MASTER OF COMPUTER APPLICATIONS

PRACTICAL RECORD WORK

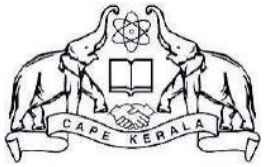
ON

20MCA136 NETWORKING & SYSTEM ADMINISTRATION LAB

Submitted

By

DILSHAD MUHAMMED K
(Reg. No. : VDA20MCA-2029)



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

OCTOBER - 2021

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

CERTIFICATE

Certified that this is a bonafide record of the practical work on the course 20MCA136 NETWORKING & SYSTEM ADMINISTRATION LAB done by Mr./Ms. DILSHAD MUHAMMED K (Reg.No.: VDA20MCA-2029) second Semester MCA
student of Department of Computer Applications at College of Engineering Vatakara in the partial fulfilment for the award of the degree of Master of Computer Applications (MCA) of APJ Abdul Kalam Technological University (KTU)

(Ms. Rajalekshmy K D)
FACULTY-IN-CHARGE

HEAD OF THE DEPARTMENT

CEV
03/10/2021

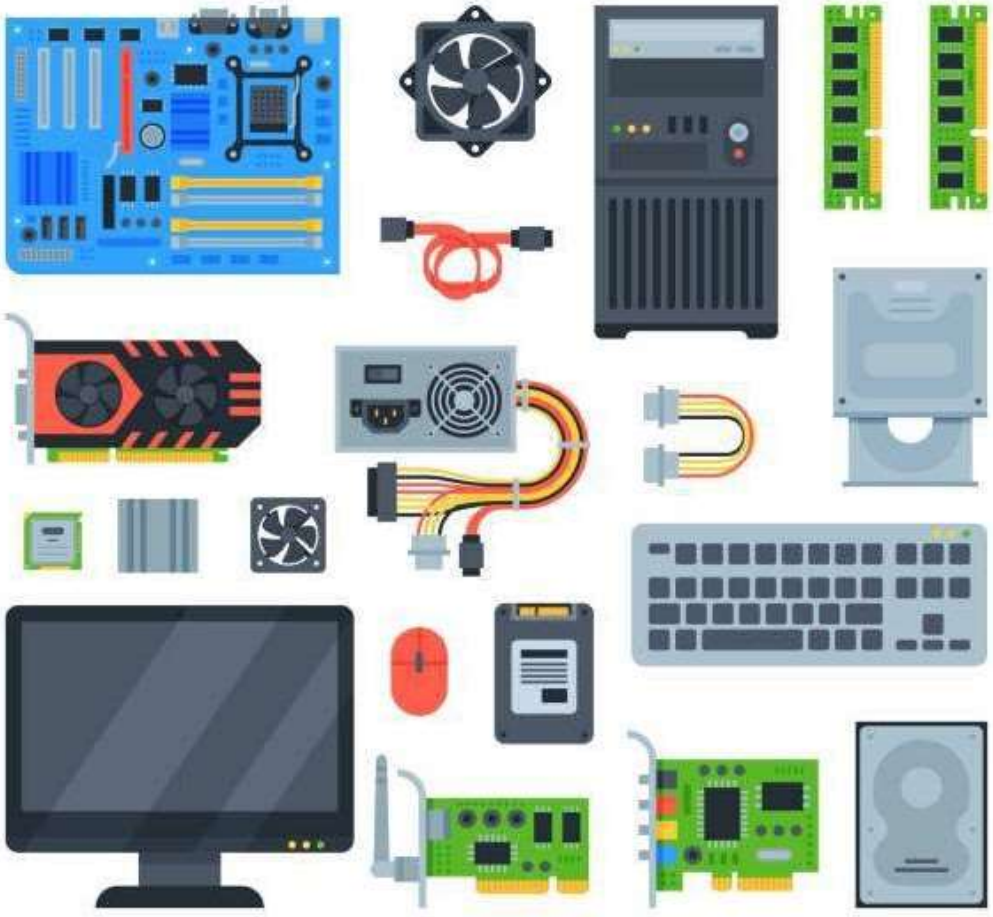
EXAMINERS:

INDEX

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.

INTRODUCTION :

Computer hardware (usually simply called hardware when a computing context is concerned) is the collection of physical elements that constitutes a computer system. **Computer hardware is the physical parts or components of a computer**, such as the monitor, mouse, keyboard,computer data storage, hard disk drive (HDD), graphic cards, sound cards, memory, motherboard, and so on, all of which are physical objects that are tangible.

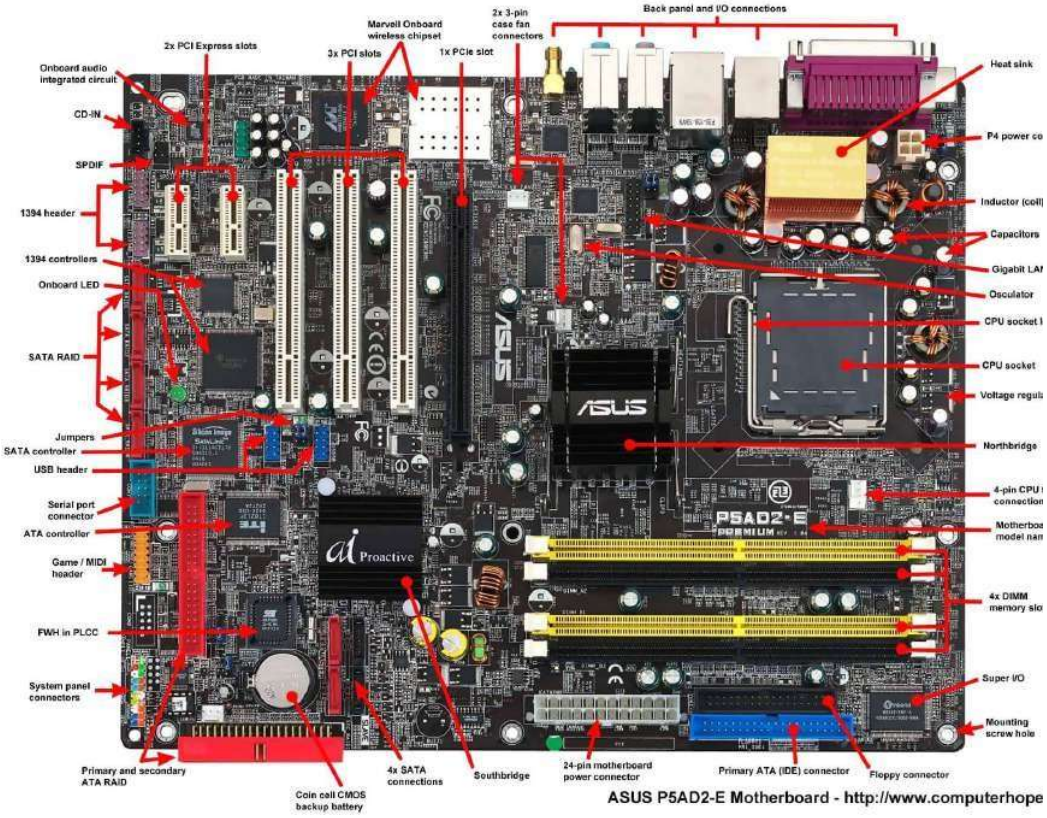


1) MOTHER BOARD :

The first motherboard is considered to be one used in the IBM Personal Computer, released in 1981. At the time, IBM called it a **"planar"** instead of a motherboard.

The motherboard is a **printed circuit board** and foundation of a computer that is the biggest board in a computer chassis. It allocates power and allows communication to and between the CPU, RAM, and all other computer hardware components.

A motherboard provides connectivity between the hardware components of a computer, like the processor (CPU), memory (RAM), hard drive, and video card.



=> NORTH BRIDGE : Alternatively referred to as the PAC (PCI/AGP Controller) and nb, the Northbridge is an integrated circuit responsible for communications

between the CPU interface, AGP, and the memory.The northbridge is usually slightly larger than the southbridge, and is positioned closer to the CPU and memory.

When the CPU needs data from RAM, a request is sent to the northbridge memory controller. After the request is received, the northbridge responds with how long the processor needs to wait to read memory over the front-side bus.

=> SOUTH BRIDGE : The southbridge is an IC on the motherboard responsible for the hard drive controller, I/O controller and integrated hardware. Integrated hardware can include the sound card and video card if on the motherboard, USB, PCI, ISA, IDE, BIOS, and Ethernet.

Although the southbridge handles most of the I/O devices, less prominent input/output devices, such as a serial port, keyboard, and non-USB mouse are handled by the **SIO (super input/output).**

=> CMOS : CMOS is short for **Complementary Metal-Oxide Semiconductor**. CMOS is an onboard, battery powered semiconductor chip inside computers that stores information. This information ranges from the system time and date to system hardware settings for your computer.

=> BIOS : Short for **Basic Input/Output System**, the BIOS is a ROM chip found on motherboards that allows you to access and set up your computer system at the most basic level.

The BIOS includes instructions on how to load basic computer hardware. It also includes a test referred to as a POST (Power-On Self-Test) that helps verify the computer meets requirements to boot up properly. If the computer does not pass the POST, you hear a combination of beeps indicating what is malfunctioning in the computer.

=> POST (Power-On Self-Test) :Test the computer hardware and make sure no errors exist before loading the operating system.

=> Bootstrap Loader : Locate the operating system. If a capable operating system is

located, the BIOS will pass control to it

2. RAM (RANDOM ACCESS MEMORY) :

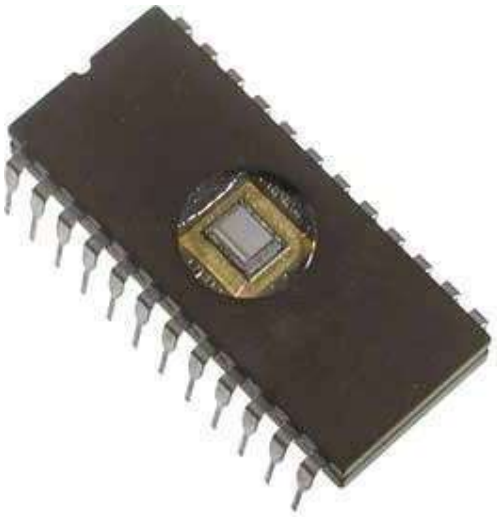
Alternatively referred to as main memory, primary memory, or system memory, RAM (random-access memory) is a hardware device that allows information to be stored and retrieved on a computer. RAM is usually associated with DRAM, which is a type of memory module. Because data is accessed randomly instead of sequentially like it is on a CD or hard drive, access times are much faster. However, unlike ROM, RAM is a **volatile** memory and requires power to keep the data accessible. If the computer is turned off, all data contained in RAM is lost.

Access time in RAM is independent of the address, that is, each storage location inside the memory is as easy to reach as other locations and takes the same amount of time. Data in the RAM can be accessed randomly but it is very expensive.

3.ROM (READ ONLY MEMORY) :

ROM stands for Read Only Memory. The memory from which we can only read but cannot write on it. This type of memory is **non-volatile**. The information is stored permanently in such memories during manufacture. A ROM stores such instructions that are required to start a computer. This operation is referred to as bootstrap. ROM chips are not only used in the computer but also in other electronic

items like washing machine and microwave oven.



4. DAUGHTER BOARD :

Alternatively referred to as a daughter card, a daughterboard is an expansion board that connects directly to the motherboard and gives added functionality (e.g., modem).

A motherboard with daughterboard

5. BUS SLOTS :

Alternatively known as a 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 you wanted to install a new video card in the computer, you'd purchase a video

expansion card and install that card into the compatible expansion slot.

An expansion slot is a socket on the motherboard that is used to insert an expansion card (or circuit board), which provides additional features to a computer such as video, sound, advanced graphics, Ethernet or memory.

The expansion card has an edge connector that fits precisely into the expansion slot as well as a row of contacts that is designed to establish an electrical connection between the motherboard and the electronics on the card, which are mostly integrated circuits. Depending on the form factor of the case and motherboard, a computer system generally can have anywhere from one to seven expansion slots. With a backplane system, up to 19 expansion cards can be installed.

6. SMPS (Switched Mode Power Supply) :

SMPS is the Switched Mode Power Supply circuit which is designed for obtaining the regulated DC output voltage from an unregulated DC or AC voltage. A SMPS is an electronic power supply that incorporates a switching regulator to convert electrical power efficiently. SMPS provide improved efficiency & space saving over traditional linear supplies, but care has to be taken to ensure noise on the output is low. Switch mode power supplies are widely used because of the advantages they offer in terms of size, weight, cost, efficiency and overall performance.

The advantages of SMPS include :

- The efficiency is as high as 80 to 90%.
- Less heat generation; less power wastage.
- Reduced harmonic feedback into the supply mains.
- The device is compact and small in size.
- The manufacturing cost is reduced.
- Provision for providing the required number of voltages.

SMPS DIAGRAM

- Interior view of a switched-mode power supply.

A -bridge rectifier

B -Input filter capacitors

C-Transformer

D -output filter

E -output filter capacitors

7. 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.**

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.

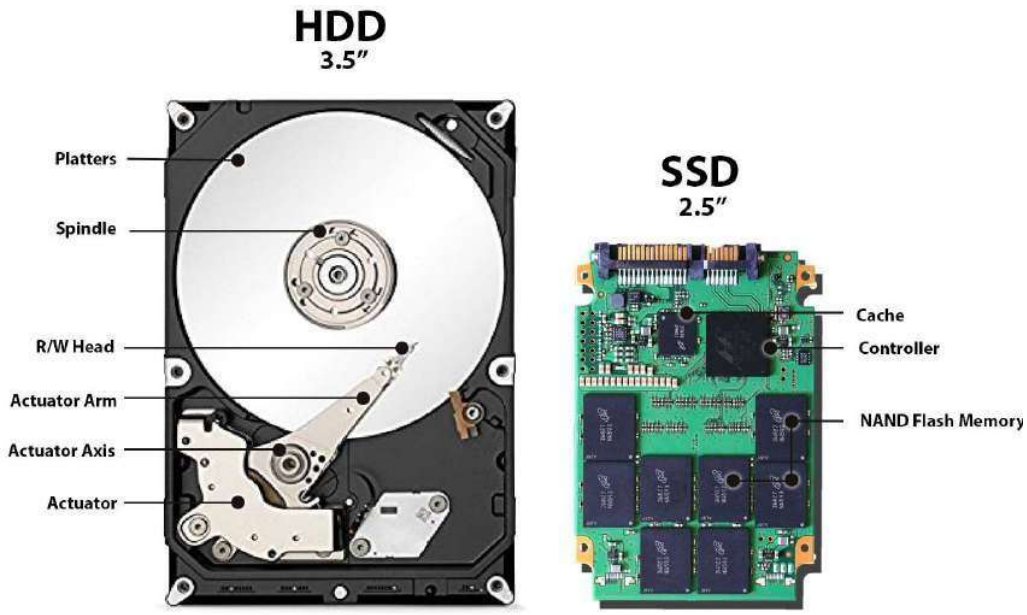
It will also be used to store the applications software that you use and more than likely, the original copies of your data files.

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.

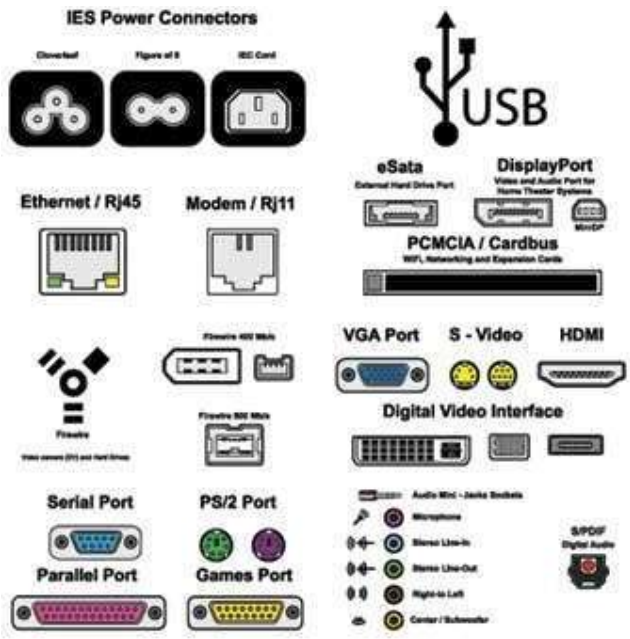


8. Interfacing ports :

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.

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



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.

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.

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.

VGA cable

VGA connector

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.

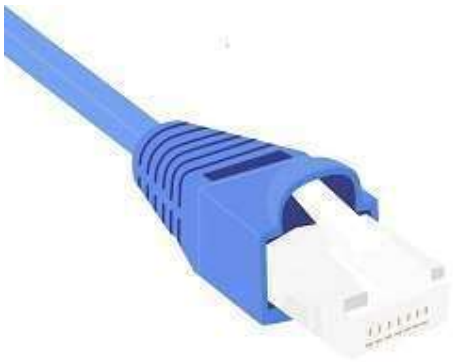
Modem Port :

- Connects a PC's modem to the telephone network.

Ethernet Port :

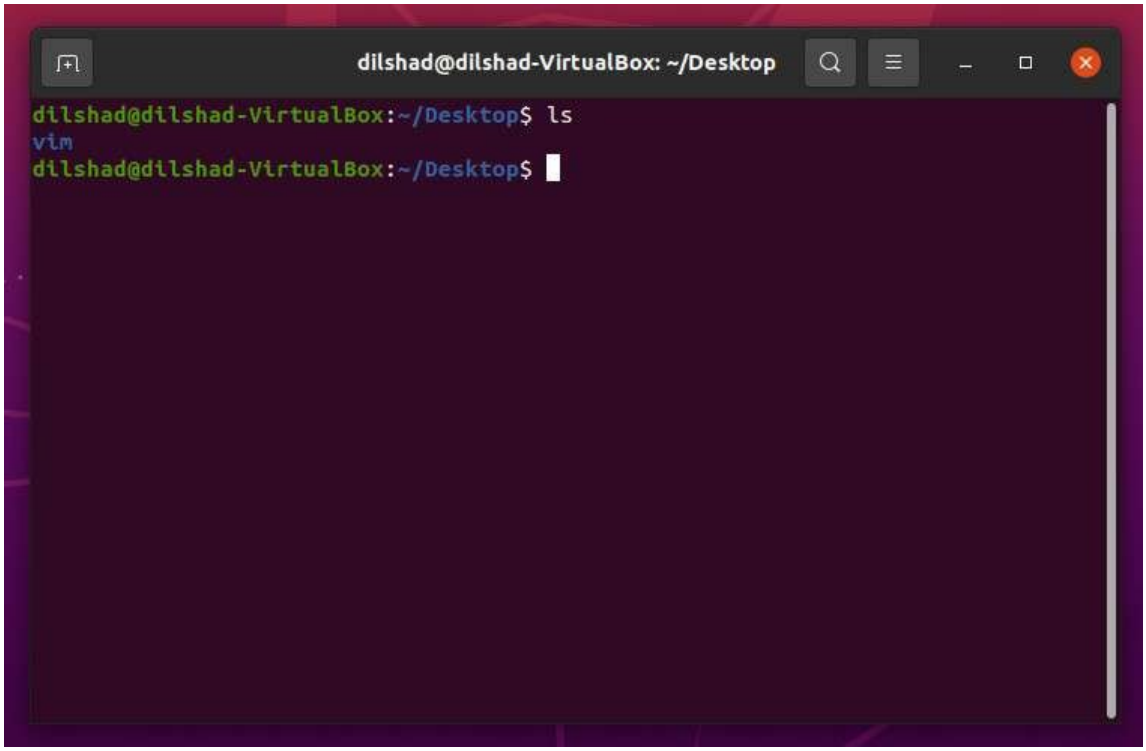
- Connects to a network and high speed Internet.
- Connects the network cable to a computer.
- This port resides on an Ethernet Card.
- Data travels at 10 megabits to 1000 megabits per seconds depending upon

the network bandwidth.

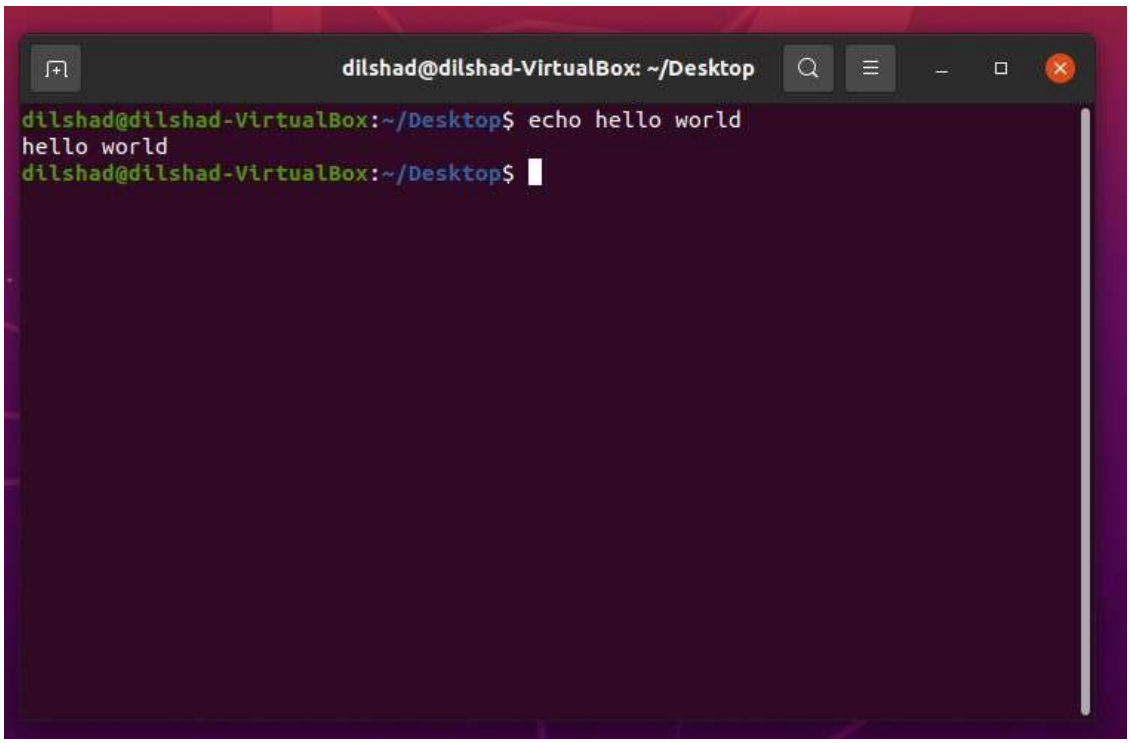


Basic Linux Commands

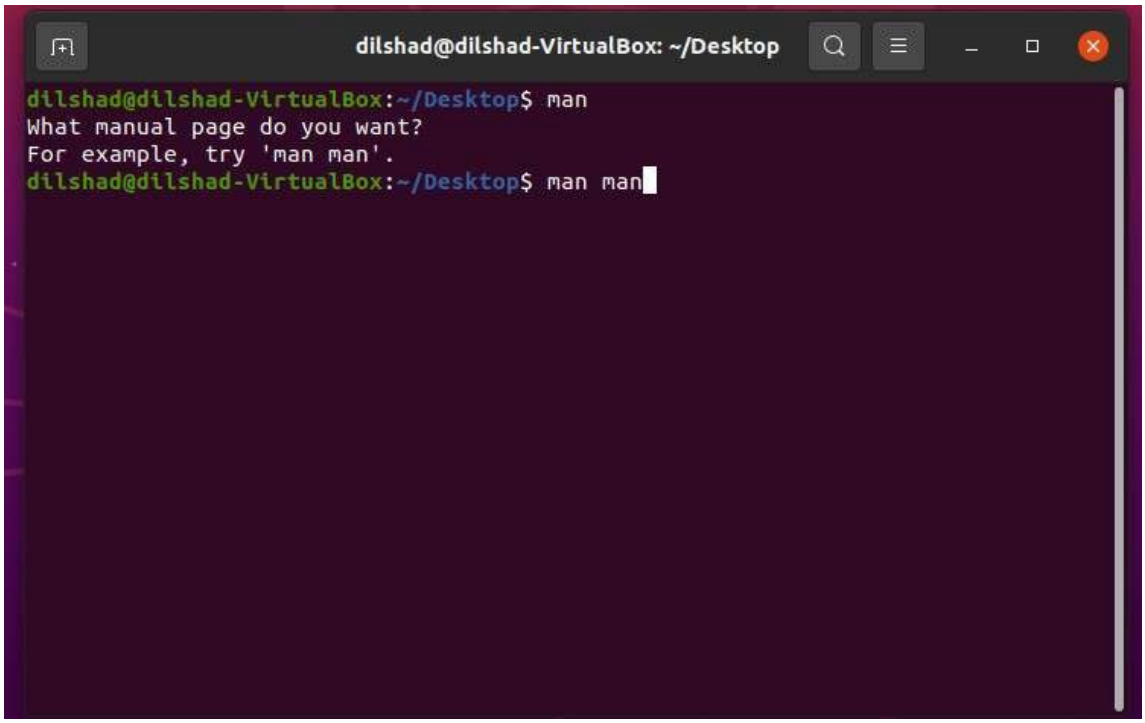
 ls :- used to list items inside a directory.

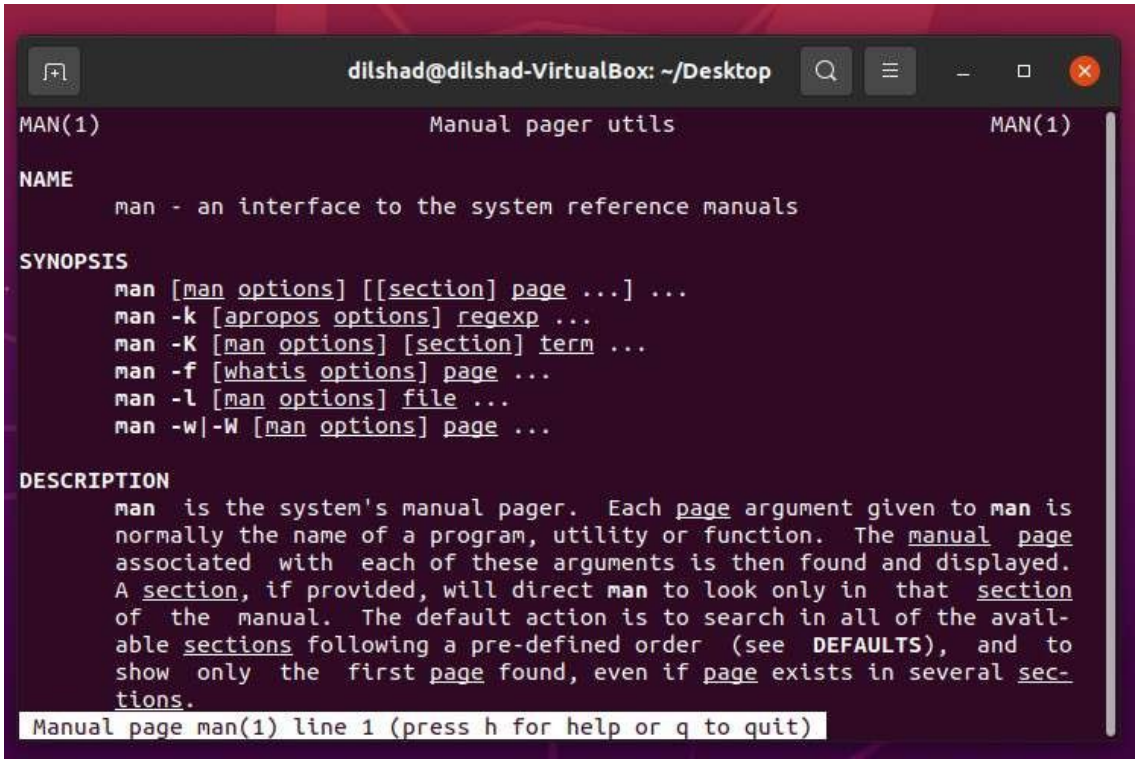


 echo :- used to display text or string.

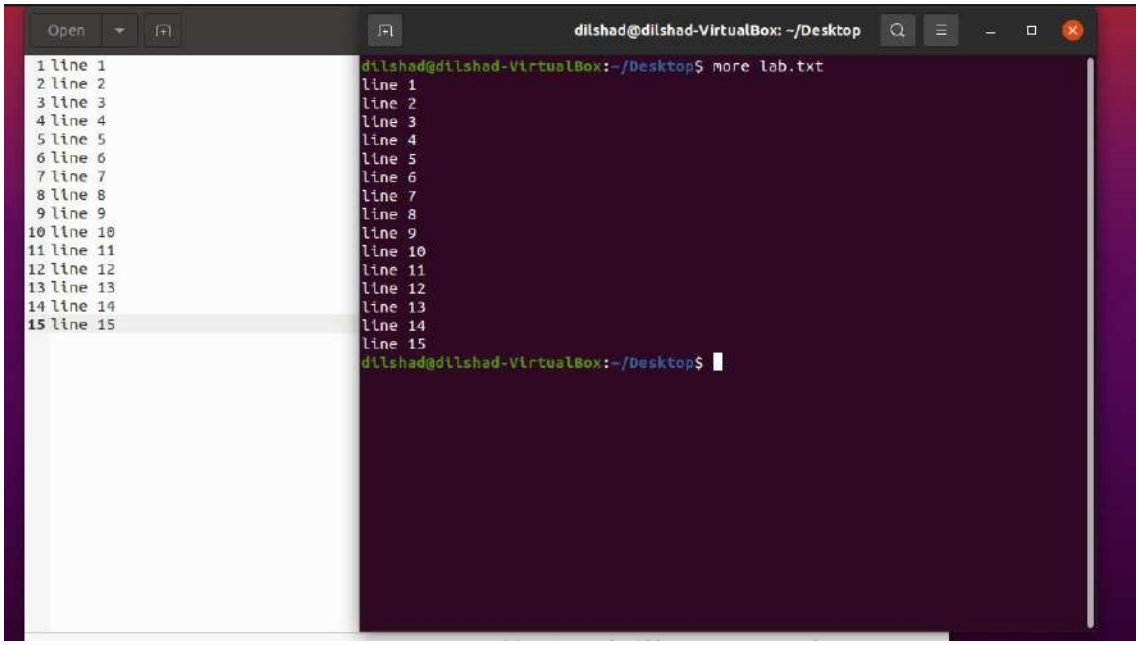


 man :-provide user manual for the command mentioned.

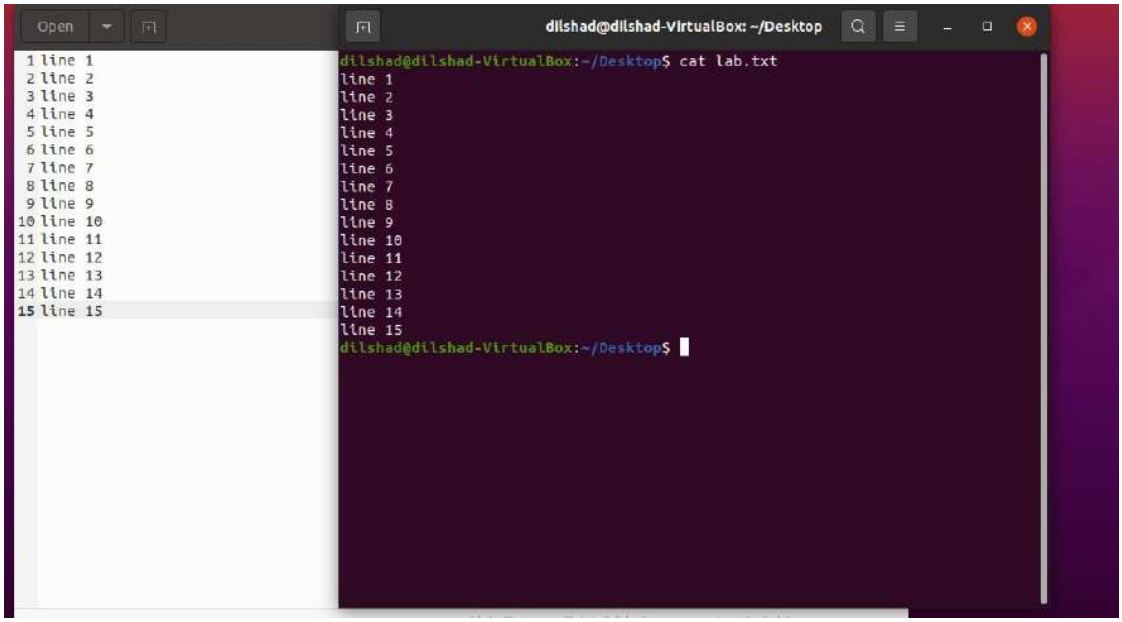




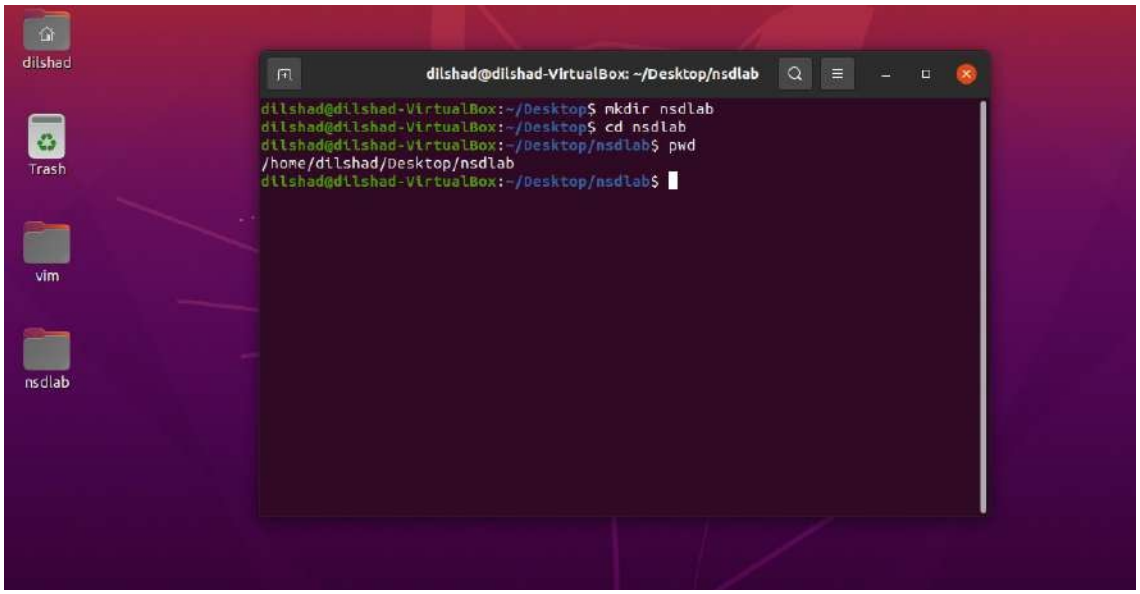
 more :- used to view text files in command prompt.



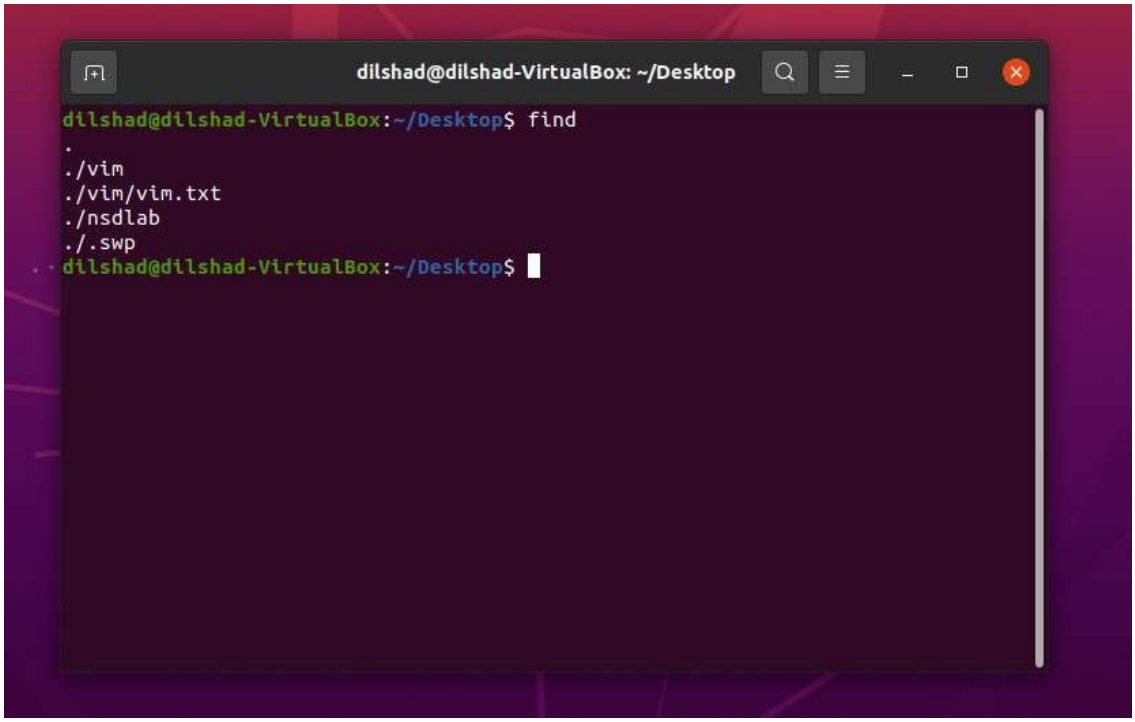
 cat :- read a file and give its content as output.



- **mkdir :-** used to create a folder.
- **Cd :-** used to change directory.
- **Pwd :-** used to print working directory.

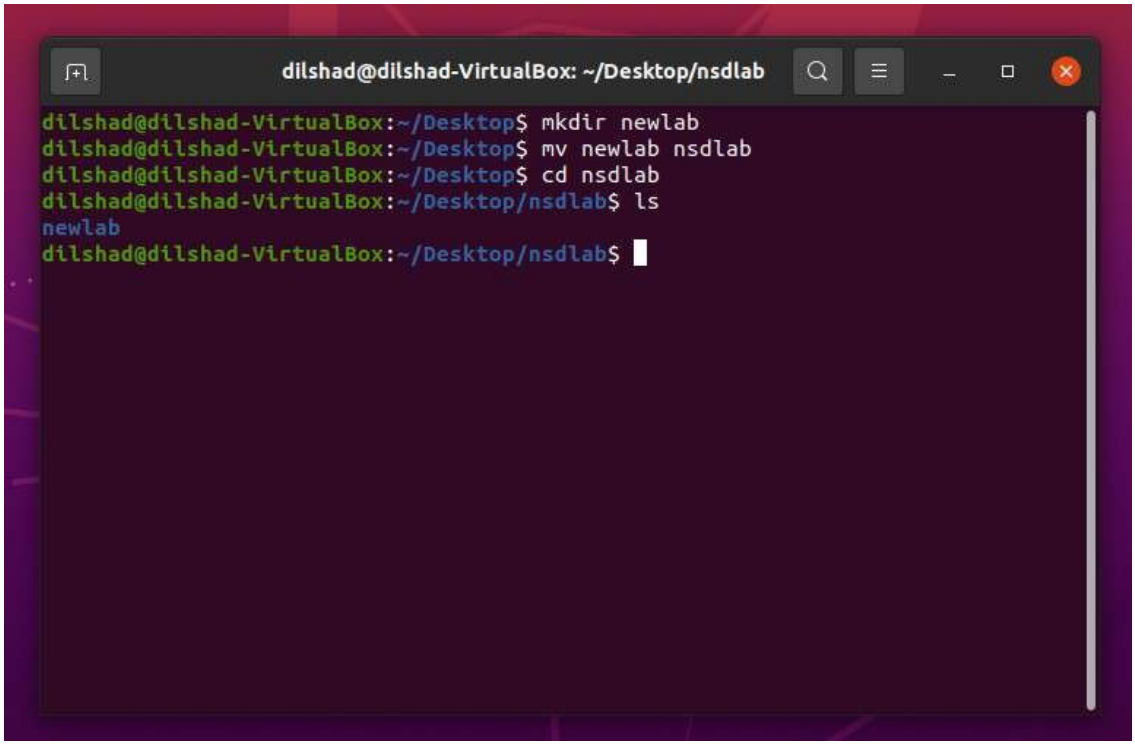


- **Find :-** gives a complete heirarchial structure of a directory.



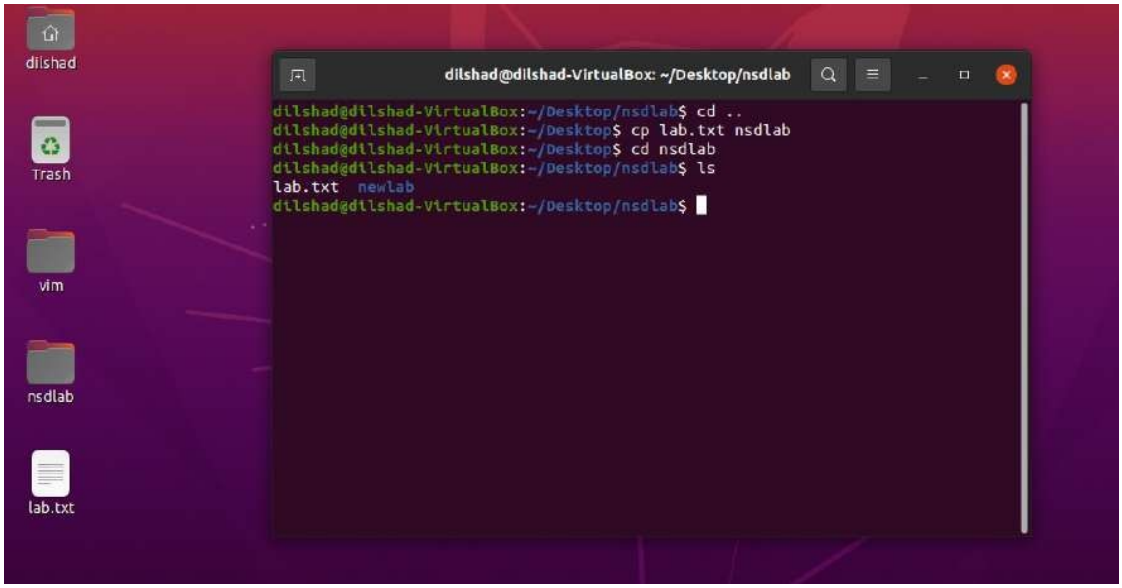
 mv :- used for moving a file or folder.

Syntax : mv (moving file)(destination)

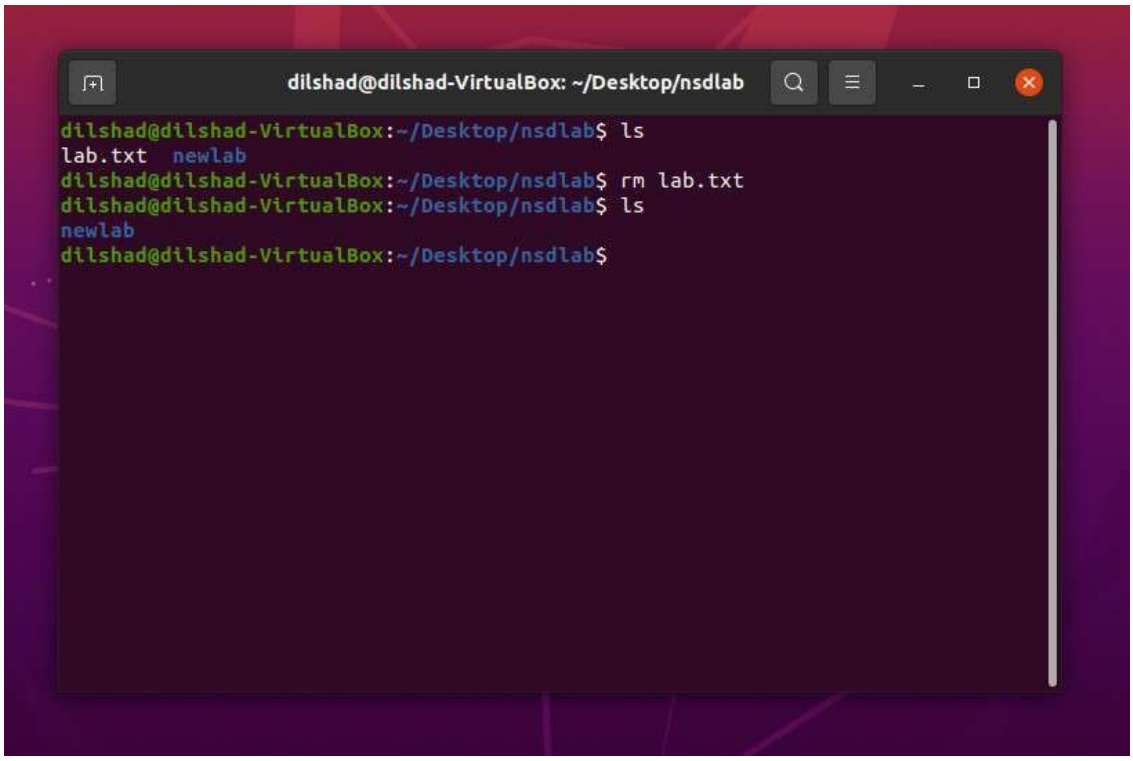


 Cp:- copy a file or folder.

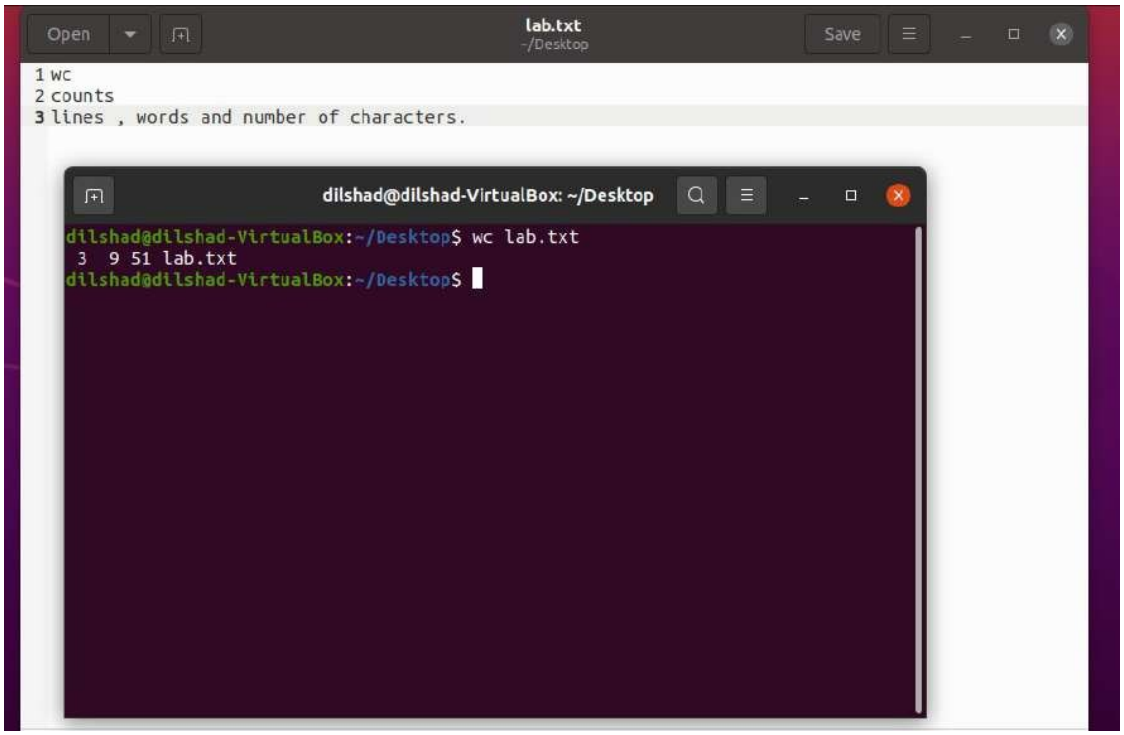
Syntax : cp copying_file destination_folder



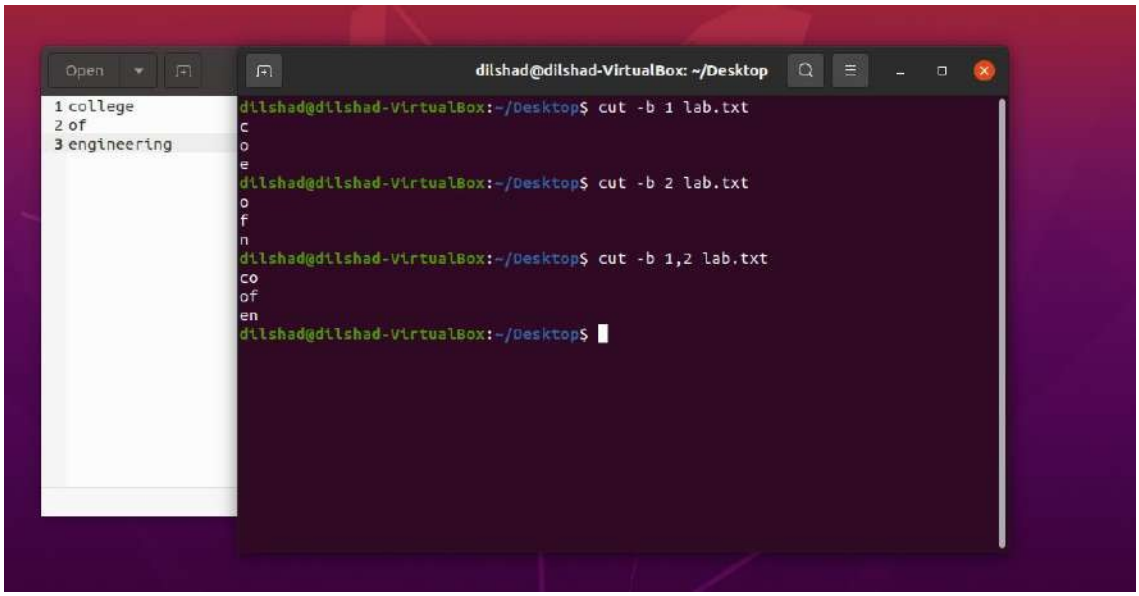
 rm :- used to remove a file.



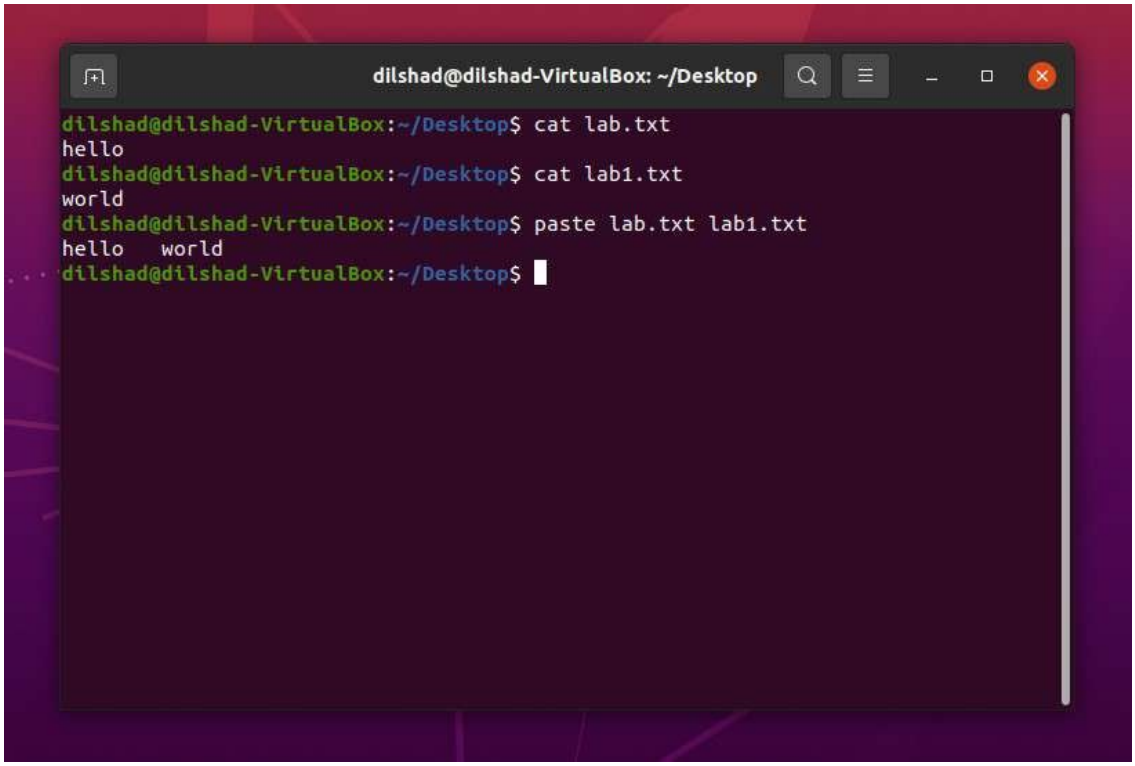
 wc :- used to count characters,words,lines in a file.



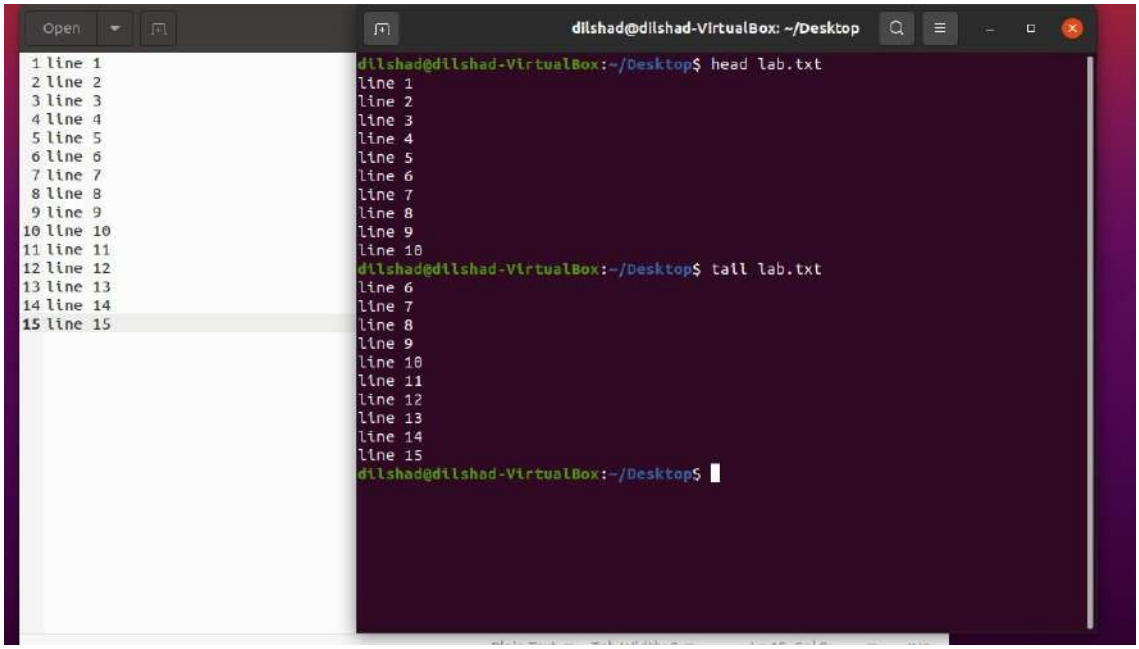
 Cut :- used to cut characters from a file.



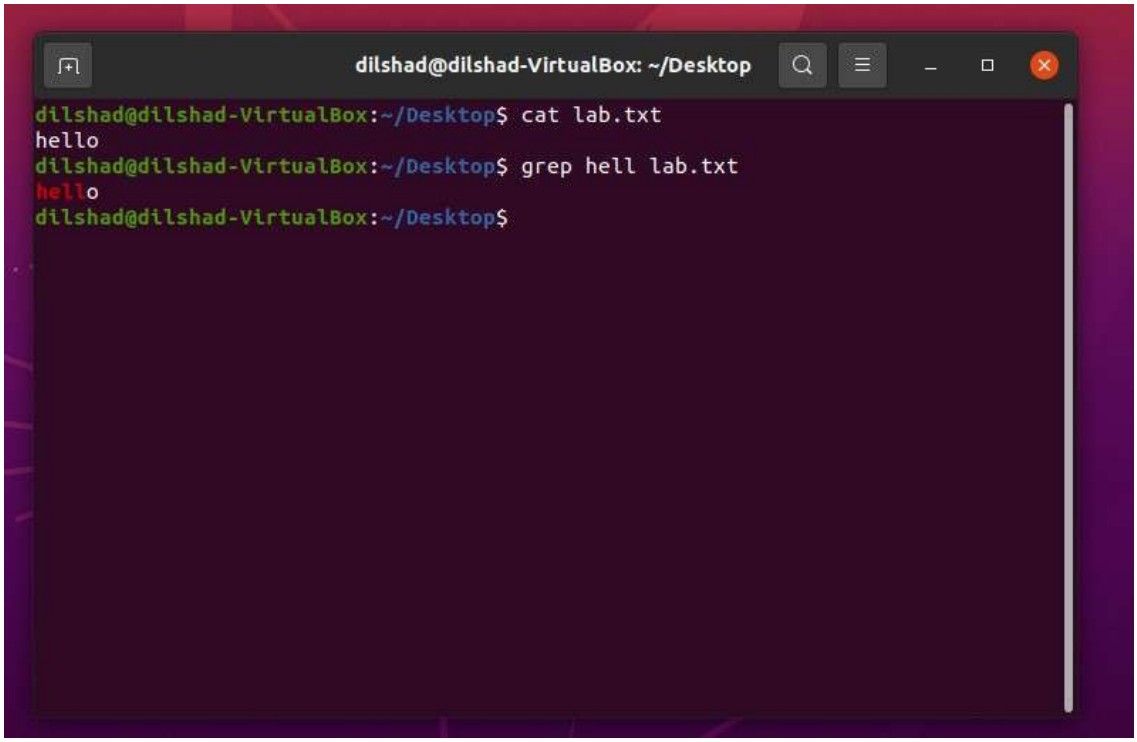
 Paste :-paste content of file1 to file2.



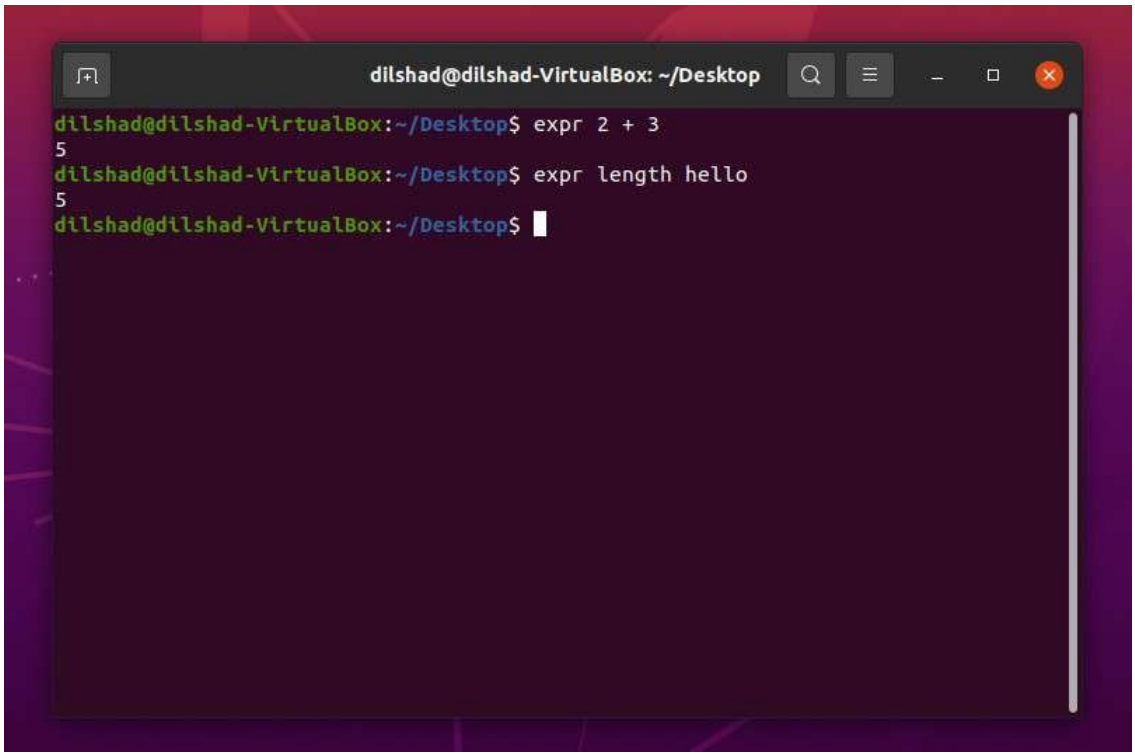
 Head:- display first 10 lines of the file.
 Tail :- display last 10 lines of a file.



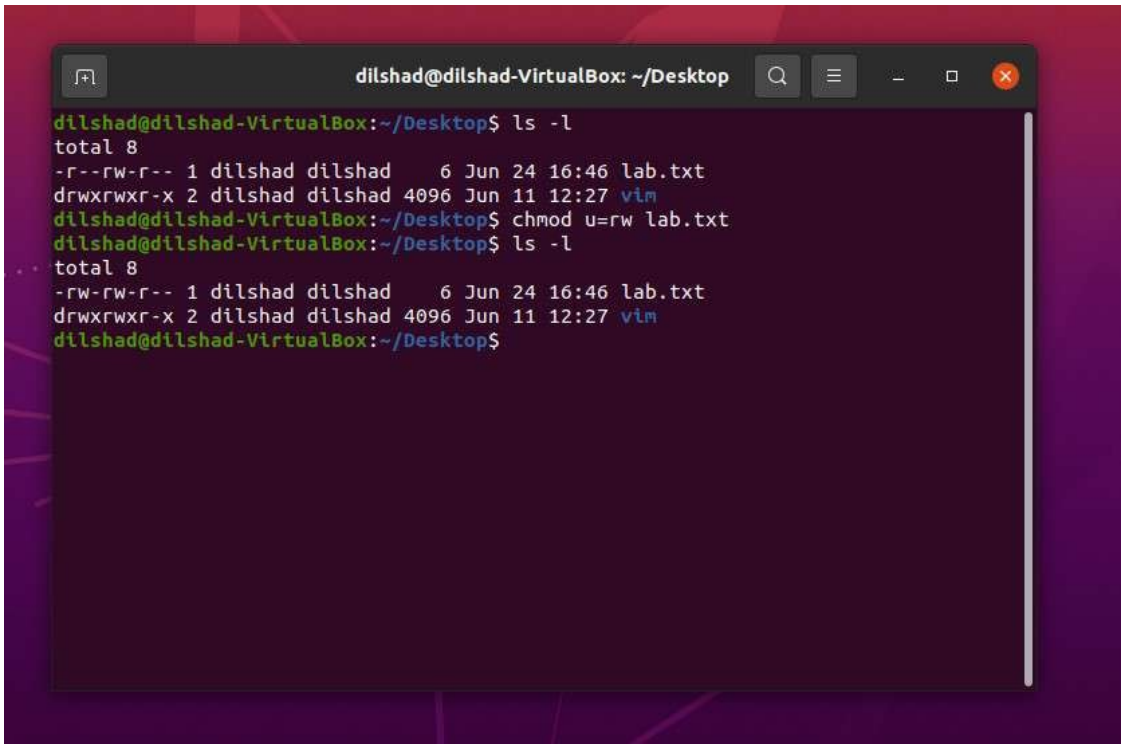
 Grep :- used to find a matching pattern in the file.



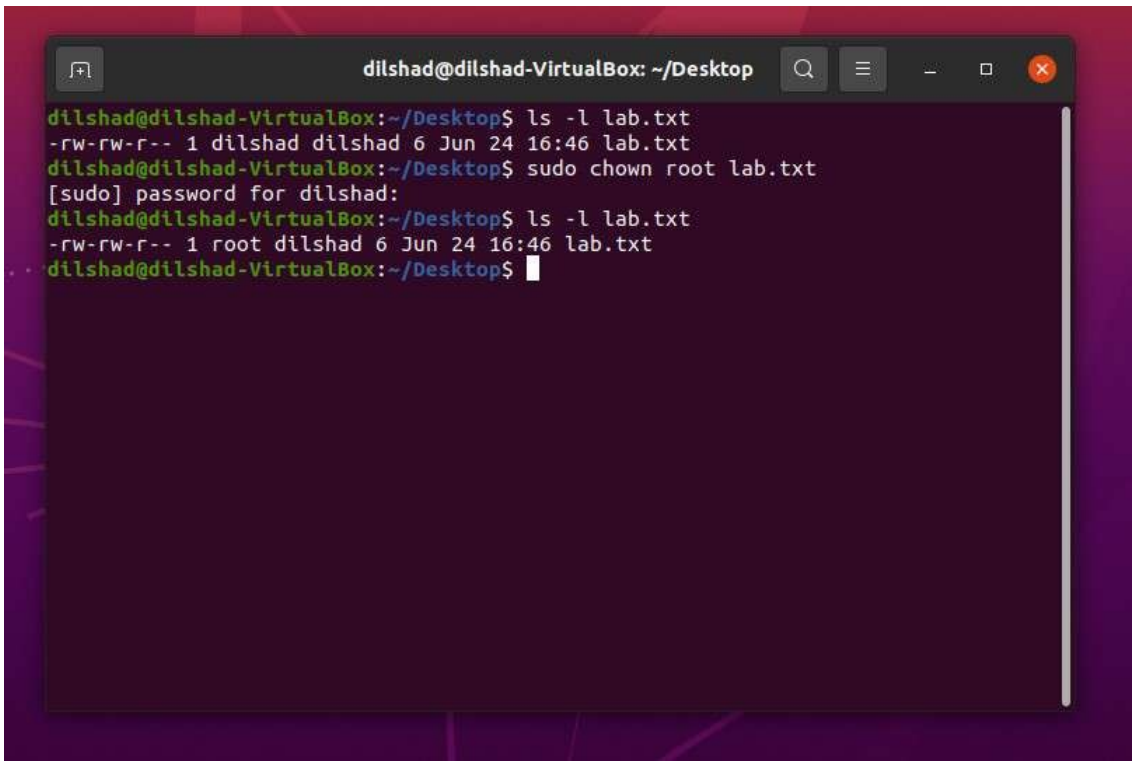
 Expr :- used to perform numerical and string operations.



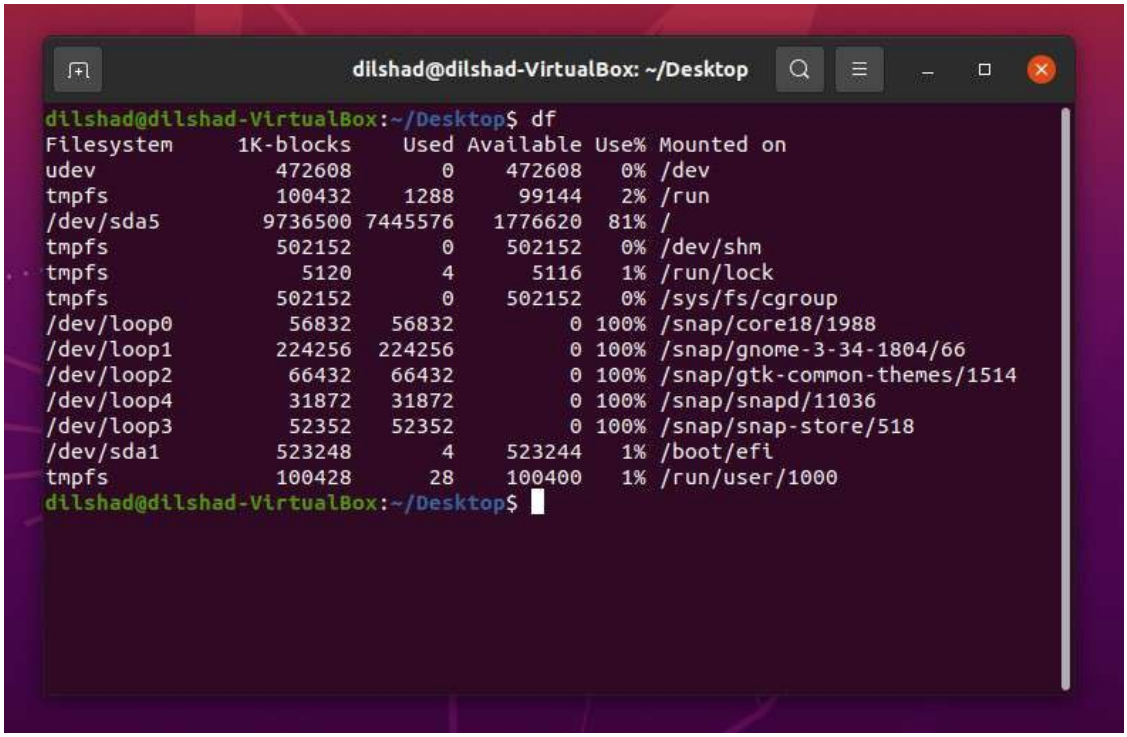
 Chmod :- used to change the operating mode of a file.read,write, read-write,execute the file.



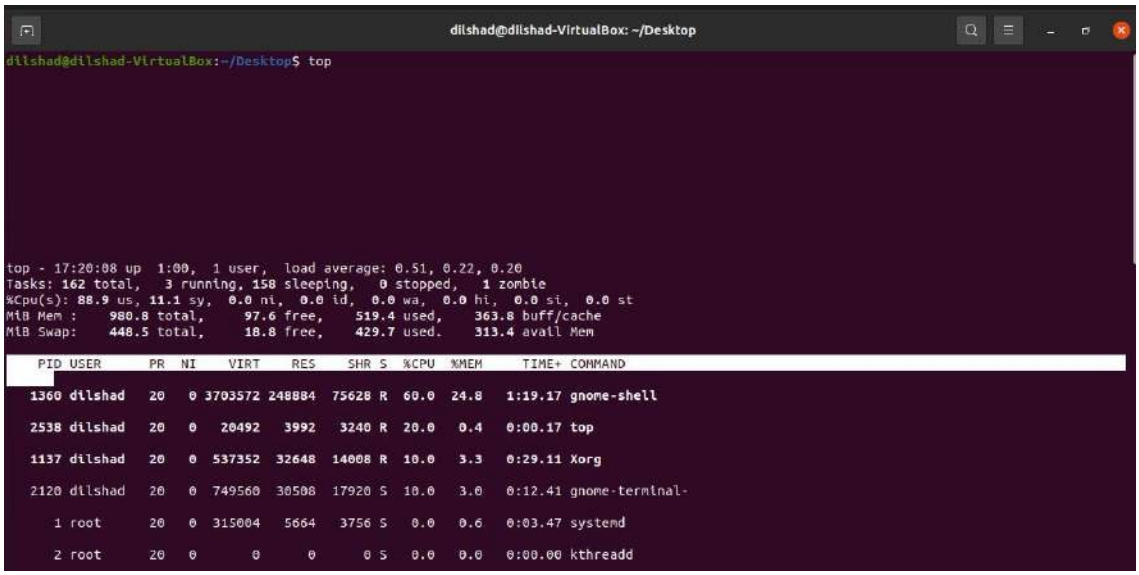
 Chown :- change the existing owner of a file.



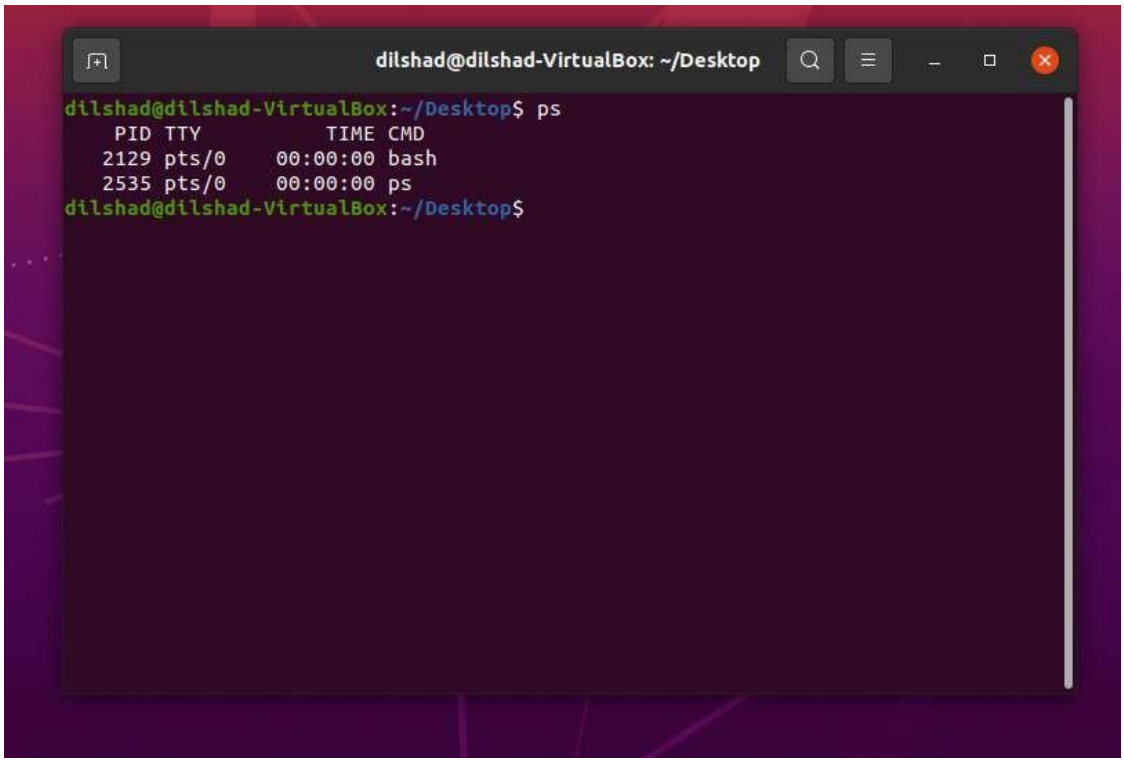
 df :- used to display available disk space for the file systems.



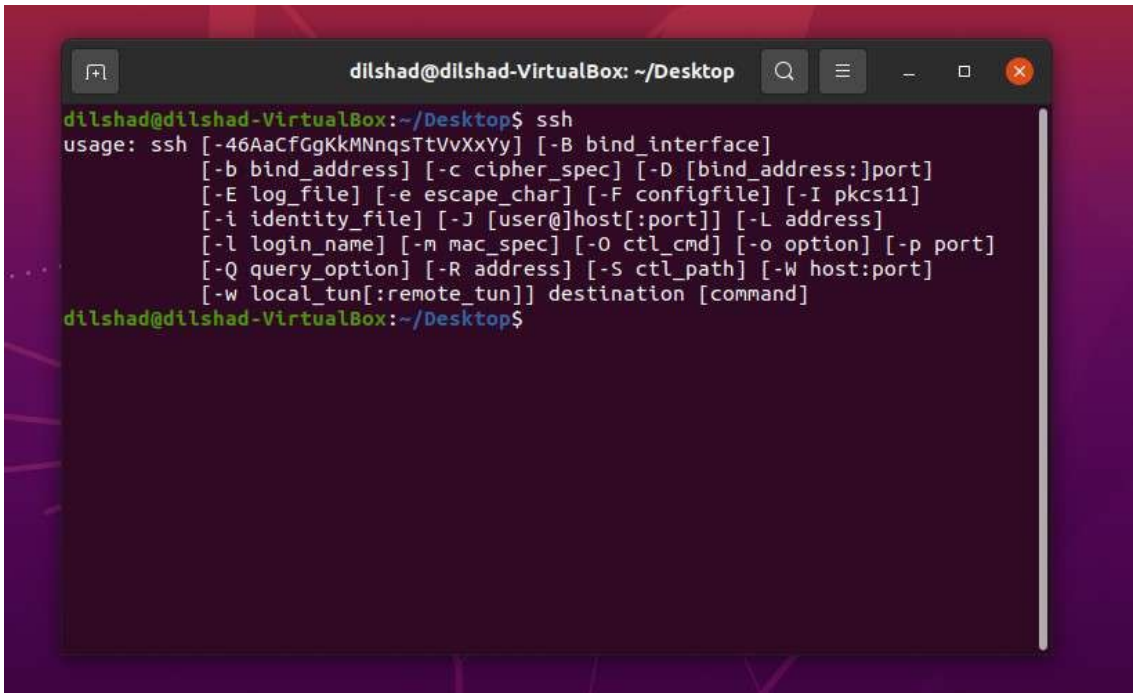
 top:-used to show processes of running systems.



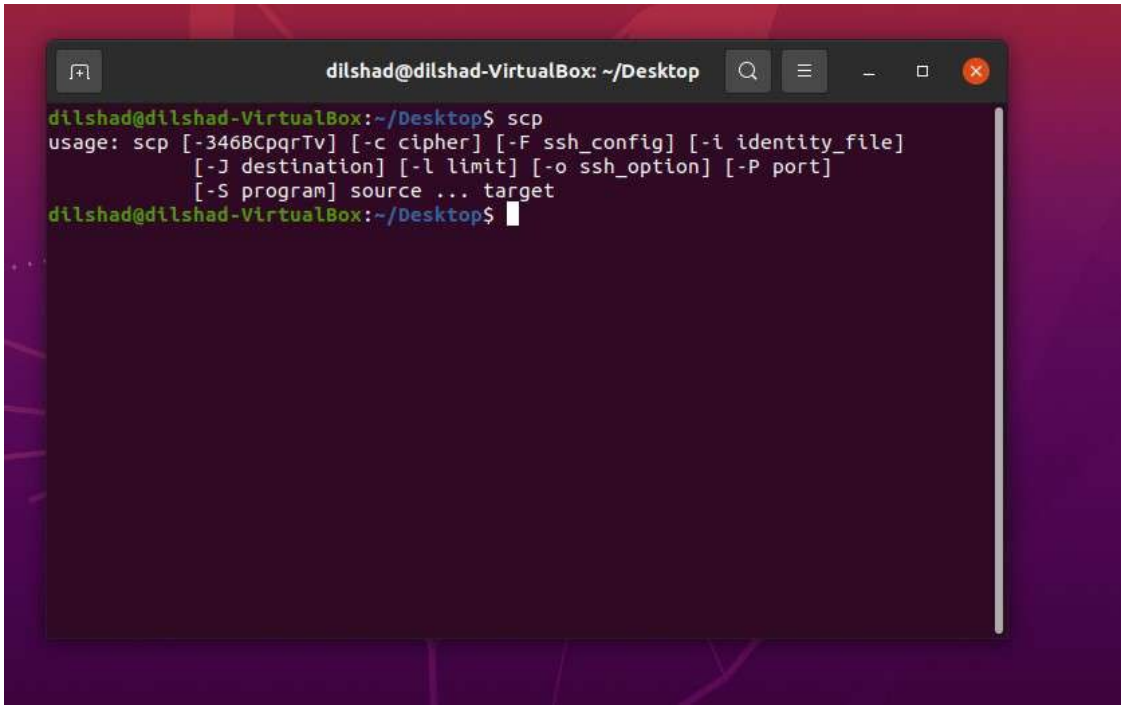
 ps:-list currently running processes.



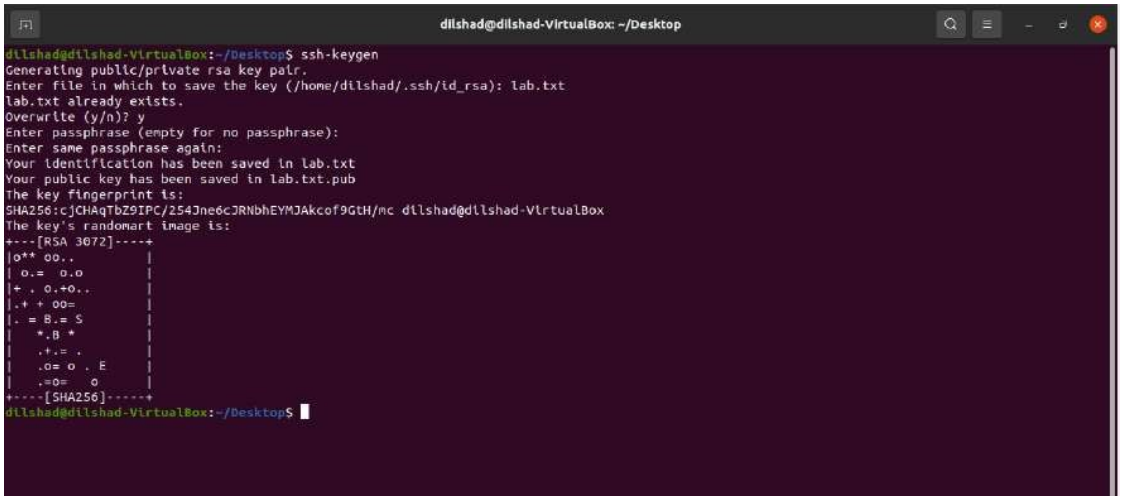
 ssh:-instruct the system to establish an encrypted secure connection with host.



 scp:- securely copy files between locations.



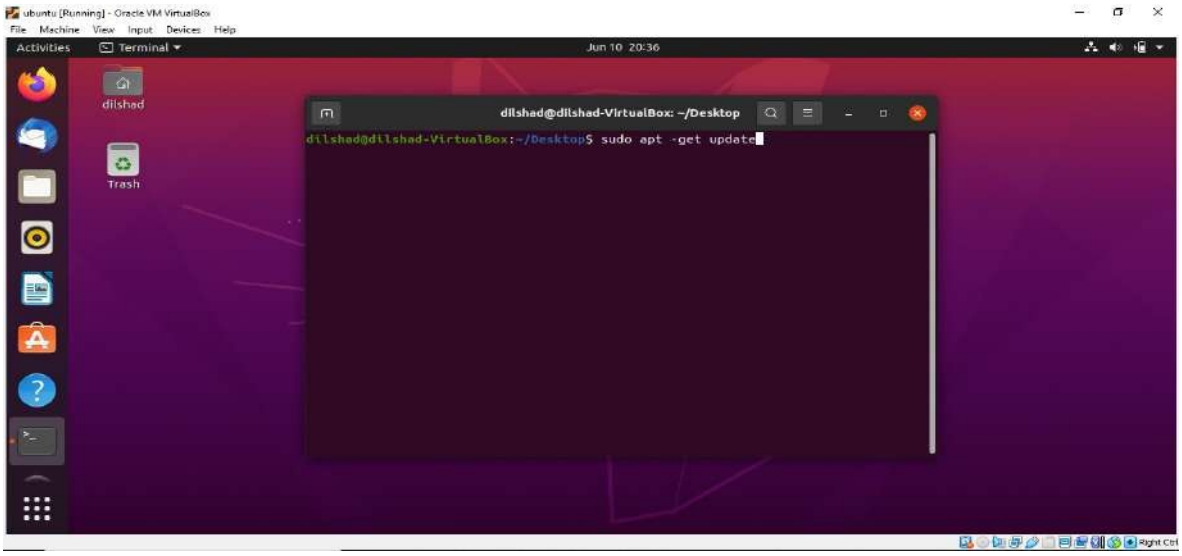
 ssh-keygen:-tool for creating new authentication key pair for **ssh.**

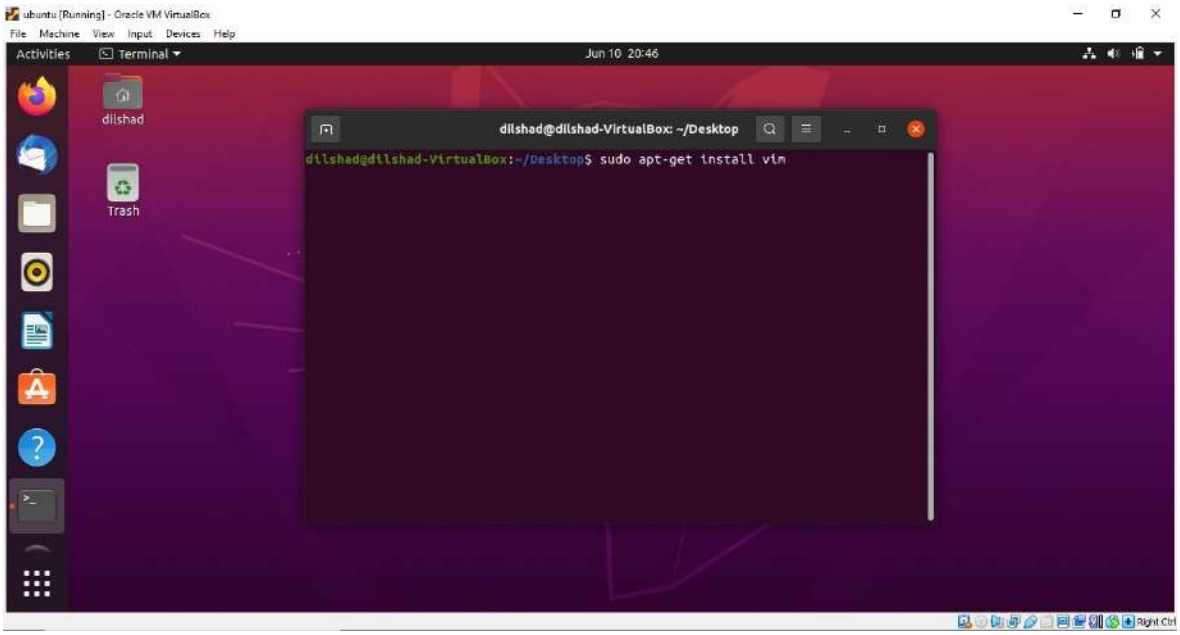


Installation of VIM Text Editor

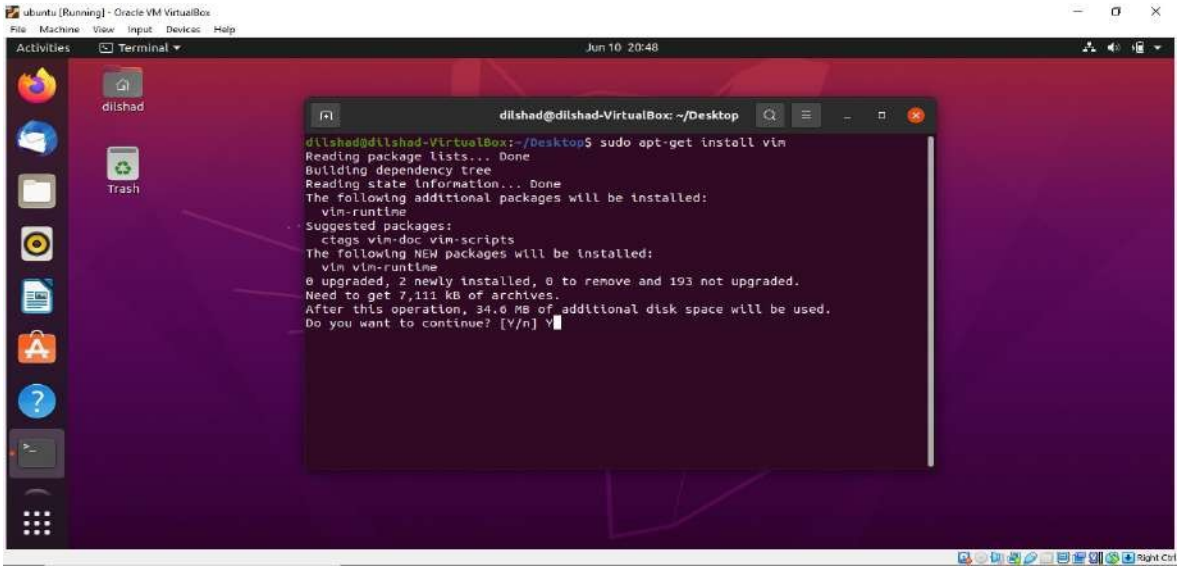
On ubuntu terminal, execute the below commands.

- ⇨ Sudo apt-get update.
- ⇨ Sudo apt-get install vim.





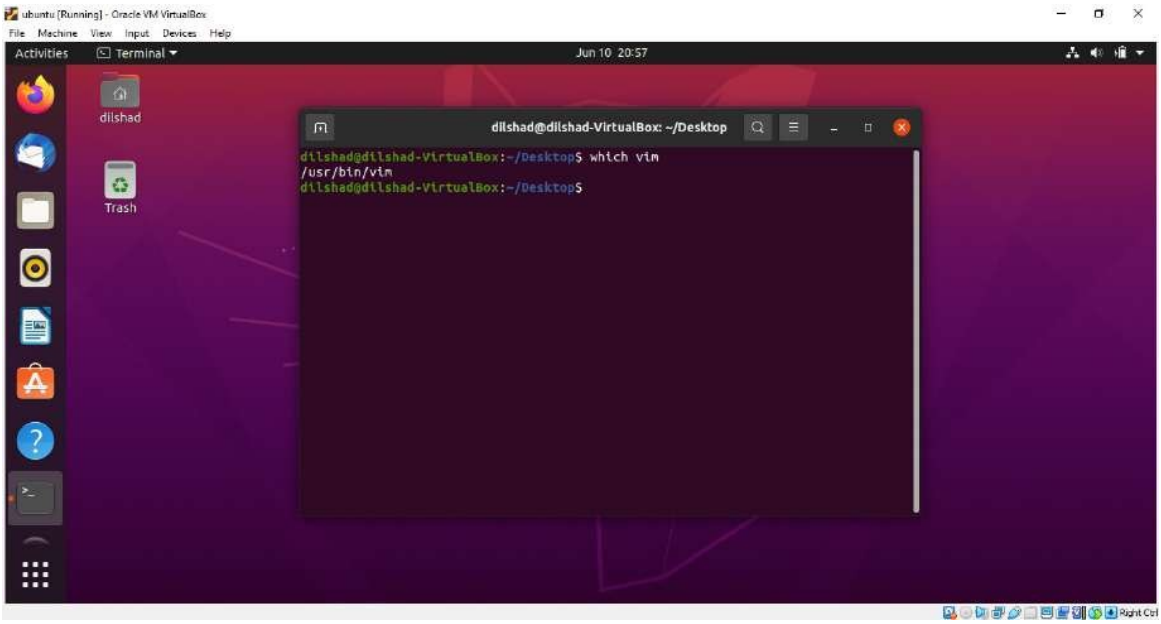
Type ‘**y**’ and press ‘**enter**’ to continue installation.



To ensure VIM is correctly installed execute below command.

⇨ ‘Which vim’

It should print the location of vim binary , if installation is successful.



Vim modes

There are some arguments as to how many modes that Vim has, but the modes you're most likely to use are command mode and insert mode. These modes will allow you to do just about anything you need, including creating your document, saving your document and doing advanced editing, including taking advantage of search and replace functions.

Command mode

This is the default mode that you'll be in once you open Vim. If you're in a different mode and want to go back to command mode, just hit the Escape key. This mode allows you to use Vim commands and move through your document. From command mode, you can also use last-:line commands, which generally start with the use of a colon. For example, :w saves your file and :q allows you to exit Vim.

Insert mode

This mode allows you to enter text into your document. You can enter insert mode by pressing the i key. Keep in mind that to save your document, you'll need to go **back to command mode** since only text input is allowed in this mode.

When you start vim, you begin in Command Mode by default ,Hitting ESCAPE will get you back to Command Mode from other modes.

Last line mode

- From command mode press :

- ⇨ Cursor jump to the last line on the screen
- ⇨ Here you can manage files, issue shell commands, change editor settings
- ⇨ Also where you go to exit.

Command mode commands :-

When using movement commands, you can put a number in front of them to make Vim complete a command multiple times. For example, 5h will move your cursor five spaces to the left, and 90j will put your cursor at the beginning of the 90th line down from where your cursor currently is.

- ⇨ **h** -: Moves the cursor to the left.
- ⇨ **l** -: Moves the cursor to the right.
- ⇨ **j** -: Moves the cursor down one line.
- ⇨ **k** -: Moves the cursor up one line.
- ⇨ **H** -: Puts the cursor at the top of the screen.
- ⇨ **M** -: Puts the cursor in the middle of the screen.
- ⇨ **L** -: Puts the cursor at the bottom of the screen.
- ⇨ **w** -: Puts the cursor at the start of the next word.
- ⇨ **b** -: Puts the cursor at the start of the previous word.
- ⇨ **e** -: Puts the cursor at the end of a word.
- ⇨ **0** -: Places the cursor at the beginning of a line.
- ⇨ **$** -: Places the cursor at the end of a line.
- ⇨ **)** -: Takes you to the start of the next sentence.

- **(** -: Takes you to the start of the previous sentence.
- **}** -: Takes you to the start of the next paragraph or block of text.
- **{** -: Takes you to the start of the previous paragraph or block of text.
- **Ctrl** + **f** -: Takes you one page forward.
- **Ctrl** + **b** -: Takes you one page back.
- **gg** -: Places the cursor at the start of the file.
- **G** -: Places the cursor at the end of the file.
- # -: Where # is the number of a line, this command takes you to the line specified.

Insert mode :-

putting a number in front of yy will increase the number of lines copied, so 5yy will copy five lines.

- **i :-** insert before cursor.
- **I :-** insert before first nonblank character on line.
- **a :-** insert after cursor.
- **A :-** insert at end of line .
- **o:-** pen line below .
- **O**:- pen line above .
- **r :-** replace current character .
- **R:-** replace characters
- **yy** -: Copies a line.

- ⇨ **yw** -: Copies a word.
- ⇨ **y$** -: Copies from where your cursor is to the end of a line.
- ⇨ **v** -: Highlight one character at a time using arrow buttons or the h, k, j, l buttons.
- ⇨ **V** -: Highlights one line, and movement keys can allow you to highlight additional lines.
- ⇨ **p** -: Paste whatever has been copied to the unnamed register.
- ⇨ **d** -: Deletes highlighted text.
- ⇨ **dd** -: Deletes a line of text.
- ⇨ **dw** -: Deletes a word.
- ⇨ **D** -: Deletes everything from where your cursor is to the end of the line.
- ⇨ **d0** -: Deletes everything from where your cursor is to the beginning of the line.
- ⇨ **dgg** -: Deletes everything from where your cursor is to the beginning of the file.
- ⇨ **dG** -: Deletes everything from where your cursor is to the end of the file.
- ⇨ **x** -: Deletes a single character.
- ⇨ **u** -: Undo the last operation; u# allows you to undo multiple actions.
- ⇨ **Ctrl** + **r** -: Redo the last undo.
- ⇨ **.** -: Repeats the last action.

Last line mode :-:

- ⇨ **w** -: write file.

- **b** -: quit .
- **w!** -: write read-:only file .
- **q!** -: quit without saving changes .
- **e** *filename* -: opens a file for editing.

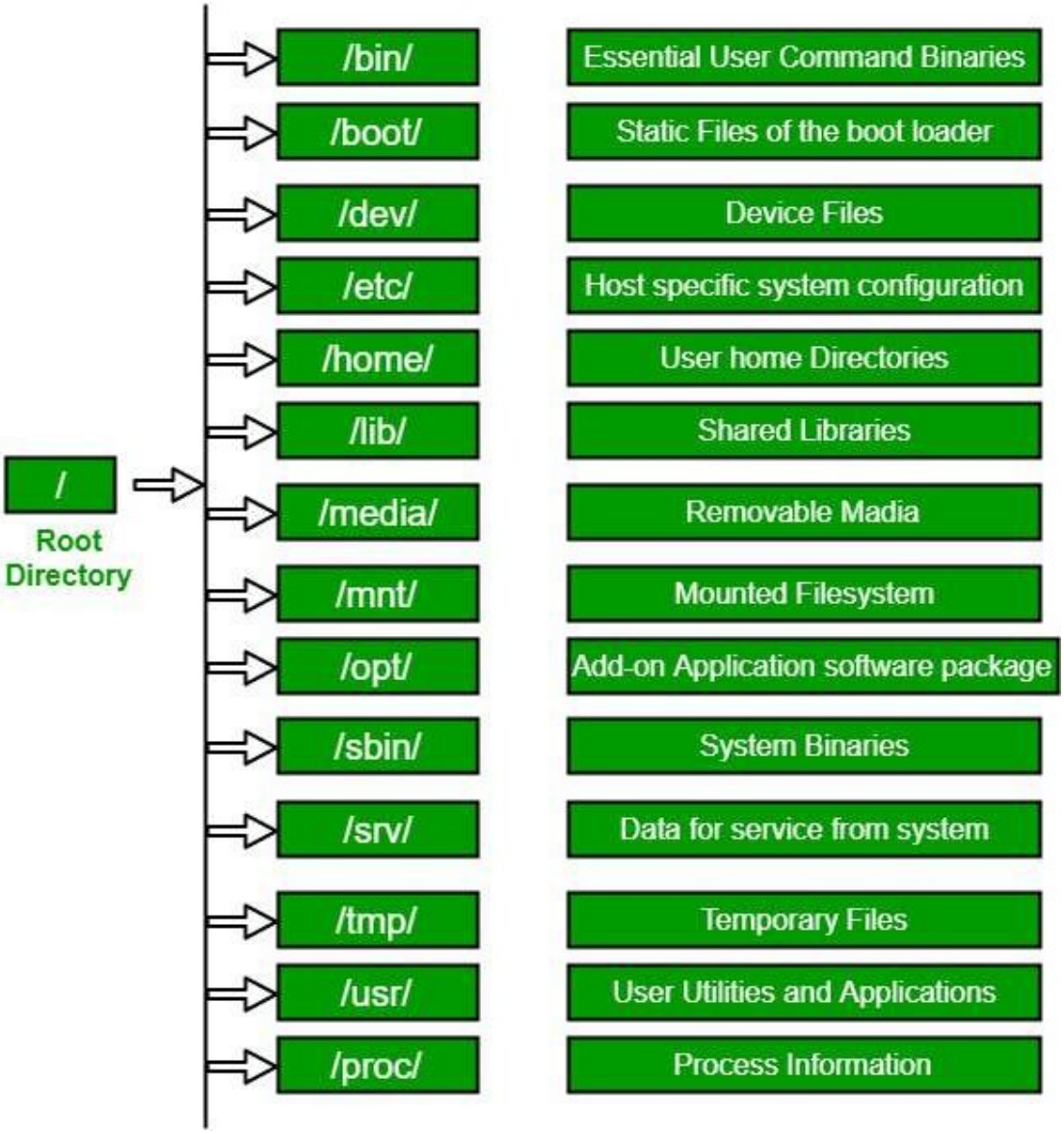
Experiment -3

Aim :- File system hierarchy in a common Linux distribution, file and device permissions, study of system configuration files in /etc, familiarizing log files for system events, user activity, network events.

File System Hierarchy in Linux :-

The Linux File Hierarchy Structure or the File system Hierarchy Standard (FHS) defines the directory structure and directory contents in Unix-like operating systems. It is maintained by the Linux Foundation.

- In the FHS, all files and directories appear under the root directory /, even if they are stored on different physical or virtual devices.
- Some of these directories only exist on a particular system if certain subsystems, such as the X Window System, are installed.
- Most of these directories exist in all UNIX operating systems and are generally used in much the same way; however, the descriptions here are those used specifically for the FHS and are not considered authoritative for platforms other than Linux.



1. / (Root): Primary hierarchy root and root directory of the entire file system hierarchy.

- Every single file and directory starts from the root directory.
- The only root user has the right to write under this directory.
- /root is the root user's home directory, which is not the same as /.

2. /bin : Essential command binaries that need to be available in single-user mode for all users, e.g., cat, ls, cp. .

- Contains binary executables.
- Common linux commands you need to use in single-user modes are located under this directory.
- Commands used by all the users of the system are located here e.g. ps, ls, ping, grep, cp.

3. /boot : Boot loader files, e.g., kernels, initrd.

- Kernel initrd, vmlinux, grub files are located under /boot
- Example: initrd.img-2.6.32-24-generic, vmlinuz-2.6.32-24-generic.

4. /dev : Essential device files, e.g., /dev/null.

- These include terminal devices, usb, or any device attached to the system.

  Example: /dev/tty1, /dev/usbmon0

5. /etc : Host-specific system-wide configuration files.

- Contains configuration files required by all programs.
- This also contains startup and shutdown shell scripts used to start/stop individual programs.

  Example: /etc/resolv.conf, /etc/logrotate.conf.

6. /home : Users' home directories, containing saved files, personal settings, etc.

- Home directories for all users to store their personal files.

  example: /home/kishlay, /home/kv

7. /lib : Libraries essential for the binaries in /bin/ and /sbin/.

- Library filenames are either ld* or lib*.so.*

Example: ld-2.11.1.so, libncurses.so.5.7

8. /mnt : Temporarily mounted filesystems.

- Temporary mount directory where sysadmins can mount filesystems.

9. /opt : Optional application software packages.

- Contains add-on applications from individual vendors.
- Add-on applications should be installed under either /opt/ or /opt/ sub-directory.

10. /tmp : Temporary files. Often not preserved between system reboots, and may be severely size restricted.

- Directory that contains temporary files created by system and users.
- Files under this directory are deleted when system is rebooted.

11. /usr : Secondary hierarchy for read-only user data; contains the majority of (multi-)user utilities and applications.

- Contains binaries, libraries, documentation, and source-code for second level programs.
- /usr/bin contains binary files for user programs. If you can't find a user binary under /bin, look under /usr/bin. For example: at, awk, cc, less, scp.
- /usr/sbin contains binary files for system administrators. If you can't find a system binary under /sbin, look under /usr/sbin. For example: atd, cron, sshd, useradd, userdel.
- /usr/lib contains libraries for /usr/bin and /usr/sbin.

- /usr/local contains users programs that you install from source. For example, when you install apache from source, it goes under /usr/local/apache2.
- /usr/src holds the Linux kernel sources, header-files and documentation.

12. /proc : Virtual filesystem providing process and kernel information as files. In Linux, corresponds to a procfs mount. Generally automatically generated and populated by the system, on the fly.

- Contains information about system process.
- This is a pseudo filesystem contains information about running process. For example: /proc/{pid} directory contains information about the process with that particular pid.
- This is a virtual filesystem with text information about system resources. For example: /proc/uptime

File And Device Permissions :-

Every file and directory on your Unix/Linux system is assigned 3 types of owner.

User

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.

File Permissions :-

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

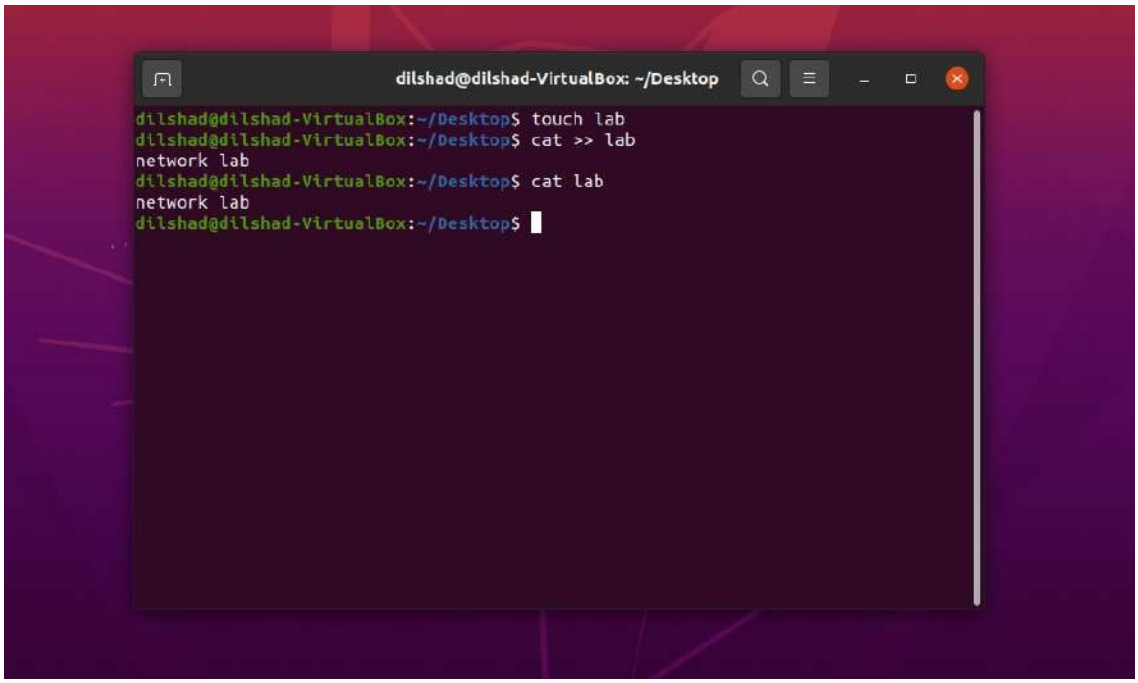
Read: This permission give you the authority to open and read a file. Read permission on a directory gives you the ability to lists its content.

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 to write permission on 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.

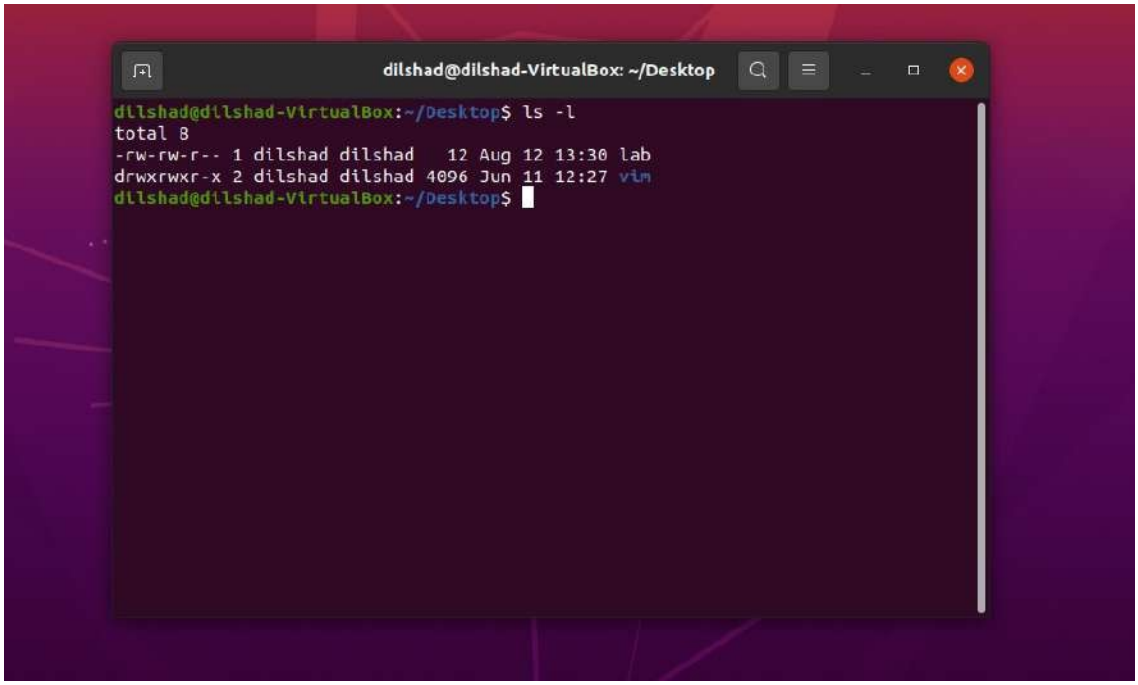
Execute: In Windows, an executable program usually has an extension ".exe" and which you can easily run. In Unix/Linux, you cannot run a program unless the execute permission is set. If the execute permission is not set, you might still be able to see/modify the program code(provided read & write permissions are set), but not run it.

Example :-

Create a new file and add some text in it.



Use the command ***ls –l*** to show the current permission details of the file.



- It shows a minus(-) In front of permission of lab and *d* for the vim. This is to show that minus(-) determines it as a file and *d* determines it as a directory.
- In the above example the file lab have the permission with 3 bit representation for the 3 types of owners.
- First 3 bit for user,then next 3 bit for group and last 3 bit for others/world.
- Here user have read and write permission on lab file,group also has read and write permission on file and others can have only read permission.

Changing file/directory permissions with 'chmod' command

Say you do not want your colleague to see your personal images. This can be achieved by changing file permissions.

We can use the '**chmod'** command which stands for 'change mode'. Using the command, we can set permissions (read, write, execute) on a file/directory for the owner, group and the world.

Syntax:

chmod permissions filename

There are 2 ways to use the command -

1. **Absolute mode**
2. **Symbolic mode**

Absolute(Numeric) Mode

In this mode, file **permissions are not represented as characters but a three-digit octal number**.

The table below gives numbers for all for permissions types.

Number	Permission Type	Symbol
0	No Permission	---
1	Execute	--x

2	Write	-w-
3	Execute + Write	-wx
4	Read	r--
5	Read + Execute	r-x
6	Read +Write	rw-
7	Read + Write +Execute	Rwx

Symbolic Mode

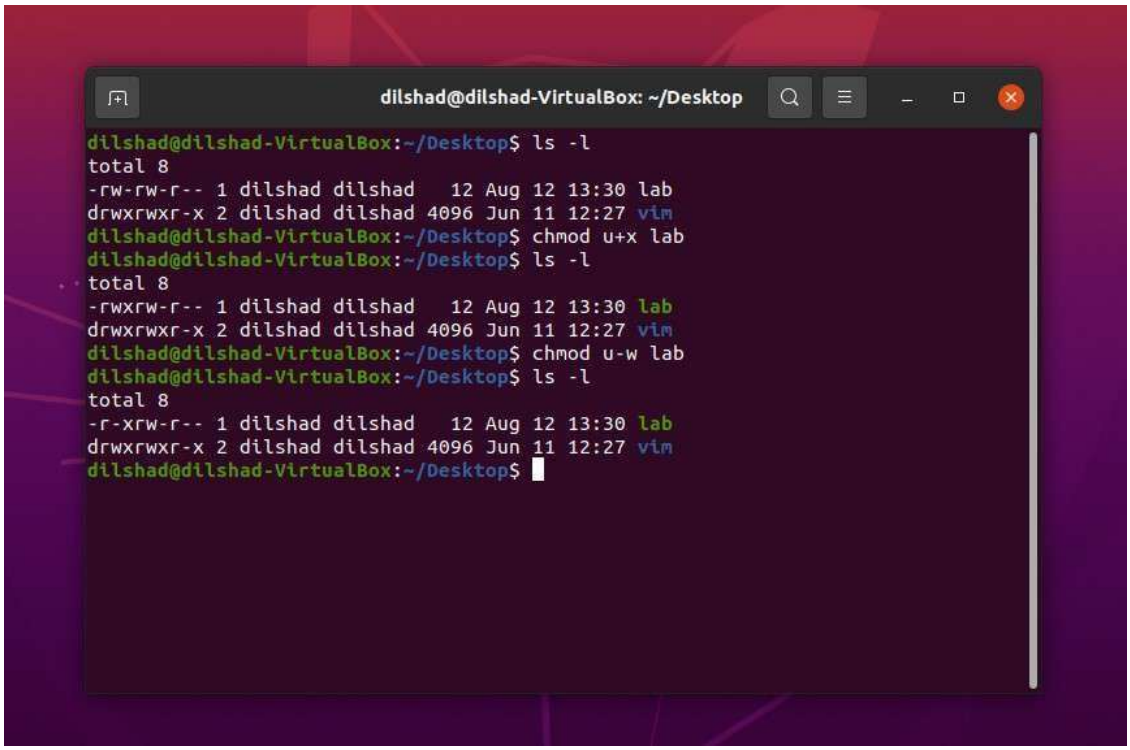
In the Absolute mode, you change permissions for all 3 owners. In the symbolic mode, you can modify permissions of a specific owner. It makes use of mathematical symbols to modify the Unix file permissions.

Operator	Description
+	Adds a permission to a file or directory
-	Removes the permission
=	Sets the permission and overrides the permissions set earlier.

The various owners are represented as -

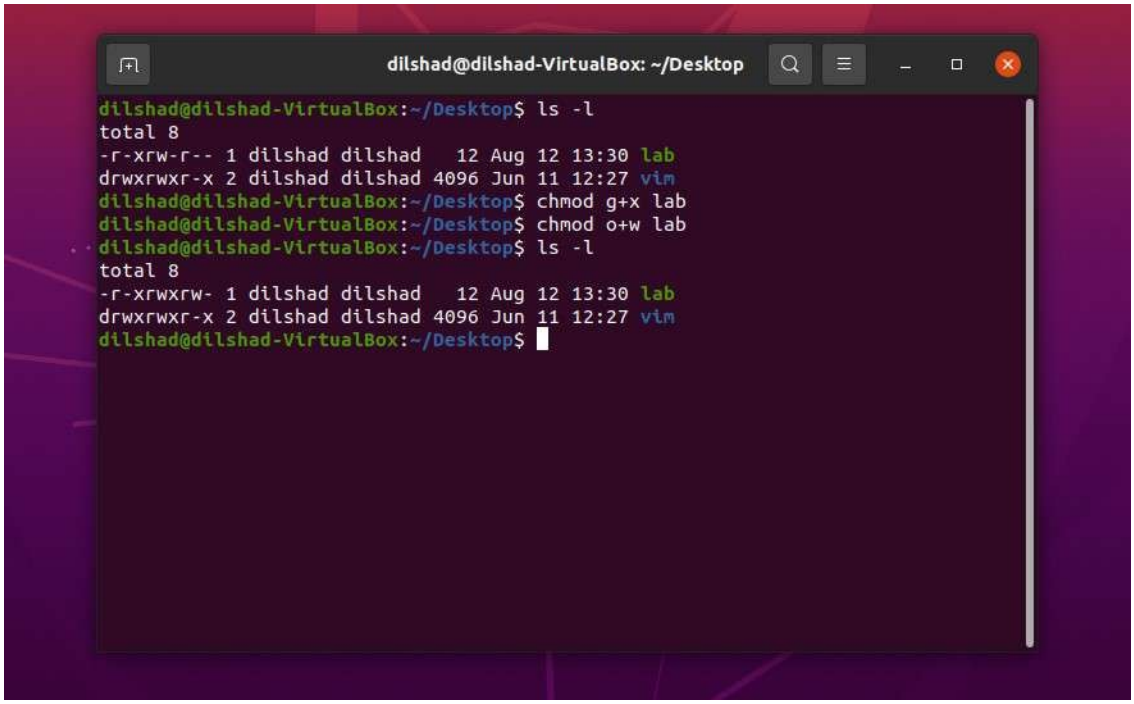
User Denotations	
U	user/owner
G	Group
O	Other
A	All

Example 1 (symbolic mode):- Chmod is the command used to change permission mode of a file.



In symbolic mode + uses to add a permission and – uses for removing a permission.

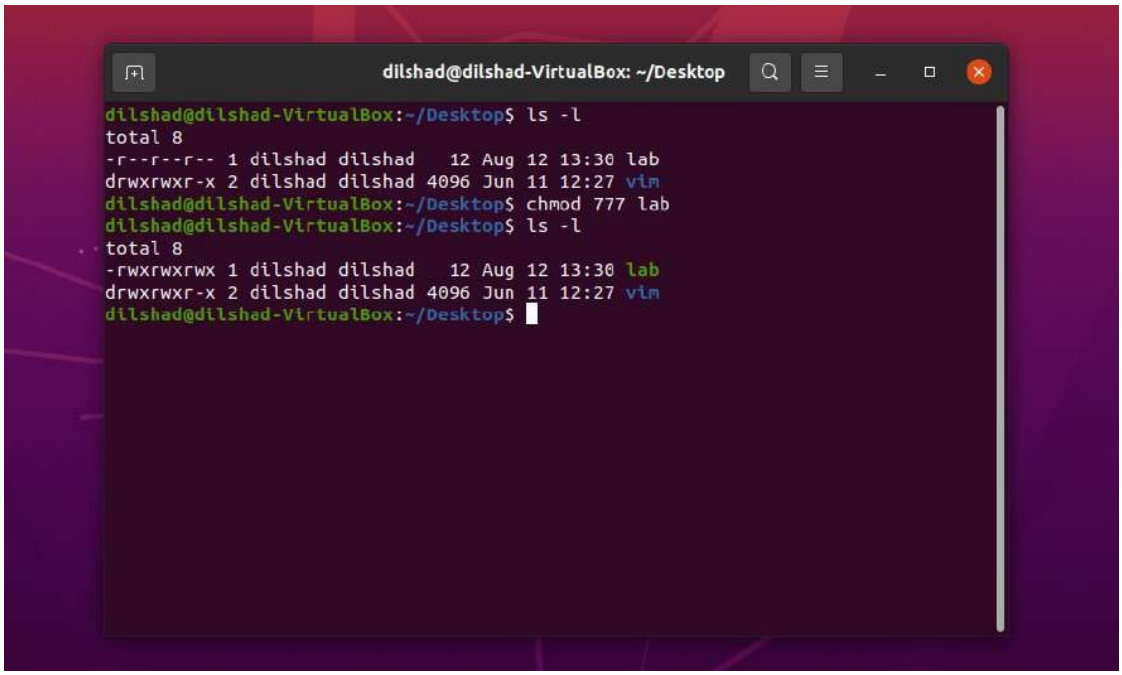
In the example we are adding execute permission to the user.And removing write permission from user.



In this we are giving execute permission to the group and write permission to the others.

Example 2 :-

With the absolute(numeric) mode we are giving all owners the permission to read , write and execute the file



System configuration files in /etc :-

The /etc hierarchy contains configuration files. A "configuration file" is a local file used to control the operation of a program; it must be static and cannot be an executable binary.

It is recommended that files be stored in subdirectories of /etc rather than directly in /etc.

/etc/opt

Host-specific configuration files for add-on application software packages must be installed within the directory /etc/opt/<subdir>, where <subdir> is the name of the subtree in /opt where the static data from that package is stored.

/etc/X11

/etc/X11 is the location for all X11 host-specific configuration. This directory is necessary to allow local control if */usr* is mounted read only.

/etc/sgml

Generic configuration files defining high-level parameters of the SGML systems are installed here. Files with names *.conf indicate generic configuration files. File with names *.cat are the DTD-specific centralized catalogs, containing references to all other catalogs needed to use the given DTD. The super catalog file catalog references all the centralized catalogs.

/etc/xml

Generic configuration files defining high-level parameters of the XML systems are installed here. Files with names *.conf indicate generic configuration files. The super catalog file catalog references all the centralized catalogs.

Log Files :-

Log files are the records that Linux stores for administrators to keep track and monitor important events about the server, kernel, services, and applications running on it.

Linux provides a centralized repository of log files that can be located under the **/var/log** directory.

/var/log/messages

- This log file contains generic system activity logs.
- It is mainly used to store informational and non-critical system messages.

/var/log/auth.log

- All authentication related events in Debian and Ubuntu server are logged here.
- If you're looking for anything involving the user authorization mechanism, you can find it in this log file.

/var/log/secure

- It is mainly used to track the usage of authorization systems.
- It stores all security related messages including authentication failures.
- It also tracks sudo logins, SSH logins and other errors logged by system security services daemon.

/var/log/faillog

This file contains information on failed login attempts.

/var/log/mail.log

All mail server related logs are stored here.

/var/log/mysql.log

- As the name suggests, this is the MySQL log file.
- All debug, failure and success messages related to the [mysqld] and [mysqld_safe] daemon are logged to this file.

SYSTEM'S SPECIFICATIONS

A System Requirement Specification [SRS] is a collection of information that embodies the requirement of a system.

The specifications of a software should match with the SRS to run the application smoothly. For example, a computer game may require your computer to have Windows 7,2.0 Ghz processor,2gb RAM,1gb graphics card and 500 mb storage space. If the system doesnot meet all these requirements the game willnot run verywell or might not run at all.

These are the necessary specifications your computer must have in order to use the software or hardware.

1. Operating system.
2. Processor speed.
3. RAM.
4. Graphics card.
5. Hard disk space.
6. I/O ports.

FACTORS THAT AFFECT COMPUTER PERFORMANCE :-

i. Processor speed and architecture :

- The **architecture** of a processor is the most important factor to determine its performance, and refers to its basic design and complexity.
- The **speed** of a computer's processor chip (technically known as its "**clock speed**") in measured in **gigahertz (GHz)**, with the fastest modern processors currently running at up to **4.7GHz**. However, for most computing tasks, including web browsing, sending e-mails,

word processing and spreadsheet work any processor running at 1GHz or more remains perfectly sufficient.

- **Cache** is a form of very fast memory integrated into the processor chip, and used to store up instructions (work for the processor) so that it has to slow down as little as possible between tasks.
- **Front Side Bus** (FSB) speed is a measure of how fast a microprocessor communicates with the computer's main circuit board (or "motherboard") into which it is physically connected.

ii. Random Access Memory (RAM) :

- The part of the computer in which information is stored temporarily when a program is being used.
- RAM is measured in megabytes (MB) and gigabytes (GB), as detailed on the storage page.

iii. Graphics system :

- Determines how well it can work with visual output. Graphics systems can either be integrated into a computer's motherboard, or plugged into the motherboard as a separate "video card".

iv. Hard Drive Speed and Capacity :

- A part of the computer that is used for storing computer data and that contains one or more hard disks.
- Two key factors that determine the speed of traditional, spinning hard disks :
  1. Rotational velocity of the physical disk itself.
  2. Interface used to connect it to the computer's motherboard.

Three types of interface :

- **Serial Advance Technology Attachment (SATA) :-**
  The most modern and now pretty much the norm on new PCs
- **Integrated Device Electronics (IDE)** (also known as UDMA):-
  A slower and older form of interface
- **Small Computer System Interface (SCSI**):-
  The oldest but in it most modern variant is still the fastest disk interface standard.

SERVER

****

Server is the centrally located and most powerful computer system in the network whose primary functions are houses the server software, stores and manage common data and serve the data to the clients and provide shared services like access to internet, printing, faxing and so on. Server may serve data to the systems on the LAN or WAN network that depend from the where the request is come. Server is a device or computer which manage network resources. If we talk about server is software terms then it is a computer program that serve as the server component in a client-server architecture and provide some of the features like centralization, Does nothing until a request is received from client, Some of the continuous operations that are running always and ready to respond to client request, background operation that interact with the user program not to the user directly, simultaneous operation that respond to multiple user at the same time.

TYPES OF SERVERs :-

1. **Proxy server** :-

   Proxy server is the software system running on a computer that act like and intermediate between client endpoint device and other servers from which user is requesting for his request. A proxy server can be a different server/computer or may be a same machine using firewall which forward request through firewall. Some of the advantages of using proxies servers are: this server's cache can serve all users, because most of the time user request for more than one internet sites so that should probably be handle by proxy server. This can also improve user response time when main server is already busy. Proxy server log its interactions with the user this would be helpful for troubleshooting.

2. **Mail Server :-**

   As the name implies mail server act like a virtual post office and store all the mail that comes from local users and sends it out to the mentioned user's mail. This server is based on client-server application model and use Simple mail transfer protocol(SMTP) to send and receive mail. We can say it with the another name which is mail transfer agent.

3. **Web Server :-**

   A server that deliver content and service over the internet that is web server. It is also known by Internet server. This server is emphasis of a physical server, Server Operating system, and software used to establish http communication. Primary function of web server is to respond HTTP request to deliver the requested content and service by the client.

4. **Application Server :-**

   Application server is a server program in a computer which based on distributed network and providing the business logic to an application program. This server is viewed as a part of three-tier application emphasis

of GUI, application server and a database server which is also known as transaction server. Application server provide support to web services and expose business logic services. These servers are heavy in terms of resource utilization.

5. **FTP Server :-**

   FTP server is one of the oldest internet service, By using FTP protocol it is possible to transfer file over internet securely between computers. FTP server emphasis of provide file security, organization of the files and the transfer control.

6. **Telnet Server :-**

   This server enable the user to log on to the host server simultaneously when he is already working on remote server.

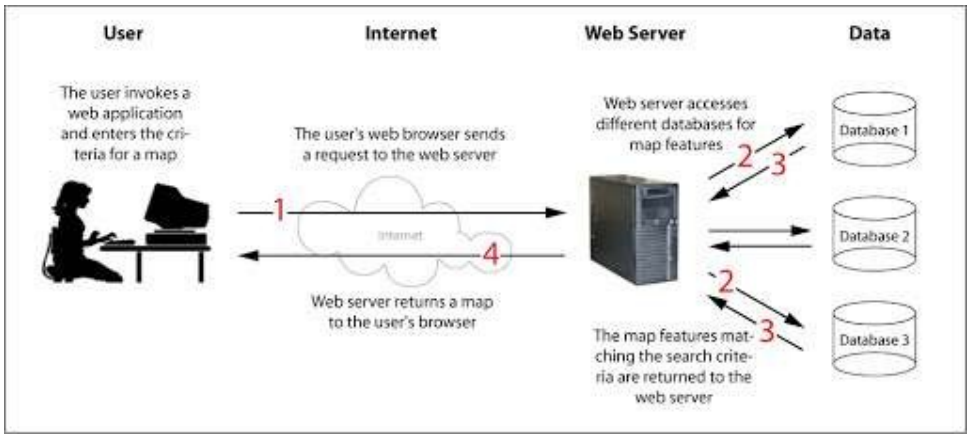
WORKING OF A SERVER :

At the most basic level, a server is a repository for web pages that respond when someone requests a certain website. This '**request**' is simply the act of entering the web address into a browser and hitting return. The server monitors these requests via ports, giving their '**response**' nearly instantly to deliver the site page requested. Once the server has received and verified the request at-hand, it gathers the assorted elements that make up a website and communicates this assembled information back to the user's web browser. At its core, request-response is the key to what work a server does, day in and day out.All that remains then is for the web browser to ensure the requested site is genuine and display the page for the user. Web browsers and servers ensure the request is genuine by using TCP (Transmission Control Protocol) or IP (Internet Protocol), with HTTP

overlaid to ensure seamless communication between the server and web browser in use. At the same time, web browsers use DNS (Domain Name System) to make it possible for different types of web browser to connect to a diverse range of server types and configurations by changing requests for domain names (like **knownhost.com**) into numeric addresses, and back again – a bit like a post office uses zip codes.

It works like this:

- The web browser requests a specific web page – looking for the correct IP address associated with that domain.
- The web browser requests a full URL for the site it wants to display – sending this information over to the server.
- The web server finds and assembles all the information needed to display the site – including things like ads, dynamic elements, content and more.
- The server then sends this complete package of information back to the web browser as a response.
- The web browser receives this complete page and displays it for the user.



Shell scripting

The basic concept of a shell script is a list of commands, which are listed in the order of execution. A good shell script will have comments, preceded by # sign, describing the steps.

There are conditional tests, such as value A is greater than value B, loops allowing us to go through massive amounts of data, files to read and store data, and variables to read and store data, and the script may include functions.

We are going to write many scripts in the next sections. It would be a simple text file in which we would put all our commands and several other required constructs that tell the shell environment what to do and when to do it.

A **Shell** provides you with an interface to the Unix system. It gathers input from you and executes programs based on that input. When a program finishes executing, it displays that program's output.

Shell is an environment in which we can run our commands, programs, and shell scripts. There are different flavors of a shell, just as there are different flavors of operating systems. Each flavor of shell has its own set of recognized commands and functions.

Shell Types :-

In Unix, 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 Bourne Shell has the following subcategories −

- Bourne shell (sh)
- Korn shell (ksh)
- Bourne Again shell (bash)
- POSIX shell (sh)

Assume we create a **test.sh** script. Note all the scripts would have the **.sh** extension. Before you add anything else to your script, you need to alert the system that a shell script is being started. This is done using the **shebang** construct

#!/bin/sh

Example :-

```
#!/bin/sh

echo "What is your name?"
read PERSON
echo "Hello, $PERSON"
```

Here is a sample run of the script −

```
$./test.sh
What is your name?
xyz
Hello, xyz
$
```

Variables :-

The name of a variable can contain only letters (a to z or A to Z), numbers (0 to 9) or the underscore character (_).

By convention, Unix shell variables will have their names in UPPERCASE.

The following examples are valid variable names −

_ALI
TOKEN_A
VAR_1
VAR_2

Following are the examples of invalid variable names −

2_VAR
-VARIABLE
VAR1-VAR2
VAR_A!

The reason you cannot use other characters such as **!**, *****, or **-** is that these characters have a special meaning for the shell.

Variable Types :-

When a shell is running, three main types of variables are present −

- **Local Variables** − A local variable is a variable that is present within the current instance of the shell. It is not available to programs that are started by the shell. They are set at the command prompt.
- **Environment Variables** − An environment variable is available to any child process of the shell. Some programs need environment variables in order to function correctly. Usually, a shell script defines only those environment variables that are needed by the programs that it runs.

- **Shell Variables** − A shell variable is a special variable that is set by the shell and is required by the shell in order to function correctly. Some of these variables are environment variables whereas others are local variables.

Defining Variables :-

Variables are defined as follows −

variable_name=variable_value

For example −

```
NAME="xyz"
```

Accessing Values :-

To access the value stored in a variable, prefix its name with the dollar sign (**$**).

For example, the following script will access the value of defined variable NAME and print it

```
#!/bin/sh

NAME="xyz"
echo $NAME
```

control constructs :-

The if...else statements

If else statements are useful decision-making statements which can be used to select an option from a given set of options.

Unix Shell supports following forms of **if…else** statement −

- if...fi statement
- if...else...fi statement
- if...elif...else...fi statement

syntax :

```
if [ expression1 ]
then
   statement1
   statement2
   .
   .
elif [ expression2 ]
then
   statement3
   statement4
   .
   .
else
   statement5
fi
```

The case...esac Statement

You can use multiple **if...elif** statements to perform a multiway branch. However, this is not always the best solution, especially when all of the branches depend on the value of a single variable.

Unix Shell supports **case...esac** statement which handles exactly this situation, and it does so more efficiently than repeated **if...elif** statements.

There is only one form of **case...esac** statement which has been described in detail here −

- case...esac statement

The **case...esac** statement in the Unix shell is very similar to the **switch...case** statement we have in other programming languages like **C** or **C++**

Syntax :

```
case in

Pattern 1) Statement 1;;

Pattern n) Statement n;;

esac
```

while statement

Here command is evaluated and based on the result loop will executed, if command raise to false then loop will be terminated.

Syntax

```
while command
do
   Statement to be executed
done
```

for statement

The for loop operate on lists of items. It repeats a set of commands for every item in a list.

Here var is the name of a variable and word1 to wordN are sequences of characters separated by spaces (words). Each time the for loop executes, the value of the variable var is set to the next word in the list of words, word1 to wordN.

Syntax

```
for var in word1 word2 ...wordn
do
   Statement to be executed
done
```

functions

Functions enable you to break down the overall functionality of a script into smaller, logical subsections, which can then be called upon to perform their individual tasks when needed.

Using functions to perform repetitive tasks is an excellent way to create **code reuse**. This is an important part of modern object-oriented programming principles.

Shell functions are similar to subroutines, procedures, and functions in other programming languages.

Creating Functions

To declare a function, simply use the following syntax −

```
function_name () {
   list of commands
}
```

The name of your function is **function_name**, and that's what you will use to call it from elsewhere in your scripts. The function name must be followed by parentheses, followed by a list of commands enclosed within braces.

Example

Following example shows the use of function −

```
#!/bin/sh

# Define your function here
Hello () {
   echo "Hello World"
}

# Invoke your function
Hello
```

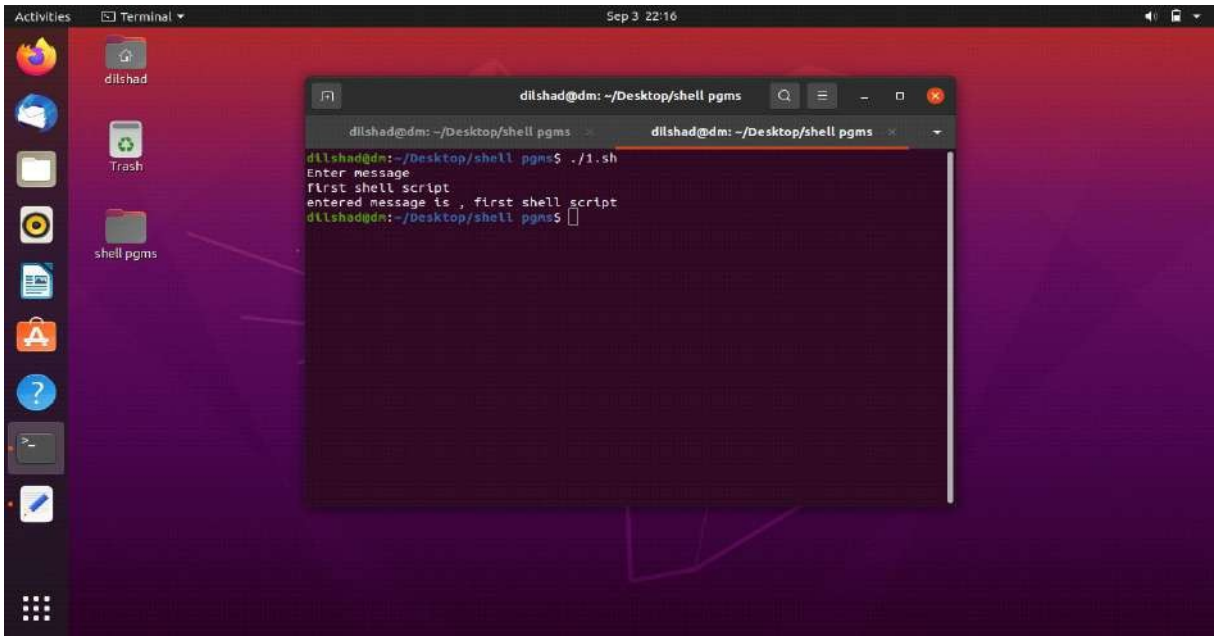
Upon execution, you will receive the following output −

```
$./test.sh
Hello World
```

Shell 15 programs :-

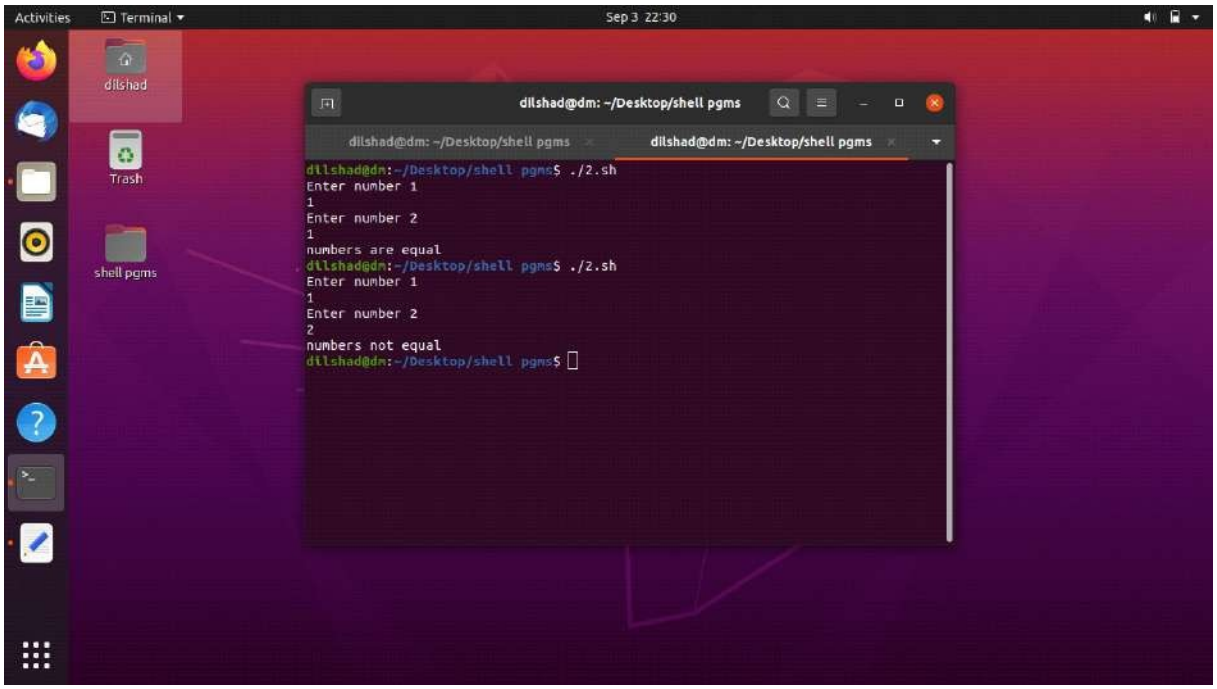
1) Write a shell script program to display a given message.

```
#!/bin/sh
echo "Enter message"
read msg
echo "entered message is , $msg"
```



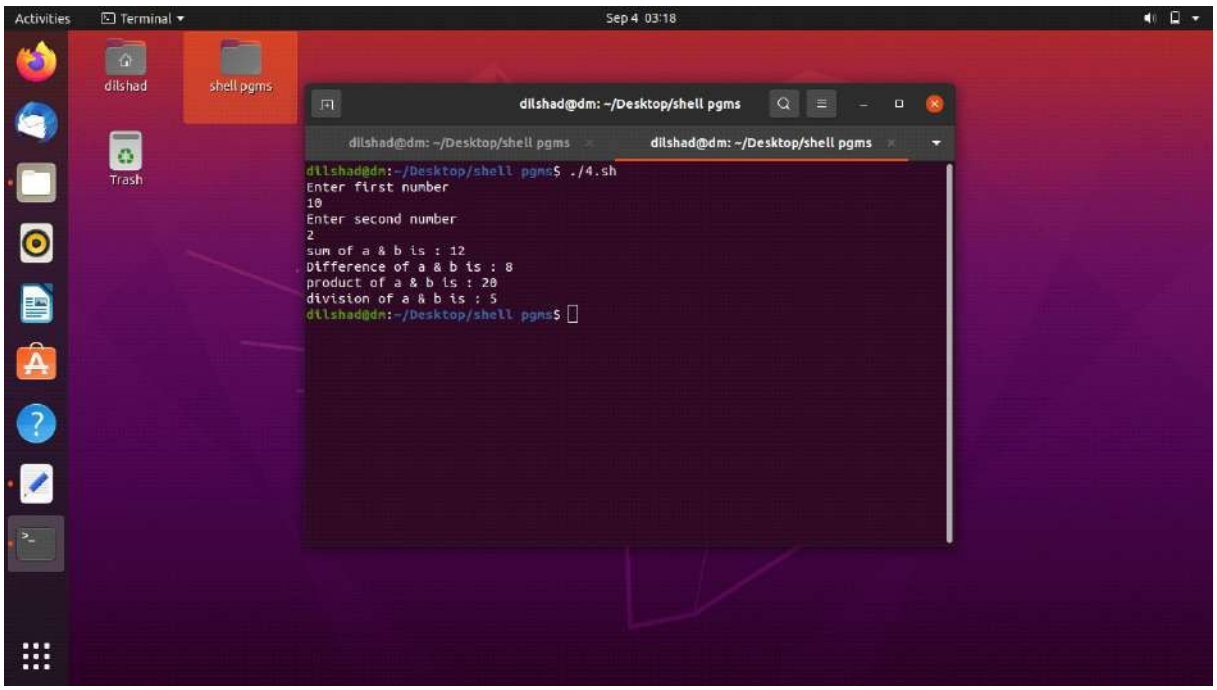
2) Write a shell script to print whether two numbers are equal or not.

```
#!/bin/sh
echo "Enter number 1"
read a
echo "Enter number 2"
read b
if [ $a -eq $b ]
then
echo "numbers are equal"
else
echo "numbers not equal"
fi
```



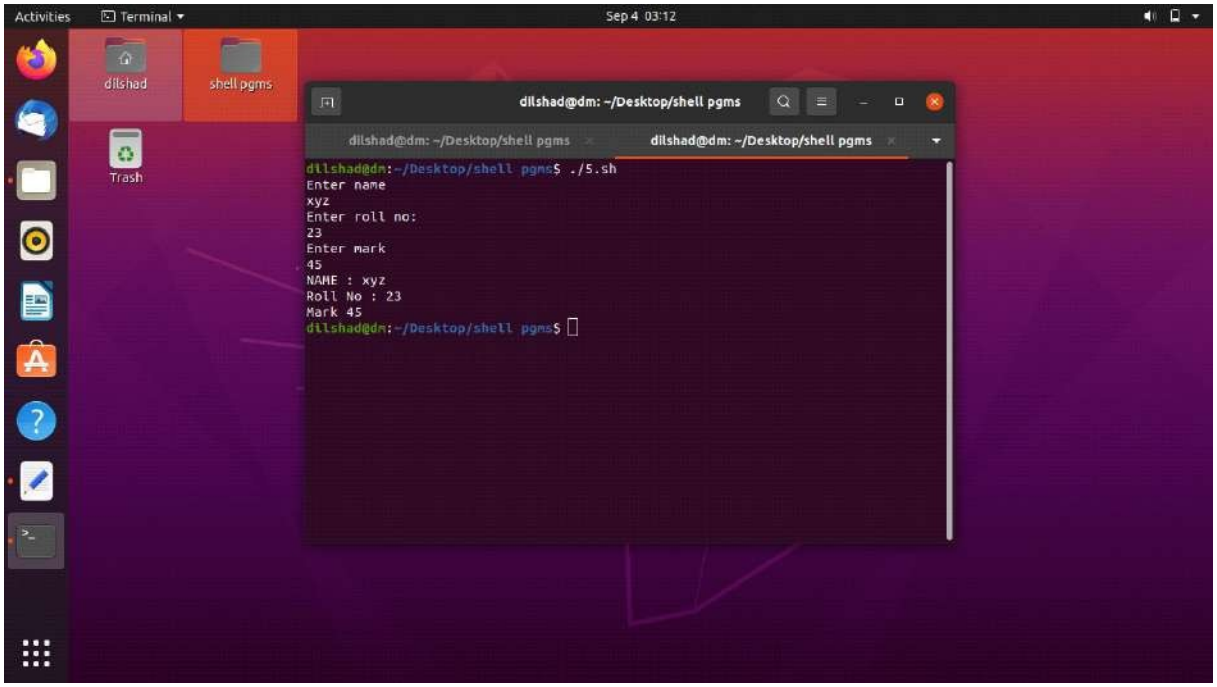
4) Write a shell script to perform integer arithmetic operations.

```
#!/bin/sh
echo "Enter first number"
read a
echo "Enter second number"
read b
add=$(( $a + $b ))
sub=$(( $a - $b ))
mul=$(( $a * $b ))
div=$(( $a / $b ))
echo "sum of a & b is : $add"
echo "Difference of a & b is : $sub"
echo "product of a & b is : $mul"
echo "division of a & b is : $div"
```



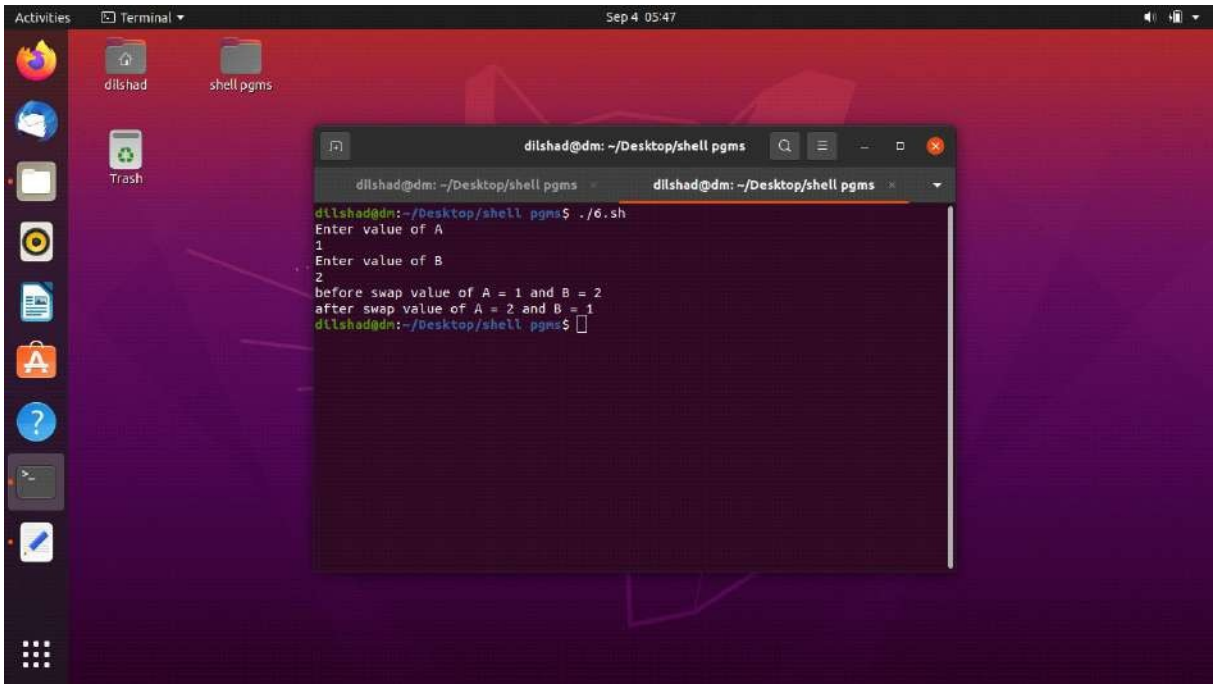
5) Write a shell script to getting input details like name, roll number and marks and print them.

```
#!/bin/sh
echo "Enter name"
read name
echo "Enter roll no:"
read no
echo "Enter mark"
read mark
echo "NAME : $name"
echo "Roll No : $no"
echo "Mark $mark "
```



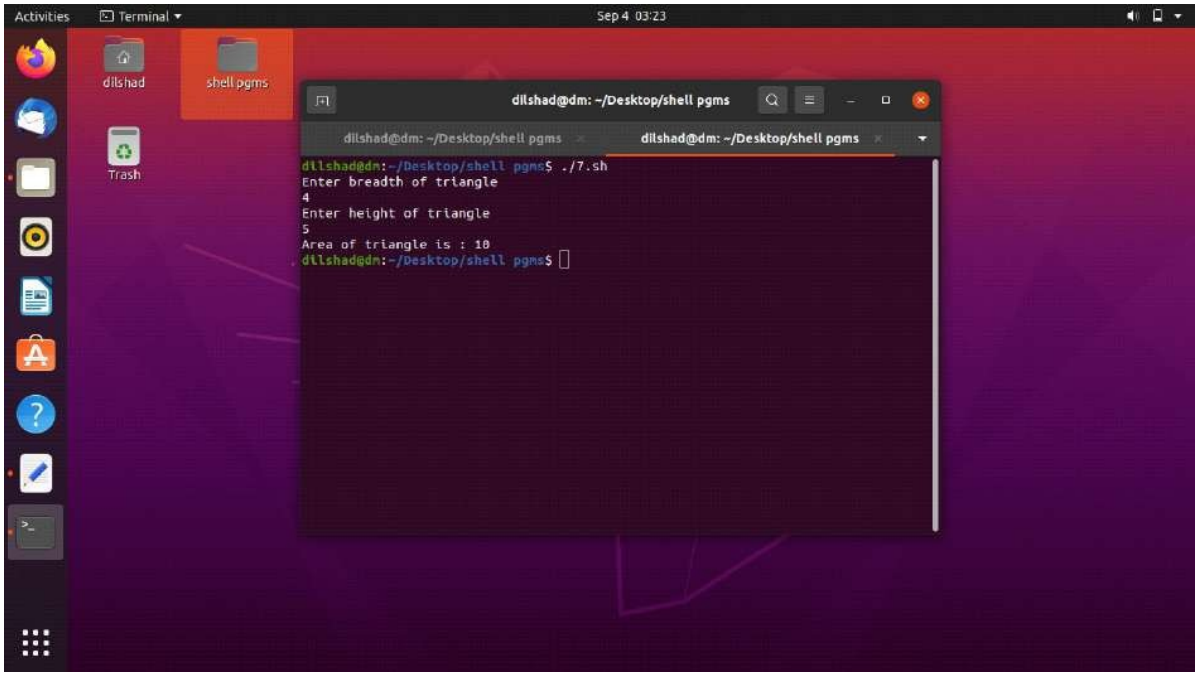
6) Write a Shell program to swap two values.

```
#!/bin/sh
echo "Enter value of A"
read a
echo "Enter value of B"
read b
echo "before swap value of A = $a and B = $b"
a=$(($a+$b))
b=$(($a-$b))
a=$(($a-$b))
echo "after swap value of A = $a and B = $b"
```



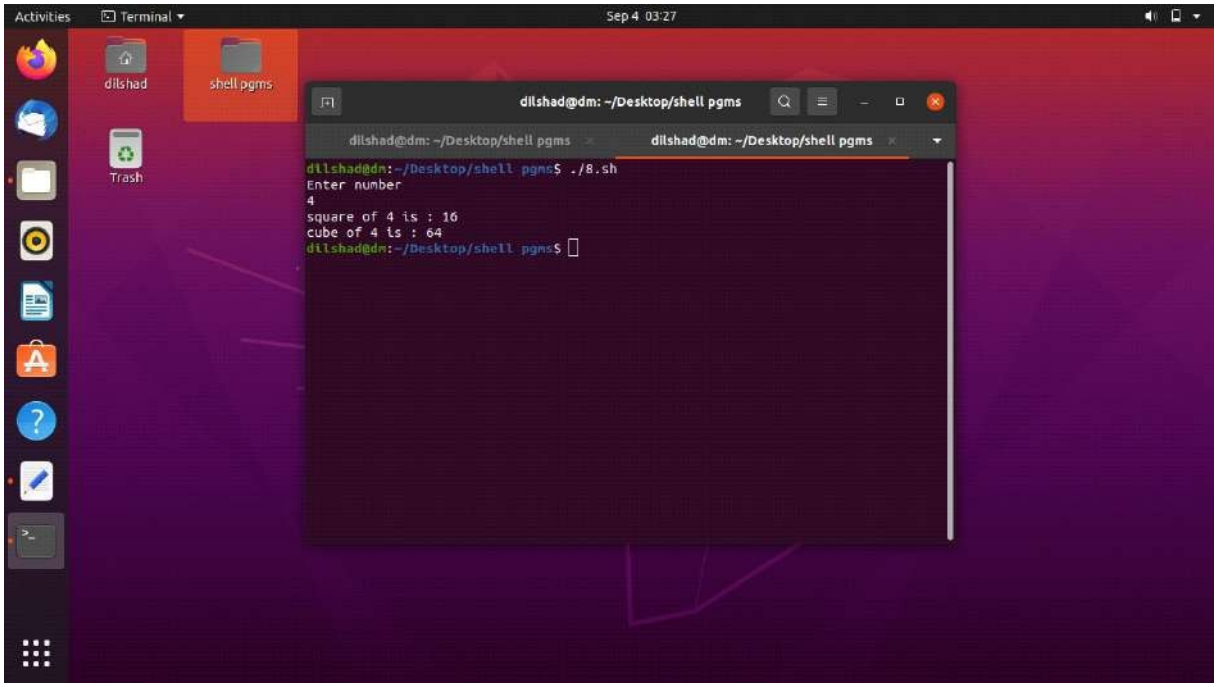
7) Write a shell program to find the area of a triangle.

```
#!/bin/sh
echo "Enter breadth of triangle"
read b
echo "Enter height of triangle"
read h
ar=$(($b * $h))
area=$(($ar/2))
echo "Area of triangle is : $area"
```



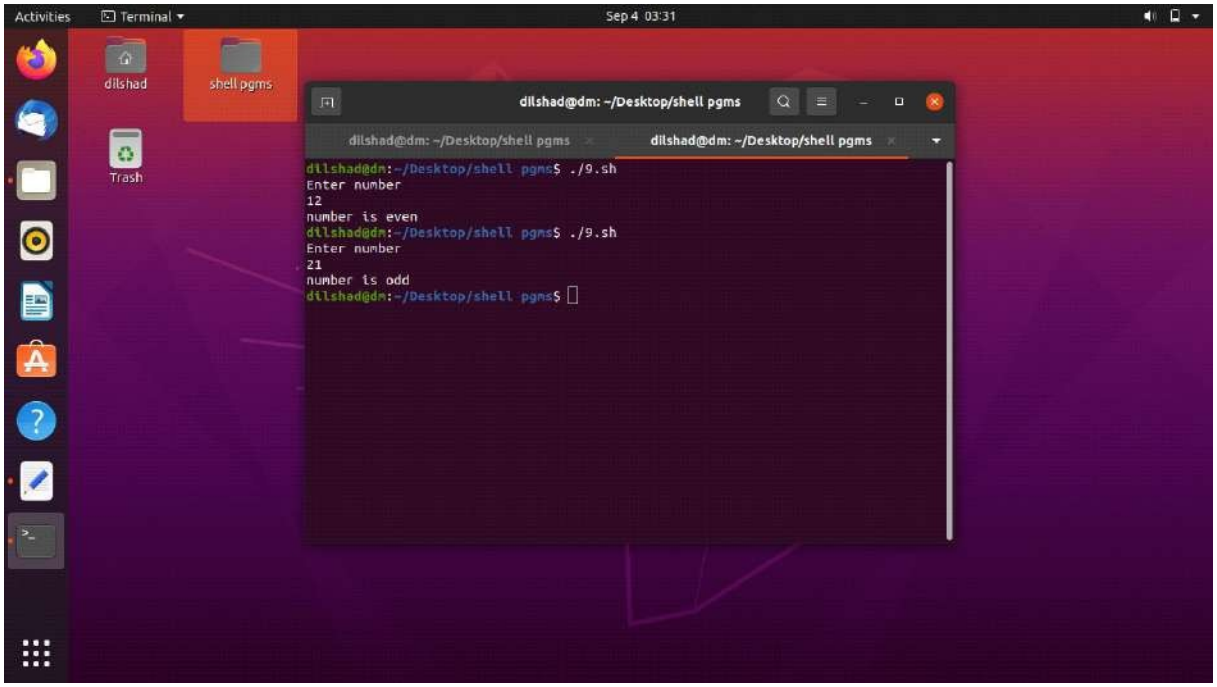
8) Write a shell program to find the square and cube of a number.

```
#!/bin/sh
echo "Enter number"
read  n
sq=$(($n*$n))
cu=$(($n*$n*$n))
echo "square of $n is : $sq"
echo "cube of $n is : $cu"
```



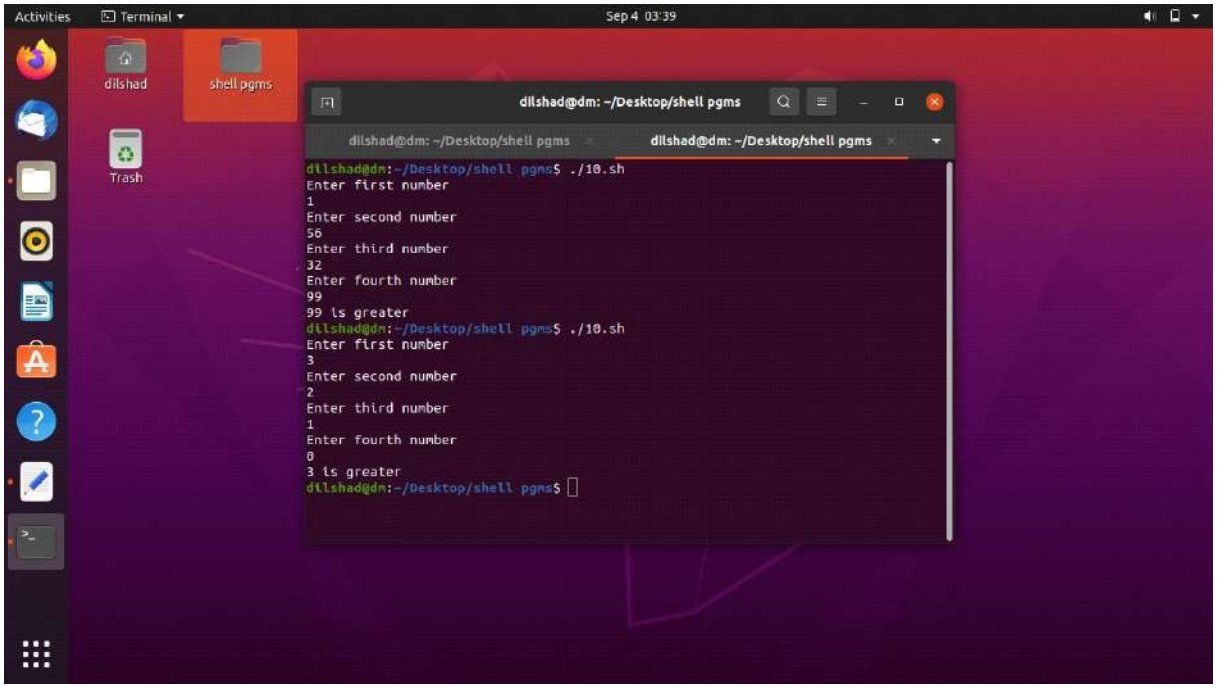
9) Write a shell program to check whether the given number is odd or even.

```
#!/bin/sh
echo "Enter number"
read n
rem=$(($n%2))
if [ $rem -eq 0 ]
then
echo "number is even "
else
echo "number is odd"
fi
```



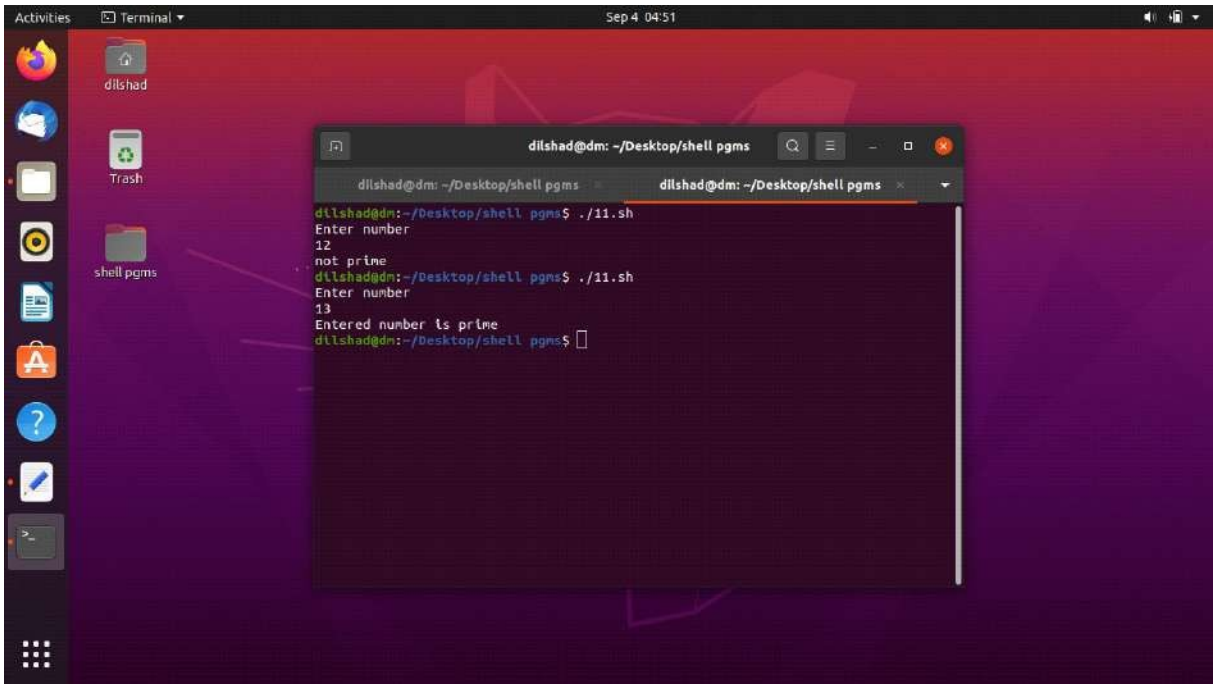
10) Write a shell program to find the minimum among four values.

```
#!/bin/sh
echo "Enter first number"
read a
echo "Enter second number"
read b
echo "Enter third number"
read c
echo "Enter fourth number"
read d
if [ $a -gt $b    ] && [ $a -gt $c ] && [ $a -gt $d ]
then
echo "$a is greater"
elif [ $b -gt $a    ] && [ $b -gt $c ] && [ $b -gt $d ]
then
echo "$b is greater"
elif [ $c -gt $a    ] && [ $c -gt $b ] && [ $c -gt $d ]
then
echo "$c is greater"
else
echo "$d is greater"
fi
```



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

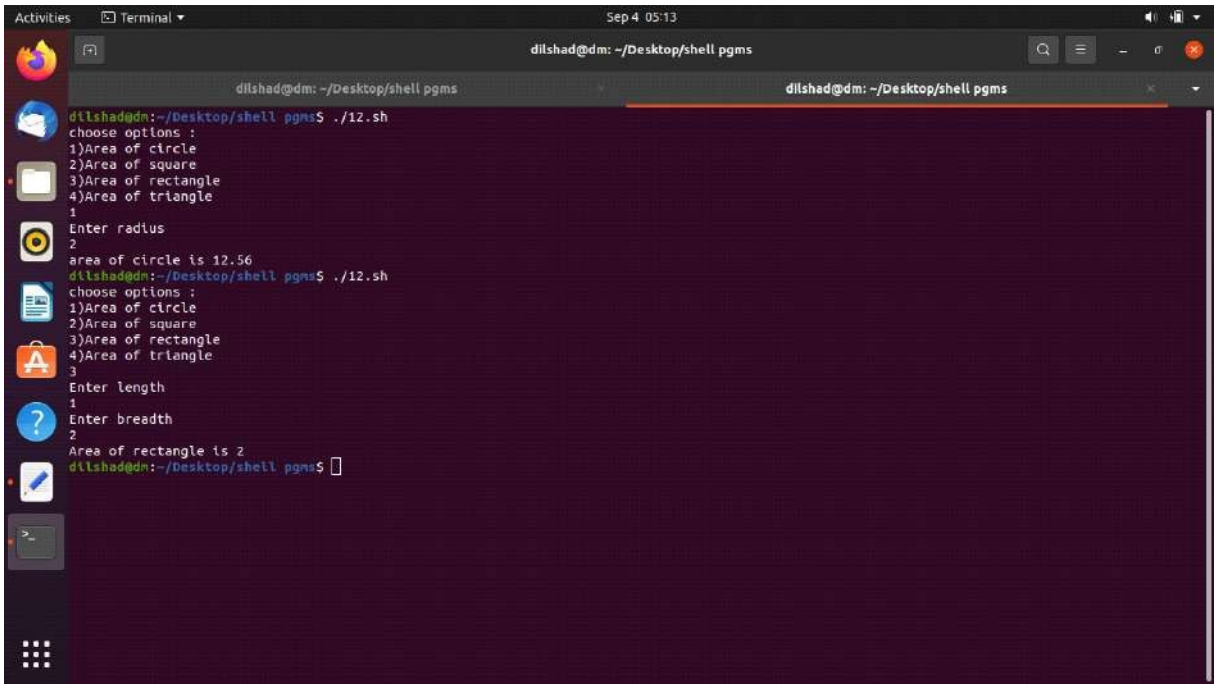
```
#!/bin/sh
echo "Enter number"
read n
f=0
for i in 2 $((n/2))
do
[ $((n%i)) -eq 0 ] && f=1
done
[ $f -eq 0 ] && echo "Entered number is prime" || echo "not prime"
```



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

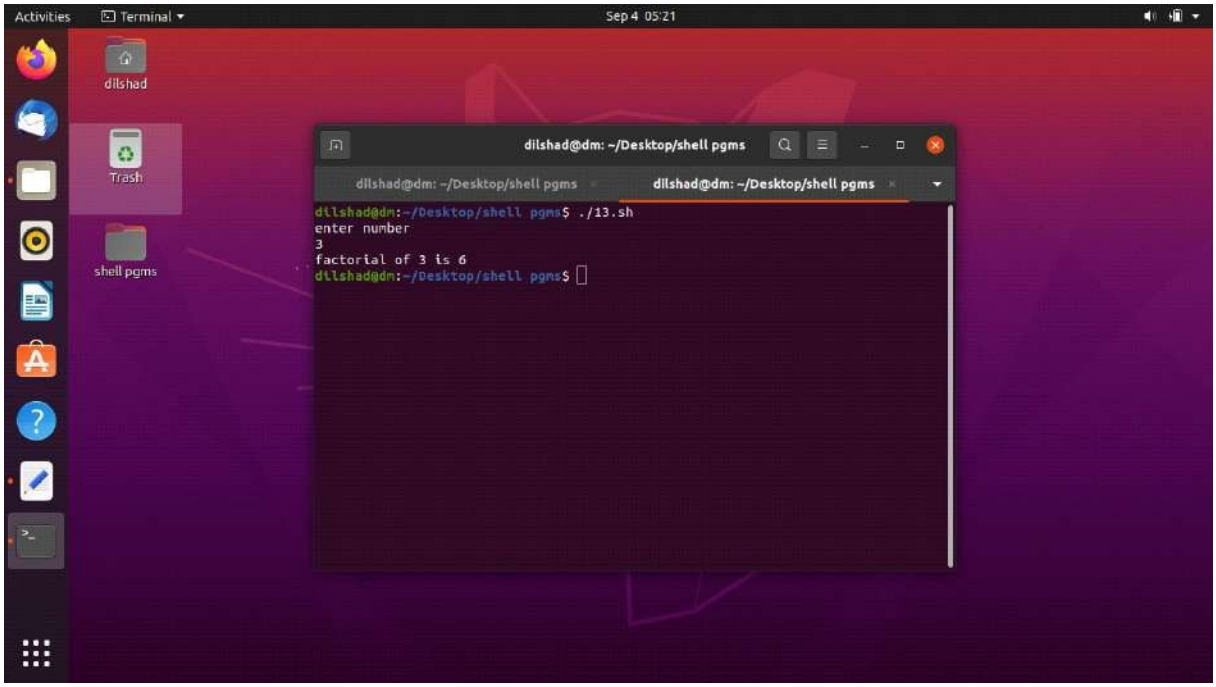
```
#!/bin/sh
echo "choose options :"
echo "1)Area of circle"
echo "2)Area of square"
echo "3)Area of rectangle"
echo "4)Area of triangle"
read ch
case $ch in
        1)echo "Enter radius"
        read r
        area=$(echo "3.14 * $r *$r" |bc)
        echo "area of circle is $area"
        break
        ;;
        2)echo "Enter side"
        read l
        area=$(($l*$l))
        echo "area of square is $area"
        break
        ;;
        3)echo "Enter length"
        read l
        echo "Enter breadth"
        read b
        area=$(($l*$b))
        echo "Area of rectangle is $area"
        break
```

```
        ;;
        4)echo "Enter breadth"
        read b
        echo "Enter height"
        read h
        ar=$(($b*$h))
        area=$(($ar/2))
        echo "Area of triangle is : $area"
        break
        ;;
        *)echo "invalid option ! ! !"
        ;;
esac
```



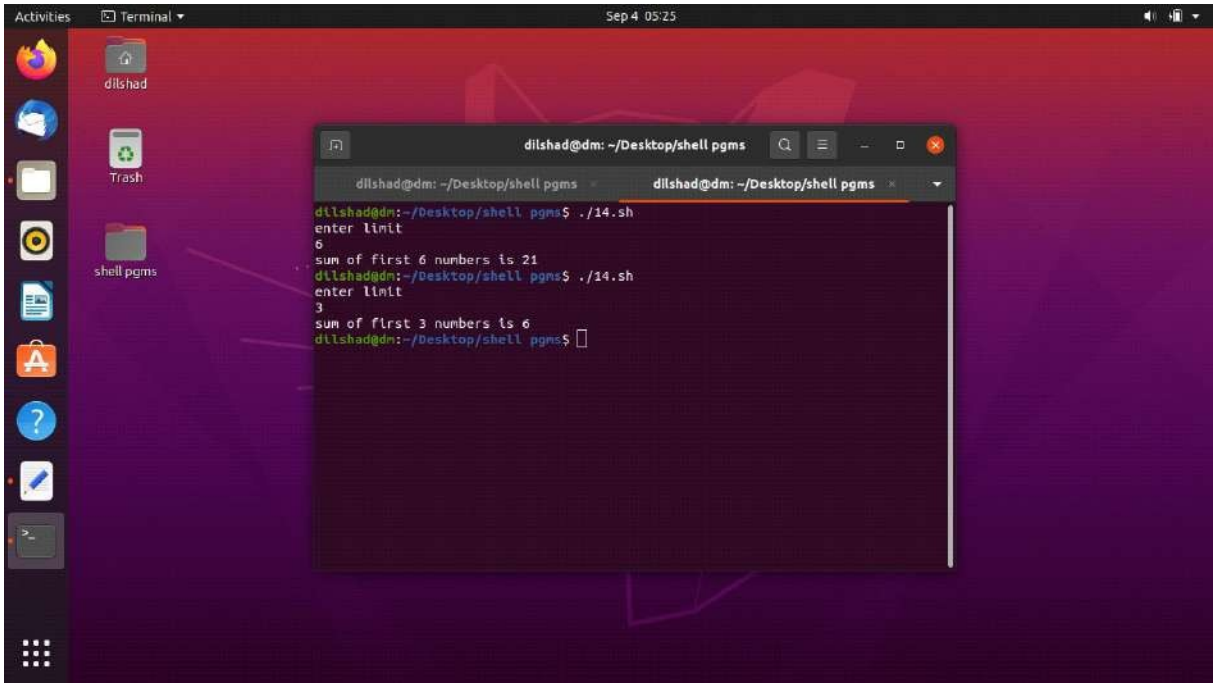
13) Write a shell program to find the factorial of a given number.

```
#!/bin/sh
echo "enter number"
read n
fact=1
i=1
while [ $i -le $n ]
do
fact=$(($fact*$i))
i=$(($i+1))
done
echo "factorial of $n is $fact"
```



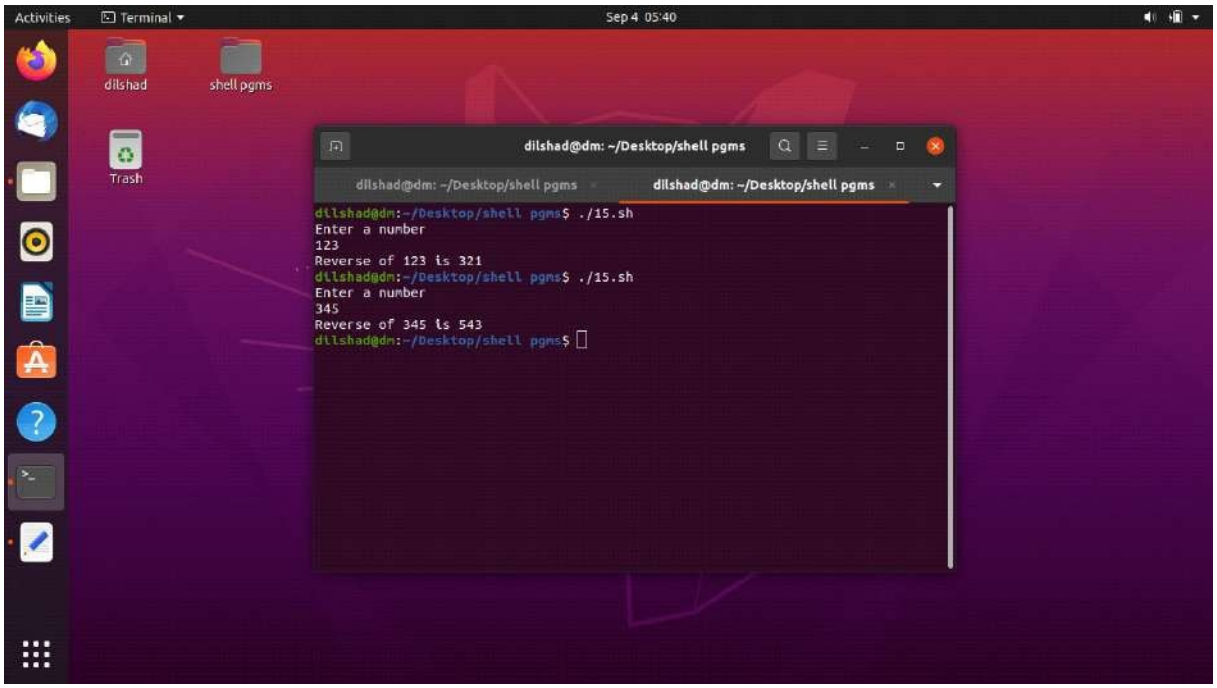
14) Write a Simple Shell script to print the sum of n natural numbers.

```
#!/bin/sh
echo "enter limit"
read n
sum=0
i=1
while [ $i -le $n ]
do
sum=$(($sum+$i))
i=$(($i+1))
done
echo "sum of first $n numbers is $sum"
```



15) Write a shell program to reverse a number.

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

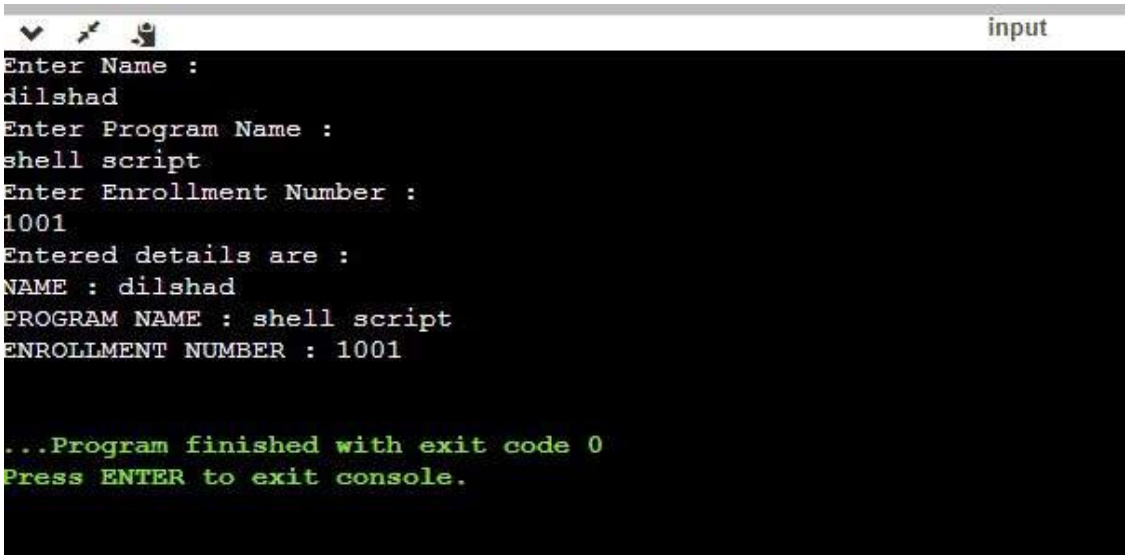


1.**Write a shell script to ask your name, program name and enrollment number and print it on the screen.**

Source code

```
echo "Enter Name :"

read name

echo "Enter Program Name :"

read pname

echo "Enter Enrollment Number :"

read eno

echo "Entered details are :"

echo "NAME : $name"

echo "PROGRAM NAME : $pname"

echo "ENROLLMENT NUMBER : $eno"
```

output

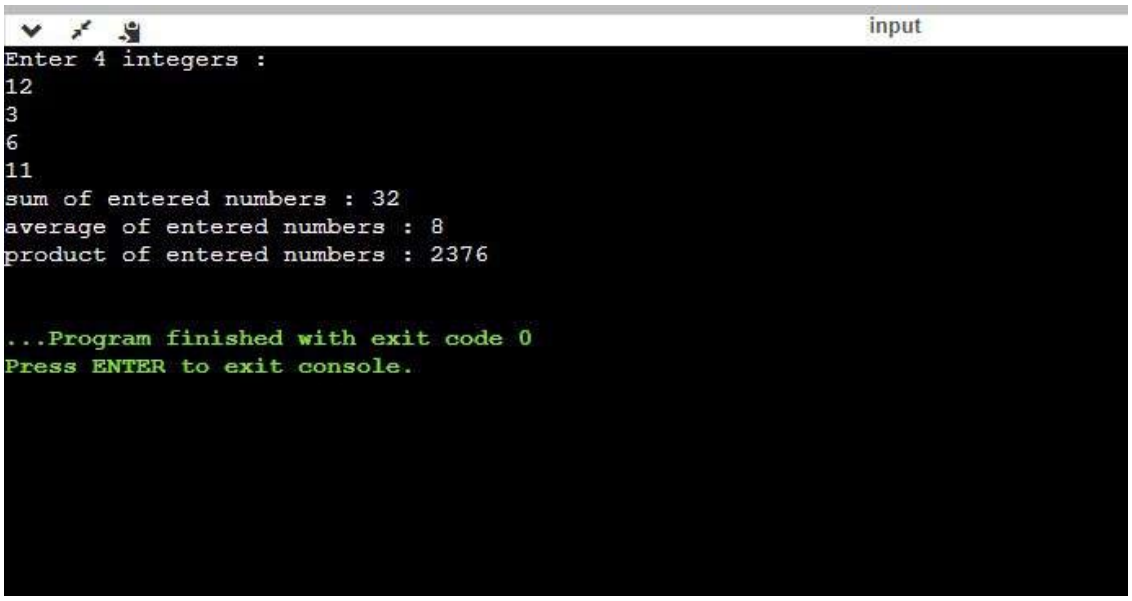


2) Write a shell script to find the sum, the average and the product of the four integers entered.

Source code

```
echo "Enter 4 integers :"
read n1
read n2
read n3
read n4
sum=$((n1+n2+n3+n4))
avg=$(($sum/4))
product=$((n1*n2*n3*n4))
echo "sum of entered numbers : $sum"
echo "average of entered numbers : $avg"
echo "product of entered numbers : $product"
```

output

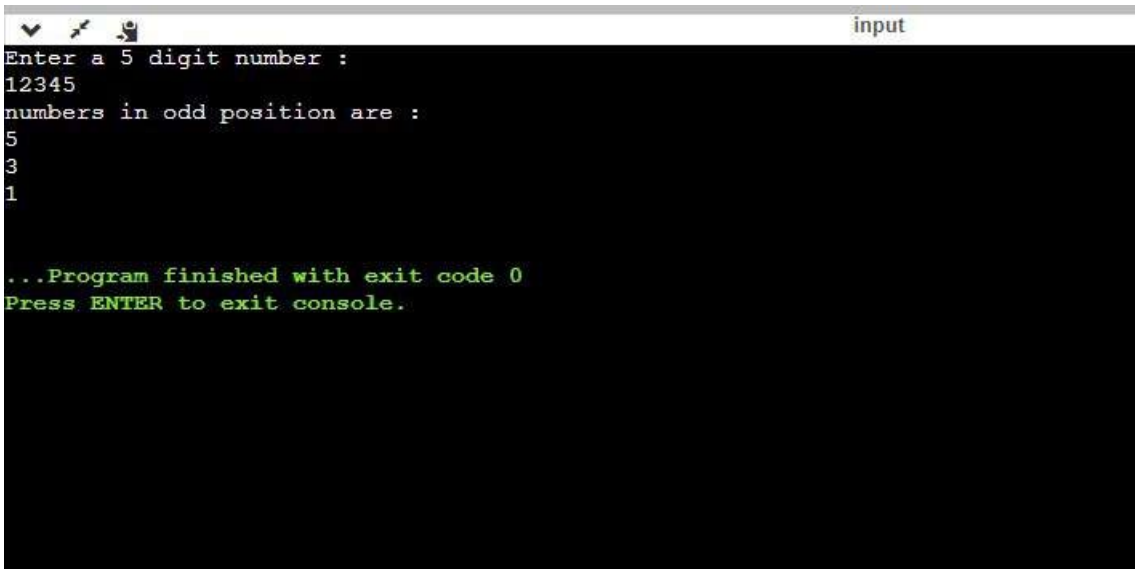


4) Write a shell script to display the digits which are in odd position in a given 5 digit number.

Source code

```
echo "Enter a 5 digit number :"

read num

c=0

rem=0

echo "numbers in odd position are :"

while [ $num -gt 0 ]

do

d=$(($num%10))

num=$(($num/10))

c=$(($c+1))

rem=$(($c%2))

if [ $rem != 0 ]

then

echo "$d"

fi

done
```

output



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

Source code

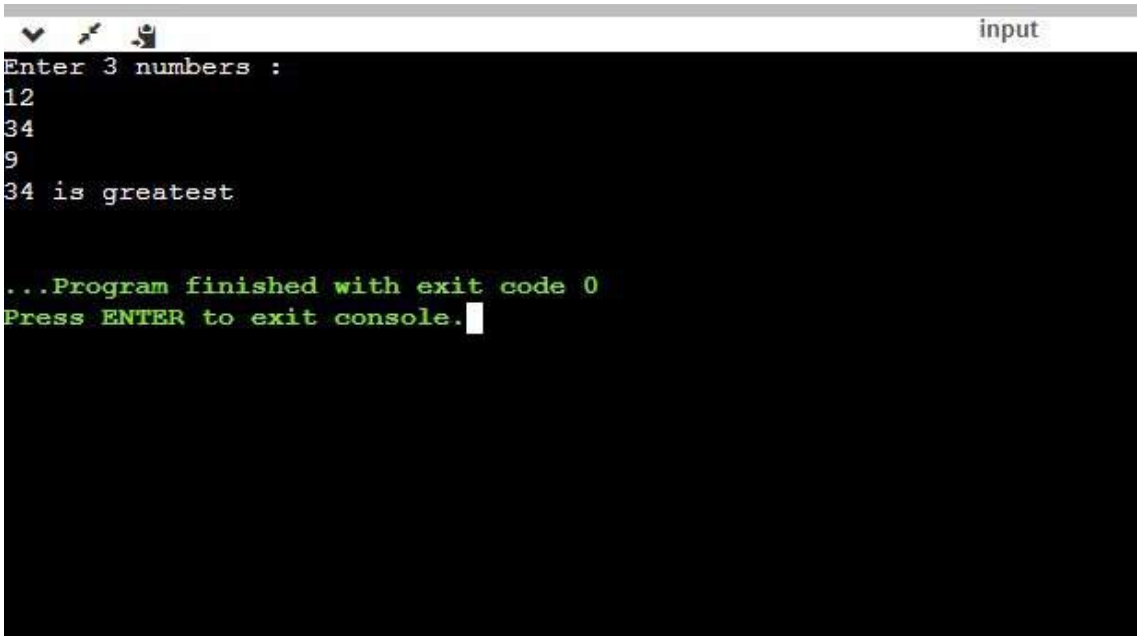
```
echo "Enter 3 numbers :"

read n1

read n2

read n3

if [ $n1 -gt $n2 ] && [ $n1 -gt $n3 ]

then

echo "$n1 is greatest"

elif [ $n2 -gt $n1 ] && [ $n2 -gt $n3 ]

then

echo "$n2 is greatest"
```

else

echo "$n3 is greatest"

fi

output



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

Source code

echo "Enter 2 strings :"

read s1

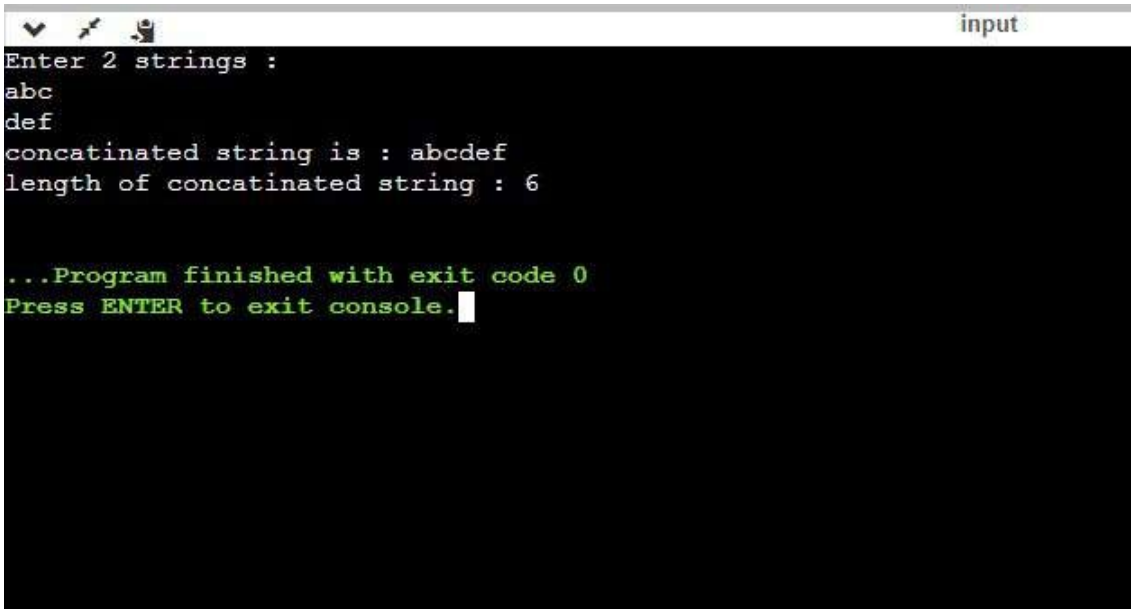
read s2

s1+=$s2

echo "concatinated string is : $s1"

echo "length of concatinated string : ${#s1}"

output



8) **Write a shell script to find the smallest of three numbers.**

Source code

```
echo "Enter 3 numbers :"

read n1

read n2

read n3

if [ $n1 -lt $n2 ] && [ $n1 -lt $n3 ]

then

echo "$n1 is smallest"

elif [ $n2 -lt $n1 ] && [ $n2 -lt $n3 ]
```

then

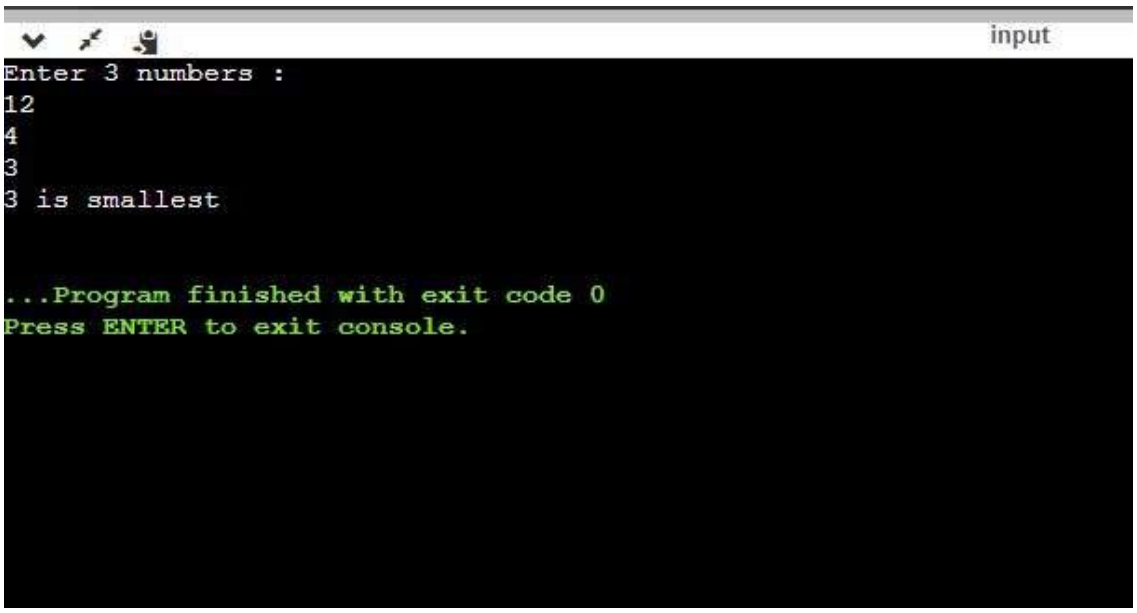
echo "$n2 is smallest"

else

echo "$n3 is smallest"

fi

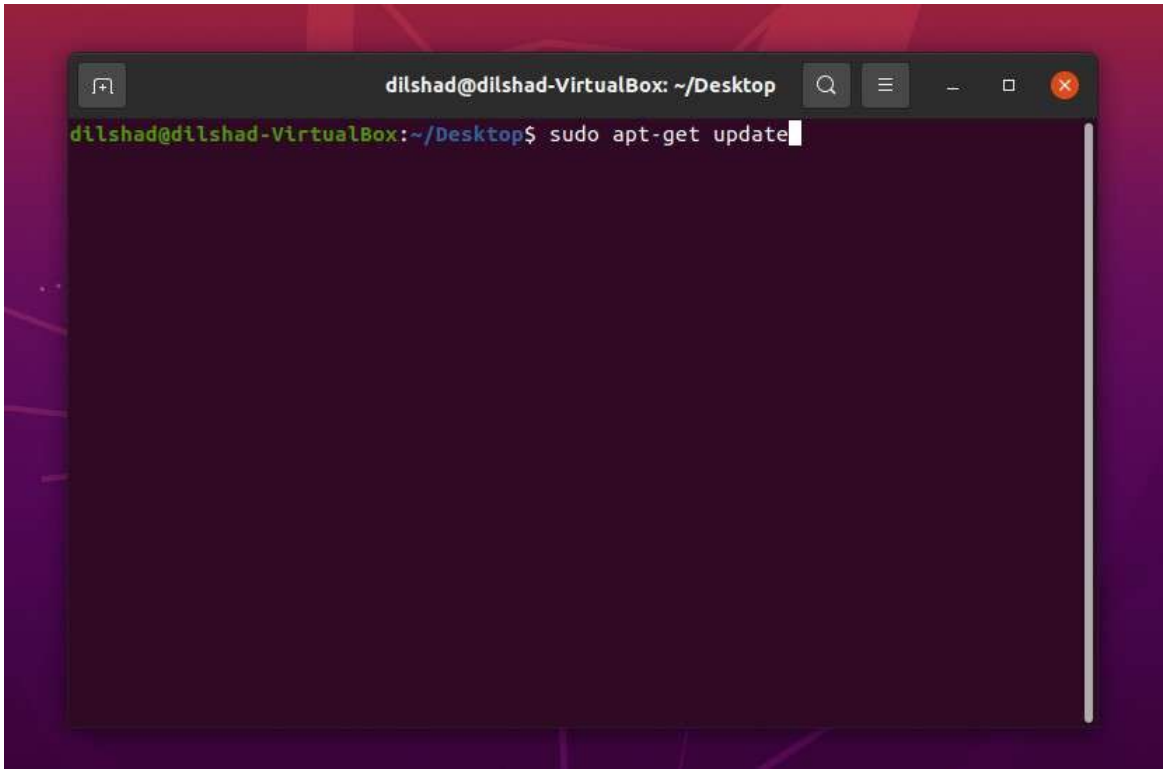
output



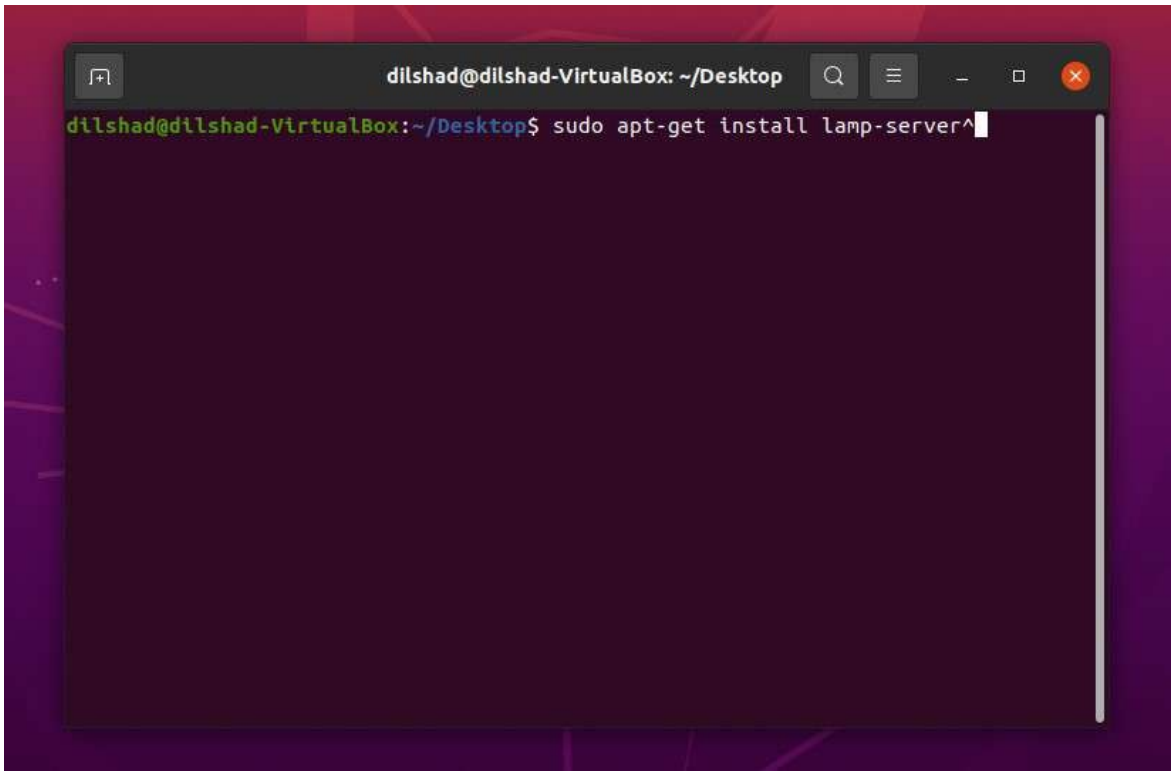
EXPERIMENT -5

Aim : Installation and configuration of LAMP stack. Deploy an open source application such as phpmyadmin and Wordpress.

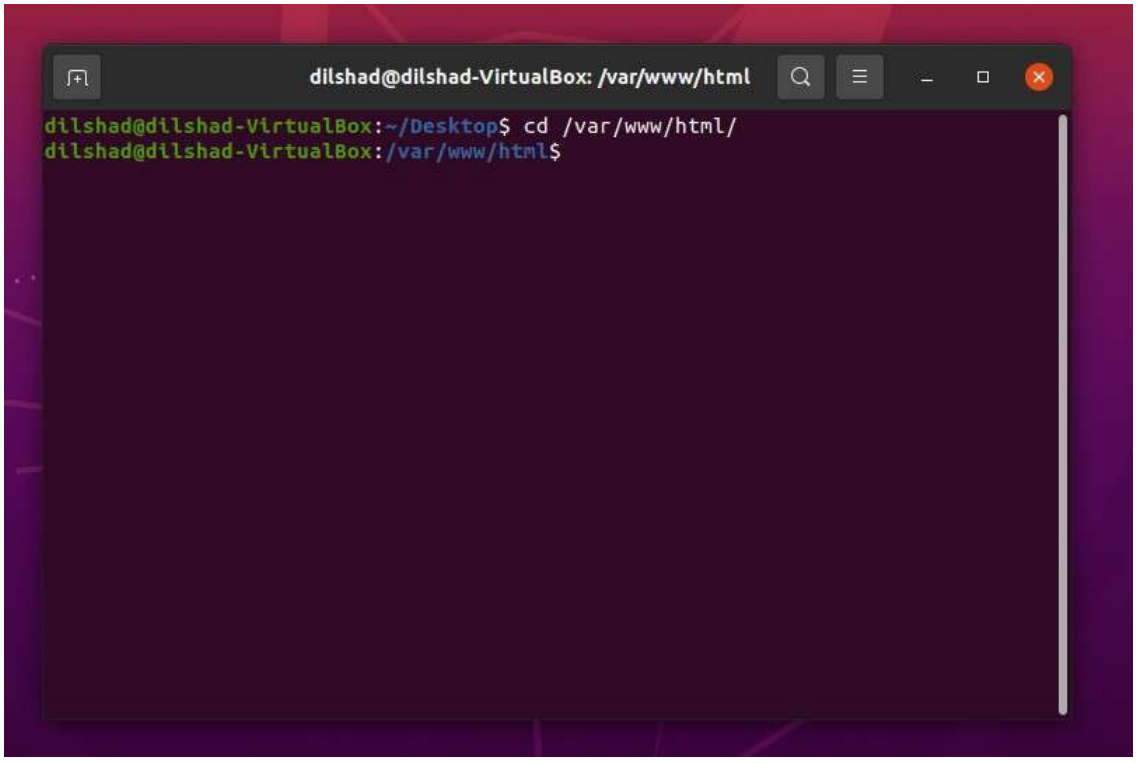
Step 1 :- get the latest updation.



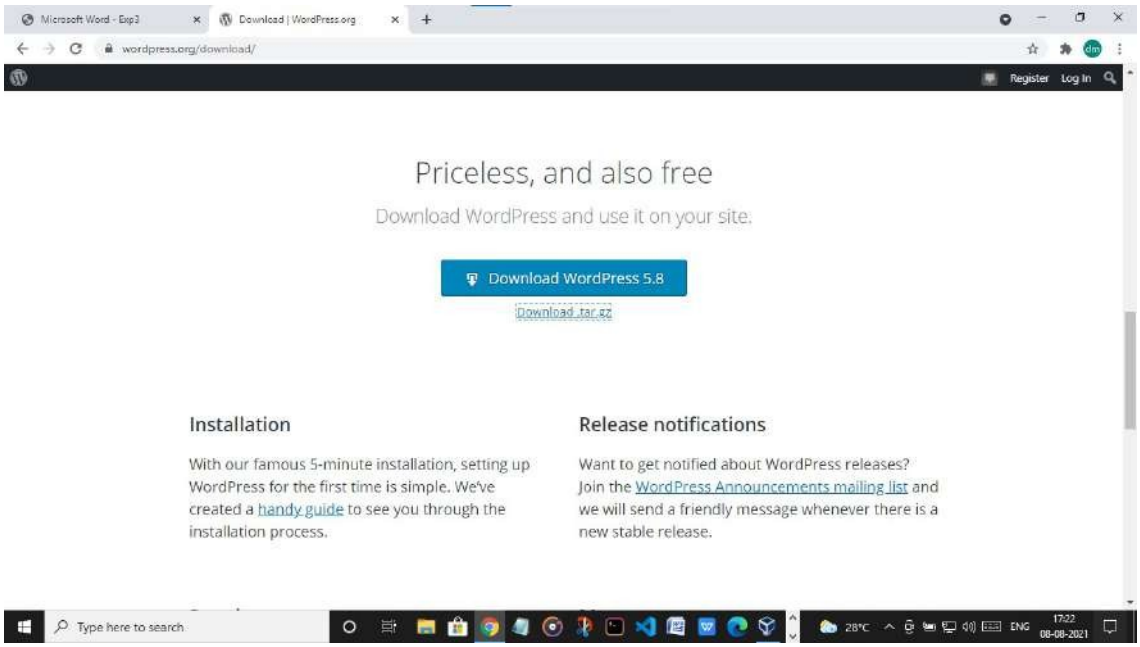
Step 2 :- install lamp server.



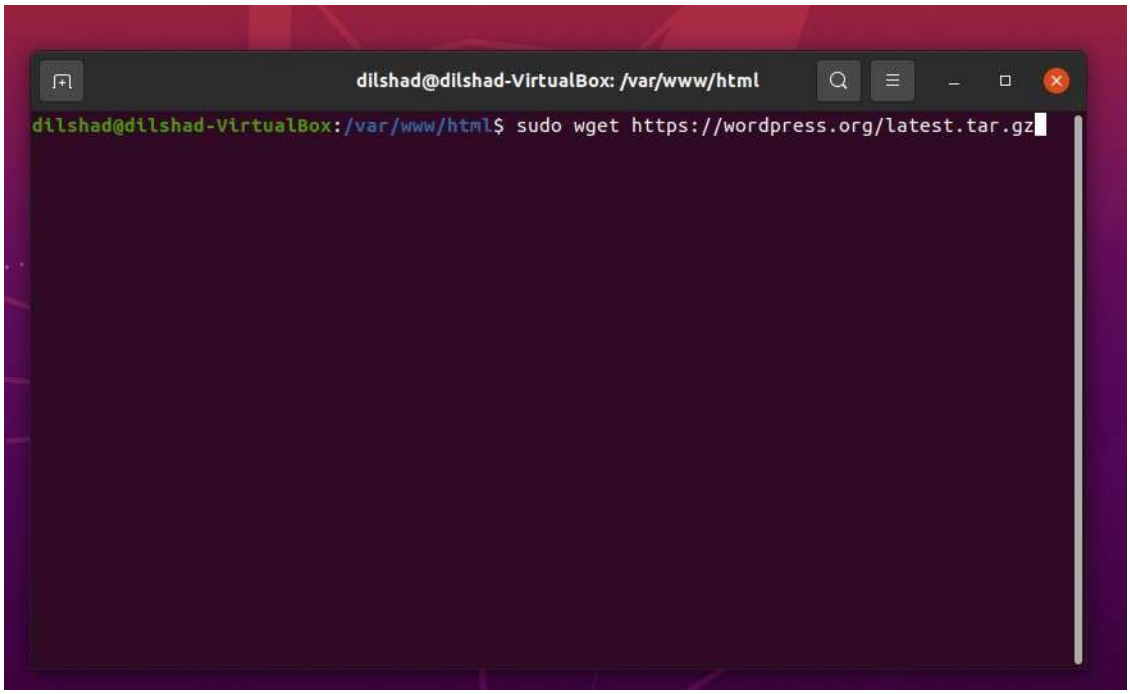
Step 3 :- change directory to root of apache.



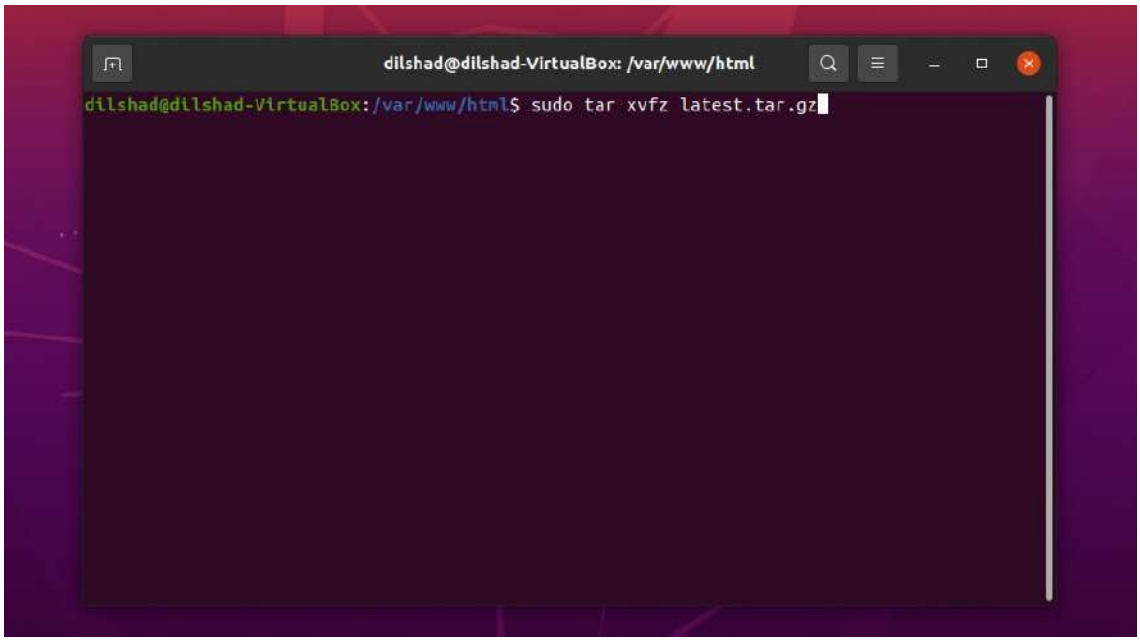
Step 4 :- copy download address of wordpress from wordpress.org



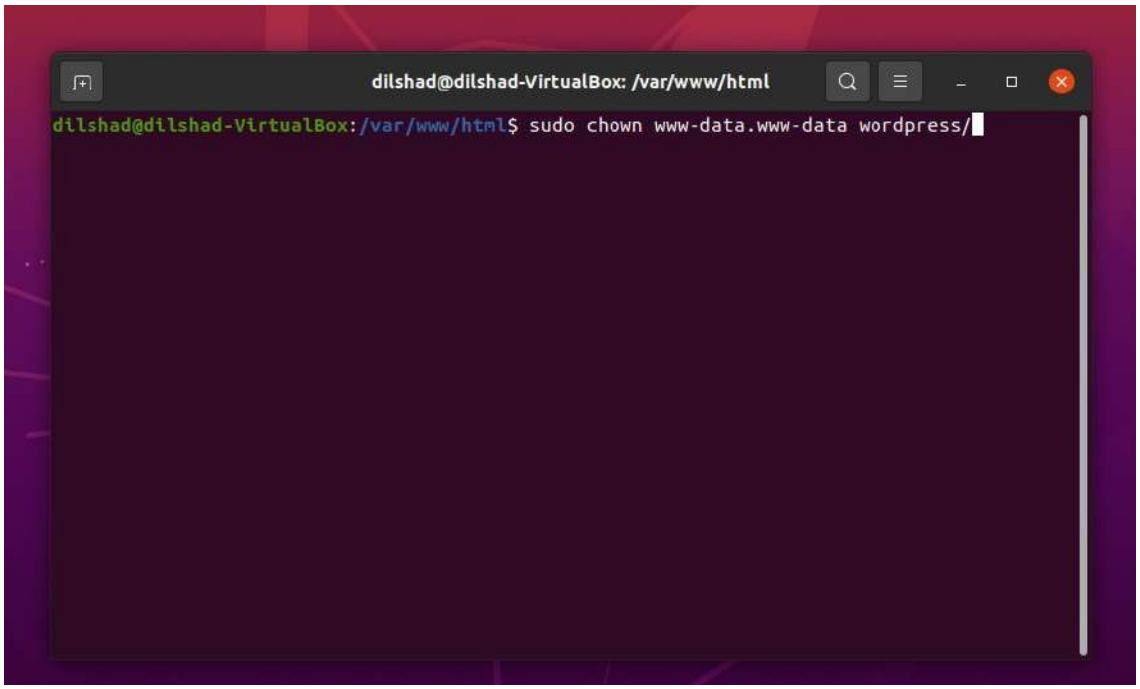
Step 5 :- download wordpress from terminal by pasting the copied address.



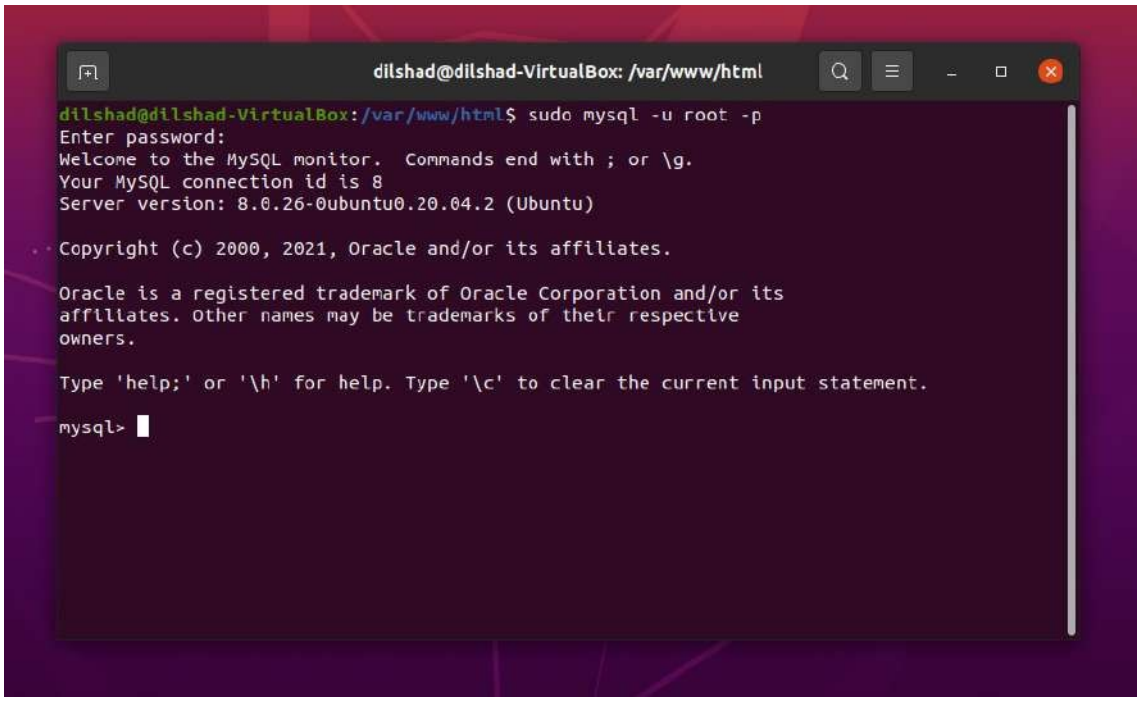
Step 6 :- unzip downloaded file.



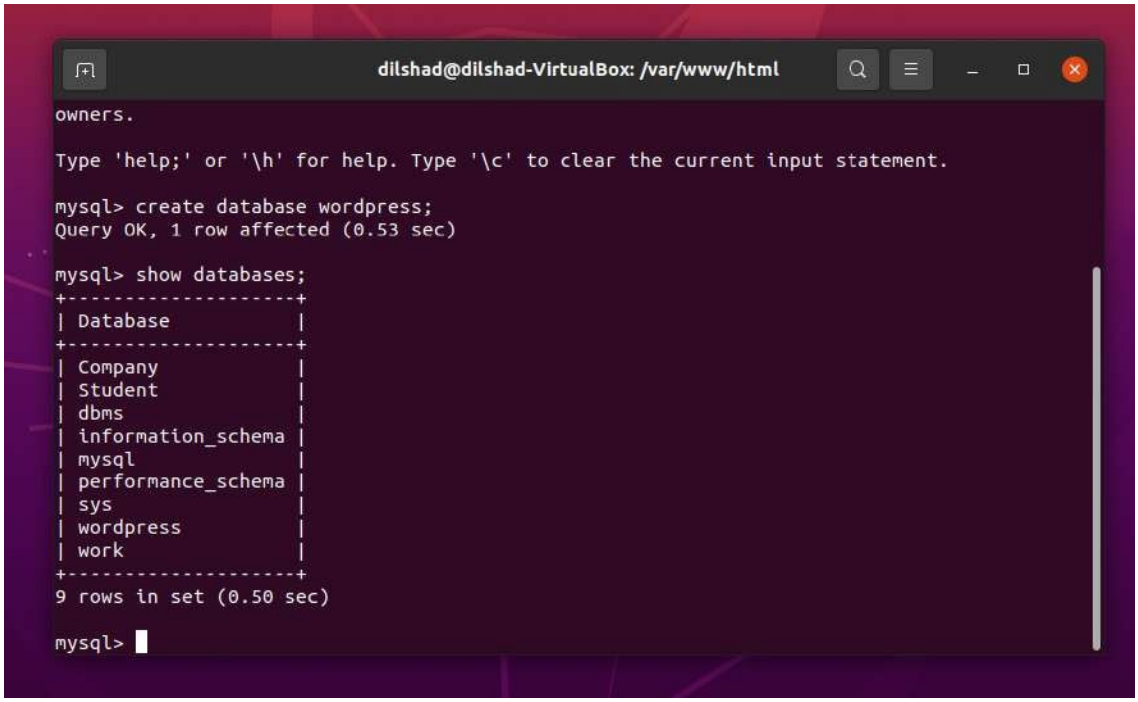
Step 7 :- change ownership of wordpress file.



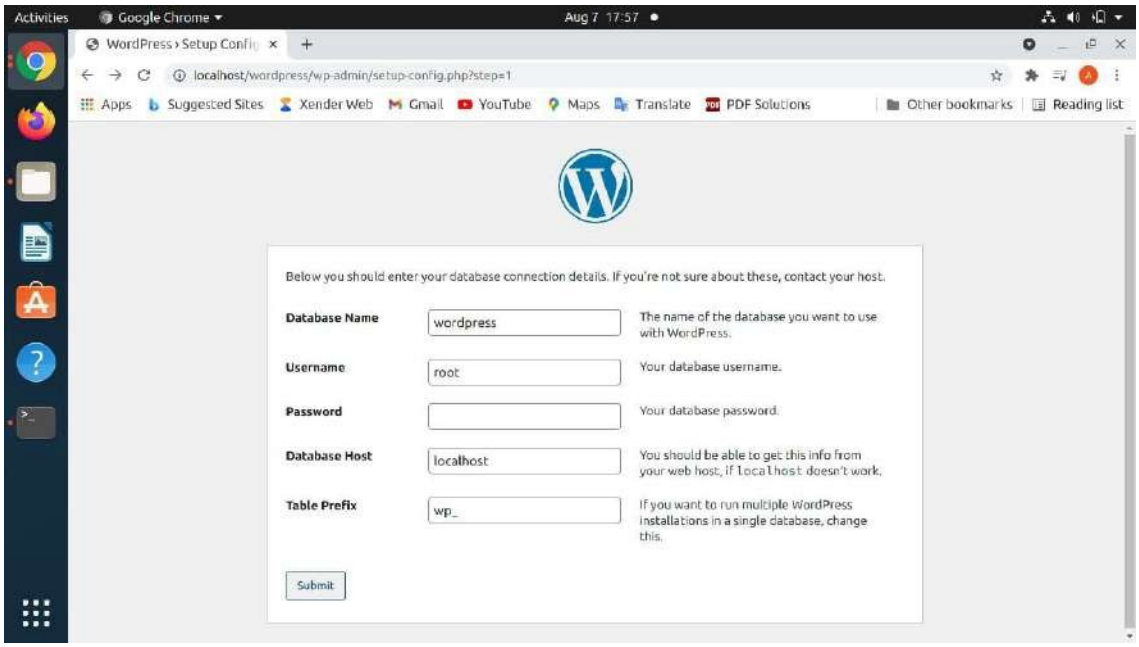
Step 8 :- open mysql in terminal.



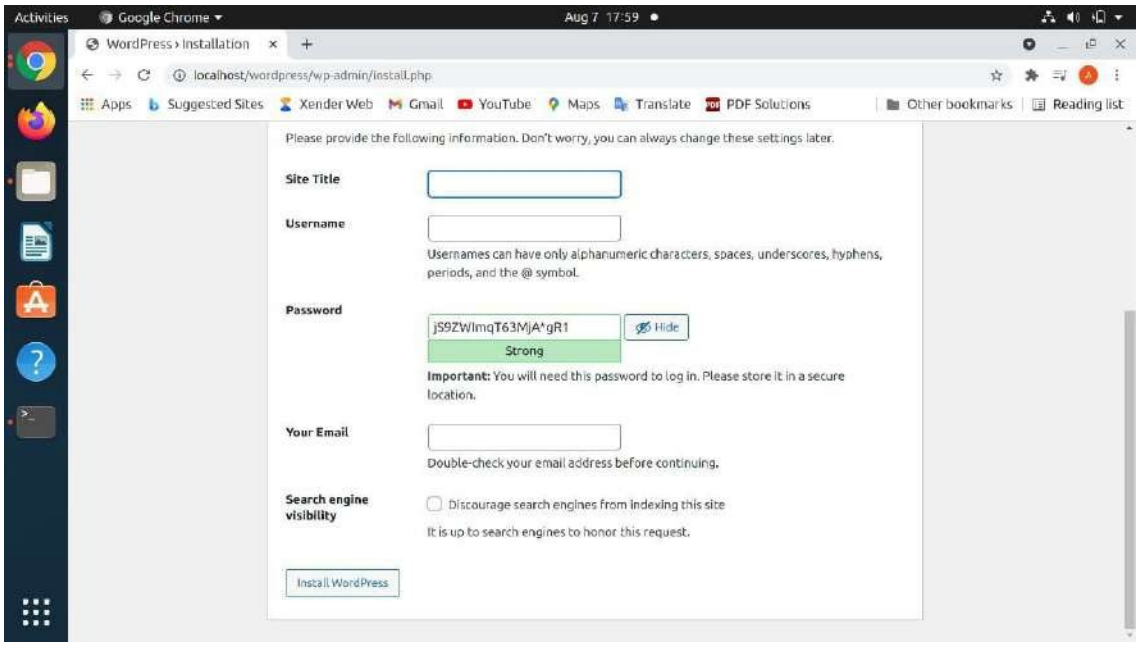
Step 9 :- create a database named wordpress.



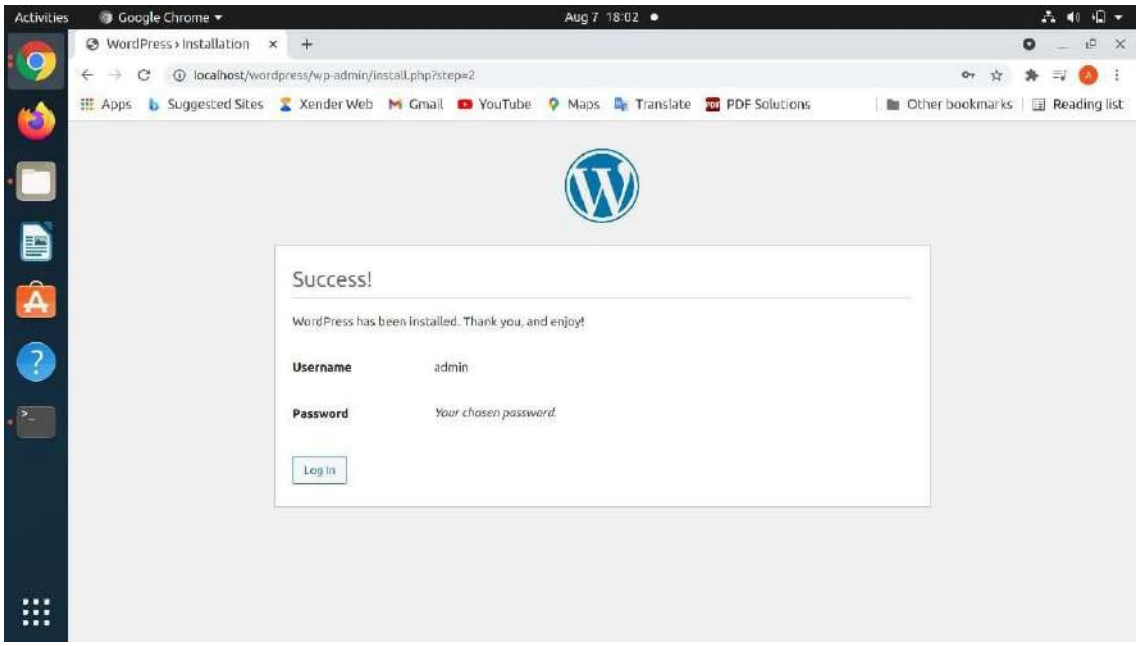
Step 10 :- open browser and localhost to wordpress .



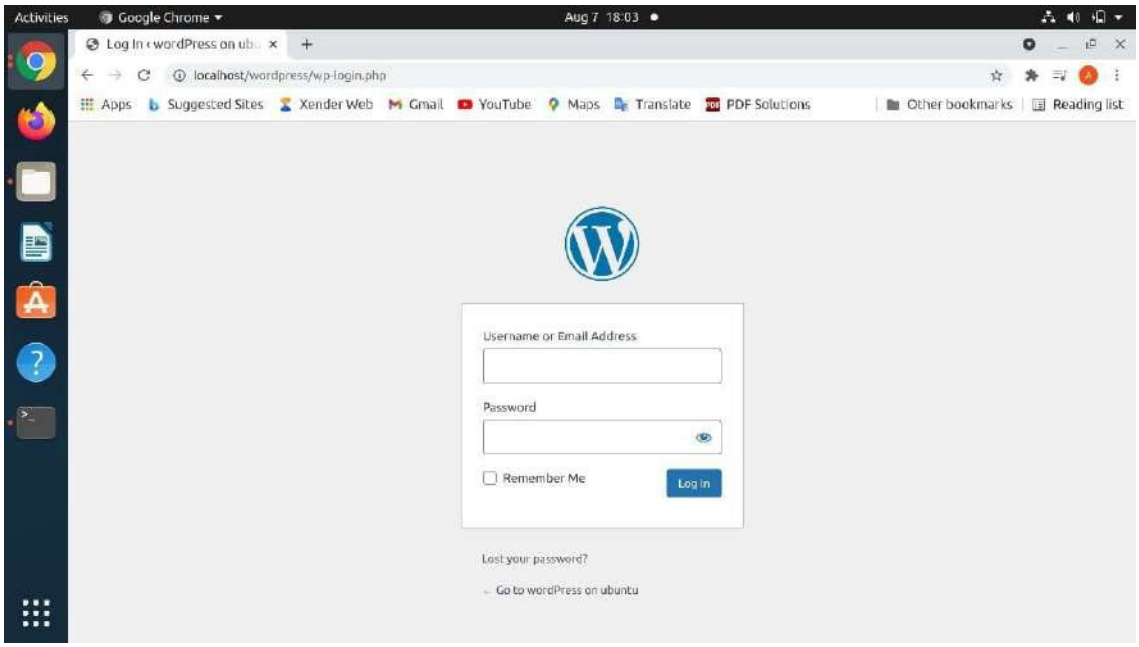
Step 11 :-

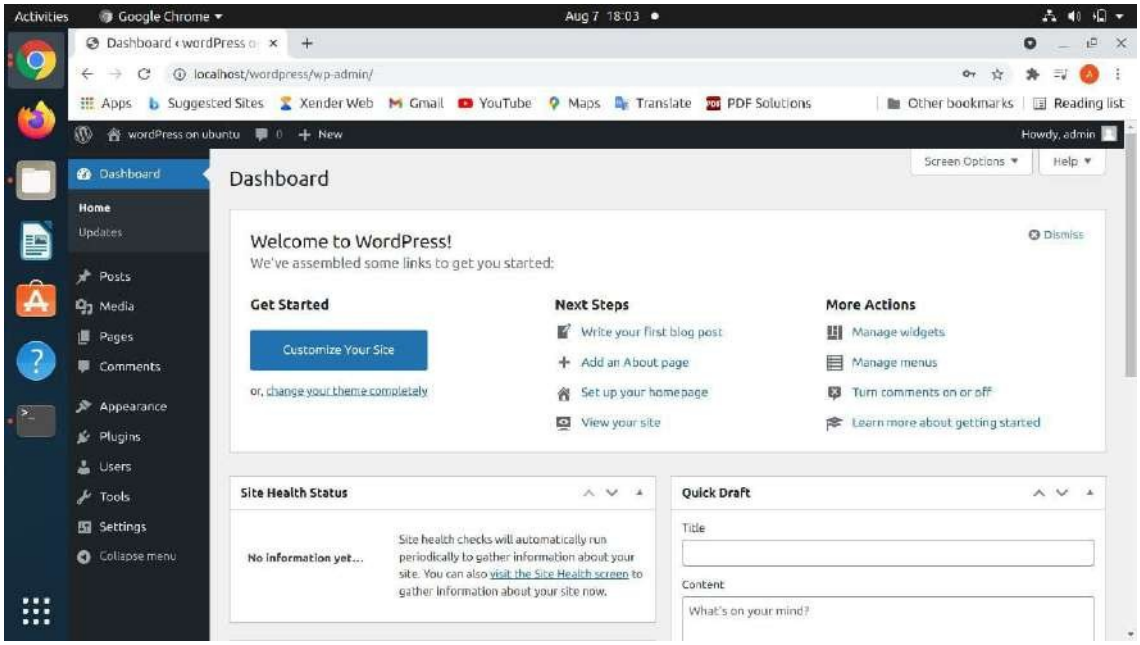


Step 12:-



Step 13 :- now login to start word press.

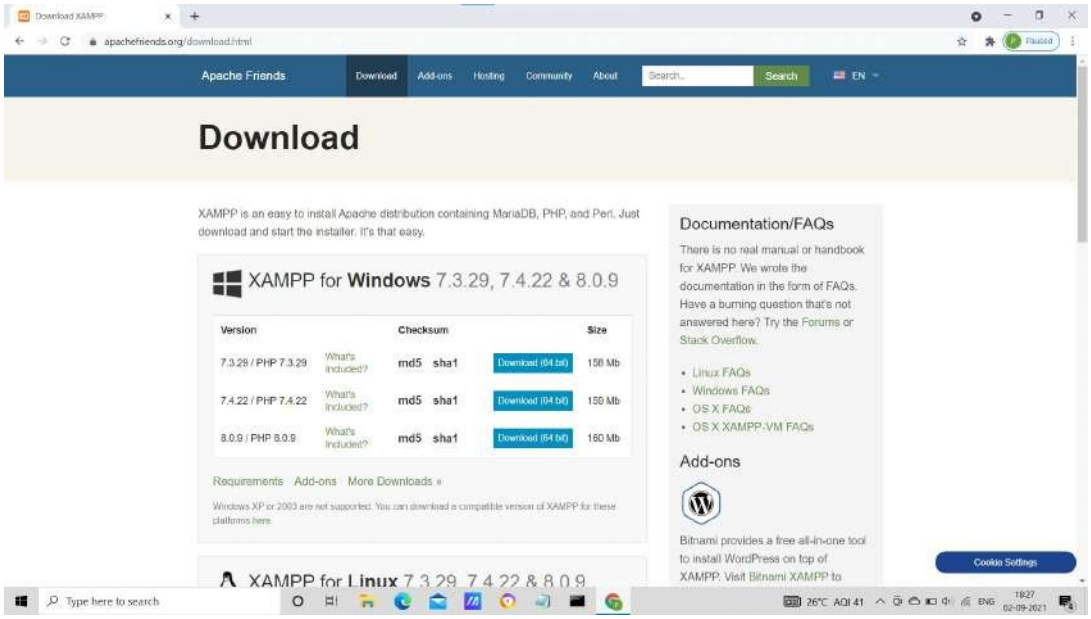



Activities
Google Chrome
Aug 7 18:03
Dashboard ‹ wordPress
localhost/wordpress/wp-admin/
Apps
Suggested Sites
Xender Web
Gmail
YouTube
Maps
Translate
PDF Solutions
Other bookmarks
Reading list
wordPress on ubuntu
New
Howdy, admin
Screen Options
Help
Dashboard
Home
Updates
Posts
Media
Pages
Comments
Appearance
Plugins
Users
Tools
Settings
Collapse menu
Dashboard
Welcome to WordPress!
We've assembled some links to get you started:
Dismiss
Get Started
Customize Your Site
or, change your theme completely
Next Steps
Write your first blog post
Add an About page
Set up your homepage
View your site
More Actions
Manage widgets
Manage menus
Turn comments on or off
Learn more about getting started
Site Health Status
No information yet...
Site health checks will automatically run periodically to gather information about your site. You can also visit the Site Health screen to gather information about your site now.
Quick Draft
Title
Content
What's on your mind?

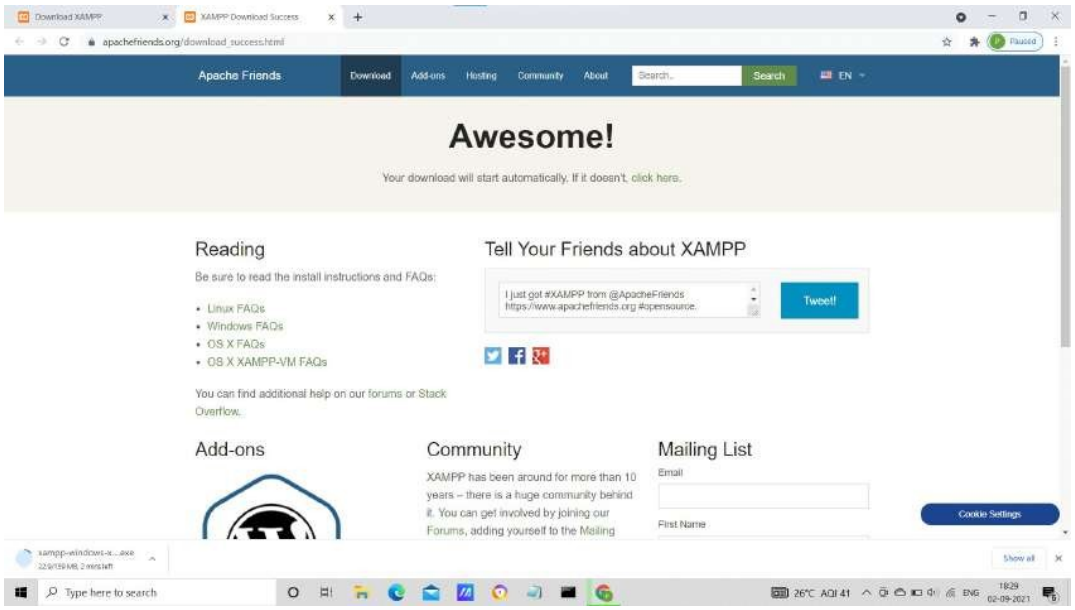
Installation of Laravel

For installing laravel we need xampp and composer.

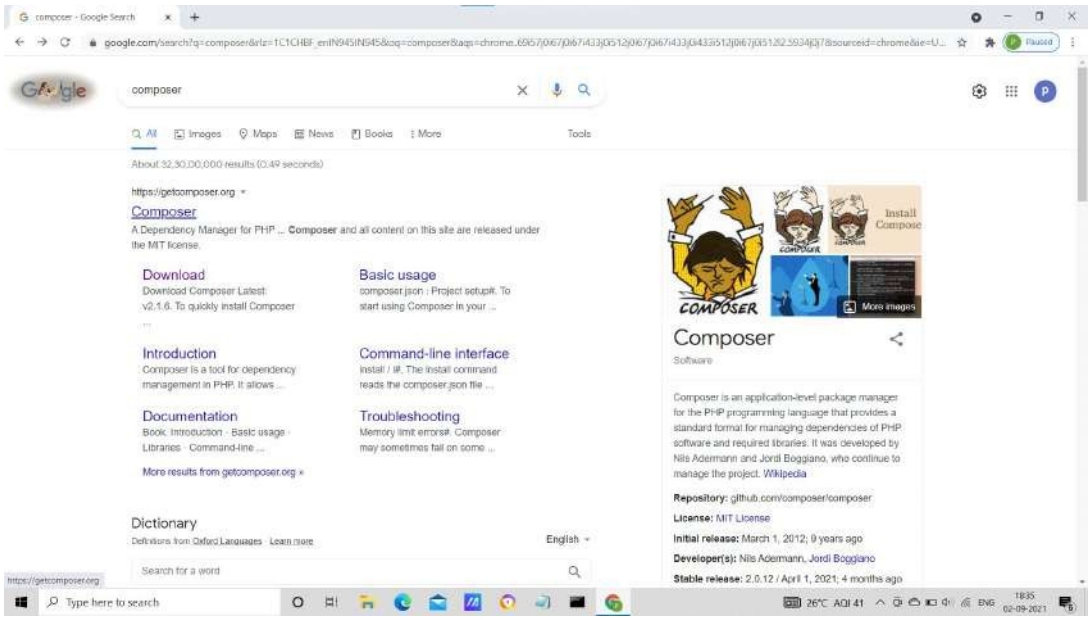
Step 1: open chrome and search for xampp and download it.



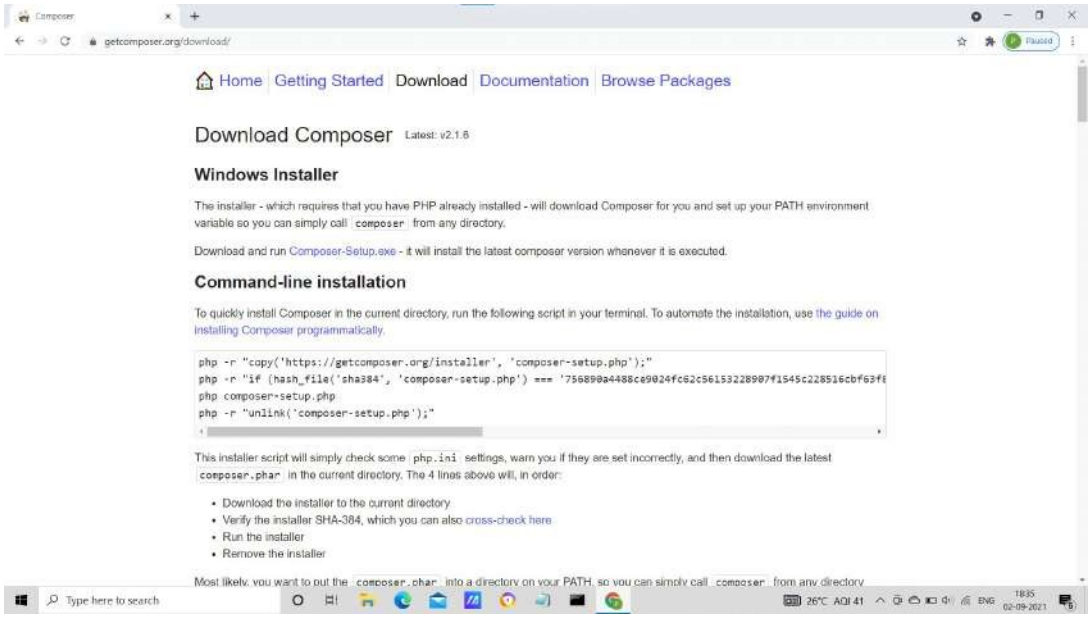
Download success



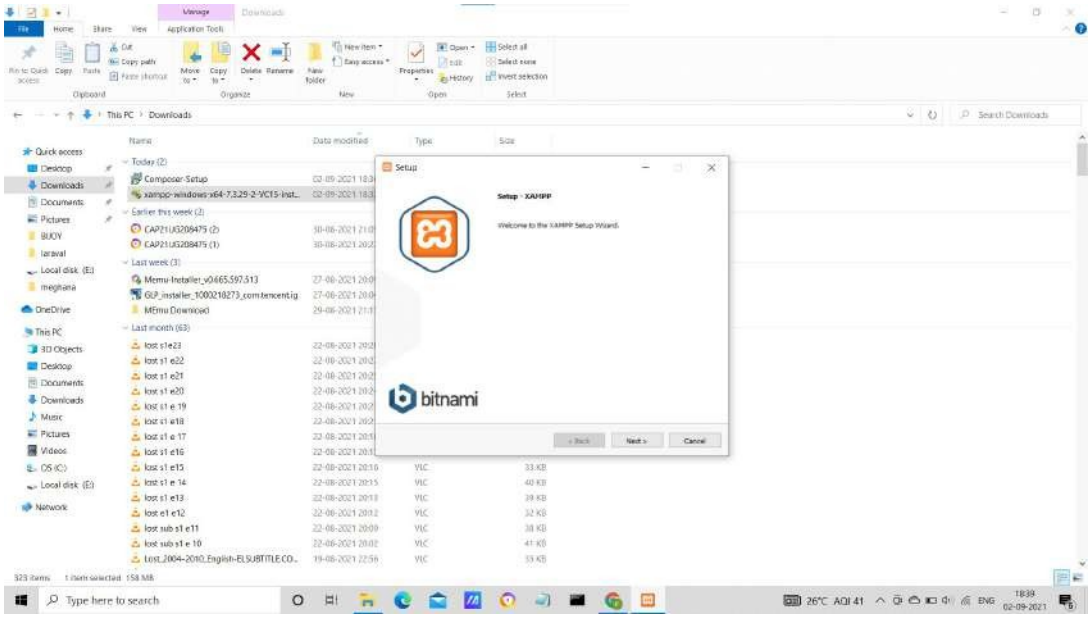
Step 2 : then download composer by searching in chrome.



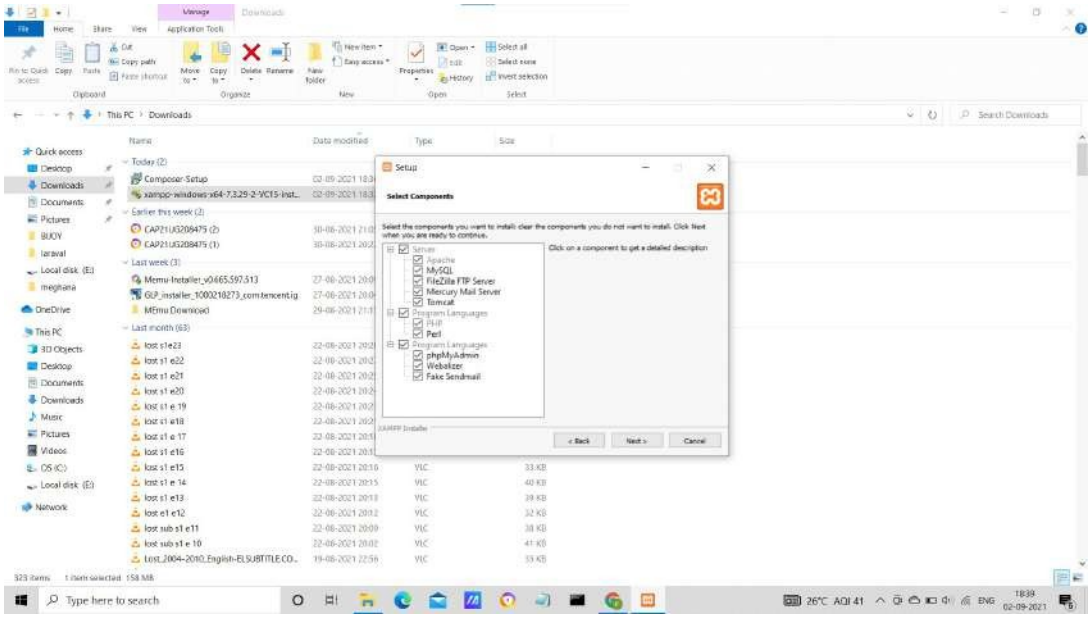
Click on the *composer-setup.exe* to download composer.



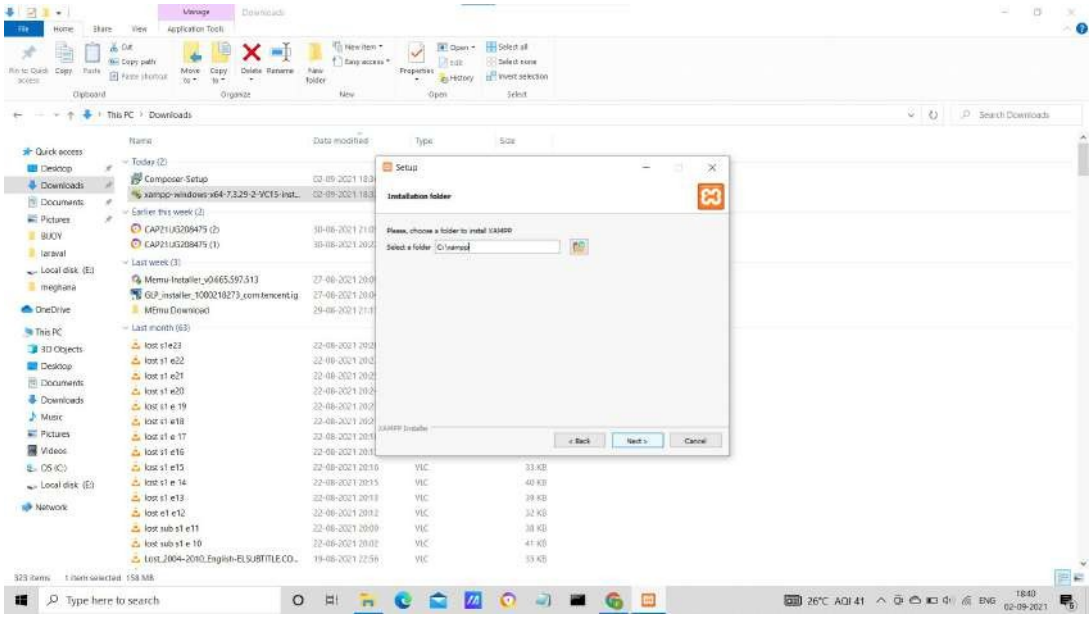
Step 3 : install the downloaded xampp software by following install procedures.



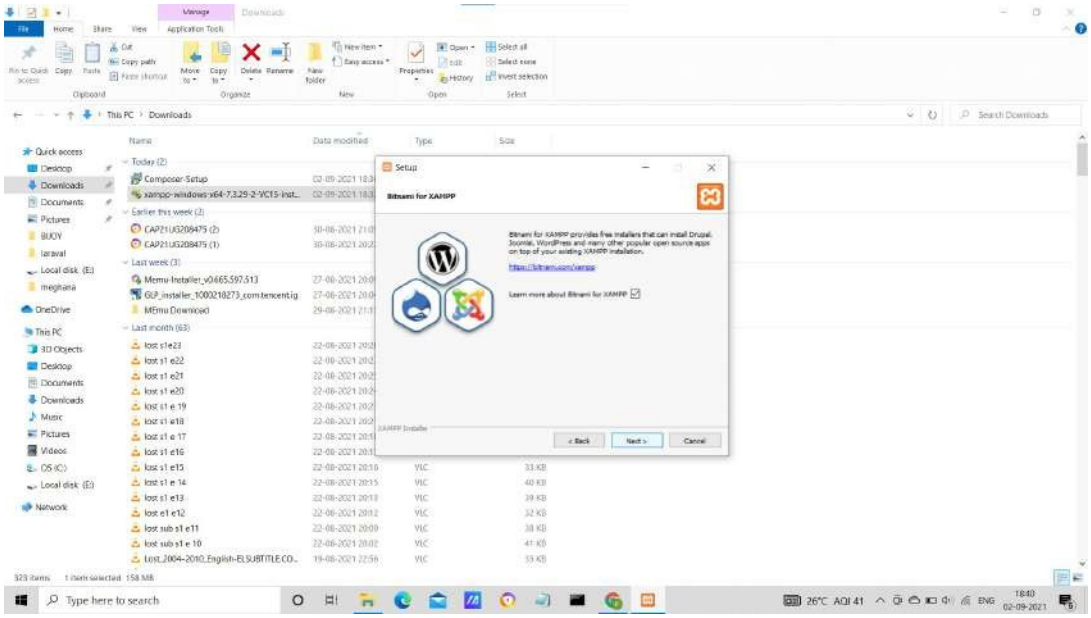
Click next



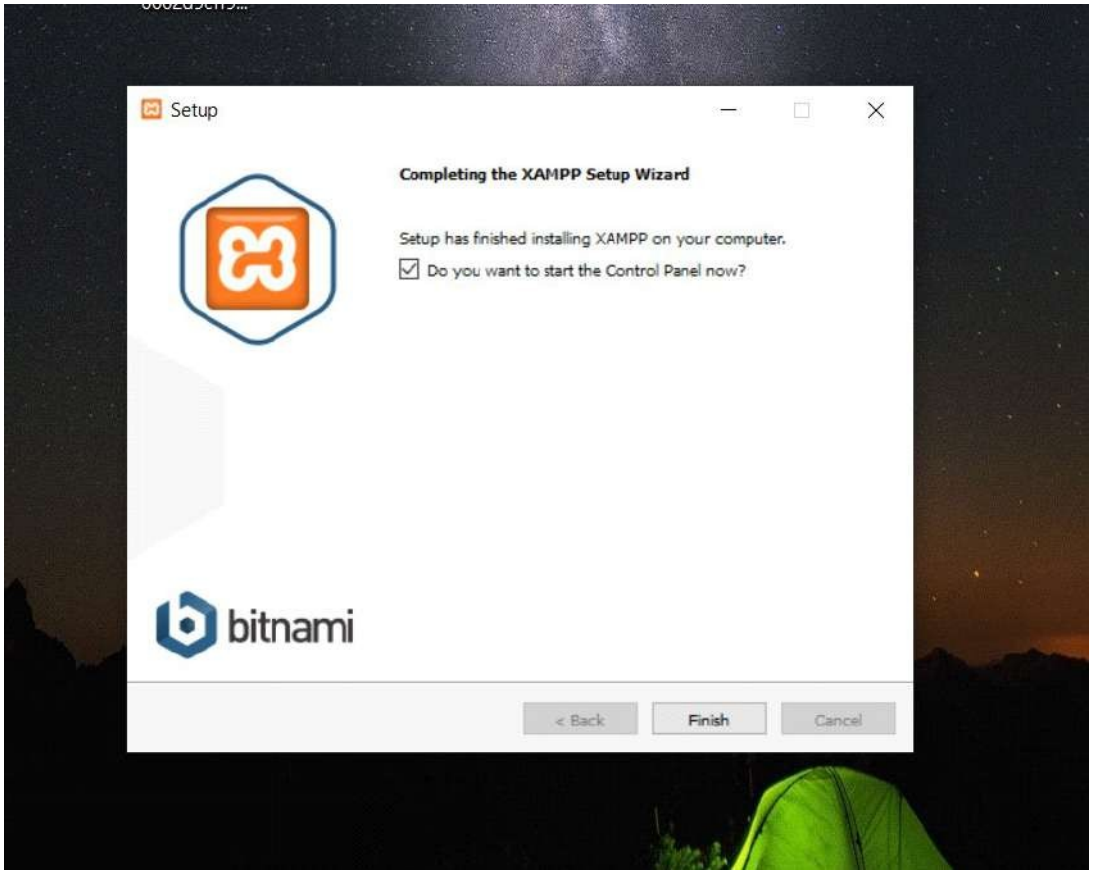
Click next



Click next

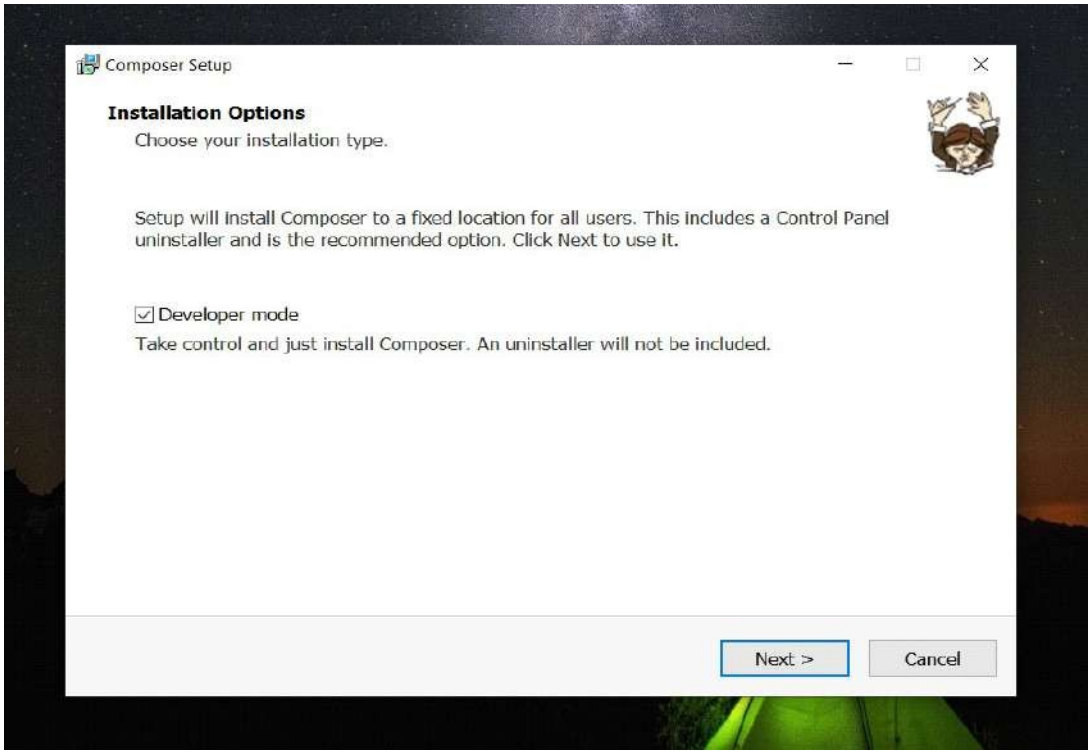


Click next

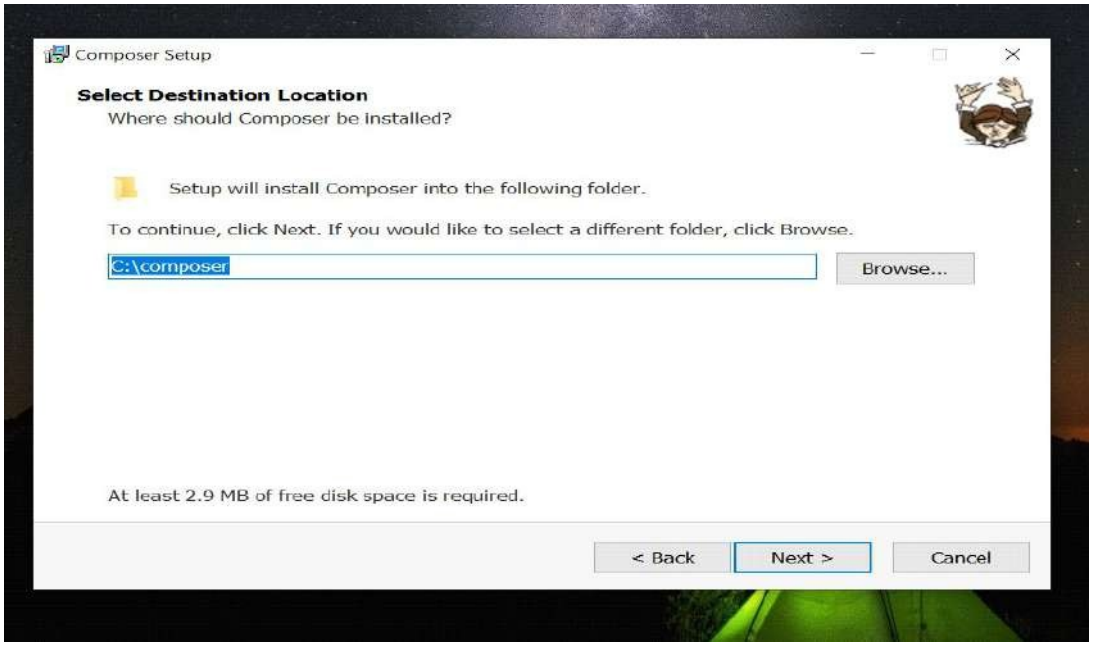


Xampp installation complete.click finish.

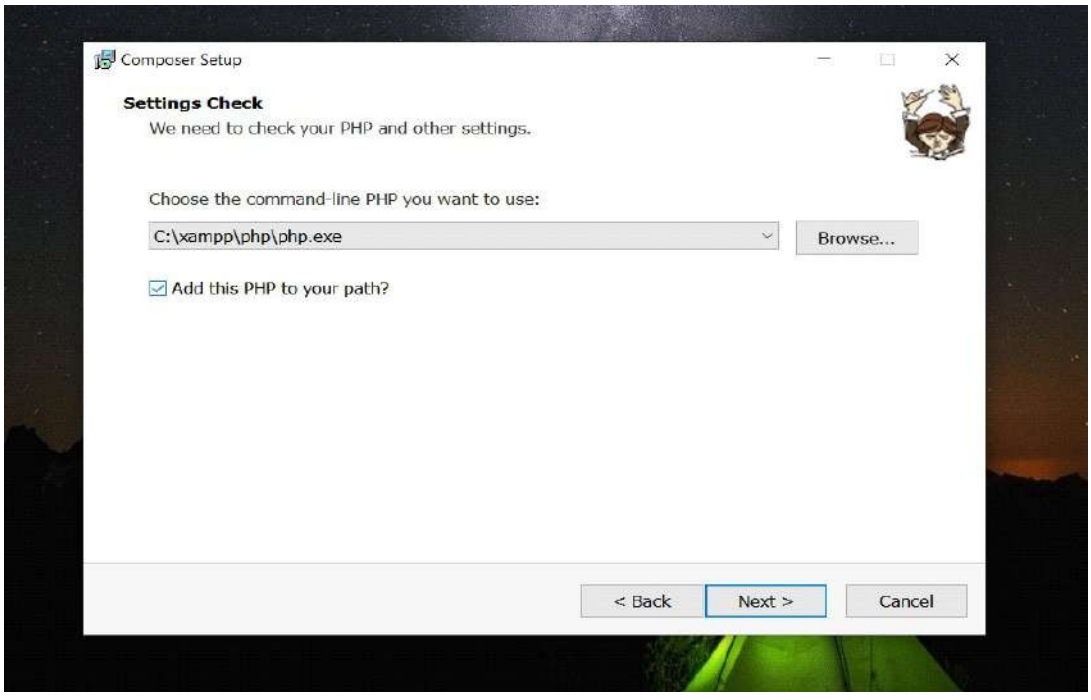
Step 4 : install composer.



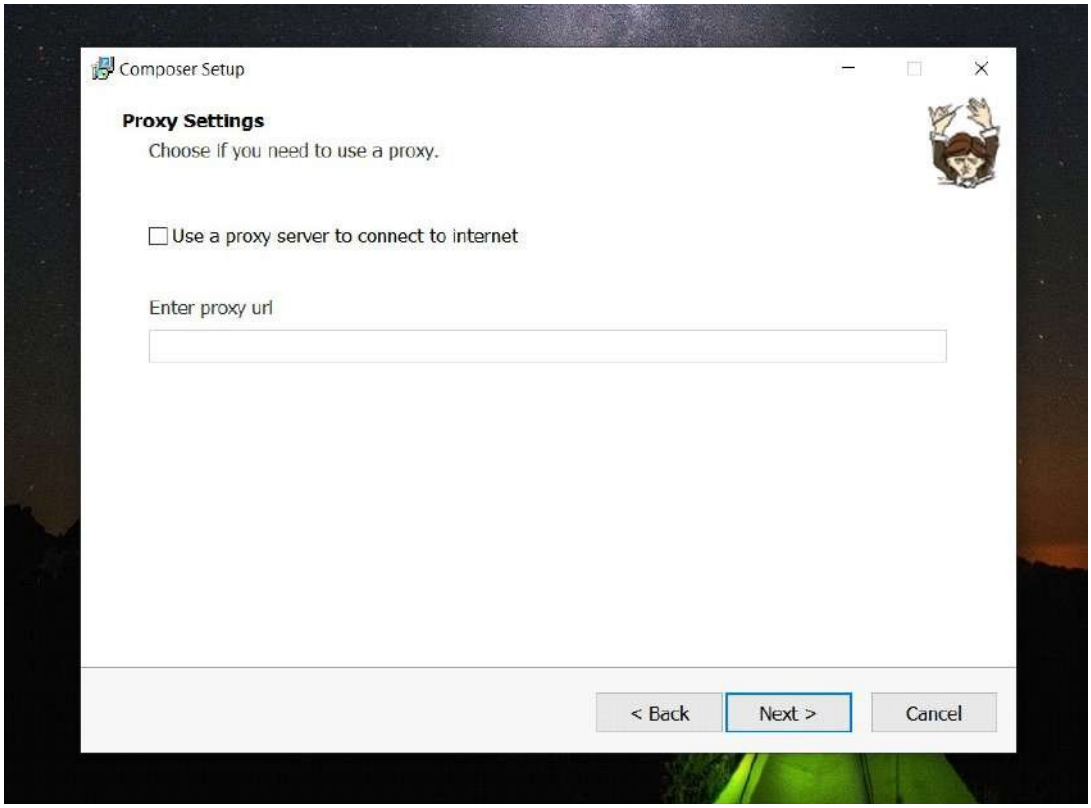
Click next



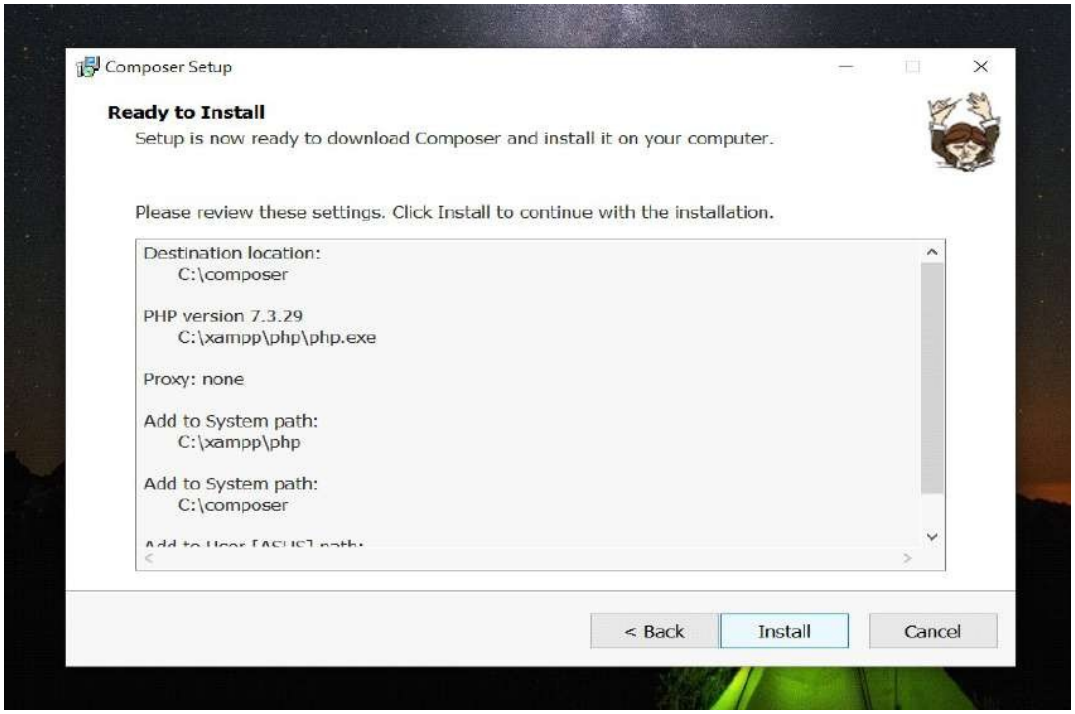
Click next



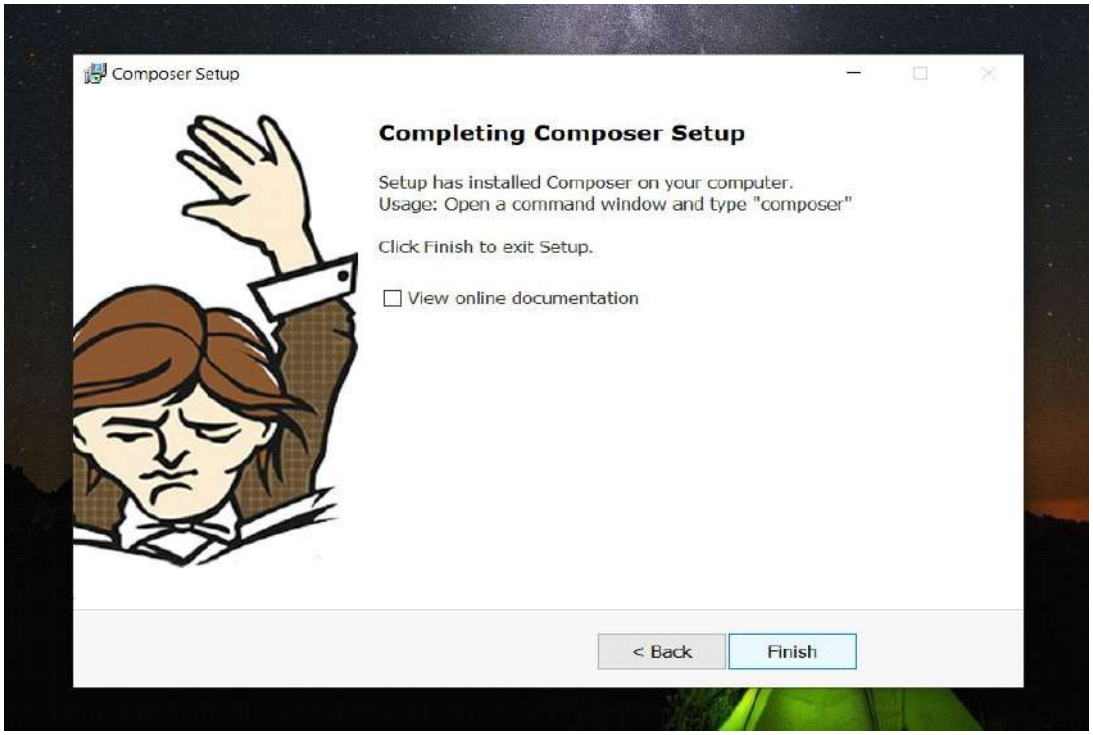
Setup the path and click next.



Click next by leaving the input as blank.



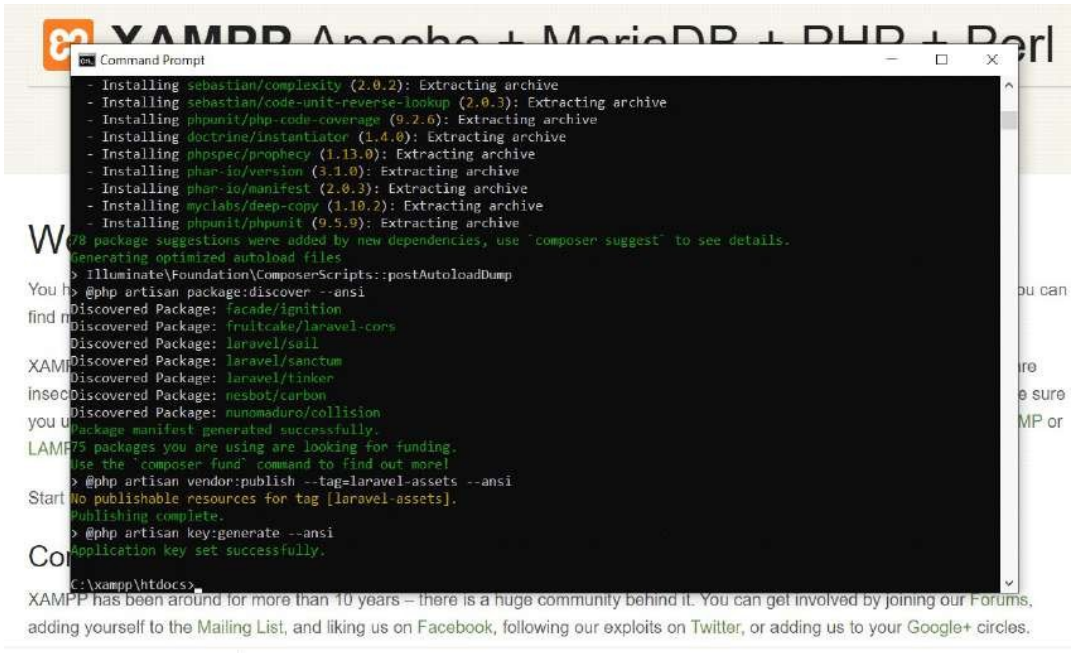
Click install.



Composer installation complete . click finish.

Step 5 : open command prompt and cd to htdocs.

Step 6 : type the command *composer create project laravel/laravel myapp* to initialise laravel with name as myapp.(install laravel)



Laravel installation complete.

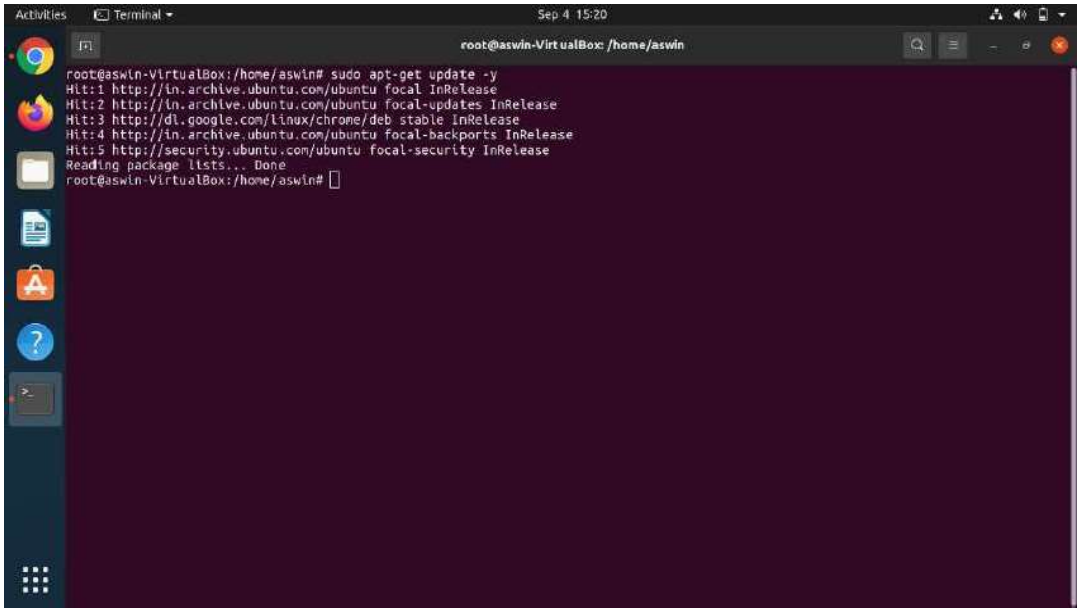
Step 7 : cd over to myapp and type the command *php artisan serve* ,this gives the laravel address to access.

EXPIRIMENT-7:

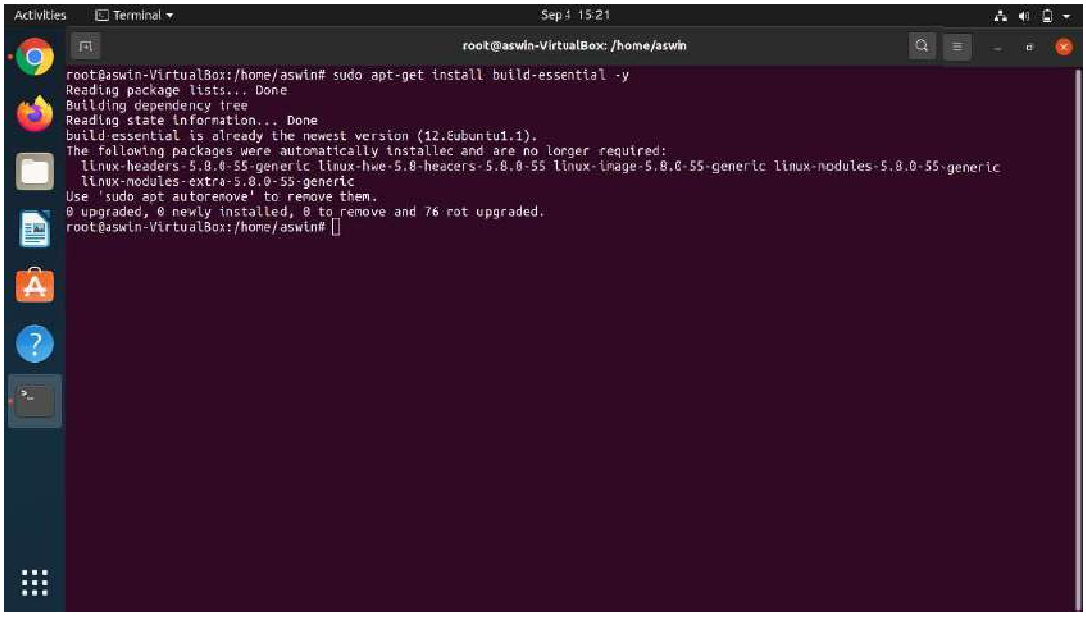
Aim: Build and install software from source code, familiarity with make and make utilities expected.

Solution :-

Step 1: make packages uptodate.

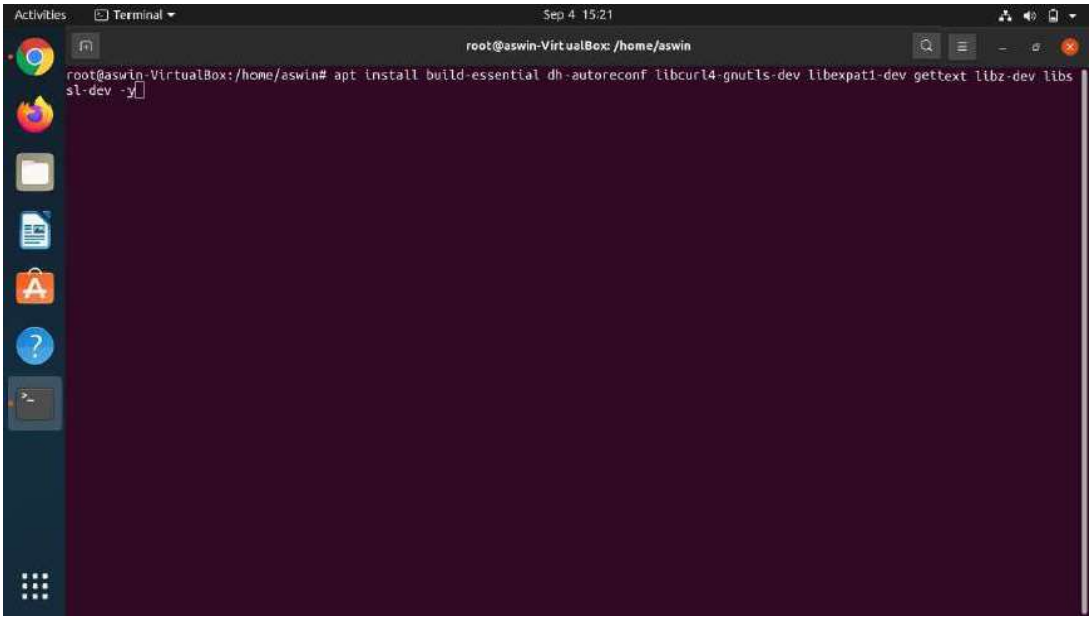


- Next, you'll need to make sure you have a compiler available. Run this command to install build-essential:



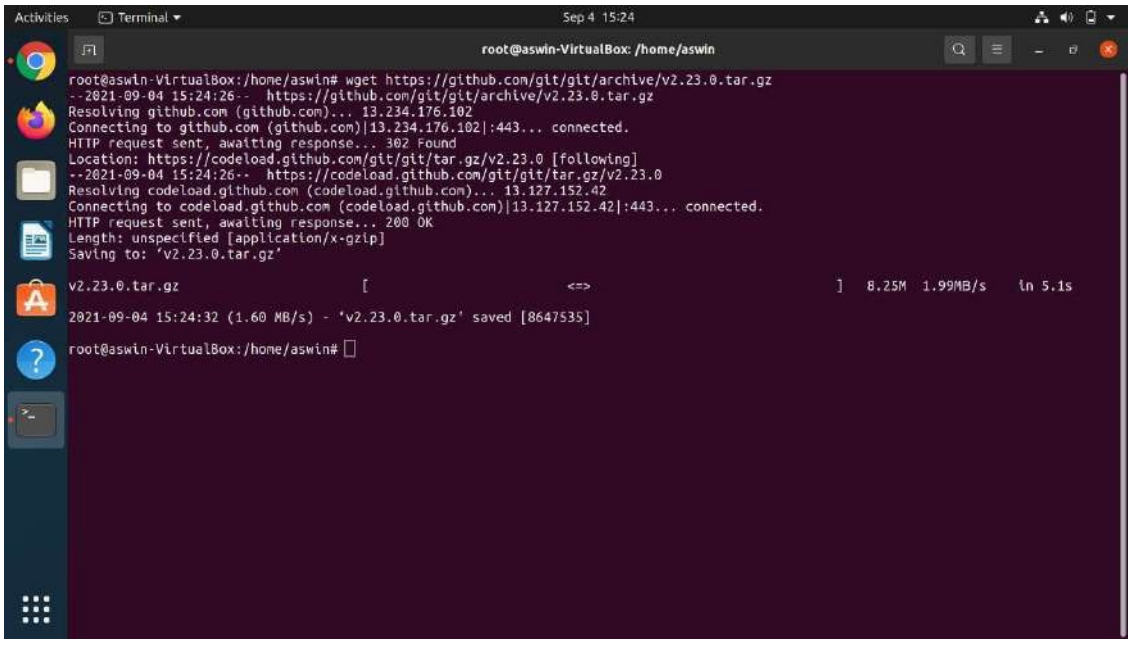
Step 2: Download Dependencies

- When installing a package from source code, you'll need to manage the installation of the package dependencies. We'll use apt-get to install git's dependencies:

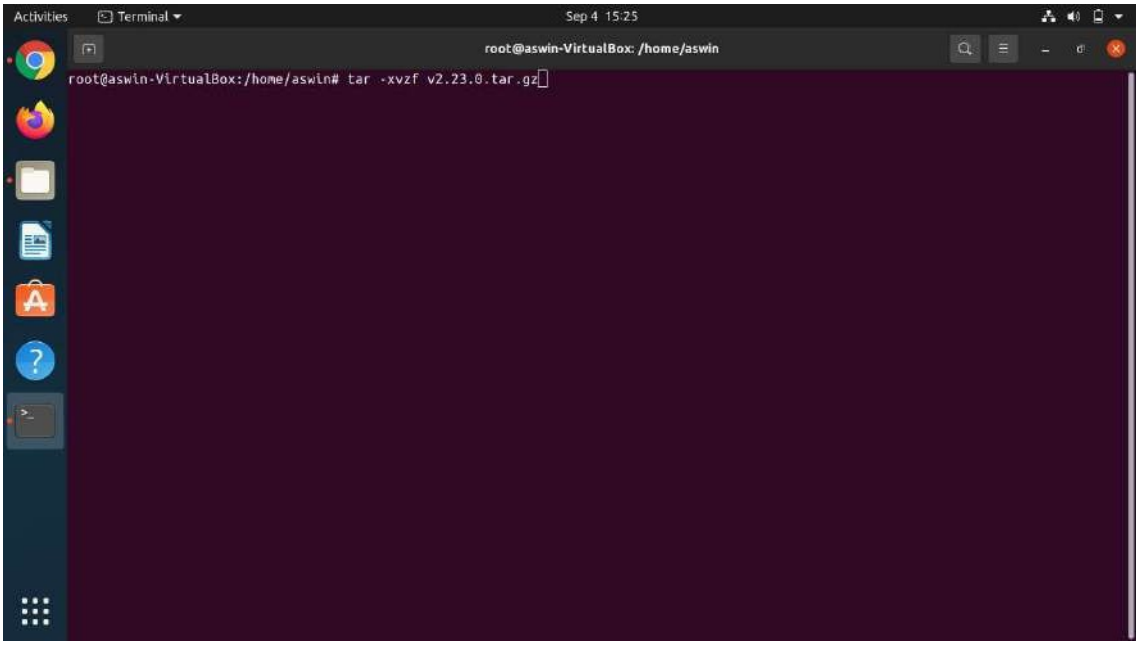


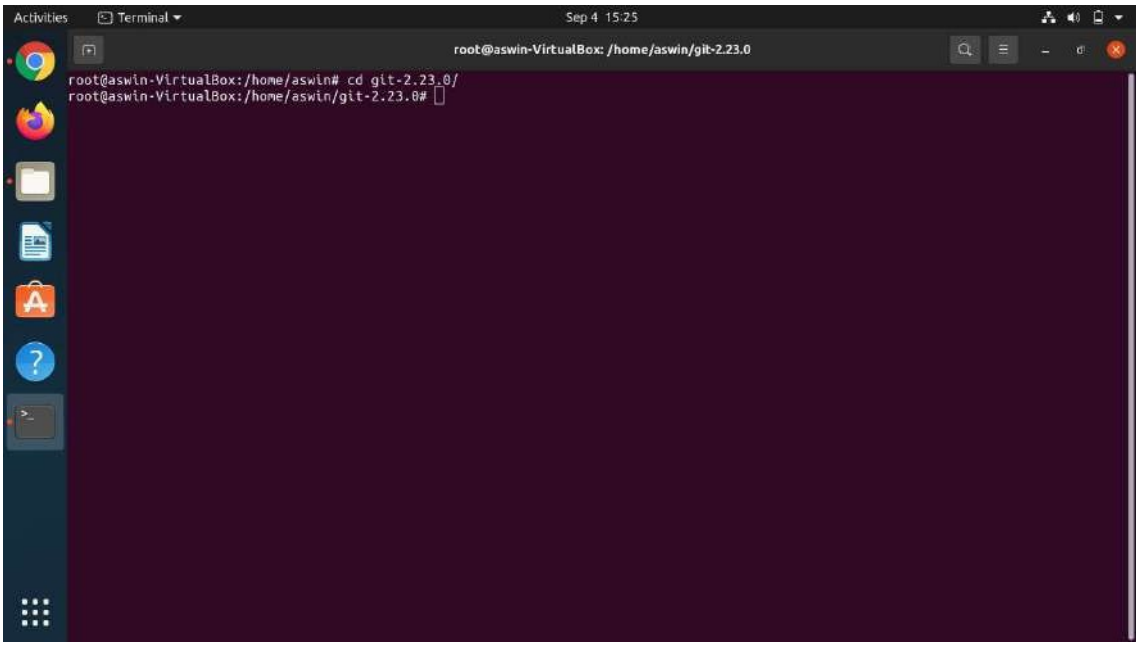
Step 3: Download the Source Package

- Once the package dependencies are in place, it’s time to download the package with wget:



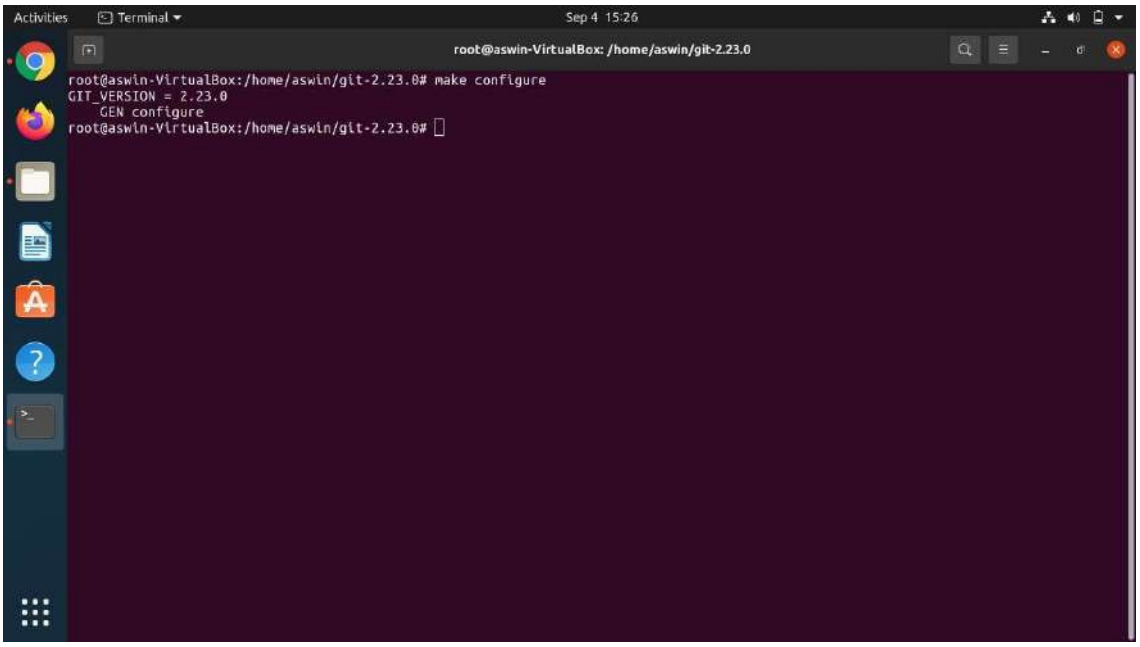
- Next, we need to extract the archive and cd (change directories) into the new git directory:



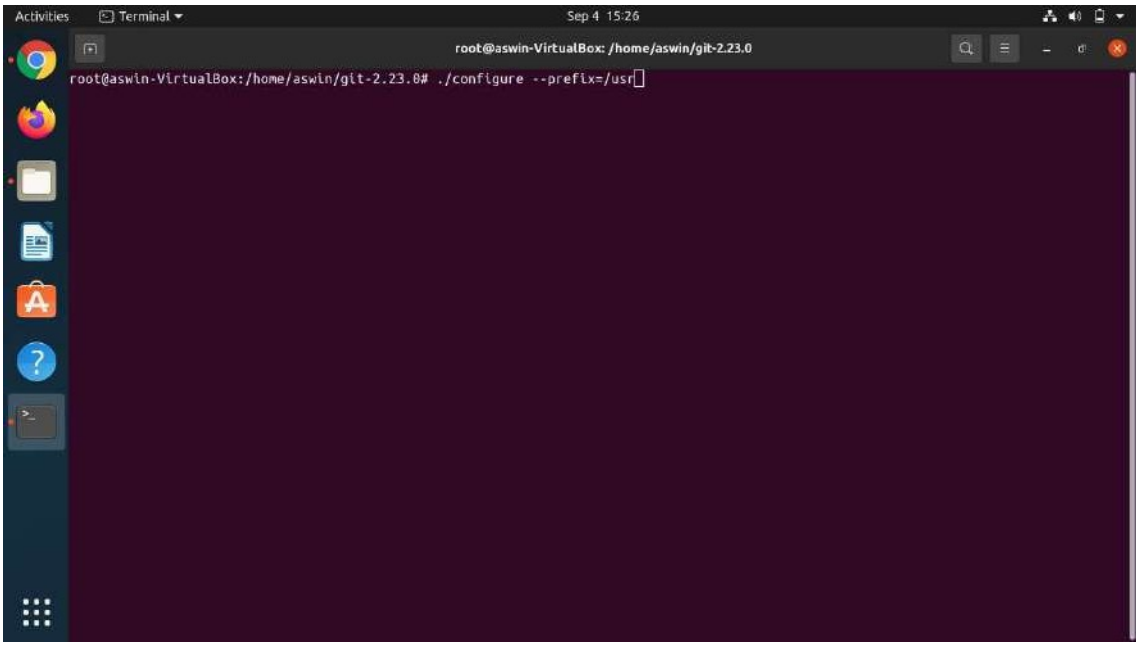


Step 4: Install Git

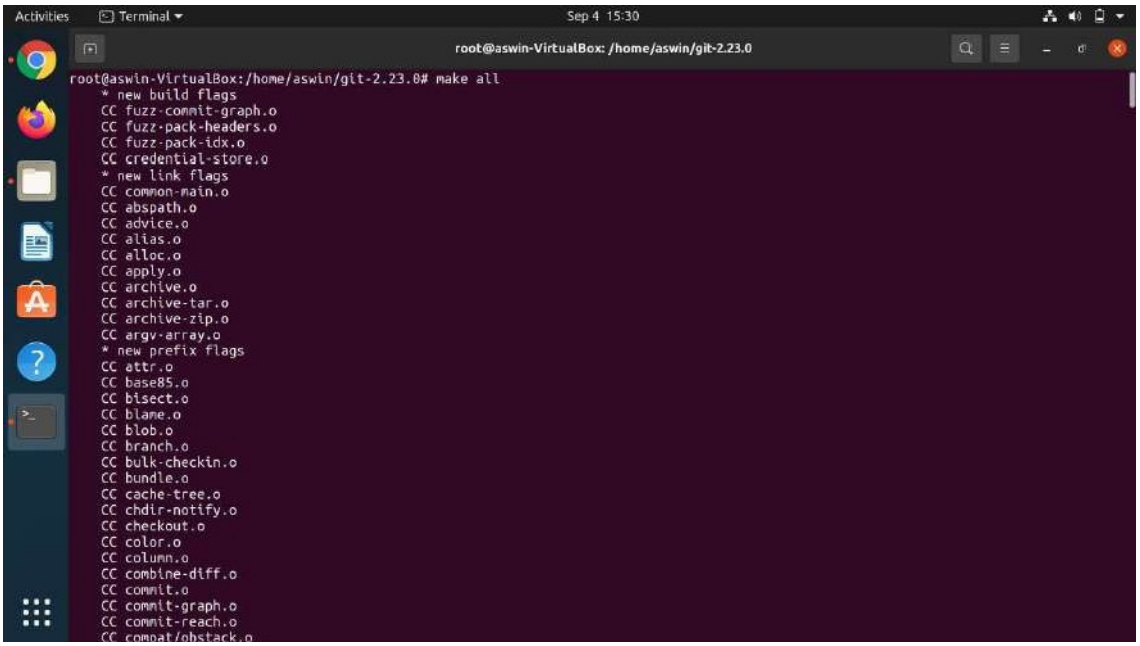
- Now that we have our package extracted and ready to go, we need to configure it:



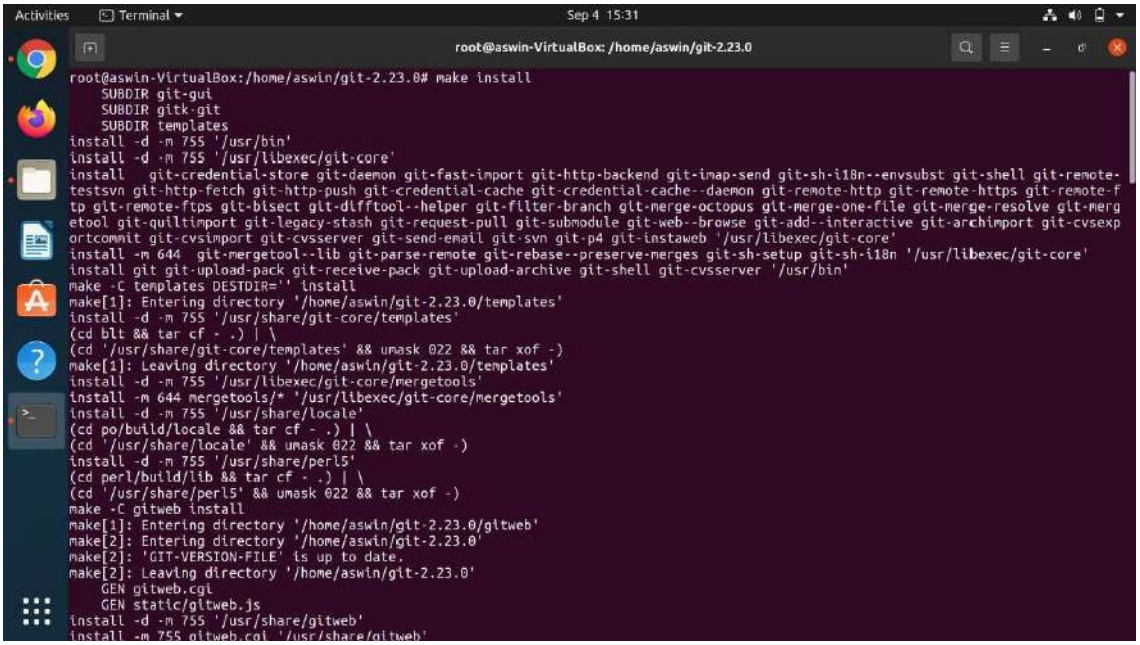
- Next, let's verify that all of the dependencies necessary to build the package are available by running this command:



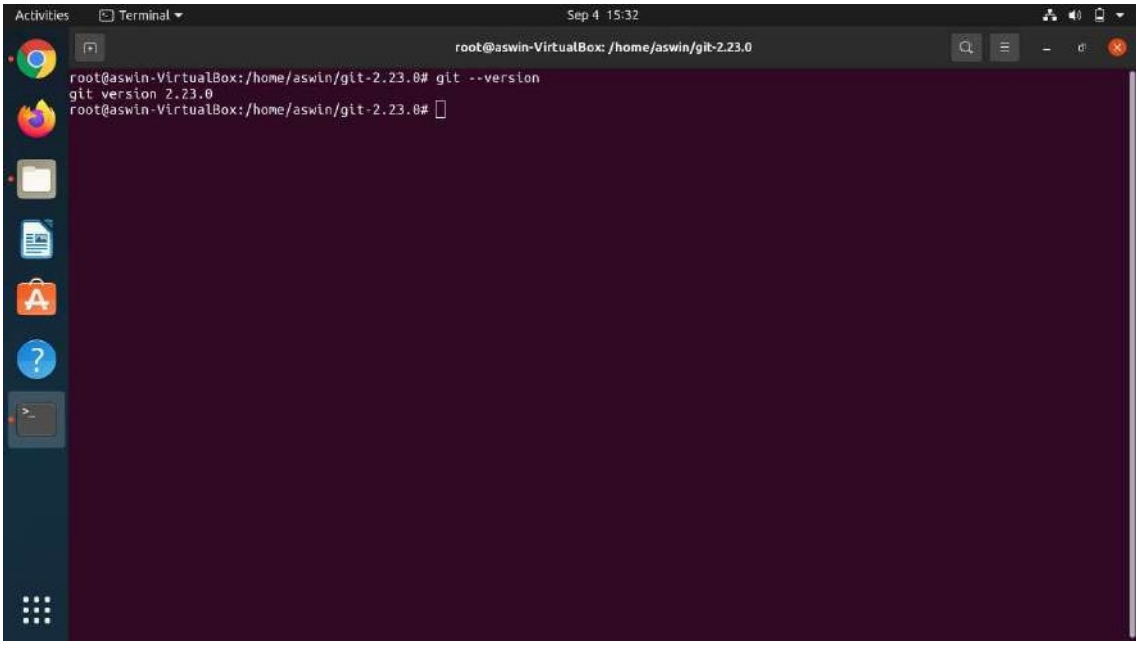
- then build the source code:



- now install git.



- installation success.

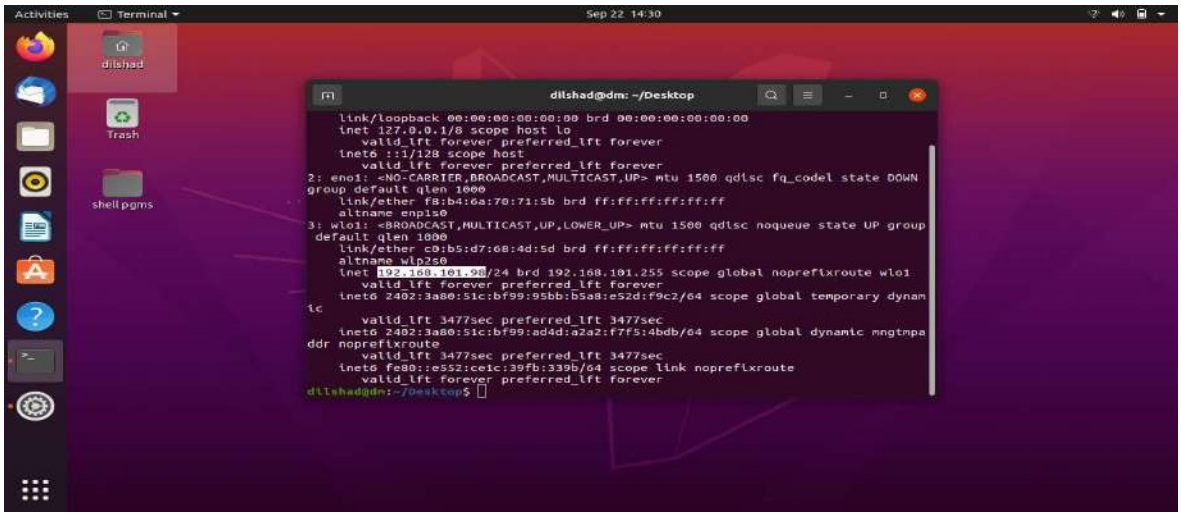


Experiment 8

Ipv4 –networking

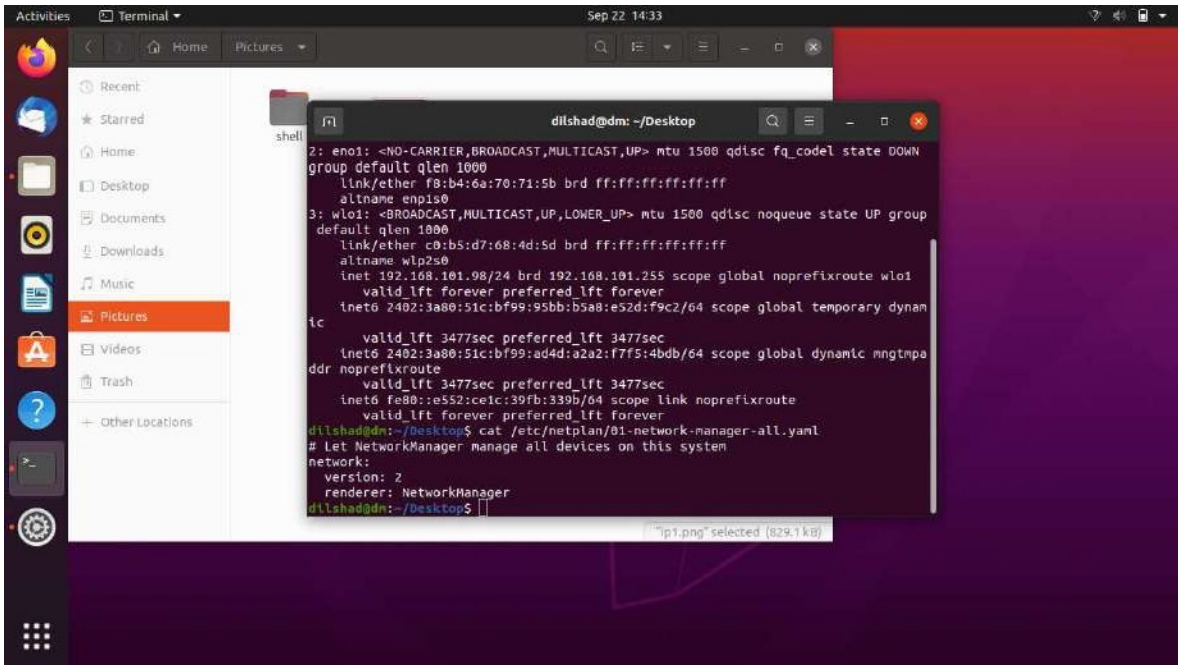
Step 1 :

Type ip address in the terminal and find out current ip address of network



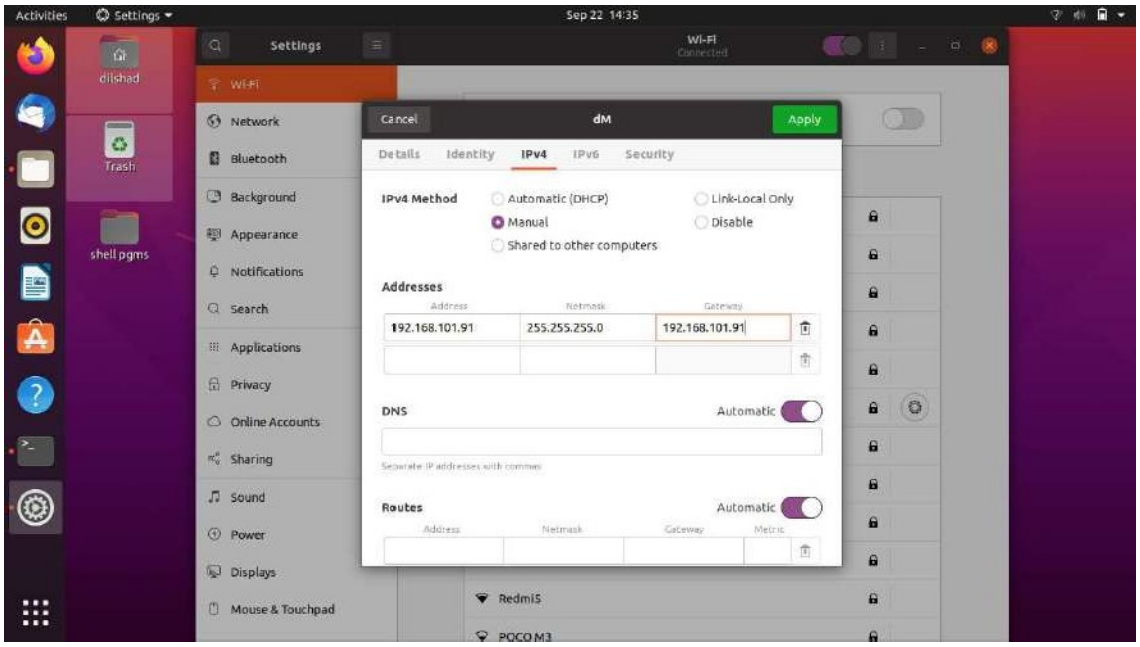
Step 2 :

Type cat /etc/netplan/01-network-manager-all.yml



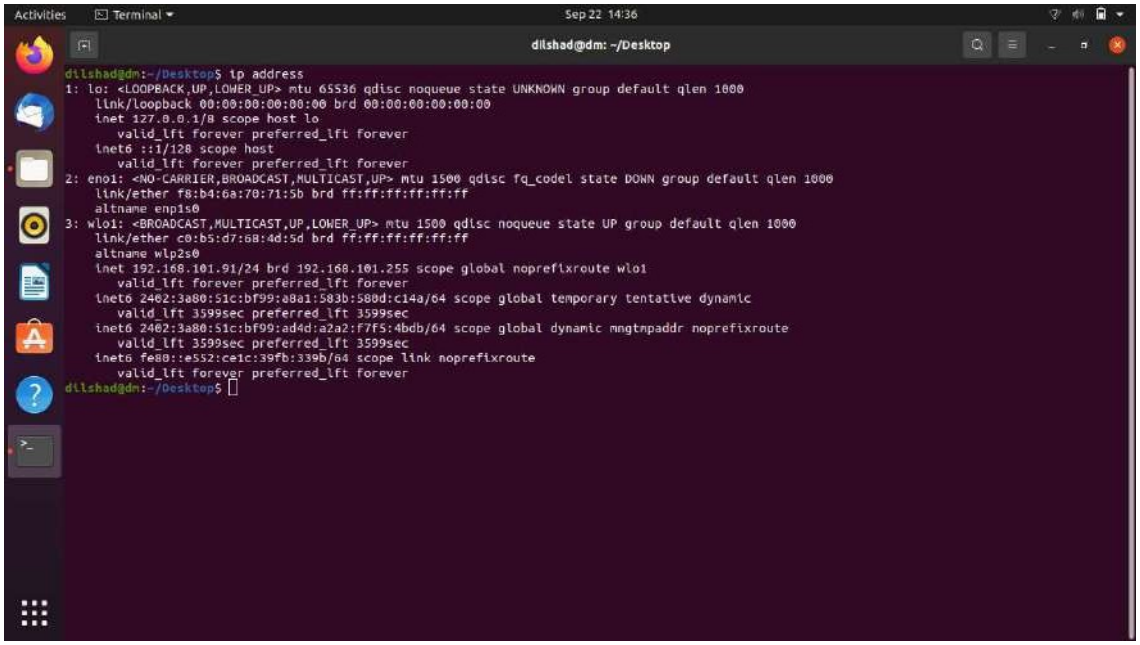
Step 3 :

Open wifi connection settings and click on ipv4 and change to manual from automatic. Change the current ip address and click apply.



Step 4:

Ip address get changed.



Configure and Set Up a Firewall on Ubuntu

UFW stands for Uncomplicated Firewall which acts as an interface to IPTABLES that simplifies the process of the configuration of firewalls it will be a very hard for a beginner to learns and configure the firewall rules where we will secure the network from unknown users are machines. UFW works on the policies we configure as rules. For this, we needed a non-root user with root permission on the machine.

Installing the UFW (Firewall)

UFW is installed by default with Ubuntu, if not installed then we will install them using the below command

: sudo apt-get install ufw –y

Enabling the UFW (Firewall)

Below is the command to enable the UFW –

sudo ufw enable

Enabling the Default Policies

As the beginner, we will first configure default policies, which control and handles the traffic which will not match the other rules. By default, the rules will deny all incoming connections and allow all outgoing connections will be allowed which stops someone trying to reach the machine from the internet world.

sudo ufw default deny incoming

sudo ufw default allow outgoing

Enabling SSH Connections

Using the above commands, we have disabled all the incoming connections, it will deny all the incoming connections, we needed to create a rule which will explicitly allow the SSH incoming connection.Below is the command to enable the incoming connection for SSH.

sudo ufw allow ssh

With the above command, the port 22 will be allowed for incoming connections. We can use the below command directly using the port no 22 to allow the SSH connections.

sudo ufw allow 22

However, if we have configured the SSH daemon to use a different port like 2022 or 1022, then we can use the below command

sudo ufw allow 1022

Checking the UFW (Firewall) Status

Below is the command to check the current status of the firewall rules.

sudo ufw status

Enabling the UFW for regular port like (HTTP, HTTPS & FTP)

At this point, we will allow others to connect to the server for the regular ports like HTPP, HTTPS, and FTP ports respectively.

HTTP port 80

sudo ufw allow 80

We can check the UFW (Firewall) status using the below command

sudo ufw status

Like that will use the below command to enable HTTPs and FTP ports (443 and 21) respectively.

sudo ufw allow https

sudo ufw allow ftp

Enabling to Allow Specific Range of Ports

We can also allow or deny particular ranges of ports with UFW to allow the multiple ports instead of allowing single ports. Below is the command to enable a specific range of ports.

sudo ufw allow 500:800/tcp

Enable to Allow specific IP Addresses

If we want to allow a particular machine to allow for all the ports. We can use the below command.

sudo ufw allow from 192.168.100.1

If we want to allow for only specific port we can use the below command.

sudo ufw allow from 192.168.100.1 to any port 8080

If we want to enable the specific subnets like we want to enable for office networks we can use the below command.

sudo ufw allow from 192.168.0.0/24

Deny the Connections or Rules

If we want to deny any ports or network we can use the below commands to deny the connections.

sudo ufw deny http

If we want to deny all the connects from a specific network we can use the below command.

sudo ufw deny from 192.168.2.1

Deleting the Rules

We can delete the rules in two ways one with the actual rules and other with the rules numbers. Actual Rules The rules can be deleted using the actual rule which we allowed using the allow command. Below is the command to delete the HTTP rules from UFW.

sudo ufw allow http

sudo ufw delete allow http

Rules Number We can use the Rules numbers to delete the firewall rules, we can get the list of firewall rules with the below command.

sudo ufw status

numbered If we want to delete the rule 14, then we can use the below command to delete the rules with the below command.

sudo ufw delete 14

EXPERIMENT NO. 9

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

RESULT:

tcpdump:

tcpdump is a packet sniffing and packet analyzing tool for a System Administrator to troubleshoot connectivity issues in Linux. It is used to capture, filter, and analyze network traffic such as TCP/IP packets going through your system. It is many times used as a security tool as well. It saves the captured information in a pcap file, these pcap files can then be opened through Wireshark or through the command tool itself.

Installing tcpdump tool in Linux

Sudo apt install tcpdump

Working with tcpdump command

1. To capture the packets of current network interface

sudo tcpdump

This will capture the packets from the current interface of the network through which the system is connected to the internet.

2. To capture packets from a specific network interface

sudo tcpdump -i wlp2s0

This command will now capture the packets from wlp2s0 network interface.

3) To display all available interfaces

sudo tcpdump -D

wireshark:

Wireshark is a software tool used to monitor the network traffic through a network interface. It is the most widely used network monitoring tool today. Wireshark is loved equally by system administrators, network engineers, network enthusiasts, network security professionals and black hat hackers. The extent of its popularity is such, that experience with Wireshark is considered as a valuable/essential trait in a computer networking related professional.
There are many reasons why Wireshark is so popular :

- It has a great GUI as well as a conventional CLI(T Shark).
- It offers network monitoring on almost all types of network standards (ethernet, wlan, Bluetooth etc)
- It is open source with a large community of backers and developers.
- All the necessary components for monitoring, analysing and documenting the network traffic are present. It is free to use.

Wireshark installation:

sudo apt install wireshark **command is used to install wireshark in linux**

Enter Y to continue

You can give appropriate option according to your need. After this the wireshark will be downloaded to the system

On launching Wireshark, you will see a screen like this:

The basic features of Wireshark are:

1) Packet Monitor: This segment visually shows the packets flowing inside the network. There are colour codes for each type of packets. The packets are shown

with following information :

1. Source address
2. Destination address
3. Packet type
4. Hex dump of the packet
5. Contents of the packet in text
6. Source port(if applicable)
7. Destination port(if applicable)

2) Import from a capture file:

This feature lets you import packets dump from a capture file to analyse further. There are many formats supported by Wireshark, some of them are:

- pcapng
- libpcap
- Oracle snoop and atmsnoop
- Finisar (previously Shomiti) Surveyor captures
- Microsoft Network Monitor captures
- Novell LANalyzer captures
- AIX iptrace captures
- Cinco Networks NetXray captures
- Network Associates Windows-based Sniffer and Sniffer Pro captures
- Network General/Network Associates DOS-based Sniffer (compressed or uncompressed) captures
- AG Group/WildPackets/Savvius EtherPeek/TokenPeek/AiroPeek/EtherHelp/PacketGrabber captures
- RADCOM's WAN/LAN Analyzer captures
- Network Instruments Observer version 9 captures
- Lucent/Ascend router debug output
- HP-UX's nettl

- Toshiba's ISDN routers dump output
- ISDN4BSD i4btrace utility
- Traces from the EyeSDN USB S0
- IPLog format from the Cisco Secure Intrusion Detection System pppd logs (pppdump format)
- the output from VMS's TCPIPtrace/TCPtrace/UCX$TRACE utilities
- the text output from the DBS Etherwatch VMS utility
- Visual Networks' Visual UpTime traffic capture
- the output from CoSine L2 debug
- the output from Accellent's 5Views LAN agents
- Endace Measurement Systems' ERF format captures
- Linux Bluez Bluetooth stack hcidump -w traces
- Catapult DCT2000 .out files
- Gammu generated text output from Nokia DCT3 phones in Netmonitor mode
- IBM Series (OS/400) Comm traces (ASCII & UNICODE)
- Juniper Netscreen snoop captures
- Symbian OS btsnoop captures
- Tamosoft CommView captures
- Textronix K12xx 32bit .rf5 format captures
- Textronix K12 text file format captures
- Apple PacketLogger captures
- Captures from Aethra Telecommunications' PC108 software

3) Export to a capture file: Wireshark lets you save the results as a capture file to continue working on them at later point of time. The supported formats are:

- pcapng (*.pcapng)

- libpcap, tcpdump and various other tools using tcpdump's capture format (*.pcap, *.cap, *.dmp)
- Accellent 5Views (*.5vw)
- HP-UX's nettl (*.TRC0, *.TRC1)
- Microsoft Network Monitor – NetMon (*.cap)
- Network Associates Sniffer – DOS (*.cap, *.enc, *.trc, *fdc, *.syc)
- Network Associates Sniffer – Windows (*.cap)
- Network Instruments Observer version 9 (*.bfr)
- Novell LANalyzer (*.tr1)
- Oracle (previously Sun) snoop (*.snoop, *.cap)
- Visual Networks Visual UpTime traffic (*.*).

Netcat:

Netcat (or nc in short) is a simple yet powerful **networking command-line tool** used for performing any operation in Linux related to TCP, UDP, or UNIX-domain sockets.

Netcat can be used for port scanning, port redirection, as a port listener (for incoming connections); it can also be used to open remote connections and so many other things. Besides, you can use it as a backdoor to gain access to a target server.

Installing netcat on linux:

```
sudo apt-get install netcat
```

Port scanning:

Netcat can be used for port scanning: to know **which ports are open** and running services on a target machine. It can scan a single or multiple or a range of open ports.

The -z option sets nc to simply scan for listening daemons, without actually sending any data to them. The -v option enables verbose mode and -w specifies a timeout for connection that can not be established.

Syntax:

```
nc -vz IP_address port
```

Connection timed out:

A *connection timed out* response indicates that your connection is not working, which could mean your firewall is blocking the port. Test the connection status by adding a rule that accepts connections on the required port.

Connection succeeded

If the initial connection succeeds, Netcat can connect to the service. Look at the connection in more detail.

Syntax:

```
nc -vt IP Address Port
```

Closing the connection

You can terminate the connection by either pressing Ctrl-C or type the service-specific quit command.

EXPERIMENT NO. 10

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

RESULT:

Hypervisor:

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.

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

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, while 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.

Benefits of hypervisors

There are several benefits to using a hypervisor that hosts multiple virtual machines:

- **Speed:** Hypervisors allow virtual machines to be created instantly, unlike bare-metal servers. This makes it easier to provision resources as needed for dynamic workloads.
- **Efficiency:** Hypervisors that run several virtual machines on one physical machine's resources also allow for more efficient utilization of one physical server. It is more cost- and energy-efficient to run several virtual machines on one physical machine than to run multiple underutilized physical machines for the same task.
- **Flexibility:** Bare-metal hypervisors allow operating systems and their associated applications to run on a variety of hardware types because the hypervisor separates the OS from the underlying hardware, so the software no longer relies on specific hardware devices or drivers.
- **Portability:** Hypervisors allow multiple operating systems to reside on the same physical server (host machine). Because the virtual machines that the hypervisor runs are independent from the physical machine, they are portable. IT teams can shift workloads and allocate networking, memory, storage and processing resources across multiple servers as needed, moving from machine to machine or platform to platform. When an application needs more processing power, the virtualization software allows it to seamlessly access additional machines.

Virtual mechine:

A **Virtual Machine** (VM) is a compute resource that uses software instead of a physical computer to run programs and deploy apps. One or more virtual "guest" machines run on a physical "host" machine. Each virtual machine runs its own operating system and functions separately from the other VMs, even when they are all running on the same host. This means that, for example, a virtual MacOS virtual machine can run on a physical PC.

Virtual machine technology is used for many use cases across on-premises and cloud environments. More recently, public cloud services are using virtual machines to provide virtual application resources to multiple users at once, for even more cost efficient and flexible compute.

What are virtual machines used for?

Virtual machines (VMs) allow a business to run an operating system that behaves like a completely separate computer in an app window on a desktop. VMs may be deployed to accommodate different levels of processing power needs, to run software that requires a different operating system, or to test applications in a safe, sandboxed environment.

Virtual machines have historically been used for server virtualization, which enables IT teams to consolidate their computing resources and improve efficiency. Additionally, virtual machines can perform specific tasks considered too risky to carry out in a host environment, such as accessing virus-infected data or testing operating systems. Since the virtual machine is separated from the rest of the system, the software inside the virtual machine cannot tamper with the host computer.

Advantages of virtual machines

Virtual machines are easy to manage and maintain, and they offer several advantages over physical machines:

- VMs can run multiple operating system environments on a single physical computer, saving physical space, time and management costs.
- Virtual machines support legacy applications, reducing the cost of migrating to a new operating system. For example, a Linux virtual machine running a distribution of Linux as the guest operating system can exist on a host server that is running a non-Linux operating system, such as Windows.
- VMs can also provide integrated disaster recovery and application provisioning options.

KVM:

KVM stands for Kernel-based Virtual Machine. It allows the kernel to operate as a hypervisor. Moreover, it requires a processor with hardware virtualization extensions such as Intel VT or AMD-V. KVM was initially designed for x86 processors. Later, it was ported to processors such as ARM, PowerPC etc. Operating systems such as FreeBSD and illumos contain KVM as loadable kernel modules. Also, KVM provides hardware-assisted virtualization for many guest operating systems such as Linux, Solaris, Windows, Haiku and OS X. Furthermore, Android 2.2, Darwin 8.-.1 etc. work with some limitations.

Furthermore, some graphical management tools having KVM are as follows.

Kimchi is KVM's web-based virtualization management tool

Virtual Machine Manager allows creating, editing, starting and stopping KVM based virtual machine

OpenQRM allows managing and controlling various data centre infrastructures.

GNOME Boxes – Gnome interface for handling libvirt guests on Linux.

oVirt is an open source virtualization management tool for KVM

Proxmox Virtual Environment is an open source virtualization management package with KVM and LXC.

Xen:

Xen or Xen Project is a type 1 hypervisor. It provides services to allow multiple computer operating systems to execute on the same computer hardware simultaneously.

In brief, KVM and Xen are two hypervisors written in C language. The main difference between KVM and Xen is that KVM is a virtualization module in Linux kernel that works similar to a hypervisor while Xen is a type 1 hypervisor that allows multiple operating systems to execute on the same computer hardware simultaneously.

Docker Container:

A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.

Container images become containers at runtime and in the case of Docker containers - images become containers when they run on Docker Engine. Available for both Linux and Windows-based applications, containerized software will always run

the same, regardless of the infrastructure. Containers isolate software from its environment and ensure that it works uniformly despite differences for instance between development and staging.

Docker containers that run on Docker Engine:

- Standard: Docker created the industry standard for containers, so they could be portable anywhere
- Lightweight: Containers share the machine's OS system kernel and therefore do not require an OS per application, driving higher server efficiencies and reducing server and licensing costs
- Secure: Applications are safer in containers and Docker provides the strongest default isolation capabilities in the industry

Installing Docker:

1) Updating the local repository

```
sudo apt update
```

2) Installing Docker

```
sudo apt install docker.io
```

Type y and hit Enter to confirm the installation. Once the install is completed, the output notifies you Docker has been installed.

3) Checking docker installation

```
docker --version
```

4) Starting docker service

* Start the Docker service by running:

```
sudo systemctl start docker
```

* Then, enable it to run at startup:

```
sudo systemctl enable docker
```

*To check the status of the service, run:

```
sudo systemctl status docker
```

The output should verify Docker is active (running).

Use Docker on Ubuntu 20.04

The basic syntax for docker commands is:

```
sudo docker [option] [command] [argument]
```

Working With Docker Images

Docker images are files that contain the source code, libraries, dependencies, tools, and other files a container needs. You can create Docker images with Dockerfiles or use existing ones available on Docker Hub.

To download a new Docker image, use the command:

```
docker pull [image_name]
```

If you don’t know the exact name of the image, search for it in Docker’s repository with:

```
docker search ubuntu
```

After working with Docker for some time, you will collect a local registry of images. Display a list of all Docker images on the system with:

```
docker images
```

Working With Docker Containers

Docker containers are isolated virtual environments that run based on the Docker image assigned to them.

To run a container based on an existing Docker image, use the command:

```
docker run [image_name]
```

Using the command above runs a container but doesn't move you inside of it. To run a container in interactive mode and change to the container command prompt, run:

```
docker run -it [image_name]
```

Another useful docker command is listing all the containers on the system. To list all active containers, type:

```
docker container ps
```

To view all containers (active and inactive), run:

```
docker container ps -a
```

Experiment No :11

Aim : Automation using Ansible: Spin up a new Linux VM using Ansible playbook.

Ansible

Ansible is an **open-source IT engine** that automates application deployment, cloud provisioning, intra service orchestration, and other IT tools.

It is an **automation and orchestration tool** popular for the following reasons:

- Simple to Install.
- Free and open source.
- Lightweight and consistent.
- OpenSSH security features make it very secure.

Ansible Concepts

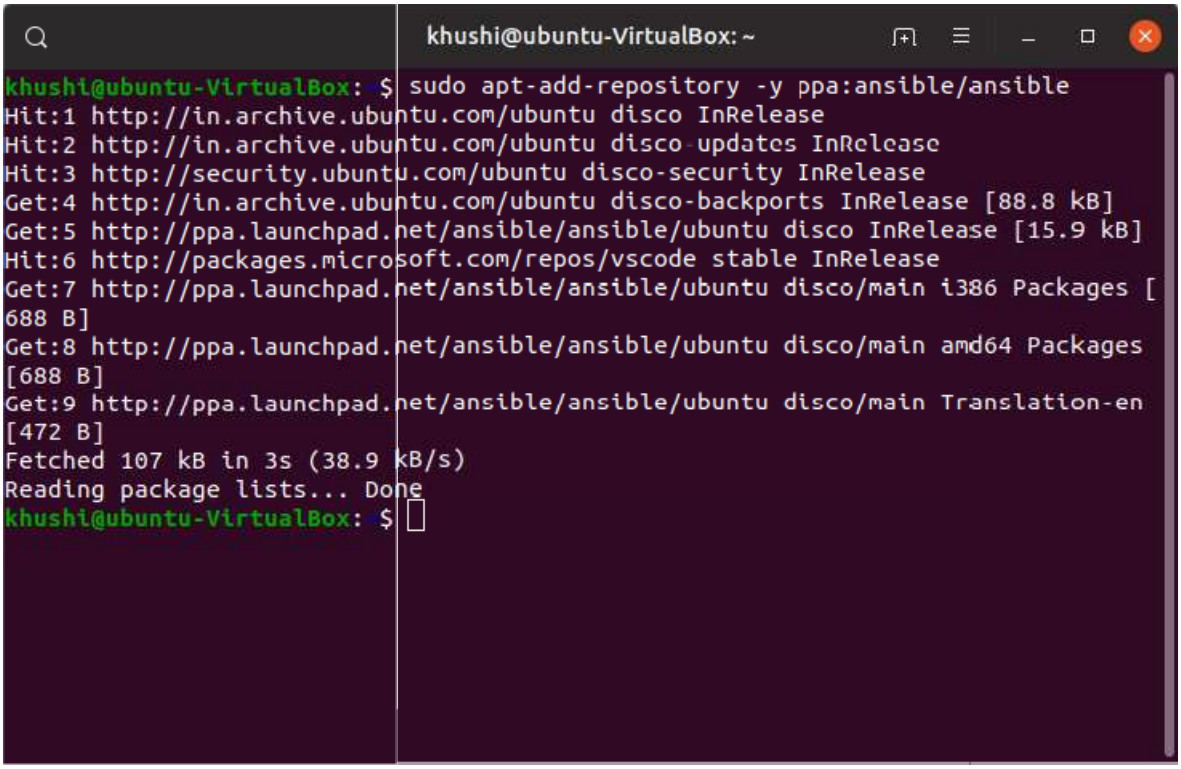
1. **Control node:**
   Commands and Playbooks can run by invoking /usr/bin/ansible or /usr/bin/ansible-playbook, from any control node. You can use any computer that has Python installed on it as a control node. However, one can not use a computer with Windows OS as a control node. One can have multiple control nodes.
2. **Managed nodes:**
   Also sometimes called “hosts”, Managed nodes are the network devices (and/or servers) you manage with Ansible.
3. **Inventory:**
   Also sometimes called “hostfile”, Inventory is the list of Managed nodes use to organize them. It is also used for creating and nesting groups for easier scaling.

4. **Modules:**
   These are the units of code executed by Ansible. Each module can be used for a specific purpose. One can invoke a single module with a task, or invoke several different modules in a playbook.
5. **Tasks:**
   The units of action in Ansible. One can execute a single task once with an ad-hoc command.
6. **Playbooks3:**
   These are the ordered list of tasks that are saved so you can run those tasks in that order repeatedly. Playbooks are written in YAML and are easy to read, write, share and understand.

Installation

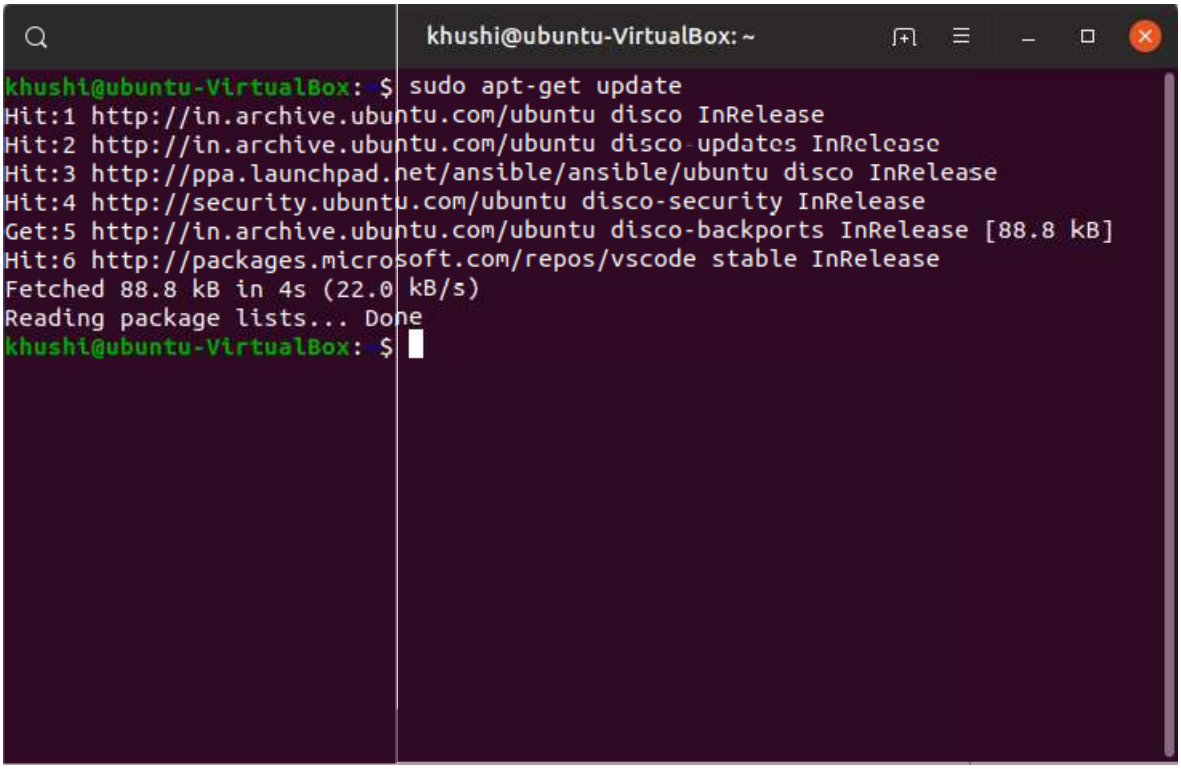
Step 1:Add the Ansible Repository.

```
sudo apt-add-repository -y ppa:ansible/ansible
```



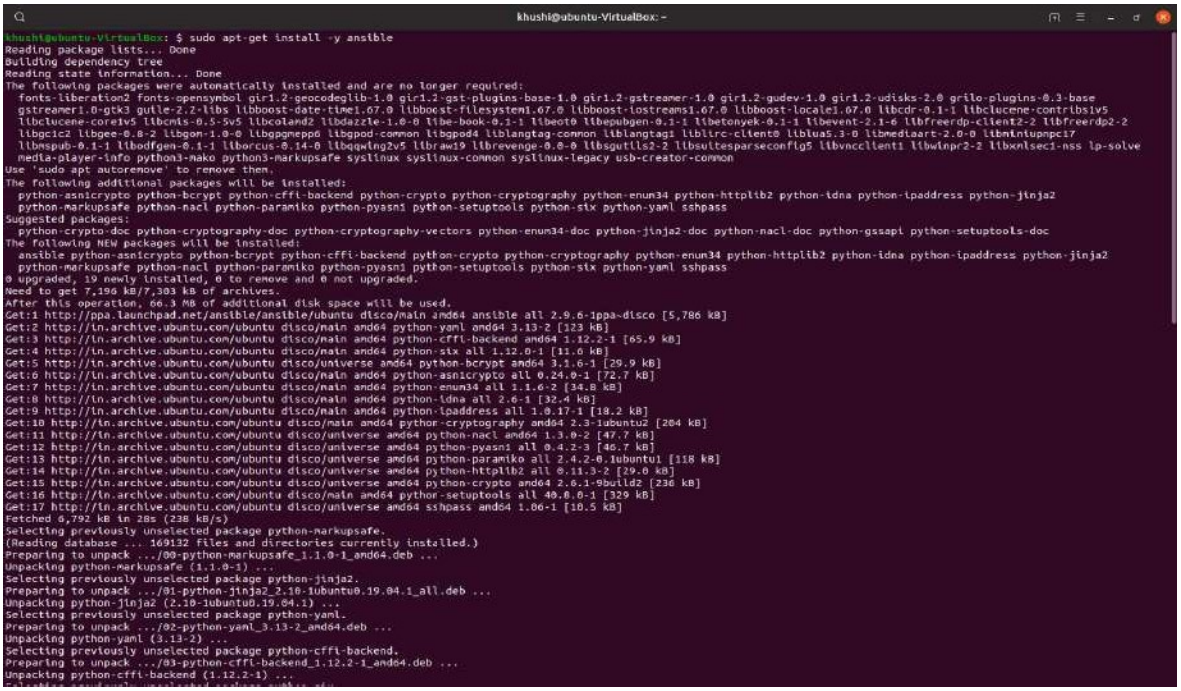
Step 2: Update the system repository listings.

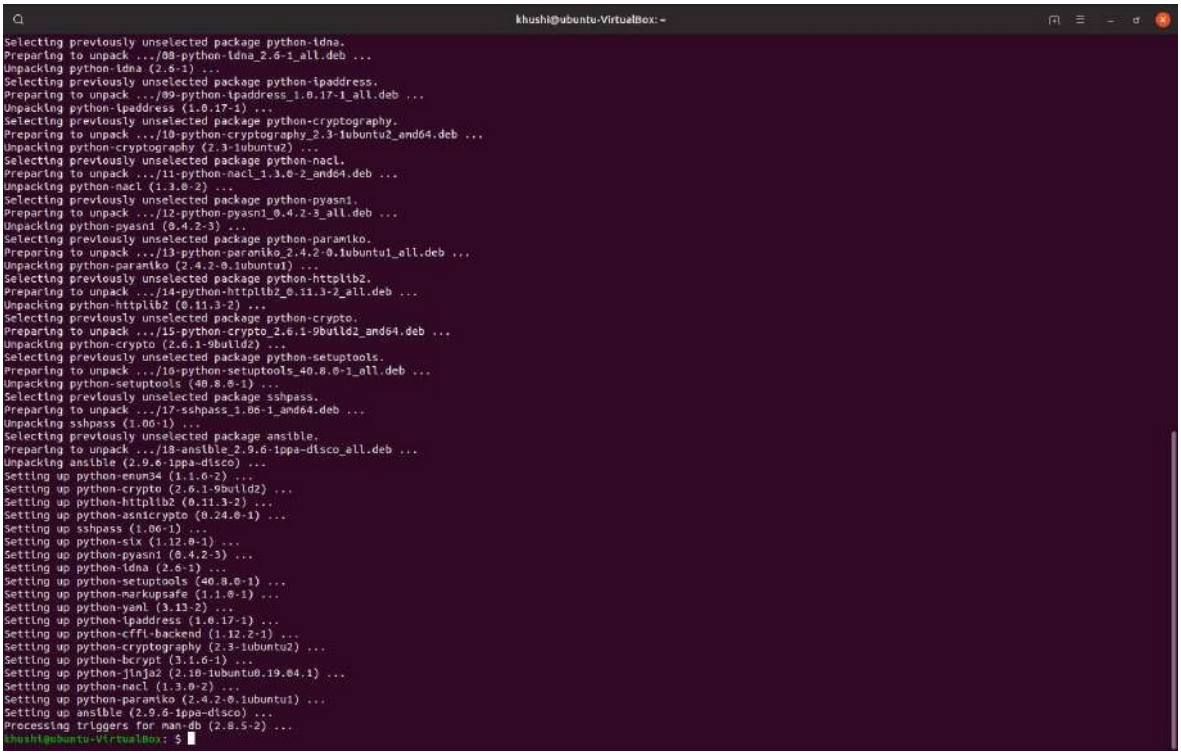
sudo apt-get update



Step 3: Install the ansible packages.

sudo apt-get install -y ansible

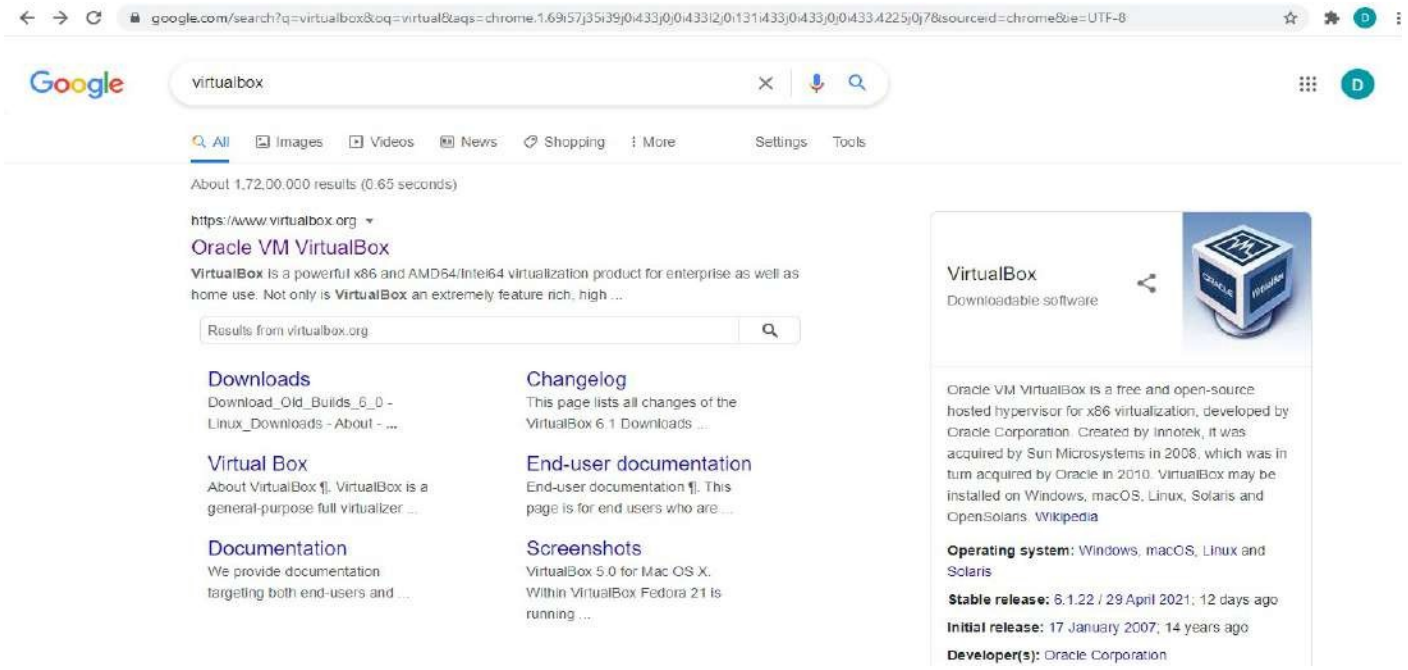



khushi@ubuntu-VirtualBox: ~
Selecting previously unselected package python-idna.
Preparing to unpack .../08-python-idna_2.6-1_all.deb ...
Unpacking python-idna (2.6-1) ...
Selecting previously unselected package python-ipaddress.
Preparing to unpack .../09-python-ipaddress_1.0.17-1_all.deb ...
Unpacking python-ipaddress (1.0.17-1) ...
Selecting previously unselected package python-cryptography.
Preparing to unpack .../10-python-cryptography_2.3-1ubuntu2_amd64.deb ...
Unpacking python-cryptography (2.3-1ubuntu2) ...
Selecting previously unselected package python-nacl.
Preparing to unpack .../11-python-nacl_1.3.0-2_amd64.deb ...
Unpacking python-nacl (1.3.0-2) ...
Selecting previously unselected package python-pyasn1.
Preparing to unpack .../12-python-pyasn1_0.4.2-3_all.deb ...
Unpacking python-pyasn1 (0.4.2-3) ...
Selecting previously unselected package python-paramiko.
Preparing to unpack .../13-python-paramiko_2.4.2-0.1ubuntu1_all.deb ...
Unpacking python-paramiko (2.4.2-0.1ubuntu1) ...
Selecting previously unselected package python-httplib2.
Preparing to unpack .../14-python-httplib2_0.11.3-2_all.deb ...
Unpacking python-httplib2 (0.11.3-2) ...
Selecting previously unselected package python-crypto.
Preparing to unpack .../15-python-crypto_2.6.1-9build2_amd64.deb ...
Unpacking python-crypto (2.6.1-9build2) ...
Selecting previously unselected package python-setuptools.
Preparing to unpack .../16-python-setuptools_40.8.0-1_all.deb ...
Unpacking python-setuptools (40.8.0-1) ...
Selecting previously unselected package sshpass.
Preparing to unpack .../17-sshpass_1.06-1_amd64.deb ...
Unpacking sshpass (1.06-1) ...
Selecting previously unselected package ansible.
Preparing to unpack .../18-ansible_2.9.6-1ppa~disco_all.deb ...
Unpacking ansible (2.9.6-1ppa~disco) ...
Setting up python-enum34 (1.1.6-2) ...
Setting up python-crypto (2.6.1-9build2) ...
Setting up python-httplib2 (0.11.3-2) ...
Setting up python-asn1crypto (0.24.0-1) ...
Setting up sshpass (1.06-1) ...
Setting up python-six (1.12.0-1) ...
Setting up python-pyasn1 (0.4.2-3) ...
Setting up python-idna (2.6-1) ...
Setting up python-setuptools (40.8.0-1) ...
Setting up python-markupsafe (1.1.0-1) ...
Setting up python-yaml (3.13-2) ...
Setting up python-ipaddress (1.0.17-1) ...
Setting up python-cffi-backend (1.12.2-1) ...
Setting up python-cryptography (2.3-1ubuntu2) ...
Setting up python-bcrypt (3.1.6-1) ...
Setting up python-jinja2 (2.10-1ubuntu0.19.04.1) ...
Setting up python-nacl (1.3.0-2) ...
Setting up python-paramiko (2.4.2-0.1ubuntu1) ...
Setting up ansible (2.9.6-1ppa~disco) ...
Processing triggers for man-db (2.8.5-2) ...
khushi@ubuntu-VirtualBox: $

Aim : Installation of ubuntu with virtual box.

Step 1 :-

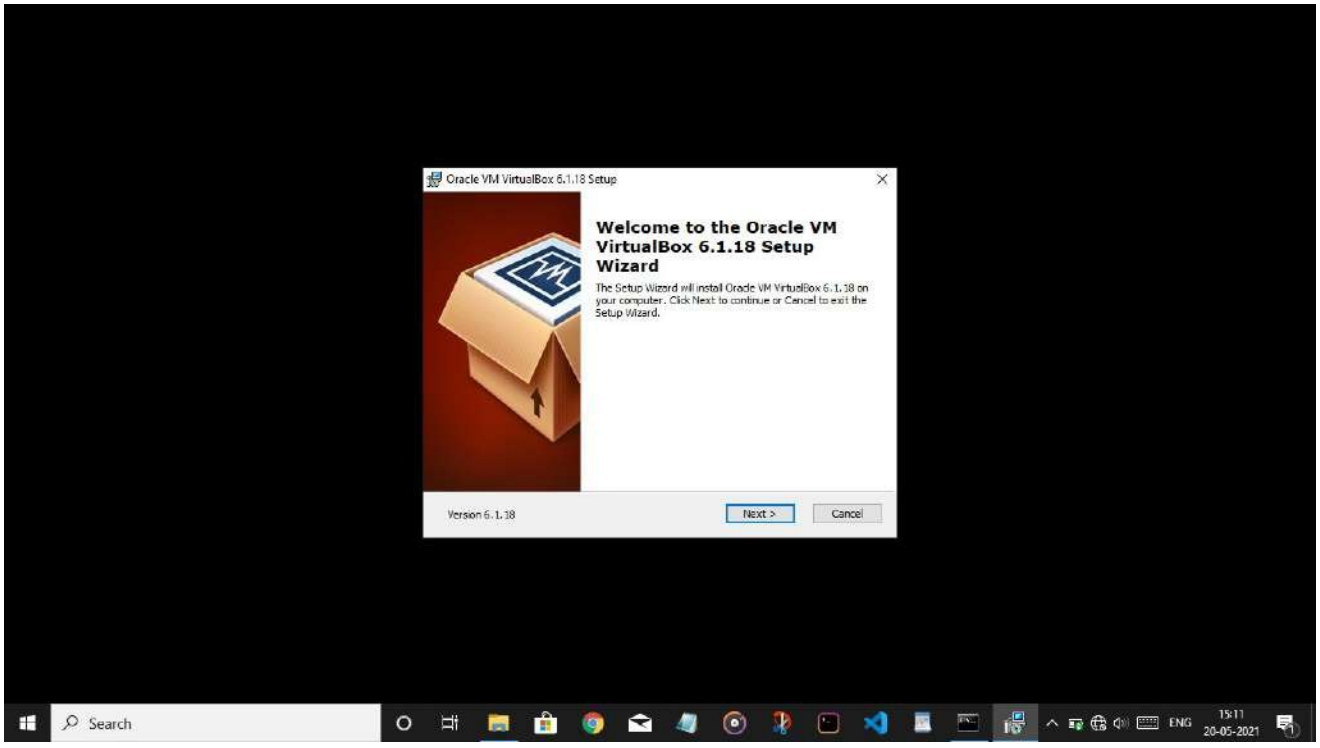
Download virtual box using browser.

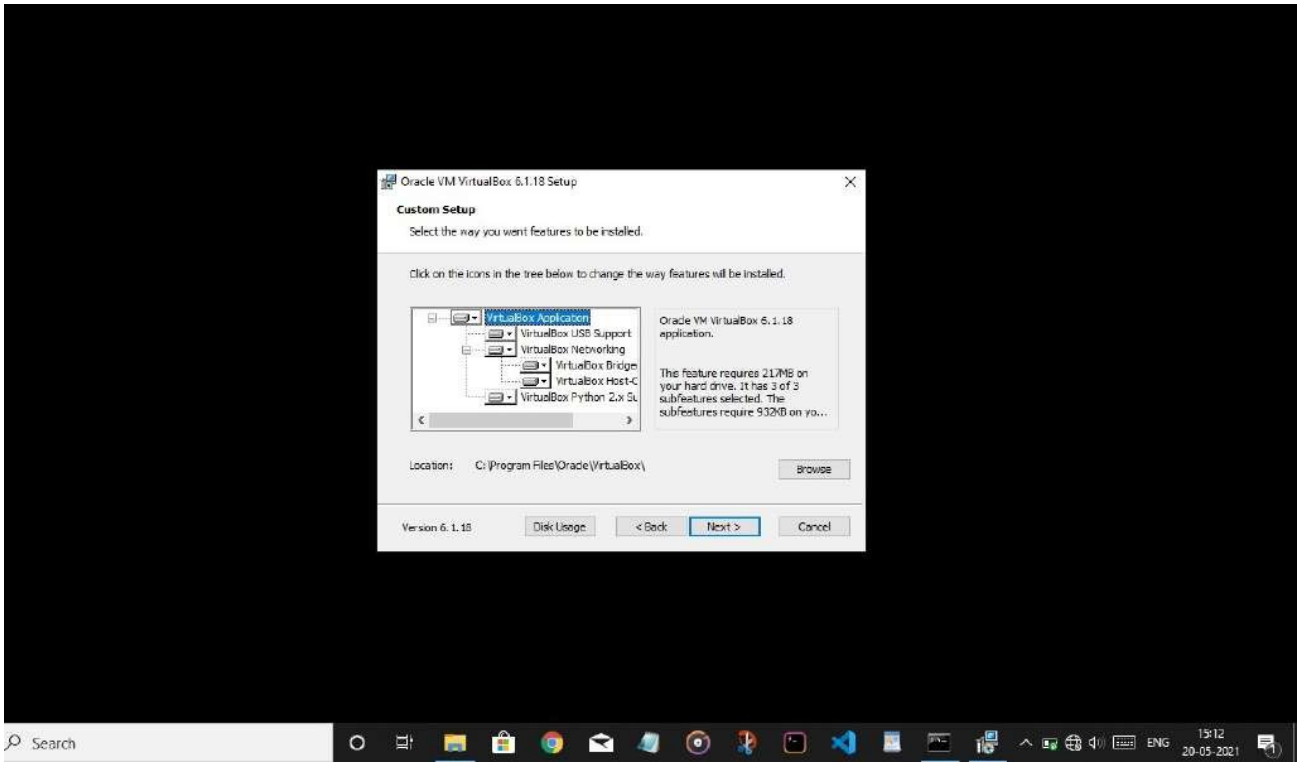


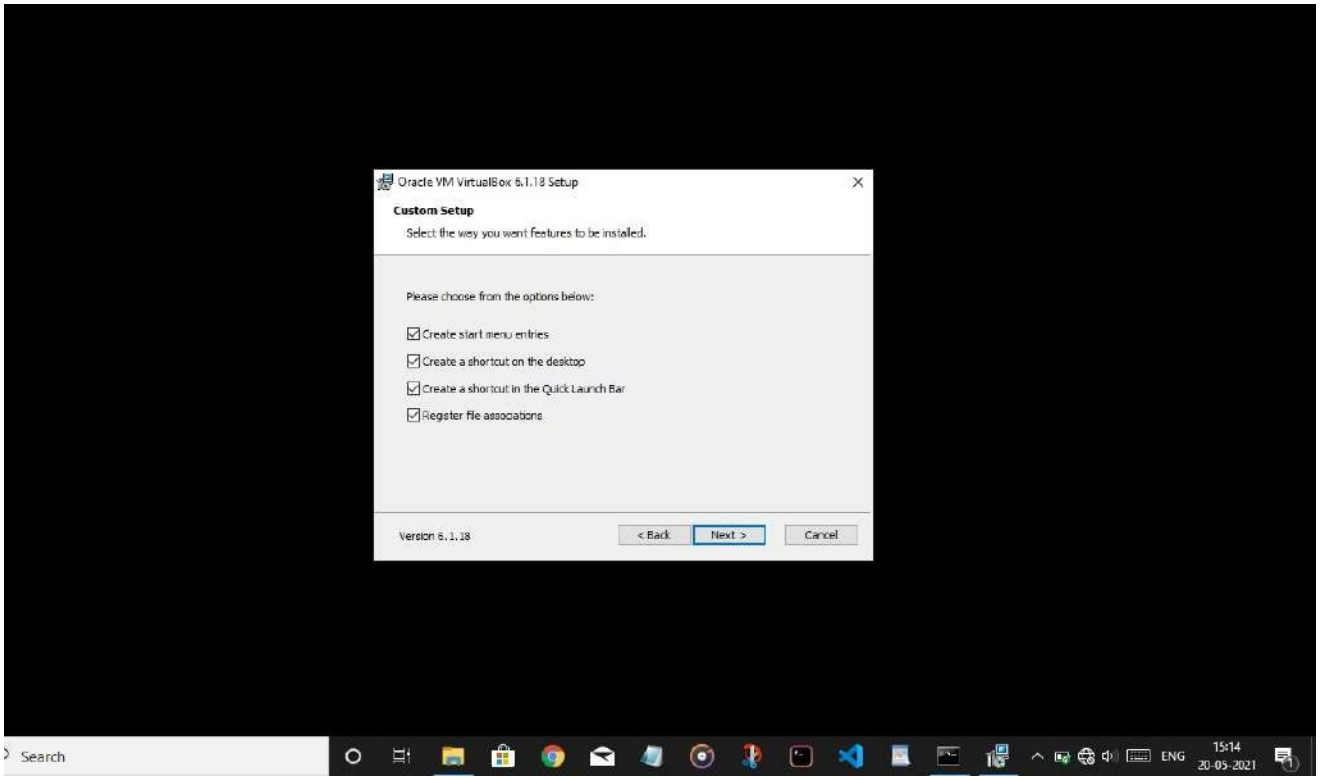
1.1 Click on www.virtualbox.org and proceed to download.

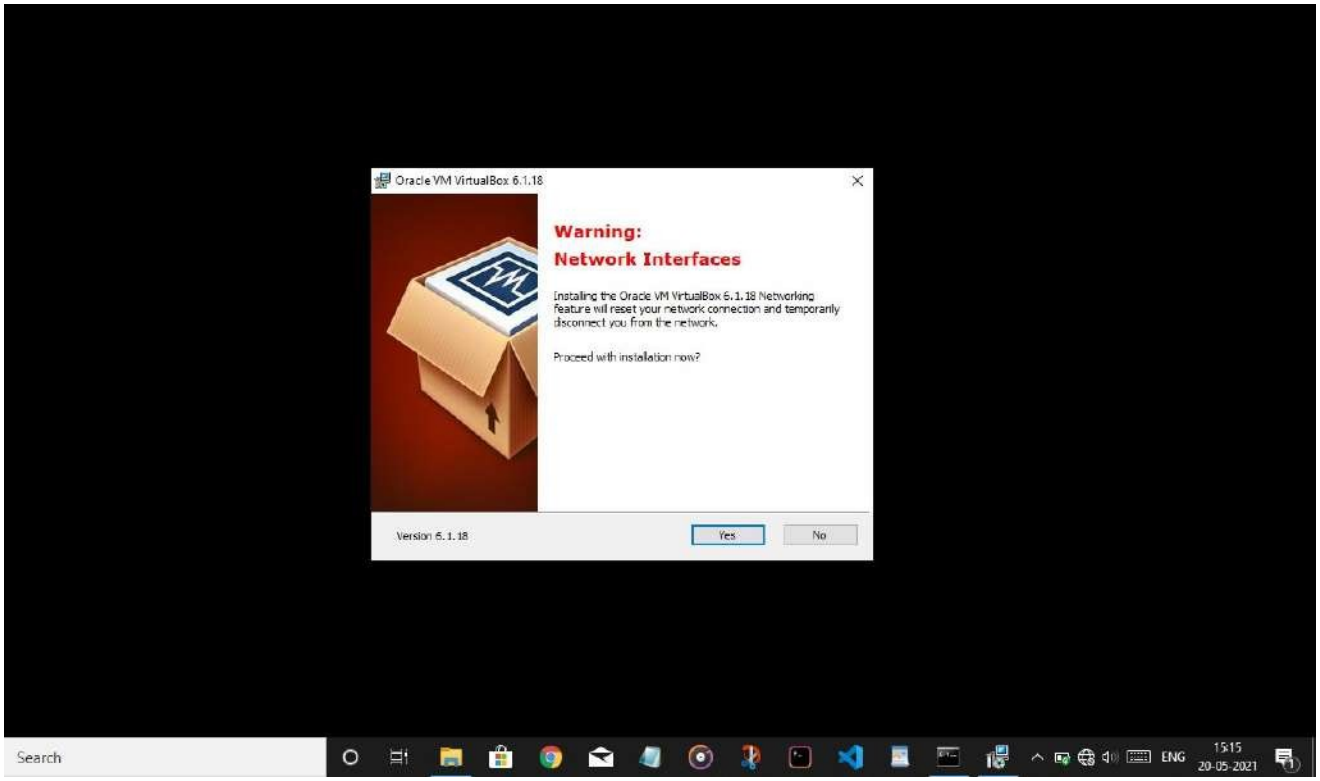
Step 2:

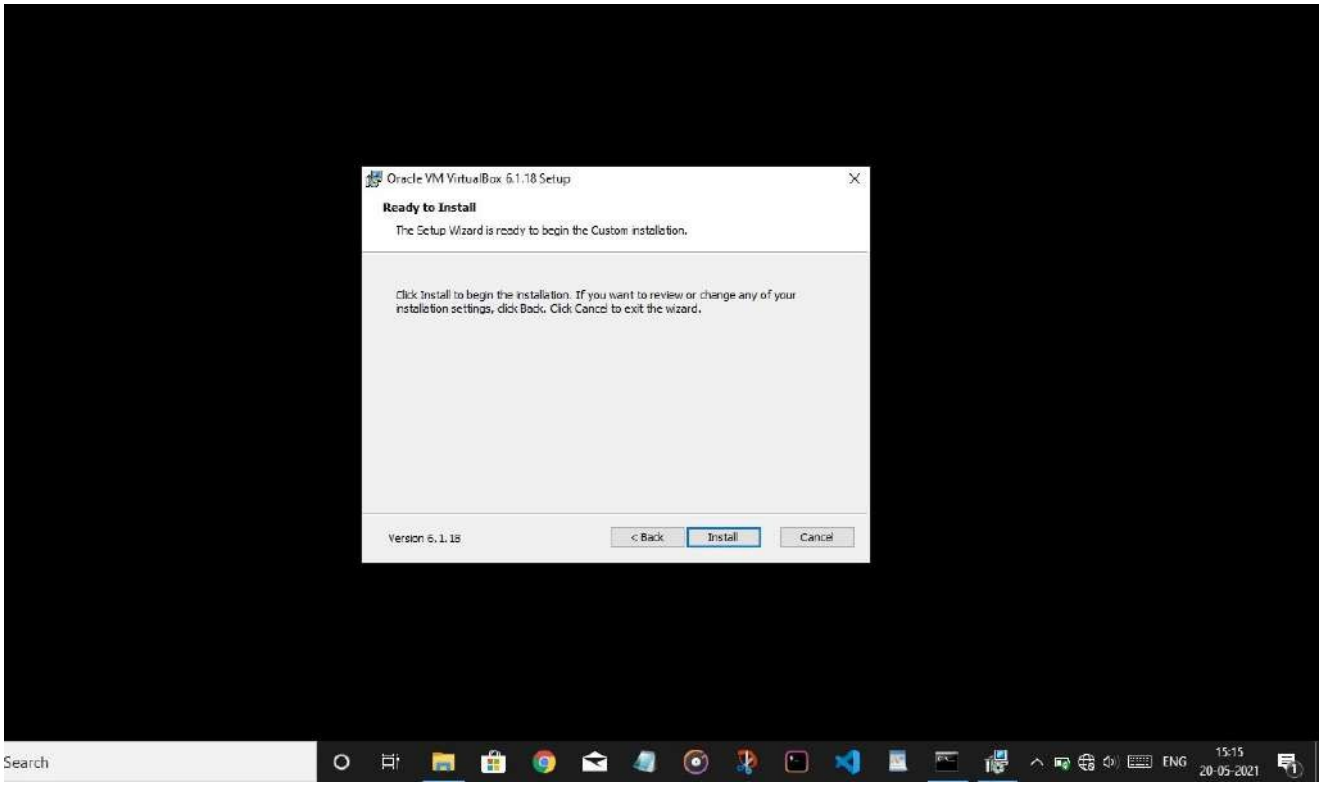
Installation of virtual box by clicking on ‘Next’,until installation get completed.

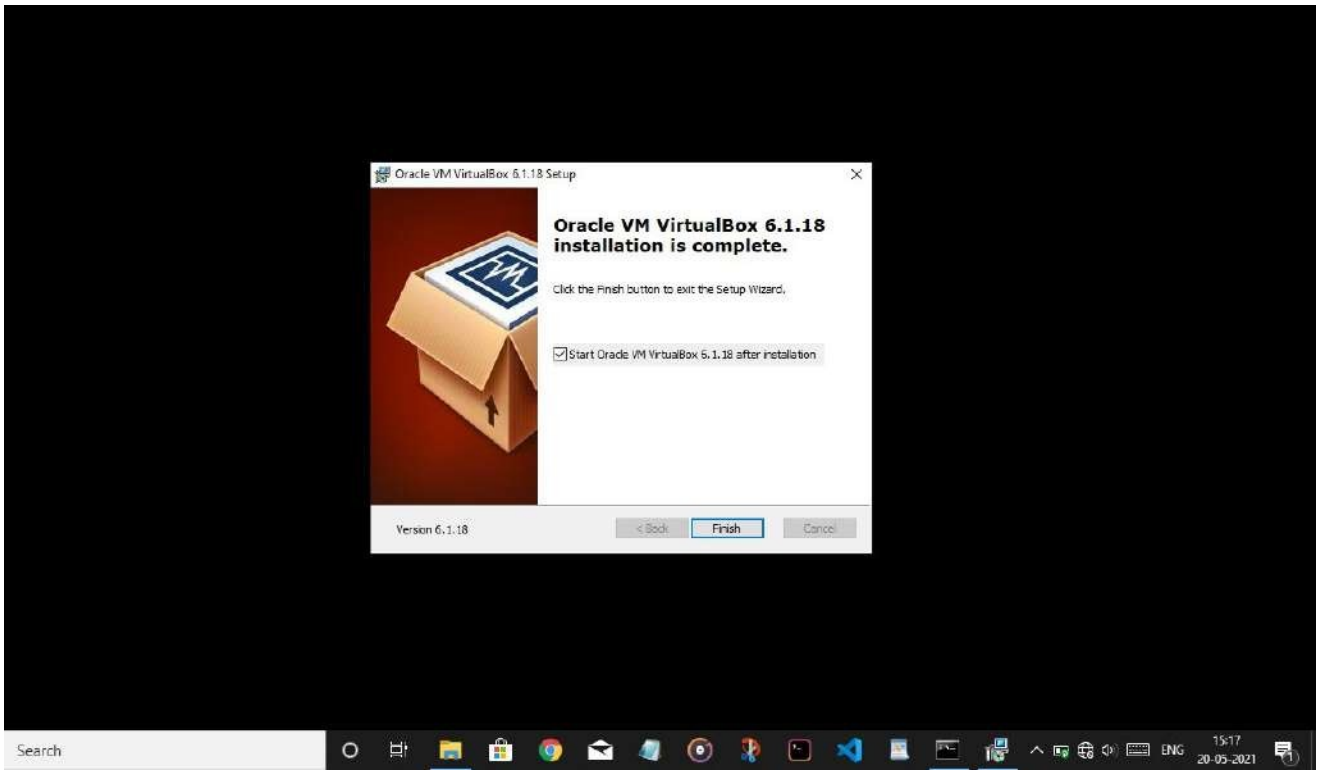




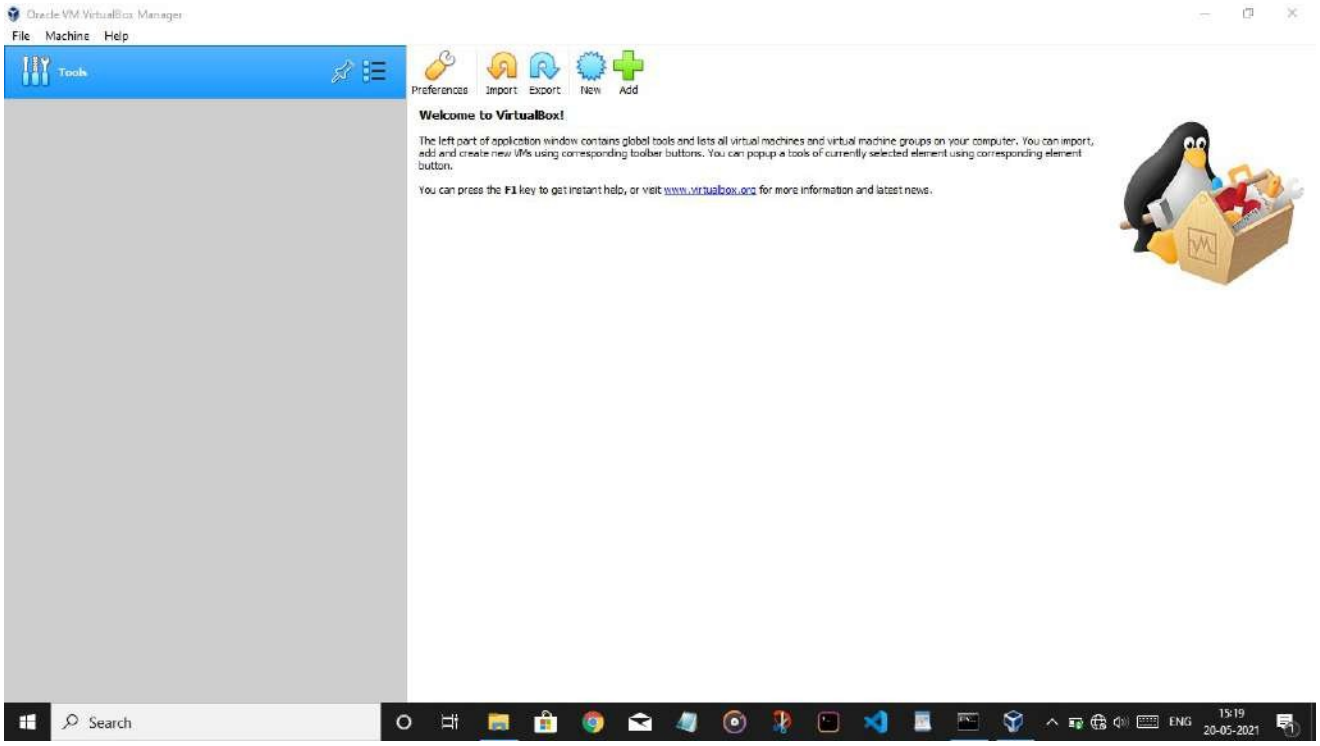

Oracle VM VirtualBox 6.1.18 Setup
Custom Setup
Select the way you want features to be installed.
Please choose from the options below:
Create start menu entries
Create a shortcut on the desktop
Create a shortcut in the Quick Launch Bar
Register file associations
Version 6.1.18
< Back
Next >
Cancel
Search
ENG
15:14
20-05-2021


Oracle VM VirtualBox 6.1.18
Warning:
Network Interfaces
Installing the Oracle VM VirtualBox 6.1.18 Networking feature will reset your network connection and temporarily disconnect you from the network.
Proceed with installation now?
Version 6.1.18
Yes
No
Search
ENG
15:15
20-05-2021


Oracle VM VirtualBox 6.1.18 Setup
Ready to Install
The Setup Wizard is ready to begin the Custom installation.
Click Install to begin the installation. If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.
Version 6.1.18
< Back
Install
Cancel
Search
ENG
15:15
20-05-2021

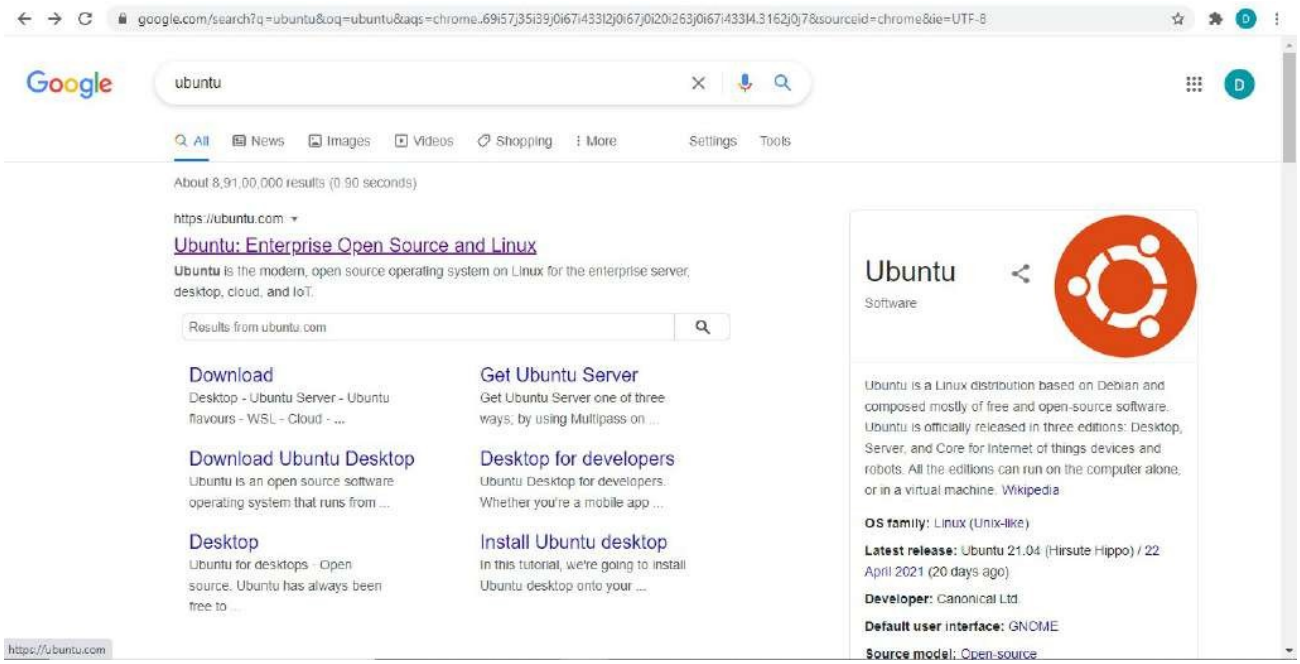

Oracle VM VirtualBox 6.1.18 Setup
Oracle VM VirtualBox 6.1.18 installation is complete.
Click the Finish button to exit the Setup Wizard.
Start Oracle VM VirtualBox 6.1.18 after installation
Version 6.1.18
< Back
Finish
Cancel
Search
ENG
15:17
20-05-2021

Virtual box installation completed.

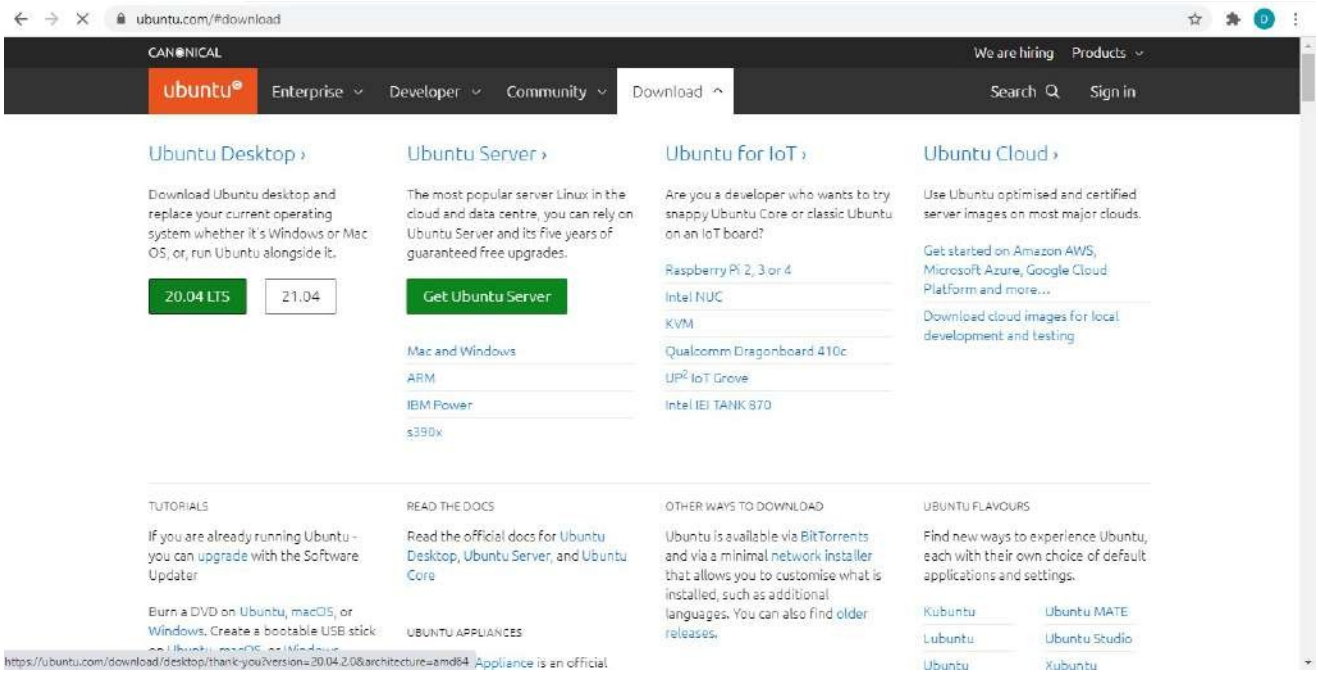


Step 3 :

Download ubuntu latest version from browser.



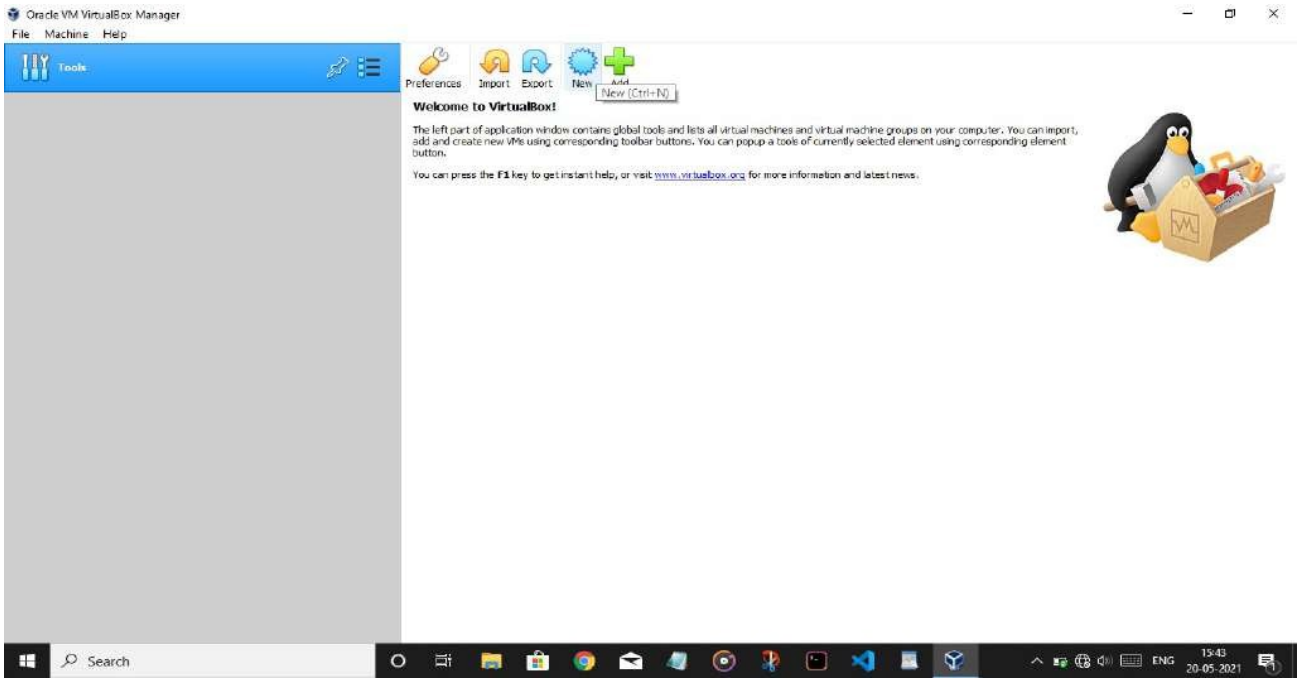
Click on ubuntu.com



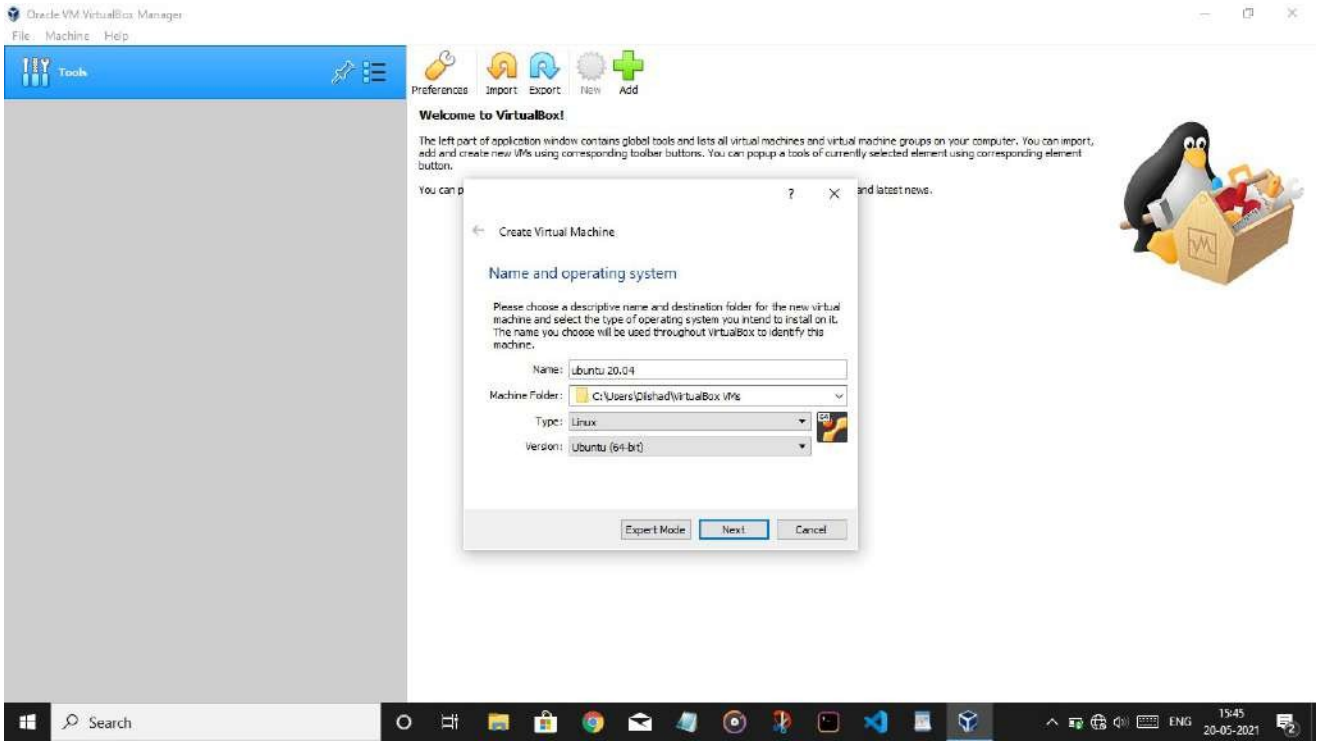
Click on Download and select 20.04 LTS version of ubuntu.

Step 4 :

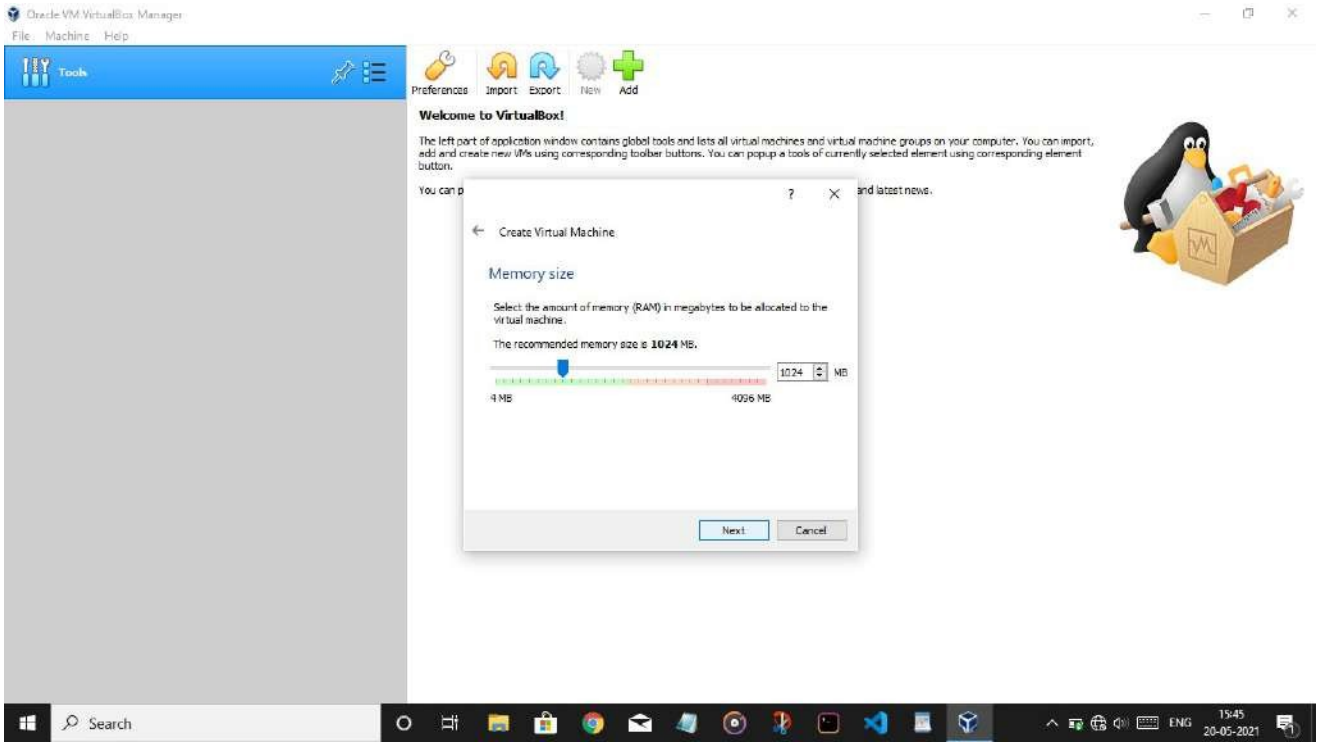
After completing ubuntu download, open virtual box and click on ‘New’ on top.



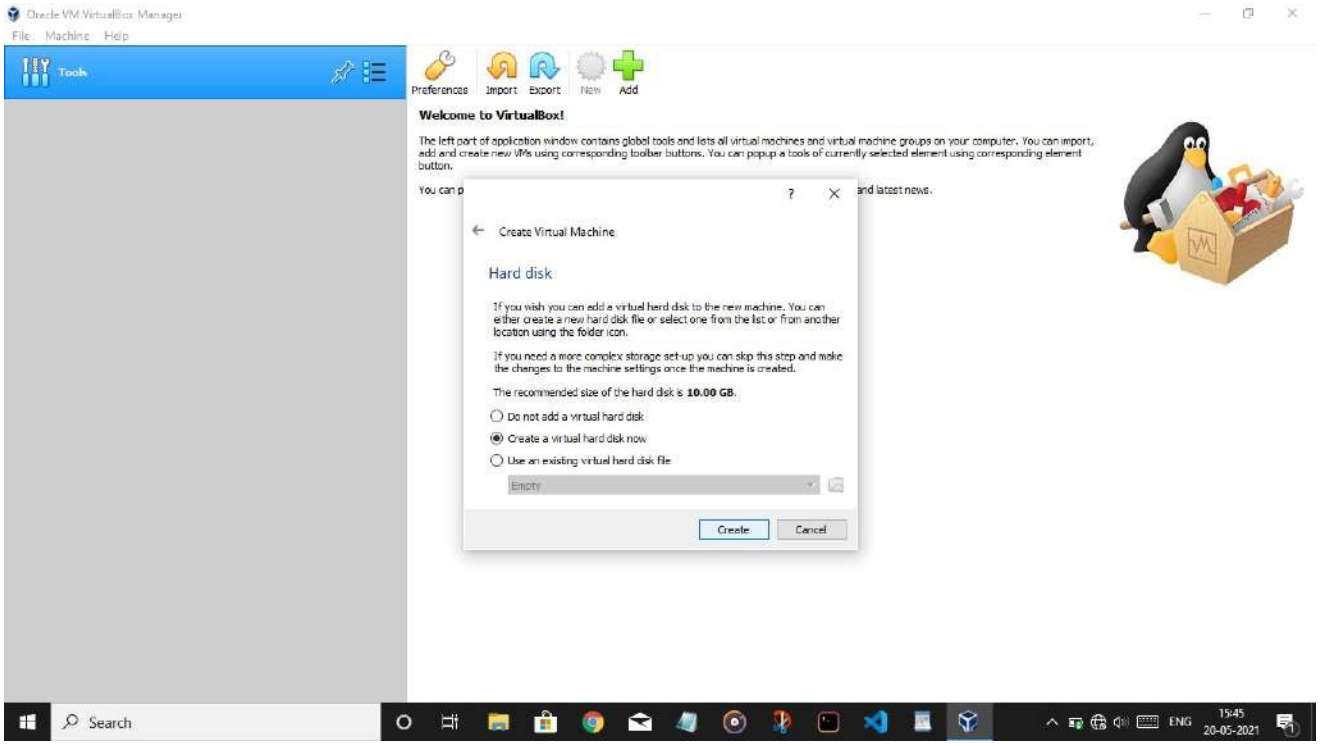
Give the details and click ‘Next’.



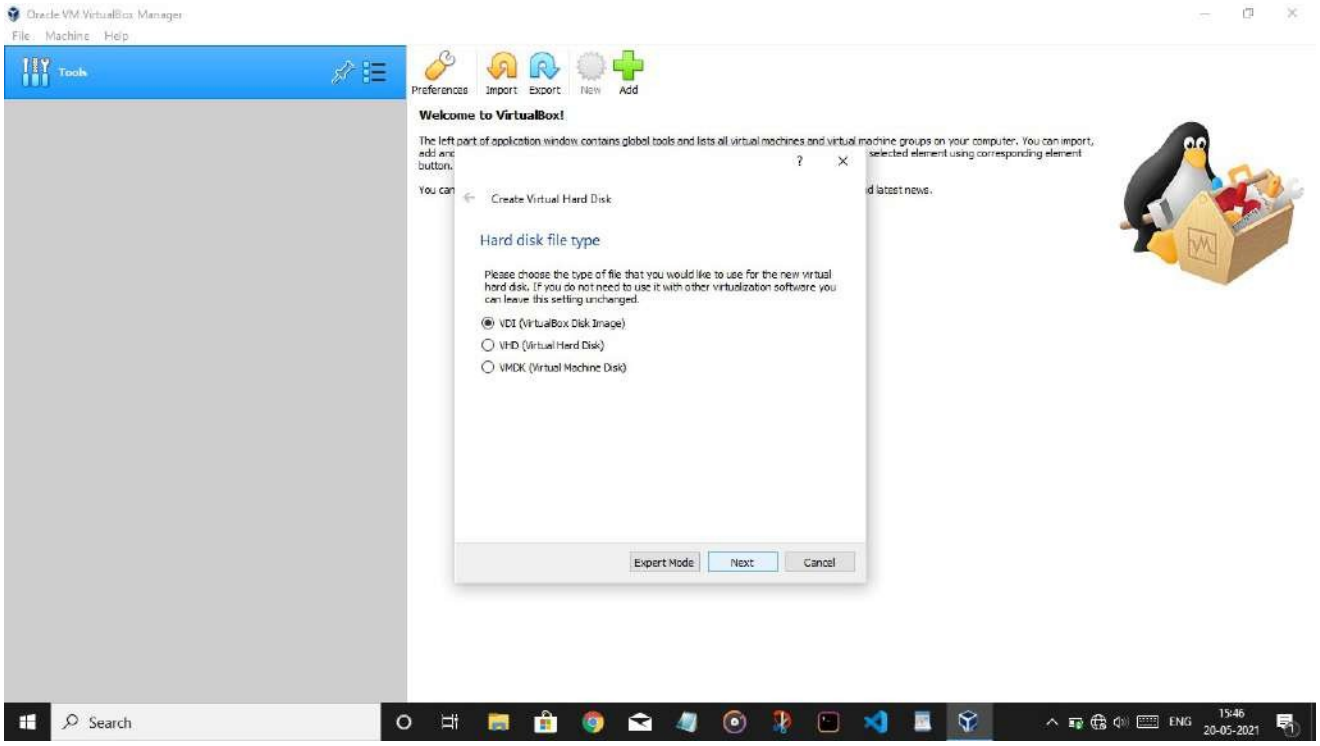
Select amount of memory [RAM] to be allocated to virtual machine.



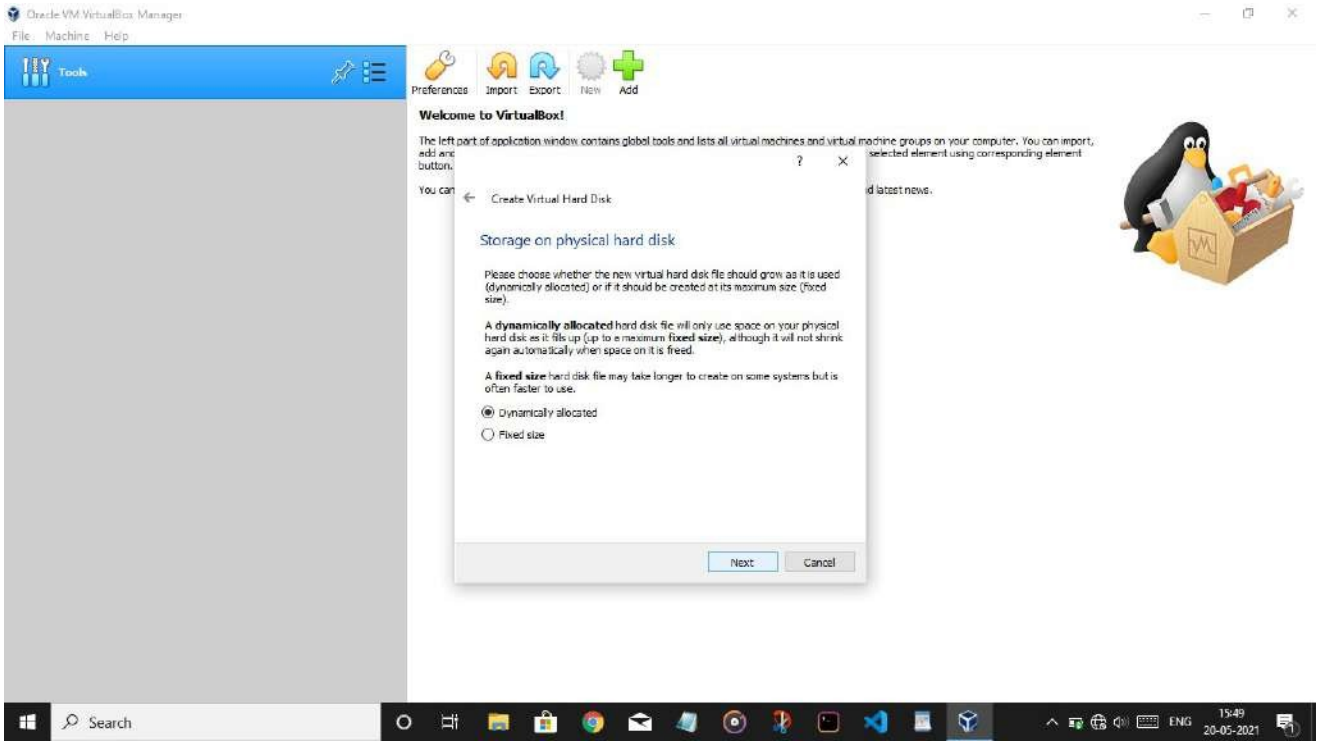
select the option and click on ‘create’.

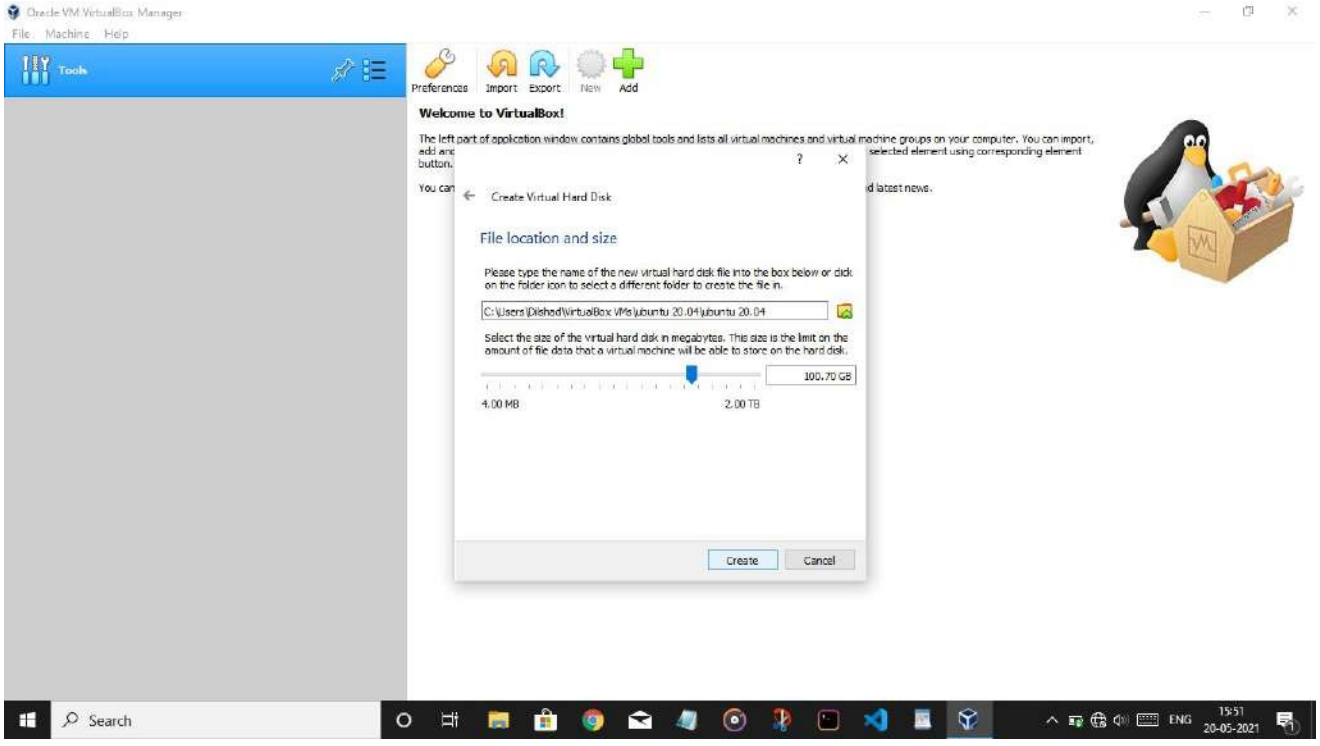


select VDI and click ‘Next’.



Allocate memory space [hard disk] for virtual machine.

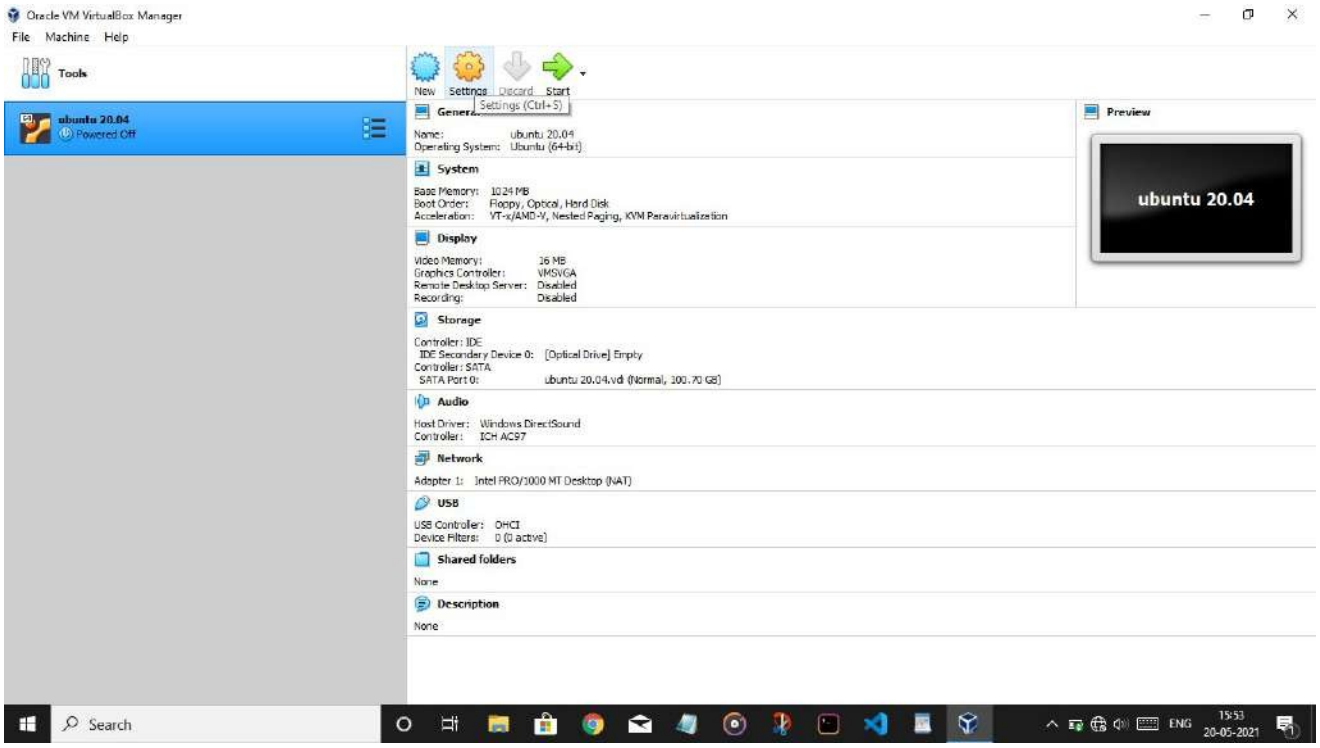




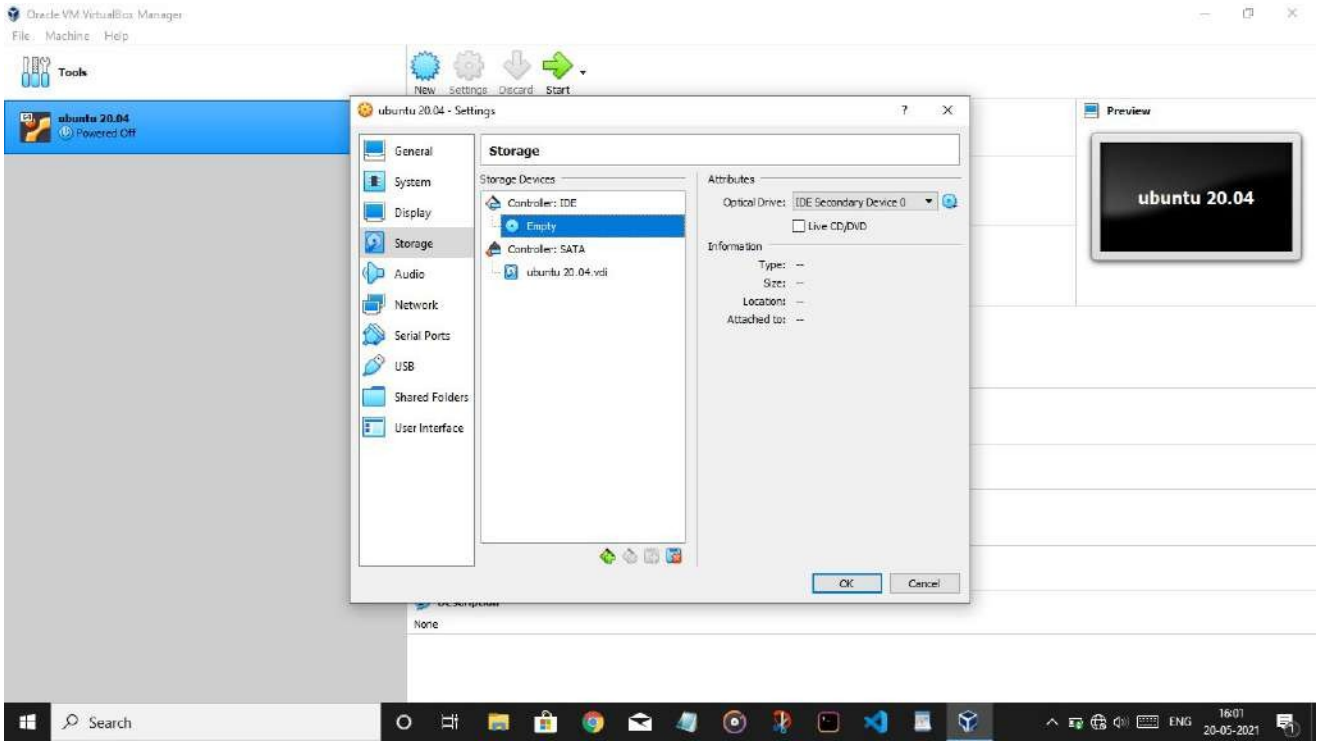
Memory allocation successful.

Step 4 :-

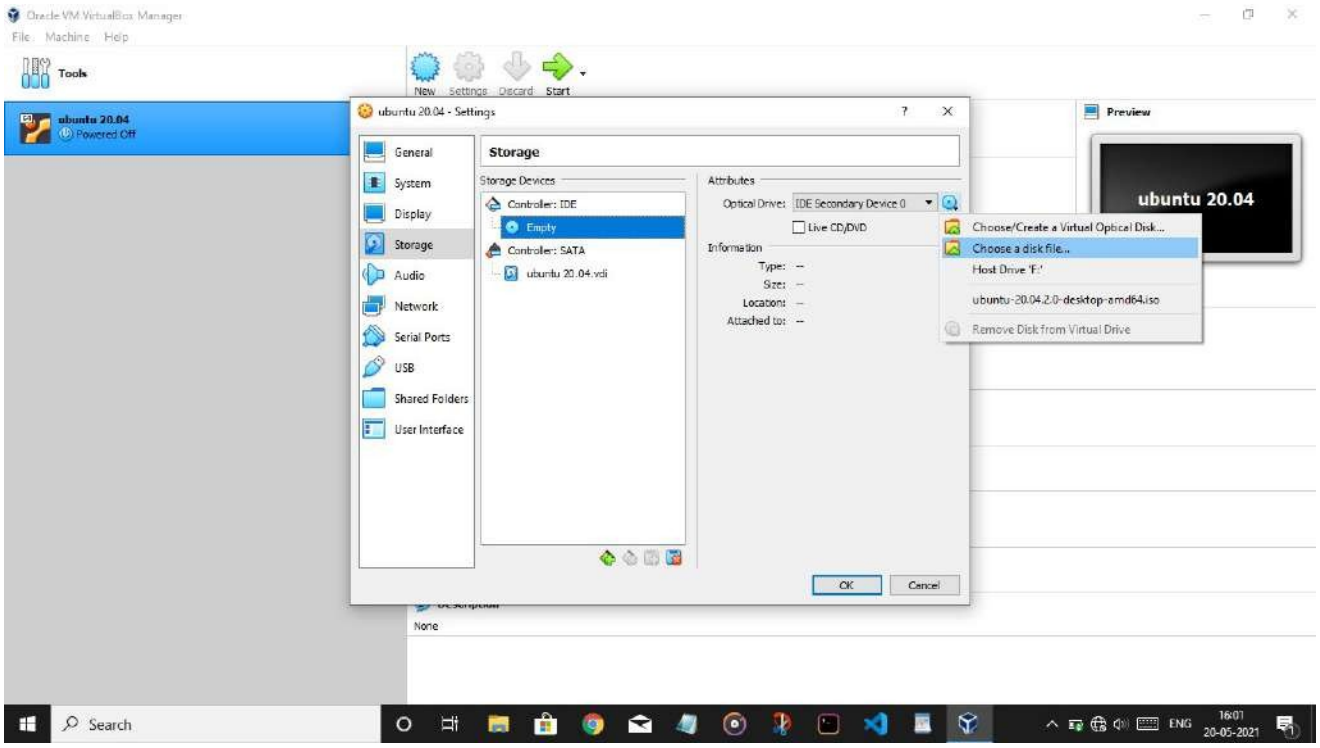
Click on ‘settings’.

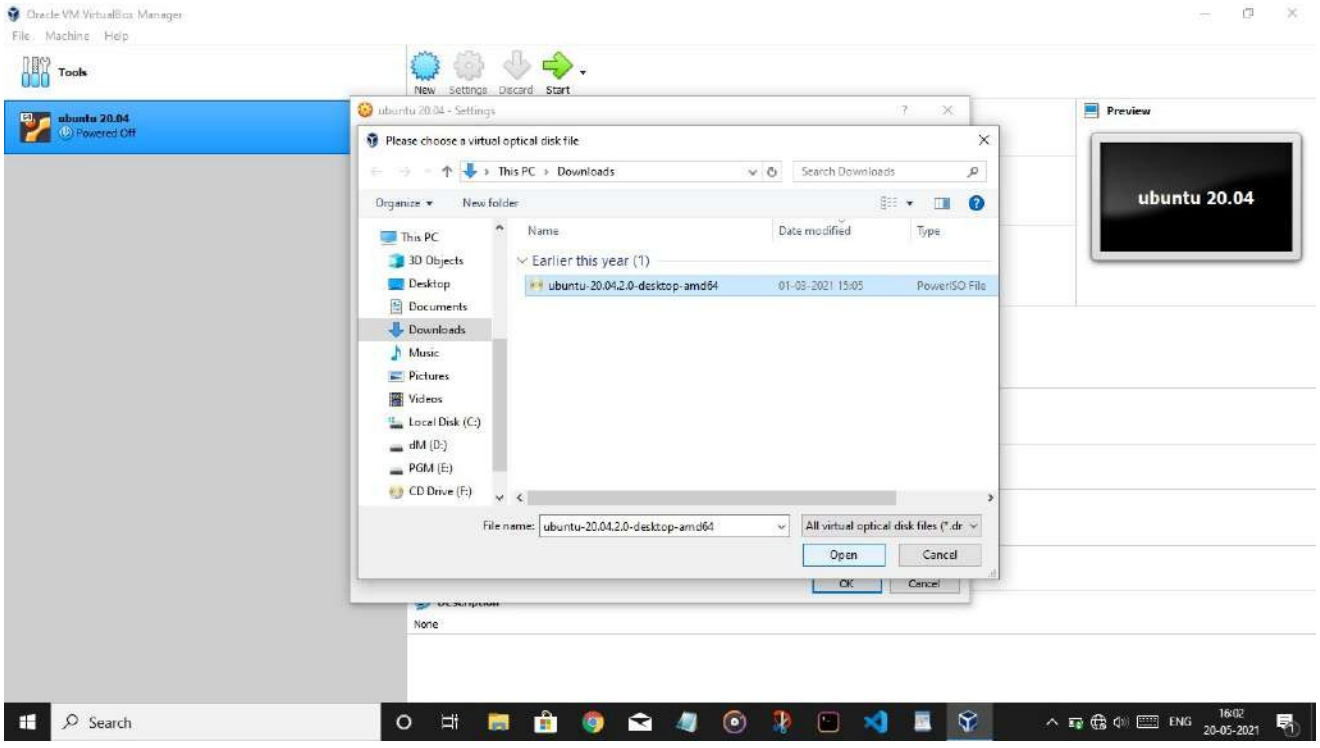


select ‘storage’.

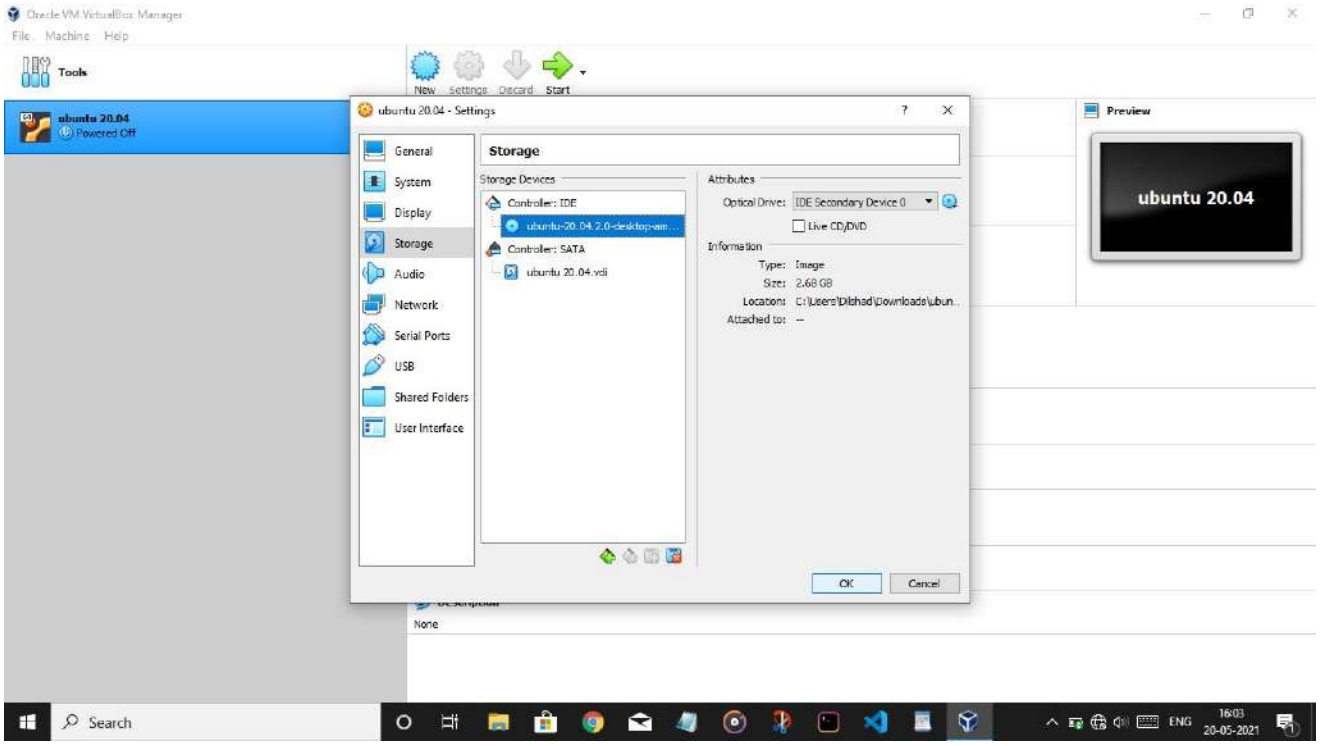


click on cd icon and choose the downloaded ubuntu file.



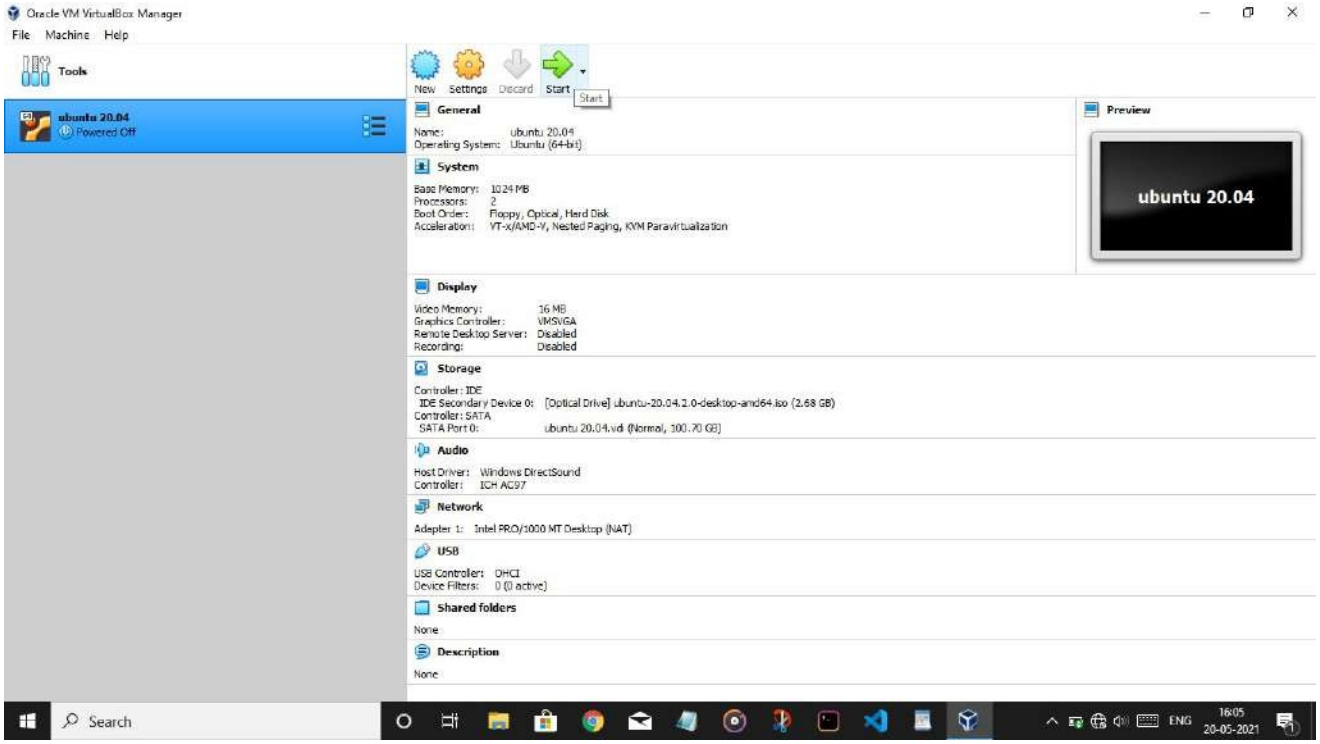


Click ‘OK’.

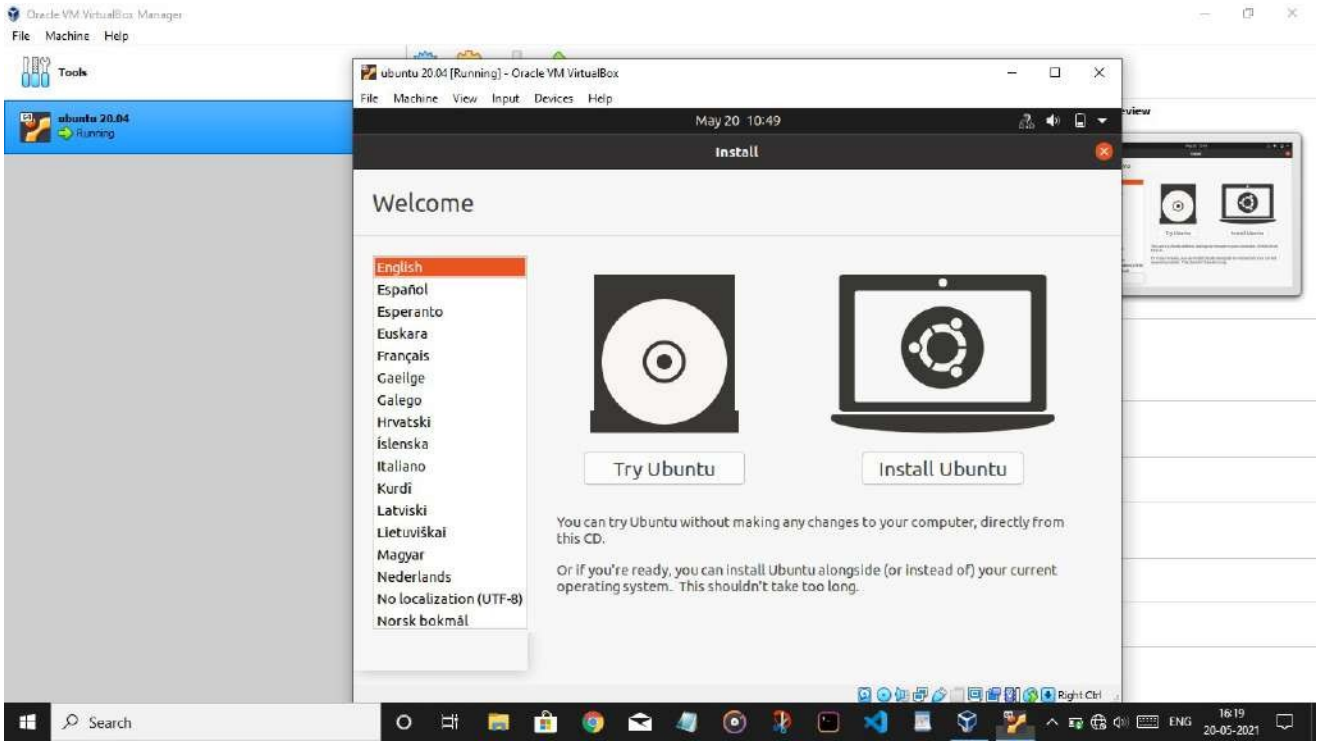


Step 5 :-

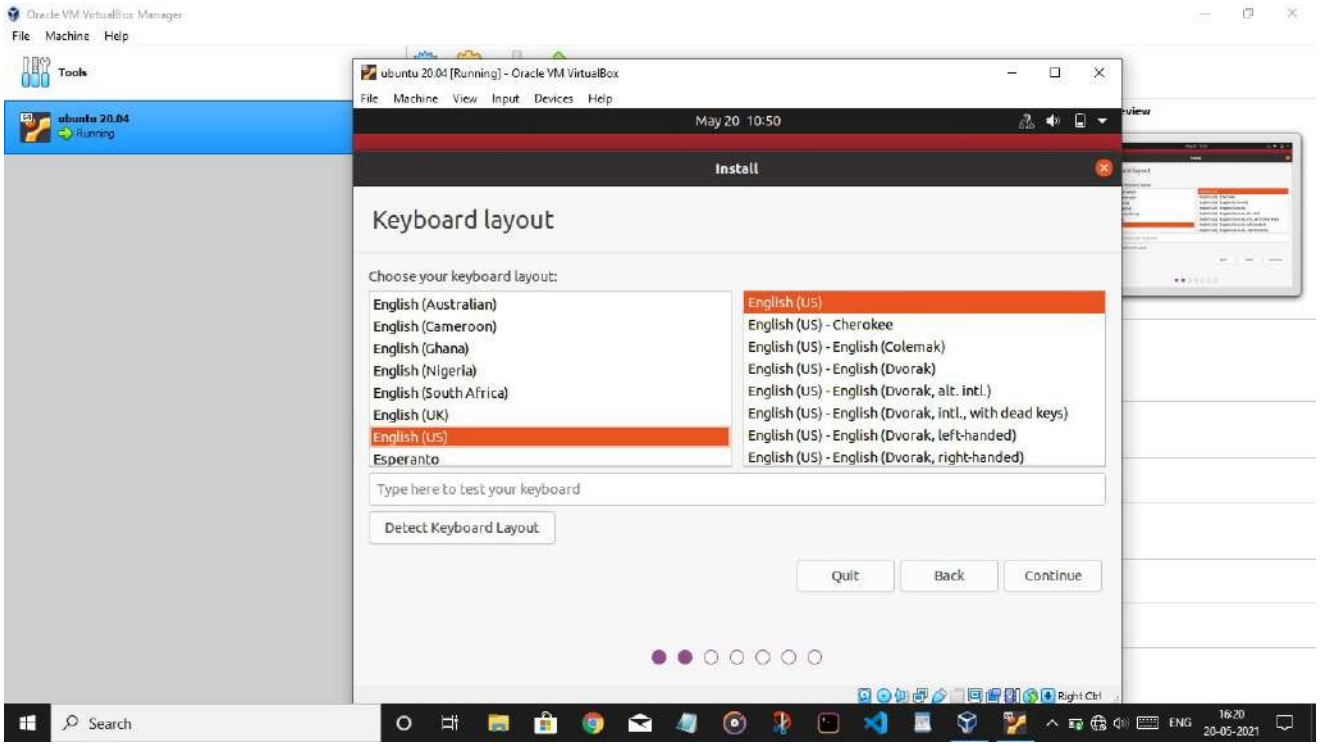
Click on ‘Start’ buton.

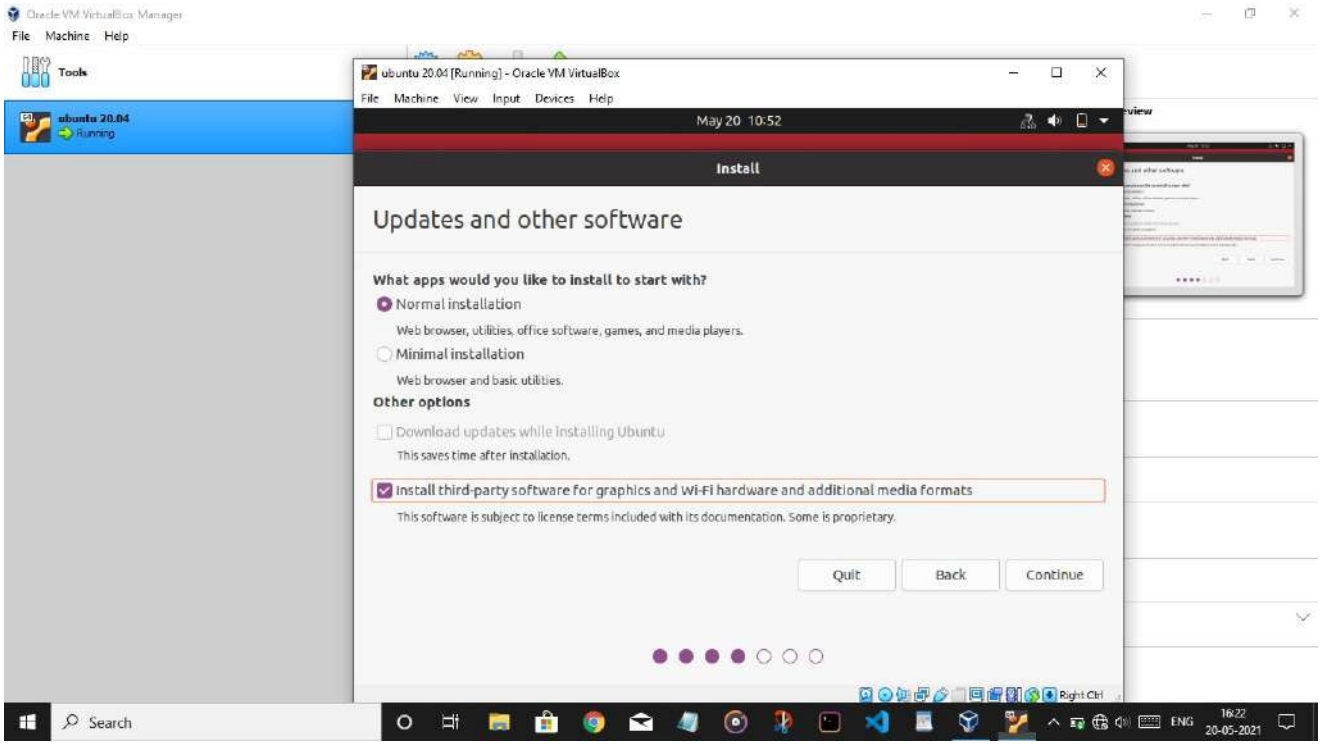


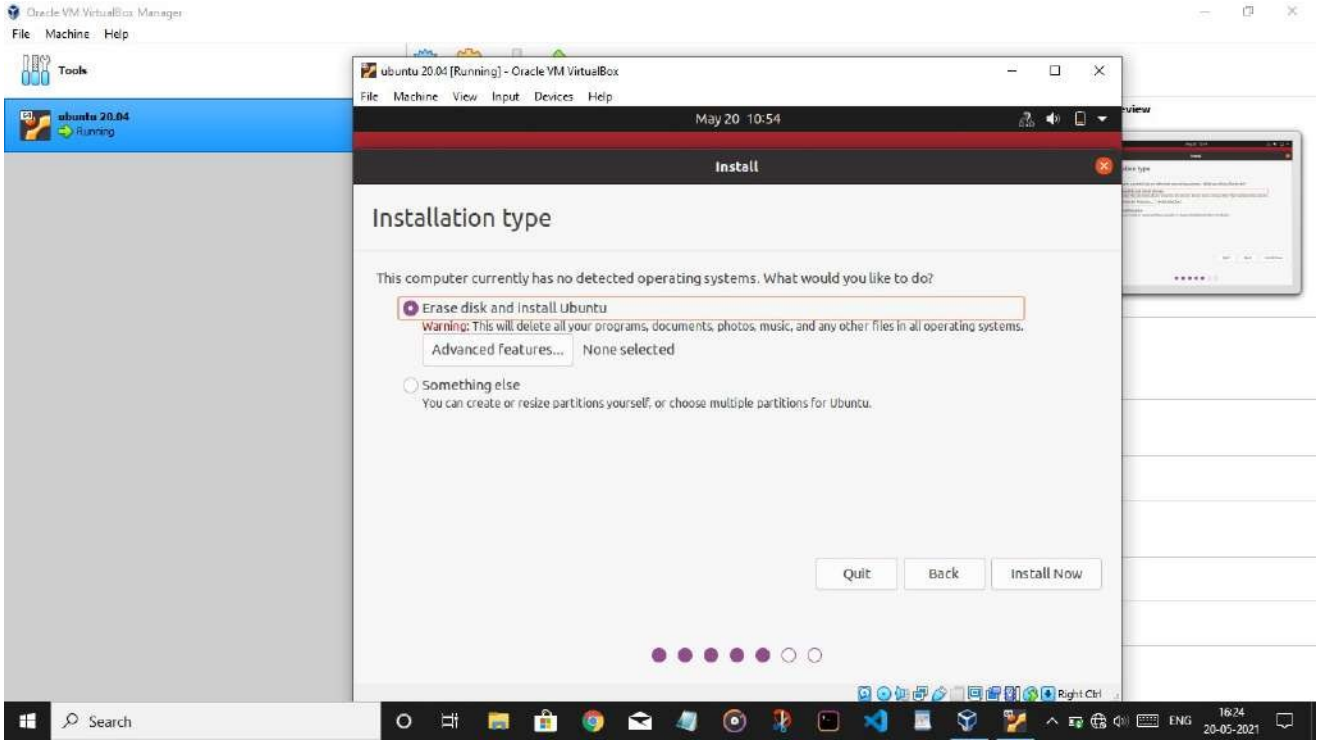
select the language and click on ‘install ubuntu’.

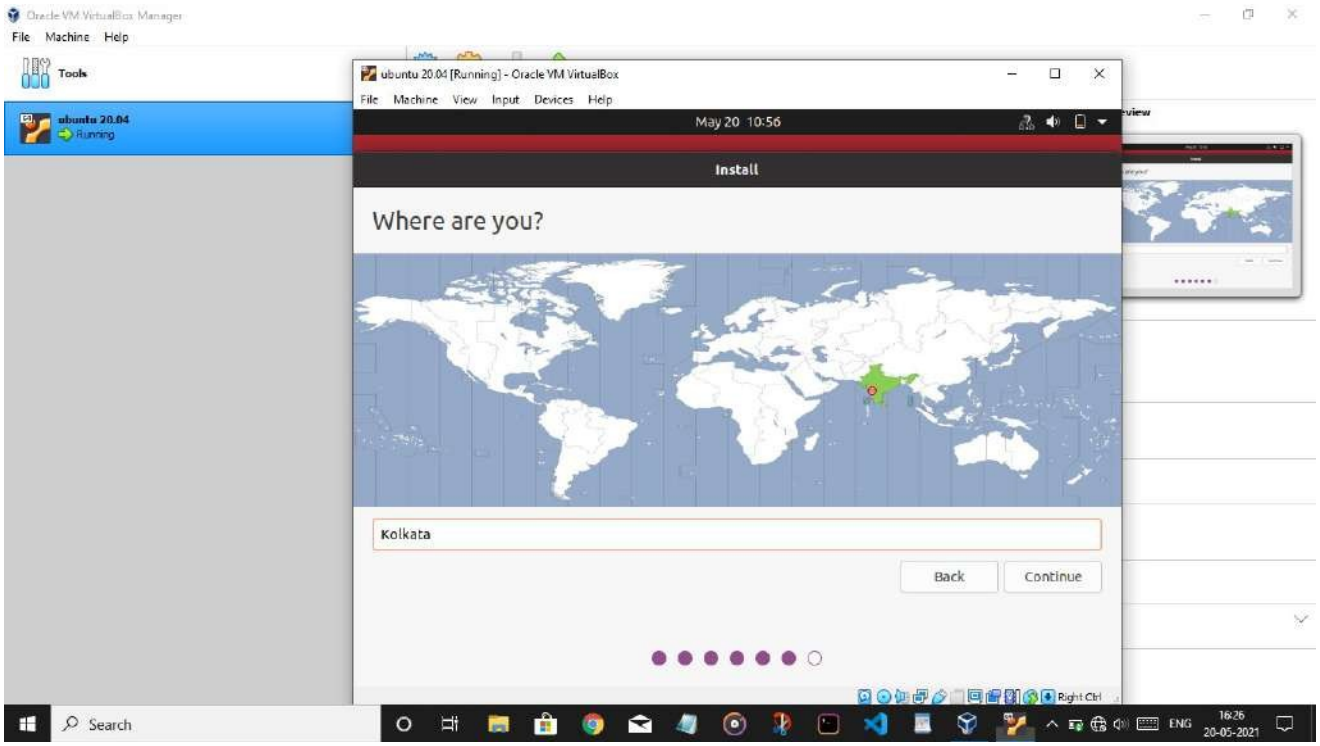


follow the procedures until the installation start.

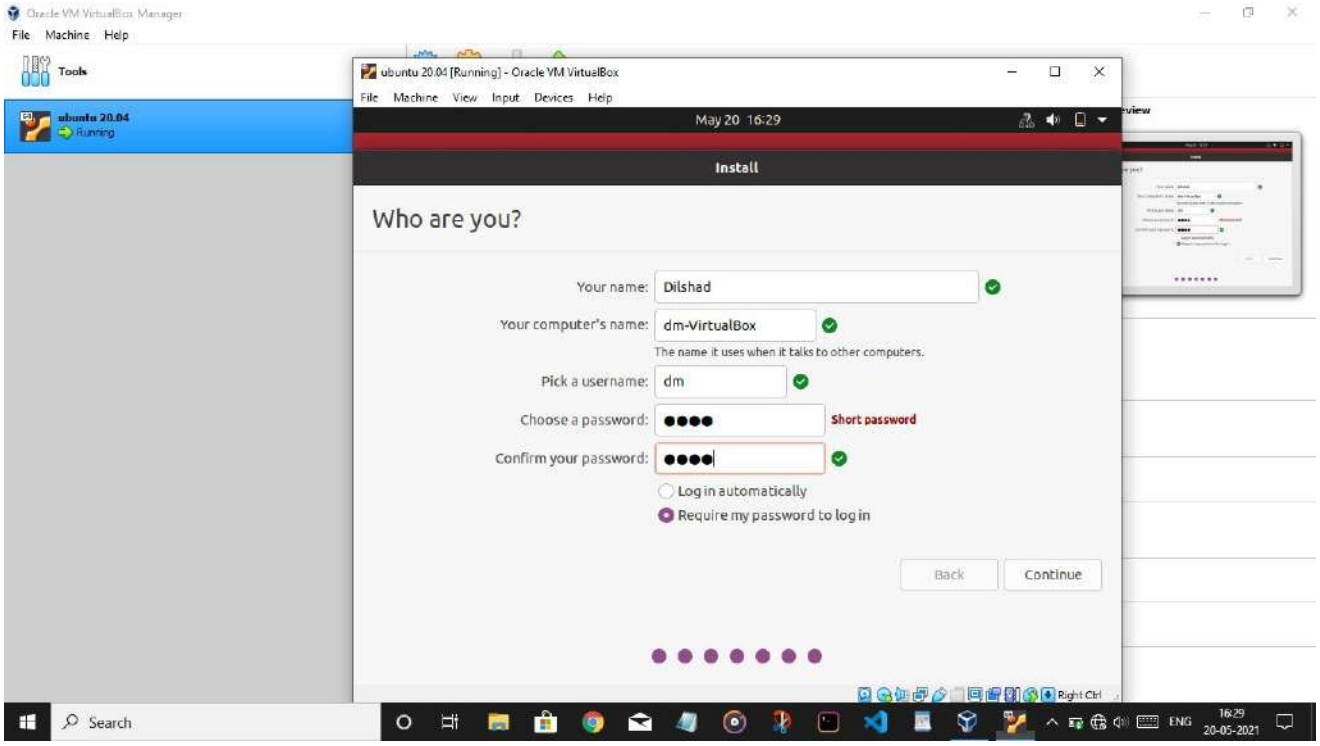



Oracle VM VirtualBox Manager
File Machine Help
Tools
ubuntu 20.04
Running
ubuntu 20.04 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
May 20 10:52
Install
Updates and other software
What apps would you like to install to start with?
Normal installation
Web browser, utilities, office software, games, and media players.
Minimal installation
Web browser and basic utilities.
Other options
Download updates while installing Ubuntu
This saves time after installation.
Install third-party software for graphics and Wi-Fi hardware and additional media formats
This software is subject to license terms included with its documentation. Some is proprietary.
Quit
Back
Continue
Right Ctrl
Search
ENG
16:22
20-05-2021


Oracle VM VirtualBox Manager
File Machine Help
Tools
ubuntu 20.04
Running
ubuntu 20.04 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
May 20 10:54
Install
Installation type
This computer currently has no detected operating systems. What would you like to do?
Erase disk and install Ubuntu
Warning: This will delete all your programs, documents, photos, music, and any other files in all operating systems.
Advanced features...
None selected
Something else
You can create or resize partitions yourself, or choose multiple partitions for Ubuntu.
Quit
Back
Install Now
Right Ctrl
Search
ENG
16:24
20-05-2021

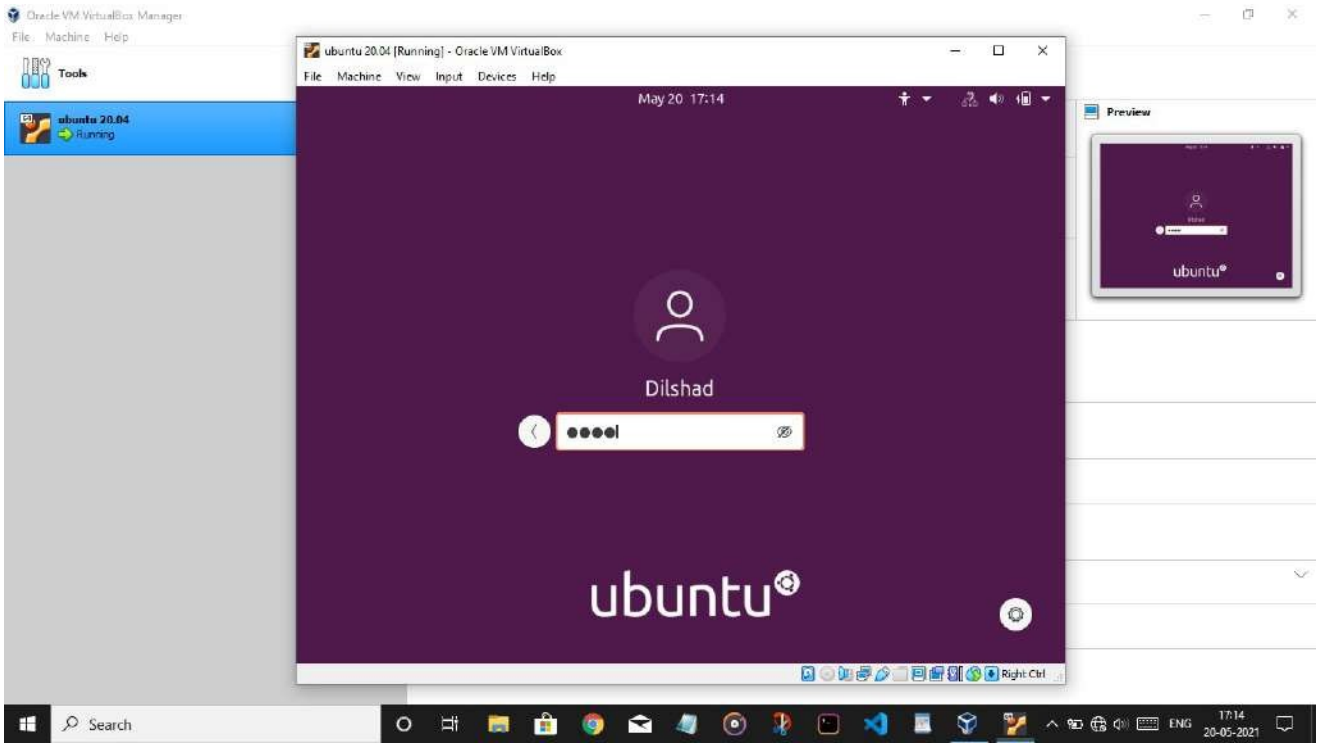


provide a username and password for ubuntu.



Step 6 :-

With the registered password, login to ubuntu OS.



INSTALLATION COMPLETED.