

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

HORMIS NAGAR, MOOKKANNOOR

**ANGAMALY-683577**



**'FOCUS ON EXCELLENCE'**

**ADVANCED COMPUTER NETWORK**

.....  
**LABORATORY RECORD**

**Name: AKSHAYA J**

**Branch: MASTER OF COMPUTER APPLICATION**

**Semester: 2   Batch: SEMESTER -2 A      Roll No: 7**

**FEDERAL INSTITUTE OF  
SCIENCE AND TECHNOLOGY**

**(FISAT)<sup>TM</sup>**

**HORMIS NAGAR, MOOKKANNOOR**

**ANGAMALY-683577**



**'FOCUS ON EXCELLENCE'**

**Name : AKSHAYA J**

**Branch : MASTER OF COMPUTER APPLICATION**

**Semester : 2 Roll No: 7**

**University Exam.RegNo: FITMCA20-2007**

## **CERTIFICATE**

This is to certify that this is a Bonafide record of the Practical work done and submitted to *Kerala Technological University* in partial fulfillment for the award of the Master Of Computer Applications is a record of the original research work done by **AKSHAYA J** in the **ADVANCED COMPUTER NETWORKS** Laboratory of the Federal Institute of Science and Technology during the academic year 2020-2021.

Signature of Staff in Charge

Name:

Date:

Signature of H.O.D

Name:

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

Signature of

Internal Examiner

Signature of

External Examiner

## CONTENT

SI No:	Date :	Name of Experiment:	Page No:	Signature of Staff -In - Charge:
1	27-11-20	COMPONENT IDENTIFICATION	5	
2	27-11-20	LINUX COMMANDS	26	
3	4-12-20	FILE SYSTEM	32	
4	17-12-20	SHELL SCRIPT	39	
5	31-12-20	LAMP STOCK SERVER	48	
6	14-01-21	LARAVEL SERVER	51	
7	15-01-21	NETWORKING COMMANDS	63	
8	21-01-21	WIRE SHARK	66	
9	21-01-21	VIRTUAL BOX	70	

## EXPERIMENT -1

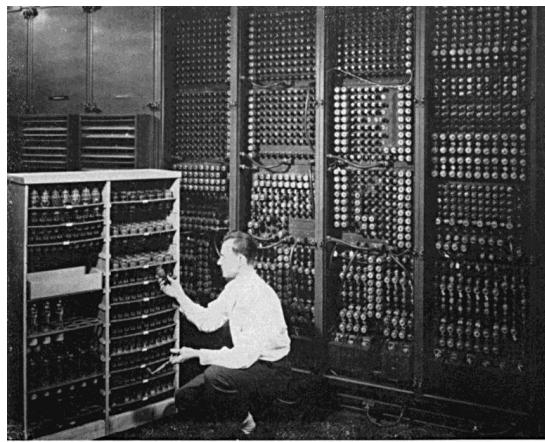
### COMPONENT IDENTIFICATION

#### 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**

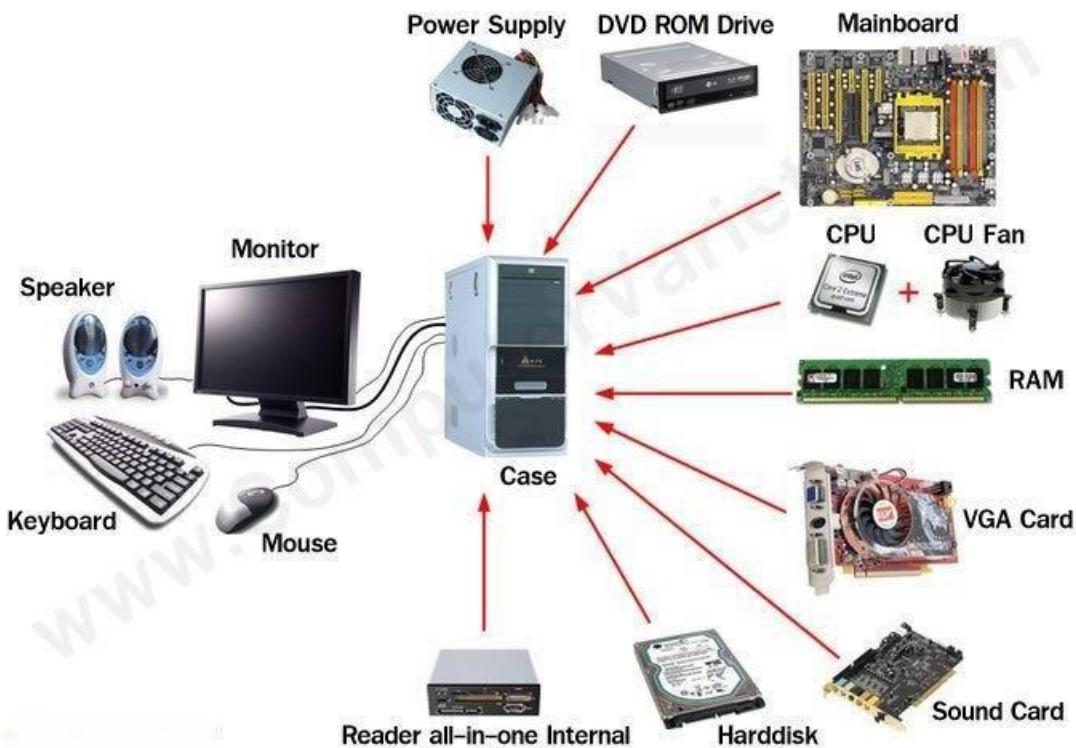


Replacing a bad tube meant checking among ENIAC's 19,000 possibilities.

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

## **2. 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.

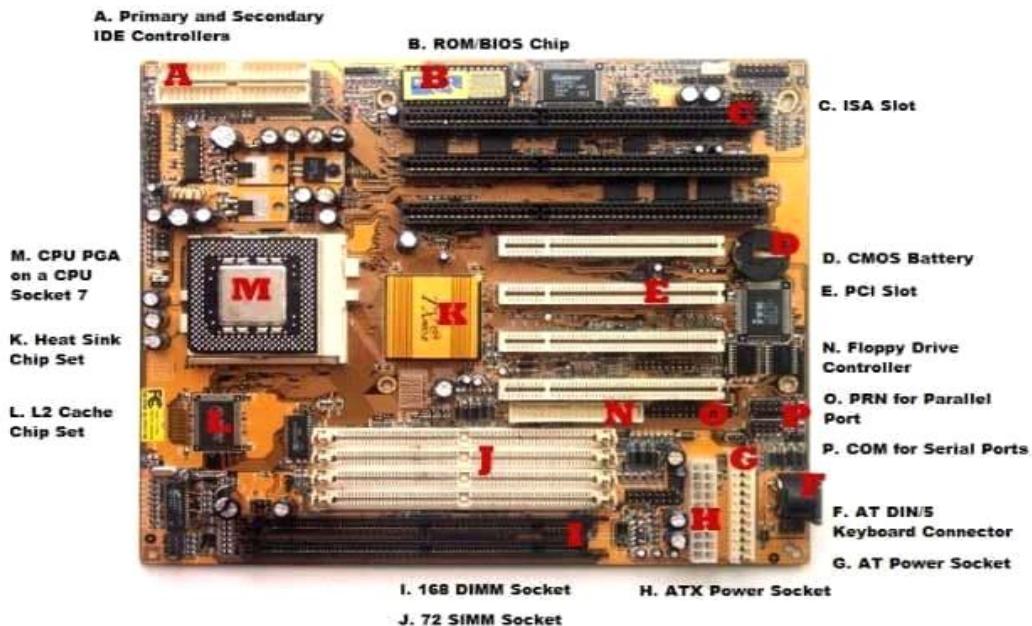
### **FEATURES OF MOTHERBOARD**

- 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**



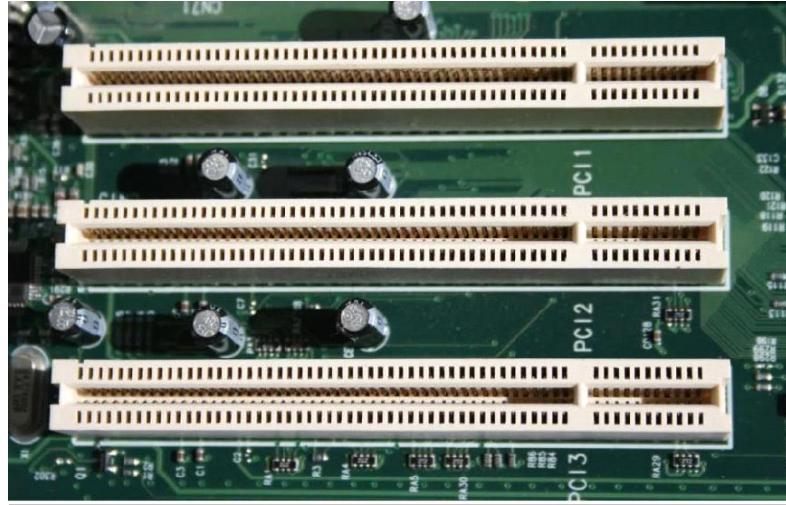
Serial Ports provide an interface to connect serial lines to prepare a serial communication. Serial ports are typically used in modem, mouse, security cameras etc. A Serial port uses DB-9 connector, a 9 pin D-Shaped Connector which connects to the transmission line. A serial port provides a serial communication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

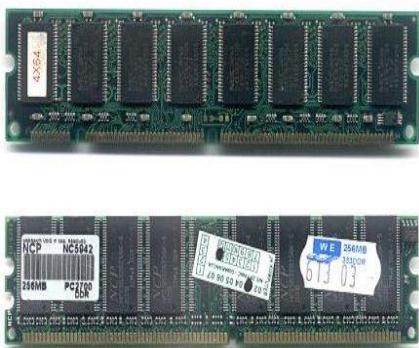
CAS (Column Address Strobe) Latency, sometimes abbreviated to just "CL," indicates the time it takes for a memory module to return data to the CPU. A lower CAS Latency indicates RAM that performs faster.

### **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.

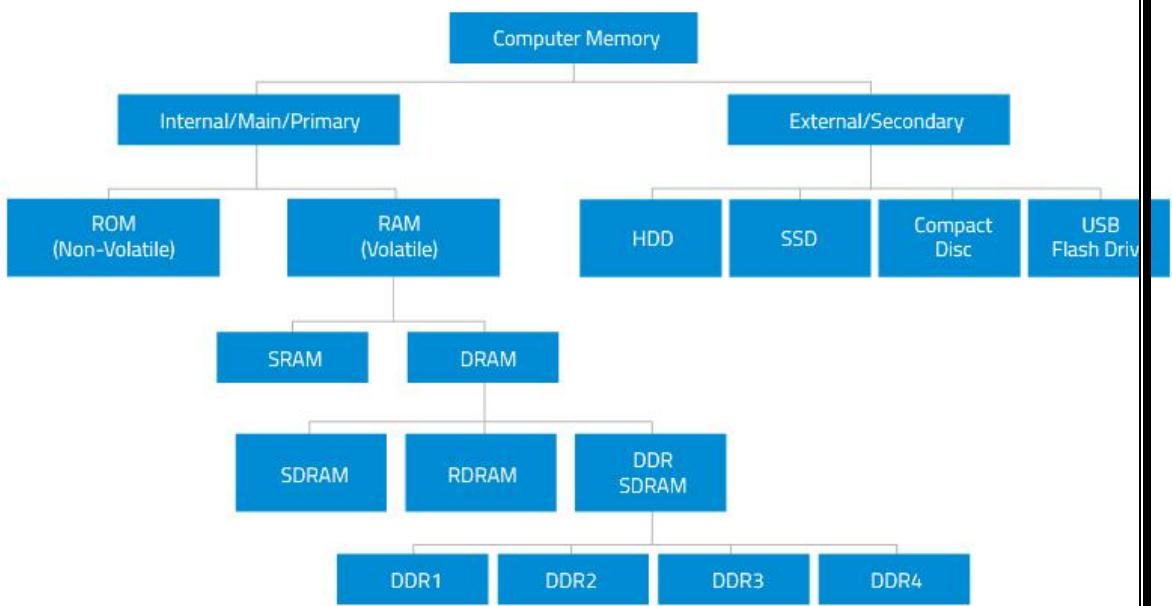


### **Different Types of RAM**

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

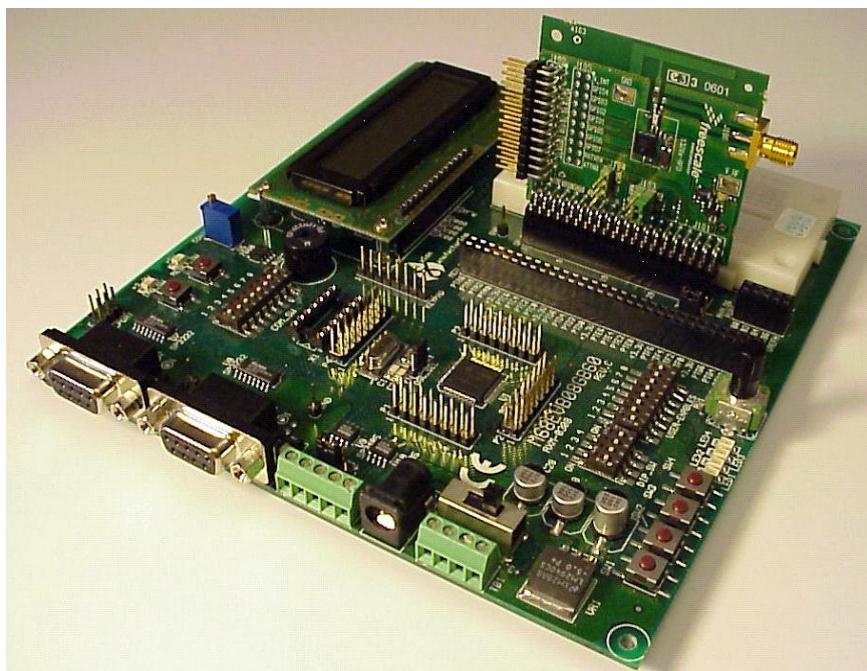
this data in RAM, programs can work with it quickly, improving speed and responsiveness.

### 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.

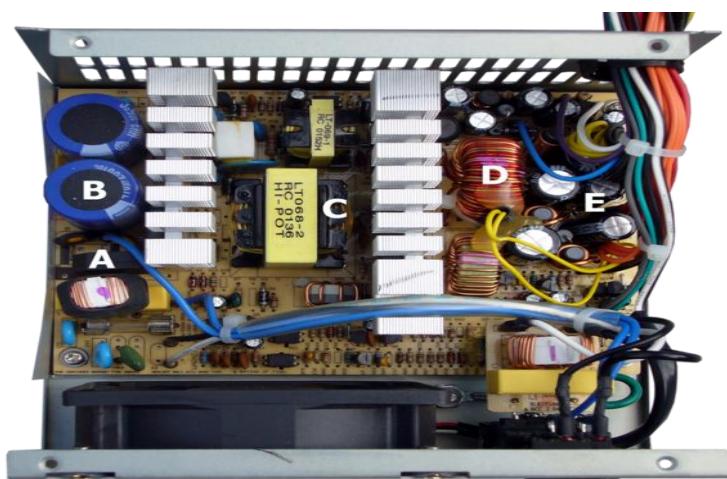
- A daughter board is a printed circuit board which is connected to the motherboard or expansion card.
- Unlike expansion card, daughter boards are directly connected to the motherboard by soldering.
- Daughter boards do not provide new functions to the circuit like an expansion but they extend the circuitry 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**

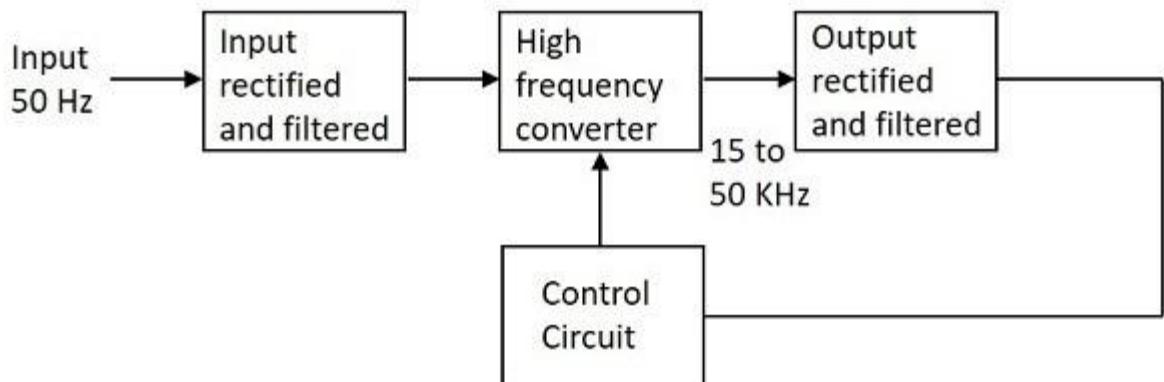
- Video Card: This is also referred to as the graphics adapter, display adapter or video adapter.
- Sound Card: To handle sound, to insert a microphone or connect a speaker this sound card is used.
- Network Interface Card: This is also referred as NIC. The computer can be connected to a network only 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.

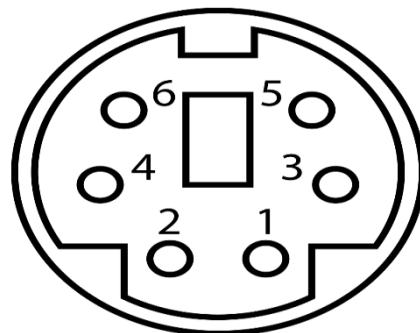
### **Types of Computer Ports**

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,

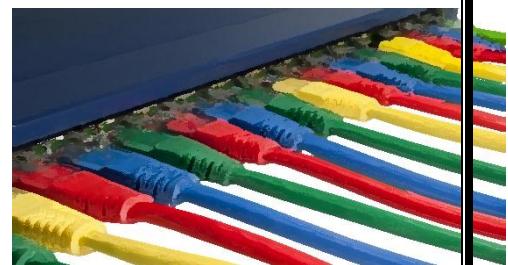


**7.3 PARALLEL PORT:** Parallel port is a type of interface found on computers (personal and otherwise) for connecting peripherals. Parallel ports send multiple bits of

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



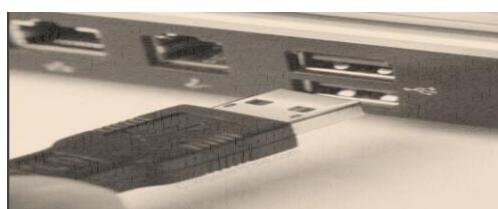
**7.4 ETHERNET:** A Ethernet port is a jack or socket on a computer that allows the use of an Ethernet connector. These ports are essential in allowing the creation of local area networks (LANs). An Ethernet port is usually found on networking devices, including computers, routers, video game consoles, 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.



## EXPERIMENT-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.

```
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ mkdir INDIA
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ cd INDIA
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/INDIA$ mkdir KERALA TN AP
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/INDIA$ ls
AP KERALA TN
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/INDIA$ cd KERALA
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/INDIA/KERALA$ touch Culture.txt Language.txt Dress.txt
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/INDIA/KERALA$ ls
Culture.txt Dress.txt Language.txt
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/INDIA/KERALA$ cd ..
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/INDIA$ ls
AP KERALA TN
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/INDIA$ cd TN
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/INDIA/TN$ touch Fam.txt Land.txt
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/INDIA/TN$ ls
Fam.txt Land.txt
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/INDIA/TN$ cd ..
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/INDIA$ cd AP
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/INDIA/AP$ touch Industry.txt Farmers.txt General.txt
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/INDIA/AP$ ls
Farmers.txt General.txt Industry.txt
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/INDIA/AP$
```

```
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ cd INDIA/AP
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/INDIA/AP$ mkdir Development
```

Q2. List your present working directory

```
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/INDIA/AP$ cd ~
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ pwd
/home/onworks
```

Q3. Move to the root directory

```
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ cd /
```

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

```
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ cd INDIA/KERALA
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/INDIA/KERALA$ ls
Culture.txt Dress.txt Language.txt
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/INDIA/KERALA$ cp Culture.txt ~/INDIA/AP
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/INDIA/KERALA$ ls
Culture.txt Dress.txt Language.txt
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/INDIA/KERALA$ cd ~/INDIA/AP
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/INDIA/AP$ ls
Culture.txt Development Farmers.txt General.txt Industry.txt
```

Q5. Display the content of the file ‘General’

```
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/INDIA/AP$ cat > General.txt
This is general file^Z
[1]+  Stopped                  cat > General.txt
```

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

```
onworks@onworks-Standard-PC-l440FX-PIIX-1996:~/INDIA/AP$ cd ~/INDIA/KERALA
onworks@onworks-Standard-PC-l440FX-PIIX-1996:~/INDIA/KERALA$ ls
Culture.txt Dress.txt Language.txt
onworks@onworks-Standard-PC-l440FX-PIIX-1996:~/INDIA/KERALA$ mv Language.txt ~/INDIA/AP/Development
onworks@onworks-Standard-PC-l440FX-PIIX-1996:~/INDIA/KERALA$ ls
Culture.txt Dress.txt
```

Q7. List all the files in the folder AP

```
onworks@onworks-Standard-PC-l440FX-PIIX-1996:~/INDIA/KERALA$ cd ~/INDIA/AP
onworks@onworks-Standard-PC-l440FX-PIIX-1996:~/INDIA/AP$ ls
Culture.txt Development Farmers.txt General.txt Industry.txt
```

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

```
onworks@onworks-Standard-PC-l440FX-PIIX-1996:~/INDIA/KERALA$ head -10 Dress.txt
line 1
line 2
line 3
line 4
line 5
line 6
line 7
line 8
line 9
line 10
```

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

```
onworks@onworks-Standard-PC-l440FX-PIIX-1996:~/INDIA/KERALA$ tail -10 Dress.txt
line 5
line 6
line 7
line 8
line 9
line 10
line 11
line 12
line 13
line 14
```

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

```
onworks@onworks-Standard-PC-l440FX-PIIX-1996:~/INDIA/AP$ ls -l
total 4
-rw-rw-r-- 1 onworks onworks 0 Aug 30 06:50 Culture.txt
drwxrwxr-x 2 onworks onworks 4096 Aug 30 06:53 Development
-rw-rw-r-- 1 onworks onworks 0 Aug 30 06:48 Farmers.txt
-rw-rw-r-- 1 onworks onworks 0 Aug 30 06:51 General.txt
-rw-rw-r-- 1 onworks onworks 0 Aug 30 06:48 Industry.txt
```

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

```
onworks@onworks-Standard-PC-l440FX-PIIX-1996:~/INDIA/AP$ ls F*
Farmers.txt
```

Q12. List the files page by page

```
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/INDIA/AP\$ ls -la |more
total 12
drwxrwxr-x 3 onworks onworks 4096 Aug 30 06:50 .
drwxrwxr-x 5 onworks onworks 4096 Aug 30 06:46 ..
-rw-rw-r-- 1 onworks onworks 0 Aug 30 06:50 Culture.txt
drwxrwxr-x 2 onworks onworks 4096 Aug 30 06:53 Development
-rw-rw-r-- 1 onworks onworks 0 Aug 30 06:48 Farmers.txt
-rw-rw-r-- 1 onworks onworks 0 Aug 30 06:51 General.txt
-rw-rw-r-- 1 onworks onworks 0 Aug 30 06:48 Industry.txt
```

Q13. Remove the file ‘general’

```
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/INDIA/AP\$ rm General.txt
```

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

```
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/INDIA/KERALA\$ chmod g-wx Culture.txt
```

Q15. List the lines of the file which contains a string ‘Saree’

```
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/INDIA/KERALA\$ cat >Dress.txt
Shirt
Pants
Churidar
Saree
Shorts
^Z
[1]+  Stopped                  cat > Dress.txt
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/INDIA/KERALA\$ grep 'Saree' Dress.txt
Saree
```

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

```
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/INDIA/KERALA\$ wc -mwl ~/INDIA/KERALA/Dress.txt
5 5 34 /home/onworks/INDIA/KERALA/Dress.txt
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/INDIA/KERALA\$
```

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

```
MAN(1)                                Manual pager utils                               MAN(1)

NAME
    man - an interface to the system reference manuals

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

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

    The table below shows the section numbers of the manual followed by the types of pages they
    contain.

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

    A manual page consists of several sections.

    Conventional section names include NAME, SYNOPSIS, CONFIGURATION, DESCRIPTION, OPTIONS,
    Manual page man(1) line 1 (press h for help or q to quit)
```

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

```
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ ls -l >> filelist.txt
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ cat filelist.txt
total 40
drwxr-xr-x 2 onworks onworks 4096 Nov 29 2020 Desktop
drwxr-xr-x 2 onworks onworks 4096 Nov 29 2020 Documents
drwxr-xr-x 2 onworks onworks 4096 Nov 29 2020 Downloads
-rw-rw-r-- 1 onworks onworks 0 Aug 30 07:27 filelist.txt
drwxrwxr-x 5 onworks onworks 4096 Aug 30 06:46 INDIA
drwxr-xr-x 2 onworks onworks 4096 Nov 29 2020 Music
drwxr-xr-x 2 onworks onworks 4096 Nov 29 2020 Pictures
drwxr-xr-x 2 onworks onworks 4096 Nov 29 2020 Public
drwxr-xr-x 3 onworks onworks 4096 Nov 29 2020 snap
drwxr-xr-x 2 onworks onworks 4096 Nov 29 2020 Templates
drwxr-xr-x 2 onworks onworks 4096 Nov 29 2020 Videos
```

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

```
onworks@onworks-Standard-PC-t440FX-PIIX-1996:~/INDIA/KERALA$ ps -aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START  TIME COMMAND
root       1  0.2  0.4 103360 12664 ?        Ss   06:41  0:06 /sbin/init splash
root       2  0.0  0.0     0   0 ?        S    06:41  0:00 [kthreadd]
root       3  0.0  0.0     0   0 ?        I<  06:41  0:00 [rcu_gp]
root       4  0.0  0.0     0   0 ?        I<  06:41  0:00 [rcu_par_gp]
root       6  0.0  0.0     0   0 ?        I<  06:41  0:00 [kworker/0:0H-kblockd]
root       8  0.0  0.0     0   0 ?        I<  06:41  0:00 [mm_percpu_wq]
root       9  0.0  0.0     0   0 ?        S    06:41  0:00 [ksoftirqd/0]
root      10  0.0  0.0     0   0 ?        I    06:41  0:02 [rcu_sched]
root      11  0.0  0.0     0   0 ?        S    06:41  0:00 [migration/0]
root      12  0.0  0.0     0   0 ?        S    06:41  0:00 [idle_inject/0]
root      14  0.0  0.0     0   0 ?        S    06:41  0:00 [cpuhp/0]
root      15  0.0  0.0     0   0 ?        S    06:41  0:00 [cpuhp/1]
root      16  0.0  0.0     0   0 ?        S    06:41  0:00 [idle_inject/1]
root      17  0.0  0.0     0   0 ?        S    06:41  0:00 [migration/1]
root      18  0.0  0.0     0   0 ?        S    06:41  0:00 [ksoftirqd/1]
root      20  0.0  0.0     0   0 ?        I<  06:41  0:00 [kworker/1:0H-kblockd]
root      21  0.0  0.0     0   0 ?        S    06:41  0:00 [kdevtmpfs]
root      22  0.0  0.0     0   0 ?        I<  06:41  0:00 [netns]
root      23  0.0  0.0     0   0 ?        S    06:41  0:00 [rcu_tasks_kthre]
root      24  0.0  0.0     0   0 ?        S    06:41  0:00 [kaudittd]
root      25  0.0  0.0     0   0 ?        S    06:41  0:00 [khungtaskd]
root      26  0.0  0.0     0   0 ?        S    06:41  0:00 [oom_reaper]
root      27  0.0  0.0     0   0 ?        I<  06:41  0:00 [writeback]
root      28  0.0  0.0     0   0 ?        S    06:41  0:00 [kcompactd0]
root      29  0.0  0.0     0   0 ?        SN   06:41  0:00 [ksmd]
root      30  0.0  0.0     0   0 ?        SN   06:41  0:00 [khugepaged]
root      77  0.0  0.0     0   0 ?        I<  06:41  0:00 [kintegrityd]
root      78  0.0  0.0     0   0 ?        I<  06:41  0:00 [kblockd]
root      79  0.0  0.0     0   0 ?        I<  06:41  0:00 [blkcg_punt_bio]
root      80  0.0  0.0     0   0 ?        I<  06:41  0:00 [tpm_dev_wq]
root      81  0.0  0.0     0   0 ?        I<  06:41  0:00 [ata_sff]
root      82  0.0  0.0     0   0 ?        I<  06:41  0:00 [md]
root      83  0.0  0.0     0   0 ?        I<  06:41  0:00 [edac-poller]
root      84  0.0  0.0     0   0 ?        I<  06:41  0:00 [devfreq_wq]
root      85  0.0  0.0     0   0 ?        S    06:41  0:00 [watchdogd]
```

Q20. List the disk partitions in your harddisk.

```
onworks@onworks-Standard-PC-t440FX-PIIX-1996:~$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
fd0     2:0    1   4K  0 disk
loop0   7:0    0  55M  1 loop /snap/core18/1880
loop1   7:1    0 255,6M  1 loop /snap/gnome-3-34-1804/36
loop2   7:2    0  62,1M  1 loop /snap/gtk-common-themes/1506
loop3   7:3    0  49,8M  1 loop /snap/snap-store/467
loop4   7:4    0  29,9M  1 loop /snap/snapd/8542
loop5   7:5    0  32,3M  1 loop /snap/snapd/128883
loop6   7:6    0  55,4M  1 loop /snap/core18/2128
loop7   7:7    0  51M   1 loop /snap/snap-store/547
loop8   7:8    0  65,1M  1 loop /snap/gtk-common-themes/1515
loop9   7:9    0  219M  1 loop /snap/gnome-3-34-1804/72
sda     8:0    0   30G  0 disk
└─sda1  8:1    0  512M  0 part /boot/efi
└─sda2  8:2    0    1K  0 part
└─sda5  8:5    0  29,5G  0 part /
sr0    11:0   1 1024M  0 rom
```

Q21. Redirect the output of the top program to a file called 'error'.

```
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ top >> error.txt
[1]+  Stopped                  top >> error.txt
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ cat error.txt
top - 07:35:42 up 54 min, 1 user, load average: 0,15, 0,10, 0,11
Tasks: 179 total, 2 running, 177 sleeping, 0 stopped, 0 zombie
%CPU(s): 1,8 us, 2,3 sy, 0,0 ni, 94,4 id, 1,2 wa, 0,0 hi, 0,3 si, 0,0 st
MiB Mem : 2920,5 total, 267,8 free, 934,4 used, 1718,4 buff/cache
MiB Swap: 1401,6 total, 1398,8 free, 2,8 used. 1800,5 avail Mem

PID USER      PR  NI    VIRT    RES   SHR S %CPU %MEM TIME+ COMMAND
 182 root      0 -20      0     0   0 I  0,3  0,0  0:03.64 kworker/0:1H-kblockd
 1052 onworks  20   0 4134612 317856 117852 S  0,3 10,6  0:50.86 gnome-shell
    1 root      20   0 103360 12664  8420 S  0,0  0,4  0:06.16 systemd
    2 root      20   0      0     0   0 S  0,0  0,0  0:00.00 kthreadd
    3 root      0 -20      0     0   0 I  0,0  0,0  0:00.00 rCU_gp
    4 root      0 -20      0     0   0 I  0,0  0,0  0:00.00 rCU_par_gp
    6 root      0 -20      0     0   0 I  0,0  0,0  0:00.00 kworker/0:0H-kblockd
    8 root      0 -20      0     0   0 I  0,0  0,0  0:00.00 mm_percpu_wq
    9 root      20   0      0     0   0 S  0,0  0,0  0:00.19 ksoftirqd/0
   10 root     20   0      0     0   0 R  0,0  0,0  0:02.86 rCU_sched
   11 root     rt  0      0     0   0 S  0,0  0,0  0:00.02 migration/0
   12 root    -51  0      0     0   0 S  0,0  0,0  0:00.00 idle_inject/0
   14 root     20   0      0     0   0 S  0,0  0,0  0:00.00 cpuhp/0
   15 root     20   0      0     0   0 S  0,0  0,0  0:00.00 cpuhp/1
   16 root    -51  0      0     0   0 S  0,0  0,0  0:00.00 idle_inject/1
   17 root     rt  0      0     0   0 S  0,0  0,0  0:00.18 migration/1
   18 root     20   0      0     0   0 S  0,0  0,0  0:00.69 ksoftirqd/1
   20 root     0 -20      0     0   0 I  0,0  0,0  0:00.00 kworker/1:0H-kblockd
   21 root     20   0      0     0   0 S  0,0  0,0  0:00.00 kdevtmpfs
   22 root     0 -20      0     0   0 I  0,0  0,0  0:00.00 netns
   23 root     20   0      0     0   0 S  0,0  0,0  0:00.00 rCU_tasks_kthre
   24 root     20   0      0     0   0 S  0,0  0,0  0:00.02 kauditd
   25 root     20   0      0     0   0 S  0,0  0,0  0:00.00 khungtaskd
   26 root     20   0      0     0   0 S  0,0  0,0  0:00.00 oom_reaper
   27 root     0 -20      0     0   0 I  0,0  0,0  0:00.00 writeback
```

## EXPERIMENT-3

### 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:~$ 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
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.

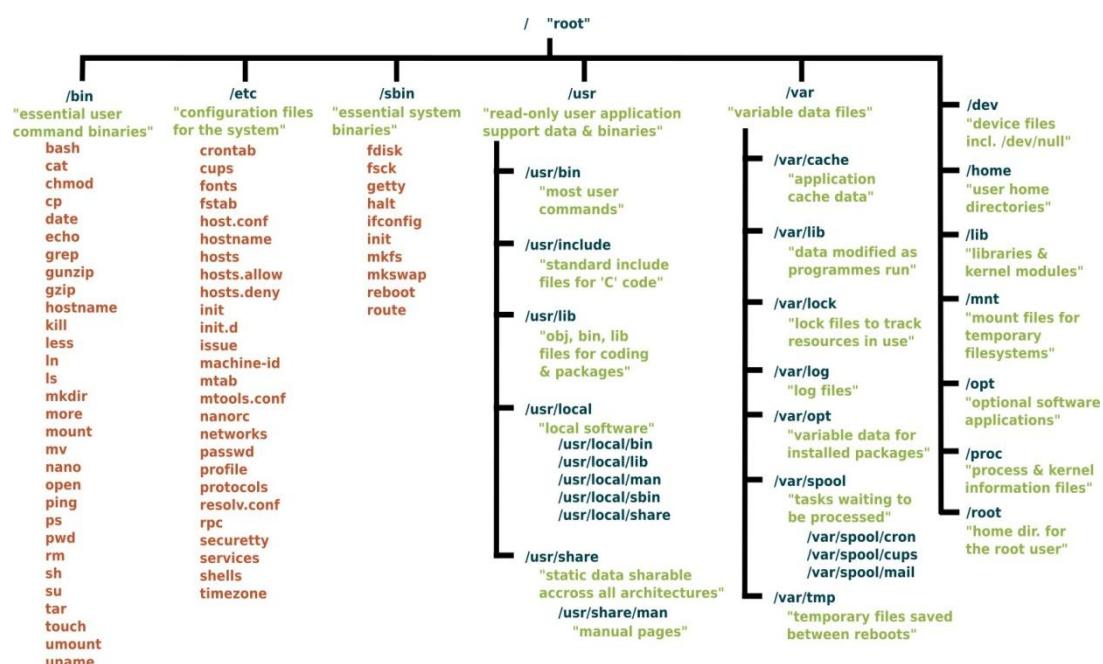
Be that as it may, */var* contains things like logs in the */var/log* subdirectories. Logs are files that register events that happen on the system. If something fails in the kernel, it will be logged in a file in */var/log*; if someone tries to break into your computer from outside, your firewall will 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 filesand 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 explore the filesystem yourself, use the cd command:

cd

will take you to the directory of your choice (*cd* stands for *change directory*). If you get confused,

pwd

will always tell you where you (*pwd* stands for *print working directory*). Also, cd with no options or parameters, will take you back to your own home directory, where things are safe and cosy.

Finally, cd ..

will take you up one level, getting you one level closer to the / root directory. If you are in */usr/share/wallpapers* and run cd .., you will move up to */usr/share*.

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.

## EXPERIMENT-4

### SHELL SCRIPT

#### **1. Write a Shell program to display a given message**

```
echo "enter a message"  
read message  
echo "The entered message is $message"
```

#### **Output**

```
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ chmod +x shell1.1.sh  
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ ./shell1.1.sh  
enter a message  
i am akshaya  
The entered message is i am akshaya  
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ █
```

#### **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 - $b`  
echo "sub=$c"  
c=`expr $a / $b`  
echo "div=$c"  
c=`expr $a \* $b`  
echo "multiplication=$c"  
c=`expr $a % $b`  
echo "remainder=$c"
```

#### **Output**

```
user@user-HP-Notebook:~/Desktop$ chmod +x shell1.sh  
user@user-HP-Notebook:~/Desktop$ ./shell1.sh  
Enter two integer number  
10  
20  
sum=30  
sub=-10  
div=0  
multiplication=200  
remainder=10
```

#### **3. Write a shell Script to determine largest among three integer number.**

```
echo "enter three integer number"
```

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

### **Output**

```
user@user-HP-Notebook:~/Desktop$ chmod +x shell2.sh  
user@user-HP-Notebook:~/Desktop$ ./shell2.sh  
enter three integer number  
10  
20  
30  
30 is largest number
```

### **4. Write a shell script to compare two string.**

```
echo "enter two string"  
read a read b  
if [ -z $a ] then echo " First String is  
empty: Null String" fi  
if [ -z $b ] then  
echo " First String is empty: Null String"  
fi if [ $a = $b ] then echo "Strings are equal:  
strings Matched" else echo "Strings are not  
equal: Strings not match"  
fi
```

### **Output**

```
user@user-HP-Notebook:~/Desktop$ chmod +x shell3.sh
user@user-HP-Notebook:~/Desktop$ ./shell3.sh
enter 2 strings
ammu
amma
Strings are not equal:strings not match
user@user-HP-Notebook:~/Desktop$ chmod +x shell3.sh
user@user-HP-Notebook:~/Desktop$ ./shell3.sh
enter 2 strings
amma
amma
Strings are equal:strings matched
```

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

```
echo "enter name of directory"
read dir
if [ -d $dir ] then echo
"Directory $dir Exists!" else
mkdir $dir
fi
```

**Output**

```
user@user-HP-Notebook:~/Desktop$ chmod +x shell4.sh
user@user-HP-Notebook:~/Desktop$ ./shell4.sh
Enter name of directory
Pictures
user@user-HP-Notebook:~/Desktop$ chmod +x shell4.sh
user@user-HP-Notebook:~/Desktop$ ./shell4.sh
Enter name of directory
Pictures
Directory Pictures exists!
```

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

```
echo "enter name of file"
read filename if [ -f
$filename ]
then echo "File $filename
Exists!"
else touch
$filename
fi
```

**Output**

```
user@user-HP-Notebook:~/Desktop$ chmod +x shell5.sh
user@user-HP-Notebook:~/Desktop$ ./shell5.sh
enter name of file
Downloads
user@user-HP-Notebook:~/Desktop$ chmod +x shell5.sh
user@user-HP-Notebook:~/Desktop$ ./shell5.sh
enter name of file
Downloads
file Downloads exists!
```

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

```
echo "enter two integer values" read a read b echo -e "Menu \n 1 for Addition \n 2 for Substraction \n 3 for Multiplication \n 4 for Division \n 5 for Remainder" echo "enter choice" read ch case $ch in
1) echo "Sum=$(expr $a + $b)";;
2) echo "Substraction=$(expr $a - $b)";;
3) echo "Multiplication=$(expr $a \* $b)";;
4) echo "Division=$(expr $a / $b)";;
5) echo "Remainder=$(expr $a % $b)";; 6)
   echo "invalid Choice:Try Again!" esac
```

## Output

```
user@user-HP-Notebook:~/Desktop$ ./shell6.sh
enter 2 integer values
10
20
menu
 1 for Addition
 2 for Subtraction
 3 for Multiplication
 4 for Division
 5 for Remainder
enter choice
6
invalid choice:try again!
user@user-HP-Notebook:~/Desktop$ ./shell6.sh
enter 2 integer values
10
20
menu
 1 for Addition
 2 for Subtraction
 3 for Multiplication
 4 for Division
 5 for Remainder
enter choice
2
Subtraction=-10
```

```

user@user-HP-Notebook:~/Desktop$ chmod +x shell6.sh
user@user-HP-Notebook:~/Desktop$ ./shell6.sh
enter 2 integer values
10
20
menu
1 for Addition
2 for Subtraction
3 for Multiplication
4 for Division
5 for Remainder
enter choice
3
Multiplication=200
user@user-HP-Notebook:~/Desktop$ ./shell6.sh
enter 2 integer values
10
20
menu
1 for Addition
2 for Subtraction
3 for Multiplication
4 for Division
5 for Remainder
enter choice
5
Remainder=10

```

### **8. Write a shell script to do:**

- a. display list of directory contents**
- b. Name of current directory**
- c. Who is logged on**
- d. Long listing of directory contents according to choose of user.**

```

echo -e "Menu \n 1 for listing directory content \n 2 for print name of current
directory \n 3 for Show who is logged on \n 4 Show directory content using long
listing format " echo "enter your choice " read ch
case $ch in
1) ls;;
2) pwd;;
3) who;;
4) ls -l;;
*) echo "Invalid Choice: Try Again!!" esac

```

## Output

```
user@user-HP-Notebook:~/Desktop$ ./shell7.sh
Menu
 1 for listing directory content
 2 for print name for current directory
 3 for show who is logged on
 4 show directory content using long listing format
enter your choice
4
ashku.ora
Cheese photos
Desktop
Downloads
Pictures
shell1.png
shell1.sh
shell2.png
shell2.sh
shell3.png
shell3.sh
shell4.png
shell4.sh
shell5.png
shell5.sh
shell6.1.png
shell6.png
shell6.sh
shell7.sh
user@user-HP-Notebook:~/Desktop$ chmod +x shell7.sh
user@user-HP-Notebook:~/Desktop$ ./shell7.sh
Menu
 1 for listing directory content
 2 for print name for current directory
 3 for show who is logged on
 4 show directory content using long listing format
enter your choice
1
ashku.ora      Pictures      shell2.sh    shell4.sh      shell6.png
Cheese photos  shell1.png    shell3.png    shell5.png    shell6.sh
Desktop        shell1.sh    shell3.sh    shell5.sh    shell7.sh
Downloads      shell2.png    shell4.png    shell6.1.png
user@user-HP-Notebook:~/Desktop$ ./shell7.sh
Menu
 1 for listing directory content
 2 for print name for current directory
 3 for show who is logged on
 4 show directory content using long listing format
enter your choice
2
/home/user/Desktop
user@user-HP-Notebook:~/Desktop$ ./shell7.sh
Menu
 1 for listing directory content
 2 for print name for current directory
 3 for show who is logged on
 4 show directory content using long listing format
enter your choice
3
user      :0          2021-09-28 18:25 (:0)
user      pts/4        2021-09-28 18:26 (:0.0)
```

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

```
cat > Studentdetail echo "Name of the  
student: $1" echo "Roll Number of the  
student: $2" echo "Marks of the student:  
$3"  
ctrl+D  
./ Studentdetail your name your roll no marks
```

**Output**

```
user@user-HP-Notebook:~/Desktop$ chmod +x shell8.sh  
user@user-HP-Notebook:~/Desktop$ ./shell8.sh sree 15 60  
Name of the student:sree  
Roll number of the student:15  
Marks of the student:60
```

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

```
cat> file2.sh  
a=12  
echo $a  
echo "$a"  
echo '$a'
```

**Output**

```
user@user -HP-Notebook:~/Desktop$ cat>shell9.sh  
a=12  
echo $a  
echo "$a"  
echo '$a'  
user@user-HP-Notebook:~/Desktop$ chmod +x shell9.sh  
user@user-HP-Notebook:~/Desktop$ ./shell9.sh  
12  
12  
$a
```

- 11. To check whether there is any entry in the month of May in the system log.**

```
if [[ -e /var/log/syslog ]]  
then  
    cat /var/log/syslog | grep "May" else  
    echo "File not found"  
fi
```

## Output

```
ann@ann-VirtualBox:~/Shell pgmng$ chmod +x s11.sh
ann@ann-VirtualBox:~/Shell pgmng$ ./s11.sh
File not found
ann@ann-VirtualBox:~/Shell pgmng$ 
```

### 12. Implement arithmetic calculator using Functions add()

```
{
    i=$1    j=$2
    ((k=i+j))
    echo Sum is $k
}
sub()
{
    i=$1
    j=$2
    ((k=i-j))
    echo Sum is $k
}
mul()
{
    i=$1    j=$2
    ((k=i*j))
    echo Sum is $k
}
echo "Enter your option: 1: Add, 2:Subtract, 3: Multiply"
read i
case $i in
1) add 1 2;;
2) sub 1 2;;    3) mul 1 2;;
esac
```

### **Output**

```
user@user-HP-Notebook:~/Desktop$ chmod +x shell11.sh
user@user-HP-Notebook:~/Desktop$ ./shell11.sh
Enter your option:1:Add,2:Subtraction,3:Multiplication
3
sum is 2
user@user-HP-Notebook:~/Desktop$ ./shell11.sh
Enter your option:1:Add,2:Subtraction,3:Multiplication
2
sum is -1
user@user-HP-Notebook:~/Desktop$ ./shell11.sh
Enter your option:1:Add,2:Subtraction,3:Multiplication
1
sum is 3
```

### **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
```

#### **b. Using While loop i=0**

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

### **Output**

```
user@user-HP-Notebook:~/Desktop$ chmod +x shell12.2.sh
user@user-HP-Notebook:~/Desktop$ ./shell12.2.sh
45
user@user-HP-Notebook:~/Desktop$ chmod +x shell12.1.sh
user@user-HP-Notebook:~/Desktop$ ./shell12.1.sh
45
```

## **EXPERIMENT- 5**

### **LAMP SERVER INSTALLATION ON UBUNTU WITH APACHE**

#### **1. STEP 1: PREPARING YOUR UBUNTU SERVER**

To begin with, you need a cloud server to run the LAMP stack software. If you are new to UpCloud, have a look at our quick started guide for [deploying your first cloud server](#) and [how to connect to it](#). First of all, ensure everything is up to date on your server:

```
sudo apt update
```

```
sudo apt upgrade
```

Now open ports 22 (for SSH), 80 and 443 and enable Ubuntu Firewall (ufw):

```
sudo ufw allow ssh
```

```
sudo ufw allow 80
```

```
sudo ufw allow 443
```

```
sudo ufw enable
```

#### **2. Step 2: Installing and testing Apache2**

Install Apache using apt:

```
sudo apt install apache2
```

Confirm that Apache is now running with the following command:

```
sudo systemctl status apache2
```

You should get an output showing that the apache2.service is running and enabled.

```
● apache2.service - The Apache HTTP Server
```

```
Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
```

```
Active: active (running) since Tue 2020-11-03 10:32:26 UTC; 1min 6s ago
```

```
Docs: https://httpd.apache.org/docs/2.4/
```

```
Main PID: 52943 (apache2)
```

```
Tasks: 7 (limit: 2282)
```

```
Memory: 11.9M
```

```
CGroup: /system.slice/apache2.service
```

```
    ├─52943 /usr/sbin/apache2 -k start
```

```
    ├─52944 /usr/sbin/apache2 -k start
```

```
    ├─52945 /usr/sbin/apache2 -k start
```

```
    ├─52946 /usr/sbin/apache2 -k start
```

```
    ├─52947 /usr/sbin/apache2 -k start
```

```
    ├─52948 /usr/sbin/apache2 -k start
```

```
    └─52953 /usr/sbin/apache2 -k start
```

```
Once installed, test by accessing your server's IP in your browser:
```

```
http://YOURSERVERIPADDRESS/
```

You should see a page with an “Apache2 Ubuntu Default” header showing that Apache2 has been installed successfully. If you do not see this, please ensure that the previous commands in this section have completed without error

### **3. STEP 3: INSTALLING AND TESTING PHP 7.4**

PHP 7.4 is the latest available right now so let's install that along with some regularly used modules:

```
sudo apt install php7.4 php7.4-mysql php-common php7.4-cli php7.4-json php7.4-common php7.4-opcache libapache2-mod-php7.4
```

Check the installation and version:

```
php --version
```

Restart Apache for the changes to take effect:

```
sudo systemctl restart apache2
```

Create a phpinfo.php test page:

```
echo '<?php phpinfo(); ?>' | sudo tee -a /var/www/html/phpinfo.php > /dev/null
```

Test everything is working by accessing the following in your browser:

```
http://YOURSERVERIPADDRESS/phpinfo.php
```

You should see a PHP Version 7.4.3 page listing all of your PHP options. If you don't or it tries to download a file, double-check that all of the above steps have completed without error. Once you've confirmed that PHP is working correctly, delete the test file.

```
sudo rm /var/www/html/phpinfo.php
```

The information displayed in the PHP info could be used to find an attack vector against your web server so best not to leave it publicly accessible.

### **4. STEP 4: INSTALLING AND SECURING MARIADB**

MariaDB is a fork of MySQL from some of the original MySQL team and is a drop-in replacement. We'll be using this over MySQL itself in this guide!

Install the required packages:

```
sudo apt install mariadb-server mariadb-client
```

Once installed, check it's running correctly:

```
sudo systemctl status mariadb
```

## **EXPERIMENT- 6**

### **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
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
        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-js
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.

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 libcroco3 libdebservice-perl libfile-stripnondeterminism-perl libltl-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 libtool 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 libcroco3 libdebservice-perl libfile-stripnondeterminism-perl
  libltl-dev libmail-sendmail-perl libpcre2-16-0 libpcre2-dev libpcre2-posix2 libsigsegv2 libssl-dev libsub-override-perl
  libsys-hostname-long-perl libtool m4 php-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 libsigsegv2 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 1:1.16.1-4ubuntu6 [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.

...

**cgi.fix\_pathinfo=0**

...

```
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.**
; http://php.net/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
;cgi.redirect_status_env = 1

; 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=1

; 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
```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo  
^X Exit ^R Read File ^A Replace ^U Paste Text ^T To Spell ^G Go To Line M-E Redo

```

GNU nano 4.8                               php.ini
; **You CAN safely turn this off for IIS, in fact, you MUST.**
; Thunderbird Mail.net/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
: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-
  generic
  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"
'

^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
```

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
[Progress Bar]
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 fakernph/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 [ Read 16 lines ]  
^X Exit ^R Read File ^\ Replace ^K Cut Text ^J Justify  
 ^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
```

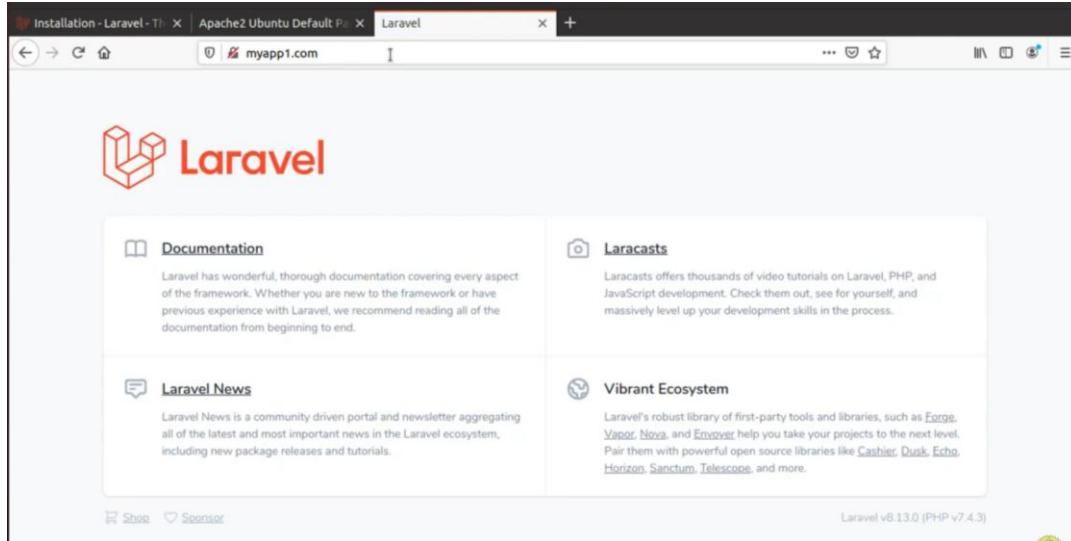
```
GNU nano 4.8                               /etc/hosts
127.0.0.1      localhost
127.0.1.1      ebin-VirtualBox

# 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      ^K Cut Text      ^J Justify
^X Exit          ^R Read File       ^\ Replace       ^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.

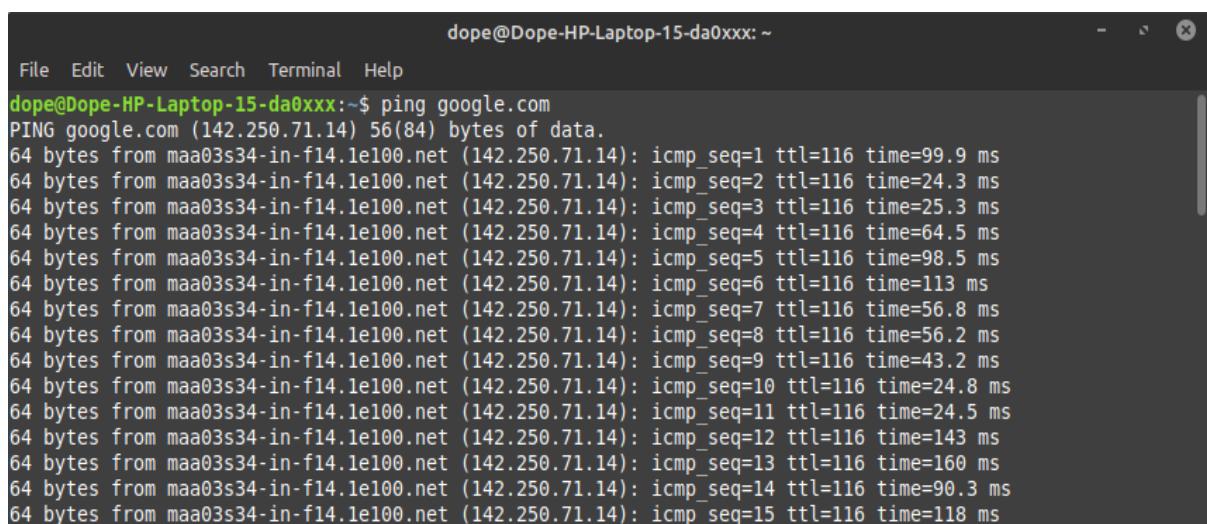
## EXPERIMENT-7

### 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:~$ 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  202.785 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)  30.012 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 network 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]

```
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:~$
```

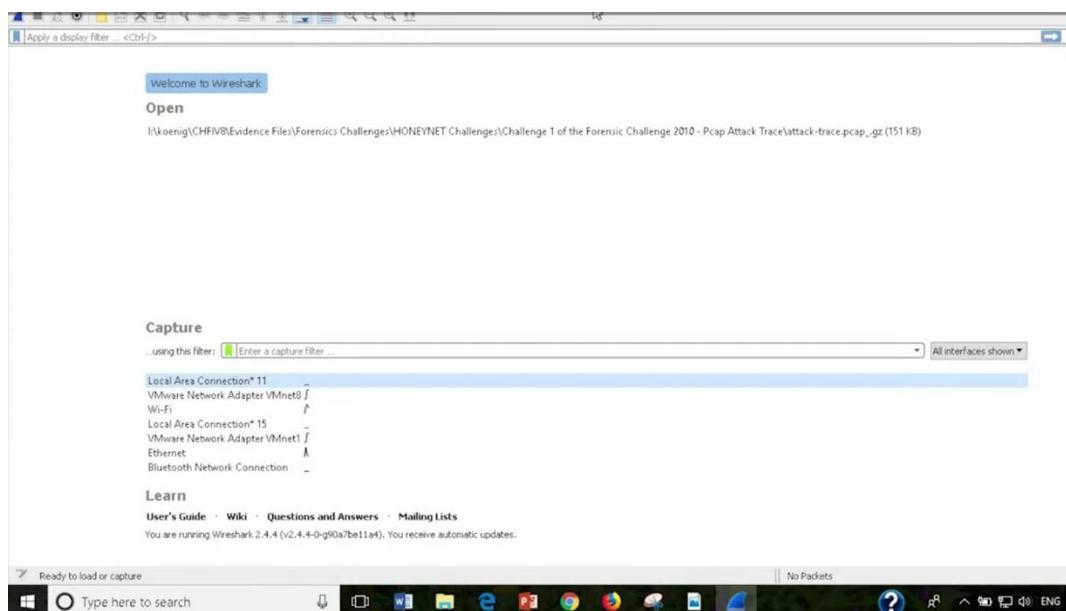
## **EXPERIMENT- 8**

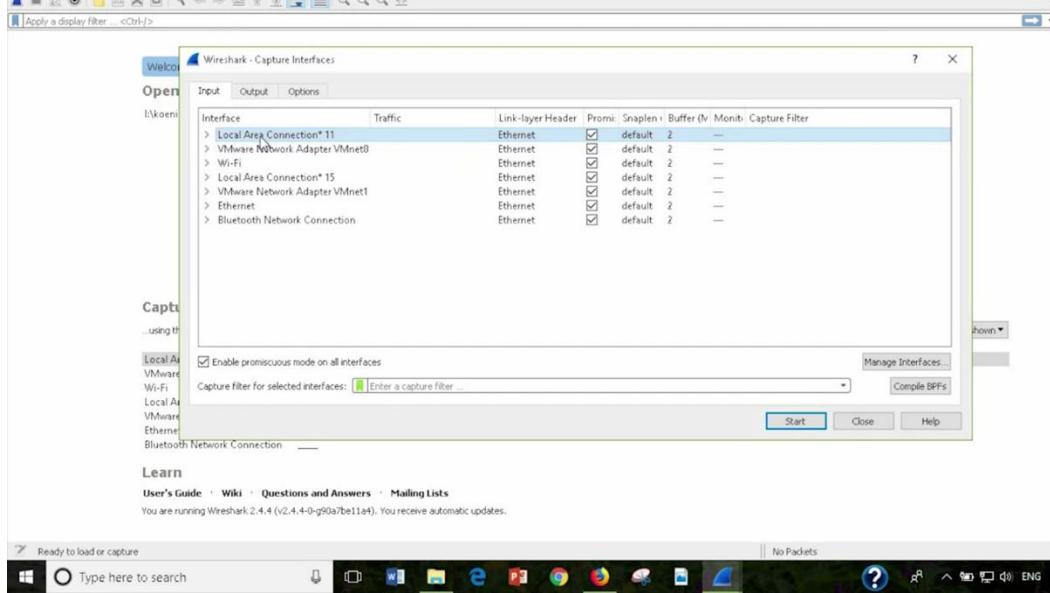
### **INTRODUCTION TO WIRESHARK**

Wireshark is an open-source packet analyzer, which is used for education, analysis, software development, communication protocol development, and network troubleshooting. It is used to track the packets so that each one is filtered to meet our specific needs. It is commonly called as a sniffer, network protocol analyzer, and network analyzer. It is also used by network security engineers to examine security problems.

#### **FEATURES OF WIRESHARK**

- Available for UNIX and Windows.
- Capture live packet data from a network interface.
- Display packets with very detailed protocol information.
- Save packet data captured.
- Filter packet data on our criteria.
- Colorize packet data on our criteria.
- Export some or all packet data in a number of capture file formats.

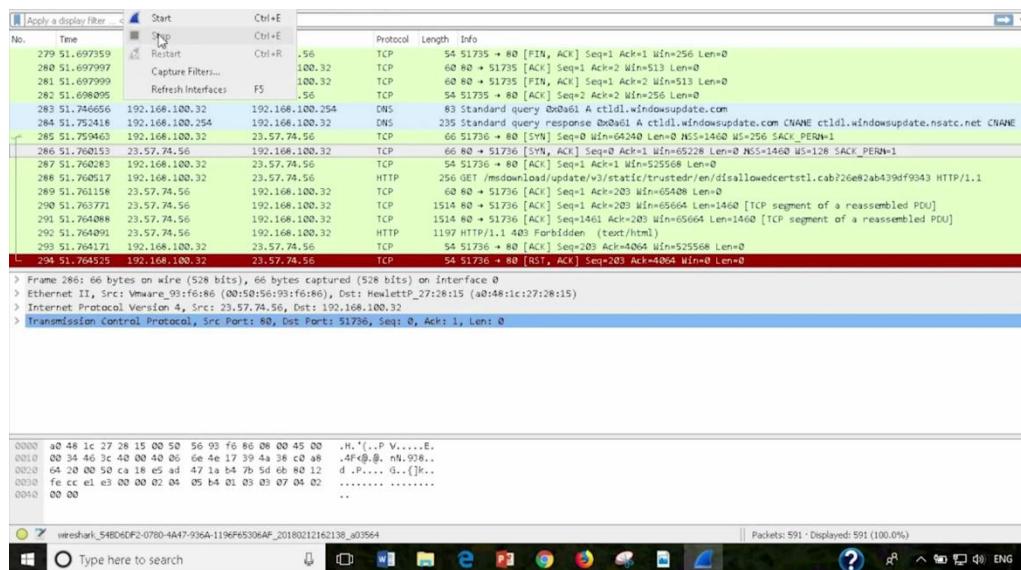
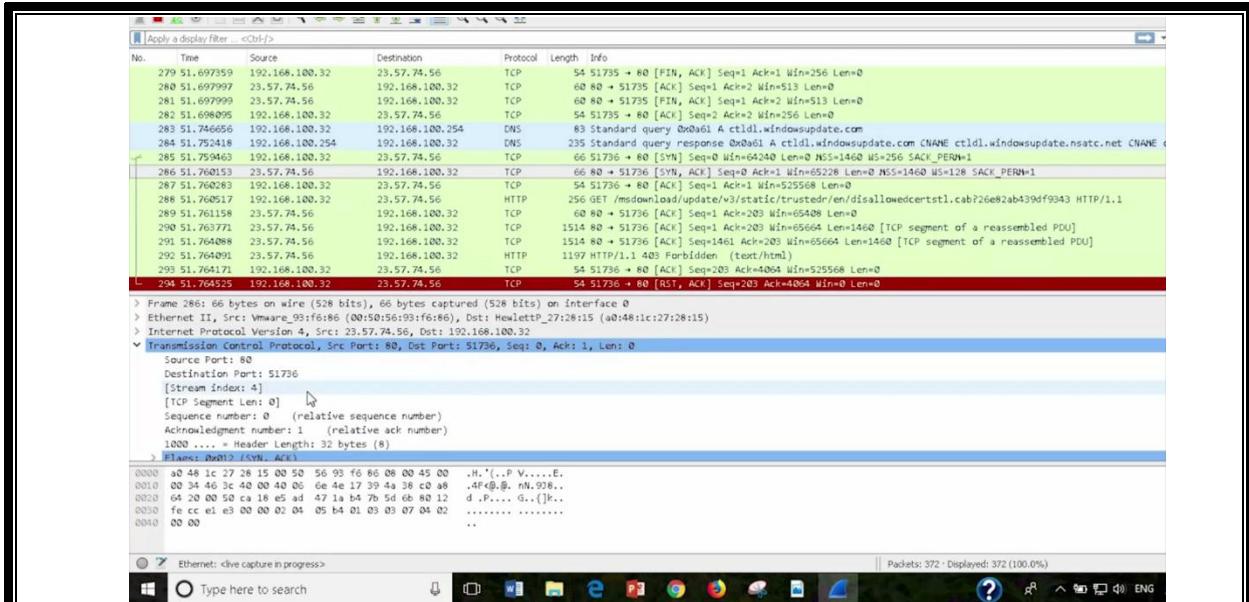




No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	fe80::264e:9189%89d4	ff02:11:2	DHCPv6	157	Solicit XID: 0xe0344 CID: 000100011b63af9f00505699139d
2	0.042910	Vmware_931b:51	Broadcast	ARP	60	Who has 172.32.100.1? Tell 172.32.0.44
3	0.105367	192.168.100.32	192.168.100.254	DNS	75	Standard query 0xa07b A docs.google.com
4	0.106411	192.168.100.254	192.168.100.32	DNS	91	Standard query response 0xa07b A docs.google.com A 172.217.24.238
5	0.146168	192.168.100.32	172.217.24.227	QUIC	1392	Client Hello, PKN: 1, CID: 187093422062879262
6	0.151319	192.168.100.32	172.217.24.238	QUIC	1392	Client Hello, PKN: 1, CID: 6575440334439518126
7	0.160495	192.168.100.32	172.217.24.227	QUIC	1392	Client Hello, PKN: 2, CID: 187093422062879262
8	0.169818	172.217.24.227	192.168.100.32	QUIC	80	Payload (Encrypted), PKN: 1, CID: 187093422062879262
9	0.172496	192.168.100.32	172.217.24.238	QUIC	1392	Client Hello, PKN: 2, CID: 6575440334439518126
10	0.182190	172.217.24.238	192.168.100.32	QUIC	80	Payload (Encrypted), PKN: 1, CID: 6575440334439518126
11	0.230375	172.217.24.238	192.168.100.32	QUIC	1392	Payload (Encrypted), PKN: 2
12	0.230379	172.217.24.238	192.168.100.32	QUIC	73	Payload (Encrypted), PKN: 3
13	0.231515	192.168.100.32	172.217.24.238	QUIC	81	Payload (Encrypted), PKN: 3, CID: 6575440334439518126
14	0.231696	192.168.100.32	172.217.24.238	QUIC	83	Payload (Encrypted), PKN: 4, CID: 6575440334439518126
15	0.251474	172.217.24.227	192.168.100.32	QUIC	1392	Payload (Encrypted), PKN: 2
16	0.251481	172.217.24.227	192.168.100.32	QUIC	73	Payload (Encrypted), PKN: 3

Frame 1: 157 bytes on wire (1256 bits), 157 bytes captured (1256 bits) on interface 0  
 Ethernet II, Src: Vmware\_931b:90 (00:50:56:93:1b:90), Dst: IPv4mcast\_01:00:02 (33:33:00:01:00:02)  
 Internet Protocol Version 6, Src: fe80::264e:9189%89d4, Dst: ff02:11:2  
 User Datagram Protocol, Src Port: 546, Dst Port: 547  
 > DHCPv6

Ethernet: Slave capture in progress>



## HOW TO RETRIEVE WEB PAGES USING HTTP AND WIRESHARK TO CAPTURE THE PACKETS:-

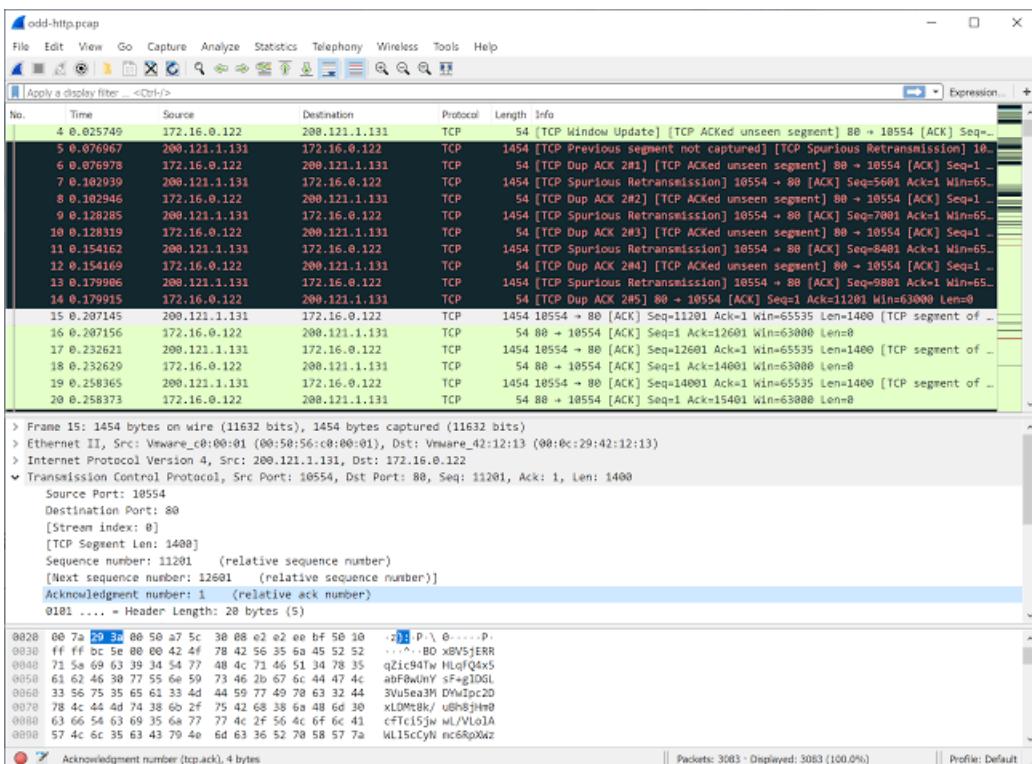
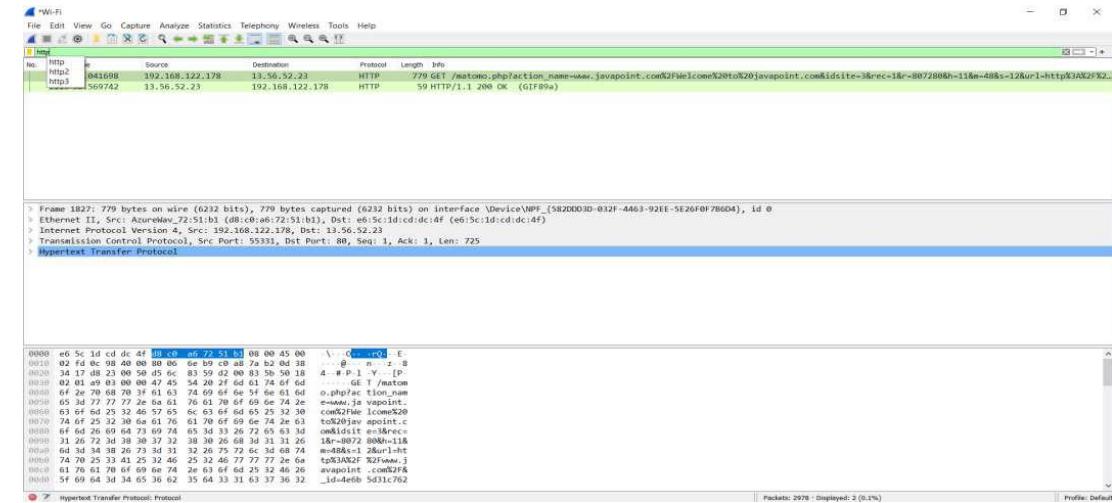
1. Start up your web browser.
2. Start up the Wireshark packet sniffer (but don't yet begin packet capture).
  - Enter “http” (just the letters, not the quotation marks) in the display-filter specification window, so that only captured HTTP messages will be displayed later in the packet-listing window.
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

TCP OR IP.ADDER == 192.168.100.32 .

Your browser should display the webpage.

## 5. Stop Wireshark packet capture.

**Your Wireshark window should look similar to the window shown in below Figure.**



## **EXPERIMENT- 9**

### **VIRTUALBOX**

#### **CREATING A VIRTUAL MACHINE**

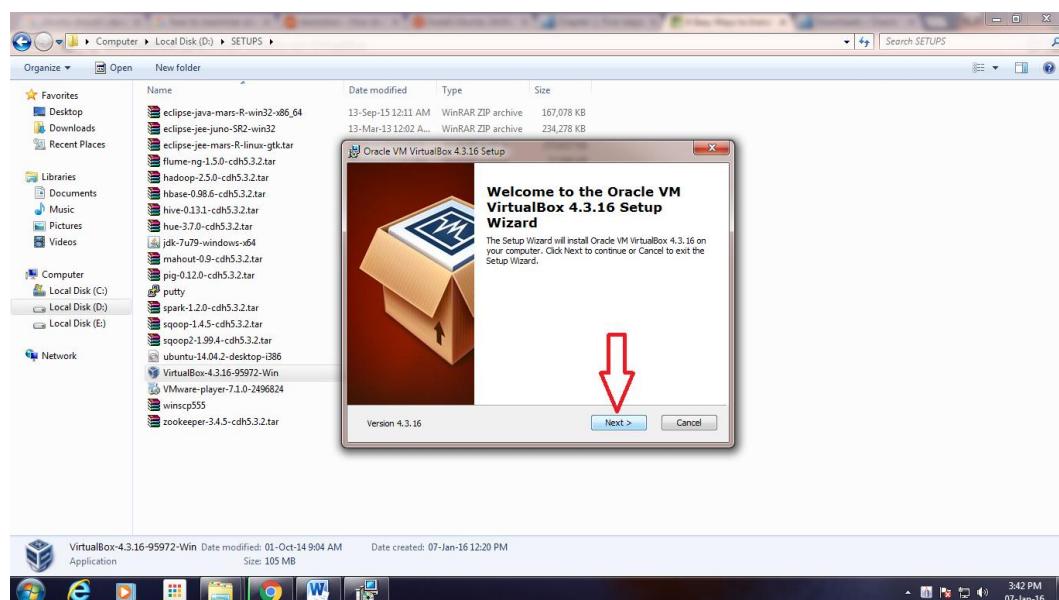
##### **1. INSTALL VIRTUALBOX**

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

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

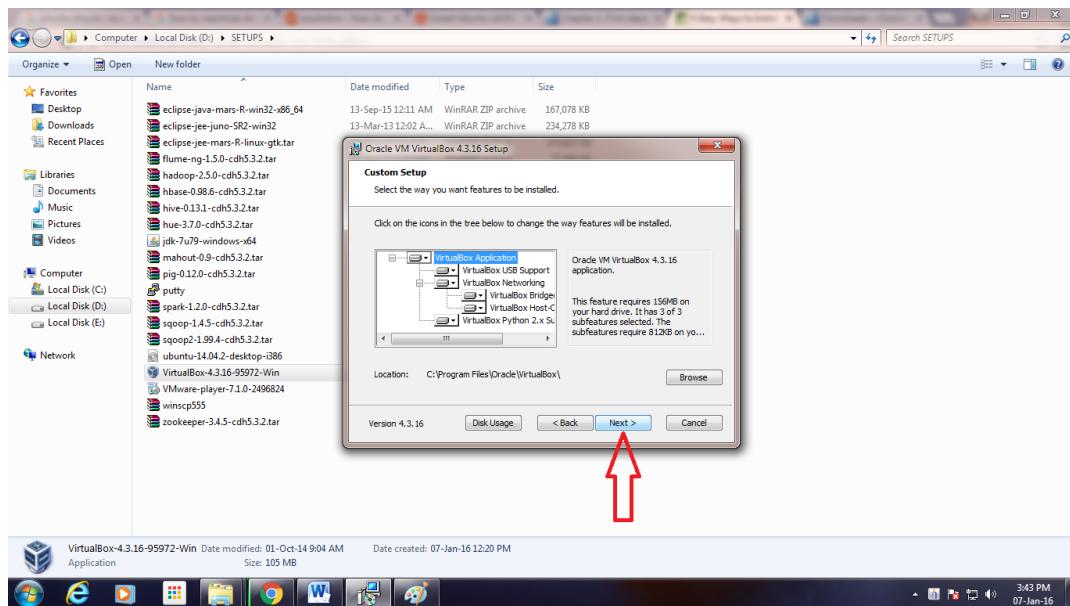
You can download the latest version of Virtual Box from the Virtual Box website: <https://www.virtualbox.org/wiki/Downloads> according to the version of your operating system Windows, Mac or Linux.

##### **1.1. CLICK NEXT**



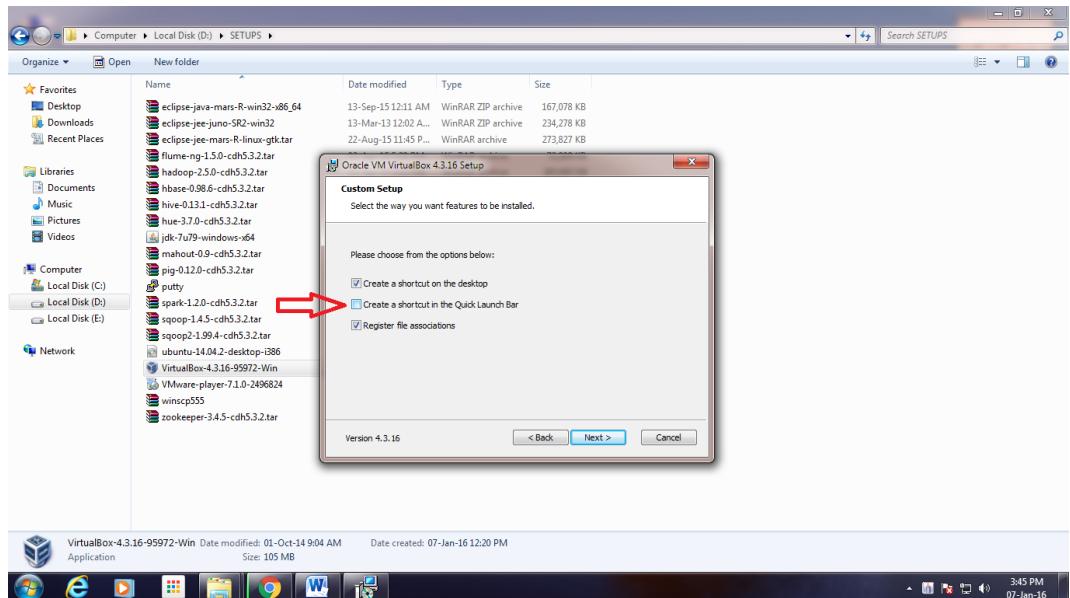
To Install Virtual Box – Setup Wizard

## 1.2. CLICK NEXT



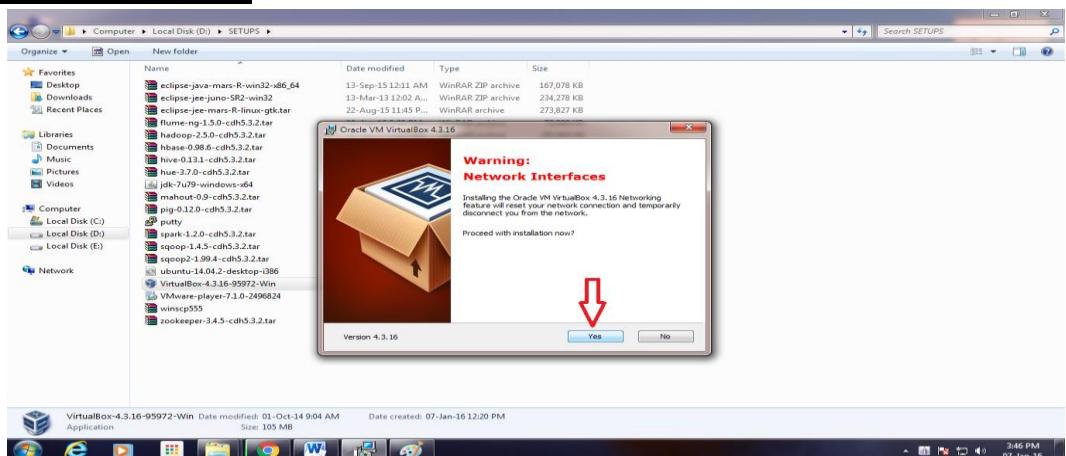
To Install VirtualBox – Custom Setup

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



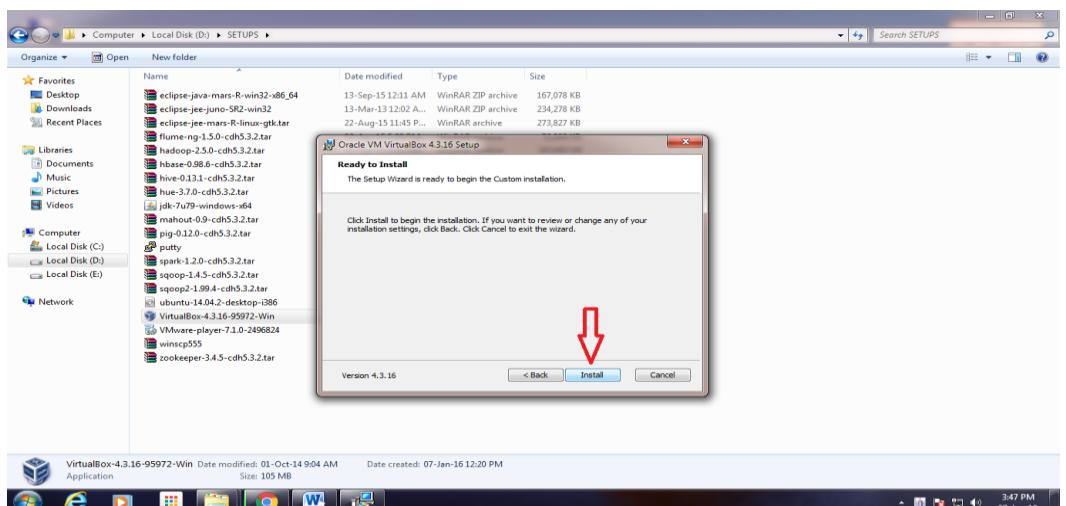
To Install Virtual Box – Features Selection

## **1.4. CLICK “YES”**



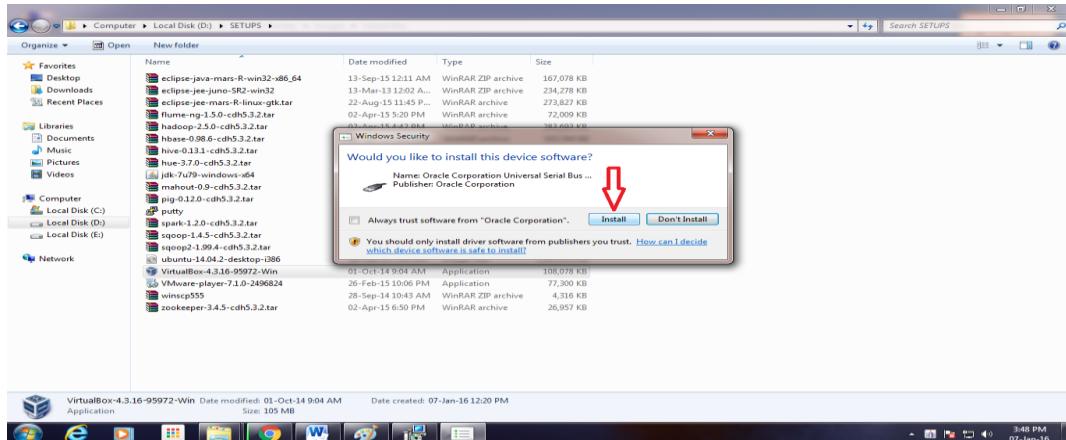
To Install Virtual Box – Network Interfaces Warning

## **1.5. CLICK “INSTALL”**



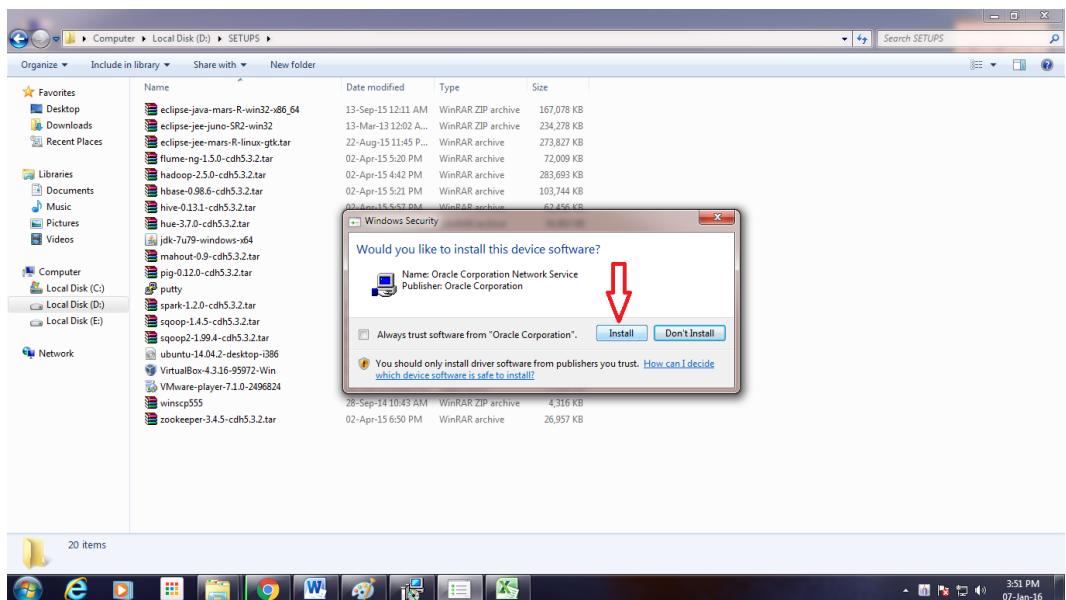
Installation of Oracle VM VirtualBox – Ready to Install

## 1.6. CLICK “INSTALL”



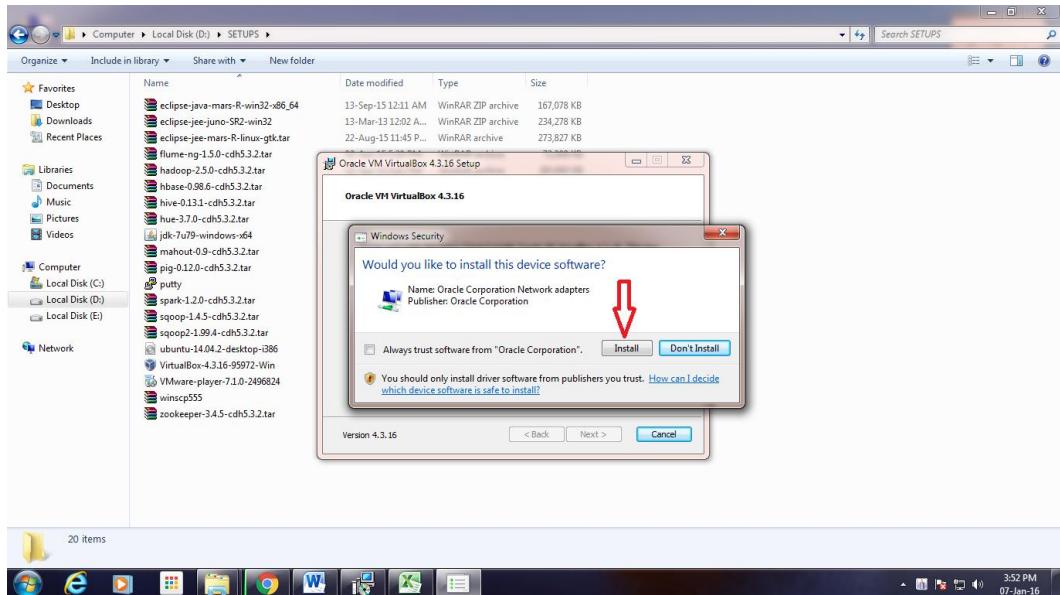
Installation of Oracle VM Virtual Box- Serial Bus Software Installation

## 1.7. CLICK “INSTALL”



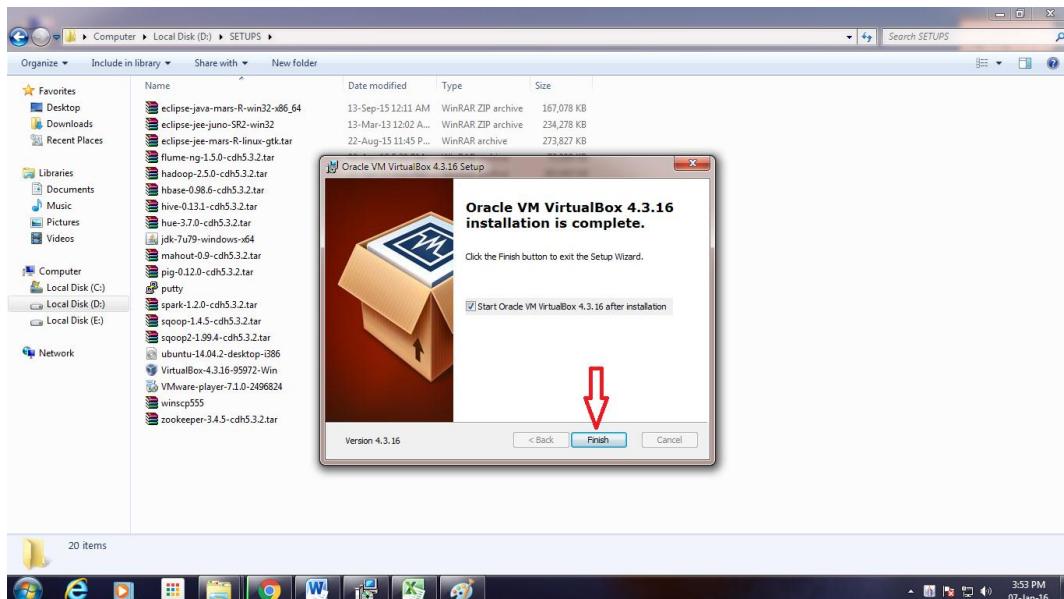
Installation of Oracle VM Virtual Box – Network Service Installation

## **1.8. CLICK “INSTALL”**



Installation of Oracle VM Virtual Box – Network Adapters Installation

## **1.9. CLICK “FINISH”**



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

3.



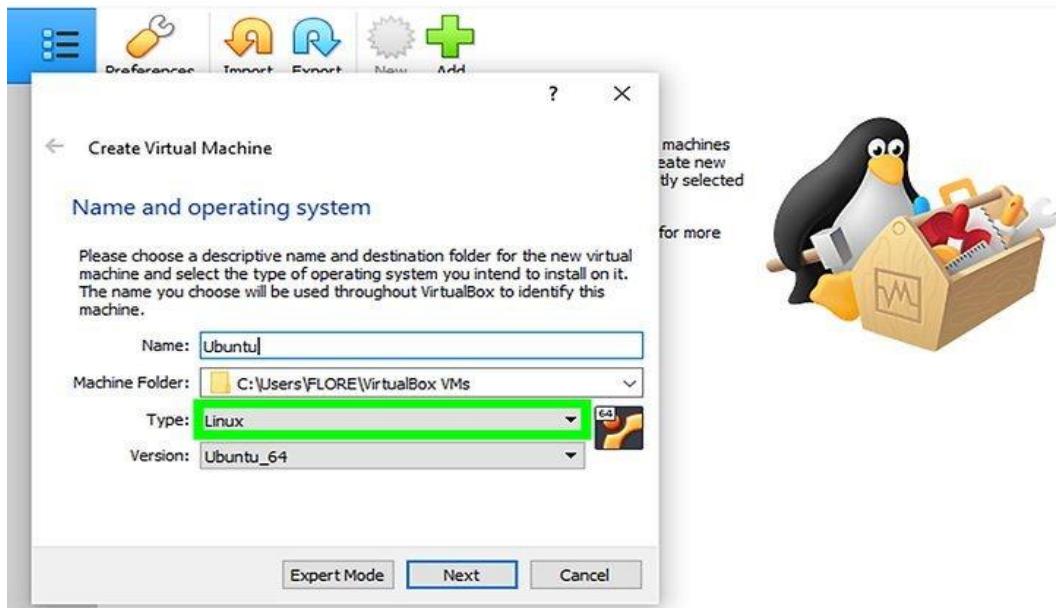
Click **New**. It's a blue badge in the upper-left corner of the Virtual Box window. Doing so opens a pop-up menu.

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.



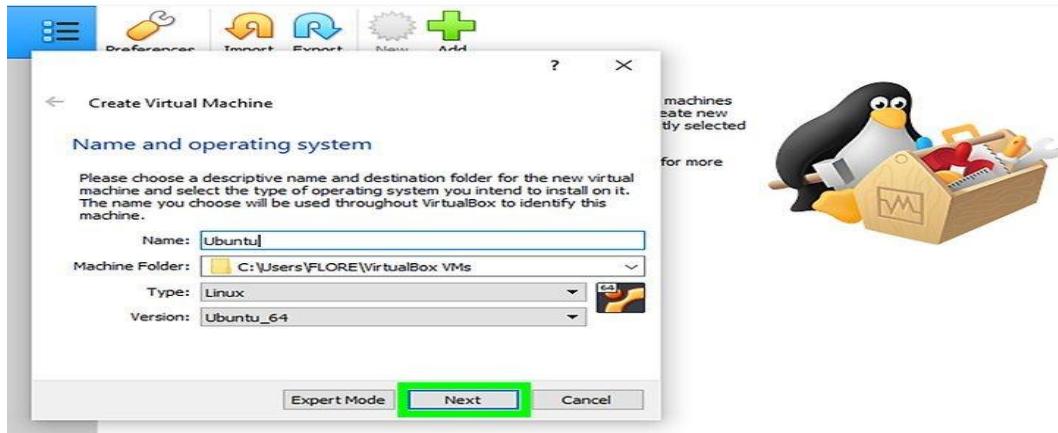
For **Operating System Type**, select the OS that you want to install.

6.



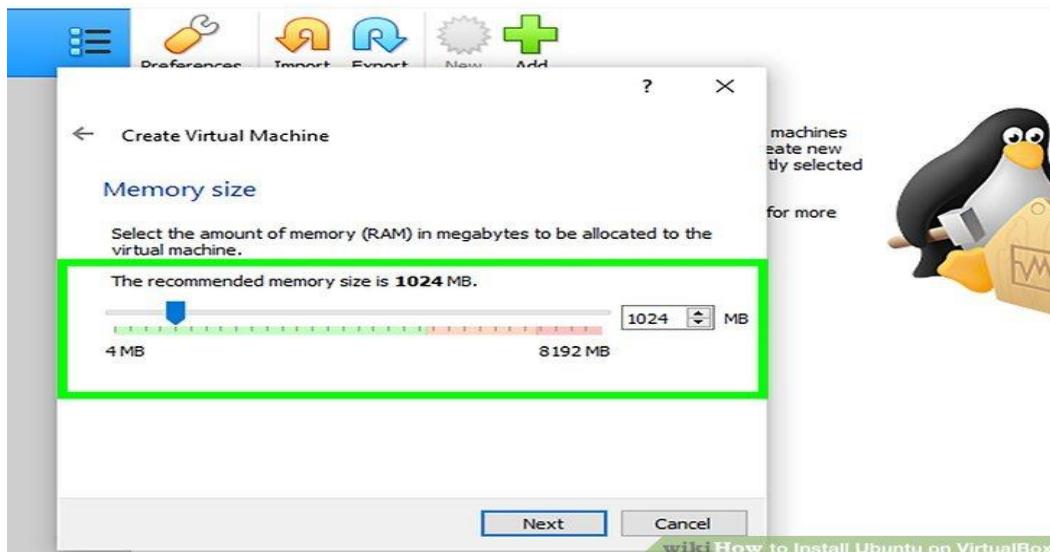
Select the **version** of the operating system.

7.



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

8.

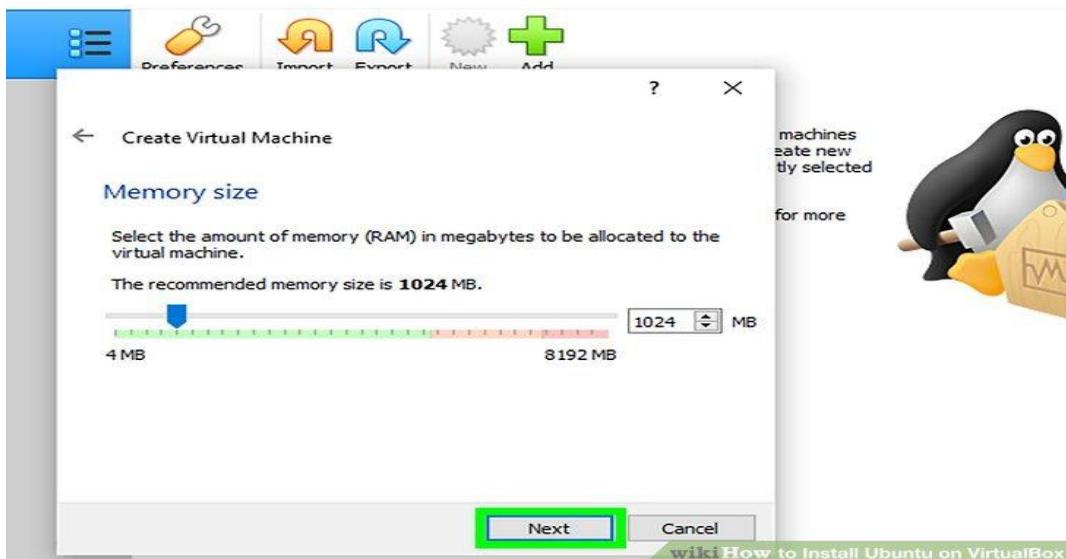


Select an amount of RAM to use. Click and drag the slider left or right to decrease or increase the amount of RAM that VirtualBox will have available for your virtual machine.

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.



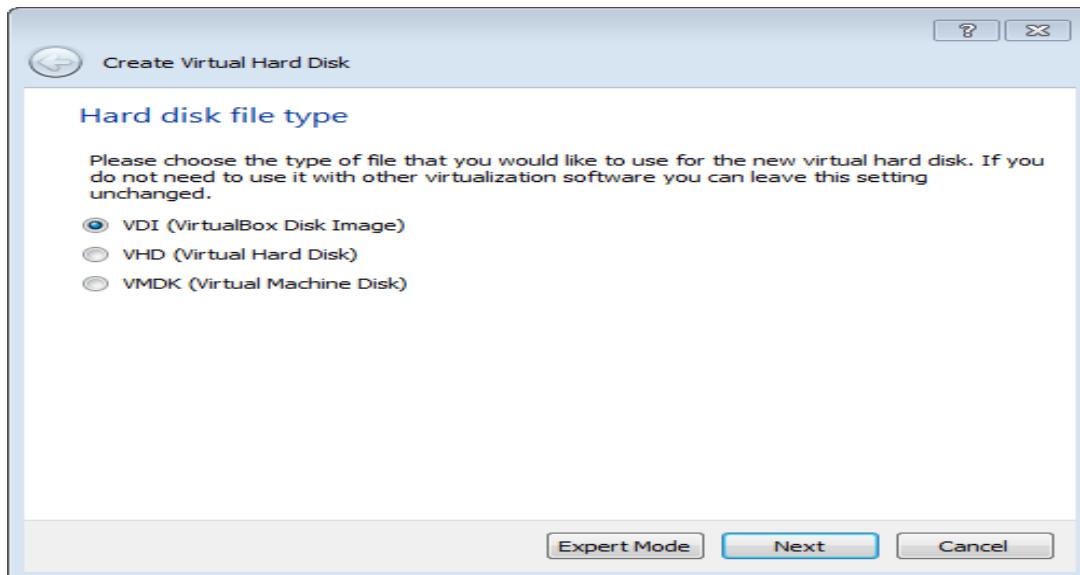
Click **Next**. It's at the bottom of the menu.

10.



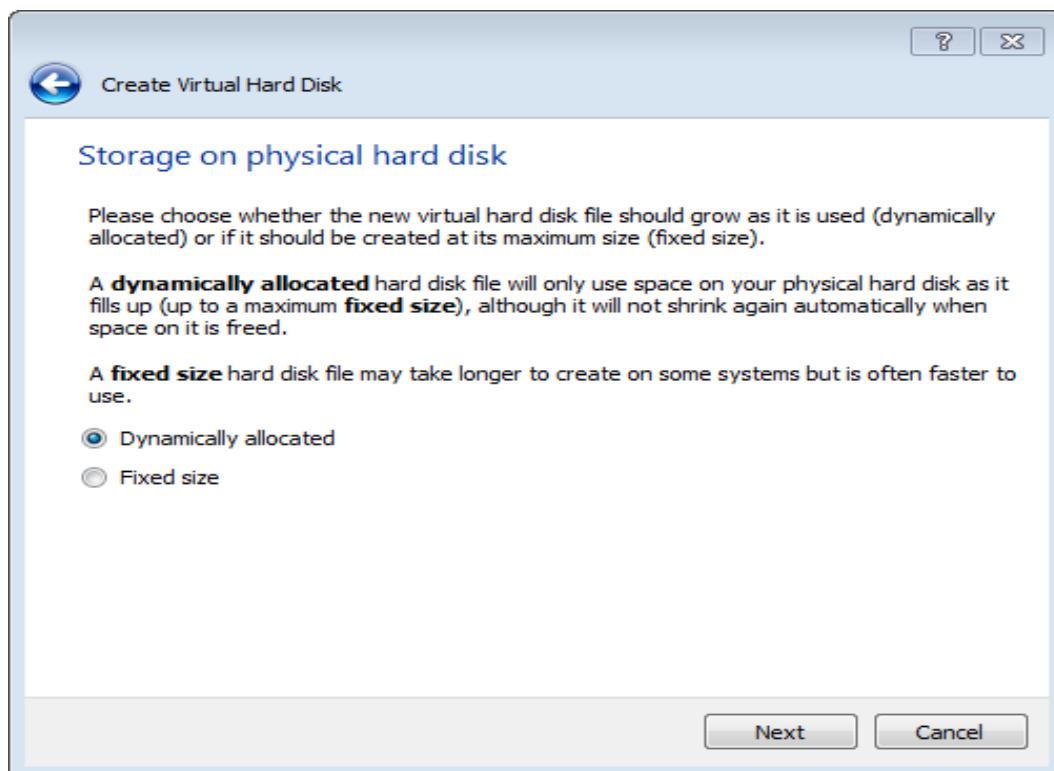
Create your virtual machine's virtual hard drive. The virtual hard drive is a section of your computer's hard drive space which will be used to store your virtual machine's files and programs:

11.



Use “VDI” to create a virtual hard disk

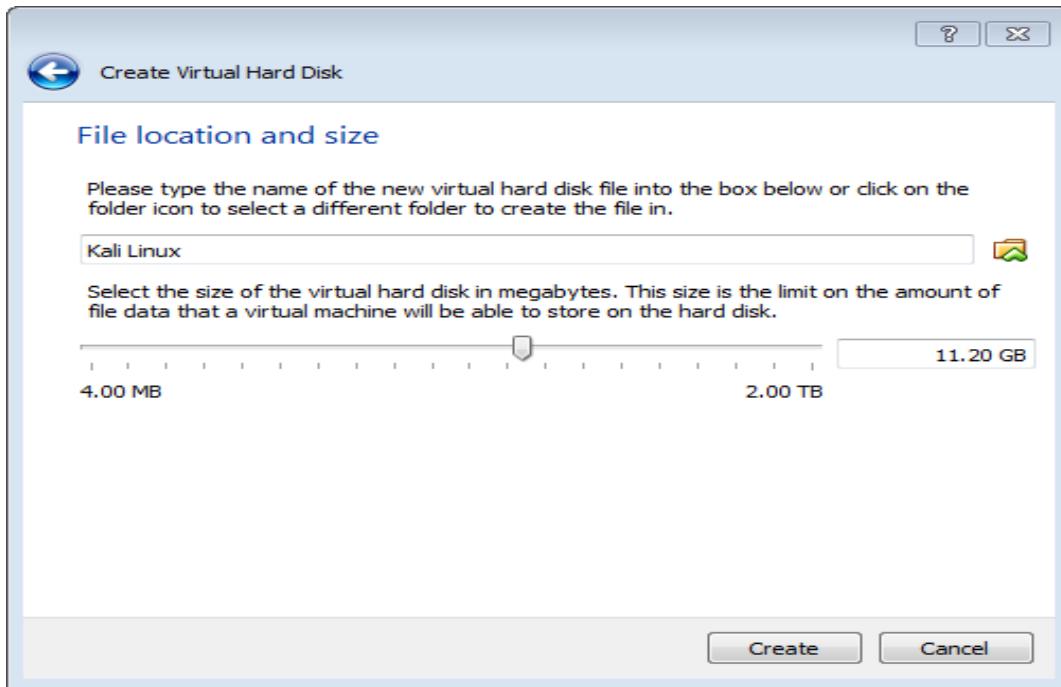
12.



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 Virtual Box 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.