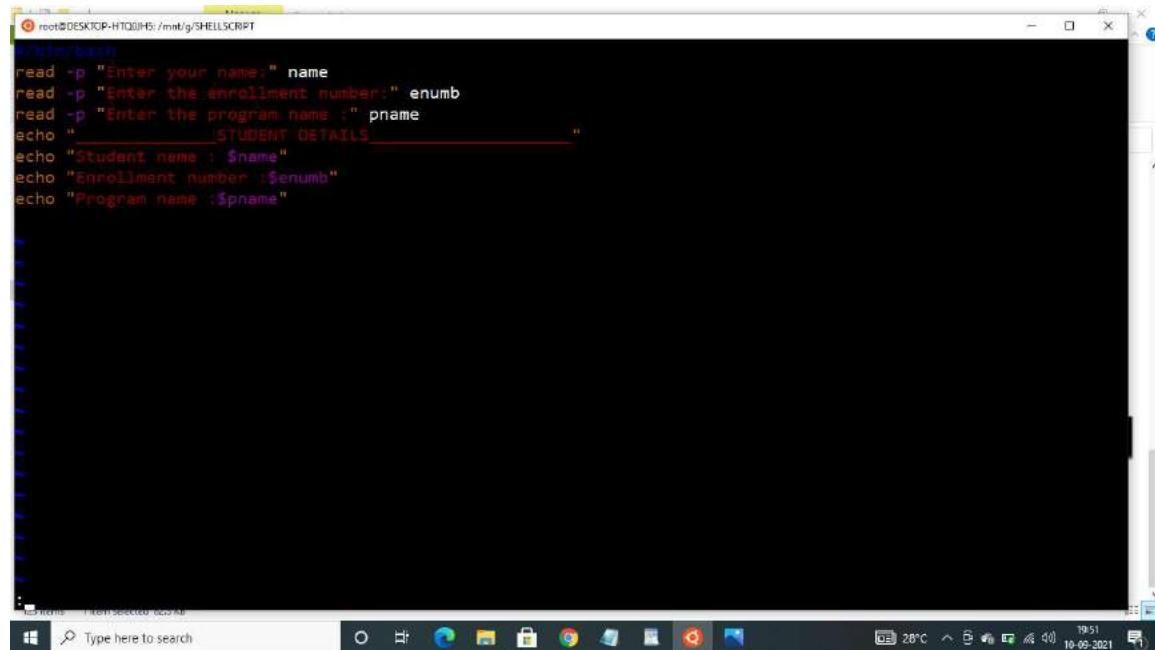
Configuration management is a process for maintaining computer systems, servers, and software in a desired, consistent state. It's a way to make sure that a system performs as it's expected to as changes are made over time.

EXPERIMENT NO :

AIM :: PROGRAMS ON SHELL SCRIPT

- Write a shell script to ask your name,program name and enrollment number and print it on the screen.

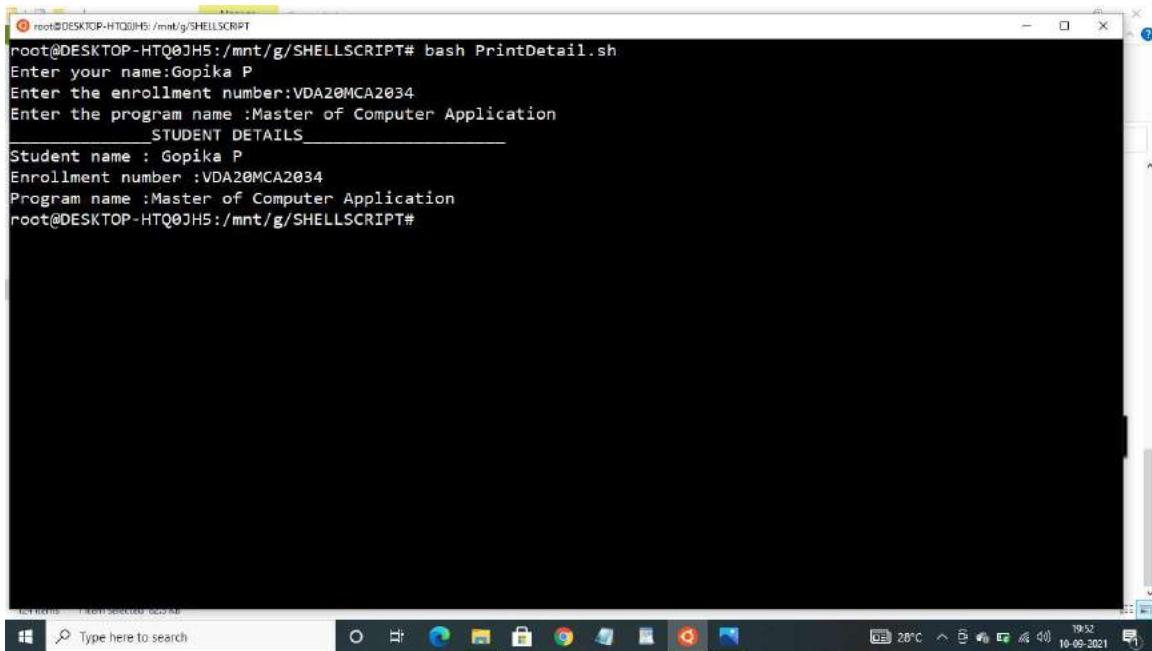
Source Code & OP:



The screenshot shows a terminal window titled "root@DESKTOP-H1Q0JH1:~/min/g/SHELLSCRIPT". The window contains the following shell script code:

```
root@DESKTOP-H1Q0JH1:~/min/g/SHELLSCRIPT
#!/bin/bash
read -p "Enter your name:" name
read -p "Enter the enrollment number:" enumb
read -p "Enter the program name :" pname
echo " _____ STUDENT DETAILS _____"
echo "Student name : $name"
echo "Enrollment number :$enumb"
echo "Program name :$pname"
```

The terminal window is running on a Windows operating system, as indicated by the taskbar at the bottom which includes icons for File Explorer, Edge, File Explorer, Task View, Start, Taskbar settings, and a system clock showing 10:51 AM on 10-09-2021.



```
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT# bash PrintDetail.sh
Enter your name:Gopika P
Enter the enrollment number:VDA20MCA2034
Enter the program name :Master of Computer Application
____STUDENT DETAILS_____
Student name : Gopika P
Enrollment number :VDA20MCA2034
Program name :Master of Computer Application
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT#
```

- Write a shell script to find the sum ,average and product of four numbers entered.

Source Code & OP:

```
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT#
#!/bin/bash
read -p "Enter the first number:" a
read -p "Enter the second number :" b
read -p "Enter the third number :" c
read -p "Enter the fourth number :" d
sum=$[ $a+$b+$c+$d ]
avg=$[ $sum/4]
prod=$[ $a*$b*$c*$d ]
echo "sum of $a,$b,$c and $d is $sum"
echo "Average of $a,$b,$c and $d is $avg"
echo "Product is $a,$b ,$c and $d $prod"
#
```

-- INSERT --

11,34 All


```
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT# bash DisplayArith.sh
Enter the first number:23
Enter the second number :12
Enter the third number :4
Enter the fourth number :5
sum of 23,12,4 and 5 is 44
Average of 23,12,4 and 5 is 11
Product is 23,12 ,4 and 5 5520
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT#
```

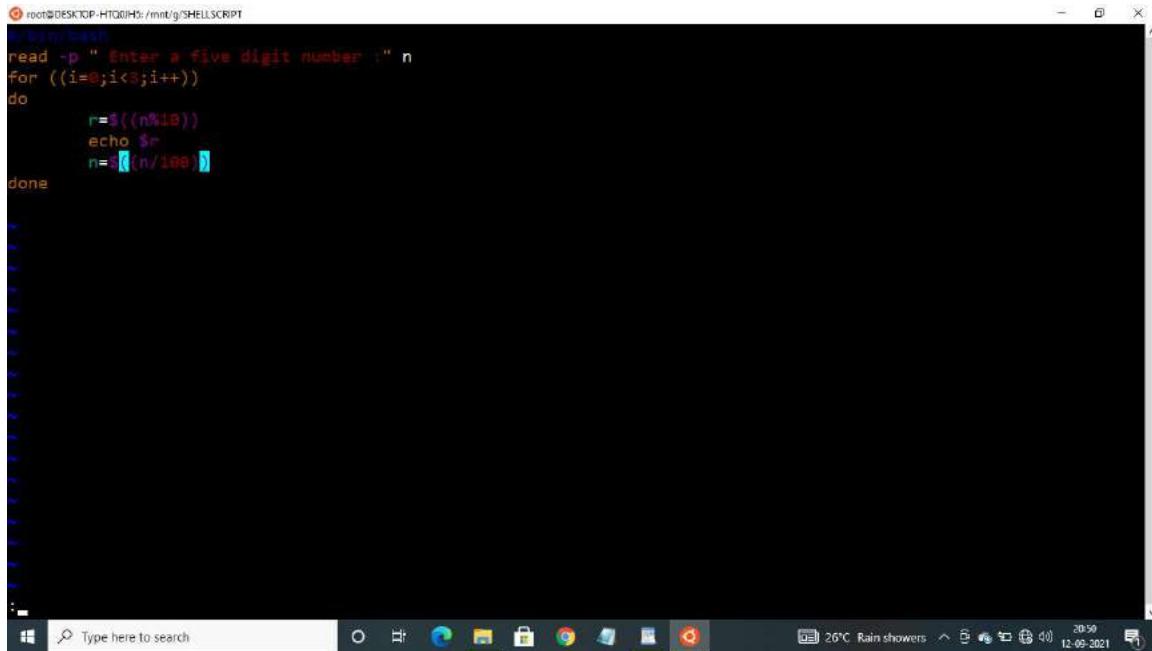

- Write a shell script to exchange the values of two variables.

Source Code & OP:

```
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT# cat swap.sh
read -p "Enter the first value : " num1
read -p "Enter the second value :" num2
echo "The value of num1 before swapping is $num1"
echo "The value of num2 before swapping is $num2"
temp=$num1
num1=$num2
num2=$temp
echo "The value of num1 after swapping is $num1"
echo "The value of num2 after swapping is $temp"
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT# vim swap.sh
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT# bash swap.sh
1,1          All
Type here to search  0  26°C Rain showers  2015  12-09-2021
root@DESKTOP-HTQ0JH5:~# cd /mnt/g
root@DESKTOP-HTQ0JH5:/mnt/g# cd SHELLSCRIPT
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT# vim swap.sh
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT# bash swap.sh
Enter the first value : 23
Enter the second value :12
The value of num1 before swapping is 23
The value of num2 before swapping is 12
The value of num1 after swapping is 12
The value of num2 after swapping is 23
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT#
```

- Write a shell script to display the digits which are in the odd position in a given 5 digit number.

Source Code & OP:



```
root@DESKTOP-HTQ0IH3:/mnt/g/SHELLSCRIPT
#!/bin/bash
read -p " Enter a five digit number :" n
for ((i=0;i<5;i++))
do
    r=$((n%10))
    echo $r
    n=$((n/100))
done
```

The screenshot shows a terminal window on a Windows desktop. The title bar of the terminal window says "root@DESKTOP-HTQ0IH3:/mnt/g/SHELLSCRIPT". The terminal itself contains a shell script. The script reads a five-digit number from the user, then loops through it five times. In each iteration, it calculates the remainder of the number divided by 10 (which gives the current digit), prints that digit, and then updates the number to be divided by 100 (removing the last digit). The terminal window has a dark background and light-colored text. Below the terminal, the Windows taskbar is visible, showing the Start button, a search bar with "Type here to search", and several pinned icons. The system tray shows the date (12-09-2021), time (20:59), battery level (20%), and weather information (20°C, Rain showers).

```
③ Select root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT# vim oddpos.sh
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT# bash oddpos.sh
Enter a five digit number :
23456
6
4
2
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT#
```

- Write a shell script to find the largest among the 3 given numbers.

Source Code & OP:

```
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT#
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT# bash larg.sh
Enter first integer: 23
Enter the second integer : 0
Enter the third integer :18
23 is largest
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT#
```



- Write a shell program to concatenate two strings and find the length of the resultant string.

Source Code & OP:

```
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT#
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT# cat concat.sh
#!/bin/bash
read -p "Enter the first string: " s1
read -p "Enter the second string : " s2
s1+=${s2}
echo "The resultant string is $s1"
len=`expr "$s1" : '\.'`
echo "length of string $len"
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT# ./concat.sh
Enter the first string: abc
Enter the second string : def
The resultant string is abcdef
length of string 6
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT#
```

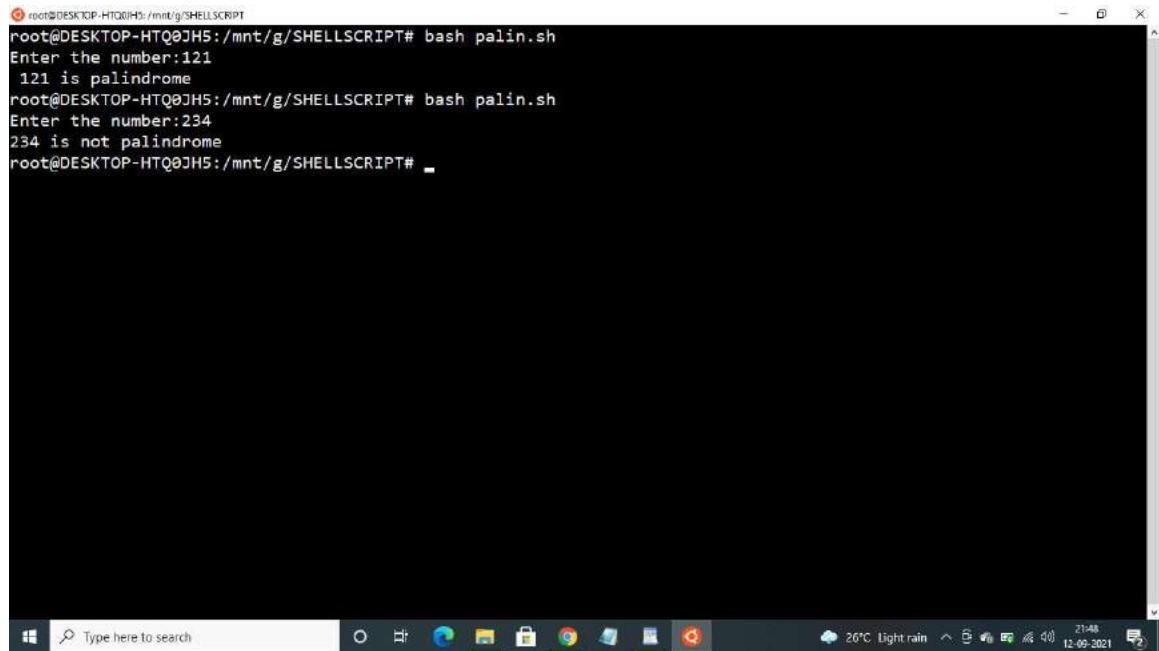


```
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT#
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT# vim concat.sh
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT# bash concat.sh
Enter the first string: computer
Enter the second string : science
The resultant string is computerscience
length of string 15
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT#
```

- Write a shell program to check whether a given string is palindrome or not.

Source Code & OP:

```
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT#
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT# cat palindrome.sh
#!/bin/bash
read -p "Enter the number:" num
s=0
r=0
n=$num
while [ $num -gt 0 ]
do
    r=$((num%10))
    s=$((($s*10)+r))
    num=$((num/10))
done
if [ $s -eq $n ]
then
    echo "$n is palindrome"
else
    echo "$n is not palindrome"
fi
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT# ./palindrome.sh
Enter the number: 12321
12321 is palindrome
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT#
```



The screenshot shows a terminal window on a Windows desktop. The command `bash palin.sh` is run twice, each time prompting for a number and outputting whether it is a palindrome or not. The terminal window has a black background with white text. The desktop taskbar at the bottom shows various icons and the system tray indicates a light rain weather forecast at 26°C.

```
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT# bash palin.sh
Enter the number:121
121 is palindrome
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT# bash palin.sh
Enter the number:234
234 is not palindrome
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT#
```

- Write a shell script to find the smallest of three numbers.

Source Code & OP:

```
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT
cat smallest3.sh
read -p "Enter the three numbers :" a b c
if [ $a -lt $b -a $a -lt $c ]
then
    echo "smallest is $a"
elif [ $b -lt $a -a $b -lt $c ]
then
    echo "smallest is $b"
else
    echo "smallest is $c"
fi

root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT#
"smallest3.sh" 12L, 206C
10,22-29      All
Windows Taskbar
```

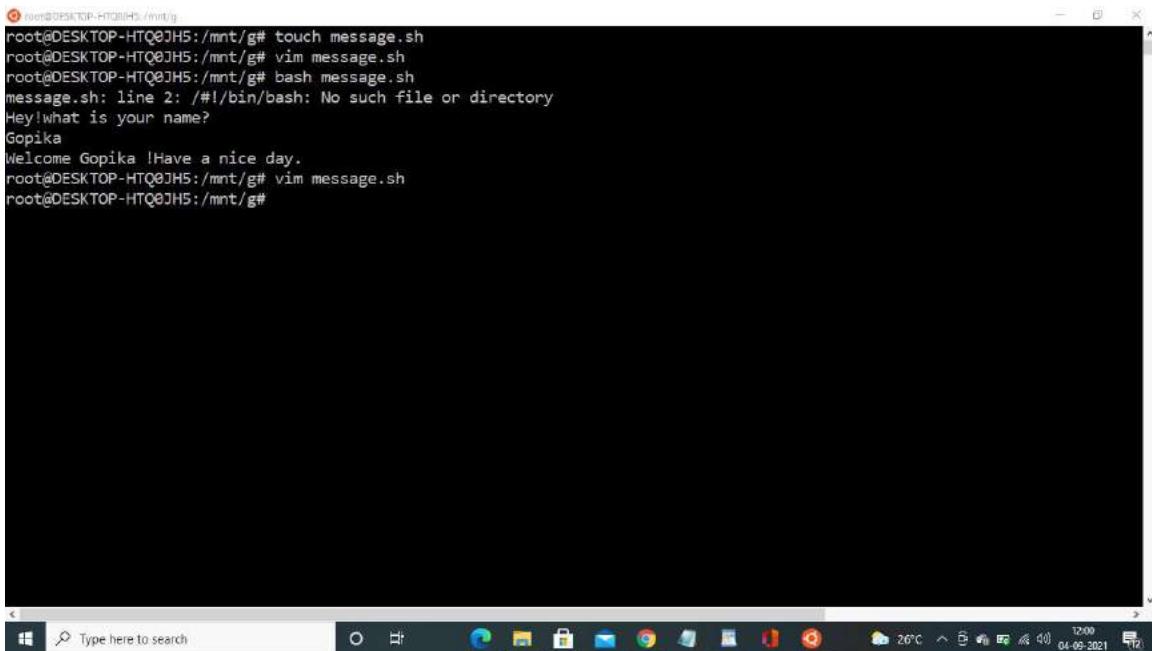
```
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT# bash smallest3.sh
Enter the three numbers :12 3 0
smallest is 0
root@DESKTOP-HTQ0JH5:/mnt/g/SHELLSCRIPT#
Windows Taskbar
```

EXPERIMENT NO:

AIM: SHELL SCRIPTING PROGRAMS

?1. Write a shell script program to display a given message.

SOURCE CODE &OUTPUT:



```
root@DESKTOP-HTQ0JH5:/mnt/g# touch message.sh
root@DESKTOP-HTQ0JH5:/mnt/g# vim message.sh
root@DESKTOP-HTQ0JH5:/mnt/g# bash message.sh
message.sh: line 2: /#!/bin/bash: No such file or directory
Hey! what is your name?
Gopika
Welcome Gopika !Have a nice day.
root@DESKTOP-HTQ0JH5:/mnt/g# vim message.sh
root@DESKTOP-HTQ0JH5:/mnt/g#
```

A screenshot of a Windows desktop environment. At the top, there's a taskbar with various icons. Below it is a terminal window titled 'mint/g'. The terminal contains the following bash script:

```
#!/bin/bash
echo "Hey! What is your name?"
read name
echo "Welcome $name ! Have a nice day."
```

The terminal window shows the script being run, with the prompt 'What is your name?' and the response 'Hello' followed by the output 'Welcome Hello ! Have a nice day.'. The bottom right corner of the screen displays system information: 26°C, 11:59, 04-09-2021, and battery status.

?2. Write a shell script to print whether two numbers are equal or not.

SOURCE CODE & OUTPUT:

```
root@DESKTOP-HTQ0JH5:/mnt/g#
root@DESKTOP-HTQ0JH5:/mnt/g# vim equaln.sh
root@DESKTOP-HTQ0JH5:/mnt/g# bash equaln.sh
Enter the first number :23
Enter the second number :56
numbers are not equal
root@DESKTOP-HTQ0JH5:/mnt/g# bash equaln.sh
Enter the first number :34
Enter the second number :34
numbers are equal
root@DESKTOP-HTQ0JH5:/mnt/g#
```

```
root@DESKTOP-HTQ0JH5:/mnt/g#
root@DESKTOP-HTQ0JH5:/mnt/g# cat equaln.sh
read -p "Enter the first number :" num1
read -p "Enter the second number :" num2
if [ $num1 -eq $num2 ]; then
    echo "numbers are equal"
else
    echo "numbers are not equal"
fi
root@DESKTOP-HTQ0JH5:/mnt/g# ./equaln.sh
Enter the first number :12
Enter the second number :12
numbers are equal
root@DESKTOP-HTQ0JH5:/mnt/g# ./equaln.sh
Enter the first number :12
Enter the second number :13
numbers are not equal
root@DESKTOP-HTQ0JH5:/mnt/g#
```

?3. Write a Shell Program to find the roots of the quadratic

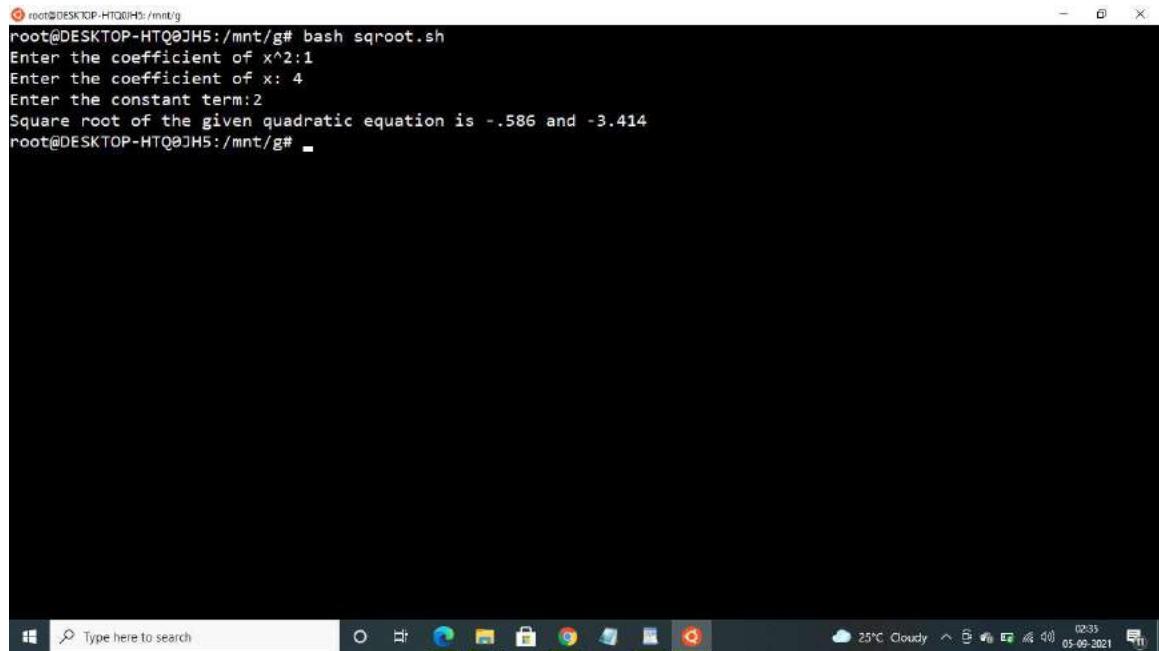
equation

SOURCE CODE & OUTPUT:

The screenshot shows a terminal window titled "root@DESKTOP-HIQ8H4: /mnt/g". The window contains the following code:

```
read -p "Enter the coefficient of x^2: " a
read -p "Enter the coefficient of x: " b
read -p "Enter the constant term: " c
D=$((($b)*($b)-4*($a)*($c)))
x1=$(echo "scale=3;(-$b+sqrt($D))/(2*$a)" | bc)
x2=$(echo "scale=3;(-$b-sqrt($D))/(2*$a)" | bc)
echo "Square root of the given quadratic equation is $x1 and $x2"
```

The terminal window also displays the file name "sqroot.sh" and its size "9L, 324C". The status bar at the bottom right shows "5,28" and "All".



A screenshot of a Windows desktop environment. In the center is a terminal window with a black background and white text. The text shows the execution of a shell script named 'sqroot.sh'. The user enters coefficients for a quadratic equation, and the script outputs the square roots. The desktop taskbar at the bottom shows various icons, and the system tray indicates it's 25°C, cloudy, and the date is 05-09-2021.

```
root@DESKTOP-HTQ0JH5:/mnt/g
root@DESKTOP-HTQ0JH5:/mnt/g# bash sqroot.sh
Enter the coefficient of x^2:1
Enter the coefficient of x: 4
Enter the constant term:2
Square root of the given quadratic equation is -.586 and -3.414
root@DESKTOP-HTQ0JH5:/mnt/g#
```

?4. Write a shell script to perform integer arithmetic operations.

SOURCE CODE & OUTPUT:

```
root@DESKTOP-HTQ0JH5:/mnt/g# vim arithmetic.sh
root@DESKTOP-HTQ0JH5:/mnt/g# bash arithmetic.sh
Enter the num1:23
Enter the num2:5
Sum of 23 and 5 is 28
Difference of 23 and 5 is 18
Product of 23 and 5 is 115
23 divided by 5 is 4
root@DESKTOP-HTQ0JH5:/mnt/g#
```

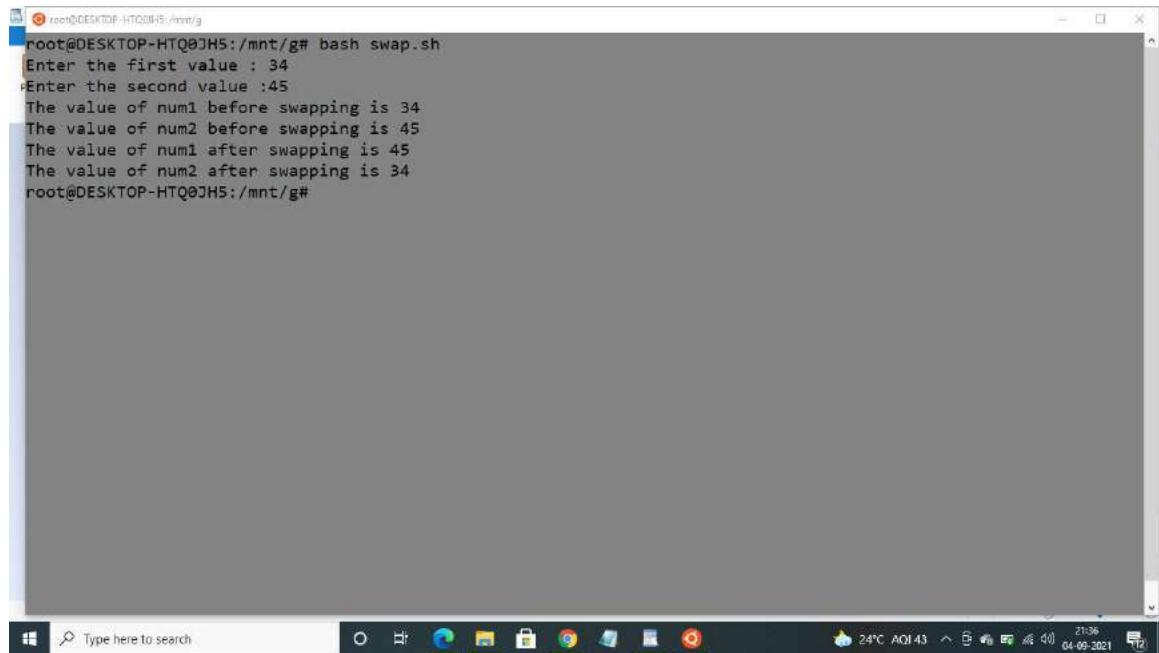
?5. Write a shell script to get input details like name, roll number and marks and print them.

source code & output:

```
root@DESKTOP-HTQ0JH5:/mnt/g# vim display.sh
root@DESKTOP-HTQ0JH5:/mnt/g# bash display.sh
Enter the name:GOPIKA P
Enter the roll no : 37
Enter the mark scored in malayalam :99
Enter the mark scored in hindi :100
Enter the mark scored in maths : 100
Enter the mark scored in IT : 100
Student name is :GOPIKA P
Roll no: 37
____Marks Scored_____
Malayalam 99
Hindi 100
Maths 100
IT 100
root@DESKTOP-HTQ0JH5:/mnt/g#
```

?6. Write a Shell program to swap two values.

SOURCE CODE & OUTPUT:



```
root@DESKTOP-HTQ0JH5:/mnt/g# bash swap.sh
Enter the first value : 34
Enter the second value :45
The value of num1 before swapping is 34
The value of num2 before swapping is 45
The value of num1 after swapping is 45
The value of num2 after swapping is 34
root@DESKTOP-HTQ0JH5:/mnt/g#
```

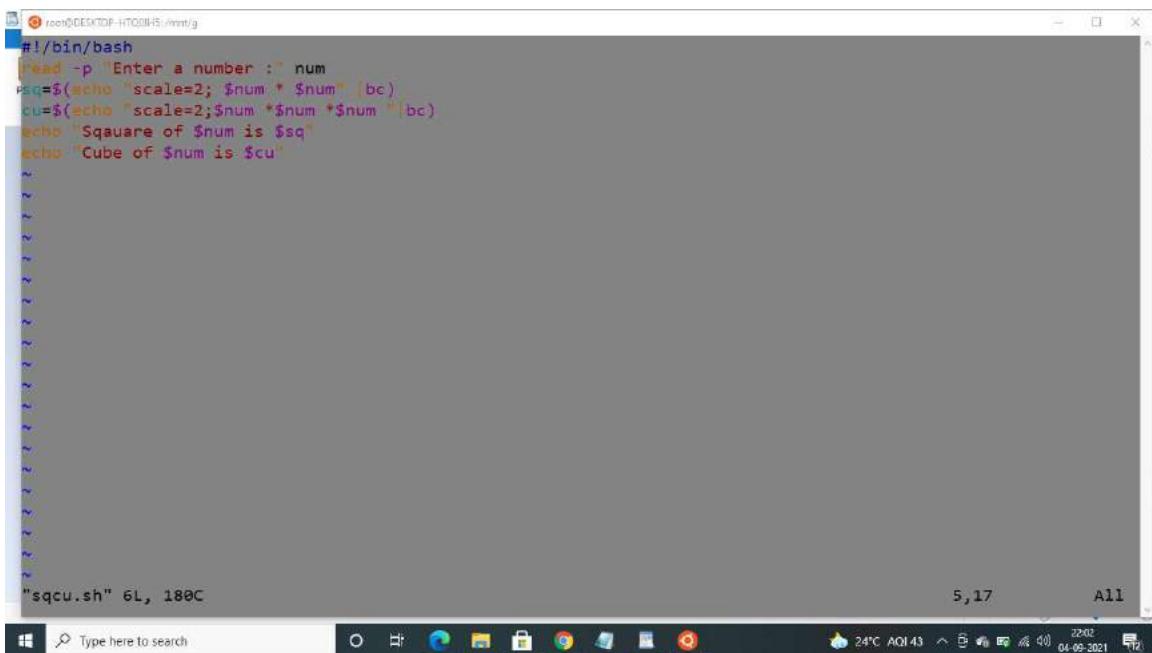
?7. Write a shell program to find the area of a triangle.

SOURCE CODE & OUTPUT:

```
root@DESKTOP-HTQ0JH5:/mnt/g# vim area.sh
root@DESKTOP-HTQ0JH5:/mnt/g# bash area.sh
Enter the base: 7
Enter the height:12
Area of triangle is : 42.00
root@DESKTOP-HTQ0JH5:/mnt/g#
```

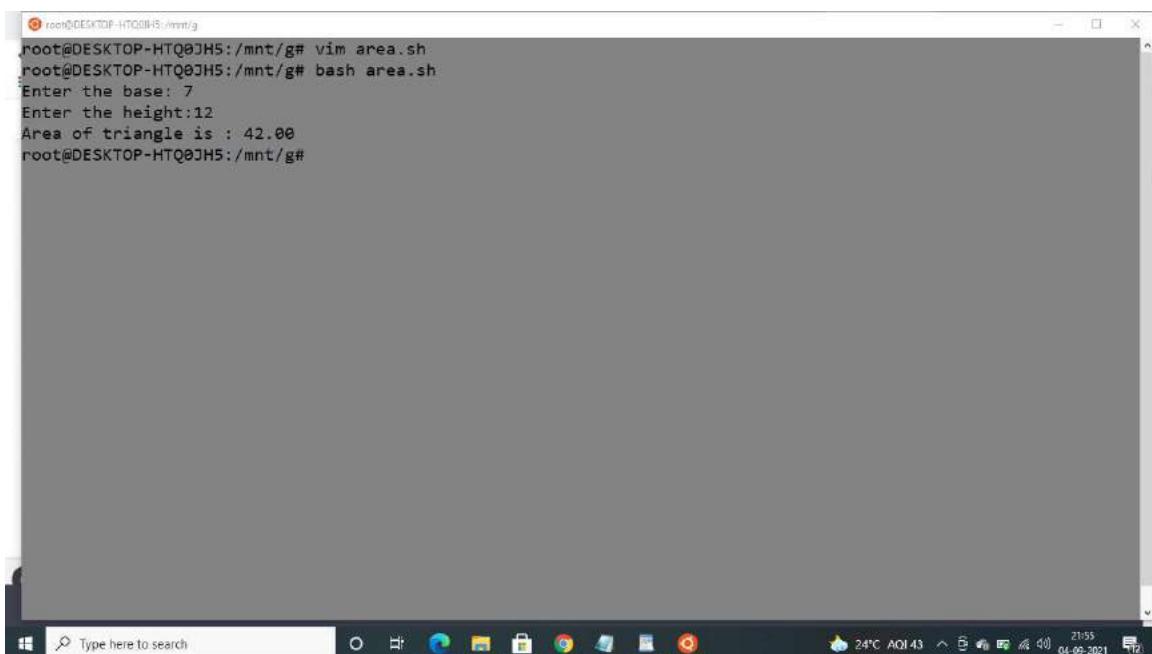
?8. Write a shell program to find the square and cube of a number.

SOURCE CODE & OUTPUT:



```
#!/bin/bash
read -p "Enter a number :" num
sq=$(echo "scale=2; $num * $num" | bc)
cu=$(echo "scale=2;$num *$num *$num" | bc)
echo "Square of $num is $sq"
echo "Cube of $num is $cu"
```

"sqcu.sh" 6L, 180C

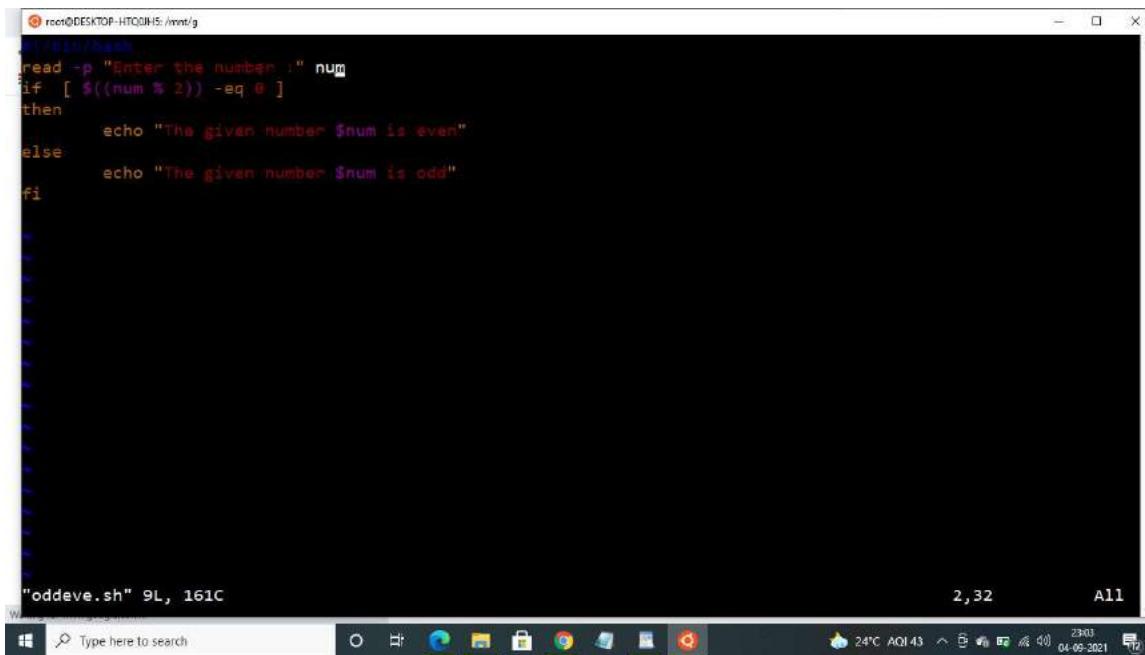


```
root@DESKTOP-HTQ0JH5:/mnt/g# vim area.sh
root@DESKTOP-HTQ0JH5:/mnt/g# bash area.sh
Enter the base: 7
Enter the height:12
Area of triangle is : 42.00
root@DESKTOP-HTQ0JH5:/mnt/g#
```

?9. Write a shell program to check whether the given

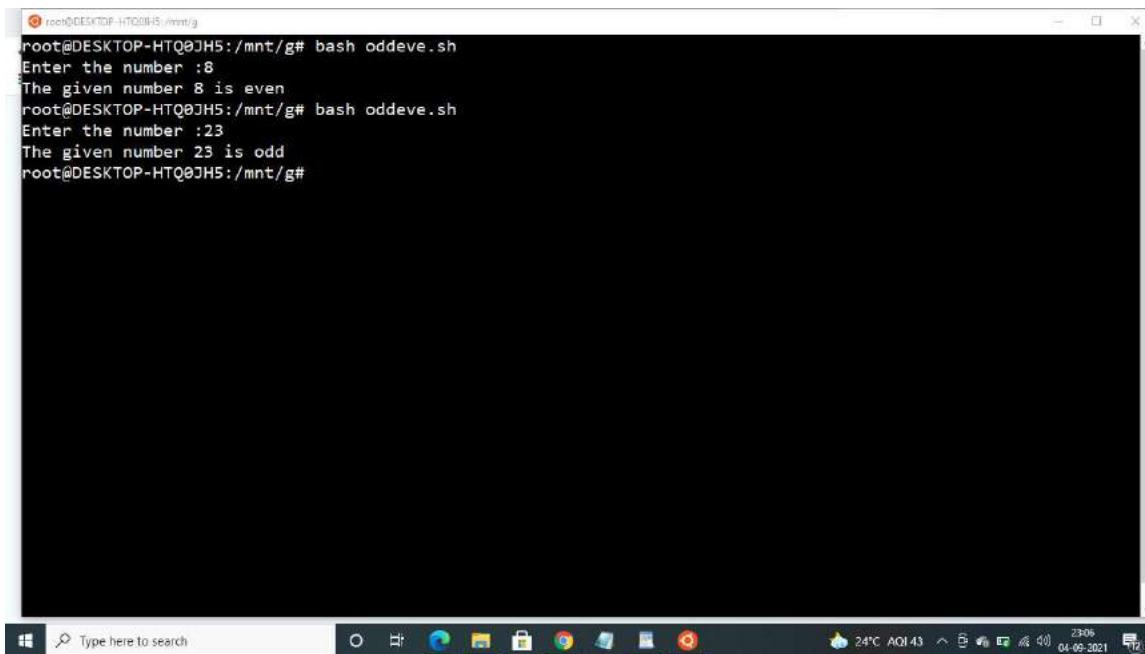
number is odd or even.

SOURCE CODE & OUTPUT:



```
root@DESKTOP-HTQ0JH5:/mnt/g
#!/bin/bash
read -p "Enter the number :" num
if [ $(($num % 2)) -eq 0 ]
then
    echo "The given number $num is even"
else
    echo "The given number $num is odd"
fi

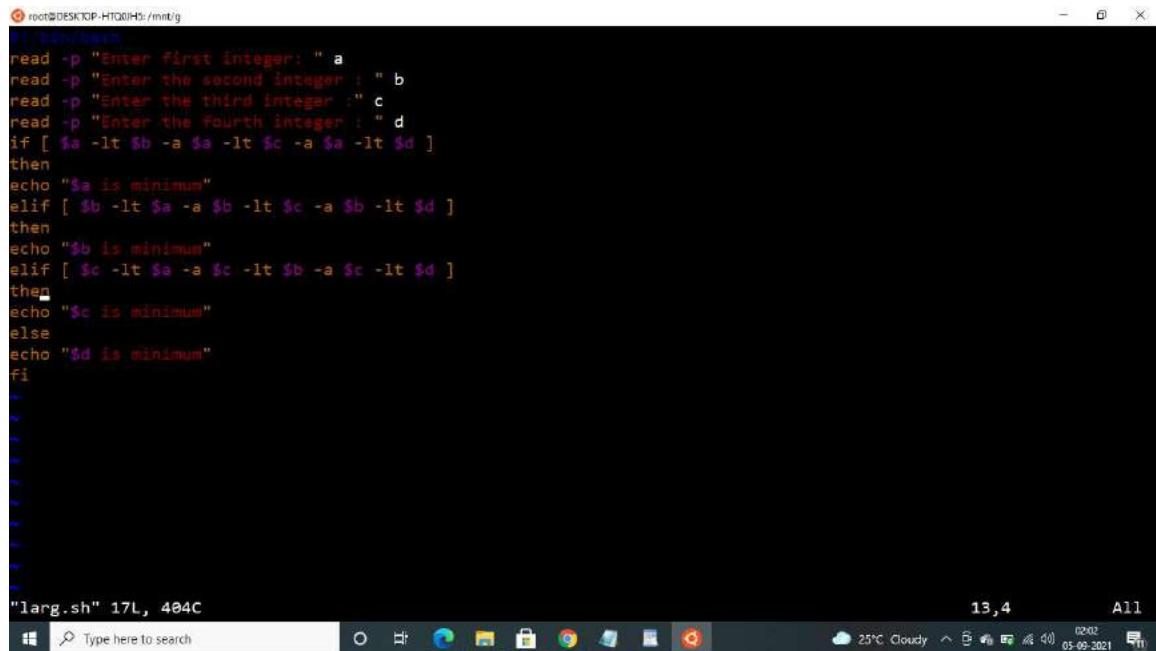
"oddeve.sh" 9L, 161C
2,32          All
```



```
root@DESKTOP-HTQ0JH5:/mnt/g# bash oddeve.sh
Enter the number :8
The given number 8 is even
root@DESKTOP-HTQ0JH5:/mnt/g# bash oddeve.sh
Enter the number :23
The given number 23 is odd
root@DESKTOP-HTQ0JH5:/mnt/g#
```

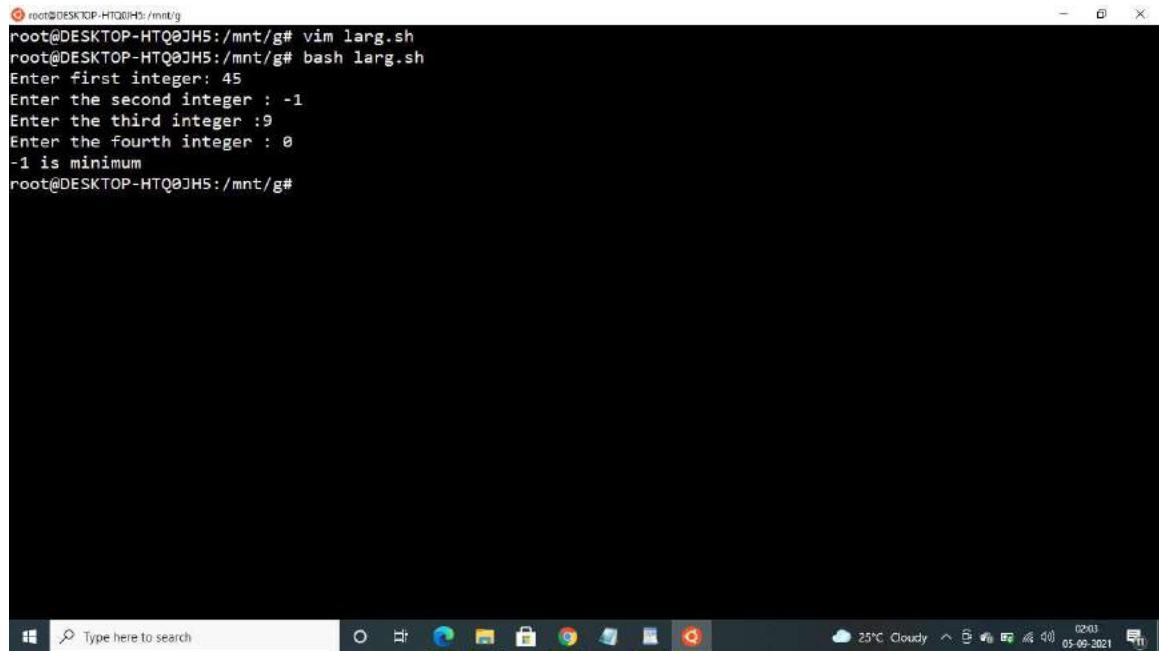
? .10 Write a shell program to find the minimum among four values.

SOURCE CODE & OUTPUT:



```
root@DESKTOP-HTQ0IH5:~# ./larg.sh
Enter first integer: 10
Enter the second integer : 20
Enter the third integer : 30
Enter the fourth integer : 40
$ a is minimum
root@DESKTOP-HTQ0IH5:~#
```

"larg.sh" 17L, 404C 13,4 All



A screenshot of a Windows desktop environment. In the center is a terminal window with a black background and white text. The terminal shows the following session:

```
root@DESKTOP-HTQ0JH5:/mnt/g#
root@DESKTOP-HTQ0JH5:/mnt/g# vim larg.sh
root@DESKTOP-HTQ0JH5:/mnt/g# bash larg.sh
Enter first integer: 45
Enter the second integer : -1
Enter the third integer :9
Enter the fourth integer : 0
-1 is minimum
root@DESKTOP-HTQ0JH5:/mnt/g#
```

The desktop taskbar at the bottom has several icons, including File Explorer, Task View, Start, Taskbar settings, and a search bar. The system tray shows the date (05-09-2021), time (02:03), battery level, signal strength, and temperature (25°C). The weather icon indicates Cloudy.

?11. Write a shell program to check whether the input number is prime or not.

SOURCE CODE & OUTPUT:

```
root@DESKTOP-HTQ0JH5:/mnt/g# bash primen.sh
Enter the number: 6
Given number 6 is not prime
root@DESKTOP-HTQ0JH5:/mnt/g# bash primen.sh
Enter the number: 7
Number is prime
root@DESKTOP-HTQ0JH5:/mnt/g#
```

? .12. Write a shell program to find the area of circle, square, rectangle and triangle using case statements.

SOURCE CODE & OUTPUT:

```
root@DESKTOP-HTQ0H4B: /mnt/g
#!/bin/bash
echo "Enter an input:"
read n
case $n in
    1)read -p "Enter the radius :" r
        area= $(echo "scale=2;($.14) * $r * $r" |bc)
        echo "Area of circle is $area";;
    2) read -p "Enter the value of one side :" a
        area1=$(echo "scale=2; $a * $a" |bc)
        echo "Area of square is $area1";;
    3)read -p "Enter the breadth :" b
        read -p "Enter the height :" h
        area2=$(echo "scale=2; $b * $h" |bc)
        echo "Area of rectangle is $area2";;
    4)read -p "Enter the base :" b
        read -p "Enter the height :" h
        area3=$(echo "scale=2; (1/2) * $b * $h" |bc)
        echo "Area of triangle is $area3";;
    *)echo "Please enter a valid input!!"
esac
```

```
root@DESKTOP-HTQ0JH5:/mnt/g#
root@DESKTOP-HTQ0JH5:/mnt/g# bash areaswitch.sh
Enter an input:
1
Enter the radius:2
Area of circle is 12.56
root@DESKTOP-HTQ0JH5:/mnt/g# bash areaswitch.sh
Enter an input:
2
Enter the value of one side :2
Area of square is 4
root@DESKTOP-HTQ0JH5:/mnt/g# bash areaswitch.sh
Enter an input:
3
Enter the length :3
Enter the breadth :5
Area of rectangle is 15
root@DESKTOP-HTQ0JH5:/mnt/g# bash areaswitch.sh
Enter an input:
4
Enter the base : 3
Enter the height :6
Area of triangle is 9.00
root@DESKTOP-HTQ0JH5:/mnt/g# bash areaswitch.sh
Enter an input:
6
Please enter a valid input!!
root@DESKTOP-HTQ0JH5:/mnt/g#
```

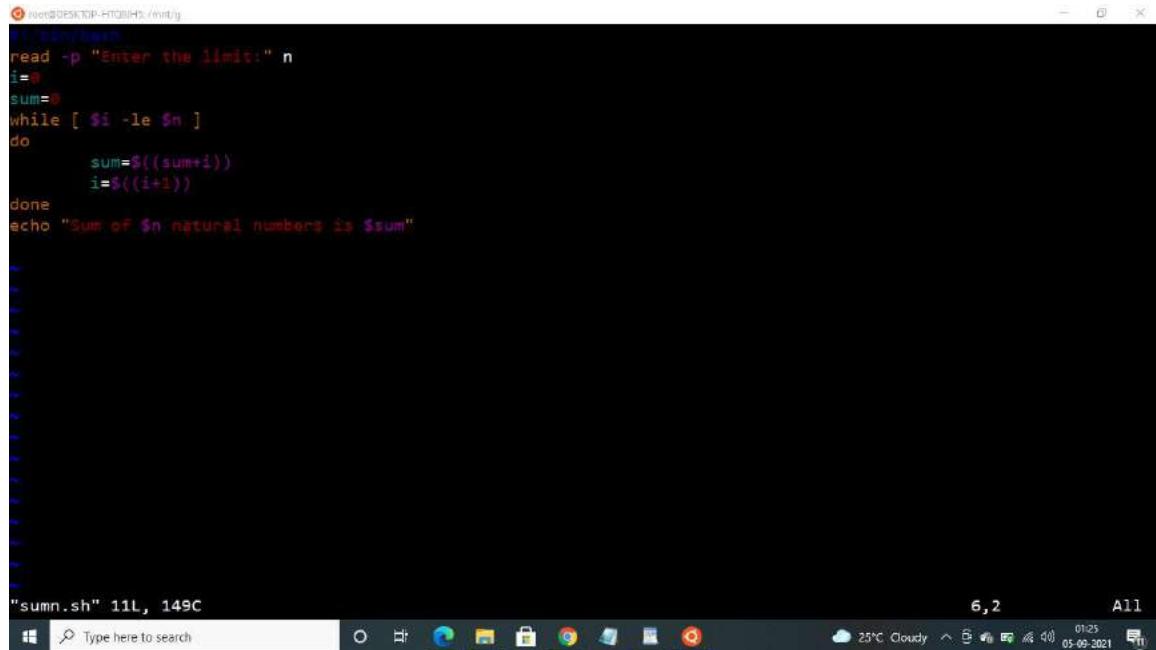
? .13. Write a shell program to find the factorial of a given number.

SOURCE CODE & OUTPUT:

```
root@DESKTOP-HTQ0JH5:/mnt/g# vi fact.sh
#!/bin/bash
read -p "enter the number: " num
i=1
fact=1
while [ $i -le $num ]
do
    fact=$((fact*i))
    i=$((i+1))
done
echo "Factorial of $num is $fact"
root@DESKTOP-HTQ0JH5:/mnt/g# bash fact.sh
enter the number: 5
Factorial of 5 is 120
root@DESKTOP-HTQ0JH5:/mnt/g#
```

?14. Write a Simple Shell script to print the sum of n natural numbers.

SOURCE CODE & OUTPUT:



```
root@DESKTOP-H7CQH4:~# vi sumn.sh
#!/bin/bash
read -p "Enter the limit:" n
i=0
sum=0
while [ $i -le $n ]
do
    sum=$((sum+i))
    i=$((i+1))
done
echo "Sum of $n natural numbers is $sum"

root@DESKTOP-H7CQH4:~# ./sumn.sh
Enter the limit: 10
Sum of 10 natural numbers is 55
root@DESKTOP-H7CQH4:~#
```

The screenshot shows a terminal window on a Windows operating system. The title bar says "root@DESKTOP-H7CQH4:~#". The command "vi sumn.sh" is run to edit a shell script named "sumn.sh". The script prompts for a limit, initializes variables, uses a while loop to calculate the sum of natural numbers, and then prints the result. The terminal shows the script's content, the command to run it, and the resulting output "Sum of 10 natural numbers is 55". The taskbar at the bottom includes icons for File Explorer, Task View, Start, Taskbar settings, and a search bar. The system tray shows the date and time as 05-09-2021 01:25.

```
root@DESKTOP-HTQ0JH5:/mnt/g#
root@DESKTOP-HTQ0JH5:/mnt/g# vim sumn.sh
root@DESKTOP-HTQ0JH5:/mnt/g# bash sumn.sh
Enter the limit:10
Sum of 10 natural numbers is 55
root@DESKTOP-HTQ0JH5:/mnt/g#
```

The screenshot shows a Windows desktop environment with a terminal window open. The terminal window has a black background and white text. It displays a shell script named 'sumn.sh' being run. The user enters '10' as the limit, and the script calculates the sum of natural numbers up to 10, which is 55. The terminal window is titled with the path 'root@DESKTOP-HTQ0JH5:/mnt/g#'. The desktop taskbar at the bottom shows various icons for applications like File Explorer, Edge, and Google Chrome.

?15. Write a shell program to reverse a number

SOURCE CODE & OUTPUT:

```
#!/bin/bash
read -p "Enter the number :" n
num=0
while [ $n -gt 0 ]
do
    rev=$((n%10))
    n=$((n/10))
    num=$(( (num*10)+rev ))
done
echo "Reverse of the given number is $num"
```

The screenshot shows a Windows desktop environment with a terminal window open. The terminal window has a black background and white text. It displays a shell script that takes a number as input and prints its reverse. The user enters '12345', and the script outputs '54321'. The terminal window is titled with the path 'root@DESKTOP-HTQ0JH5:/mnt/g#'. The desktop taskbar at the bottom shows various icons for applications like File Explorer, Edge, and Google Chrome.

```
root@DESKTOP-HTQ0JH5:/mnt/g#
root@DESKTOP-HTQ0JH5:/mnt/g# bash reverse.sh
Enter the number :234
Reverse of the given number is 432
root@DESKTOP-HTQ0JH5:/mnt/g#
```

