

**FEDERAL INSTITUTE OF  
SCIENCE AND TECHNOLOGY  
(FISAT)<sup>TM</sup>**

**HORMIS NAGAR, MOOKKANNOOR**

**ANGAMALY-683577**



**'FOCUS ON EXCELLENCE'**

**NETWORKING & SYSTEM ADMINISTRATION**

.....

**LABORATORY RECORD**

**Name : AQUIN JOHNSON**

**Branch : MASTER OF COMPUTER APPLICATION**

**Semester : 2      Batch : SEMESTER -A      Roll No : 32**

# **FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY**

## **(FISAT)™**

## HORMIS NAGAR, MOOKKANNOOR

**ANGAMALY-683577**



## **‘FOCUS ON EXCELLENCE’**

**Name : AQUIN JOHNSON**

**Branch : MASTER OF COMPUTER APPLICATION**

Semester : 2 Roll No: 32

**University Exam.Reg. No :** .....

## **CERTIFICATE**

*Certified that this is the Bonafide record of the Practical work done by  
AQUIN JOHNSON in the NETWORKING & SYSTEM ADMINISTRATION  
Laboratory of the Federal Institute of Science and Technology during the academic year  
2020-21.*

Signature of Staff in Charge  
Name :  
Date :

Signature of H.O.D  
Name:

*Date of University practical examination* .....

**Signature of  
Internal Examiner**

## CONTENTS

<b>SI No:</b>	<b>Date :</b>	<b>Name of Experiment:</b>	<b>Page No:</b>	<b>Signature of Staff –In– Charge:</b>
1	27-11-20	COMPONENT IDENTIFICATION	1	
2	27-11-20	LINUX COMMANDS	23	
3	4-12-20	LINUX FILE SYSTEM	30	
4	17-12-20	SHELL SCRIPT	39	
5	31-12-20	LAMP STOCK SERVER	49	
6	14-01-21	LARAVEL SERVER	56	
7	15-01-21	NETWORKING COMMANDS	66	
8	21-01-21	WIRE SHARK	69	
9	21-01-21	INTRODUCTION TO VIRTUAL MACHINES	73	

## EXPERIMENT -1

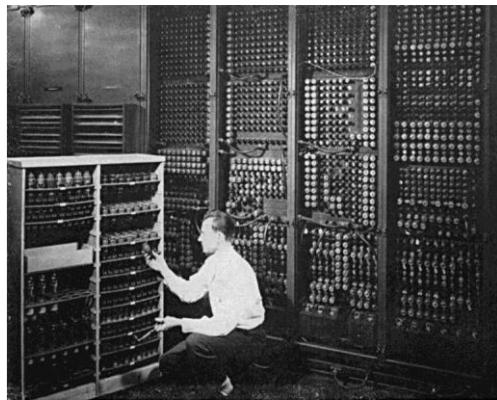
### BASIC INTRODUCTORY CONCEPT OF COMPUTER

### HARDWARE

#### **What is Computer?**

A computer is a programmable electronic device that accepts raw data as input and processes it with a set of instructions (a program) to produce the result as output. It renders output just after performing mathematical and logical operations and can save the output for future use. It can process numerical as well as non-numerical calculations.

#### **History of Computers**

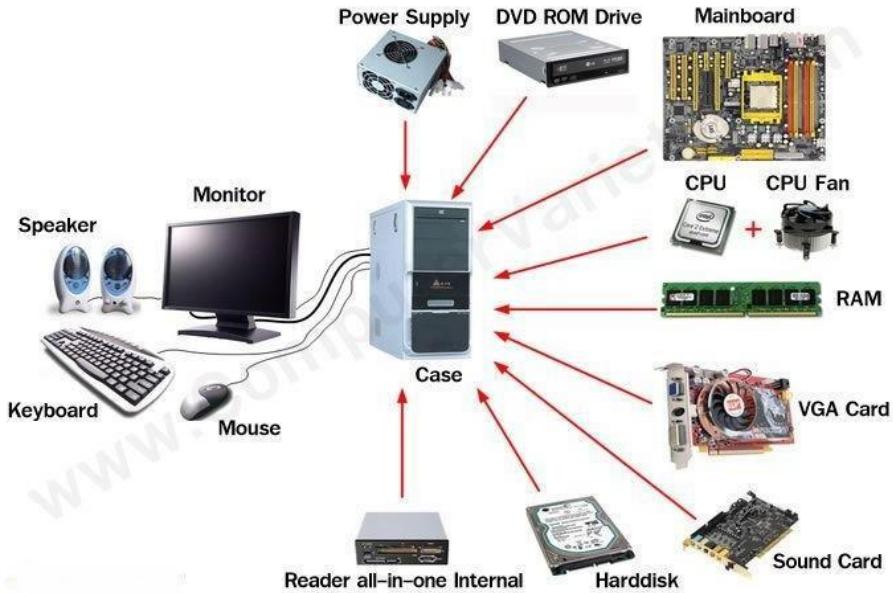


Replacing a bad tube meant checking among ENIAC's 19,000 possibilities.

Since the evolution of humans, devices have been used for calculations for thousands of years. One of the earliest and well-known devices was an abacus. Then in 1822, the father of computers, **Charles Babbage** began developing the first mechanical computer.

#### **Computer Hardware**

Computer hardware includes the physical parts of a computer, such as the case, central processing unit (CPU), monitor, mouse, keyboard, computer data storage, graphics card, sound card, speakers and motherboard. Hardware is typically directed by the software to execute any command or instruction. A combination of hardware and software forms a usable computing system, although other systems exist with only hardware.



The following are the basic components that will be detailed :-

1. Motherboard
2. RAM Modules
3. Daughter cards
4. Bus slots
5. SMPS
6. Internal Storage Devices
7. Interfacing Ports

## **1. MOTHERBOARD**

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



Motherboard means specifically a PCB with expansion capabilities. As the name suggests, this board is often referred to as the "mother" of all components attached to it, which often include peripherals, interface cards, and daughter cards: soundcards, video cards, network cards, host bus adapters, TV tuner cards, IEEE 1394 cards; and a variety of other custom components.

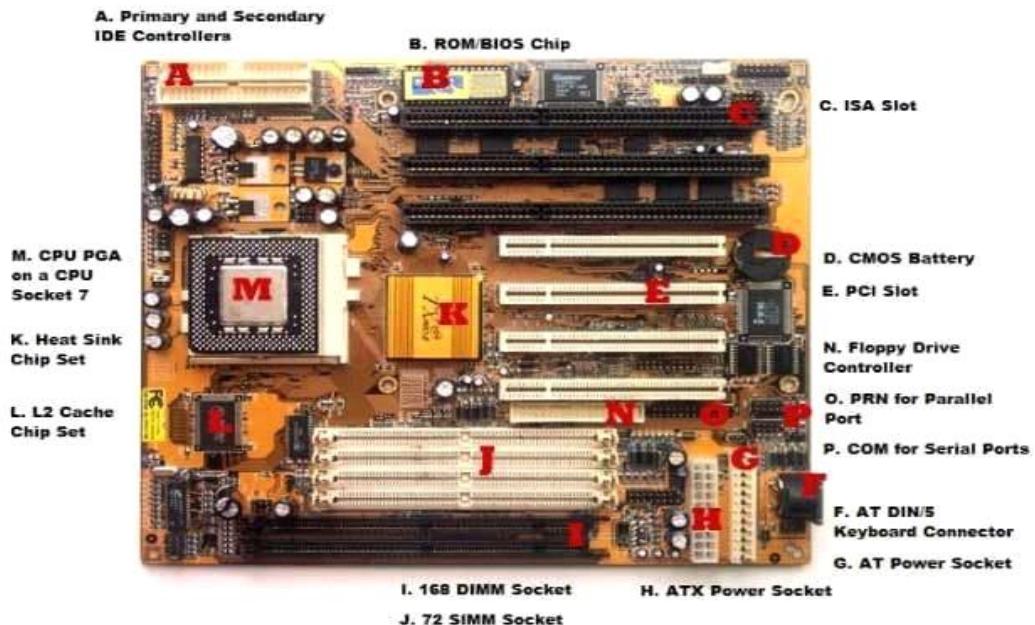
### **FEATURES OF MOTHERBOARD**

- Motherboard varies greatly in supporting various types of components.
- Motherboard supports a single type of CPU and few types of memories.
- Video cards, hard disks, sound cards have to be compatible with the motherboard to function properly.
- Motherboards, cases, and power supplies must be compatible to work properly together.

### **TYPES OF MOTHERBOARD**

#### **2.1 AT Motherboard**

These motherboards have bigger physical dimensions of hundreds of millimetres and hence they are not the right fit for the mini desktop category of computers.



## 2.2 ATX Motherboards

ATX denotes Advanced technology extended, It was developed by Intel during the 1990s and it was an improved version over an earlier version of AT motherboard. It is smaller in size when compared to AT and it provides interchangeability of the connected components.

## 2.3 BTX Motherboard

BTX denotes Balanced Technology Extended, intended to manage demands of new technologies in terms of more power requirements hence generation of more heat.

## 2.4 Pico BTX Motherboard

These boards are smaller in size and hence the word Pico. Two expansion slots are supported in spite of being sharing the top half of BTX. Half-height or riser cards are its unique features and it supports the demands of digital applications.

## 2.5 Mini ITX Motherboard

It's a miniature version of motherboard. Designed in the early 2000s and its dimension is 17 x 17 cm. Mainly used in small form factor (SFF) computer due to its lower power consumption and faster cooling ability.

## MAIN COMPONENTS OF MOTHERBOARD

### 2.1 FLOPPY DISC CONTROLLER

A floppy disk controller (FDC) is an electronic chip controller used as an interface between a computer and a floppy disk drive. Modern computers have this chip embedded

in the motherboard, whereas they were a separate component when they were originally introduced.

A floppy disk controller (FDC) is a specially designed chip that controls the reading and writing functionality of a floppy drive. An FDC can support up to four floppy disk drives at a time. The controller is connected to the system bus of the CPU and appears as a set of I/O ports to the computer. It is usually also linked to a serial bus of the direct memory access (DMA) controller. In an x86 computer, the floppy disk controller uses IRQ6, whereas interrupt schemes are used on other systems. Data transmission is often done by FDC while in DMA mode.



#### Floppy disk controller functions (FDC)

- Translate data bits into FM, MFM, M<sup>2</sup>FM, or GCR format to be able to record them
- Interpret and execute commands such as seek, read, write, format, etc.
- Error detection with check sums generation and verification, like CRC
- Synchronize data with phase-locked loop(PLL)

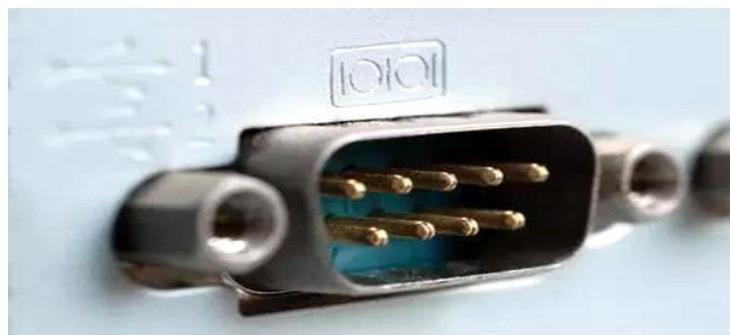
## 2.2 Serial Ports



Serial Ports provide an interface to connect serial lines to prepare a serial communication. Serial ports are typically used in modem, mouse, security cameras etc. A Serial port uses DB-9 connector, a 9 pin D-Shaped Connector which connects to the transmission line. A

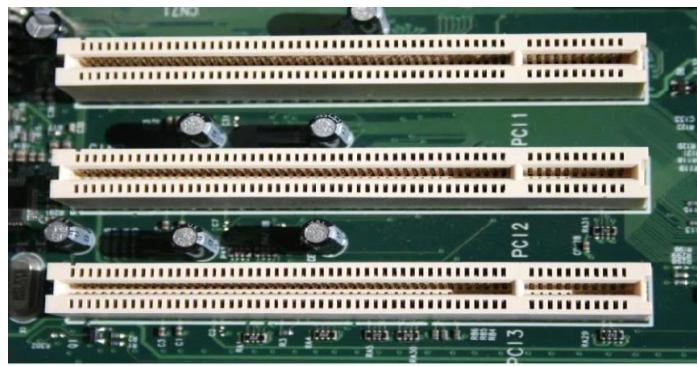
serial port provides a serial communication using one line and thus have no dependency on other wire's speed and its length can be extended as per the need.

### **2.3 Parallel Ports**



Parallel ports provide an interface to connect multiple lines to prepare a parallel communication to send large data at a time. Parallel ports are used in connecting printers, hard-drives, CD-drives etc. All lines speed should be same to avoid error and cross-talk issues. To avoid such issues, the wires are kept small in length. A parallel port uses D-25 connector, a 25 pin D- Shaped connector which connects to the transmission wires.

### **2.4 The Expansion Buses**



An expansion bus is an input/output pathway from the CPU to peripheral devices and it is typically made up of a series of slots on the motherboard. Expansion boards(cards) plug into the bus. PCI is the most common expansion bus in a PC and other hardware platforms. PCI stands for Peripheral Component Interface; PCI slot allows you to insert expansion cards into your computer. Buses carry signals such as data, memory addresses, power, and control signals from component to component. Other types of buses include ISA and EISA. Expansion buses enhance the PCs capabilities by allowing users to add missing features in their computers by slotting adapter cards into expansion slots.

## 2.5 The Computer Chip-sets



A chipset is a group of small circuits that coordinate the flow of data to and from a PC's key components. These key components include the CPU itself, the main memory, the secondary cache, and any devices situated on the buses. A chip set also controls data flow to and from hard disks and other devices connected to the IDE channels.

A computer has got two main chipsets:

- The North Bridge (also called the memory controller) is in charge of controlling transfers between the processor and the RAM, which is why it is located physically near the processor. It is sometimes called the GMCH, for Graphic and Memory Controller Hub.
- The South Bridge (also called the input/output controller or expansion controller) handles communications between slower peripheral devices. It is also called the ICH (I/O Controller Hub). The term "bridge" is generally used to designate a component which connects two buses.

Chipset manufacturers include SIS, VIA, ALI, and OPTI.

## 3. RANDOM ACCESS MEMORY

### Introduction

RAM, which stands for Random Access Memory, is a hardware device generally located on the motherboard of a computer and acts as an internal memory of the CPU. It allows CPU store data, program, and program results when you switch on the computer. It is the read and write memory of a computer, which means the information can be written to it as well as read from it.



RAM comes in the form of a chip that is individually mounted on the motherboard or in the form of several chips on a small board connected to the motherboard. It is the main memory of a computer. It is faster to write to and read from as compared to other memories such as a hard disk drive (HDD), solid-state drive (SSD), optical drive, etc. A computer's performance mainly depends on the size or storage capacity of the RAM. If it does not have sufficient RAM (random access memory) to run the OS and software programs, it will result in slower performance. So, the more RAM a computer has, the faster it will work. Information stored in RAM is accessed randomly, not in a sequence as on a CD or hard drive. So, its access time is much faster.

## **CHARACTERISTICS OF RAM**

### **3.1 SDRAM AND DDR**

Memory modules are labelled with either SDRAM (Synchronous Dynamic Random-Access Memory) or DDR (Double Data Rate). DDR RAM, as the "double data rate" name suggests, offers much faster speeds than SDRAM. Each generation of DDR, such as DDR2 and DDR3, offers performance improvements over the one preceding it.

### **3.2 SPEED**

The two numbers often quoted first on memory module specifications -- for example, "DDRxxx/PCxxxx" -- indicate the maximum clock speed and maximum transfer rate the device can operate at -- and the higher the better. The stated clock speed is actually double the real figure, so a stick of RAM labelled DDR3-1333 PC3200 offers a clock speed of 666 MHz and a transfer rate of 3,200 MB/s.

### **3.3 PINS**

Essentially, the number of pins a memory module has indicates the number of connections it has to the motherboard -- and thus which motherboards it's compatible with. More pins mean more data can be transferred at once, for faster operation overall, though performance is based on a variety of different factors, including CPU speed and the motherboard configuration.

### **3.4 VOLTS**

The voltage rating associated with a memory module -- for example, 2.5 V -- indicates how much power it draws from the motherboard in order to operate properly. RAM sticks that can work at lower voltages use up less power and give off less heat, and are therefore more suitable for smaller systems such as laptops.

### **3.5 CAS LATENCY**

CAS (Column Address Strobe) Latency, sometimes abbreviated to just "CL," indicates the time it takes for a memory module to return data to the CPU. A lower CAS Latency indicates RAM that performs faster.

### **3.6 TIMING**

Memory modules feature other timings besides CAS Latency, usually listed as a series of numbers after the other specifications. In order after CAS Latency, they are RAS (Row Address Strobe) to CAS delay, RAS Precharge, Active to Precharge delay and, optionally, command rate. These timings are only really of interest to advanced technical users, as the impact they have on performance is very small.

### **3.7 REDUNDANCY**

The redundancy built into a memory module indicates its ability to recover from errors and to alert the operating system to a problem, rather than just allowing it to crash and lose your data. More expensive and critical server memory uses Error Checking and Correcting Redundancy, or ECC, in order to detect and correct errors wherever possible.



### **Different Types of RAM**

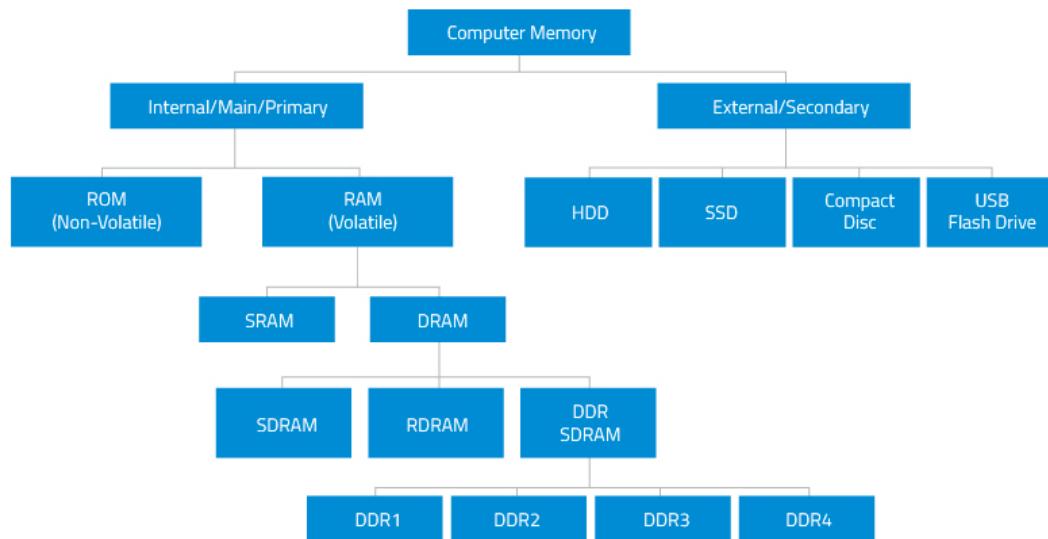
RAM(Random Access Memory) is a part of computer's Main Memory which is directly accessible by CPU. RAM is used to Read and Write data into it which is accessed by CPU randomly. RAM is volatile in nature, it means if the power goes off, the stored

information is lost. RAM is used to store the data that is currently processed by the CPU.

Most of the programs and data that are modifiable are stored in RAM.

Integrated RAM chips are available in two form:

1. SRAM(Static RAM)
2. DRAM(Dynamic RAM)



### **3.1 SRAM**

The SRAM memories consist of circuits capable of retaining the stored information as long as the power is applied. That means this type of memory requires constant power. SRAM memories are used to build Cache Memory.

### **3.2 DRAM**

DRAM stores the binary information in the form of electric charges that applied to capacitors. The stored information on the capacitors tend to lose over a period of time and thus the capacitors must be periodically recharged to retain their usage. The main memory is generally made up of DRAM chips.

#### **Functions**

##### **3.1. Reading Files**

Hard drives can store vast numbers of files, but compared to other computer components, drives run very slowly. Accessing hard drive files -- especially when those files are scattered across the drive due to fragmentation -- requires the drive to move its mechanical read/write head back and forth and to wait for the spinning platters to spin into the correct position. Even though drives spin at thousands of rotations per minute,

this process causes a noticeable delay when reading files. To lessen the slowdown, computers store files in RAM after the files are first read from the drive. RAM has no moving parts, so the files can load very quickly during subsequent uses.

### **3.2 Temporary Storage**

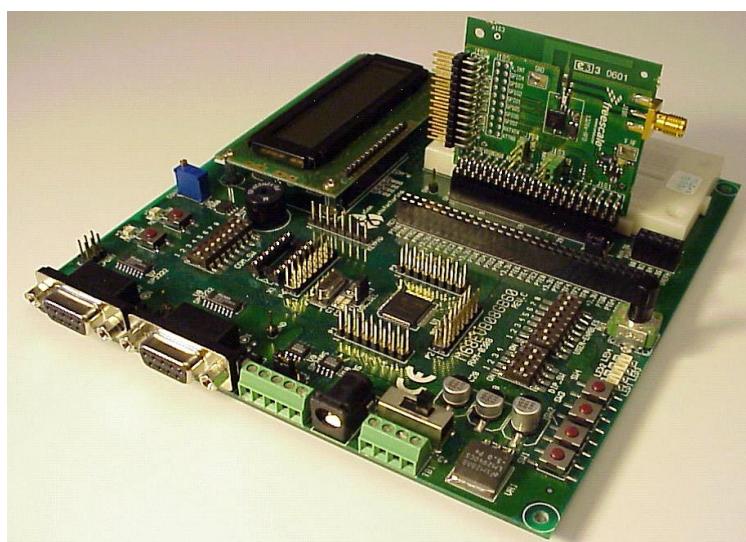
In addition to storing files read from the hard drive, RAM also stores data that programs are actively using but that doesn't need to be saved permanently. By keeping this data in RAM, programs can work with it quickly, improving speed and responsiveness.

### **3.3 RAM Size**

If RAM works so much faster than the hard drive, why not load all of the computer's data into RAM? One major reason: computers have far less RAM than drive space. As of publication, hard drive sizes range from a few hundred gigabytes in laptops to 10TB in high-end enterprise systems. Most home computers have between 1 and 4TB of drive space.

## **4. DAUGHTER CARD**

Referred to as a piggyback board and **daughter card**, a daughterboard is an expansion board that connects directly to the motherboard and gives added functionality.



Motherboard with daughter card

To disable a daughter board, the user must physically remove it from the motherboard. Daughter boards do not provide new functions to the circuit like an expansion but they extend the circuitry of the circuit in which they are plugged into.

**Functionalities of daughter board:**

- It is known as the piggyback board, riser card, daughtercard etcetera.
- A daughter board is smaller than a motherboard and may have some slots like the motherboard.
- A daughter board is a printed circuit board which is connected to the motherboard or expansion card.
- Unlike expansion card, daughter boards are directly connected to the motherboard by soldering.
- Daughter boards do not provide new functions to the circuit like an expansion but they extend the circuitry of the circuit in which they are plugged into.
- Daughter boards are released by the vendors as an update of motherboard or expansion card.

**List of daughter cards**

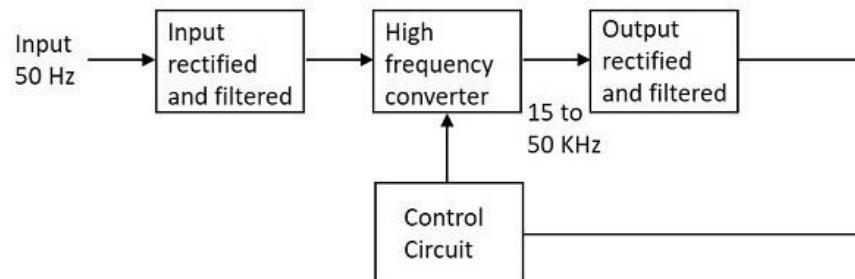
- Video Card: This is also referred to as the graphics adapter, display adapter or video adapter.
- Sound Card: To handle sound, to insert a microphone or connect a speaker this sound card is used.
- Network Interface Card: This is also referred as NIC. The computer can be connected to a network only with the use of this network interface card.
- Ethernet Card: Ethernet card is used to connect computers to computers. A cable is used to connect the Ethernet cards in each computer to make a network.

## **5. SWITCHED- MODE POWER SUPPLY**

Switched-mode power supply Introduction A switched-mode power supply (switching-mode power supply, switch-mode power supply, switched power supply, SMPS, or switcher) is an electronic power supply that incorporates a switching regulator to convert electrical power efficiently. Its function is to convert a level of voltage to the voltage or current required by the client through different forms of architecture.



### **WORKING**



The working of SMPS is simply understood by knowing that the transistor used in LPS is used to control the voltage drop while the transistor in SMPS is used as a controlled switch. The AC input supply signal 50 Hz is given directly to the rectifier and filter circuit combination without using any transformer. A fast switching device such as a Power transistor is employed in this section, which switches ON and OFF according to the variations and this output is given to the primary of the transformer. This is a regulated output voltage which is then given to the control circuit, which is a feedback circuit.

## **6. INTERNAL STORAGE DEVICES**

A storage device is any type of computing hardware that is used for storing, porting or extracting data files and objects. Storage devices can hold and store information both temporarily and permanently. They may be internal or external to a computer.

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

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

### **Examples of Internal storage devices**

- Hard Disk
- SSD
- RAM

#### **6.1 HARD DISK**

A **hard disk drive** (sometimes abbreviated as a **hard drive**, **HD**, or **HDD**) is a non-volatile data storage device. It is usually installed internally in a computer, attached directly to the disk controller of the computer's motherboard. It contains one or more platters, housed inside of an air-sealed casing. Data is written to the platters using a magnetic head, which moves rapidly over them as they spin.

Internal hard disks reside in a drive bay, connected to the motherboard using an ATA, SCSI, or SATA cable. They are powered by a connection to the computer's PSU (power supply unit).

Hard disk, Magnetic storage medium for a microcomputer. Hard disks are flat, circular plates made of aluminum or glass and coated with a magnetic material. Hard disks for personal computers can store up to several gigabytes (billions of bytes) of information. Data are stored on their surfaces in concentric tracks. A small electromagnet, called a magnetic head, writes a binary digit (1 or 0) by magnetizing tiny spots on the spinning disk in different directions and reads digits by detecting the magnetization direction of the spots. A computer's hard drive is a device consisting of several hard disks, read/write heads, a drive motor to spin the disks, and a small amount of circuitry, all sealed in a metal case to protect the disks from dust. In addition to referring to the disks themselves, the term hard disk is also used to refer to the whole hard drive.

Computers rely on hard disk drives (HDDs) to store data permanently. They are storage devices used to save and retrieve digital information that will be required for future reference.

Hard drives are non-volatile, meaning that they retain data even when they do not have power. The information stored remains safe and intact unless the hard drive is destroyed or interfered with. The information is stored or retrieved in a random-access manner as opposed to sequential access. This implies that blocks of data can be accessed at any time they are required without going through other data blocks.



## **6.2 SOLID STATE DRIVE**

A solid-state drive (SSD) is a solid-state storage device that uses integrated circuit assemblies to store data persistently , typically using flash memory, and functioning as secondary storage in the hierarchy of computer storage. It is also sometimes called a **solid-state device** or a **solid-state disk**, even though SSDs lack the physical spinning disks and movable read–write heads used in hard disk drives (HDDs) and floppy disks.

Compared with electromechanical drives, SSDs are typically more resistant to physical shock, run silently, and have quicker access time and lower latency. SSDs store data in semiconductor cells. SSDs have a limited number of writes, and slow as they reach storage capacity.



## **6.3 RAM**

RAM (Random Access Memory) is the internal memory of the CPU for storing data, program, and program result. It is a read/write memory which stores data until the machine is working. As soon as the machine is switched off, data is erased.

RAM is volatile, i.e. data stored in it is lost when we switch off the computer or if there is a power failure. RAM is small, both in terms of its physical size and in the amount of data it can hold.



RAM is of two types –

- Static RAM (SRAM)
- Dynamic RAM (DRAM)

### **Static RAM (SRAM)**

The word **static** indicates that the memory retains its contents as long as power is being supplied. However, data is lost when the power gets down due to volatile nature. SRAM chips use a matrix of 6-transistors and no capacitors. Transistors do not require power to prevent leakage, so SRAM need not be refreshed on a regular basis.

### **Dynamic RAM (DRAM)**

DRAM, unlike SRAM, must be continually **refreshed** in order to maintain the data. This is done by placing the memory on a refresh circuit that rewrites the data several hundred times per second. DRAM is used for most system memory as it is cheap and small.

## **6.4 SOME OTHER STORAGE DEVICES**

- Magnetic Storage Device
- Optical Storage Device
- Flash Memory Device
- Online and Cloud
- Paper Storage



## **7 . COMPUTER PORTS**

### **WHAT IS MEANT BY A PORT?**

A port in a computer network is a communication endpoint whereas, in an operating system, it is a logical construct, recognizes precise method otherwise a network service type. These endpoints recognize the combination of every protocol and its address through 16-bit unsigned numbers, called the port number. The protocols that use port numbers are the TCP (Transmission Control Protocol) and UDP (User Datagram Protocol). The port number in every computer networking uses an IP address of the type of protocol & the host

### **What is Port in Computer/Computer Port?**

A computer port or a communication port is a connection point used as an interface between the computer & the peripherals like keyboard, mouse, printer, display unit, monitor, flash drive and speaker. The computer port transmits the data from any peripheral to the computer. In general, the communication ports are available in two type like Serial Ports as well as Parallel Ports.



### **Characteristics of Computer Ports**

The characteristics of the computer port include the following.

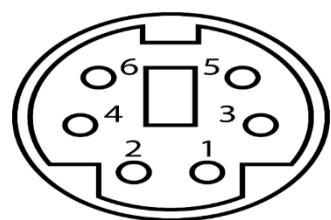
- It is an interface between external devices as well as a computer.
- Ports on the motherboard can be connected using an external device cable by plugging in.
- The external devices which are connected through via ports are the keyboard, mouse, microphone, monitor, speakers, etc.

### **Types of Computer Ports**

There are different types of ports available in a computer network. Some of them are:

- PS/2
- Serial Port
- Parallel Port
- Ethernet
- VGA Port
- USB Port
- DVI Port
- HDMI Port
- Display Port

**7.1 PS/2 PORTS:** PS/2 is a type of port used by older computers for connecting input devices such as keyboards and mice. The port was introduced with IBM's Personal System/2 computer in 1987 (which was abbreviated "PS/2"). The PS/2 port has six pins and is roughly circular in shape



**7.2 SERIAL PORT:** A serial port is an interface that allows a PC to transmit or receive data one bit at a time. It is one of the oldest types of interfaces commonly used to connect printers and external modems to a PC. Compared to a parallel port, the data transfer rate of a serial port is slower. Normally, a serial port is a male port, while a parallel port is a female port. The serial port standard is RS-232. This standard is used for transmitting serial communication between devices,



**7.3 PARALLEL PORT:** Parallel port is a type of interface found on computers (personal and otherwise) for connecting peripherals. Parallel ports send multiple bits of data at once (parallel communication), as opposed to serial communication, in which bits are sent one at a time. Parallel port is an interface between computer and peripheral devices like printers with parallel communication



**7.4 ETHERNET:** An Ethernet port is a jack or socket on a computer that allows the use of an Ethernet connector. These ports are essential in allowing the creation of local area networks (LANs). An Ethernet port is usually found on networking devices, including computers, routers, video game consoles, modems, and televisions. Ethernet is a communication system that allows multiple local devices to share information and work together.



**7.5 VGA PORT:** Abbreviated VGA, Video Graphics Array is a standard type of connection for video devices such as monitors and projectors. Generally, VGA refers to the types of cables, ports, and connectors used to connect monitors to video cards.



**7.6 USB PORT:** A **USB port** is a standard cable connection interface for personal computers and consumer electronics devices. USB stands for Universal Serial Bus, USB ports allow USB devices to be connected to each other with and transfer digital data over USB cables



**7.7 HDMI PORT:** Full form of HDMI is **High-Definition Multimedia Interface**.

HDMI is a proprietary audio/video interface for transmitting uncompressed video data and compressed or uncompressed digital audio data from an HDMI-compliant source device, such as a display controller, to a compatible computer monitor, video projector, digital television, or digital audio device..



**7.8 DISPLAY PORT:** DisplayPort is a digital display interface developed by a consortium of PC and chip manufacturers and standardized by the Video Electronics Standards Association. Display Port is a digital display interface with optional multiple channel audio and other forms of data. Display Port is developed with an aim of replacing VGA and DVI ports as the main interface between a computer and monitor.



## EXERCISE 2

### LINUX COMMANDS

Q1. Create the directories and files as given in the above directory structure. Directories are mentioned as (d). Files should be filled with necessary text data

```
hp@aquin:~$ mkdir INDIA
hp@aquin:~$ cd INDIA
hp@aquin:~/INDIA$ mkdir KERALA TN AP
hp@aquin:~/INDIA$ ls
AP KERALA TN
hp@aquin:~/INDIA$ cd KERALA
hp@aquin:~/INDIA/KERALA$ touch culture.txt Language.txt Dress.txt
hp@aquin:~/INDIA/KERALA$ ls
culture.txt Dress.txt Language.txt
hp@aquin:~/INDIA/KERALA$ cd ..
hp@aquin:~/INDIA$ ls
AP KERALA TN
hp@aquin:~/INDIA$ cd TN
hp@aquin:~/INDIA/TN$ touch Fam.txt Land.txt
hp@aquin:~/INDIA/TN$ ls
Fam.txt Land.txt
hp@aquin:~/INDIA/TN$ cd ..
hp@aquin:~/INDIA$ cd AP
hp@aquin:~/INDIA/AP$ touch Industry.txt Farmers.txt General.txt
hp@aquin:~/INDIA/AP$ ls
Farmers.txt General.txt Industry.txt
hp@aquin:~/INDIA/AP$ mkdir Development
hp@aquin:~/INDIA/AP$
```

Q2. List your present working directory

```
hp@aquin:~$ cd ~
hp@aquin:~$ pwd
/home/hp
hp@aquin:~$
```

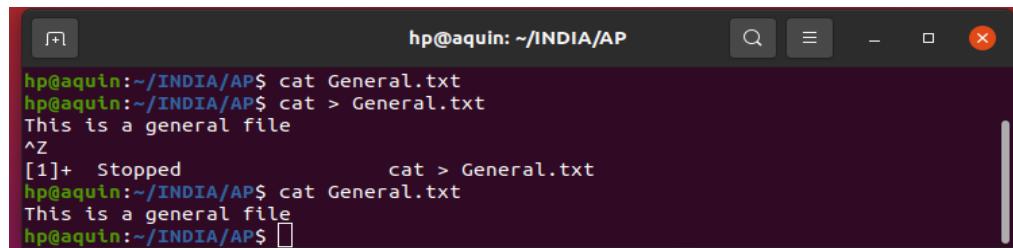
Q3. Move to the root directory.

```
hp@aquin:~$ cd /
hp@aquin:/$
```

Q4. Copy the file ‘Culture’to the folder AP

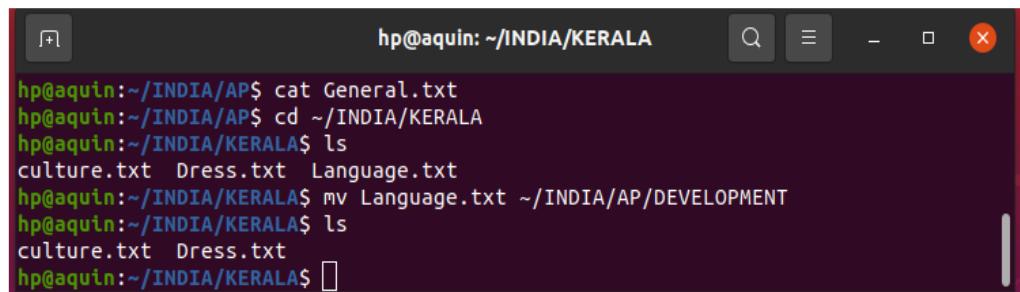
```
hp@aquin:~$ cd INDIA/KERALA
hp@aquin:~/INDIA/KERALA$ ls
culture.txt Dress.txt Language.txt
hp@aquin:~/INDIA/KERALA$ cp culture.txt ~/INDIA/AP
hp@aquin:~/INDIA/KERALA$ ls
culture.txt Dress.txt Language.txt
hp@aquin:~/INDIA/KERALA$ cd ~/INDIA/AP
hp@aquin:~/INDIA/AP$ ls
culture.txt Development Farmers.txt General.txt Industry.txt
hp@aquin:~/INDIA/AP$
```

Q5. Display the content of the file ‘general’



```
hp@aquin:~/INDIA/AP$ cat General.txt
hp@aquin:~/INDIA/AP$ cat > General.txt
This is a general file
^Z
[1]+  Stopped                  cat > General.txt
hp@aquin:~/INDIA/AP$ cat General.txt
This is a general file
hp@aquin:~/INDIA/AP$ 
```

Q6. Move the file ‘language’to the directory AP/Development



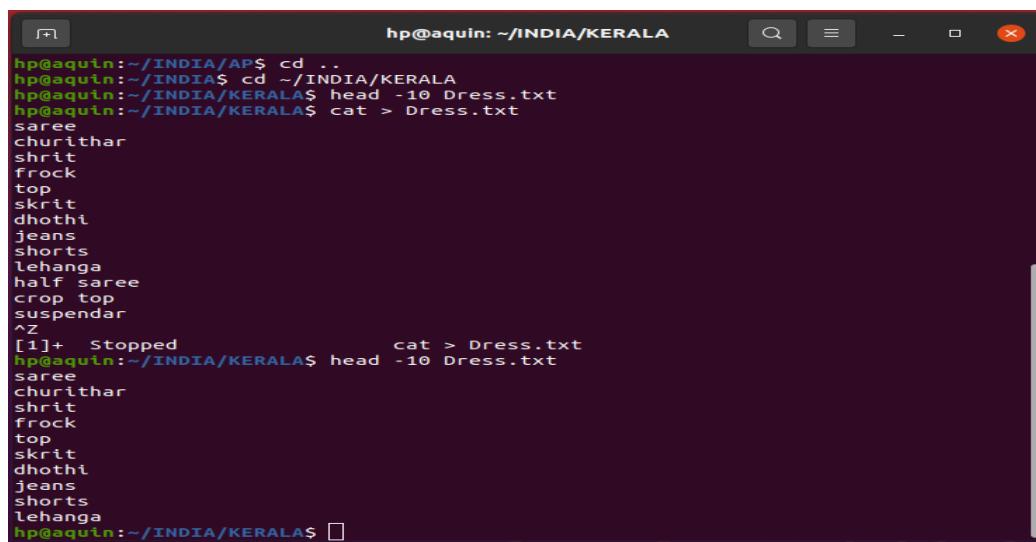
```
hp@aquin:~/INDIA/AP$ cat General.txt
hp@aquin:~/INDIA/AP$ cd ~/INDIA/KERALA
hp@aquin:~/INDIA/KERALA$ ls
culture.txt Dress.txt Language.txt
hp@aquin:~/INDIA/KERALA$ mv Language.txt ~/INDIA/AP/DEVELOPMENT
hp@aquin:~/INDIA/KERALA$ ls
culture.txt Dress.txt
hp@aquin:~/INDIA/KERALA$ 
```

Q7. List all the files in the folder AP



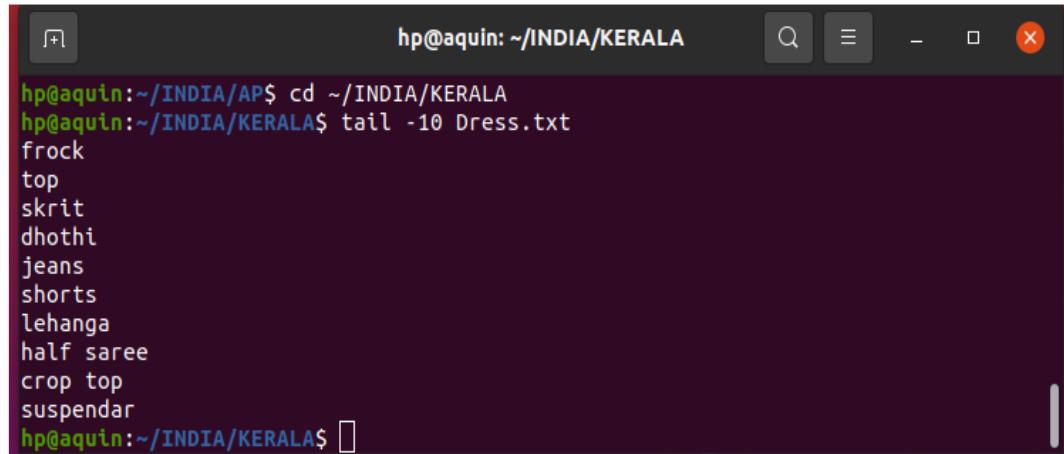
```
hp@aquin:~/INDIA/KERALA$ cd ~/INDIA/AP
hp@aquin:~/INDIA/AP$ ls
culture.txt Development DEVELOPMENT Farmers.txt General.txt Industry.txt
hp@aquin:~/INDIA/AP$ 
```

Q8. List first 10 lines of the file ‘Dress’



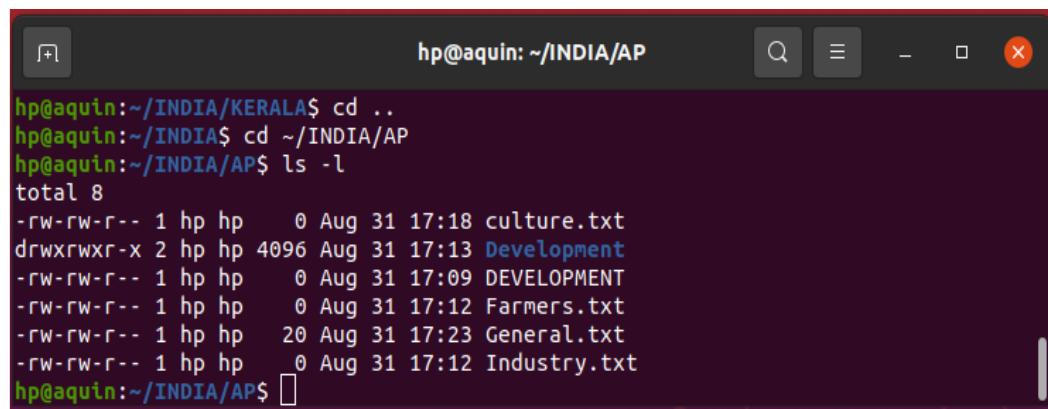
```
hp@aquin:~/INDIA/AP$ cd ..
hp@aquin:~/INDIA$ cd ~/INDIA/KERALA
hp@aquin:~/INDIA/KERALA$ head -10 Dress.txt
hp@aquin:~/INDIA/KERALA$ cat > Dress.txt
saree
churithar
shrit
frock
top
skrit
dhothi
jeans
shorts
lehanga
half saree
crop top
suspendar
^Z
[1]+  Stopped                  cat > Dress.txt
hp@aquin:~/INDIA/KERALA$ head -10 Dress.txt
saree
churithar
shrit
frock
top
skrit
dhothi
jeans
shorts
lehanga
hp@aquin:~/INDIA/KERALA$ 
```

Q9. List the last 10 lines of the file ‘Dress’.



```
hp@aquin:~/INDIA/AP$ cd ~/INDIA/KERALA
hp@aquin:~/INDIA/KERALA$ tail -10 Dress.txt
frock
top
skirt
dhoti
jeans
shorts
lehanga
half saree
crop top
suspendar
hp@aquin:~/INDIA/KERALA$
```

Q10. List all the files in AP in long listing format.



```
hp@aquin:~/INDIA/KERALA$ cd ..
hp@aquin:~/INDIA$ cd ~/INDIA/AP
hp@aquin:~/INDIA/AP$ ls -l
total 8
-rw-rw-r-- 1 hp hp    0 Aug 31 17:18 culture.txt
drwxrwxr-x 2 hp hp 4096 Aug 31 17:13 Development
-rw-rw-r-- 1 hp hp    0 Aug 31 17:09 DEVELOPMENT
-rw-rw-r-- 1 hp hp    0 Aug 31 17:12 Farmers.txt
-rw-rw-r-- 1 hp hp   20 Aug 31 17:23 General.txt
-rw-rw-r-- 1 hp hp    0 Aug 31 17:12 Industry.txt
hp@aquin:~/INDIA/AP$
```

Q11. List the files in AP which begin with the character ‘f’.



```
hp@aquin:~/INDIA/KERALA$ cd ..
hp@aquin:~/INDIA$ cd ~/INDIA/AP
hp@aquin:~/INDIA/AP$ ls F*
Farmers.txt
hp@aquin:~/INDIA/AP$
```

Q12. List the files page by page



A terminal window titled "hp@aquin: ~/INDIA/AP". The command "ls" is run, listing a single file "Industry.txt". The file has permissions -rw-rw-r--, belongs to user hp, was created on Aug 31 17:12, and is 0 bytes large.

```
-rw-rw-r-- 1 hp hp 0 Aug 31 17:12 Industry.txt
hp@aquin:~/INDIA/AP$ ls
hp@aquin:~/INDIA/AP$ 
```

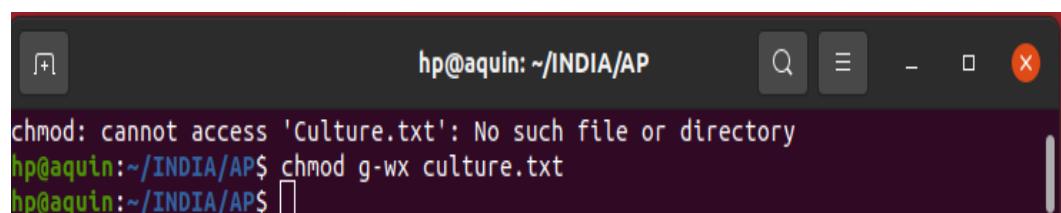
Q13. Remove the file 'general'.



A terminal window titled "hp@aquin: ~/INDIA/AP". The command "rm General.txt" is run, removing the file "General.txt". The file had permissions -rw-rw-r--, belonged to user hp, and was created on Aug 31 17:12.

```
-rw-rw-r-- 1 hp hp 0 Aug 31 17:12 Industry.txt
hp@aquin:~/INDIA/AP$ rm General.txt
hp@aquin:~/INDIA/AP$ 
```

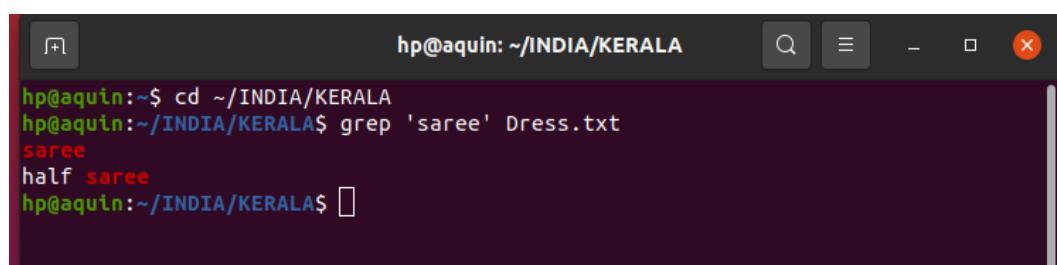
Q14. Change the permission of the file 'Culture' as only read permission to all.



A terminal window titled "hp@aquin: ~/INDIA/AP". The command "chmod g-wx culture.txt" is run, attempting to change the permissions of "culture.txt". The command fails with the error "chmod: cannot access 'Culture.txt': No such file or directory".

```
chmod: cannot access 'Culture.txt': No such file or directory
hp@aquin:~/INDIA/AP$ chmod g-wx culture.txt
hp@aquin:~/INDIA/AP$ 
```

Q15. List the lines of the file which contains a string 'saree'.



A terminal window titled "hp@aquin: ~/INDIA/KERALA". The command "grep 'saree' Dress.txt" is run, searching the file "Dress.txt" for the string "saree". It finds two lines: "saree" and "half saree".

```
hp@aquin:~$ cd ~/INDIA/KERALA
hp@aquin:~/INDIA/KERALA$ grep 'saree' Dress.txt
saree
half saree
hp@aquin:~/INDIA/KERALA$ 
```

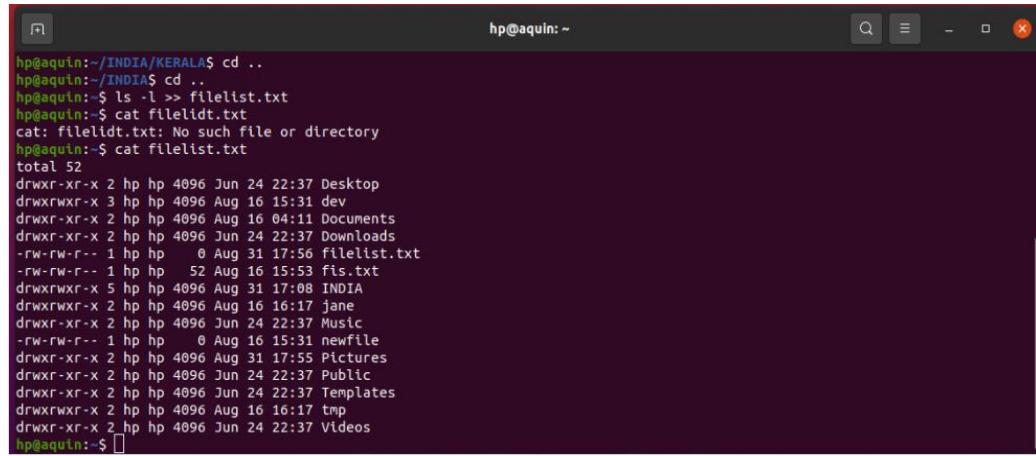
Q16. Use man command to get the syntax of wc command.

The screenshot shows a terminal window titled "User Commands" with the command "wc(1)" entered. The window displays the man page for the "wc" command, which includes sections for NAME, SYNOPSIS, DESCRIPTION, OPTIONS, AUTHOR, REPORTING BUGS, and COPYRIGHT. The "OPTIONS" section lists various flags such as -c, -m, -l, -E, -L, -w, --help, --version, and --bytes, --chars, --lines, --max-line-length, and --words. The "DESCRIPTION" section explains that it prints newline, word, and byte counts for each file. The "OPTIONS" section also describes how options like -c, -m, and -l work. The "AUTHOR" section credits Paul Rubin and David MacKenzie. The "REPORTING BUGS" section provides links for online help and reporting bugs. The "COPYRIGHT" section states the license information under GPLv3+. The bottom of the window shows the status bar with the text "Manual page wc(1) line 1/55 80% (press h for help or q to quit)".

Q17. Count the number of characters, words, lines in the directory listing

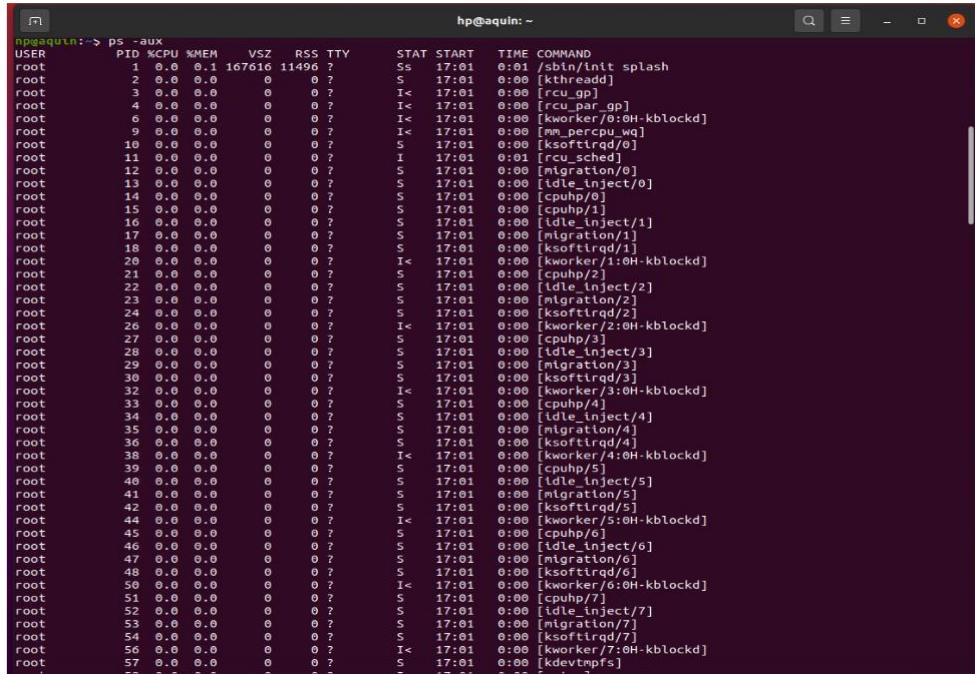
The screenshot shows a terminal window with the command "hp@aquin:~/INDIA/KERALA\$ wc -mwl ~/INDIA/Kerala/Dress.txt" entered. The output shows the counts for the file "Dress.txt": 13 characters, 15 words, and 96 lines. The window title is "User Commands".

Q18. Put a listing of the files in your directory into a file called filelist.



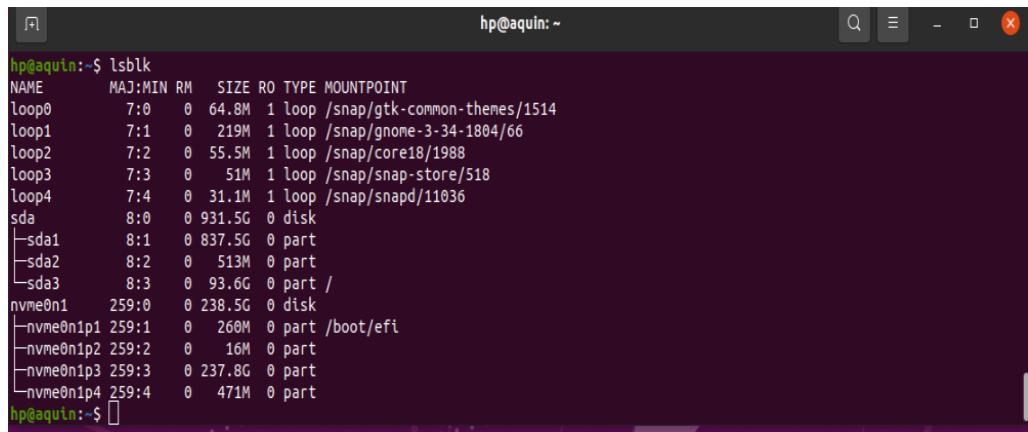
```
hp@equin:~/INDIA/KERALA$ cd ..
hp@equin:~/INDIA$ cd ..
hp@equin:~$ ls .l >> filelist.txt
hp@equin:~$ cat filelist.txt
cat: filelist.txt: No such file or directory
hp@equin:~$ cat filelist.txt
total 52
drwxr-xr-x 2 hp hp 4096 Jun 24 22:37 Desktop
drwxrwxr-x 3 hp hp 4096 Aug 16 15:31 dev
drwxr-xr-x 2 hp hp 4096 Aug 16 04:11 Documents
drwxr-xr-x 2 hp hp 4096 Jun 24 22:37 Downloads
-rw-rw-r-- 1 hp hp 0 Aug 31 17:56 filelist.txt
-rw-rw-r-- 1 hp hp 52 Aug 16 15:53 fis.txt
drwxrwxr-x 5 hp hp 4096 Aug 31 17:08 INDIA
drwxrwxr-x 2 hp hp 4096 Aug 16 16:17 jane
drwxr-xr-x 2 hp hp 4096 Jun 24 22:37 Music
-rw-rw-r-- 1 hp hp 0 Aug 16 15:31 newfile
drwxr-xr-x 2 hp hp 4096 Aug 31 17:55 Pictures
drwxr-xr-x 2 hp hp 4096 Jun 24 22:37 Public
drwxr-xr-x 2 hp hp 4096 Jun 24 22:37 Templates
drwxrwxr-x 2 hp hp 4096 Aug 16 16:17 tmp
drwxr-xr-x 2 hp hp 4096 Jun 24 22:37 Videos
hp@equin:~$
```

Q19. List the status of all process running in your system.



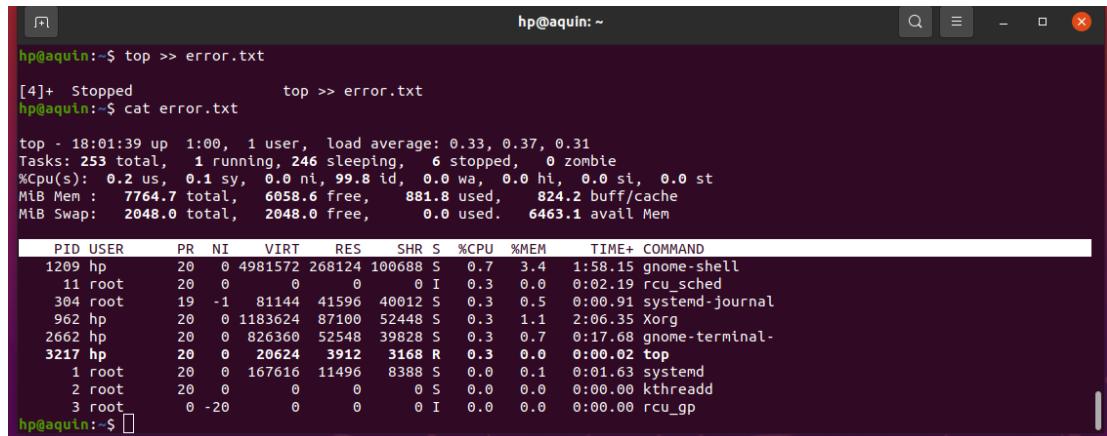
USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.1	167616	11496	?	Ss	17:01	0:01	/sbin/init splash
root	2	0.0	0.0	0	0	?	S	17:01	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	I<	17:01	0:00	[rcu_gp]
root	4	0.0	0.0	0	0	?	I<	17:01	0:00	[rcu_par_gp]
root	6	0.0	0.0	0	0	?	I<	17:01	0:00	[kworker/0:0:H-kblockd]
root	9	0.0	0.0	0	0	?	I<	17:01	0:00	[mm_percpu_wq]
root	10	0.0	0.0	0	0	?	S	17:01	0:00	[ksoftirqd/0]
root	11	0.0	0.0	0	0	?	L	17:01	0:00	[migration/0]
root	12	0.0	0.0	0	0	?	S	17:01	0:00	[idle_inject/0]
root	13	0.0	0.0	0	0	?	S	17:01	0:00	[cpuhp/0]
root	14	0.0	0.0	0	0	?	S	17:01	0:00	[cpuhp/1]
root	15	0.0	0.0	0	0	?	S	17:01	0:00	[cpuhp/1]
root	16	0.0	0.0	0	0	?	S	17:01	0:00	[idle_inject/1]
root	17	0.0	0.0	0	0	?	S	17:01	0:00	[migration/1]
root	18	0.0	0.0	0	0	?	S	17:01	0:00	[ksoftirqd/1]
root	20	0.0	0.0	0	0	?	I<	17:01	0:00	[kworker/1:0:H-kblockd]
root	21	0.0	0.0	0	0	?	S	17:01	0:00	[cpuhp/2]
root	22	0.0	0.0	0	0	?	S	17:01	0:00	[idle_inject/2]
root	23	0.0	0.0	0	0	?	S	17:01	0:00	[migration/2]
root	24	0.0	0.0	0	0	?	S	17:01	0:00	[ksoftirqd/2]
root	26	0.0	0.0	0	0	?	I<	17:01	0:00	[kworker/2:0:H-kblockd]
root	27	0.0	0.0	0	0	?	S	17:01	0:00	[cpuhp/3]
root	28	0.0	0.0	0	0	?	S	17:01	0:00	[idle_inject/3]
root	29	0.0	0.0	0	0	?	S	17:01	0:00	[migration/3]
root	30	0.0	0.0	0	0	?	S	17:01	0:00	[ksoftirqd/3]
root	32	0.0	0.0	0	0	?	I<	17:01	0:00	[kworker/3:0:H-kblockd]
root	33	0.0	0.0	0	0	?	S	17:01	0:00	[cpuhp/4]
root	34	0.0	0.0	0	0	?	S	17:01	0:00	[idle_inject/4]
root	35	0.0	0.0	0	0	?	S	17:01	0:00	[migration/4]
root	36	0.0	0.0	0	0	?	S	17:01	0:00	[ksoftirqd/4]
root	38	0.0	0.0	0	0	?	I<	17:01	0:00	[kworker/4:0:H-kblockd]
root	39	0.0	0.0	0	0	?	S	17:01	0:00	[cpuhp/5]
root	40	0.0	0.0	0	0	?	S	17:01	0:00	[idle_inject/5]
root	41	0.0	0.0	0	0	?	S	17:01	0:00	[migration/5]
root	42	0.0	0.0	0	0	?	S	17:01	0:00	[ksoftirqd/5]
root	44	0.0	0.0	0	0	?	I<	17:01	0:00	[kworker/5:0:H-kblockd]
root	45	0.0	0.0	0	0	?	S	17:01	0:00	[cpuhp/6]
root	46	0.0	0.0	0	0	?	S	17:01	0:00	[idle_inject/6]
root	47	0.0	0.0	0	0	?	S	17:01	0:00	[migration/6]
root	48	0.0	0.0	0	0	?	S	17:01	0:00	[ksoftirqd/6]
root	50	0.0	0.0	0	0	?	I<	17:01	0:00	[kworker/6:0:H-kblockd]
root	51	0.0	0.0	0	0	?	S	17:01	0:00	[cpuhp/7]
root	52	0.0	0.0	0	0	?	S	17:01	0:00	[idle_inject/7]
root	53	0.0	0.0	0	0	?	S	17:01	0:00	[migration/7]
root	54	0.0	0.0	0	0	?	S	17:01	0:00	[ksoftirqd/7]
root	56	0.0	0.0	0	0	?	I<	17:01	0:00	[kworker/7:0:H-kblockd]
root	57	0.0	0.0	0	0	?	S	17:01	0:00	[kdevtmpfs]

Q20. List the disk partitions in your harddisk.



```
hp@aquin:~$ lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0      7:0    0 64.8M  1 loop /snap/gtk-common-themes/1514
loop1      7:1    0 219M  1 loop /snap/gnome-3-34-1804/66
loop2      7:2    0 55.5M  1 loop /snap/core18/1988
loop3      7:3    0 51M   1 loop /snap/snap-store/518
loop4      7:4    0 31.1M  1 loop /snap/snapd/11036
sda       8:0    0 931.5G 0 disk 
├─sda1     8:1    0 837.5G 0 part /
├─sda2     8:2    0 513M  0 part 
└─sda3     8:3    0 93.6G  0 part /
nvme0n1   259:0   0 238.5G 0 disk 
├─nvme0n1p1 259:1   0 260M  0 part /boot/efi
├─nvme0n1p2 259:2   0 16M   0 part 
├─nvme0n1p3 259:3   0 237.8G 0 part 
└─nvme0n1p4 259:4   0 471M  0 part
hp@aquin:~$
```

Q21. Redirect the output of the top program to a file called ‘errors’.



```
hp@aquin:~$ top >> error.txt
[4]+  Stopped                  top >> error.txt
hp@aquin:~$ cat error.txt
top - 18:01:39 up 1:00, 1 user, load average: 0.33, 0.37, 0.31
Tasks: 253 total, 1 running, 246 sleeping, 6 stopped, 0 zombie
%CPU(s): 0.2 us, 0.1 sy, 0.0 ni, 99.8 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
Mem Mem : 7764.7 total, 6058.6 free, 881.8 used, 824.2 buff/cache
Mem Swap: 2048.0 total, 2048.0 free, 0.0 used. 6463.1 avail Mem

          PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM TIME+ COMMAND
 1209 hp        20   0 4981572 268124 100688 S  0.7  3.4 1:58.15 gnome-shell
 11 root      20   0      0      0      0 I  0.3  0.0 0:02.19 rcu_sched
 304 root     19  -1  81144 41596 40012 S  0.3  0.5 0:00.91 systemd-journal
 962 hp        20   0 1183624 87100 52448 S  0.3  1.1 2:06.35 Xorg
 2662 hp      20   0 826360 52548 39828 S  0.3  0.7 0:17.68 gnome-terminal-
 3217 hp      20   0 20624 3912 3168 R  0.3  0.0 0:00.02 top
  1 root      20   0 167616 11496 8388 S  0.0  0.1 0:01.63 systemd
  2 root      20   0      0      0      0 S  0.0  0.0 0:00.00 kthreadd
  3 root      20  -20     0      0      0 I  0.0  0.0 0:00.00 rcu_gp
hp@aquin:~$
```

## **EXERCISE 3**

### **LINUX FILE SYSTEM**

It makes sense to explore the Linux filesystem from a terminal window,

In fact, that is the name of the first tool you'll install to help you on the way: *tree*. If you are using Ubuntu or Debian, you can do:

```
sudo apt install tree
```

Once installed, stay in your terminal window and run *tree* like this:

```
$ tree /
```

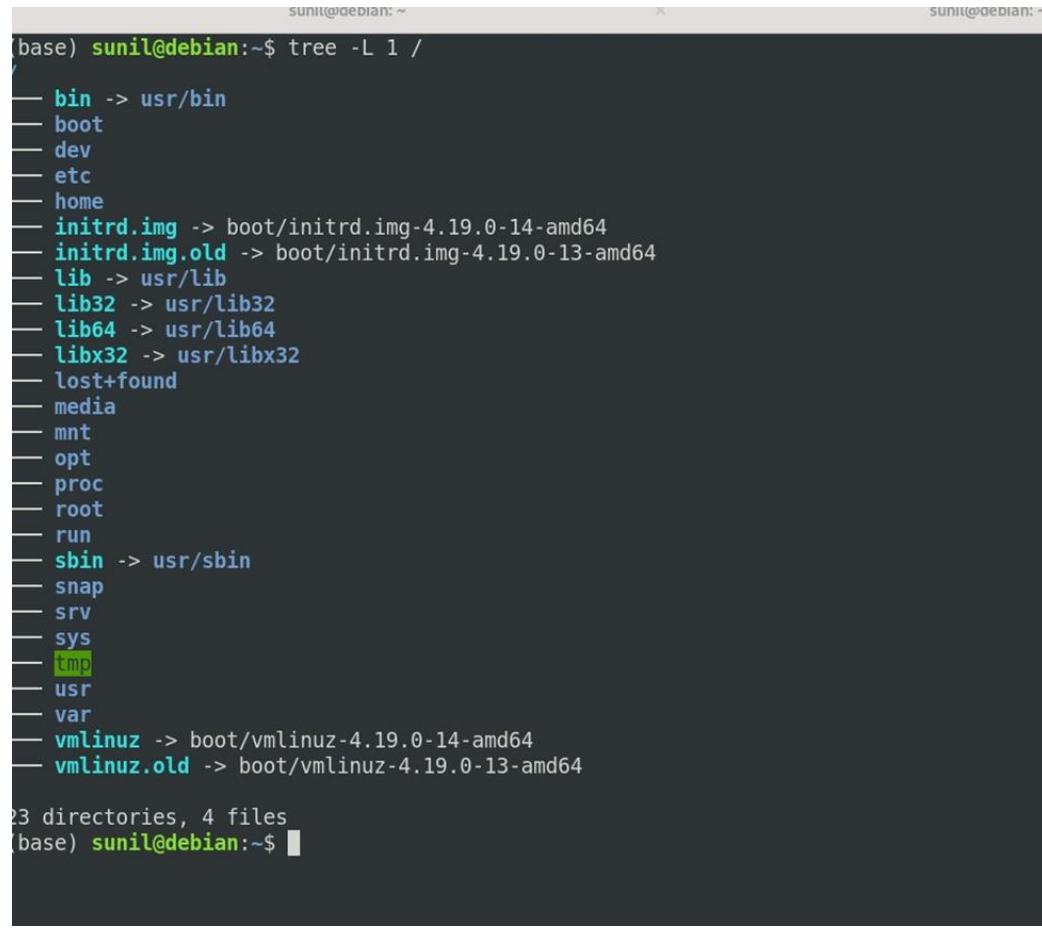
The / in the instruction above refers to the *root* directory. The root directory is the one from which all other directories branch off from. When you run *tree* and tell it to start with /, you will see the whole directory tree, all directories and all the subdirectories in the whole system, with all their files, fly by.

If you have been using your system for some time, this may take a while, because, even if you haven't generated many files yourself, a Linux system and its apps are always logging, caching, and storing temporary files. The number of entries in the file system can grow quite quickly.

Instead, try this

```
tree -L 1 /
```

And you should see a listing similar to what is shown in Figure 1.



```
sunil@debian: ~
(base) sunil@debian:~$ tree -L 1 /
/
├── bin -> usr/bin
├── boot
├── dev
├── etc
├── home
├── initrd.img -> boot/initrd.img-4.19.0-14-amd64
├── initrd.img.old -> boot/initrd.img-4.19.0-13-amd64
├── lib -> usr/lib
├── lib32 -> usr/lib32
├── lib64 -> usr/lib64
├── libx32 -> usr/libx32
├── lost+found
├── media
├── mnt
├── opt
├── proc
├── root
├── run
├── sbin -> usr/sbin
├── snap
├── srv
├── sys
└── tmp
├── usr
└── var
├── vmlinuz -> boot/vmlinuz-4.19.0-14-amd64
└── vmlinuz.old -> boot/vmlinuz-4.19.0-13-amd64

23 directories, 4 files
(base) sunil@debian:~$
```

Figure 1: tree

The instruction above can be translated as “*show me only the 1st Level of the directory tree starting at / (root)*“. The *-L* option tells *tree* how many levels down you want to see.

Most Linux distributions will show you the same or a very similar layout to what you can see in the image above. This means that even if you feel confused now, master this, and you will have a handle on most, if not all, Linux installations in the whole wide world.

Now, let’s look at what each directory is used for. While we go through each, you can peek at their contents using *ls*.

## **Directories**

From top to bottom, the directories you are seeing are as follows.

### **/bin**

*/bin* is the directory that contains *binaries*, that is, some of the applications and programs you can run. You will find the *ls* program mentioned above in this directory, as well as other basic tools for making and removing files and directories, moving them around, and so on. There are more *bin* directories in other parts of the file system tree, but we'll be talking about those in a minute.

### **/boot**

The */boot* directory contains files required for starting your system. If you mess up one of the files in here, you may not be able to run your Linux and it is a pain to repair. On the other hand, don't worry too much about destroying your system by accident: you have to have superuser privileges to do that.

### **/dev**

*/dev* contains *device* files. Many of these are generated at boot time or even on the fly. For example, if you plug in a new webcam or a USB pendrive into your machine, a new device entry will automatically pop up here.

### **/etc**

*/etc* is the directory where names start to get confusing. */etc* gets its name from the earliest Unixes and it was literally “et cetera” because it was the dumping ground for system files administrators were not sure where else to put.

Nowadays, it would be more appropriate to say that *etc* stands for “Everything to configure,” as it contains most, if not all system-wide configuration files. For example, the files that contain the name of your system, the users and their passwords, the names of machines on your network and when and where the partitions on your hard disks should be mounted are all in here. Again, if you are new to Linux, it may be best if you don't touch too much in here until you have a better understanding of how things work.

### **/home**

*/home* is where you will find your users' personal directories. In my case, under */home* there are two directories: */home/paul*, which contains all my stuff; and */home/guest*, in case anybody needs to borrow my computer.

### **/lib**

*lib* is where *libraries* live. Libraries are files containing code that your applications can use. They contain snippets of code that applications use to draw windows on your desktop, control peripherals, or send files to your hard disk.

There are more *lib* directories scattered around the file system, but this one, the one hanging directly off of */* is special in that, among other things, it contains the all-important kernel modules. The kernel modules are drivers that make things like your video card, sound card, WiFi, printer, and so on, work.

### **/media**

The */media* directory is where external storage will be automatically mounted when you plug it in and try to access it. As opposed to most of the other items on this list, */media* does not hail back to 1970s, mainly because inserting and detecting storage (pendrives, USB hard disks, SD cards, external SSDs, etc) on the fly, while a computer is running, is a relatively new thing.

### **/mnt**

The */mnt* directory, however, is a bit of remnant from days gone by. This is where you would manually mount storage devices or partitions. It is not used very often nowadays.

### **/opt**

The */opt* directory is often where software you compile (that is, you build yourself from source code and do not install from your distribution repositories) sometimes lands. Applications will end up in the */opt/bin* directory and libraries in the */opt/lib* directory.

A slight digression: another place where applications and libraries end up in is */usr/local*. When software gets installed here, there will also be */usr/local/bin* and */usr/local/lib* directories. What determines which software goes where is how the developers have configured the files that control the compilation and installation process.

### ***/proc***

*/proc*, like */dev* is a virtual directory. It contains information about your computer, such as information about your CPU and the kernel your Linux system is running. As with */dev*, the files and directories are generated when your computer starts, or on the fly, as your system is running and things change.

### ***/root***

*/root* is the home directory of the superuser (also known as the “Administrator”) of the system. It is separate from the rest of the users’ home directories BECAUSE YOU ARE NOT MEANT TO TOUCH IT. Keep your own stuff in your own directories, people.

### ***/run***

*/run* is another new directory. System processes use it to store temporary data for their own nefarious reasons.

### ***/sbin***

*/sbin* is similar to */bin*, but it contains applications that only the superuser (hence the initial *s*) will need. You can use these applications with the *sudo* command that temporarily concedes you superuser powers on many distributions. */sbin* typically contains tools that can install stuff, delete stuff and format stuff. As you can imagine, some of these instructions are lethal if you use them improperly, so handle with care.

### ***/usr***

The */usr* directory was where users’ home directories were originally kept back in the early days of UNIX. However, now */home* is where users kept their stuff as we saw above. These days, */usr* contains a mish-mash of directories which in turn contain applications, libraries, documentation, wallpapers, icons and a long list of other stuff that need to be shared by applications and services.

You will also find *bin*, *sbin* and *lib* directories in */usr*. What is the difference with their root-hanging cousins? Not much nowadays. Originally, the */bin* directory (hanging off of root) would contain very basic commands, like ls, mv and rm; the kind of commands that would come pre-installed in all UNIX/Linux installations, the bare minimum to run and maintain a system.

*/usr/bin* on the other hand would contain stuff the users would install and run to use the system as a work station, things like word processors, web browsers, and other apps.

But many modern Linux distributions just put everything into */usr/bin* and have */bin* point to */usr/bin* just in case erasing it completely would break something. So, while Debian, Ubuntu and Mint still keep */bin* and */usr/bin* (and */sbin* and */usr/sbin*) separate; others, like Arch and its derivatives just have one “real” directory for binaries, */usr/bin*, and the rest or \**bins* are “fake” directories that point to */usr/bin*.

#### */srv*

The */srv* directory contains data for servers. If you are running a web server from your Linux box, your HTML files for your sites would go into */srv/http* (or */srv/www*). If you were running an FTP server, your files would go into */srv/ftp*.

#### */sys*

*/sys* is another virtual directory like */proc* and */dev* and also contains information from devices connected to your computer.

In some cases you can also manipulate those devices. I can, for example, change the brightness of the screen of my laptop by modifying the value stored in the

*/sys/devices/pci0000:00/0000:00:02.0/drm/card1/card1-eDP-1/intel\_backlight/brightness* file (on your machine you will probably have a different file). But to do that you have to become superuser. The reason for that is, as with so many other virtual directories, messing with the contents and files in */sys* can be dangerous and you can trash your system. DO NOT TOUCH until you are sure you know what you are doing.

### **/tmp**

*/tmp* contains temporary files, usually placed there by applications that you are running. The files and directories often (not always) contain data that an application doesn't need right now, but may need later on.

You can also use */tmp* to store your own temporary files — */tmp* is one of the few directories hanging off / that you can actually interact with without becoming superuser.

### **/var**

*/var* was originally given its name because its contents was deemed *variable*, in that it changed frequently. Today it is a bit of a misnomer because there are many other directories that also contain data that changes frequently, especially the virtual directories we saw above.

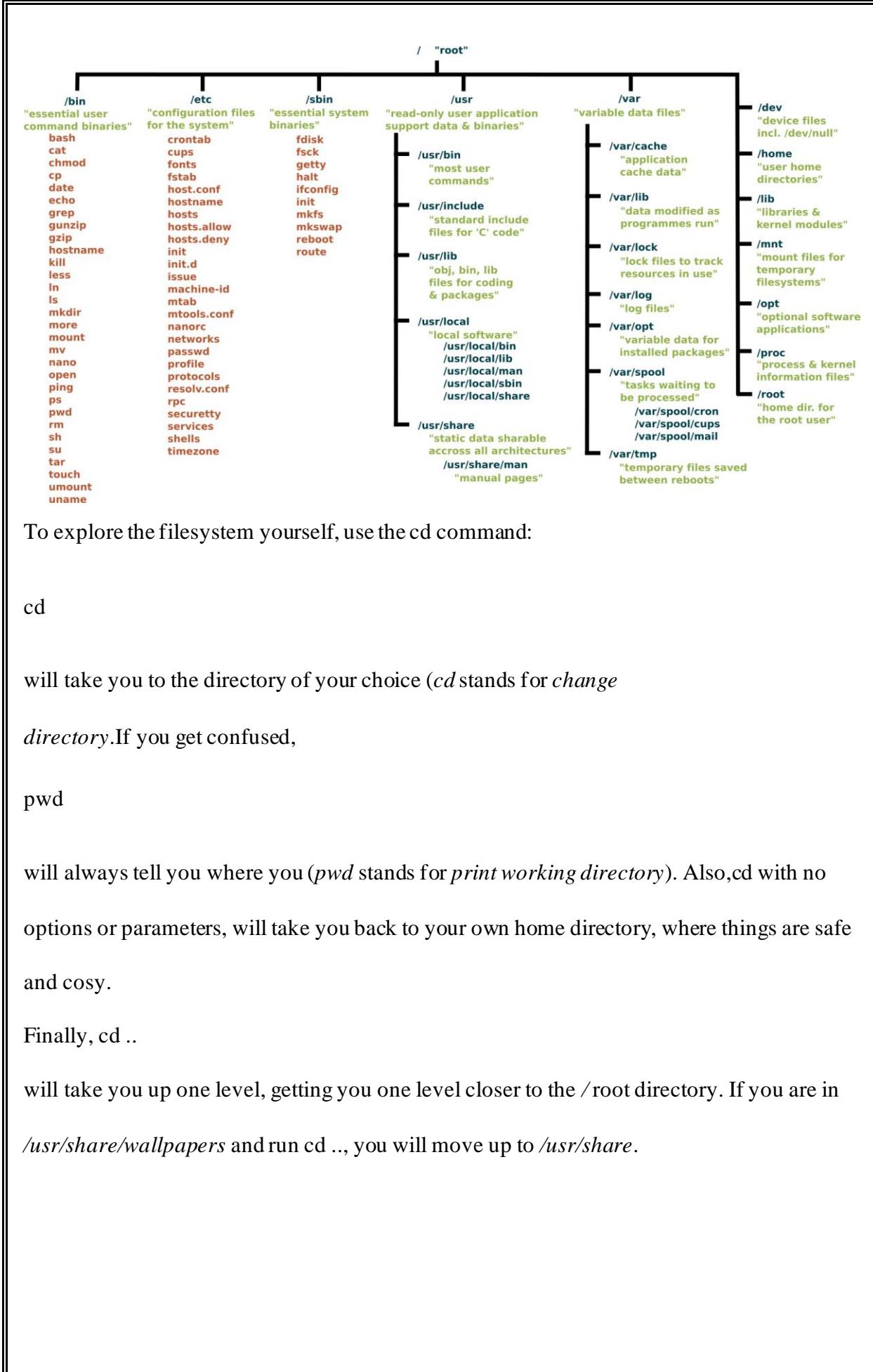
Be that as it may, */var* contains things like logs in the */var/log* subdirectories. Logs are files that register events that happen on the system. If something fails in the kernel, it will be logged in a file in */var/log*; if someone tries to break into your computer from outside, your firewall will also log the attempt here. It also contains *spools* for tasks. These “tasks” can be the jobs you send to a shared printer when you have to wait because another user is printing a long document, or mail that is waiting to be delivered to users on the system.

Your system may have some more directories we haven't mentioned above. In the screenshot, for example, there is a */snap* directory. That's because the shot was captured on an Ubuntu system. Ubuntu has recently incorporated [snap](#) packages as a way of distributing software. The

*/snap* directory contains all the files and the software installed from snaps.

*/var/log/syslog* contains lot of system related logfiles.

That is the root directory covered, but many of the subdirectories lead to their own set of files and subdirectories. Figure 2 gives you an overall idea of what the basic file system tree looks like (the image is kindly supplied under a CC By-SA license by Paul Gardner) and [Wikipedia has a break down with a summary of what each directory is used for.](#)



To see what a directory contains, use

ls

or simply

ls

to list the contents of the directory you are in right now.

And, of course, you always have tree to get an overview of what lays within a directory. Try it on */usr/share* — there is a lot of interesting stuff in there.

Although there are minor differences between Linux distributions, the layout for their filesystems are mercifully similar. So much so that you could say: once you know one, you know them all. And the best way to know the filesystem is to explore it. So go forth with tree, ls, and cd into uncharted territory.

You cannot damage your filesystem just by looking at it, so move from one directory to another and take a look around.

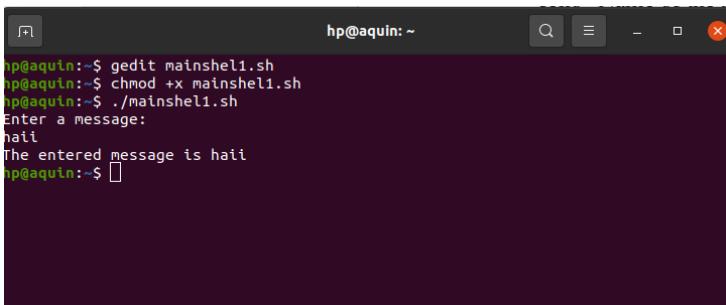
## **EXERCISE 4**

### **SHELL SCRIPT**

1. Write a Shell program to display a given message

```
read name  
echo "Your name is = $name"
```

#### **Output**



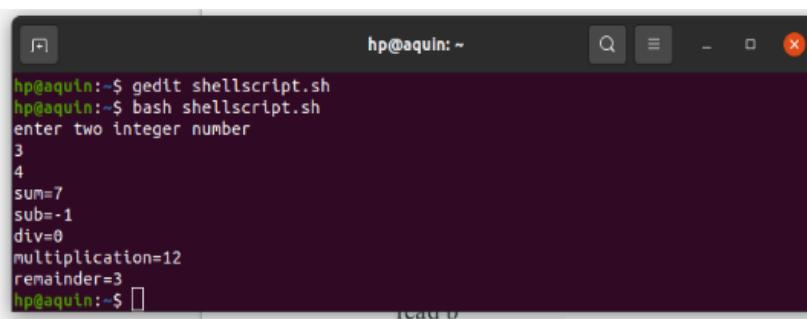
A terminal window titled 'hp@aquin: ~' showing the execution of a shell script. The script reads 'name' from the user, then prints 'Your name is = \$name'. The user enters 'hail' as input, and the script outputs 'The entered message is hail'.

```
hp@aquin:~$ gedit mainshel1.sh  
hp@aquin:~$ chmod +x mainshel1.sh  
hp@aquin:~$ ./mainshel1.sh  
Enter a message:  
hail  
The entered message is hail  
hp@aquin:~$
```

2. Write a shell script to evaluate arithmetic operations.

```
echo "enter two integer number"  
read a  
read b  
c=`expr $a + $b`  
echo "sum=$c"  
c=`expr $a - $b`  
echo "Difference=$c"  
c=`expr $a / $b`  
echo "Division=$c"  
c=`expr $a % $b`  
echo "Remainder=$c"  
c=`expr $a \* $b`  
echo "Multiplication=$c"
```

#### **Output**



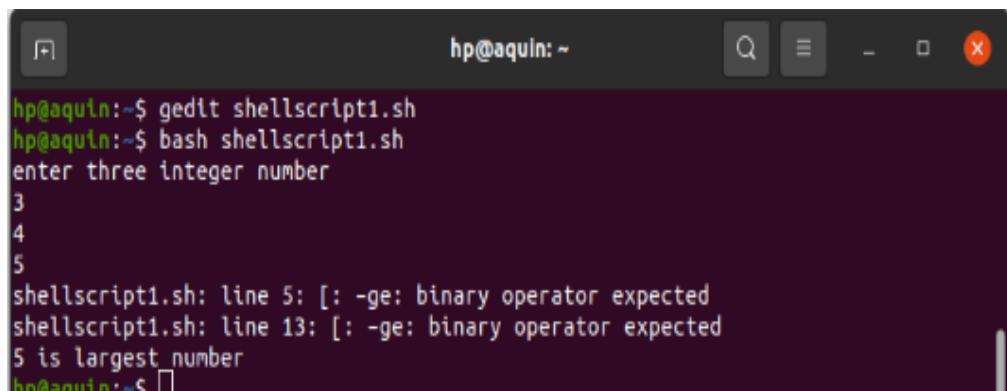
A terminal window titled 'hp@aquin: ~' showing the execution of a shell script. The script prompts for two integers, performs addition, subtraction, division, modulus, and multiplication, and prints the results. The user inputs '3' and '4'.

```
hp@aquin:~$ gedit shellscrip.sh  
hp@aquin:~$ bash shellscrip.sh  
enter two integer number  
3  
4  
sum=7  
sub=-1  
div=0  
multiplication=12  
remainder=3  
hp@aquin:~$
```

3. Write a shell Script to determine largest among three integer number  
echo "Enter numbers"

```
read a
read b
read c
if [ $a -ge $c ]
then
if [ $a -ge $c ]
then
echo "$a is the largest number"
fi
elif [ $b -ge $c ]
then
echo "$b is the largest number"
else
echo "$c is the largest number"
fi
```

### **Output**



A screenshot of a terminal window titled 'hp@aquin:~'. The window shows the following command-line interaction:

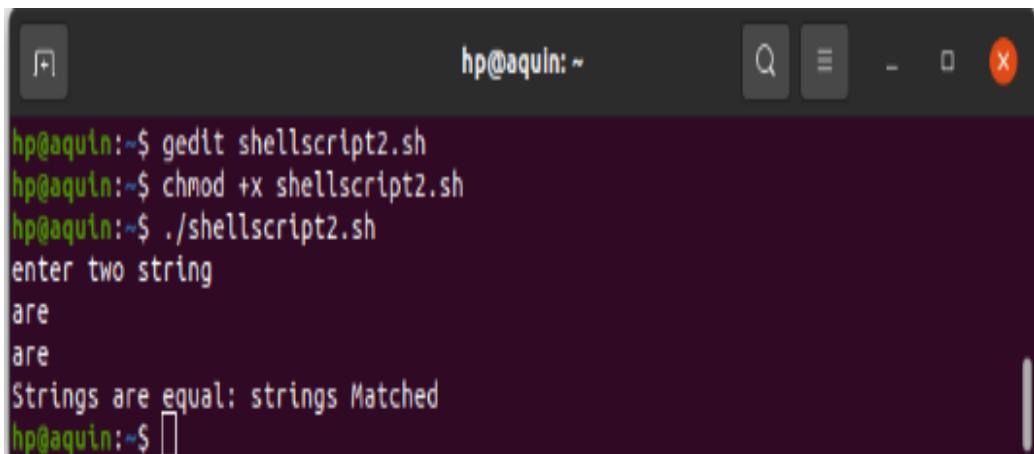
```
hp@aquin:~$ gedit shellscrip1.sh
hp@aquin:~$ bash shellscrip1.sh
enter three integer number
3
4
5
shellscrip1.sh: line 5: [: -ge: binary operator expected
shellscrip1.sh: line 13: [: -ge: binary operator expected
5 is largest number
hp@aquin:~$
```

The terminal window has a dark background and light-colored text. It includes standard window controls (minimize, maximize, close) at the top right.

4. Write a shell script to compare two string.

```
echo "Enter two strings"
read a
read b
if [ -z $a ]
then
echo "First string is empty"
fi
if [ -z $b ]
then
echo "Second string is empty"
fi
if [ $a = $b ]
then
echo "Strings are equal"
else
echo "Strings are not equal"
fi
```

### **Output**



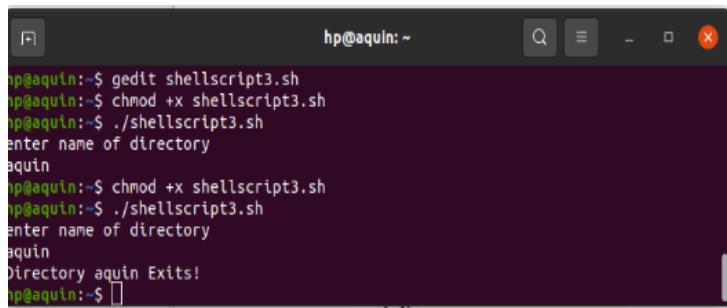
The screenshot shows a terminal window with a dark theme. The title bar says "hp@aquin:~". The terminal window contains the following text:

```
hp@aquin:~$ gedit shellscript2.sh
hp@aquin:~$ chmod +x shellscript2.sh
hp@aquin:~$ ./shellscript2.sh
enter two string
are
are
Strings are equal: strings Matched
hp@aquin:~$
```

5. Write a shell script to read and check the directory exists or not, if not make directory.

```
echo "Enter name of the directory"
read dir
if [ -d $dir ]
then
echo "Directory $dir exists"
else
mkdir $dir
echo "Directory $dir created"
fi
```

### **Output**

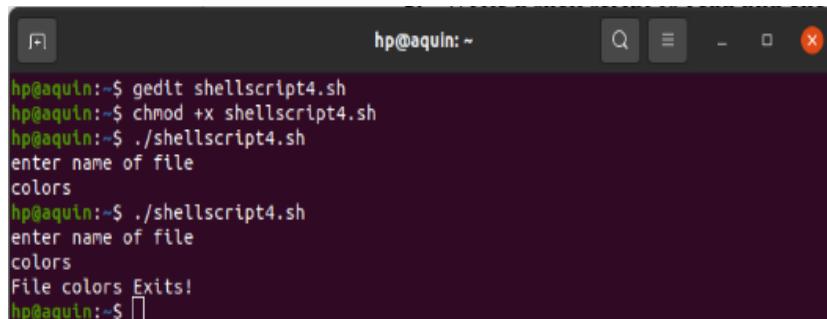


```
hp@aquin:~$ gedit shellscript3.sh
hp@aquin:~$ chmod +x shellscript3.sh
hp@aquin:~$ ./shellscript3.sh
enter name of directory
aquin
hp@aquin:~$ chmod +x shellscript3.sh
hp@aquin:~$ ./shellscript3.sh
enter name of directory
aquin
Directory aquin Exists!
hp@aquin:~$
```

6. Write a shell script to read and check the file exists or not, if not make file.

```
echo "Enter filename"
read filename
if [ -f $filename ]
then
echo "File $filename exists"
else
touch filename
echo "$filename created"
fi
```

### **Output**

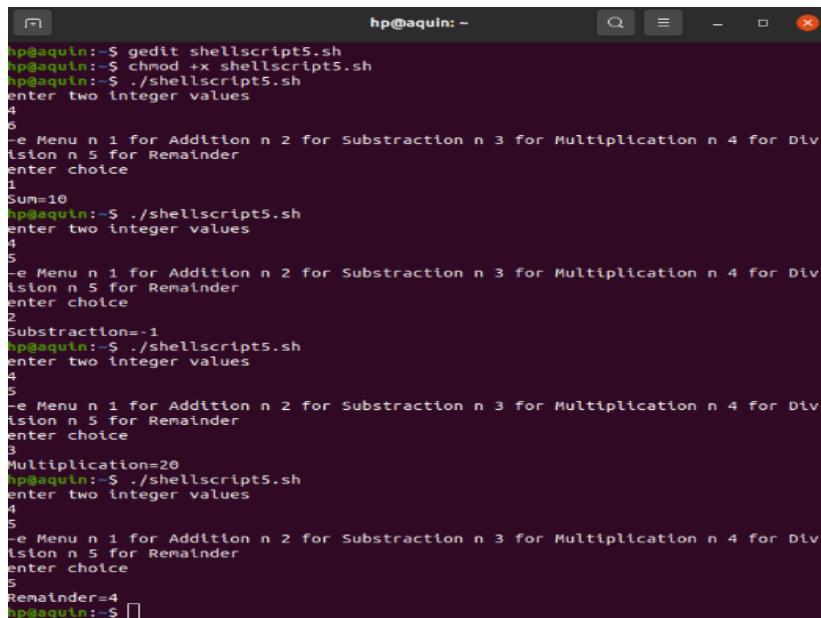


```
hp@aquin:~$ gedit shellscript4.sh
hp@aquin:~$ chmod +x shellscript4.sh
hp@aquin:~$ ./shellscript4.sh
enter name of file
colors
hp@aquin:~$ ./shellscript4.sh
enter name of file
colors
File colors Exists!
hp@aquin:~$
```

7. Write a shell script to implement menu driven program to perform all arithmetic operation using case statement.

```
echo "Enter two numbers"
read a
read b
echo -e "Menu \n 1 for Addition \n 2 for Subtraction \n 3 for Division \n 4 for Remainder \n
      5 Multiplication"
echo "Enter Choice"
read ch
case $ch in
  1) echo "Sum=$(expr $a + $b)";;
  2) echo "Difference=$(expr $a - $b )";;
  3) echo "Division=$(expr $a / $b )";;
  4) echo "Remainder=$(expr $a % $b )";;
  5) echo "Multiplication=$(expr $a \* $b )";;
  *) echo "Invalid Choice..." ;;
esac
```

## Output

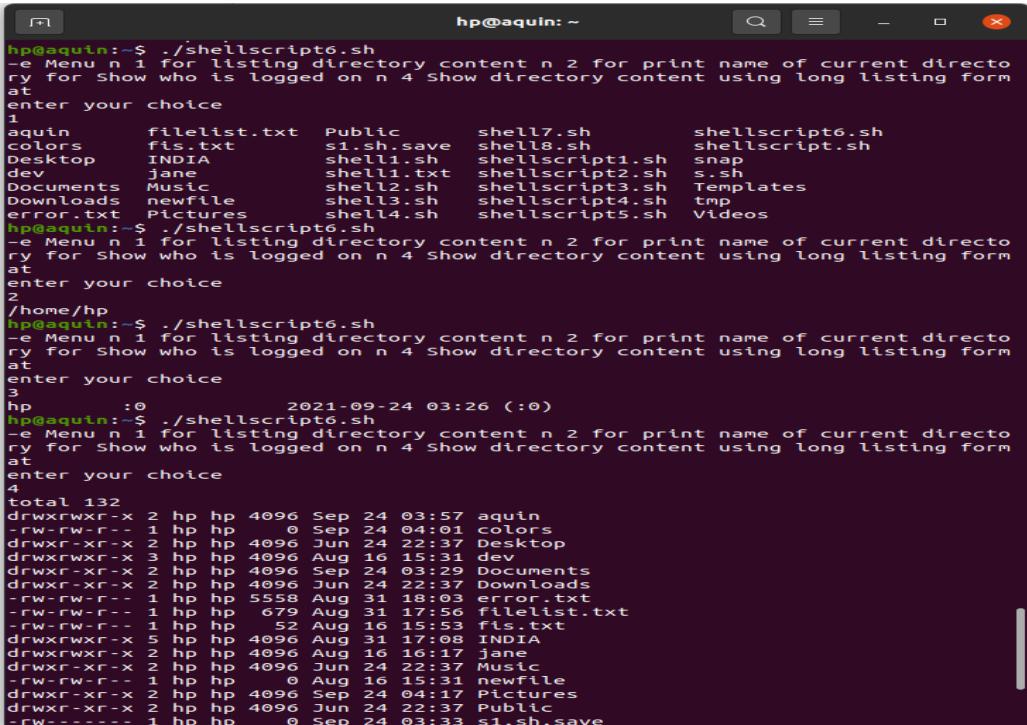


```
hp@aquila:~$ gedit shellscript5.sh
hp@aquila:~$ chmod +x shellscript5.sh
hp@aquila:~$ ./shellscript5.sh
enter two integer values
4
5
-e Menu n 1 for Addition n 2 for Subtraction n 3 for Multiplication n 4 for Div
ision n 5 for Remainder
enter choice
1
Sum=10
hp@aquila:~$ ./shellscript5.sh
enter two integer values
4
5
-e Menu n 1 for Addition n 2 for Subtraction n 3 for Multiplication n 4 for Div
ision n 5 for Remainder
enter choice
2
Subtraction=-1
hp@aquila:~$ ./shellscript5.sh
enter two integer values
4
5
-e Menu n 1 for Addition n 2 for Subtraction n 3 for Multiplication n 4 for Div
ision n 5 for Remainder
enter choice
3
Multiplication=20
hp@aquila:~$ ./shellscript5.sh
enter two integer values
4
5
-e Menu n 1 for Addition n 2 for Subtraction n 3 for Multiplication n 4 for Div
ision n 5 for Remainder
enter choice
5
Remainder=4
hp@aquila:~$
```

8. Write a shell script to do:
- display list of directory contents
  - Name of current directory
  - Who is logged on
  - Long listing of directory contents according to choice of user.

```
echo -e "Menu \n 1 for listing directory content \n 2 for print name of currentdirectory \n 3\n for Show who is logged on \n 4 Show directory content"\n\necho "enter your choice"\n\nread ch\n\ncase $ch in\n1)ls;;\n2)pwd;;\n3)who;;\n4)ls -l;;\n*)echo "Invalid Choice"\nesac
```

## Output

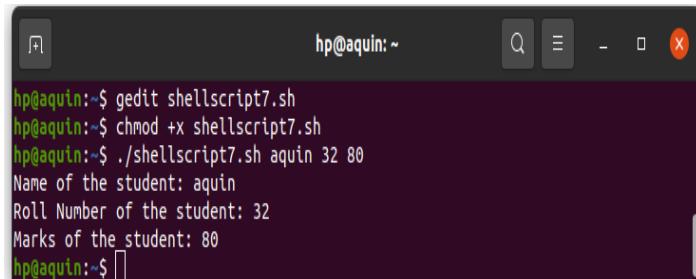


```
hp@aquin:~$ ./shellscript6.sh\n-e Menu n 1 for listing directory content n 2 for print name of current directory for Show who is logged on n 4 Show directory content using long listing form\nat\nenter your choice\n1\naquin    filelist.txt  Public      shell7.sh      shellscript6.sh\ncolors   fis.txt       s1.sh.save  shell8.sh      shellscript.sh\nDesktop  INDIA        shell1.sh    shellscript1.sh  snap\ndev      jane         shell1.txt   shellscript2.sh  s.sh\nDocuments Music        shell2.sh    shellscript3.sh  Templates\nDownloads newfile     shell3.sh    shellscript4.sh  tmp\nerror.txt Pictures     shell4.sh    shellscript5.sh  Videos\nhp@aquin:~$ ./shellscript6.sh\n-e Menu n 1 for listing directory content n 2 for print name of current directory for Show who is logged on n 4 Show directory content using long listing form\nat\nenter your choice\n2\n/home/hp\nhp@aquin:~$ ./shellscript6.sh\n-e Menu n 1 for listing directory content n 2 for print name of current directory for Show who is logged on n 4 Show directory content using long listing form\nat\nenter your choice\n3\nhp@aquin:~$ ./shellscript6.sh\n-e Menu n 1 for listing directory content n 2 for print name of current directory for Show who is logged on n 4 Show directory content using long listing form\nat\nenter your choice\n4\n\ntotal 132\ndrwxrwxr-X 2 hp hp 4096 Sep 24 03:57 aquin\n-rw-rw-r-- 1 hp hp 0 Sep 24 04:01 colors\ndrwxr-xr-x 3 hp hp 4096 Jun 24 22:37 Desktop\ndrwxrwxr-x 3 hp hp 4096 Aug 16 15:31 dev\ndrwxr-xr-x 2 hp hp 4096 Sep 24 03:29 Documents\ndrwxr-xr-x 2 hp hp 4096 Jun 24 22:37 Downloads\n-rw-rw-r-- 1 hp hp 5558 Aug 31 18:03 error.txt\n-rw-rw-r-- 1 hp hp 679 Aug 31 17:56 filelist.txt\n-rw-rw-r-- 1 hp hp 52 Aug 16 15:53 fis.txt\ndrwxrwxr-x 5 hp hp 4096 Aug 31 17:08 INDIA\ndrwxrwxr-x 2 hp hp 4096 Aug 16 16:17 jane\ndrwxr-xr-x 2 hp hp 4096 Jun 24 22:37 Music\n-rw-rw-r-- 1 hp hp 0 Aug 16 15:31 newfile\ndrwxr-xr-x 2 hp hp 4096 Sep 24 04:17 Pictures\ndrwxr-xr-x 2 hp hp 4096 Jun 24 22:37 Public\n-rw----- 1 hp hp 0 Sep 24 03:33 s1.sh.save
```

9. Write a shell script to getting input details like name, roll number and marks and print them using command line arguments

```
cat>Studentdetails  
echo "Name of the student: $1"  
echo "Rollno of the student: $1"  
echo "Mark of the student: $3"
```

### **Output**



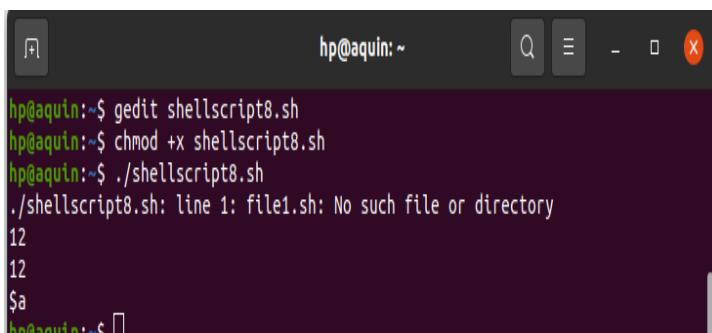
A terminal window titled 'hp@aquin:~' showing the execution of a shell script named 'shellscript7.sh'. The script prompts for 'Name of the student', 'Roll Number of the student', and 'Marks of the student', and prints them back. The output is:

```
hp@aquin:~$ gedit shellscript7.sh  
hp@aquin:~$ chmod +x shellscript7.sh  
hp@aquin:~$ ./shellscript7.sh aquin 32 80  
Name of the student: aquin  
Roll Number of the student: 32  
Marks of the student: 80  
hp@aquin:~$
```

10. Understand the differences between Echo statement using single quote , double quote and without quotes.

```
cat>tenfile.sh  
a=12  
echo $a  
echo "$a"  
echo '$a'
```

### **Output**



A terminal window titled 'hp@aquin:~' showing the execution of a shell script named 'shellscript8.sh'. The script contains three echo statements: one with a variable, one with a quoted variable, and one with an unquoted variable. The output shows that the unquoted variable is expanded to its value, while the quoted variable is printed as is. The output is:

```
hp@aquin:~$ gedit shellscript8.sh  
hp@aquin:~$ chmod +x shellscript8.sh  
hp@aquin:~$ ./shellscript8.sh  
. ./shellscript8.sh: line 1: file1.sh: No such file or directory  
12  
12  
$a  
hp@aquin:~$
```

11. Understand the differences between Echo statement using single quote, double quote and without quotes.

## Output



```
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~ $ cat > a.txt
INDIA is my country.
FISAT is my college.
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~ $ cat a.txt|grep 'FISAT'
FISAT is my college.
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~ $
```

12. Implement arithmetic calculator using Functions

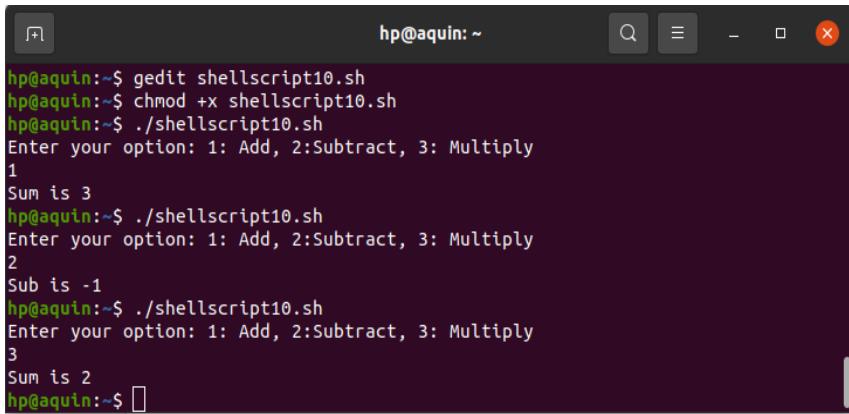
```
add()
{
i=$1
j=$2
((k=i+j))
echo "Sum is $k"
}

sub()
{
i=$1
j=$2
((k=j-i))
echo "Difference is $k"
}

mul()
{
i=$1
j=$2
((k=j*i))
echo "Multiplication is $k"
}

echo "Enter your option--1: Add 2:Subtract 3: Multiply"
read i
case $i in
1)add 2 3;;
2)sub 5 3;;
3)mul 3 3;;
esac
```

## Output



A screenshot of a terminal window titled "hp@aquin: ~". The window shows a shell script named "shellscript10.sh" being run. The script prompts the user for an option (1: Add, 2: Subtract, 3: Multiply) and performs the corresponding arithmetic operation. The output shows three runs of the script with options 1, 2, and 3 respectively.

```
hp@aquin:~$ gedit shellscript10.sh
hp@aquin:~$ chmod +x shellscript10.sh
hp@aquin:~$ ./shellscript10.sh
Enter your option: 1: Add, 2:Subtract, 3: Multiply
1
Sum is 3
hp@aquin:~$ ./shellscript10.sh
Enter your option: 1: Add, 2:Subtract, 3: Multiply
2
Sub is -1
hp@aquin:~$ ./shellscript10.sh
Enter your option: 1: Add, 2:Subtract, 3: Multiply
3
Sum is 2
hp@aquin:~$
```

13. To find the sum of n natural numbers.

a. Using for loop

```
sum=0
for((i=0;i<10;i++))
do
((sum=sum+i))
done
echo $sum
```

## Output



A screenshot of a terminal window titled "hp@aquin: ~". The window shows a shell script named "shellscript11a.sh" being run. The script uses a for loop to calculate the sum of natural numbers from 1 to 10. The output displays the result, which is 45.

```
hp@aquin:~$ gedit shellscript11a.sh
hp@aquin:~$ chmod +x shellscript11a.sh
hp@aquin:~$ ./shellscript11a.sh
45
```

b. Using While loop

```
i=0  
sum=0  
while((i<10));  
do  
((sum=sum+i))  
((i=i+1))  
done  
echo $sum
```

**Output**

```
hp@aquitn:~$ gedit shellscript11b.sh  
hp@aquitn:~$ chmod +x shellscript11b.sh  
hp@aquitn:~$ ./shellscript11b.sh  
45  
hp@aquitn:~$ █
```

## EXERCISE 5

### LAMP SERVER

#### **LAMP INSTALLATION**

##### **Step 1: Update Package Repository Cache**

Before you begin:

1. Open the terminal either by using the **CTRL+ALT+T** keyboard shortcut or by searching for the word *terminal* in **Ubuntu**
2. Make sure to update the package repository cache to ensure it installs the latest versions of the software. To do so, type in the following command:

**sudo apt-get update**

```
hp@hp-HP-Laptop-15s-du0xxx:~$ sudo apt-get update
Get:1 http://in.archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Hit:2 https://screenrec.com/download/ubuntu stable InRelease
Get:3 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [27 .6 kB]
Get:5 http://security.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 Metadata [61.0 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 DEP-11 Metadata [2,464 B]
Get:1 http://in.archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:1 http://in.archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:1 http://in.archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Hit:7 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:8 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease
Fetched 463 kB in 2min 36s (2,977 B/s)
Reading package lists... Done
hp@hp-HP-Laptop-15s-du0xxx:~$ □
```

##### **Step 2: Install Apache**

1. To install Apache, run the following command in the terminal:

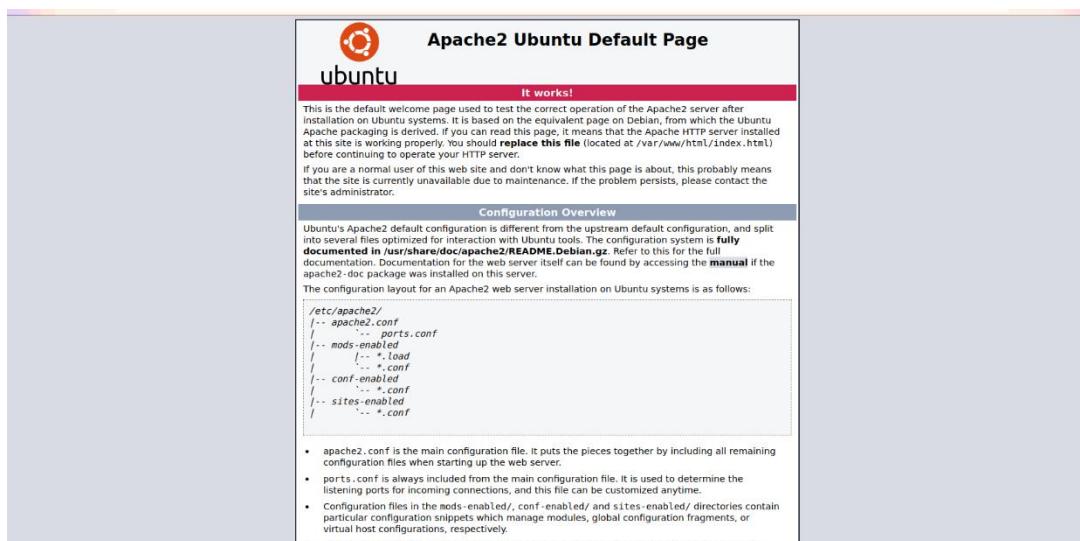
**sudo apt-get install apache2**

```
Reading package lists... Done
hp@hp-HP-Laptop-15s-du0xxx:~$ sudo apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  enchant geoip-database libbind9-161 libboost-filesystem1.67.0
  libboost-iostreams1.67.0 libdns-export1107 libdns1107 libdns1109 libenchant1c2a
  libexiv2-14 libfprint0 libgeoip1 libgspell-1-1 libgutenprint-common
  libgutenprint9 libiptc0 libirs161 libisc-export1104 libisc1104 libisc1105
  libisccc161 libisccfg163 libl1vm9 liblwres161 libnfs12 liboauth0
  printer-driver-gutenprint python3-asn1crypto shim ubuntu-software
  ubuntu-system-service
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0
0 upgraded, 9 newly installed, 0 to remove and 66 not upgraded.
Need to get 1,819 kB of archives.
After this operation, 7,938 kB of additional disk space will be used.
Do you want to continue? [Y/n] □
```

Press **y** (yes) and hit **ENTER** to permit the installation.

2. To ensure Apache is running, enter the Localhost of your server in the address bar and press **ENTER**.

The test Apache web server page should display as below.



Step 3: Install PHP,  
To install PHP, run the following command:

**\$ sudo apt-get install php7.4**

```
hp@hp-HP-Laptop-15s-du0xxx:~$ sudo apt-get install php7.4
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  enchant geopip-database libbind9-161 libboost filesystem1.67.0 libboost iostreams1.67.0 libdns-export1107 libdns1107 libdns1109 libenchantic2a libexiv2-14 libfprint0 libgeoip1 libgsspell-1-1
  libgutenprint-common libgutenprint9 libiptc0 libirs161 libisc-export1104 libisc1104 libisc1105 libisccc161 libiscfg163 liblvm9 liblwres161 libnfs12 liboauth0 printer-driver-gutenprint
  python3-asnircrypto shim ubuntu-software ubuntu-system-service
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libapache2-mod-php7.4 php-common php7.4-cli php7.4-common php7.4-json php7.4-opcache php7.4-readline
Suggested packages:
  php-pear
The following NEW packages will be installed:
  libapache2-mod-php7.4 php-common php7.4-cli php7.4-common php7.4-json php7.4-opcache php7.4-readline
0 upgraded, 8 newly installed, 0 to remove and 66 not upgraded.
Need to get 4,815 kB of archives.
After this operation, 18.0 MB of additional disk space will be used.
Do you want to continue? [Y/n] ■
```

Press **y** and **ENTER** to allow the installation.

Step 4: Restart Apache

After the php installation you must restart the Apache service.

Enter the command:

**\$ sudo /etc/init.d/apache2 restart**

Step 5: Test PHP Processing on Web Server

1. Create a basic **PHP script** and save it to the “web root” directory. This is necessary for Apache to find and serve the file correctly. This directory is located at **/var/www/html/**.

To create a file in that directory, type in the following command:

**\$ sudo nano /var/www/html/test.php**

```
hp@hp-HP-Laptop-15s-du0xxx:~$ sudo nano /var/www/html/test.php
[sudo] password for hp:
hp@hp-HP-Laptop-15s-du0xxx:~$ ■
```

This command opens the **bank file**.

2. Inside the file, type in the valid PHP code:

```
<?php  
Echo " test php ";  
?>
```

3. Press **CTRL + X** to save and close the file. Press **y** and **ENTER** to confirm.

4. Then check the code are run correctly in php.open the browser enter the Ip address (localhost/test.php).

It show the below image



#### Step 6: Install Mysql server

1. To install Mysql server, run the following command:

```
$ sudo apt-get install mysql-server
```

```
OTHER OPTIONS.
hp@hp-HP-Laptop-15s-du0xxx:~$ sudo apt-get install mysql-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  enchant geoip-database libbind9-161 libboost-filesystem1.67.0
  libboost-iostreams1.67.0 libdns-export1107 libdns1107 libdns1109
  libenchant1c2a libexiv2-14 libfprint0 libgeoip1 libgspell-1-1
  libgutenprint-common libgutenprint9 libiptc0 libirs161 libisc-export1104
  libisc1104 libisc1105 libisccc161 libisccfg163 liblvm9 liblwres161 libnfs12
  liboauth0 printer-driver-gutenprint python3-asn1crypto shim ubuntu-software
  ubuntu-system-service
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libaio1 libcgi-fast-perl libcgi-pm-perl libevent-core-2.1-7
  libevent-pthreads-2.1-7 libfcgi-perl libhtml-template-perl libmecab2
  mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0
  mysql-client-core-8.0 mysql-server-8.0 mysql-server-core-8.0
Suggested packages:
```

2. Then it's asking us for a root password . enter the password . Again we get to repeat it

#### Step 7: Check the Mysql server

1. To check Mysql server, run the following command

```
$ mysql -u root -p
```

Enter the root password and press enter

```
hp@hp-HP-Laptop-15s-du0xxx:~$ mysql -u root -p
Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
hp@hp-HP-Laptop-15s-du0xxx:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 19
Server version: 8.0.26-0ubuntu0.20.04.2 (Ubuntu)

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

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

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

mysql> create database testdb;
Query OK, 1 row affected (0.01 sec)

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

mysql> |
```

2 . Create a database testdb and show it

Enter the command

Create database testdb;

Show databases;

So mysql is working then exit the mysql prompt just enter **exit;**

### Step 8: Install PHP Myadmin

1. To install PHP Myadmin, run the following command:

**\$ sudo apt-get install phpmyadmin**

```
hp@hp-HP-Laptop-15s-du0xxx:~$ sudo apt-get install phpmyadmin
[sudo] password for hp:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  acl app-color-data enchant geoip-database gnome-control-center-faces gsfonts hplip-data libbind9-161 libboost-filesystem1.67.0 libboost-iostreams1.67.0 libcolord-gtk1
  libcolorhug2 libdns-export1107 libdns1107 libenchant1c2a libexiv2-14 libfprint0 libgeoip1 libgsound0 libgspell-1-1 libgssdp-1.2-0 libgnupnp-1.2-0 libgnupnp-av-1.0-2 libgnupnp-dlna-2.0-3
  libgbpcommon libgbtunprint9 libieee1284-3 libimagequant0 libiptc0 libirs161 libisc-export104 libisc1104 libisc1105 libiscfg163 liblvm9 liblwres161 libnfs12 libauth0
  librgel-core-2.6-2 librgel-renderer-2.6-2 librgel-server-2.6-2 libsane-common libbsmp-base libwebpnu3 mobile-broadband-provider-info network-manager-gnome
  printer-driver-gutenprint printer-driver-postscript-hp python3-asn1crypto python3-macaroonbakery python3-olefile python3-pil python3-protobuf python3-pynacaroons python3-reportlab
  python3-reportlab-accel python3-rfc3399 python3-tz rygel shim ubuntu-software ubuntu-system-service
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  dbconfig-common dbconfig-mysql libjs-jquery libjs-openlayers libjs-sphinxdoc libjs-underscore libonig5 libzip5 php-b22 php-curl php-gd php-google-recaptcha
  php-mbstring php-mysql php-phpmyadmin-notranslator php-phpmyadmin-shapefile php-phpmyadmin-sql-parser php-psr-cache php-psr-container php-psr-log php-symfony-cache
  php-symfony-cache-contracts php-symfony-expression-language php-symfony-service-contracts php-symfony-var-exporter php-tcpdf php-twigi php-twigi-extensions php-xml php-zip php7.4-b22 php7.4-curl
  php7.4-gd php7.4-mbstring php7.4-mysql php7.4-xml php7.4-zip
Suggested packages:
  php-base php-libodium php-ncrypt php-gmp php-symfony-service-implementation php-inagick php-twigi-doc php-symfony-translation php-recode php-gd2 php-pragmarx-google2fa php-bacon-qr-code
  php-symfony-uzf php-server
Recommended packages:
  php-mcrypt
The following NEW packages will be installed:
  dbconfig-common dbconfig-mysql libjs-jquery libjs-openlayers libjs-sphinxdoc libjs-underscore libonig5 libzip5 php-b22 php-curl php-gd php-google-recaptcha
  php-mbstring php-mysql php-phpmyadmin-notranslator php-phpmyadmin-shapefile php-phpmyadmin-sql-parser php-psr-cache php-psr-container php-psr-log php-symfony-cache
  php-symfony-cache-contracts php-symfony-expression-language php-symfony-service-contracts php-symfony-var-exporter php-tcpdf php-twigi php-twigi-extensions php-xml php-zip php7.4-b22 php7.4-curl
  php7.4-gd php7.4-mbstring php7.4-mysql php7.4-xml php7.4-zip phpmyadmin
0 upgraded, 41 newly installed, 0 to remove and 61 not upgraded.
Need to get 16.0 MB of archives.
After this operation, 71.8 MB of additional disk space will be used.
Do you want to continue? [Y/n] 
```

Press **y** and **ENTER** to allow the installation

1. Then its ask what type of server, we have Apache2 is set by default that's what we want then press ok
  2. Then a configuration prompt are open . here we're going to just choose yes and then it ask the input password for phpmyadmin
  3. Then check it correct . go to the localhost/phpmyadmin. Here we can not found it so
  4. We have to actually edit the file php is located in Apache2 folder.
  5. Enter the following command to edit the file  
**\$ sudo nano/etc/php7.4/apache2.php.ini**
  6. Then we need to uncomment an **extension=mysql.so.** find it the file just remove the Semicolon.
7. Then enter **ctrl+x** to save

### Step 9: Restart Apache

After the php installation you must restart the Apache service.

Enter the command:

```
$ sudo /etc/init.d/apache2 restart
```

### Step 9.1:Include phpmyadmin in apache configuration

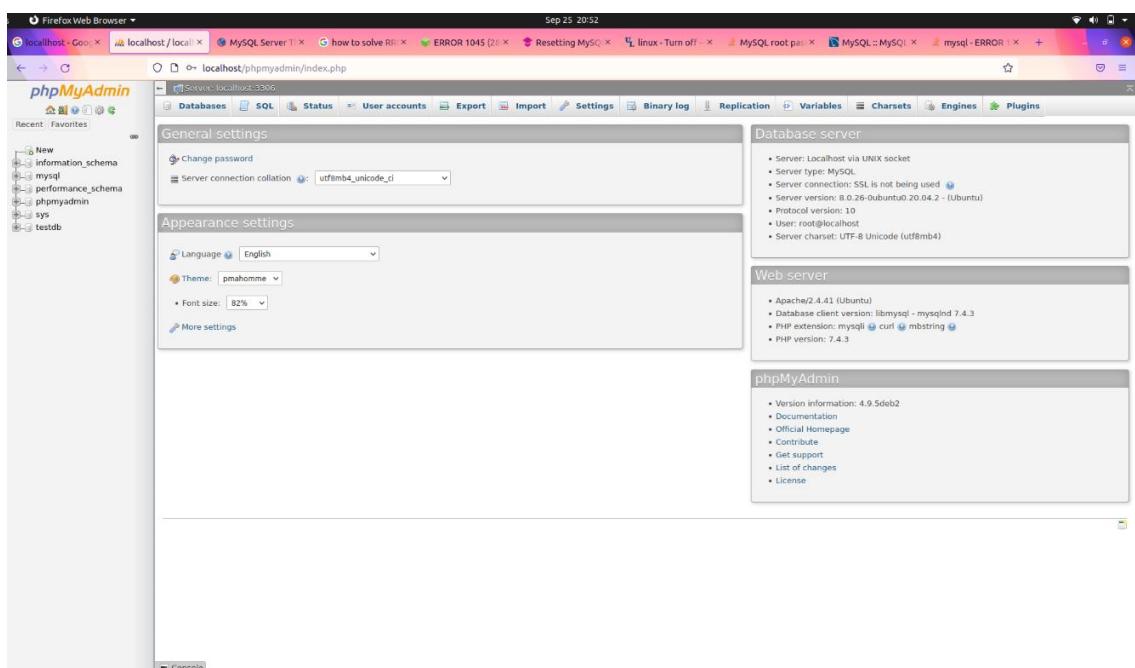
1. Enter the command:

```
$ sudo nano/etc/apache2/apache2.conf
```

2. Type the following command to the nano editor

**Include /etc/phpmyadmin/apache.conf**

3. Then enter **ctrl+x** to save
4. Then again restart the apache



## **EXERCISE 6**

### **LARAVEL SERVER**

#### **Laravel installation On Ubuntu with Apache**

##### **Step 1 – Install Apache Web Server**

Let's open up a Terminal and do first thing first update your package list using Sudo apt update command.

**\$ sudo apt update**

After updating your package list install apache webserver

**\$ sudo apt install apache2**

**\$ systemctl status apache2**

```
ebin@ebin-VirtualBox:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  linux-headers-5.11.0-25-generic linux-hwe-5.11-headers-5.11.0-25 linux-image-5.11.0-25-generic linux-modules-5.11.0-25-generic
  linux-modules-extra-5.11.0-25-generic
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following packages will be upgraded:
  apache2 apache2-bin apache2-data apache2-utils
4 to upgrade, 0 to newly install, 0 to remove and 78 not to upgrade.
Need to get 1,518 kB of archives.
After this operation, 4,096 B of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 apache2 amd64 2.4.41-4ubuntu3.5 [95.5 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 apache2-bin amd64 2.4.41-4ubuntu3.5 [1,180 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 apache2-data all 2.4.41-4ubuntu3.5 [159 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 apache2-utils amd64 2.4.41-4ubuntu3.5 [84.2 kB]
Fetched 1,518 kB in 1s (1,216 kB/s)
```

Now, check the status of apache server whether it is running or not.

If the Apache server not running then use the following command to start apache serve and add to boot startup.

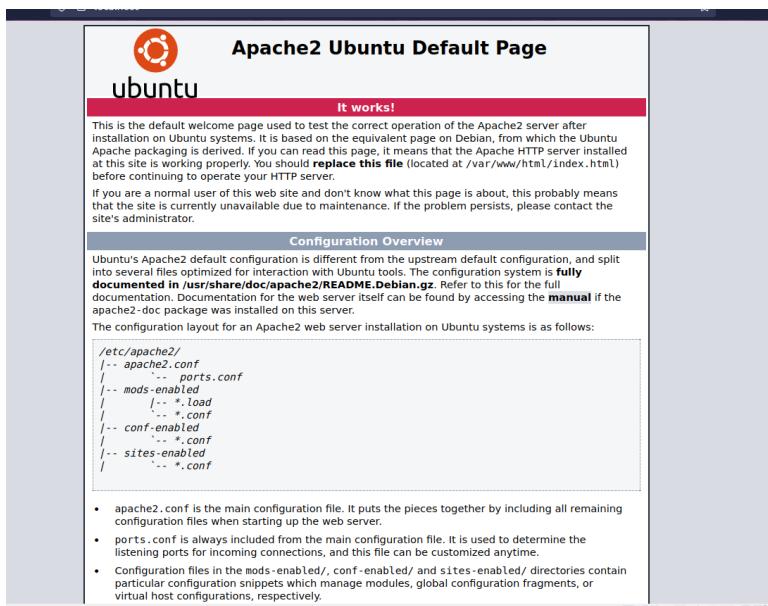
**\$ systemctl start apache2**

**\$ systemctl enable apache2**

```
ebin@ebin-VirtualBox:~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2021-09-28 13:53:55 IST; 6min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 2749 (apache2)
      Tasks: 7 (limit: 4650)
     Memory: 14.8M
        CPU: 0.000 CPU(s) since start
       CGroup: /system.slice/apache2.service
               ├─2749 /usr/sbin/apache2 -k start
               ├─2750 /usr/sbin/apache2 -k start
               ├─2753 /usr/sbin/apache2 -k start
               ├─2754 /usr/sbin/apache2 -k start
               ├─2755 /usr/sbin/apache2 -k start
               ├─2756 /usr/sbin/apache2 -k start
               └─2757 /usr/sbin/apache2 -k start

Sep 28 13:53:55 ebin-VirtualBox systemd[1]: Starting The Apache HTTP Server...
Sep 28 13:53:55 ebin-VirtualBox apachectl[2747]: AH00558: apache2: Could not reliably determine the server's fully qualified name, using ebin-VirtualBox for Port 80
Sep 28 13:53:55 ebin-VirtualBox systemd[1]: Started The Apache HTTP Server.
lines 1-19/19 (END)
```

Open browser, goto localhost and check if default apache server page is available or not



## Step 2 – Install and Configure PHP 7.4

To install Laravel 8.x, at least you must have PHP  $\geq 7.3$  on your system. And by default, the official Ubuntu 20.04 repository provides PHP 7.4 packages. Install PHP 7.4 packages using the apt command below.

```
$ sudo apt install libapache2-mod-php php php-common php-xml php-gd php-opcache  
php-mbstring php-tokenizer php-json php-bcmath php-zip unzip
```

```
ebin@ebin-VirtualBox:~$ sudo apt install libapache2-mod-php php php-common php-xml php-gd php-opcache php-mbstring php-tokenizer php-jis
on php-bcmath php-zip unzip
[sudo] password for ebin:
Reading package lists... Done
Building dependency tree
Reading state information... Done
Package php-opcache is a virtual package provided by:
  php8.1-opcache 8.1.0~rc2-1+ubuntu20.04.1+deb.sury.org+1
  php8.0-opcache 8.0.11-1+ubuntu20.04.1+deb.sury.org+1
  php7.4-opcache 7.4.24-1+ubuntu20.04.1+deb.sury.org+1
  php7.3-opcache 7.3.31-1+ubuntu20.04.1+deb.sury.org+1
  php7.2-opcache 7.2.34-24+ubuntu20.04.1+deb.sury.org+1
  php7.1-opcache 7.1.33-41+ubuntu20.04.1+deb.sury.org+1
  php7.0-opcache 7.0.33-54+ubuntu20.04.1+deb.sury.org+1
  php5.6-opcache 5.6.40-54+ubuntu20.04.1+deb.sury.org+1
You should explicitly select one to install.
```

```
ebin@ebin-VirtualBox:~$ php7.4-dev php7.4-zip php7.4-mbstring php7.4-mysql php7.4-xml curl -y
php7.4-dev: command not found
ebin@ebin-VirtualBox:~$ sudo apt install php7.4 libapache2-mod-php7.4 php7.4-curl php-pear php7.4-gd php7.4-dev php7.4-zip php7.4-mbstr
ing php7.4-mysql php7.4-xml curl -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
 linux-headers-5.11.0-25-generic linux-hwe-5.11-headers-5.11.0-25 linux-image-5.11.0-25-generic linux-modules-5.11.0-25-generic
 linux-modules-extra-5.11.0-25-generic
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
 autoconf automake autopoint autotools-dev debhelper dh-autoreconf dh-strip-nondeterminism dwz gettext intltool-debian
 libarchive-cpio-perl libarchive-zip-perl libcrc03_ll libdebhelper-perl libfile-stripnondeterminism-perl libltdl-dev
 libmail-sendmail-perl libpcre2-16-0 libpcre2-32-0 libpcre2-8-0 libpcre2-dev libpcre2-posix2 libsigsegv2 libssl-dev
 libsub-override-perl libsys-hostname-long-perl libtinfo m4 php7.4-cli php7.4-common php7.4-json php7.4-opcache php7.4-readline
 pkg-php-tools po-debconf shtool
Suggested packages:
 autoconf-archive gnu-standards autoconf-doc dh-make gettext-doc libasprintf-dev libgettextpo-dev libtool-doc libssl-doc gfortran
 | fortran95-compiler gcj-jdk m4-doc dh-php libmail-box-perl
The following NEW packages will be installed
 autoconf automake autopoint autotools-dev curl debhelper dh-autoreconf dh-strip-nondeterminism dwz gettext intltool-debian
 libapache2-mod-php7.4 libarchive-cpio-perl libarchive-zip-perl libcrc03_ll libdebhelper-perl libfile-stripnondeterminism-perl
 libltdl-dev libmail-sendmail-perl libpcre2-16-0 libpcre2-dev libpcre2-posix2 libsigsegv2 libssl-dev libsub-override-perl
 libsys-hostname-long-perl libtinfo m4 php7.4-pear php7.4 php7.4-common php7.4-curl php7.4-dev php7.4-gd php7.4-json
 php7.4-mbstring php7.4-mysql php7.4-opcache php7.4-readline php7.4-xml php7.4-zip pkg-php-tools po-debconf shtool
The following packages will be upgraded:
 libpcre2-32-0 libpcre2-8-0
2 to upgrade, 45 to newly install, 0 to remove and 76 not to upgrade.
Need to get 13.0 MB of archives.
After this operation, 54.0 MB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu focal/main amd64 libtsigsegv2 amd64 2.12-2 [13.9 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu focal/main amd64 m4 amd64 1.4.18-4 [199 kB]
Get:3 http://ppa.launchpad.net/ondrej/php/ubuntu focal/main amd64 libpcre2-8-0 amd64 10.36-2+ubuntu20.04.1+deb.sury.org+2 [201 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu focal/main amd64 autoconf all 2.69-11.1 [321 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu focal/main amd64 autotools-dev all 20180224.1 [39.6 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu focal/main amd64 automake all 1:1.16.1-4ubuntu0 [522 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu focal/main amd64 autopoint all 0.19.8.1-10build1 [412 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 curl amd64 7.68.0-1ubuntu2.7 [161 kB]
```

Now go ahead and make tweak changes in PHP ini file and set `cgi.fix_pathinfo` set to be 0. If this number is kept as a 1, the php interpreter will do its best to process the file that is as near to the requested file as possible. This is a possible security risk. If this number is set to 0, conversely, the interpreter will only process the exact file path—a much safer alternative.

```
$ cd /etc/php/7.4/apache2
```

```
$ sudo nano php.ini
```

Press **ctrl+w** and search for the word “cgi.fix” the uncomment the line and set it to 0.

...  
- ein fiktives mathinf - 0

6

```
ebin@ebin-VirtualBox:~$ cd /etc/php
ebin@ebin-VirtualBox:/etc/php$ ls
7.4 8.0
ebin@ebin-VirtualBox:/etc/php$ cd 7.4/
ebin@ebin-VirtualBox:/etc/php/7.4$ ls
apache2 cli mods-available
ebin@ebin-VirtualBox:/etc/php/7.4$ cd apache2/
ebin@ebin-VirtualBox:/etc/php/7.4/apache2$ ls
conf.d php.ini
ebin@ebin-VirtualBox:/etc/php/7.4/apache2$ sudo nano php.ini
ebin@ebin-VirtualBox:/etc/php/7.4/apache2$ sudo nano php.ini
ebin@ebin-VirtualBox:/etc/php/7.4/apache2$ █
```

```
GNU nano 4.8                                     php.ini
; **You CAN safely turn this off for IIS, in fact, you MUST.**
; Thunderbird Mail et/cgi.force-redirect
;cgi.force_redirect = 1

; if cgi.nph is enabled it will force cgi to always sent Status: 200 with
; every request. PHP's default behavior is to disable this feature.
;cgi.nph = 1

; if cgi.force_redirect is turned on, and you are not running under Apache or Netscape
; (iPlanet) web servers, you MAY need to set an environment variable name that PHP
; will look for to know it is OK to continue execution. Setting this variable MAY
; cause security issues, KNOW WHAT YOU ARE DOING FIRST.
; http://php.net/cgi.redirect-status-env
; http://php.net/cgi.redirect-status-env
;cgi.redirect_status_env =

; cgi.fix_pathinfo provides *real* PATH_INFO/PATH_TRANSLATED support for CGI. PHP's
; previous behaviour was to set PATH_TRANSLATED to SCRIPT_FILENAME, and to not grok
; what PATH_INFO is. For more information on PATH_INFO, see the cgi specs. Setting
; this to 1 will cause PHP CGI to fix its paths to conform to the spec. A setting
; of zero causes PHP to behave as before. Default is 1. You should fix your scripts
; to use SCRIPT_FILENAME rather than PATH_TRANSLATED.
; http://php.net/cgi.fix-pathinfo
cgi.fix_pathinfo=0

; if cgi.discard_path is enabled, the PHP CGI binary can safely be placed outside
; of the web tree and people will not be able to circumvent .htaccess security.
;cgi.discard_path=1

; FastCGI under IIS supports the ability to impersonate
; security tokens of the calling client. This allows IIS to define the
; security context that the request runs under. mod_fastcgi under Apache
```

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

Press Ctrl + x then y to Save and Exit.

Now Restart The apache service.

```
$ systemctl restart apache2
```

### **Step 3 – Install Composer PHP Packages Management**

Install the composer package manager go ahead and download and install Composer. and move the composer .phar file to usr/local/bin/composer directory.

```
$ sudo apt install curl
```

```
$ curl -sS https://getcomposer.org/installer | php
```

```
$ sudo mv composer.phar /usr/local/bin/composer
```

```
ebin@ebin-VirtualBox:~$ sudo apt install curl
Reading package lists... Done
Building dependency tree
Reading state information... Done
curl is already the newest version (7.68.0-1ubuntu2.7).
The following packages were automatically installed and are no longer required:
  linux-headers-5.11.0-25-generic linux-hwe-5.11-headers-5.11.0-25 linux-image-5.11.0-25-generic linux-modules-5.11.0-25-ge
  linux-modules-extra-5.11.0-25-generic
Use 'sudo apt autoremove' to remove them.
0 to upgrade, 0 to newly install, 0 to remove and 76 not to upgrade.
ebin@ebin-VirtualBox:~$ curl -sS https://getcomposer.org/installer | php
All settings correct for using Composer
Downloading...

Composer (version 2.1.8) successfully installed to: /home/ebin/composer.phar
Use it: php composer.phar

ebin@ebin-VirtualBox:~$ sudo mv composer.phar /usr/local/bin/composer
ebin@ebin-VirtualBox:~$ 
```

```
ebin@ebin-VirtualBox:~$ composer --version
Composer version 2.1.8 2021-09-15 13:55:14
ebin@ebin-VirtualBox:~$ 
```

#### **Step 4 – Install Laravel 8.x on Ubuntu 20.04**

Now install Laravel Framework using composer, just type composer global require Laravel/installer It will take a while to complete download its dependencies.

```
ebin@ebin-VirtualBox:~$ composer global require laravel/installer
Changed current directory to /home/ebin/.config/composer
Using version ^4.2 for laravel/installer
./composer.json has been created
Running composer update laravel/installer
Loading composer repositories with package information
Updating dependencies
Lock file operations: 13 installs, 0 updates, 0 removals
- Locking laravel/installer (v4.2.8)
- Locking psr/container (1.1.1)
- Locking symfony/console (v5.3.7)
- Locking symfony/deprecation-contracts (v2.4.0)
- Locking symfony/polyfill-ctype (v1.23.0)
- Locking symfony/polyfill-intl-grapheme (v1.23.1)
- Locking symfony/polyfill-intl-normalizer (v1.23.0)
- Locking symfony/polyfill-mbstring (v1.23.1)
- Locking symfony/polyfill-php73 (v1.23.0)
- Locking symfony/polyfill-php80 (v1.23.1)
- Locking symfony/process (v5.3.7)
- Locking symfony/service-contracts (v2.4.0)
- Locking symfony/string (v5.3.7)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 13 installs, 0 updates, 0 removals
- Downloading symfony/polyfill-php80 (v1.23.1)
- Downloading symfony/process (v5.3.7)
```

As you had seen above image, all packages have been installed on the ‘`~/.config/composer`’ directory. Next, we need to add the ‘`bin`’ directory to the PATH environment through the `~/.bashrc` configuration. So Now Edit the `~/.bashrc` configuration using nano command.

`$ nano ~/.bashrc`

And add the following line at the end of the file.

```
...
export PATH="$HOME/.config/composer/vendor/bin:$PATH"
...
```

```

GNU nano 4.8                               /home/ebin/.bashrc
# Add an "alert" alias for long running commands. Use like so:
# sleep 10; alert
alias alert='notify-send --urgency=low -i "$( [ $? = 0 ] && echo terminal ||

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi
export PATH="$HOME/.config/composer/vendor/bin:$PATH"

```

Press Ctrl + x then y to Save and Exit.

Now reload your bashrc configuration using the source command.

```
$ source ~/.bashrc
```

Now echo \$PATH. It will return your “Bin” directory path for the Composer package.

```
$ echo $PATH
```

```

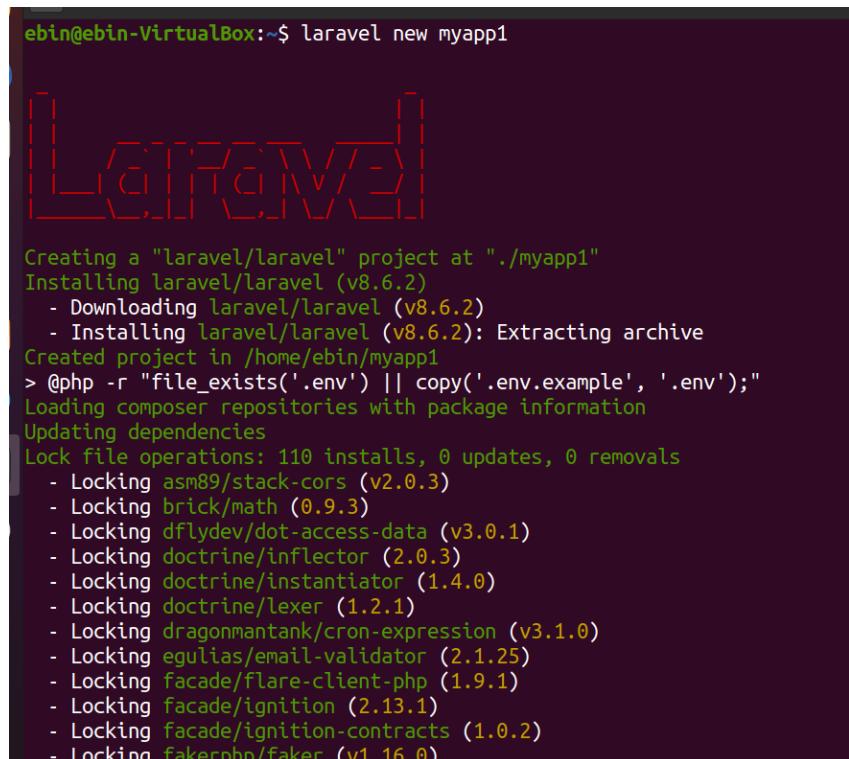
ebin@ebin-VirtualBox:~$ nano ~/.bashrc
ebin@ebin-VirtualBox:~$ nano ~/.bashrc
ebin@ebin-VirtualBox:~$ source ~/.bashrc
ebin@ebin-VirtualBox:~$ echo $PATH
/home/ebin/.config/composer/vendor/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/local/games:/snap/bin
ebin@ebin-VirtualBox:~$ 

```

The ‘bin’ directory for the composer packages has been added to the \$PATH environment variable. And as a result, you can use the command ‘laravel’ to start and create a new project. Now go ahead and type Laravel new then your project name to start a new Laravel project.

**\$ laravel new myapp1**

This will take a while to download all dependencies required by Laravel.



```
ebin@ebin-VirtualBox:~$ laravel new myapp1
[laravel logo]
Creating a "laravel/laravel" project at "./myapp1"
Installing laravel/laravel (v8.6.2)
- Downloading laravel/laravel (v8.6.2)
- Installing laravel/laravel (v8.6.2): Extracting archive
Created project in /home/ebin/myapp1
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 110 installs, 0 updates, 0 removals
- Locking asm89/stack-cors (v2.0.3)
- Locking brick/math (0.9.3)
- Locking dflydev/dot-access-data (v3.0.1)
- Locking doctrine/inflector (2.0.3)
- Locking doctrine/instantiator (1.4.0)
- Locking doctrine/lexer (1.2.1)
- Locking dragonmantank/cron-expression (v3.1.0)
- Locking egulias/email-validator (2.1.25)
- Locking facade/flare-client-php (1.9.1)
- Locking facade/ignition (2.13.1)
- Locking facade/ignition-contracts (1.0.2)
- Locking fakerphp/faker (v1.16.0)
```

Here you can see the installation of my new project myapp1 finished. You can also see inside my home directory a new directory has been created with my project name.

### **Step 5 – Finally Configure Apache for Laravel and test it**

First, add your project directory to www-data group use the following command

**\$ sudo chgrp -R www-data /home/ebin/myapp1**

-R flag is recursive, Recursive means all subdirectory and files under your project directory become changed to the “www-data” group.

Also, you need to change access permission 775 of the storage directory under your project. So, go ahead and use the following command.

**\$ sudo chmod -R 775 /home/ebin/myapp1/storage**

```
ebin@ebin-VirtualBox:~$ cd home
bash: cd: home: No such file or directory
ebin@ebin-VirtualBox:~$ pwd
/home/ebin
ebin@ebin-VirtualBox:~$ /home/ebin/myapp1
bash: /home/ebin/myapp1: Is a directory
ebin@ebin-VirtualBox:~$ cd /home/ebin/myapp1
ebin@ebin-VirtualBox:~/myapp1$ cd ..
ebin@ebin-VirtualBox:~$ sudo chgrp -R www-data /home/ebin/myapp1
[sudo] password for ebin:
ebin@ebin-VirtualBox:~$ sudo chmod -R 775 /home/ebin/myapp1/storage
ebin@ebin-VirtualBox:~$
```

Now create an apache vhost configuration go to the following directory and create a vhost config file using nano file editor.

```
$ cd /etc/apache2/sites-available/
```

```
$ sudo nano myapp1.com.conf
```

And paste the following line inside the file.

```
<VirtualHost *:80>
    ServerName myapp1.com
    ServerAdmin admin@myapp1.com
    DocumentRoot /home/ebin/myapp1/public

    <Directory /home/ebin/myapp1>
        Options Indexes MultiViews
        AllowOverride None
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log

    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

```
ebin@ebin-VirtualBox:~$ cd /etc/apache2/sites-available/
ebin@ebin-VirtualBox:/etc/apache2/sites-available$ sudo nano myapp1.com.conf
ebin@ebin-VirtualBox:/etc/apache2/sites-available$
```

```
GNU nano 4.8                                         myapp1.com.conf
<VirtualHost *:80>
    ServerName myapp1.com

    ServerAdmin admin@myapp1.com
    DocumentRoot /home/ebin/myapp1/public

    <Directory /home/ebin/myapp1>
        Options Indexes MultiViews
        AllowOverride None
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>

[ Read 16 lines ]
^G Get Help      ^O Write Out     ^W Where Is      ^K Cut Text      ^J Justify
^X Exit          ^R Read File     ^\ Replace       ^U Paste Text    ^T To Spell
```

Now enable mod rewrite for apache2 just type

**\$ sudo a2enmod rewrite**

Now enable your site, just type

**\$ sudo a2ensite myapp1.com.conf**

Finally, Restart the apache service, type

**\$ systemctl restart apache2**

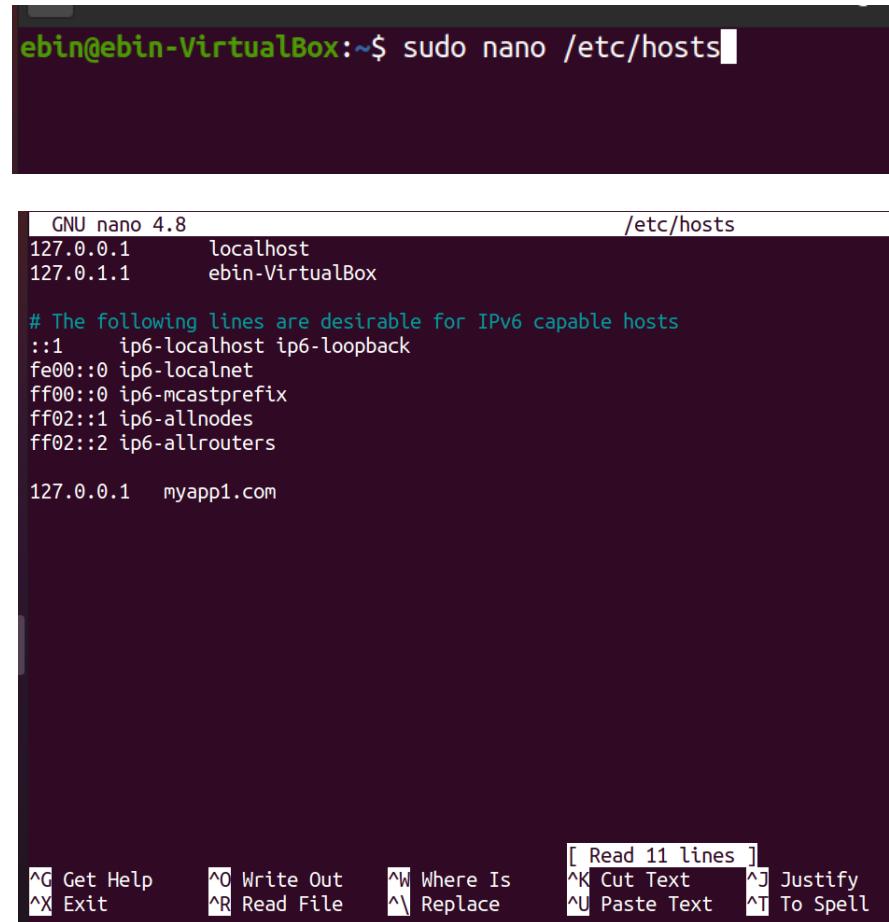
```
ebin@ebin-VirtualBox:~$ cd /etc/apache2/sites-available/
ebin@ebin-VirtualBox:/etc/apache2/sites-available$ sudo nano myapp1.com.conf
ebin@ebin-VirtualBox:/etc/apache2/sites-available$ sudo nano myapp1.com.conf
ebin@ebin-VirtualBox:/etc/apache2/sites-available$ sudo a2enmod rewrite
Enabling module rewrite.
To activate the new configuration, you need to run:
    systemctl restart apache2
ebin@ebin-VirtualBox:/etc/apache2/sites-available$ sudo a2ensite myapp1.com.conf
Enabling site myapp1.com.
To activate the new configuration, you need to run:
    systemctl reload apache2
ebin@ebin-VirtualBox:/etc/apache2/sites-available$ systemctl restart apache2
ebin@ebin-VirtualBox:/etc/apache2/sites-available$
```

As you are in a local environment you need a local dns resolver for your site. Go ahead and edit /etc/hosts file, add a dns record for your site then save the file.

**\$ sudo nano /etc/hosts**

...

**127.0.0.1 myapp1.com**



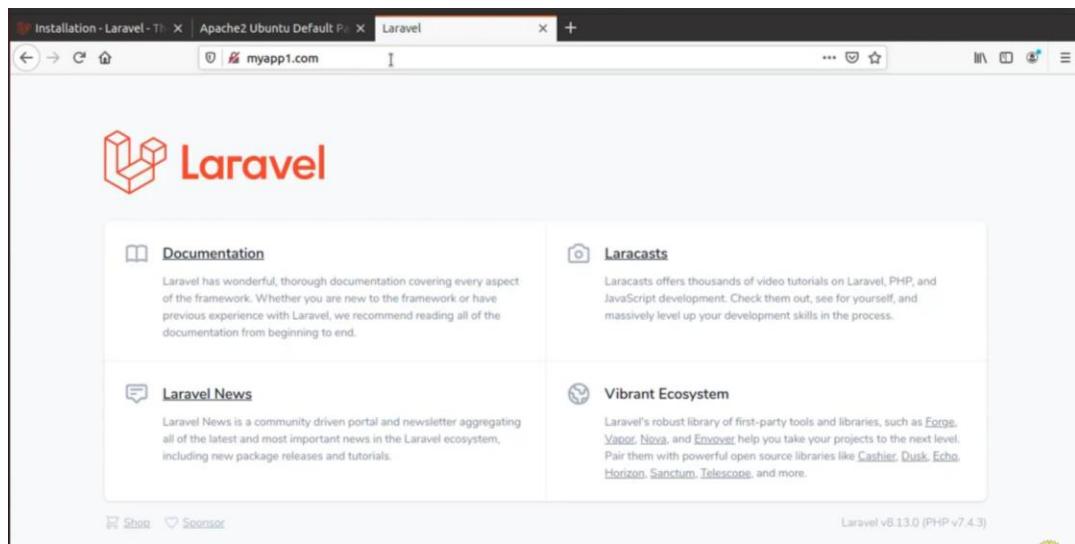
```
ebin@ebin-VirtualBox:~$ sudo nano /etc/hosts
GNU nano 4.8                               /etc/hosts
127.0.0.1      localhost
127.0.1.1      ebin-VirtualBox

# The following lines are desirable for IPv6 capable hosts
::1          ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

127.0.0.1    myapp1.com

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

Now get back to the web browser and open a tab then type your project hostname.



And here it is it's working. Here you can see the Laravel version and PHP version.

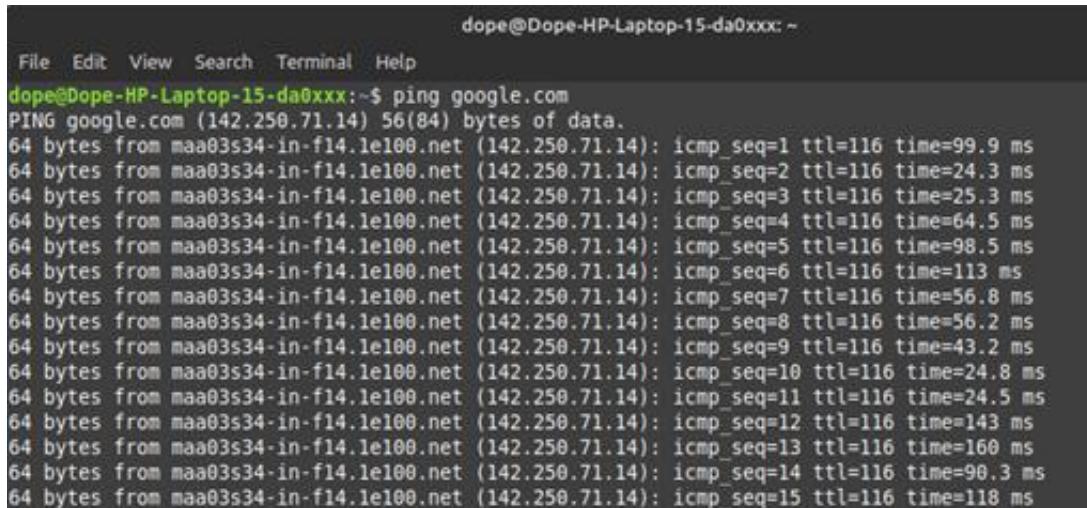
## EXPERIMENT-7

### NETWORKING COMMANDS

#### PING COMMAND

PING (Packet Internet Groper) command is used to check the network connectivity between host and server/host. This command takes as input the IP address or the URL and sends a data packet to the specified address with the message “PING” and get a response from the server/host this time is recorded which is called latency. Fast ping low latency means faster connection. Ping uses **ICMP(Internet Control Message Protocol)** to send an **ICMP echo message** to the specified host if that host is available then it sends **ICMP reply message**. Ping is generally measured in millisecond every modern operating system has this ping pre-installed.

**Syntax:**      ping [OPTIONS] DESTINATION

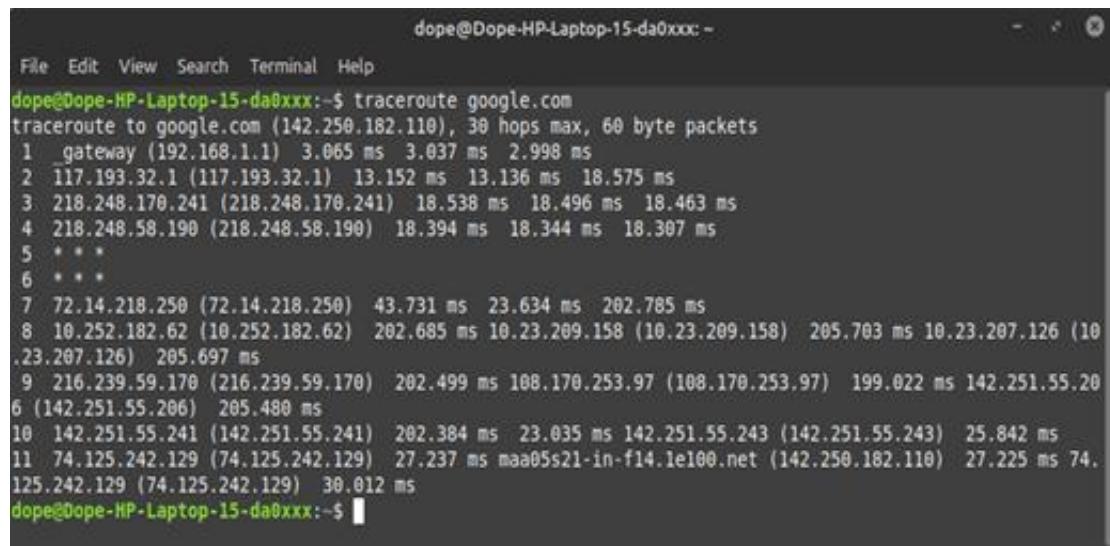


```
dope@Dope-HP-Laptop-15-da0xxx: ~
File Edit View Search Terminal Help
dope@Dope-HP-Laptop-15-da0xxx:~$ ping google.com
PING google.com (142.250.71.14) 56(84) bytes of data.
64 bytes from maa03s34-in-f14.1e100.net (142.250.71.14): icmp_seq=1 ttl=116 time=99.9 ms
64 bytes from maa03s34-in-f14.1e100.net (142.250.71.14): icmp_seq=2 ttl=116 time=24.3 ms
64 bytes from maa03s34-in-f14.1e100.net (142.250.71.14): icmp_seq=3 ttl=116 time=25.3 ms
64 bytes from maa03s34-in-f14.1e100.net (142.250.71.14): icmp_seq=4 ttl=116 time=64.5 ms
64 bytes from maa03s34-in-f14.1e100.net (142.250.71.14): icmp_seq=5 ttl=116 time=98.5 ms
64 bytes from maa03s34-in-f14.1e100.net (142.250.71.14): icmp_seq=6 ttl=116 time=113 ms
64 bytes from maa03s34-in-f14.1e100.net (142.250.71.14): icmp_seq=7 ttl=116 time=56.8 ms
64 bytes from maa03s34-in-f14.1e100.net (142.250.71.14): icmp_seq=8 ttl=116 time=56.2 ms
64 bytes from maa03s34-in-f14.1e100.net (142.250.71.14): icmp_seq=9 ttl=116 time=43.2 ms
64 bytes from maa03s34-in-f14.1e100.net (142.250.71.14): icmp_seq=10 ttl=116 time=24.8 ms
64 bytes from maa03s34-in-f14.1e100.net (142.250.71.14): icmp_seq=11 ttl=116 time=24.5 ms
64 bytes from maa03s34-in-f14.1e100.net (142.250.71.14): icmp_seq=12 ttl=116 time=143 ms
64 bytes from maa03s34-in-f14.1e100.net (142.250.71.14): icmp_seq=13 ttl=116 time=160 ms
64 bytes from maa03s34-in-f14.1e100.net (142.250.71.14): icmp_seq=14 ttl=116 time=90.3 ms
64 bytes from maa03s34-in-f14.1e100.net (142.250.71.14): icmp_seq=15 ttl=116 time=118 ms
```

#### TRACEROUTE COMMAND

**Traceroute** command in Linux prints the route that a packet takes to reach the host. This command is useful when you want to know about the route and about all the hops that a packet takes. Below image depicts how traceroute command is used to reach the Google (172.217.26.206) host from the local machine and it also prints detail about all the hops that it visits in between.

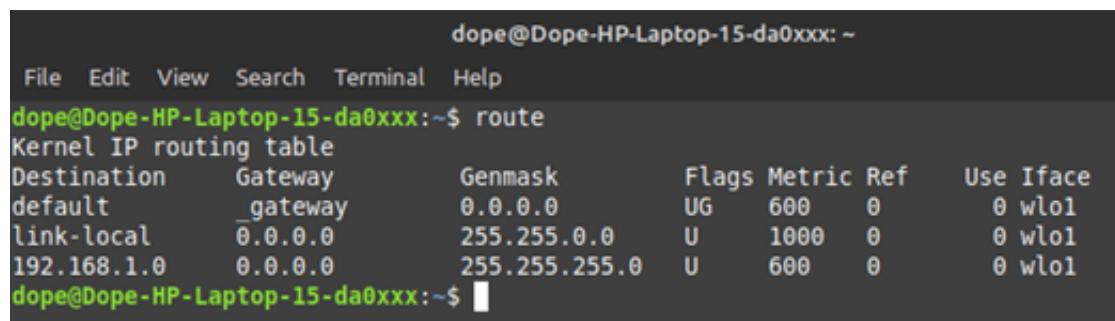
**Syntax:** traceroute [options] host\_Address [pathlength]



```
dope@Dope-HP-Laptop-15-da0xxx:~$ traceroute google.com
traceroute to google.com (142.250.182.110), 30 hops max, 60 byte packets
 1  gateway (192.168.1.1)  3.065 ms  3.037 ms  2.998 ms
 2  117.193.32.1 (117.193.32.1)  13.152 ms  13.136 ms  18.575 ms
 3  218.248.170.241 (218.248.170.241)  18.538 ms  18.496 ms  18.463 ms
 4  218.248.58.190 (218.248.58.190)  18.394 ms  18.344 ms  18.307 ms
 5  * * *
 6  * * *
 7  72.14.218.250 (72.14.218.250)  43.731 ms  23.634 ms  282.785 ms
 8  10.252.182.62 (10.252.182.62)  202.685 ms  10.23.209.158 (10.23.209.158)  205.703 ms  10.23.207.126 (10.23.207.126)  205.697 ms
 9  216.239.59.170 (216.239.59.170)  202.499 ms  108.170.253.97 (108.170.253.97)  199.022 ms  142.251.55.206 (142.251.55.206)  205.480 ms
10  142.251.55.241 (142.251.55.241)  202.384 ms  23.035 ms  142.251.55.243 (142.251.55.243)  25.842 ms
11  74.125.242.129 (74.125.242.129)  27.237 ms  maa05s21-in-f14.1e100.net (142.250.182.110)  27.225 ms  74.125.242.129 (74.125.242.129)  30.812 ms
dope@Dope-HP-Laptop-15-da0xxx:~$
```

## **ROUTE COMMAND**

**route** command in Linux is used when you want to work with the IP/kernel routing table. It is mainly used to set up static routes to specific hosts or networks via an interface. It is used for showing or update the IP/kernel routing table.



```
dope@Dope-HP-Laptop-15-da0xxx:~$ route
File Edit View Search Terminal Help
dope@Dope-HP-Laptop-15-da0xxx:~$ route
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref    Use Iface
default         _gateway       0.0.0.0        UG    600    0        0 wlo1
link-local      0.0.0.0        255.255.0.0   U     1000   0        0 wlo1
192.168.1.0    0.0.0.0        255.255.255.0 U     600    0        0 wlo1
dope@Dope-HP-Laptop-15-da0xxx:~$
```

**Syntax:** route

## **NSLOOKUP COMMAND**

**nslookup** (stands for “Name Server Lookup”) is a useful command for getting information from DNS server. It is a network administration tool for querying the Domain Name System (DNS) to obtain domain name or IP address mapping or any other specific DNS record. It is also used to troubleshoot DNS related problems.

Syntax: nslookup [option]

```
dope@Dope-HP-Laptop-15-da0xxx: ~
File Edit View Search Terminal Help
dope@Dope-HP-Laptop-15-da0xxx:~$ nslookup google.com
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.195.110
Name:   google.com
Address: 2404:6800:4007:818::200e

dope@Dope-HP-Laptop-15-da0xxx:~$ █
```

## **IFCONFIG COMMAND**

**ifconfig**(interface configuration) command is used to configure the kernel-resident netwo

```
dope@Dope-HP-Laptop-15-da0xxx: ~
File Edit View Search Terminal Help
dope@Dope-HP-Laptop-15-da0xxx:~$ ifconfig
en0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
      ether f4:39:09:73:b2:32 txqueuelen 1000 (Ethernet)
      RX packets 0 bytes 0 (0.0 B)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 0 bytes 0 (0.0 B)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
      inet6 ::1 prefixlen 128 scopeid 0x10<host>
          loop txqueuelen 1000 (Local Loopback)
          RX packets 215 bytes 21078 (21.0 KB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 215 bytes 21078 (21.0 KB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlo1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.1.107 netmask 255.255.255.0 broadcast 192.168.1.255
      inet6 fe80::540:283a:90c7:58f6 prefixlen 64 scopeid 0x20<link>
          ether dc:a2:66:54:43:1b txqueuelen 1000 (Ethernet)
          RX packets 25136 bytes 30759181 (30.7 MB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 8145 bytes 1687314 (1.6 MB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

dope@Dope-HP-Laptop-15-da0xxx:~$ █
```

interfaces. It is used at the boot time to set up the interfaces as necessary. After that, it is usually used when needed during debugging or when you need system tuning. Also, this command is used to assign the IP address and netmask to an interface or to enable or disable a given interface  
Syntax: ifconfig [...OPTIONS] [INTERFACE]

## **EXERCISE 8**

### **WIRESHARK**

#### **Wireshark**

**Wireshark** is a network packet analyzer. A network packet analyzer presents captured packet data in as much detail as possible. You could think of a network packet analyzer as a measuring device for examining what's happening inside a network cable, just like an electrician uses a voltmeter for examining what's happening inside an electric cable (but at a higher level, of course).

In the past, such tools were either very expensive, proprietary, or both. However, with the advent of Wireshark, that has changed. Wireshark is available for free, is open source, and is one of the best packet analyzers available today. It captures network traffic on the local network and stores that data for offline analysis.

Wireshark captures network traffic from Ethernet, Bluetooth, Wireless (IEEE 802.11), Token Ring, Frame Relay connections, and more.

Here are some reasons people use Wireshark:

- | Network administrators use it to *troubleshoot network problems*
- | Network security engineers use it to *examine security problems*
- | QA engineers use it to *verify network applications*
- | Developers use it to *debug protocol implementations*
- | People use it to *learn network protocol internals*

Wireshark can also be helpful in many other situations.

#### **FEATURES**

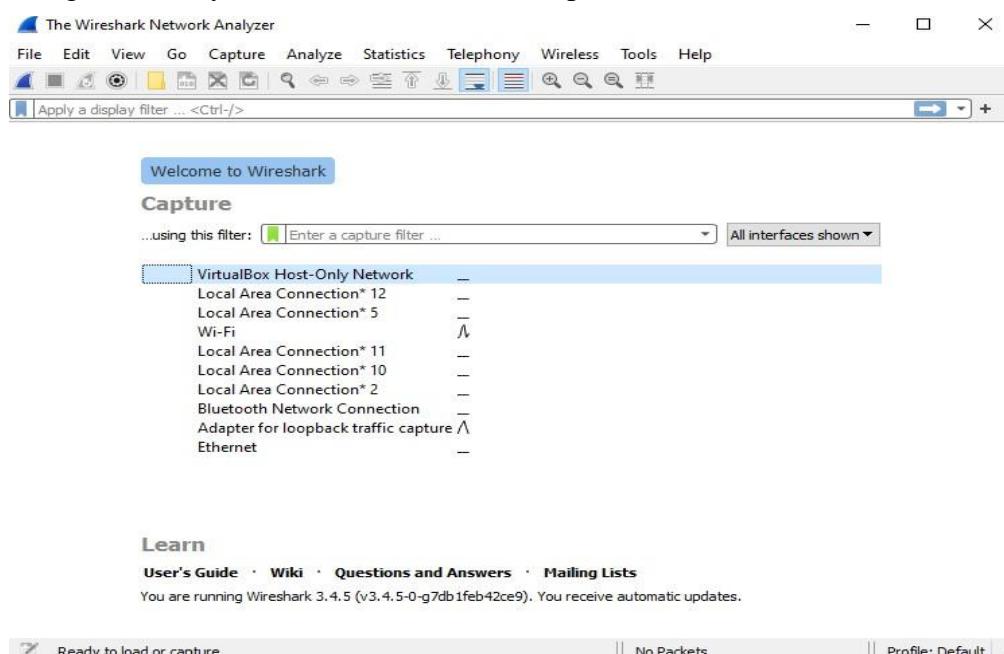
- | Data can be captured "from the wire" from a live network connection or read from a file of already-captured packets.
- | Live data can be read from different types of networks, including Ethernet, IEEE802.11, PPP, and loopback.

- | Captured network data can be browsed via a GUI, or via the terminal (command line) version of the utility, TShark.
- | Captured files can be programmatically edited or converted via command-line switches to the "editcap" program.
- | Data display can be refined using a display filter.
- | Plug-ins can be created for dissecting new protocols.
- | VoIP calls in the captured traffic can be detected. If encoded in a compatible encoding, the media flow can even be played.
- | Raw USB traffic can be captured
- | Wireless connections can also be filtered as long as they traverse the monitored Ethernet
- | Various settings, timers, and filters can be set to provide the facility of filtering the output of the captured traffic.

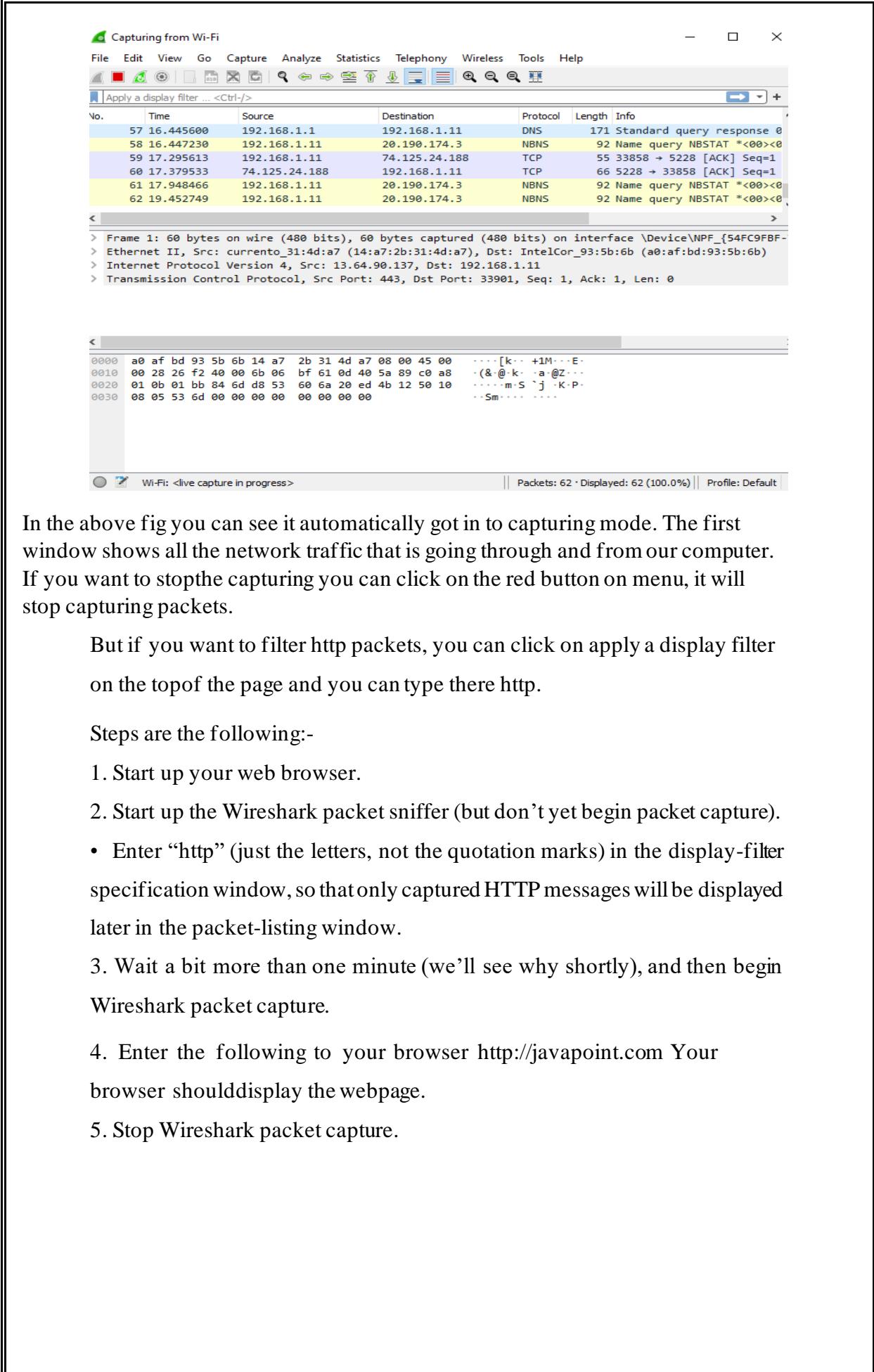
### **How to use wireshark to capture the packet for analysis?**

#### Analyzing data packet on wireshark

Wireshark shows you three different panes for inspecting packet data. The packet list, the top pane, is the list of all packets that are captured. When you click on a packet, the other two panes change to show you about the details of the packet



The above fig shows the first or opening page of wireshark(here you can select any of the option for eg:-wifi

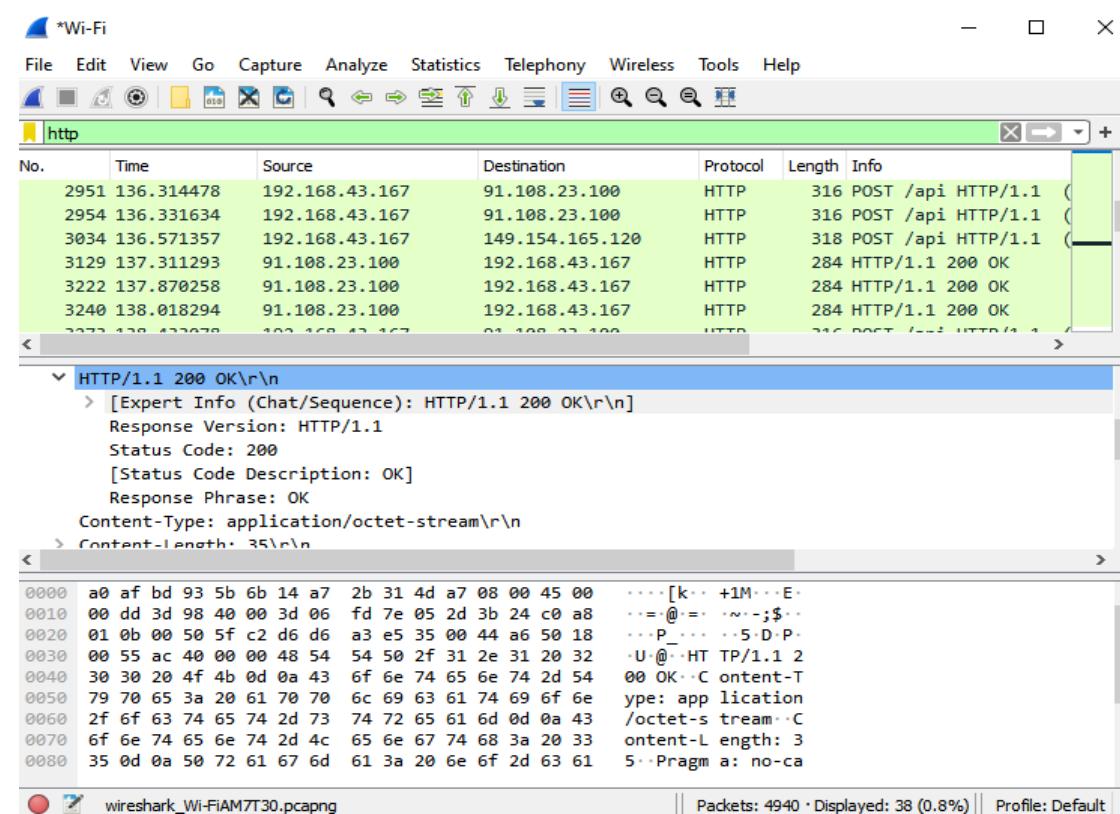


In the above fig you can see it automatically got in to capturing mode. The first window shows all the network traffic that is going through and from our computer. If you want to stop the capturing you can click on the red button on menu, it will stop capturing packets.

But if you want to filter http packets, you can click on apply a display filter on the top of the page and you can type there http.

Steps are the following:-

1. Start up your web browser.
2. Start up the Wireshark packet sniffer (but don't yet begin packet capture).
  - Enter "http" (just the letters, not the quotation marks) in the display-filter specification window, so that only captured HTTP messages will be displayed later in the packet-listing window.
3. Wait a bit more than one minute (we'll see why shortly), and then begin Wireshark packet capture.
4. Enter the following to your browser <http://javapoint.com> Your browser should display the webpage.
5. Stop Wireshark packet capture.



The above fig shows capturing http packets. After browsing any web page on your browser, come back to wireshark and start capturing. You can select any of the packets from the list. Now the selected packet details will be specified in the second window. The content is displayed in the third window.

There are columns specifying time, source IP address, destination IP address, the protocol, length of the packet and its information.

## **EXPERIMENT-9**

### **INTRODUCTION TO VIRTUAL MACHINES**

#### **Creating a Virtual Machine**

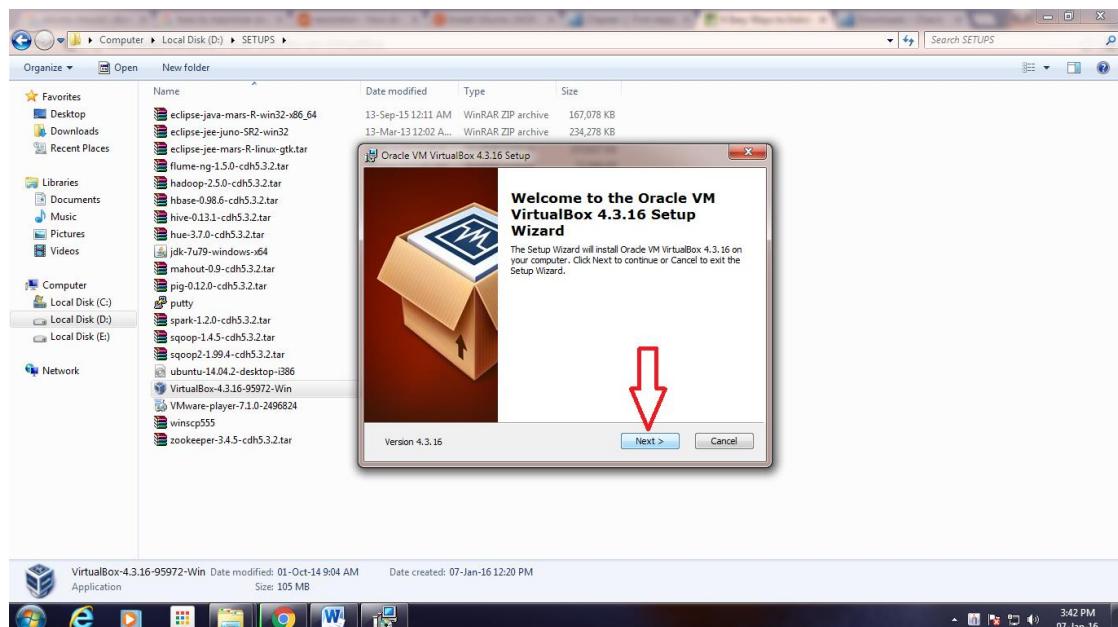
##### **1. Install VirtualBox .**

If you don't already have VirtualBox installed on your Windows or Mac computer, you'll need to install it before proceeding.

Following are the steps required to install VirtualBox(Oracle VM VirtualBox):

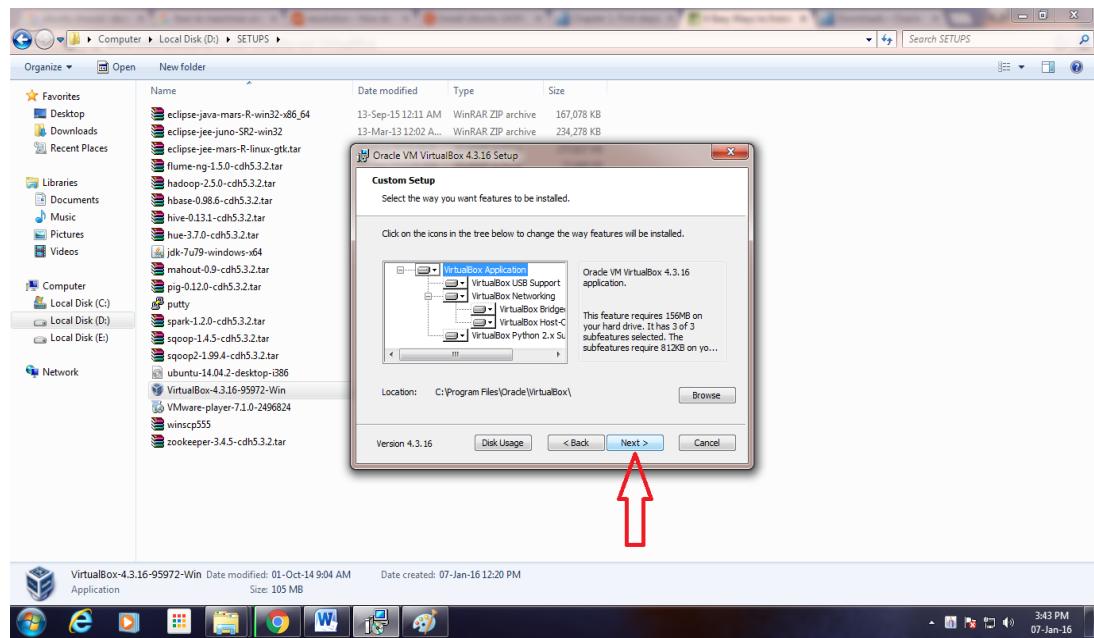
You can download the latest version of VirtualBox from the Virtual Box website: <https://www.virtualbox.org/wiki/Downloads> according to the version of your operating system Windows, Mac or Linux.

##### **1.1. Click Next**



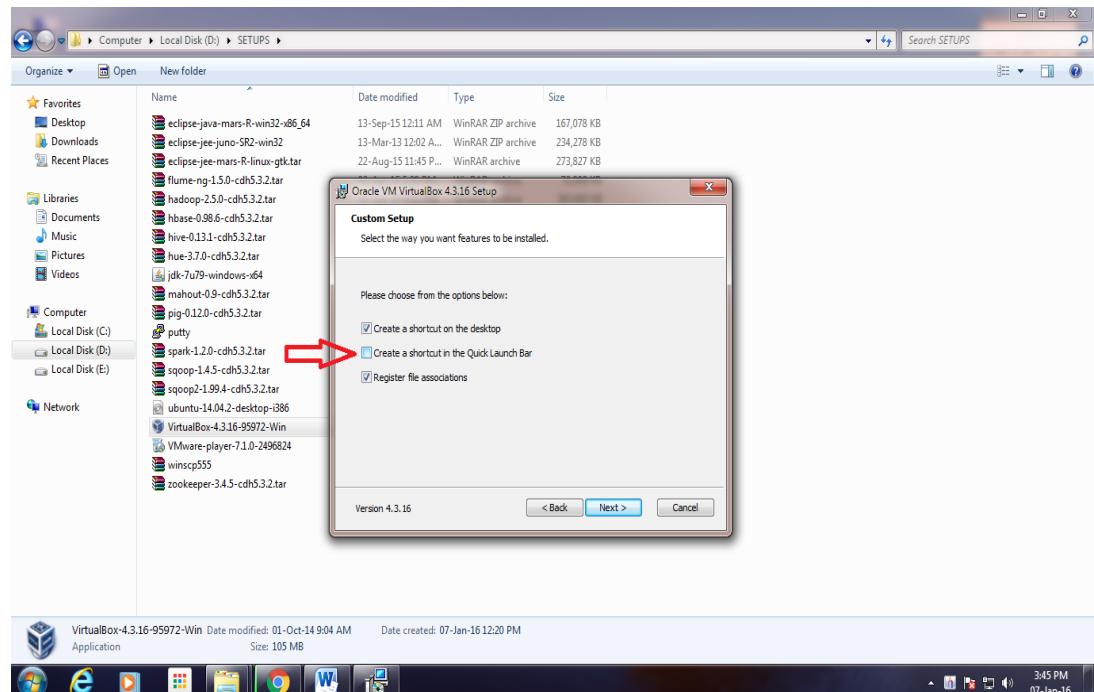
To Install VirtualBox – Setup Wizard

## 1.2. Click Next



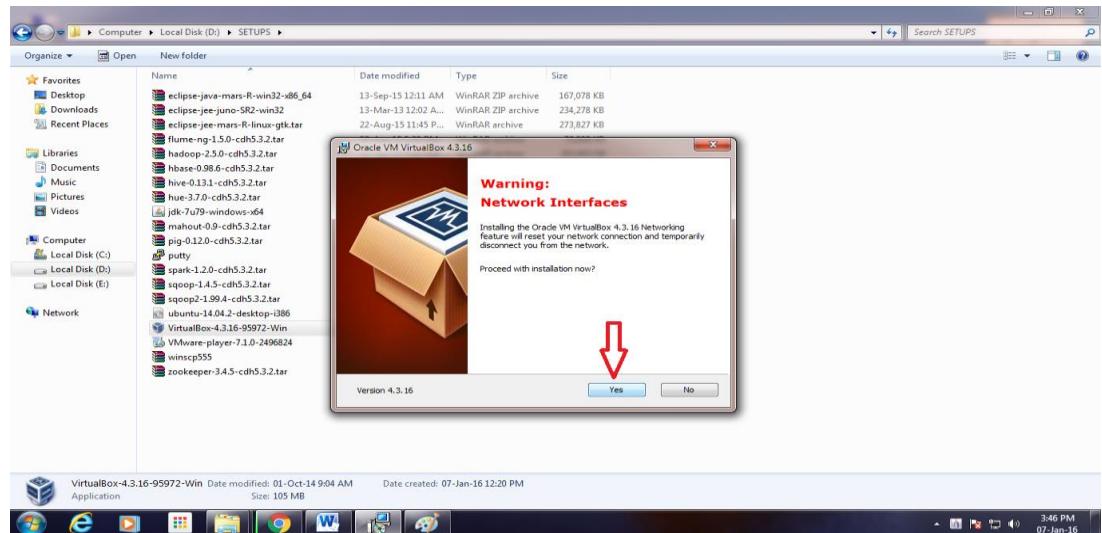
To Install VirtualBox – Custom Setup

## 1.3. Uncheck “Create a shortcut in the Quick Launch Bar” and click “Next”



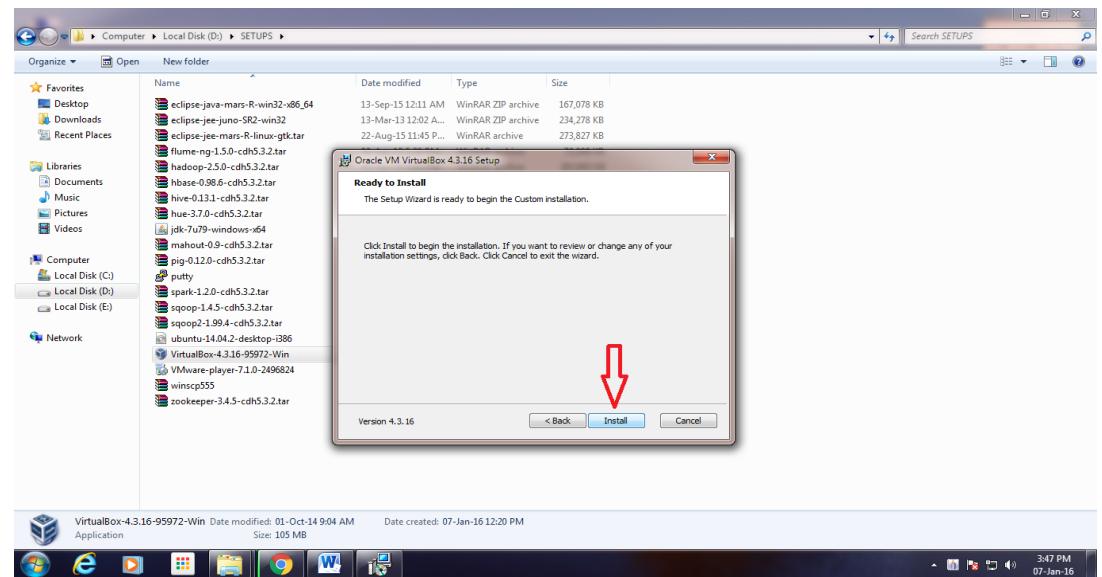
To Install VirtualBox – Features Selection

### 1.4. Click “Yes”



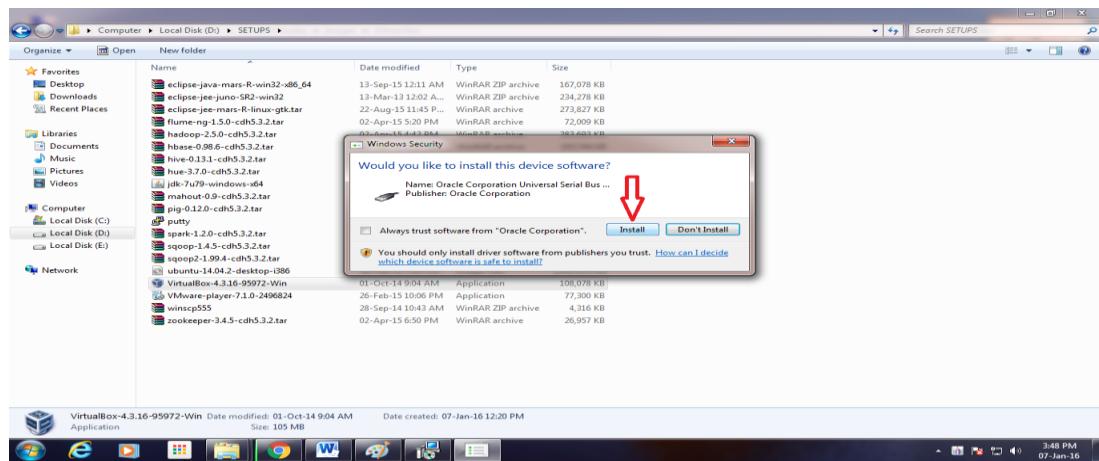
To Install VirtualBox – Network Interfaces Warning

### 1.5. Click “Install”



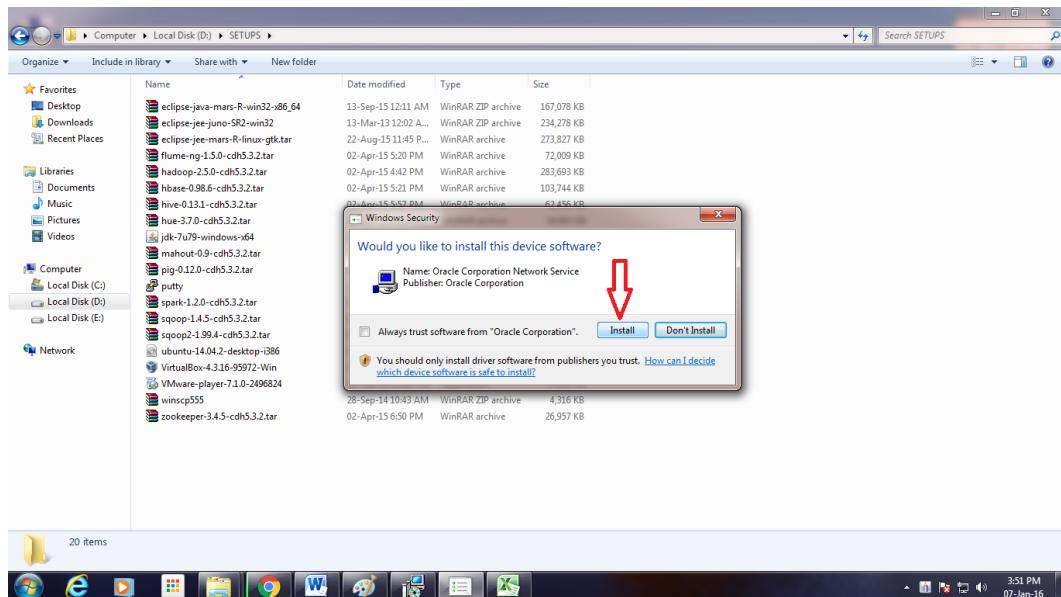
Installation of Oracle VM VirtualBox – Ready to Install

### 1.6. Click “Install”



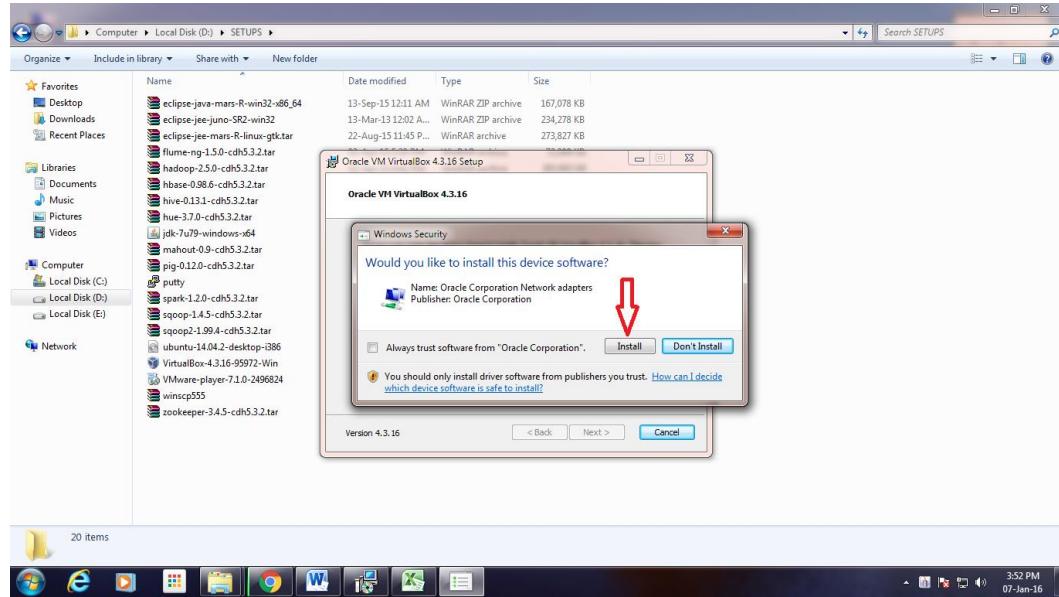
### Installation of Oracle VM VirtualBox - Serial Bus Software Installation

### 1.7. Click “Install”



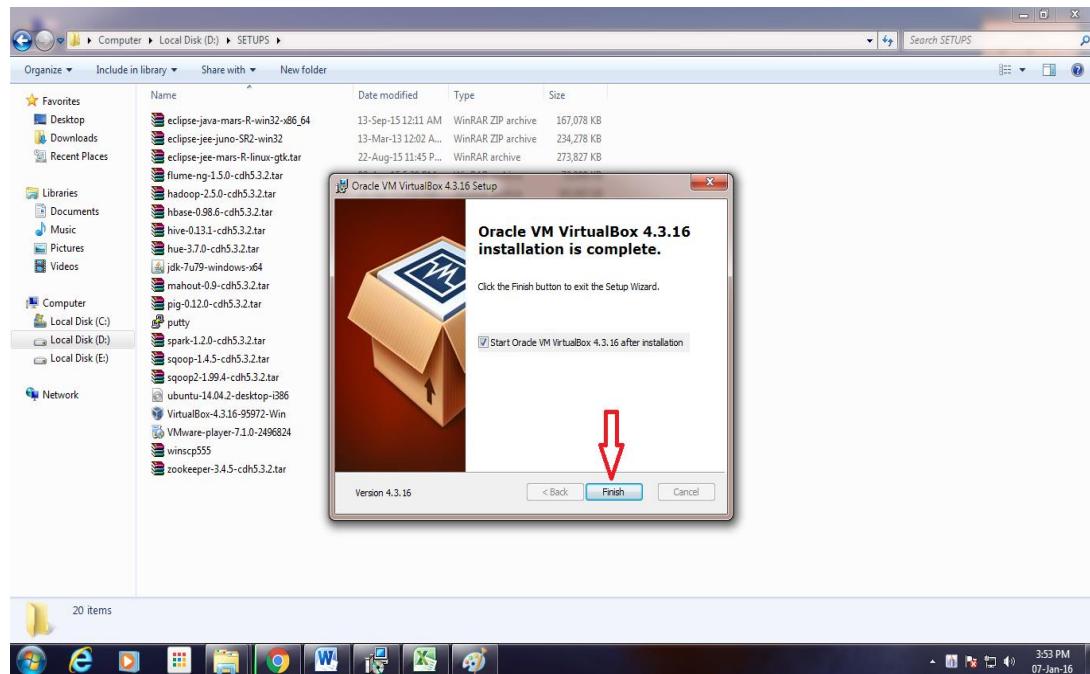
### Installation of Oracle VM VirtualBox – Network Service Installation

**1.8. Click “Install”**



Installation of Oracle VM VirtualBox – Network Adapters Installation

**1.9. Click “Finish”**



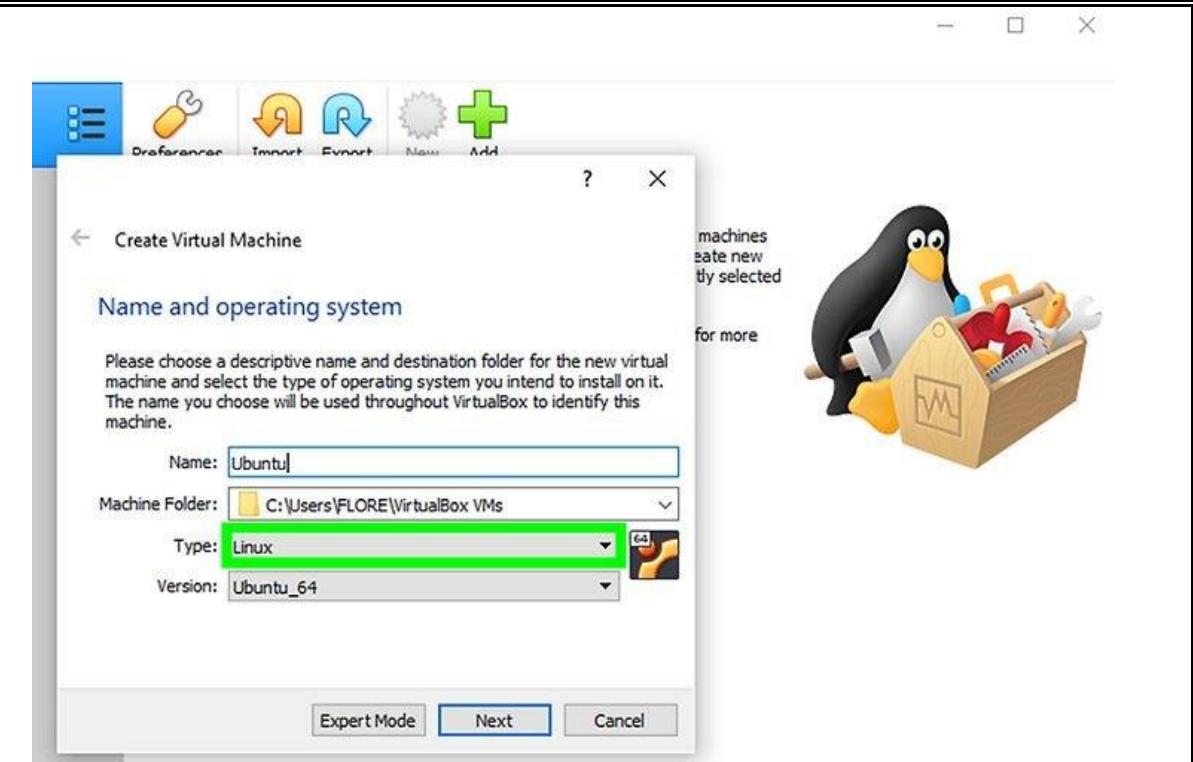
**2. Open VirtualBox. Double-click (or click once on a Mac) the VirtualBox app icon.**



**3 Click New.** It's a blue badge in the upper-left corner of the VirtualBox window. Doing so opens a pop-up menu.



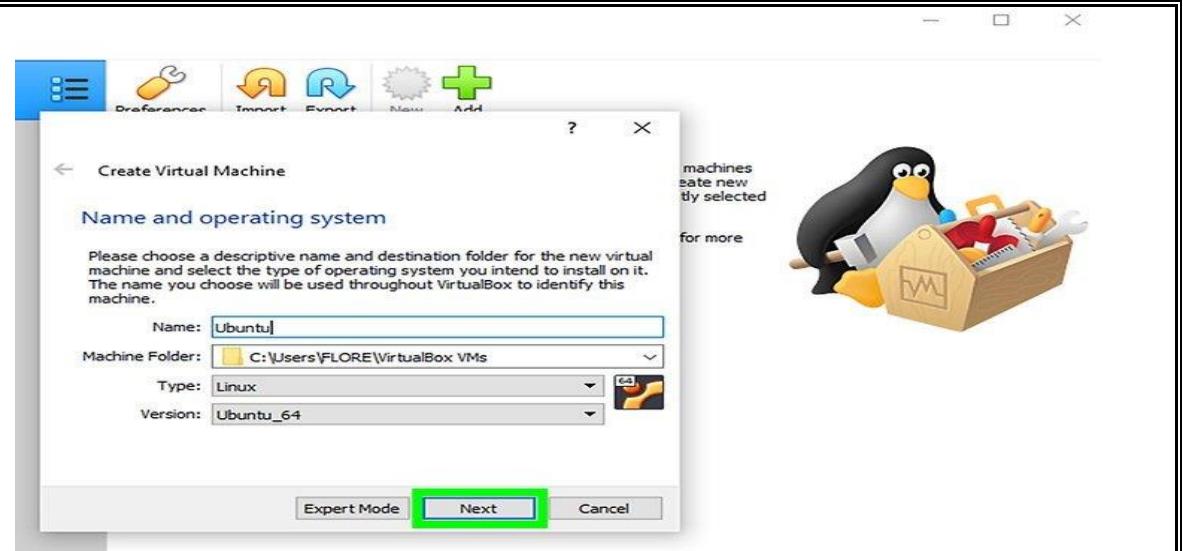
**4 Enter a name for your virtual machine.** Type whatever you want to name your virtual machine (e.g., Ubuntu) into the "Name" text field that's near the top of the pop-up menu.



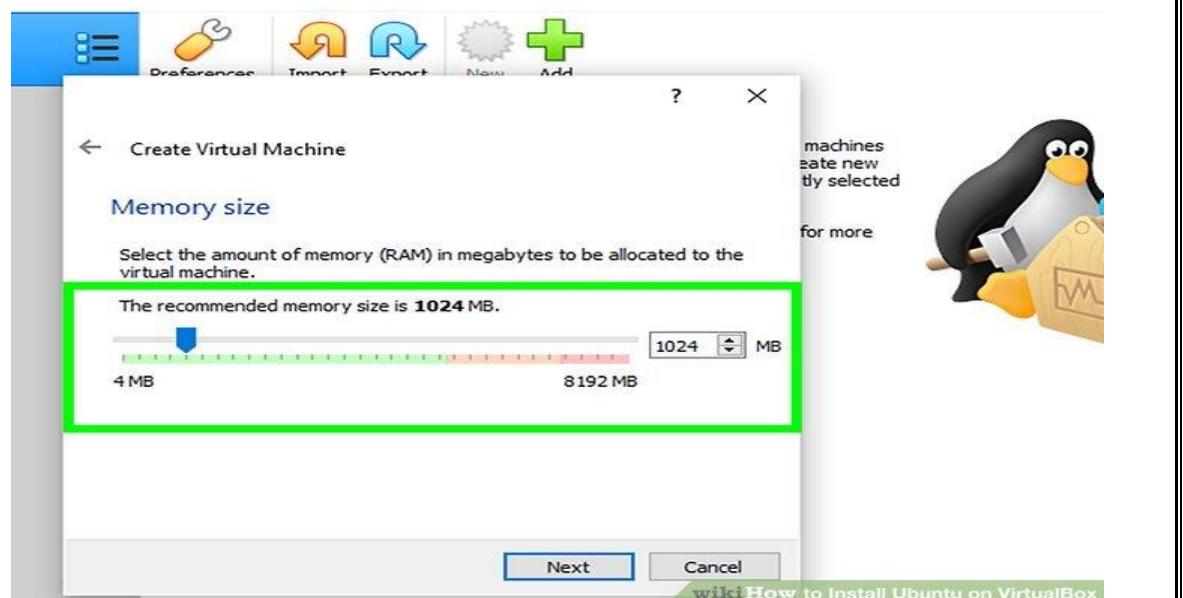
5. For **Operating System Type**, select the OS that you want to install.



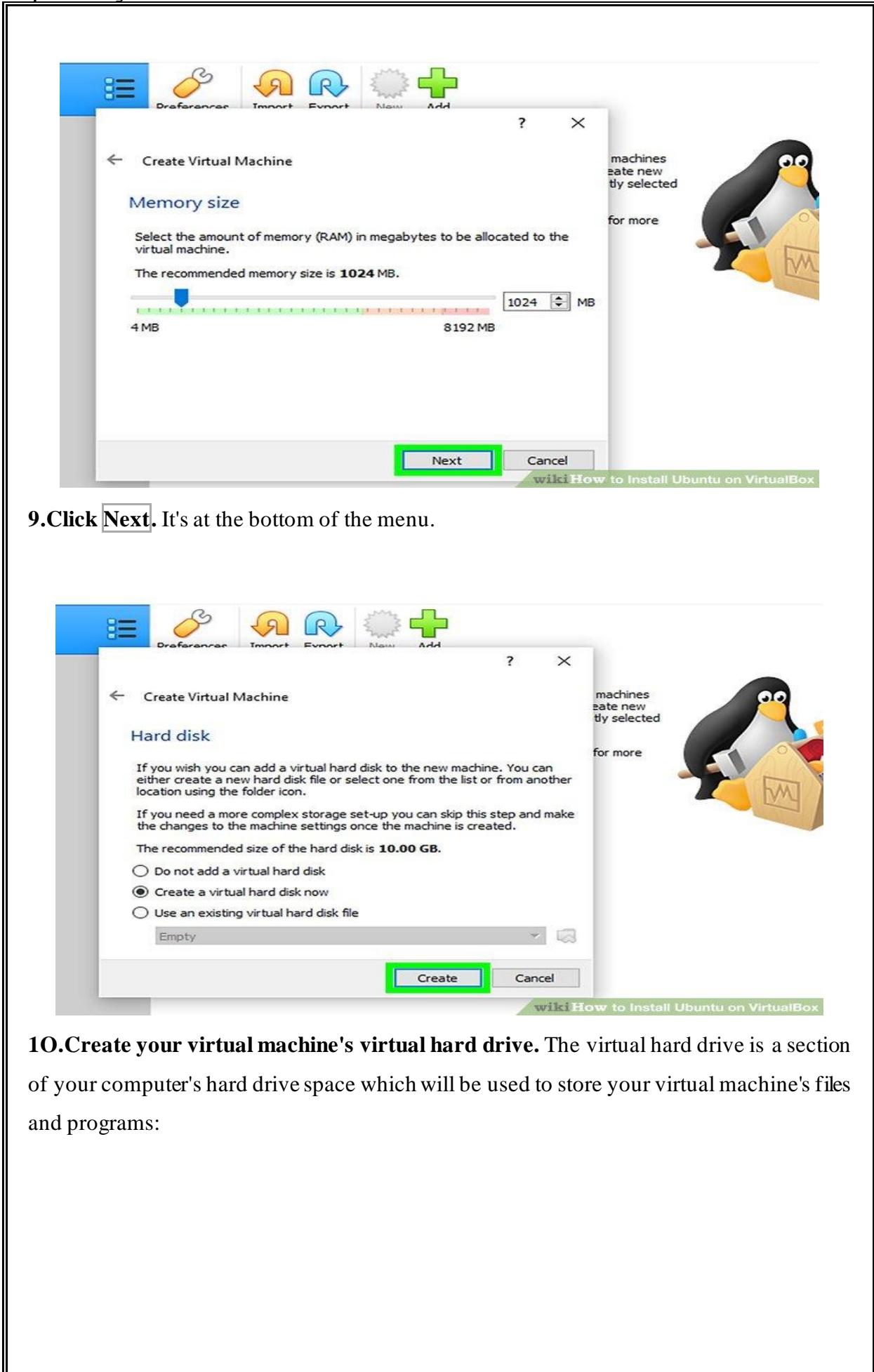
6. Select the **version** of the operating system.



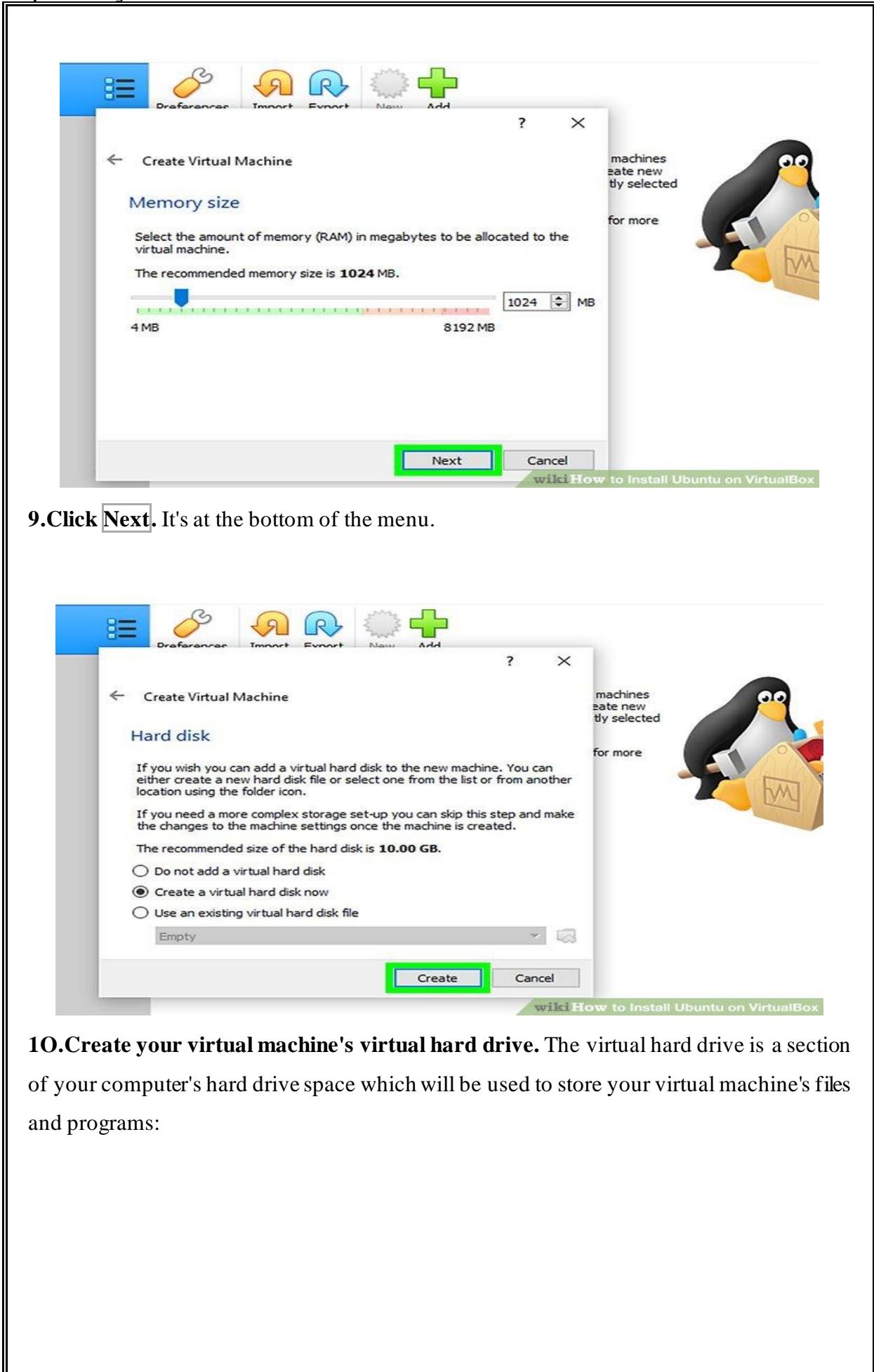
7. Click **Next**. It's at the bottom of the menu.



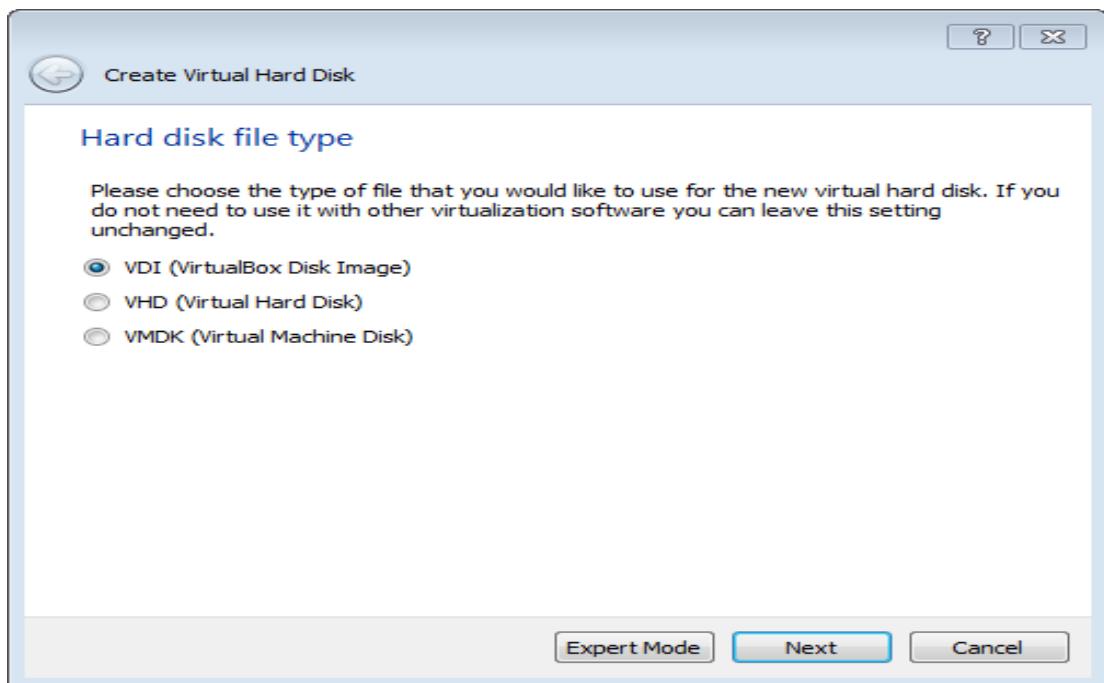
8. Select an amount of RAM to use. Click and drag the slider left or right to decrease or increase the amount of RAM that VirtualBox will have available for your virtual machine. The ideal amount of RAM will automatically be selected when you get to this page. Make sure not to increase the RAM into the red section of the slider; try to keep the slider in the green.



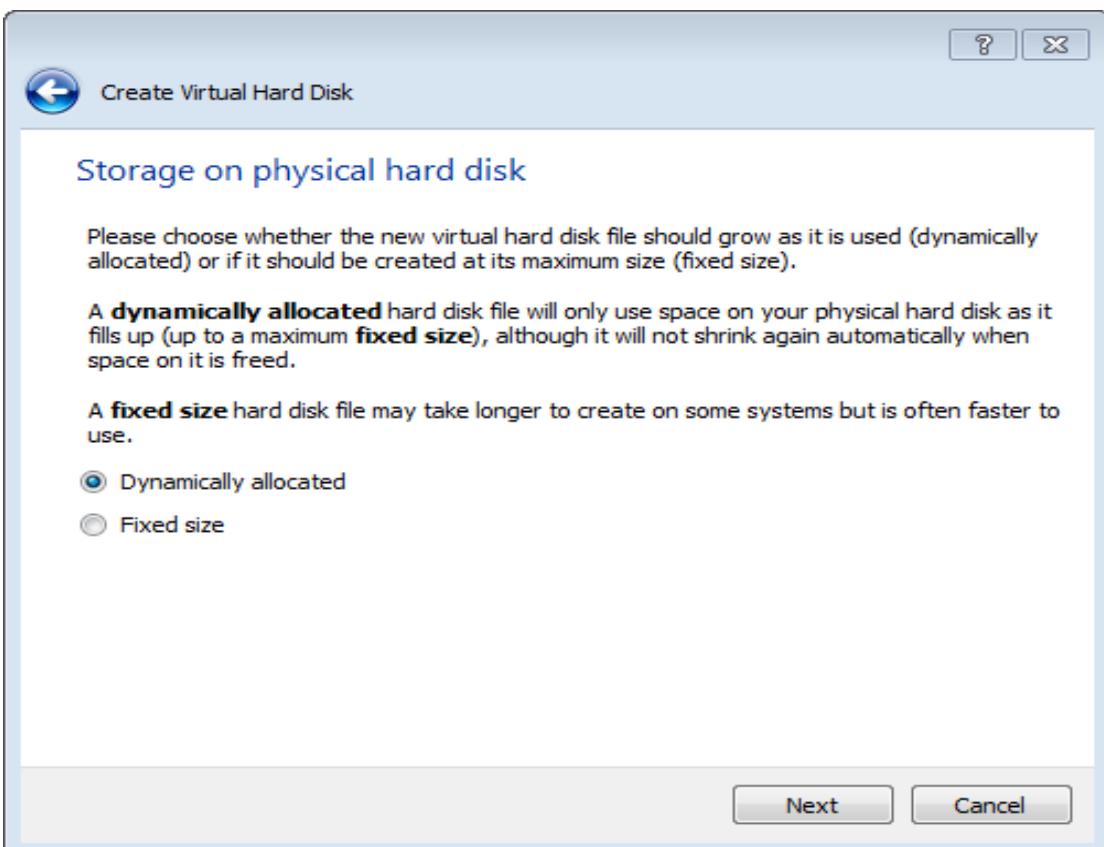
**9.Click Next.** It's at the bottom of the menu.



**10.Create your virtual machine's virtual hard drive.** The virtual hard drive is a section of your computer's hard drive space which will be used to store your virtual machine's files and programs:

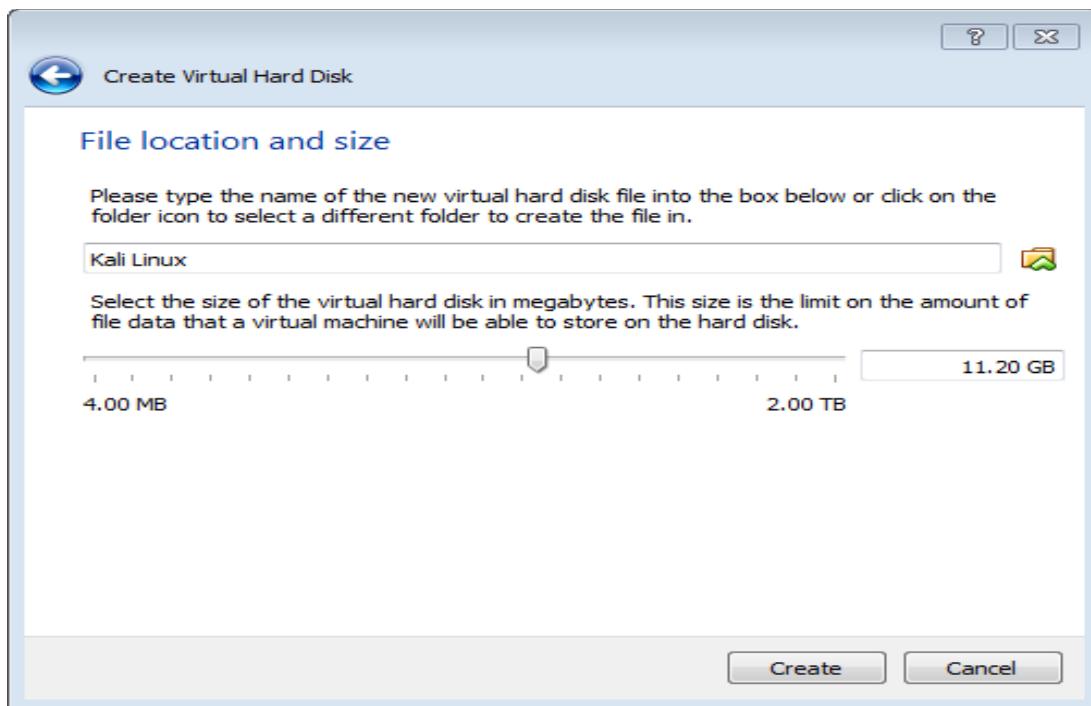


11. Use “VDI” to create a virtual hard disk



12 .Choose “Dynamically allocated”

13. Allocate at Minimum 8 GB (recommended 10 or more).



Click **Create**, to create your new virtual machine. The virtual machine is displayed in the list on the left side of the VirtualBox Manager window, with the name that you entered initially.

VMs can run multiple operating system environments on a single physical computer, saving physical space, time and management costs.