Experiment: 1

Component Identification of a Computer

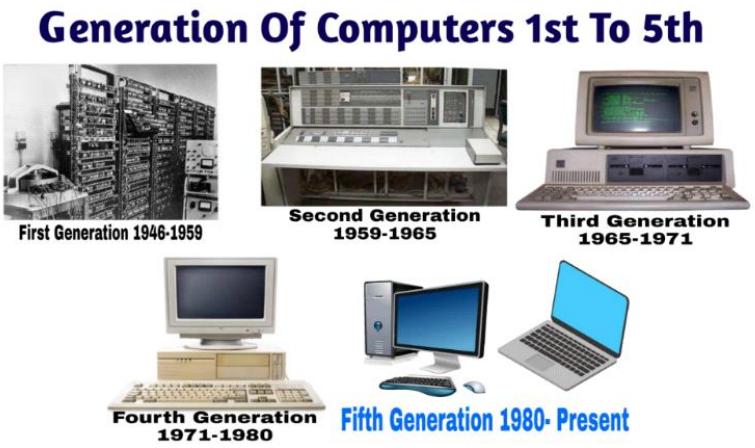
Aim:

To familiarize and identify various parts of a computer.

What is a Computer?

A computer is a machine that accepts data as input, processes that data using programs and outputs the processed data as information.

A computer can be defined as an advanced electronic device that takes raw data as input from the user. It uses a set of instructions (called program) to process the data and give the result (output). The result can be used immediately or saved for future use.



Hardware and Software

In the process of converting data to information, a computer uses hardware and software. At the simplest level, all computers consist of these two basic components; the hardware and the software.

•**Hardware** is any part of the computer that has a physical structure that can be seen and touched, though some may be so tiny that they are invisible to the naked eye. This includes the computer case, monitor, keyboard, and mouse. It also includes all the parts inside the computer case, such as the hard disk drive, motherboard, video card, and many others.

•**Software** is the instruction set that tells the computer how to perform tasks. Software is intangible i.e., that cannot be seen and touched, but its effect is clearly defined.

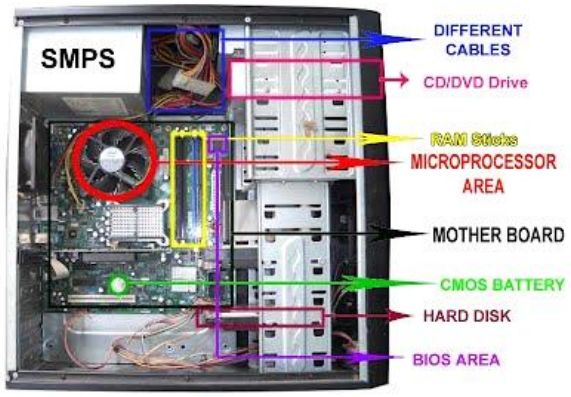
Hardware components are classified into following categories:

- External Devices: Outside the System Unit
- Internal Devices: Inside the System Unit

System Unit

A System Unit is the main component of a personal computer, which houses the other devices necessary for the computer to function. It is comprised of a chassis and the internal components of a personal computer such as the system board (mother board), the microprocessor, memory

modules, disk drives, adapter cards, the power supply, a fan or other cooling device and ports for connecting external components such as monitors, keyboards, mice, and other devices.



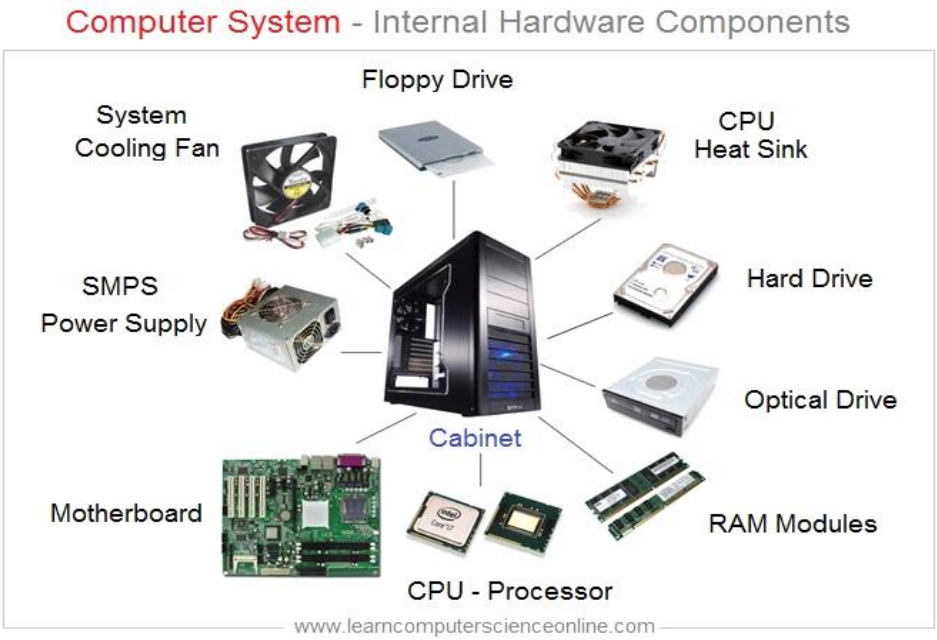
External Devices

Different External devices are discussed below

External Devices		Functionality
	Keyboard	Type data into the system
	Mouse	Point and select objects
	Microphone	Provides audio input
	Scanner	Provides graphical input
	Speaker	Provides audio output

	Printer	Provides printed output
	Monitor	Displays the Information
	External drive	Provides additional data storage
	Pen Drive	External storage device
	Web Camera	A webcam is an input device that captures digital images.
	Joystick	A joystick is an input device that can be used for controlling the movement of the cursor or a pointer in a computer device.

Internal Components/ System Unit Components

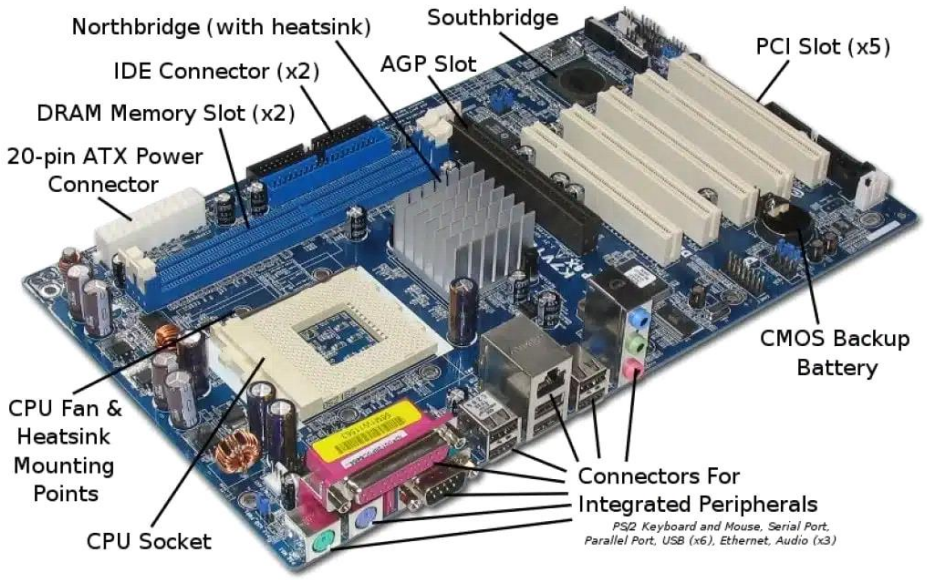


The following are the basic components that will be detailed:-

1. Motherboard
2. CPU
3. RAM
4. Network Card
5. Graphics Card
6. Sound Card
7. Hard Disk
8. Power Supply
9. Cooling System

- **Motherboard**

  The computer motherboard also knows as the printed circuit board is the foundation of a computer that allocates power and allows communication to and between the CPU, RAM, and all other computer hardware components. There are multiple types of motherboards, designed to work with specific types of processors and memory. And almost every major component (such as CPU, Memory, expansion slots and more) that crucial for the functioning of the computer is attached to the motherboard.



Parts of computer motherboard:

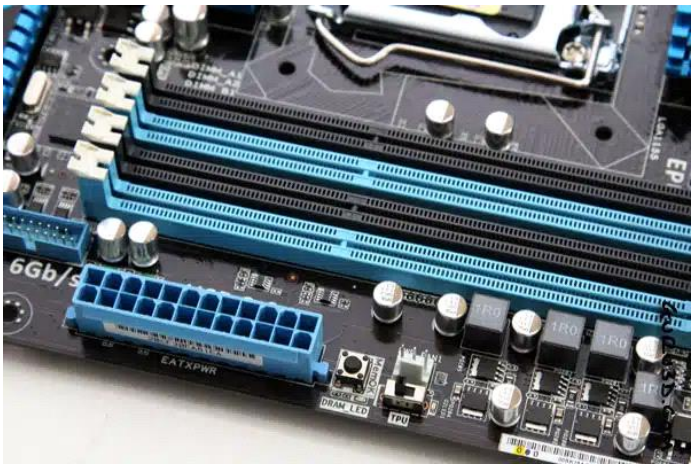
- CPU socket on the motherboard
- Memory Slot on the motherboard
- Basic Input/output System (BIOS)
- The CMOS Battery
- The Computer Cache Memory
- PCI slots – The Expansion Buses
- IDE or SATA
- The Computer Chip-sets
- Input/output Ports on the motherboard

- **CPU socket on the motherboard**

The CPU Central Processing Unit is the most important part of your Computer. It also call the brain of the computer that is responsible for fetching, decoding, and executing program instructions as well as performing mathematical and logical calculations. And the CPU socket is where your CPU (processor) is installed.

- **Memory Slot on the motherboard**

  RAM (Random Access Memory) also called computer memory is another crucial part of the computer. It's volatile that temporarily stores dynamic data to enhance computer performance while you are working and it loses its contents once power is turned off. Well, the memory slots are where we insert the RAM. Most computer motherboards have two to four memory slots, which determine the type of RAM used with the computer. And the most common types of RAM are SDRAM and DDR for desktop computers and SODIMM for laptop computers, each having various types and speeds.

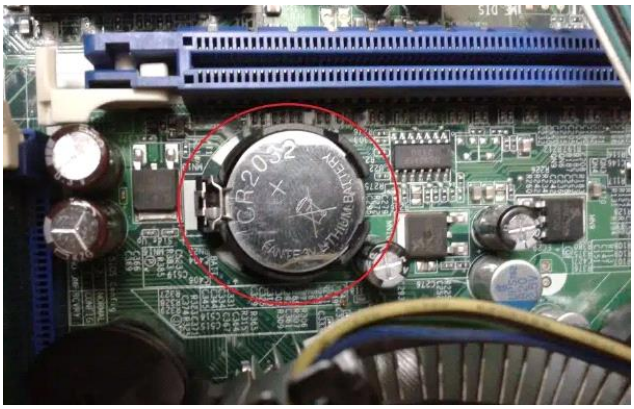


- **Basic Input/Output System (BIOS)**

  BIOS stands for Basic Input/Output System is where all the information and settings for the motherboard are stored. And It can be accessed, updated, and modified via the BIOS mode. BIOS is essentially the link between the computer hardware and software in a system. The BIOS is stored on a ROM chip because ROM retains information even when no power is being supplied to the computer and is used during the startup routine (boot process) to check out the system and prepare to run the hardware.

- **The CMOS Battery**

  Complementary Metal Oxide Semiconductor also known as a CMOS battery is what's responsible for keeping all the information intact when the entire system is shut down. And all motherboards include a small separate block for CMOS which is kept alive by a battery (known as a CMOS battery) even when the PC's power is off. This prevents reconfiguration when the PC is powered on. Again the CMOS battery is removable that can be removed to reset the BIOS after a failed update or if you overclock your RAM beyond its capabilities.



- **The Computer Cache Memory**

  Cache memory is a small block of high-speed memory (RAM) that that acts as a buffer between RAM and the CPU. It holds frequently requested data and instructions so that they are immediately available to the processor on demand.

  Well, Most CPUs have an internal cache memory built into the processor that is referred to as Level 1 or primary cache memory. And this can be supplemented by external cache memory fitted on the motherboard which is the Level 2 or secondary cache.

- **PCI slots – The Expansion Buses**

  PCI stands for Peripheral Component Interconnect and the expansion bus is an input/output pathway from the CPU to peripheral devices. These are the slots that allow inserting expansion cards such as graphics cards, sound cards, LAN card,s or several other functional computer parts. PCI is the most common expansion bus in a PC and other hardware platforms. Buses carry signals such as data, memory addresses, power, and control signals from component to component. Other types of buses include ISA and EISA.

- **IDE or SATA**

  On older computer motherboards, you found Integrated Drive Electronics (IDE) sots. These are the standard interface for connecting a motherboard to storage devices such as hard drives and CD-ROM/DVD drives. But now the latest motherboards make use of SATA technology. A serial advanced technology attachment (serial ATA, SATA or S-ATA) is a computer bus interface used to connect host bus adapters (disk drive controllers) with mass storage devices like optical drives and hard drives.

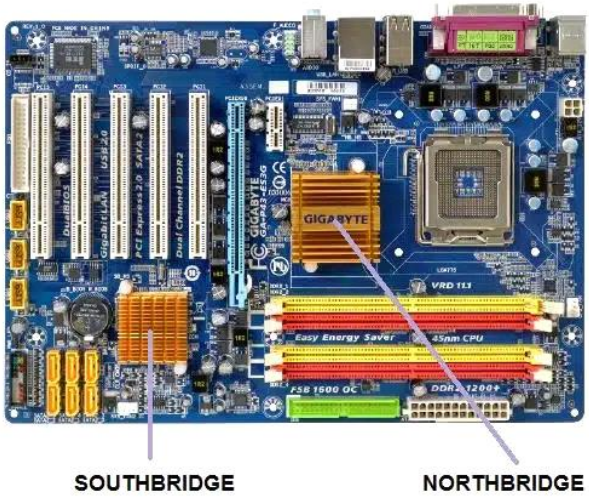
- **The Computer Chip-sets**

  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 chipset 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 NorthBridge (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.

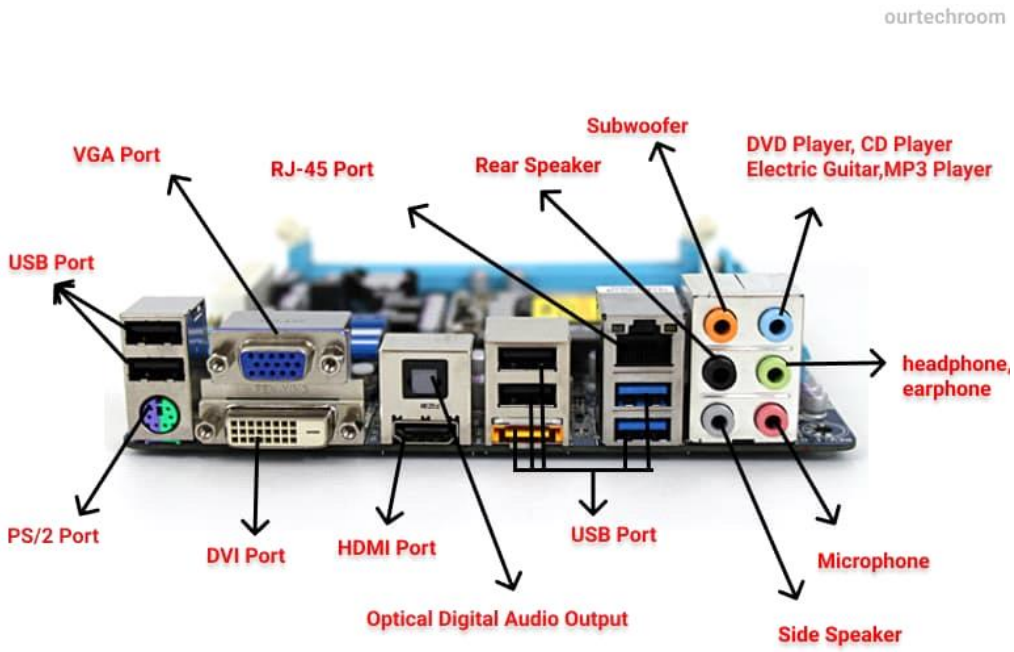
  The SouthBridge (also called the input/output controller or expansion controller) handles communications between slower peripheral devices such as USB, audio, serial, the system BIOS, the ISA bus, the interrupt controller and the IDE channels. It is also called the ICH (I/O Controller Hub). The term "bridge" is generally used to designate a component that connects two buses.



- **Input/Output Ports on the motherboard**

  These ports are located at the back of the computer and are often color-coded.

- Microphone- Pink 3.5mm jack port
- Speakers and Headphones / Headsets / Earbuds- Bold green 3.5mm jack port
- Monitor- Older motherboards are equipped with a solid blue VGA port at the back, but newer motherboards use the HDMI and black or white DVI port as standard
- Ethernet network cable- Colorless port
- Keyboard and Mouse- PS/2 port (Keyboard- purple; Mouse- green)
- USB devices- USB 2.0 colorless port; USB 3.0/3.1 solid blue port (Yes, VGA ports are a similar color, but this only goes to show how outdated VGA is)
- Some modern motherboards feature USB C type connections



- **CPU**

The CPU or central processing unit is basically like the brain of computer systems. It processes all the information on a computational level. It takes all the processes from the RAM and processes them to perform the tasks required by the computer system.

The central processing unit is usually seated in a socket that utilizes a lever or a latch with a hinged plate with a cut-out in the center to secure the CPU onto the motherboard.

It has many copper pads underneath it for the socket contacts to push up against them to make electrical contact. A processor generates a decent amount of heat, especially when it is working under high loads.

It will run even hotter when it is set to a higher clock speed to make it run faster. This is called overclocking.

This is why a heatsink and fan assembly are required to draw the heat away from the central processing unit and distribute it to thin sheets or fins of metal for the fan to cool down.

- **RAM: Random Access Memory**
  RAM is a data storage device that can provide fast read and write access. RAM is also volatile, which means that it loses all the stored data when power is lost. The RAM keeps data ready for the CPU to process. The speed of the RAM is a big contributor to the overall speed of a computer. It plugs directly into a long slot that has contacts on either side of the slot.

RAM is of two types –

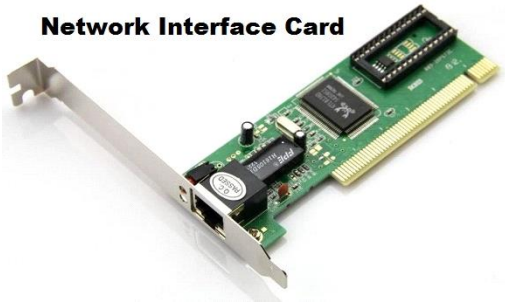
- • Static RAM (SRAM)
- • Dynamic RAM (DRAM)

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

- **Network Card:**

An Ethernet network requires that you install or attach network adapters to each computer or peripheral you want to connect to the network.



- **Graphics Card**

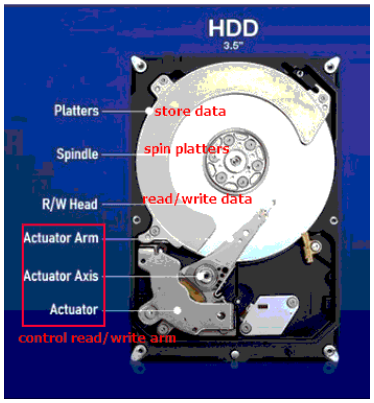
A graphics card processes the data from the motherboard and sends the appropriate information to the monitor for it to be displayed A graphics card can also be referred to as a video card or a display card. It takes the burden of all the video processing from the main CPU. A graphics card plugs into a PCI Express (Peripheral Component Interconnect Express) slot on the motherboard.

- **Sound Card**

Most of the time, the sound chip built into the motherboard is used for audio output.
But, if you are a sound enthusiast or prefer higher detailed audio while playing a game, you might be inclined to use a sound card. A sound processing chip on the card does all of the audio processing and is usually not a very powerful processor. A sound card can offer a wide range of connectivity with various audio equipment.

- **Hard Drive**

Hard disk, also called hard disk drive or hard drive, magnetic storage medium for a computer. Hard disks are flat circular plates made of aluminum or glass and coated with a magnetic material. Hard disks for personal computers can store terabytes (trillions of bytes) of information.



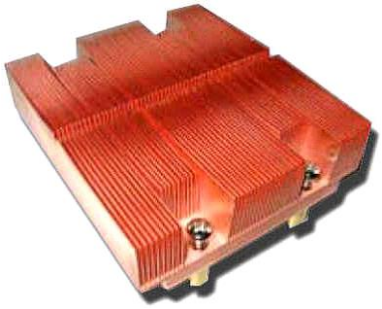
- **Power Supply**

SMPS (Switched-mode power supply) is an electronic power supply system that makes use of a switching regulator to transfer electrical power effectively. It is a PSU (power supply unit) and is usually used in computers to change the voltage to the appropriate range for the computer.

- **Cooling System**

Cooling may be required for CPU, Video Card, Mother Board, Hard Drive, etc. Without proper cooling, the computer hardware may suffers from overheating. This overheating causes slowdowns, system error messages, and crashing. The following are commonly used techniques for cooling the PC or Server components:

Heat Sinks: The purpose of a heat sink is to conduct the heat away from the processor or any other component (such as chipset) to which it is attached.



Intel Dual Core Xeon LGA 771 heat sink

Fan: The Fan is primarily used to force cooler air in to the system or remove hot air out of the system. A fan keeps the surrounding cooler by displacing air around the heat sink and other parts of the computer. A typical CPU fan is shown below.

CPU Fan

Heat sink with Fan

Result:

The different parts of the computer were identified and familiarized.

Experiment: 2
How to Assemble a PC

Aim:

To assemble the Computer and switch it on to work.

General Instructions:

- Familiarize yourself with the components and their specifications: Before you start assembling, make sure you know what components you have and what they are capable of. Read the manuals and familiarize yourself with the specifications of each component.

- Gather all necessary tools: Make sure you have all the tools you need, such as screwdrivers, anti-static wrist strap, thermal paste, SATA cables, power cables, installation media, and any other components that may be required.

- Prepare a static-free environment: Electrostatic discharge (ESD) can damage sensitive electronic components, so make sure you prepare a static-free environment to work in. Wear an anti-static wrist strap and work on a non-conductive surface, such as a wooden table.

- Make sure all the components are compatible with each other, including the motherboard, CPU, memory, storage drives, power supply, and any peripheral devices.

- It's important to research each component thoroughly and to read customer reviews and technical specifications before making a purchase. This will help you select components that are reliable, high-quality, and best suited to your needs.

Tools Required:

- Screwdriver: To install standoffs, mount components, and secure cables.
- Anti-static wrist strap: To protect sensitive components from electrostatic discharge.
- Thermal paste: To ensure proper thermal transfer between the CPU and cooler.
- Pliers or tweezers: To help with cable management and installation of small components.
- SATA cables: To connect storage drives to the motherboard.
- Power cables: To connect the power supply to components.
- Installation media: To install an operating system, such as a CD or USB drive.

Components Required:

Desktop Computers

- Central processing unit (CPU): The heart of the computer, responsible for executing instructions and performing calculations.

- Graphics processing unit (GPU): A specialized processor for handling graphics and visual tasks.

- Motherboard: The main circuit board of the computer, it houses the CPU, GPU, RAM, and other components and provides connections to peripherals.
- Memory (RAM): The computer's working memory, responsible for holding data and instructions for the CPU.
- Storage (hard drive or solid-state drive (SSD)): The device that holds the operating system, applications, and data.
- Power supply unit (PSU): The component that supplies power to the other components.
- Case: The protective housing that holds all the components.
- External Devices like Keyboard, mouse, monitor, speaker, printer etc.

Server-Side Computers:

The specifications of server-side computers can vary depending on the specific needs and requirements of the server. However, some common specifications to consider include:

- Processor (CPU: Key specifications to consider include the number of cores, clock speed, and cache size. Consider choosing a high-performance CPU that can handle the demands of a server.
- Memory (RAM): The amount of RAM required will depend on the intended use, but it is common to have 16 GB or more of RAM in a server.
- Storage: Consider using redundant arrays of independent disks (RAID) to improve the reliability and performance of the storage.
- Networking: The networking components determine the speed and performance of the server's network connections. Consider the number of Ethernet ports, the type of Ethernet ports (e.g., Gigabit Ethernet, 10 Gigabit Ethernet), and the type of network adapter (e.g., PCI, PCI Express).
- Power Supply Unit (PSU): Consider the power consumption of all the components and select a power supply with enough capacity to meet your needs. Consider using redundant power supplies to ensure the server remains operational even if one power supply fails.
- Motherboard: The motherboard is the main circuit board in a server and serves as the connection point for all other components. Consider the number of memory slots, SATA ports, and USB ports available. Consider using server-specific motherboards that are designed to handle the demands of a server.
- Case: Consider the size of the case, the number of drive bays, and the number of expansion slots available. Consider using rack-mountable cases to allow the server to be mounted in a server rack.

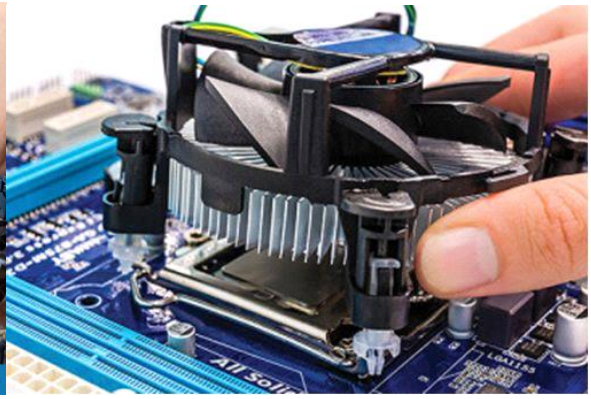
- Operating System: The operating system is the software that runs on the server and determines the functionality of the server. Consider using a server-specific operating system, such as Microsoft Windows Server or Linux, to ensure the server has the necessary features and performance.

Steps to assemble the Computer

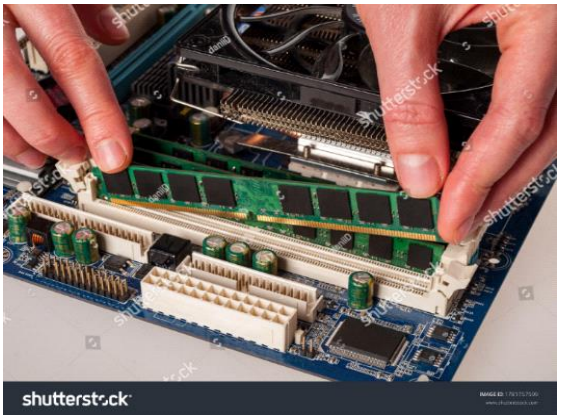
- Prepare the case: Remove any screws or accessories that might be blocking the motherboard standoffs, then install the standoffs into the case.

- Mount the Motherboard:
  - Carefully place the motherboard into the case, aligning the screw holes with the standoffs.
  - Secure the motherboard: Use screws to secure the motherboard to the standoffs. Make sure the screws are tight to prevent the motherboard from wobbling.
  - Connect the power supply cables: Connect the 24-pin ATX main power cable and the 4/8-pin ATX 12V power cable to the motherboard.

- Install the CPU and cooler:
  a. Open the CPU socket by lifting the latch on the socket.
  b. Carefully place the CPU into the socket, making sure the notches on the CPU and socket are aligned.
  c. Close the latch to secure the CPU in place.
  d. Attach the cooler to the top of the CPU, securing it in place with screws or clips.



- Install the RAM:
  a. Locate the RAM slots on the motherboard.
  b. Align the notches on the RAM with the notches on the motherboard.
  c. Firmly press down on the tabs on either side of the slot until the RAM clicks into place.

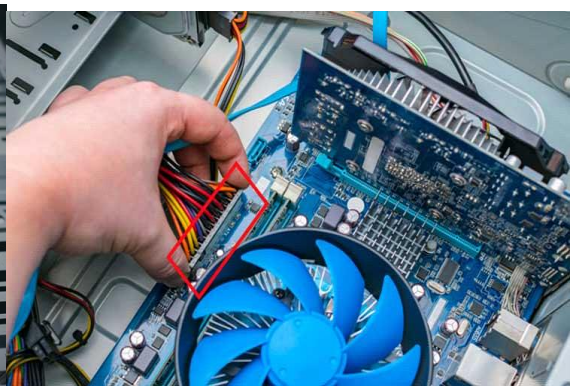


- Install the storage devices:
  a. Mount the storage drive into the case, usually by using screws to secure it in place.
  b. Connect the SATA cable and power cable to the storage drive.
  c. Connect the SATA cable to the SATA port on the motherboard.
  d. Connect the power cable to the storage drive.

- Install the GPU:
  a. Locate the PCIe slot on the motherboard.

b. Carefully align the GPU with the PCIe slot, making sure the notches on the GPU and the slot are aligned.
c. Firmly press down on the tabs on either side of the slot until the GPU clicks into place.
d. Connect any necessary power cables to the GPU.

- Connect the power supply:
  a. Connect the power supply cables to the components, including the motherboard, GPU, storage, and cooler.
  b. Connect the power supply unit to the wall outlet.



- Connect case fans and front panel connectors:
  - Locate the fan headers. Look for the fan headers on the motherboard, usually labeled as "CPU_FAN," "SYS_FAN," or similar.
  - Connect the case fans to the appropriate fan headers on the motherboard, or to the power supply directly.
  - Locate the front panel headers: Look for the front panel headers on the motherboard, usually labeled as "FP_Audio," "PWR_SW," "RESET," "LED," or similar.
  - Connect the power button: Connect the reset button header to the appropriate header on the motherboard, usually labeled as "PWR_SW.".
  - Connect the audio header: If your case has built-in audio ports, connect the audio header to the "FP_Audio" header on the motherboard.
  - Connect the power LED: Connect the power LED header to the appropriate header on the motherboard, usually labeled as "PWR_LED."
  - Connect the reset button: Connect the reset button header to the appropriate header on the motherboard, usually labeled as "RESET."

- Connect peripherals:
  a. Close the cabinet and fix it with screws.
  b. Connect the keyboard, mouse, and monitor to the appropriate ports on the motherboard or GPU.
  c. Use cables, such as HDMI, DisplayPort, or VGA, to connect the monitor to the GPU.
  d. Use USB cables to connect the keyboard and mouse.

Result:

The computer was successfully assembled and switched on.

Experiment -3

Linux Commands Familiarization

Aim:

To familiarize with various Linux Commands.

- Create a directory structure specified files in it with contents.(Use mkdir, cd, ls, cat commands)

  ACN-Notes, Tutorial, Assignments

  Notes- Chapter1.txt, Chapter2.txt

  Tutorials-Tutorial1.txt, Tutorial2.txt

  Assignment-Assignment1.txt, Assignment2.txt

Commands:

$ mkdir ACN

$ cd ACN

$ mkdir Notes Tutorial Assignments

$ cd Notes

$ cat > Chapter1.txt

$ cat > Chapter2.txt

$ cd ..

$ cd Tutorial

$ cat > Tutorial1.txt

$ cat > Tutorial2.txt

$ cd ..

$ cd Assignments

$ cat > Assignment1.txt

$ cat > Assignment2.txt

Output:

```
ubuntu@CCF:~$ cd Soorya
ubuntu@CCF:~/Soorya$ pwd
/home/ubuntu/Soorya
ubuntu@CCF:~/Soorya$ mkdir ACN
ubuntu@CCF:~/Soorya$ cd ACN
ubuntu@CCF:~/Soorya/ACN$ mkdir Notes Tutorial Assignment
ubuntu@CCF:~/Soorya/ACN$ cd Notes
ubuntu@CCF:~/Soorya/ACN/Notes$ cat > chapter1.txt
This is chapter 1
^C
ubuntu@CCF:~/Soorya/ACN/Notes$ cat > chapter2.txt
This is chapter 2.
^C
ubuntu@CCF:~/Soorya/ACN/Notes$ cd ..
ubuntu@CCF:~/Soorya/ACN$ cd Tutorial
ubuntu@CCF:~/Soorya/ACN/Tutorial$ cat > tutorial1.txt
This is tutorial 1
^C
ubuntu@CCF:~/Soorya/ACN/Tutorial$ cat > tutorial2.txt
This is tutorial 2.
^C
ubuntu@CCF:~/Soorya/ACN/Tutorial$ cd ..
ubuntu@CCF:~/Soorya/ACN$ cd Assignment
ubuntu@CCF:~/Soorya/ACN/Assignment$ cat > assignment1.txt
This is assignment 1.
^C
ubuntu@CCF:~/Soorya/ACN/Assignment$ cat > assignment2.txt
This is assignment 2.
^C
ubuntu@CCF:~/Soorya/ACN/Assignment$ cd ..
ubuntu@CCF:~/Soorya/ACN$
```

2. Display the present working directory. Create a new directory, Delete it and show it using the ls commands

Commands:

$ pwd

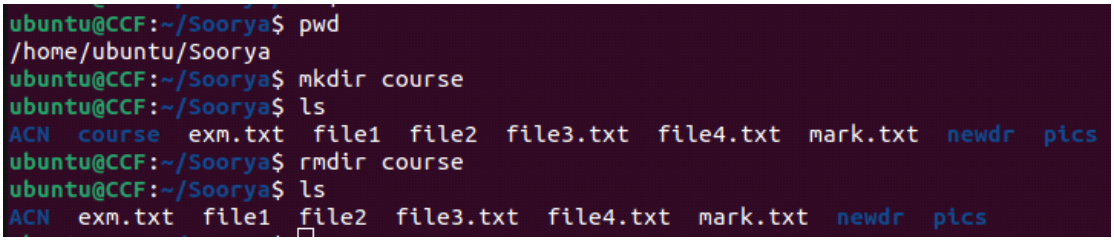
$ mkdir course

$ ls

$ rmdir course

$ ls

Output:



3. Start from the home directory. Move to the root directory, move back to home directory, move into a sub directory, move to the parent directory.

Commands:

$ cd ~

$ cd /

$ cd ~

$ mkdir book

$ cd book

$ mkdir pages

$ cd pages

$ cd ..

Output:

```
ubuntu@CCF:~/Soorya$ cd ~
ubuntu@CCF:~$ pwd
/home/ubuntu
ubuntu@CCF:~$ cd /
ubuntu@CCF:/$ pwd
/
ubuntu@CCF:/$ cd ~
ubuntu@CCF:~$ pwd
/home/ubuntu
ubuntu@CCF:~$ mkdir book
ubuntu@CCF:~$ cd book
ubuntu@CCF:~/book$ mkdir pages
ubuntu@CCF:~/book$ cd pages
ubuntu@CCF:~/book/pages$ cd ..
ubuntu@CCF:~/book$ pwd
/home/ubuntu/book
```

4. Go to the home directory. Copy a file from the home directory to a subfolder in that directory. Copy another file from the subfolder to the home folder. (Don't change directory)

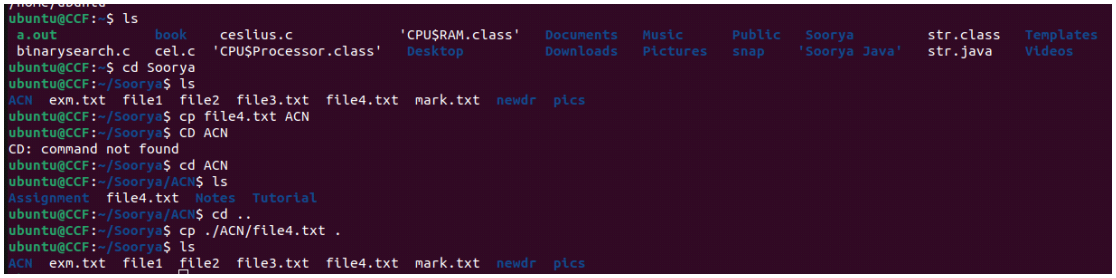
Commands:

$ cd ~

$ cp file4.txt ACN

.

Output:



5. Make three files with extension .txt in a folder (Use touch command). Delete one file from the folder. Change directory to parent directory and delete other two files. (Use ls command to demonstrate)

Commands:

$ cd Soorya

$ pwd

$ touch newfile1.txt newfile2.txt newfile3.txt

$ ls

$ rm newfile1.txt

$ ls

$ cd -

$ rm ./Soorya/*.txt

$ cd Soorya

$ ls Soorya

Output:

6. Go to the home directory. Move a file from the home directory to a subfolder in that directory. Move another file from the subfolder to the home folder. (Don't change directory)

Commands:

$ cd Soorya

$ mv file1.txt ACN

$ ls CAN

$ cd -

$ mv ./ACN/file1.txt

$ ls Soorya

Output:

```
ubuntu@CCF:~$ cd Soorya
ubuntu@CCF:~/Soorya$ mv file1.txt ACN
ubuntu@CCF:~/Soorya$ cd ACN
ubuntu@CCF:~/Soorya/ACN$ ls
Assignment  file1.txt  file4.txt  Notes  Tutorial
ubuntu@CCF:~/Soorya/ACN$ cd -
/home/ubuntu/Soorya
ubuntu@CCF:~/Soorya$ mv ./ACN/file1.txt .
ubuntu@CCF:~/Soorya$ ls
ACN  file1  file1.txt  file2  newdr  pics
ubuntu@CCF:~/Soorya$
```

7. Create a file using cat command, and contents must be the output of man ls command. Display the first 10 lines and last 10 lines of a new file.

Commands:

$ man ls | cat > file.txt

$ head file.txt

$ tail file.txt

Output:

8. Display the calendar of any year page by page using more command.

Commands:

$ cal 2000 |more

Output:

```
ubuntu@CCF:~/Soorya$ cal 2000| more
                                2000
      January                   February                  March
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
                   1          1  2  3  4  5              1  2  3  4
 2  3  4  5  6  7  8   6  7  8  9 10 11 12   5  6  7  8  9 10 11
 9 10 11 12 13 14 15  13 14 15 16 17 18 19  12 13 14 15 16 17 18
16 17 18 19 20 21 22  20 21 22 23 24 25 26  19 20 21 22 23 24 25
23 24 25 26 27 28 29  27 28 29              26 27 28 29 30 31
30 31

       April                     May                     June
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
                   1      1  2  3  4  5  6              1  2  3
 2  3  4  5  6  7  8   7  8  9 10 11 12 13   4  5  6  7  8  9 10
 9 10 11 12 13 14 15  14 15 16 17 18 19 20  11 12 13 14 15 16 17
16 17 18 19 20 21 22  21 22 23 24 25 26 27  18 19 20 21 22 23 24
23 24 25 26 27 28 29  28 29 30 31           25 26 27 28 29 30
30

        July                    August                 September
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
                   1          1  2  3  4  5                 1  2
 2  3  4  5  6  7  8   6  7  8  9 10 11 12   3  4  5  6  7  8  9
 9 10 11 12 13 14 15  13 14 15 16 17 18 19  10 11 12 13 14 15 16
16 17 18 19 20 21 22  20 21 22 23 24 25 26  17 18 19 20 21 22 23
23 24 25 26 27 28 29  27 28 29 30 31        24 25 26 27 28 29 30
30 31

      October                  November                December
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7              1  2  3  4                 1  2
 8  9 10 11 12 13 14   5  6  7  8  9 10 11   3  4  5  6  7  8  9
15 16 17 18 19 20 21  12 13 14 15 16 17 18  10 11 12 13 14 15 16
22 23 24 25 26 27 28  19 20 21 22 23 24 25  17 18 19 20 21 22 23
29 30 31              26 27 28 29 30        24 25 26 27 28 29 30
                                            31
```

9. Create three files with contents using cat command. Display the content of a file. Concatenate the three file's content to new file using cat command.

Commands:

$ cat > file1.txt

$ cat > file2.txt

$ cat > file3.txt

$ cat file1

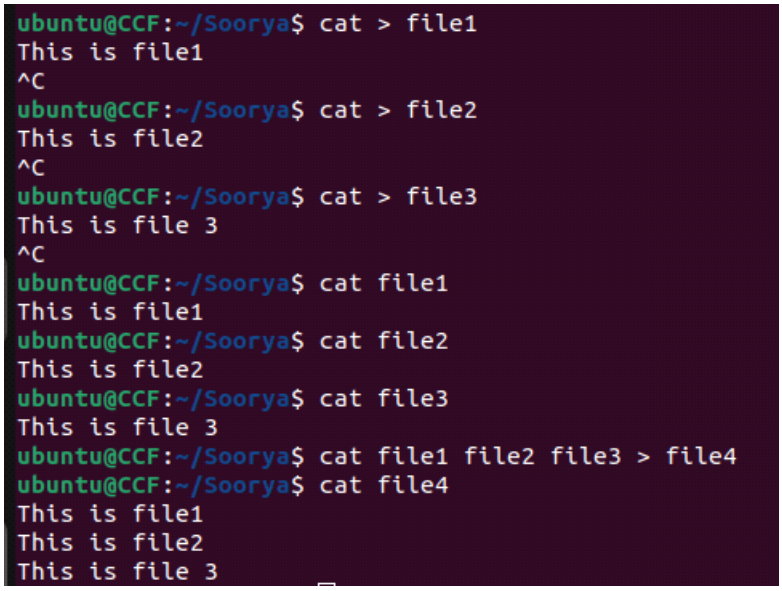
$ cat file2

$ cat file3

$ cat file1 file2 file3 > file4

$ cat file4

Output:



10. Create a file with contents. Use wc command to display the number of characters, words and lines in that file.

Commands:

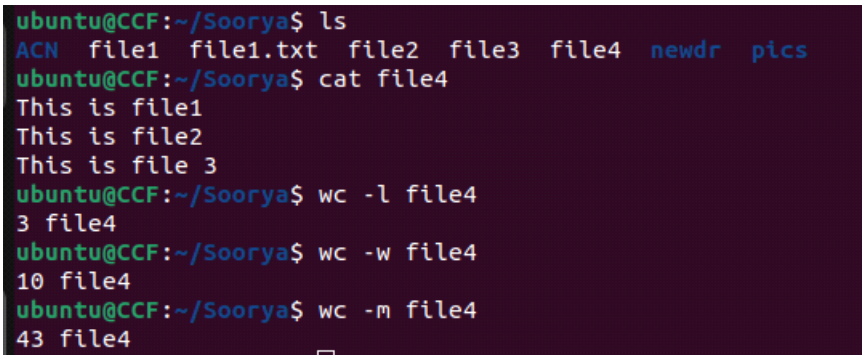
$ cat > file4.txt

$ wc -l file4

$ wc -w file4

$ wc -m file4

Output:

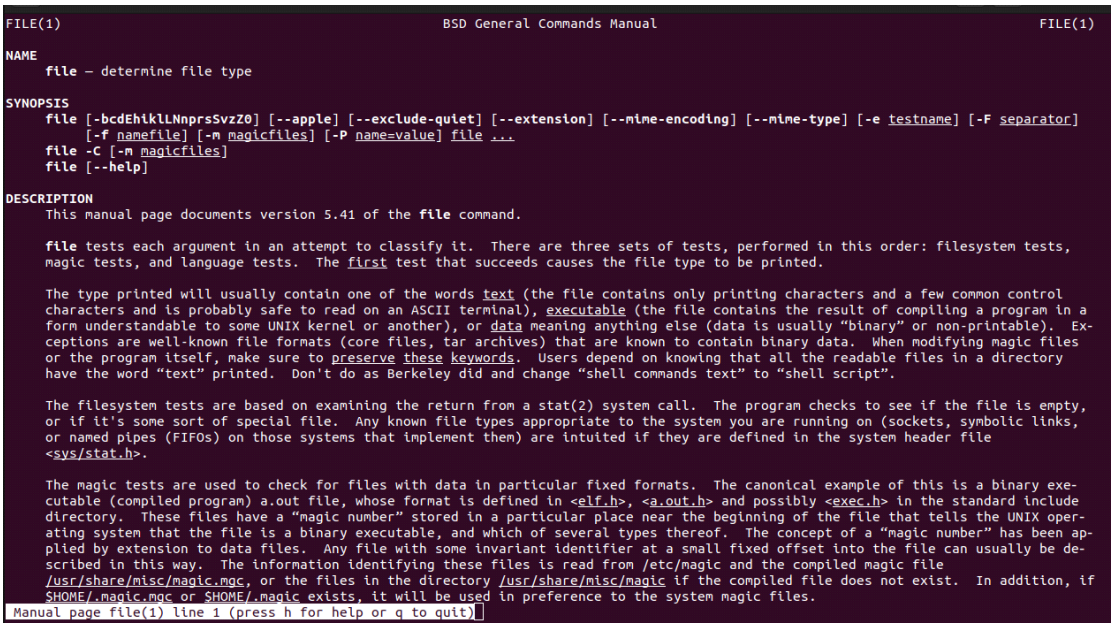


11. Use man command to get the syntax of file command.

Commands:

$ man file

Output:



12. List the lines of the file which contains a string using grep command.

Commands:

$ pwd

$ cat > marks

$ grep string marks

$ grep Shreyas marks

Output:

```
ubuntu@ccf67:~/soorya$ pwd
/home/ubuntu/soorya
ubuntu@ccf67:~/soorya$ cat > marks
Zain 50
Rhoyif 25
Shreyas 10
^C
ubuntu@ccf67:~/soorya$ cat marks
Zain 50
Rhoyif 25
Shreyas 10
ubuntu@ccf67:~/soorya$ grep string marks
ubuntu@ccf67:~/soorya$ grep Shreyas marks
Shreyas 10
```

13. Write a listing of the files in your directory into a file called filelist using output redirection.

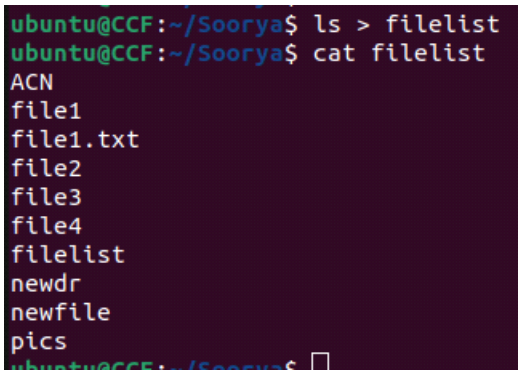
Commands:

To create a new file

$ ls > filelist

$ cat filelist

Output:



14. Create three files Number, State and Capital with proper contents. Display the contents of all three files delimited with a delimiter using paste command.

Commands:

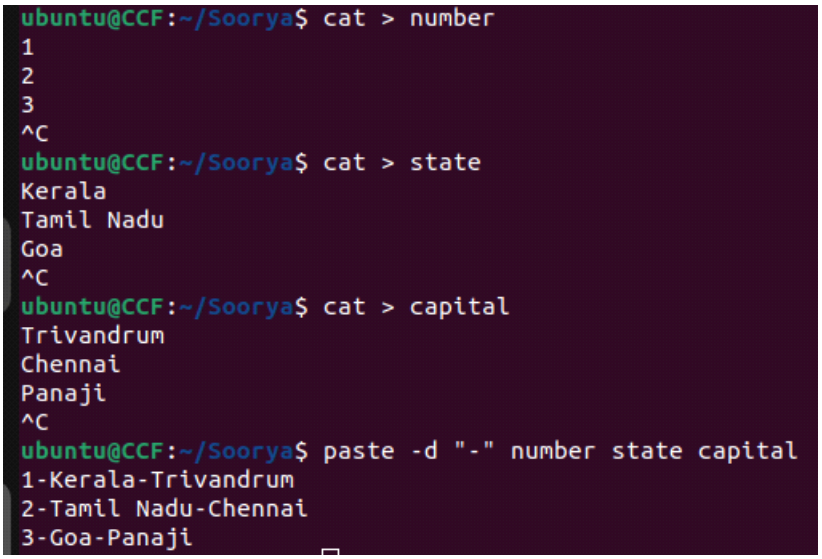
$ cat > Number

$ cat > state

$ cat > capital

$ paste -d "-" number state capital

Output:



15. Change the permission of the file as only read permission to all.

Commands:

$ cd Soorya

$ ls

$ ls -l marks

$ chmod a-wx marks

$ ls -l marks

Output:

```
ubuntu@ccf67:~$ pwd
/home/ubuntu
ubuntu@ccf67:~$ cd soorya
ubuntu@ccf67:~/soorya$ pwd
/home/ubuntu/soorya
ubuntu@ccf67:~/soorya$ ls
aaaaa.txt  aaa.txt  ab.txt  a.txt  file.txt  h.java  marks
ubuntu@ccf67:~/soorya$ ls -l marks
-rw-rw-r-- 1 ubuntu ubuntu 29 Apr 13 15:45 marks
ubuntu@ccf67:~/soorya$ chmod -wx marks
ubuntu@ccf67:~/soorya$ ls -l marks
-r--r--r-- 1 ubuntu ubuntu 29 Apr 13 15:45 marks
ubuntu@ccf67:~/soorya$
```

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

Commands:

$ cd Soorya

$ ps –ef

Output:

```
ubuntu@ccf67:~/soorya$ ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 15:38 ?        00:00:01 /sbin/init splash
root           2       0  0 15:38 ?        00:00:00 [kthreadd]
root           3       2  0 15:38 ?        00:00:00 [rcu_gp]
root           4       2  0 15:38 ?        00:00:00 [rcu_par_gp]
root           5       2  0 15:38 ?        00:00:00 [slub_flushwq]
root           6       2  0 15:38 ?        00:00:00 [netns]
root           8       2  0 15:38 ?        00:00:00 [kworker/0:0H-events_highpri
root          10       2  0 15:38 ?        00:00:00 [mm_percpu_wq]
root          11       2  0 15:38 ?        00:00:00 [rcu_tasks_rude_]
root          12       2  0 15:38 ?        00:00:00 [rcu_tasks_trace]
root          13       2  0 15:38 ?        00:00:00 [ksoftirqd/0]
root          14       2  0 15:38 ?        00:00:00 [rcu_sched]
root          15       2  0 15:38 ?        00:00:00 [migration/0]
root          16       2  0 15:38 ?        00:00:00 [idle_inject/0]
root          18       2  0 15:38 ?        00:00:00 [cpuhp/0]
root          19       2  0 15:38 ?        00:00:00 [cpuhp/1]
root          20       2  0 15:38 ?        00:00:00 [idle_inject/1]
root          21       2  0 15:38 ?        00:00:00 [migration/1]
root          22       2  0 15:38 ?        00:00:00 [ksoftirqd/1]
root          24       2  0 15:38 ?        00:00:00 [kworker/1:0H-kblockd]
root          25       2  0 15:38 ?        00:00:00 [cpuhp/2]
root          26       2  0 15:38 ?        00:00:00 [idle_inject/2]
root          27       2  0 15:38 ?        00:00:00 [migration/2]
root          28       2  0 15:38 ?        00:00:00 [ksoftirqd/2]
root          30       2  0 15:38 ?        00:00:00 [kworker/2:0H-events_highpri
root          31       2  0 15:38 ?        00:00:00 [cpuhp/3]
root          32       2  0 15:38 ?        00:00:00 [idle_inject/3]
root          33       2  0 15:38 ?        00:00:00 [migration/3]
root          34       2  0 15:38 ?        00:00:00 [ksoftirqd/3]
root          36       2  0 15:38 ?        00:00:00 [kworker/3:0H-events_highpri
root          37       2  0 15:38 ?        00:00:00 [kdevtmpfs]
root          38       2  0 15:38 ?        00:00:00 [inet_frag_wq]
root          39       2  0 15:38 ?        00:00:00 [kauditd]
root          40       2  0 15:38 ?        00:00:00 [khungtaskd]
root          41       2  0 15:38 ?        00:00:00 [oom_reaper]
root          42       2  0 15:38 ?        00:00:00 [writeback]
root          43       2  0 15:38 ?        00:00:00 [kcompactd0]
root          44       2  0 15:38 ?        00:00:00 [ksmd]
root          45       2  0 15:38 ?        00:00:00 [khugepaged]
root          92       2  0 15:38 ?        00:00:00 [kintegrityd]
root          93       2  0 15:38 ?        00:00:00 [kblockd]
root          94       2  0 15:38 ?        00:00:00 [blkcg_punt_bio]
root          95       2  0 15:38 ?        00:00:00 [tpm_dev_wq]
root          96       2  0 15:38 ?        00:00:00 [ata_sff]
```

17. List the disk partitions in your hard disk.

Commands:

$ cd Soorya

$ lsblk

Output:

```
ubuntu@ccf67:~/soorya$ lsblk
NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
loop0    7:0    0     4K  1 loop /snap/bare/5
loop1    7:1    0  55.6M  1 loop /snap/core18/2721
loop2    7:2    0  55.6M  1 loop /snap/core18/2714
loop3    7:3    0  63.3M  1 loop /snap/core20/1828
loop4    7:4    0  63.3M  1 loop /snap/core20/1852
loop5    7:5    0 314.6M  1 loop /snap/eclipse/66
loop6    7:6    0 218.4M  1 loop /snap/gnome-3-34-1804/90
loop7    7:7    0   219M  1 loop /snap/gnome-3-34-1804/77
loop8    7:8    0 346.3M  1 loop /snap/gnome-3-38-2004/119
loop9    7:9    0 349.7M  1 loop /snap/gnome-3-38-2004/137
loop10   7:10   0  91.7M  1 loop /snap/gtk-common-themes/1535
loop11   7:11   0  81.3M  1 loop /snap/gtk-common-themes/1534
loop12   7:12   0  45.9M  1 loop /snap/snap-store/599
loop13   7:13   0    46M  1 loop /snap/snap-store/638
loop14   7:14   0  49.9M  1 loop /snap/snapd/18357
loop15   7:15   0  49.9M  1 loop /snap/snapd/18596
sda      8:0    0 232.9G  0 disk
├─sda1   8:1    0    50M  0 part
├─sda2   8:2    0   120G  0 part
├─sda3   8:3    0   506M  0 part
├─sda4   8:4    0     1K  0 part
├─sda5   8:5    0  50.3G  0 part
├─sda6   8:6    0   1.9G  0 part [SWAP]
└─sda7   8:7    0  59.7G  0 part /
```

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

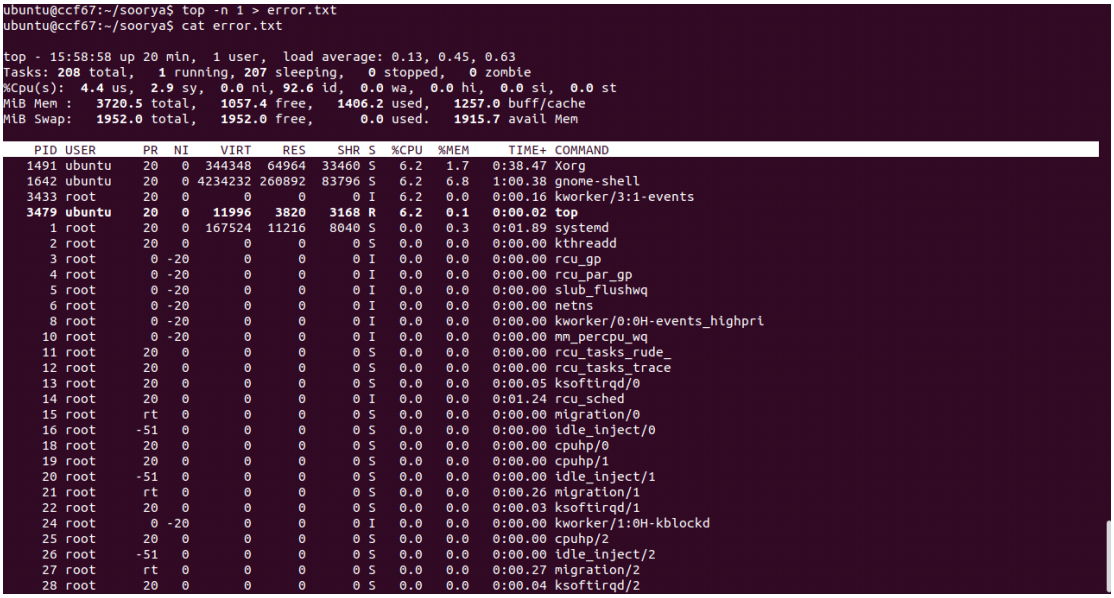
Commands:

$ cd Soorya

$ top -n 1 > error.txt

$ cat error.txt

Output:



Result:

The Linux Commands were successfully executed and received the output.

Experiment 4

Shell Scripts

Aim:

To learn how to write and execute shell scripts from command prompt using vi/vim editor.

1.Write a Shell program to display a given message “Hello, Welcome to Shell Scripting”.

Algorithm:

STEP 1: Start

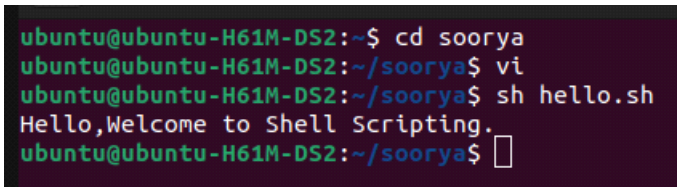
STEP 2: Print “Hello Welcome to shell Scripting”

STEP 3: Stop

Shell Script:

```
echo "Hello welcome to Shell Scripting"
```

Output:



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

Algorithm:

STEP 1: Start

STEP 2: Display message in single quotes

STEP 3: Display message in double quotes

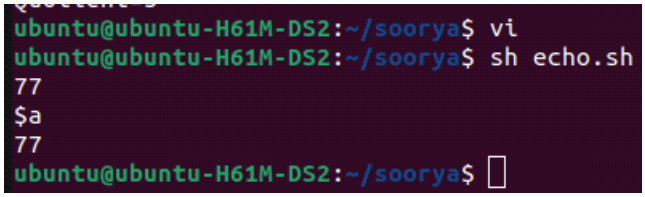
STEP 4: Display message without quotes

STEP 5: Stop

Shell Script:

```
a=77
echo "$a"
echo '$a'
echo $a
```

Output:



3.Write a shell script to evaluate arithmetic operations.

Algorithm:

STEP 1: Start

STEP 2: Read two variables from user

STEP 3: Compute addition

STEP 4: Compute subtraction

STEP 5: Compute Multiplication

STEP 6: Compute Division

STEP 7: Print the results

STEP 8: Stop

Shell Script:

```
a=10
b=2
c=` echo $a + $b | bc `
d=` echo $a - $b | bc `
e=` echo $a \* $b |bc `
f=` echo $a /$b | bc `
echo $c $d $e $f

read -p "Enter the value of a :" a
read -p "Enter the value of b:" b
echo "\n"
echo "Sum=\c"
expr $a + $b
echo "Difference=\c"
expr $a - $b
echo "Product=\c"
expr $a \* $b
echo "Quotient=\c"
expr $a / $b
```

Output:

```
ubuntu@ubuntu-H61M-DS2:~/soorya$ sh bccmnd.sh
Enter the value of a:10
Enter the value of b:2
Sum=12
Difference=8
Product=20
Quotient=5
ubuntu@ubuntu-H61M-DS2:~/soorya$
```

4.Write a shell Script to determine largest among three integer number.

Algorithm:

STEP 1: Start

STEP 2: Input three numbers from users

STEP 3: If num1 is greater than num 2 and num 3

STEP 4: Then print num1 is greater

STEP 5: else if num 2 is greater than num 1 and num 3

STEP 6: Then print num 2 is greater

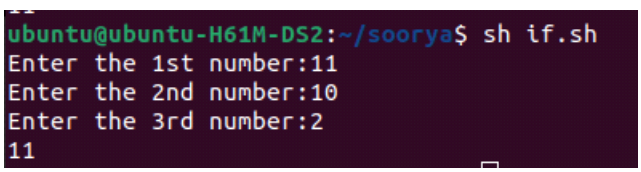
STEP 7: else print num 3 is greater

STEP 8: Stop

Shell Script:

```
echo "Enter the 1st number:"
read num1
echo "Enter the 2nd number:"
read num2
echo "Enter the 3rd number:"
read num3
if [ $num1 -gt $num2 ] && [ $num1 -gt $num3 ]
then
   echo $num1
elif [ $num2 -gt $num1 ] && [ $num2 -gt $num3 ]
then
   echo $num2
else
   echo $num3
fi
```

Output:



5.Write a shell script to compare two string.

Algorithm:

STEP 1: Start

STEP 2: Input String 1 and String 2

STEP 3: If String 1 is equal to String 2

STEP 4: Then print equal

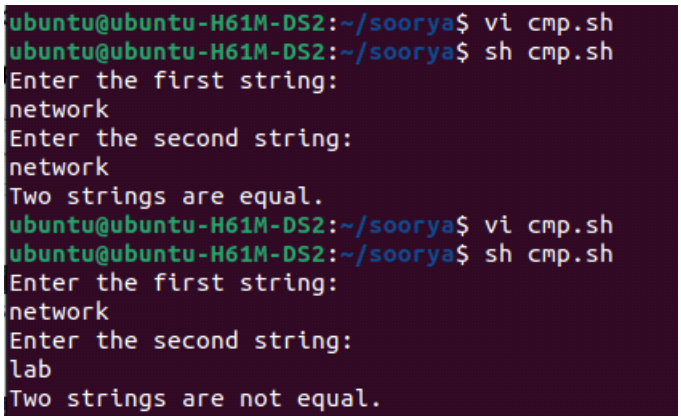
STEP 5: else echo not equal

STEP 6: Stop

Shell Script:

```
read -p "Enter the first string : " str1
read -p "Enter the second string : " str2
if [ $str1 = $str2 ]
then
 echo "Two strings are equal."
else
 echo "Two strings are not equal."
fi
```

Output:



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

Algorithm:

STEP 1: Start

STEP 2: Input the directory name

STEP 3: If the directory exists

STEP 4: Print the directory exists

STEP 5: else

STEP 6: Print the directory does not exist

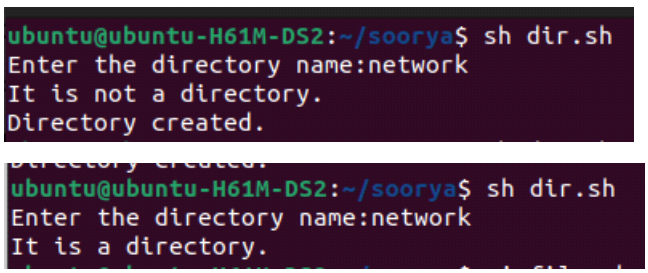
STEP 7: Create the directory

STEP 8: Stop

Shell Script:

```
read -p "Enter the  directory name : " DIR
if [ -d "$DIR" ]
then
   echo "It is a directory "
else
   echo "It is not a directory. "
   mkdir -p $DIR
   echo "Directory Created. "
fi
```

Output:



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

Algorithm:

STEP 1: Start

STEP 2: Accept a file name from user

STEP 3: Check whether it is a file or not, if it is a file print file exists

STEP 4: If it is not a file, create a file using touch command and print file created

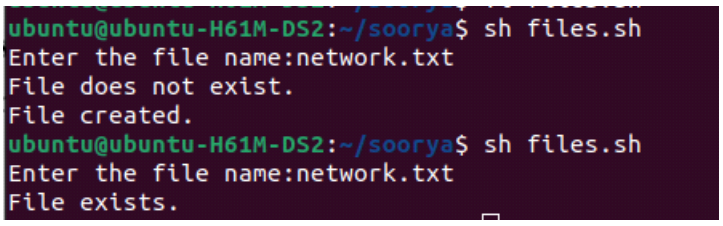
STEP 5: Use ls command to list all the items

STEP 6: Stop

Shell Script:

```
read -p "Enter the file name : " file
if [ -f "$file" ]
then
   echo "File exists."
else
   echo "File does not exist. "
   touch $file
   echo "File created."
fi
```

Output:



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

Algorithm:

STEP 1: Start

STEP 2: Accept 2 values

STEP 3: Print enter the choice

STEP 4: If the choice is Addition, then perform addition

STEP 5: If the choice is Subtraction, then perform subtraction

STEP 6: If the choice is Multiplication, then perform multiplication

STEP 8: If the choice is Division, then perform division

STEP 9: Print result

STEP 10: Stop

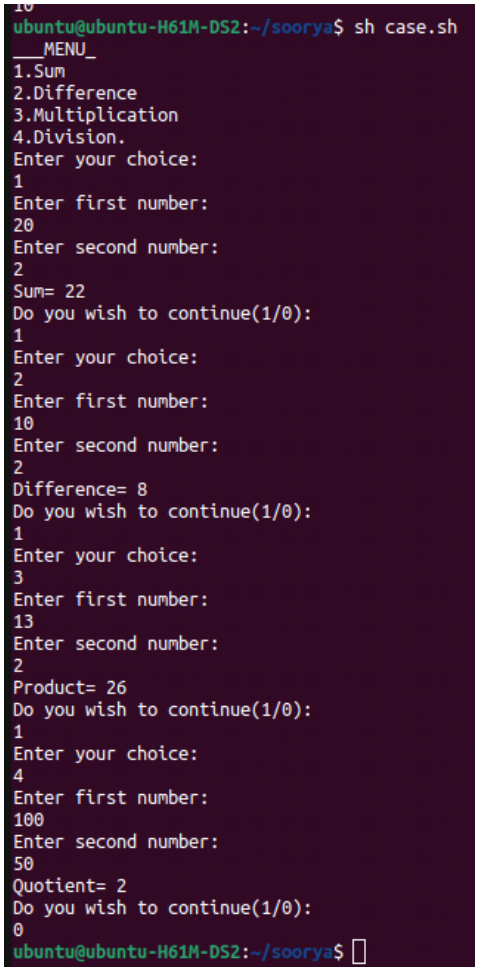
Shell Script:

```
echo”__MENU__\n1.Sum\n2.Difference\n3.Multiplication\n4.Division.”
c=1
while [ $c = 1 ]
do
echo "Enter your choice"
read ch
echo "Enter first number :"
read a
echo "Enter second number :"
read b
case $ch in

        1)s=`echo $a + $b |bc`
         echo "Sum =$s”;;
        2)diff=`echo $a - $b |bc`
         echo "Difference = $diff";;
         3)mul=`echo $a \* $b |bc`
```

```
        echo "Product = $mul”;;
        4)div=`echo $n1 / $n2 | bc`
        echo "Quotient = $div”;;
esac
echo "Do u wish to continue( 1/0):”
read c
done
```

Output:



9.Write a shell script to do:

- •display list of directory contents
- •Name of current directory
- •Who is logged in
- •Long listing of directory contents according to choose of user.

Algorithm:

STEP 1: Display list of directory contents

STEP 2: Name of current directory

STEP 3: If ch=1, then display list of directory contents

STEP 4: If ch=2, then print name of current directory

STEP 5: If ch=3, then print who is logged in

STEP 6: If ch=4, then print long list of directory contents according to choose of user

STEP 7: Otherwise, print invalid choice

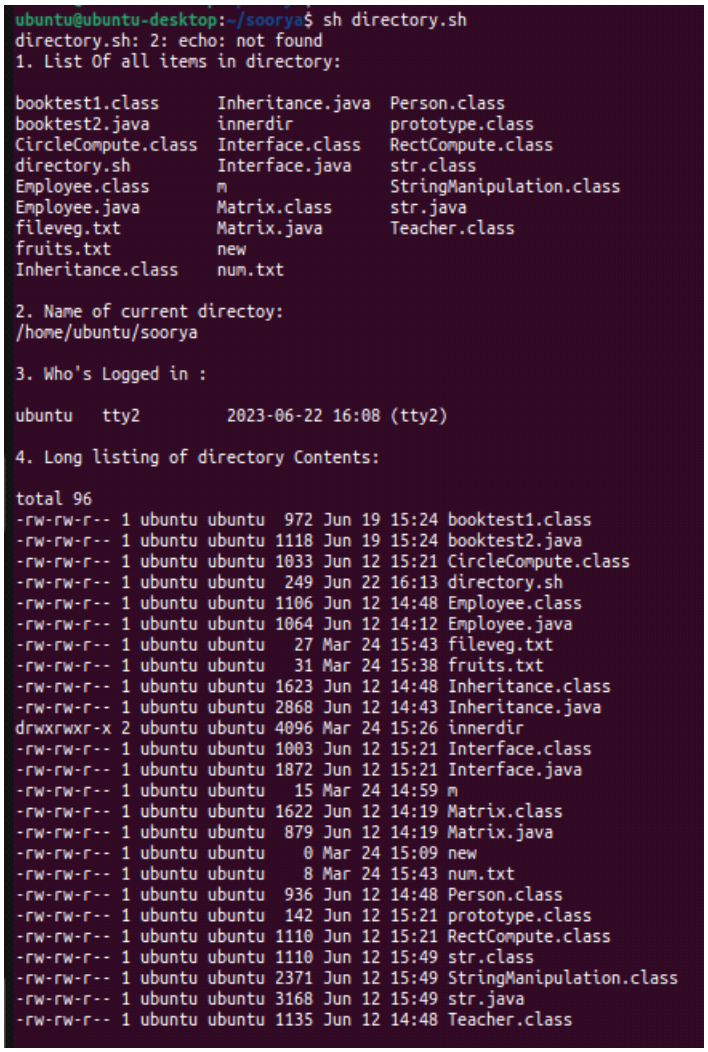
STEP 8: Stop

Shell Script:

```
i=1;
while [ $i -eq 1 ]
do
 echo "Enter choice"
 echo "1.Display list of directory contents."
 echo "2.Name of current directory."
 echo "3.Who is logged in."
 echo "4.Long listing of directory contents according to choose of   user"
 read ch
 case $ch in
  1) ls;;
  2) pwd;;
  3) who;;
  4) ls -l;;
```

```
 *) echo "Invalid choice";;
esac
echo "Do you want to continue..."
read i
if [ $i != 1 ]
then
 exit
fi
done
```

Output:



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

Algorithm:

STEP 1: Start

STEP 2: Print Roll number

STEP 3: Print name

STEP 4: Print mark

STEP 5: Stop

Shell Script:

```
echo "Name of the student:" $1;
echo "Roll number :" $2;
echo "Marks :" $3;
```

Output:

```
ubuntu@ubuntu-desktop:~/soorya$ vi CLA.sh
ubuntu@ubuntu-desktop:~/soorya$ sh CLA.sh SOORYA 22 50

Name of the student :SOORYA
Roll number :22
Marks :50
```

11.To find the sum of n natural numbers.

- •Using for loop
- •Using while loop

Algorithm:

STEP 1: Start

STEP 2: Read the limit into variable n

STEP 3: set i=1 and sum=0

STEP 4: Display the menu

STEP 5: While true do step 6 to step 16

STEP 6: Read choice as ch

STEP 7: Use case with ch as key

STEP 8: if ch=1 do step 9 to step 11

STEP 9: For i=1 to n do step 10

STEP 10: sum=sum+1

STEP 11: Display sum

STEP 12: if ch=2 while $i less then $n do step 13 to step 15

STEP 13: sum=sum+1

STEP 14: increment i

STEP 15: Display sum

STEP 16: if ch=3 do exit case

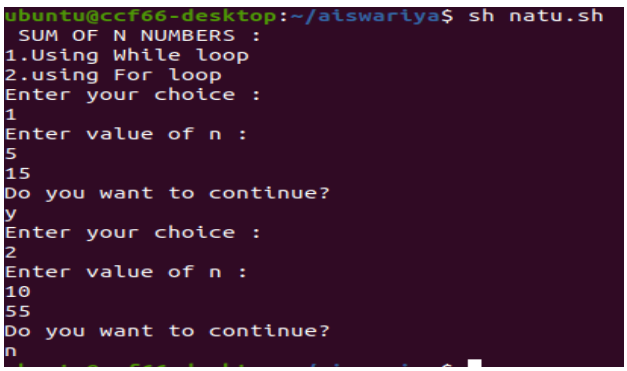
STEP 17: Stop

Shell Script:

```
a="y"
echo " SUM OF N NUMBERS : "
echo "1.Using While Loop"
echo "2.using For loop"
while [ $a = "y" ]
```

```
do
echo "Enter your choice :"
read ch
echo "Enter value of n :"
read n
case $ch in
1)i=1
  s=0
  while [ $i -le $n ]
  do
  s=$(($s + $i))
  i=$(($i + 1))
  done
  echo $s;;
2)s=0
  i=1
  for i in $(seq $n)
  do
  s=$((s + i))
  done
  echo $s;;
*)echo "invalid choice";;
esac
echo "Do you want to continue ?"
read a
done
```

Output:



12.Implement arithmetic calculator using Functions.

Algorithm:

STEP 1: Start

STEP 2: Define a function add do step 3 to step 4

STEP 3: Variable sum = (($1+$2))

STEP 4: Display sum

STEP 5: Define a function sub do step 6 to step 7

STEP 6: Variable sub = (($1+$2))

STEP 7: Display sub

STEP 8: Define a function mul to do step 9 to step 10

STEP 9: variable mul = (($1+$2))

STEP 10: Display mul

STEP 11: Define a function div to do step 12 to step 13

STEP 12: Variable div = (($1+$2))

STEP 13: Display div

STEP 14: Define a function modulo to do step 15 to step 16

STEP 15: Variable q = (($1+$2))

STEP 16: Display q

STEP 17: Display menu

STEP 18: While true do step 19 to step 27

STEP 19: Read choice as ch

STEP 20: In case use ch as key

STEP 21: If ch=1 call add()

STEP 22: If ch=2 call sub()

STEP 23: if ch=3 call mul()

STEP 24: if ch=4 call div()

STEP 25: if ch=5 do exit from while

STEP 26: If ch is invalid do display “Invalid choice”

STEP 27: Stop

Shell Script:

```
add(){
    sum=$(( $1 + $2 ))
    echo "Sum = $sum”
}
diff(){
     dif=$((  $1 - $2 ))
    echo "Difference = $dif”
}
mult(){
    mul=$(( $1 * $2 ))
    echo "Product=$mul”
}

div(){
    quo=$((  $1 / $2 ))
    echo "Quotient=$quo”
}

echo”__MENU__\n1.Sum\n2.Difference\n3.Multiplication\n4.Division/n5.Exit.”
c=1
while [ $c = 1 ]
do
echo "Enter your choice"
read ch
echo "Enter first number :"
read a
echo "Enter second number :"
read b
case $ch in
            1) add $a $b;;

            2) diff $a $b;;

            3) mult $a $b;;
```

```
        4) div $a $b;;

        5) exit;;

 esac
echo “Do you wish to continue(1/0):”
read c
done
```

Output:

```
ubuntu@ubuntu-H61M-DS2:~/soorya$ sh calcfn.sh
__MENU__
1.SUM.
2.DIFFERENCE.
3.MULTIPLICATION.
4.DIVISION.
5.EXIT.

Enter your choice:
1
Enter 1st number:
5
Enter 2nd number:
2
Sum = 7
Do you wish to continue(1/0):
1
Enter your choice:
2
Enter 1st number:
10
Enter 2nd number:
2
Difference=8
Do you wish to continue(1/0):
1
Enter your choice:
3
Enter 1st number:
4
Enter 2nd number:
5
Product=20
Do you wish to continue(1/0):
1
Enter your choice:
4
Enter 1st number:
10
Enter 2nd number:
2
Quotient=5
Do you wish to continue(1/0):
0
```

Result:

Successfully executed the Shell Scripts and received the outputs.

Experiment 5

Linux File System Hierarchy Familiarization

Aim:

To familiarize the Linux File System Hierarchy, permission, log files etc.

Linux File System

In a typical Linux distribution, the file system hierarchy follows the File system Hierarchy Standard (FHS) which defines the directory structure and organization of files on a Linux-based operating system. Here are the most common directories you would find in a Linux distribution:

/: The root directory, which contains all other directories and files in the file system.

/bin: Essential binary files that are required to boot the system and run basic commands, such as ls, cp, and mv.

/boot: Contains the files needed for the boot process, including the Linux kernel and boot loader.

/dev: Device files, which represent hardware devices such as hard drives, USB drives, and printers.

/etc: Configuration files for the system and applications installed on the system.

/home: Home directories for user accounts.

/lib: Libraries required for binaries located in /bin and /sbin.

/media: Mount points for removable media, such as USB drives and DVDs.

/mnt: Mount points for temporarily mounted file systems, such as network shares.

/opt: Optional software applications that are not part of the core system.

/proc: A virtual file system that provides information about running processes and system resources.

/root: The home directory for the root user.

/run: A temporary file system that contains data that needs to be available early in the boot process.

/sbin: System binaries that are used for system administration, such as fdisk and iptables.

/srv: Data for services provided by the system, such as web pages and FTP directories.

/sys: A virtual file system that provides information about hardware devices and drivers.

/tmp: A directory for temporary files created by applications and users.

/usr: Secondary hierarchy for read-only user data and program files.

/var: Variable data files, such as log files and spool directories for printing and mail.

Note that the above directory structure is not exhaustive, and different Linux distributions may have slightly different directory structures.

Tree command

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. 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.

$ tree -L 1 /

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.

File Permissions

In Linux, every file and directory has a set of permissions that determine who can read, write, and execute the file. File permissions are an important aspect of Linux security and are used to control access to files and directories.

There are three basic permissions for files in Linux:

- Read (r): Allows users to view the contents of the file.
- Write (w): Allows users to modify the contents of the file.
- Execute (x): Allows users to run the file as a program or a script.

These permissions are organized into three categories:

- Owner: The user who owns the file.
- Group: A group of users who have been granted access to the file.
- Other: Any user who is not the owner or a member of the group.

Each category can be granted or denied the permissions separately, resulting in a three-digit code that represents the permissions for the file. The code is calculated as follows:

- The first digit represents the owner's permissions.

- The second digit represents the group's permissions.
- The third digit represents everyone else's permissions.

The permissions are represented by numbers as follows:

- Read permission (r) = 4
- Write permission (w) = 2
- Execute permission (x) = 1
- No permission (-) = 0

For example, if a file has the permissions -rw-r--r--, this means that the owner can read and write the file, while members of the group and other users can only read the file.

To view the permissions of a file or directory, you can use the ls -l command. This will show the file permissions in the first column of the output, along with other information about the file or directory.

Device Permissions

In Linux, device permissions are used to control access to hardware devices such as disk drives, printers, and USB devices. Device permissions are similar to file permissions, but they apply specifically to device files, which are used to interact with hardware devices.

Device files are located in the /dev directory and are organized into two types: block devices and character devices.

Block devices are used to store and retrieve data in fixed-size blocks, while character devices are used for streaming data in a continuous flow. Examples of block devices include hard drives, USB drives, and memory cards. Examples of character devices include keyboards, mice, and printers.

Device files have a major and minor number associated with them, which identifies the device driver that is used to interact with the hardware device. The permissions for device files are set using the chmod command, just like file permissions.

The permissions for device files consist of three parts:

- The file type (c for character devices or b for block devices).
- The permissions for the device owner.
- The permissions for the group and everyone else.

The permissions for the device owner and group are the same as file permissions: read (r), write (w), and execute (x). However, the execute permission is not used for device files. Instead, the execute bit is used to indicate whether the file is a character device (set to 1) or a block device (set to 0).

To change the permissions for a device file, you can use the chmod command, followed by the permissions in octal notation. For example, to set the permissions for a block device file to 660, you would use the following command:

chmod 660 /dev/sda

This would give the owner and group read and write permissions, but no one else would have access to the device file.

System Configuration files in /etc

In Linux, system configuration files are stored in the /etc directory. These files contain system-wide settings and configurations for various applications and services on the system. Here are some common examples of system configuration files found in the /etc directory:

- /etc/passwd: This file contains user account information such as usernames, user IDs, home directories, and login shells.
- /etc/group: This file contains group information such as group names and group IDs.
- /etc/fstab: This file contains information about file systems and how they should be mounted at boot time.
- /etc/resolv.conf: This file contains information about the system's DNS (Domain Name System) configuration, such as the IP addresses of the DNS servers.
- /etc/hosts: This file contains mappings of hostnames to IP addresses, which are used for local name resolution.
- /etc/ssh/sshd_config: This file contains configuration settings for the OpenSSH server, such as the port number to listen on, authentication methods, and allowed users.
- /etc/sudoers: This file contains configuration settings for the sudo command, which allows users to run commands with elevated privileges.
- /etc/crontab: This file contains the system-wide crontab, which is used to schedule tasks to run at specific times.
- /etc/sysctl.conf: This file contains system-level kernel settings, such as network buffer sizes and file system limits.
- /etc/hosts.allow and /etc/hosts.deny: These files are used to control access to the system's network services, such as SSH and FTP, by specifying which hosts are allowed or denied access.

Log files for System events

In Linux, log files are used to record system events and provide a detailed history of system activity. System log files are typically stored in the /var/log directory and are divided into several categories based on the type of system event being logged. Here are some common examples of system log files and their purposes:

- syslog: This file contains messages from the system logging daemon (syslogd) that relate to general system activity, such as system startup and shutdown, user logins and logouts, and kernel messages.
- auth.log: This file contains messages related to system authentication, such as user logins, password changes, and authentication failures.
- dmesg: This file contains messages generated by the kernel at boot time, including hardware detection, driver initialization, and system configuration.
- messages: This file contains messages generated by various system daemons, such as the mail server (sendmail) and the print spooler (cups).
- secure: This file contains messages related to security events, such as successful and failed login attempts, and changes to user account information.
- cron: This file contains messages related to scheduled tasks that are executed by the cron daemon.
- kernel.log: This file contains messages related to kernel activity, such as system crashes and kernel panics.

The grep command can be used to search through log files and filter out relevant information. For example, to search for authentication-related messages in the auth.log file, you could use the following command:

grep "authentication" /var/log/auth.log

Log files for user activity

In Linux, user activity log files record information about user activities on the system. These log files can be used to track user actions, detect unauthorized access, and troubleshoot issues related to user accounts. Here are some common examples of log files that track user activity:

- wtmp: This file contains a record of user logins and logouts, including the date and time of the event, the username, and the terminal or IP address used to login.
- lastlog: This file contains a record of the last login time and location for each user account on the system.
- btmp: This file contains a record of failed login attempts, including the username, the date and time of the event, and the terminal or IP address used.
- history: This file contains a list of commands executed by each user in their respective home directory. This file is only available for the user who executed the commands and can be accessed by running the history command.
- sudo log: This file contains a record of commands executed with elevated privileges using the sudo command.

The grep command can be used to search through log files and filter out relevant information. For example, to search for failed login attempts in the btmp file, you could use the following command:

grep "failed login" /var/log/btmp

Log file for network events

In Linux, network activity log files record information about network events on the system, such as network connections and data transfers. These log files can be used to monitor network activity, detect network-related issues, and identify potential security breaches. Here are some common examples of log files that track network activity:

- syslog: This file contains messages related to network events, such as network interface status changes and network-related errors.
- kern.log: This file contains messages related to kernel-level network events, such as network interface driver initialization and network stack events.
- auth.log: This file contains messages related to network authentication, such as login attempts and access control events.
- messages: This file contains messages generated by various system daemons, including network-related daemons such as the Domain Name System (DNS) resolver (dnsmasq) and the Dynamic Host Configuration Protocol (DHCP) client (dhclient).
- tcpdump: This is not a log file, but a network packet capture tool that can be used to capture and analyze network traffic in real time.

The grep command can be used to search through log files and filter out relevant information. For example, to search for network-related events in the syslog file, you could use the following command:

grep "network" /var/log/syslog

Result:

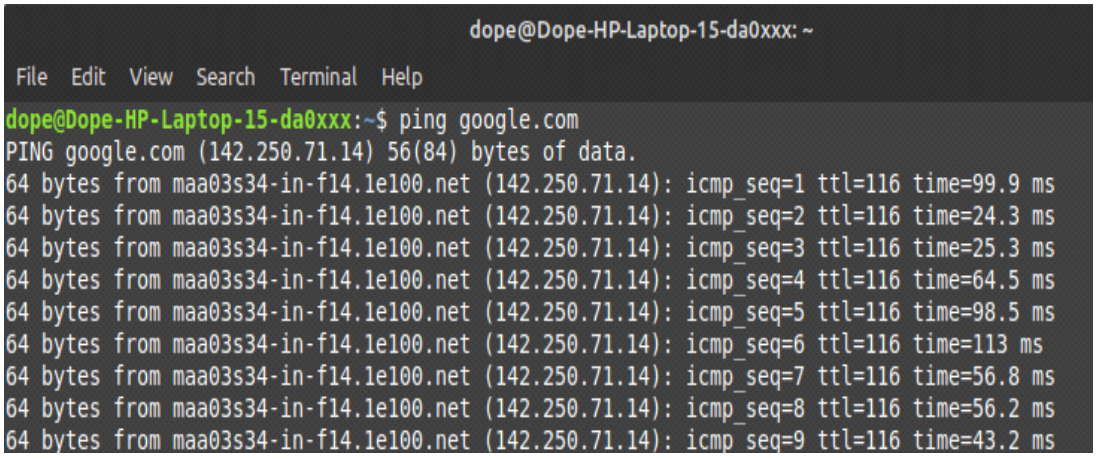
Familiarized with the Linux File System Hierarchy, permission, log files etc.

Experiment 6

Linux Network 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.

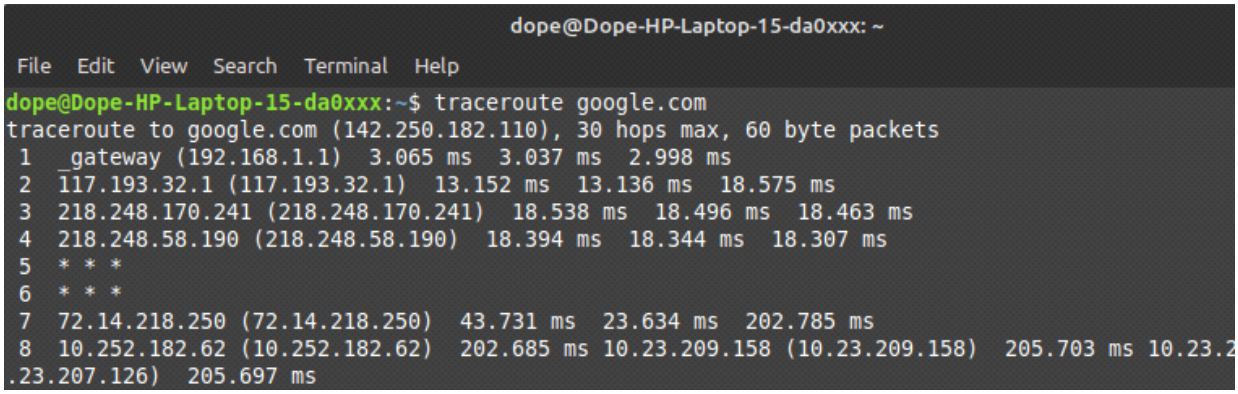


Syntax:

ping [OPTIONS] DESTINATION

Traceroute command

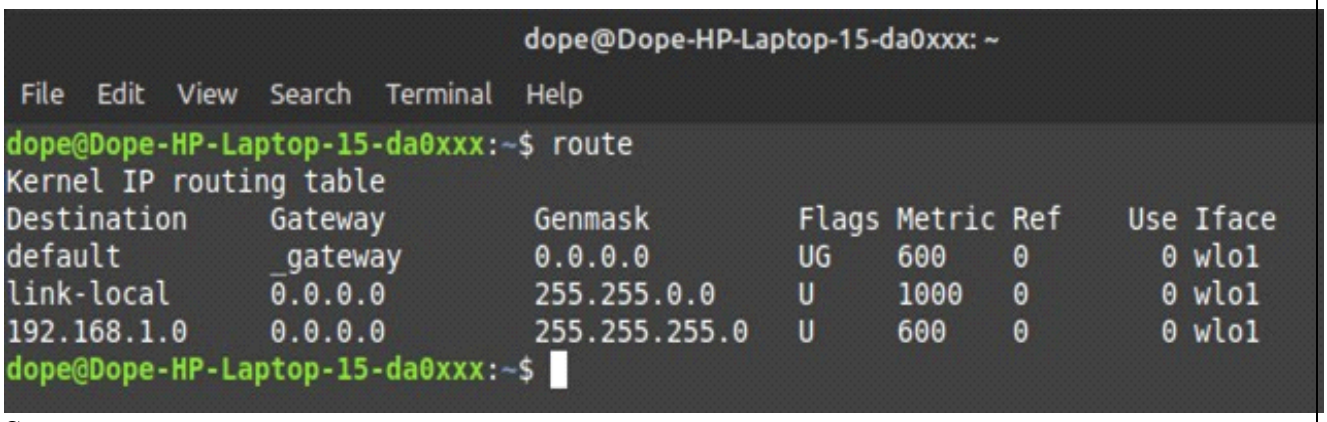
Traceroute command in Linux prints the route that a packet takes to reach the host. This command is useful when you want to know about the route and about all the hops that a packet takes. Below image depicts how traceroute command is used to reach the Google (172.217.26.206) host from the local machine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



Syntax: traceroute [options] host_Address [pathlength]

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.

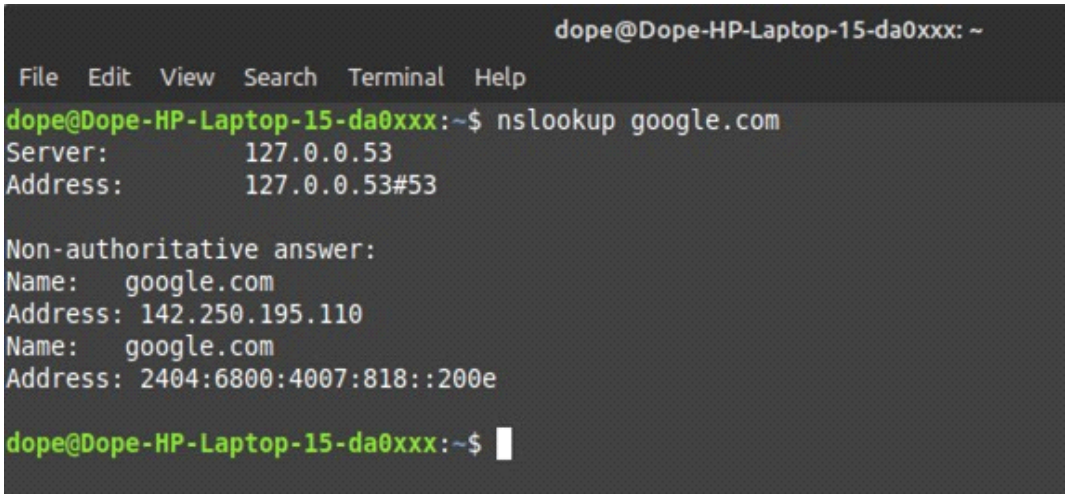


Syntax: route

Nslookup command

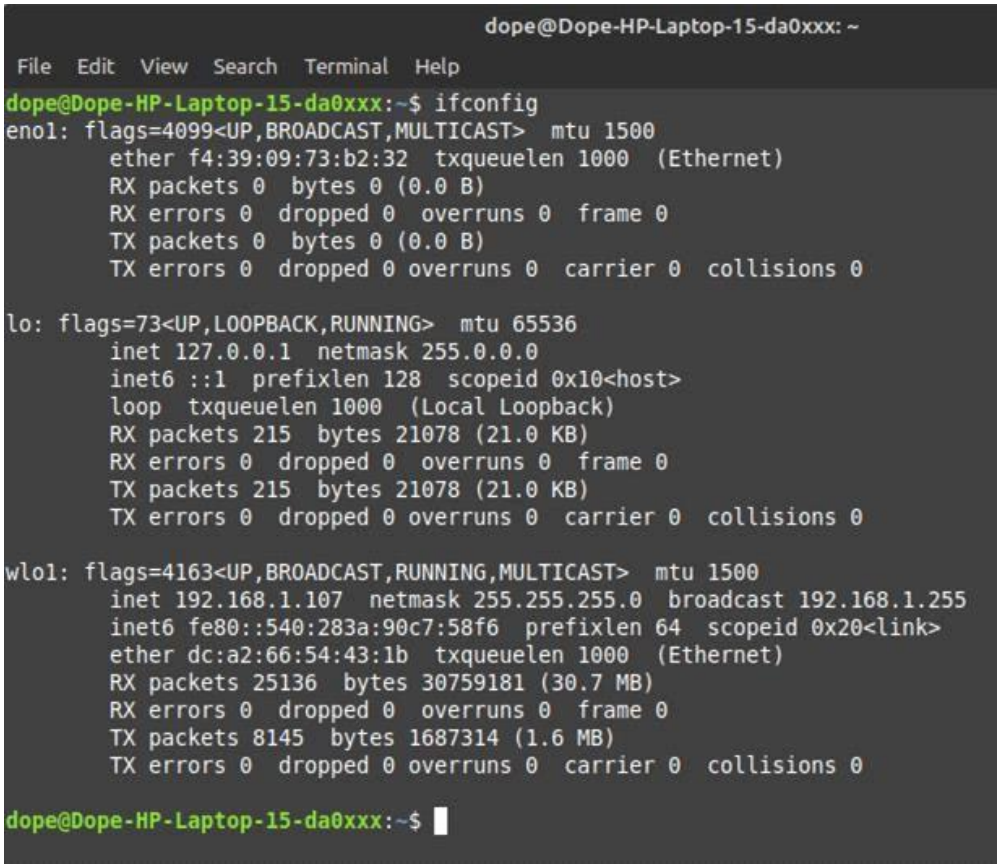
nslookup (stands for "Name Server Lookup") is a useful command for getting information from DNS server. It is a network administration tool for querying the Domain Name System (DNS) to obtain domain name or IP address mapping or anyother specific DNS record. It is also used to troubleshoot DNS related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

Syntax: nslookup [option]



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 interfaceor to enable or disable a given interface.



Syntax: ifconfig [...OPTIONS] [INTERFACE]

Experiment 7

VirtualBox installation

VirtualBox is a free virtualization program that makes it extremely convenient to set up virtual machines on different operating systems. Oracle VM VirtualBox is cross-platform virtualization software. It allows users to extend their existing computer to run multiple operating systems including Microsoft Windows, Mac OS X, Linux, and Oracle Solaris, at the same time.

Prerequisites

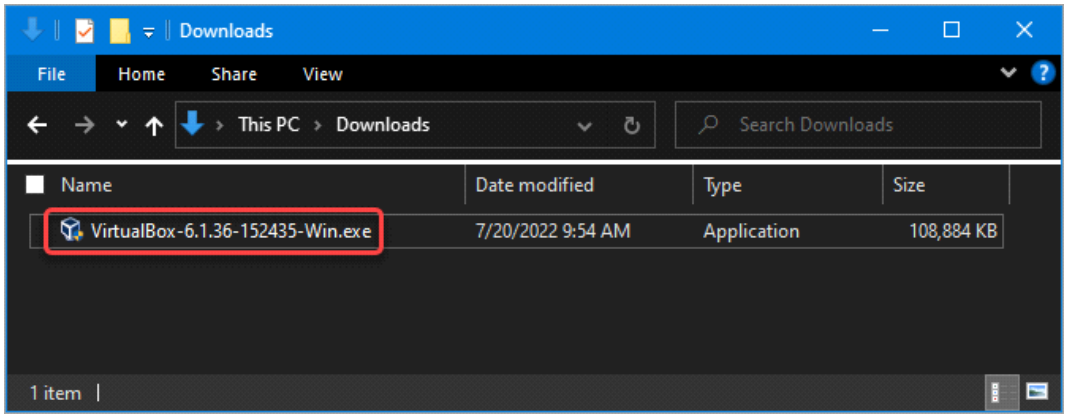
- A Windows 10 computer. VirtualBox should work fine on any recent version of Windows 10 or 11.
- An ISO file for the operating system you want to install. This ISO could be of any VirtualBox-supported guest OSes.
- Free storage space. How much you need will depend on the size of the ISO you're using, but in general, there should be at least 20 GB of free disk space.
- RAM. How much you need will depend on the operating system you want to install and how you plan on using it.

Installing VirtualBox on Windows 10

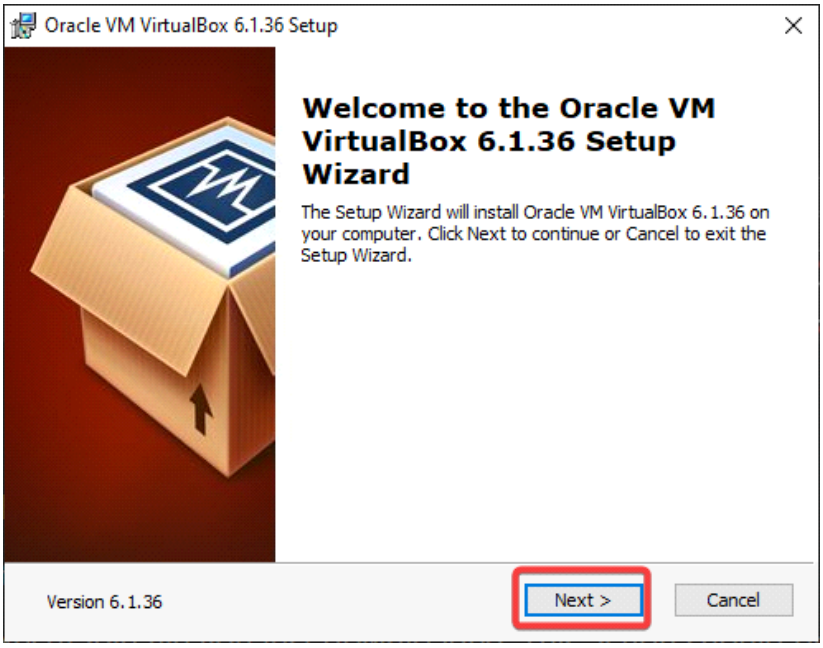
To install VirtualBox on Windows 10, you must first download the appropriate installation file for your host. From there, the installation process uses a Wizard interface, which should not be too complicated, even for a beginner.

1. Open a browser and open the VirtualBox downloads page and click the Windows hosts link. The latest VirtualBox version as of this writing is 6.1.36. Save the installer somewhere you can quickly locate it, like your Downloads folder.

- Locate the VirtualBox installer file using your File Explorer. Double-click the file to launch the VirtualBox Setup wizard.



- Click Next on the first screen. This action tells the Wizard that you want to install VirtualBox.

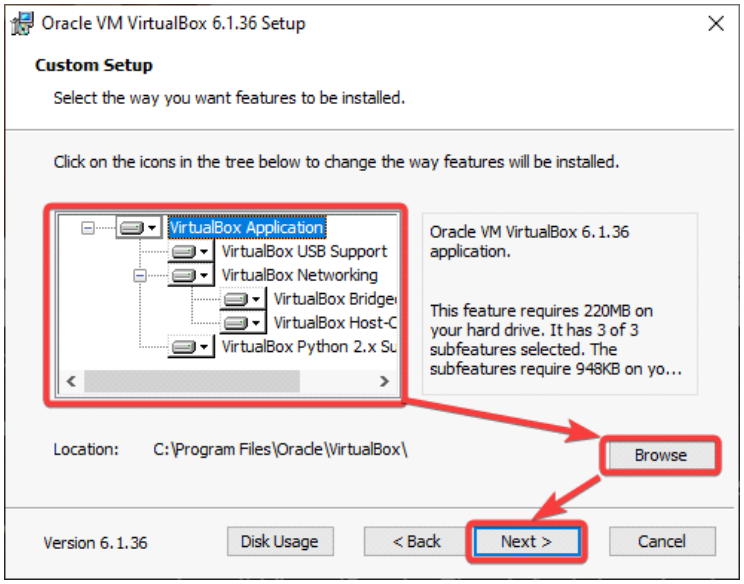


4. Do the following actions on the Custom Setup screen.

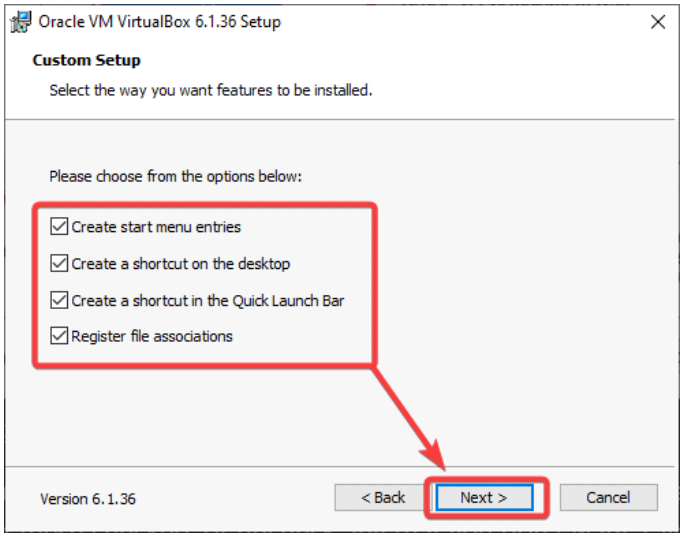
You'll see a list of the features the Wizard will install. In this example, leave the default selection.

Browse and select the location you want to install VirtualBox in. The default location is fine, but feel free to change it if you prefer.

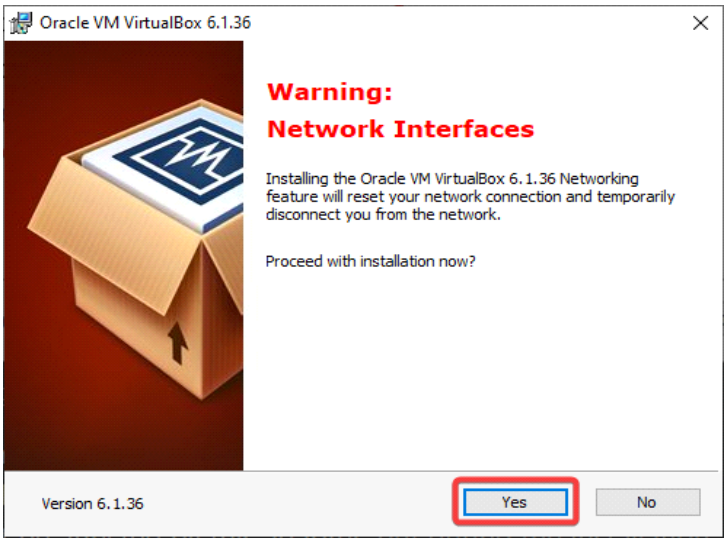
Click Next when you're ready to continue.



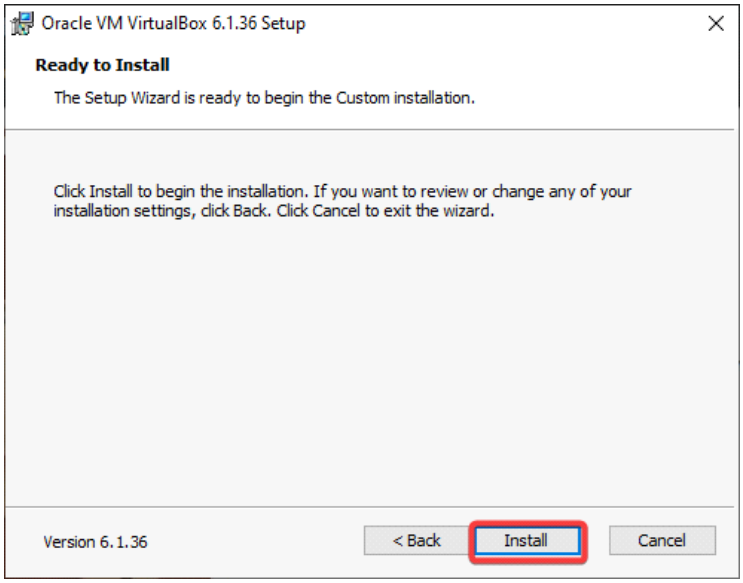
- On the next screen, you'll see a list of the shortcuts and file associations the installation will create. Check or uncheck the items you wish to include or exclude and click Next. This example leaves all options checked.



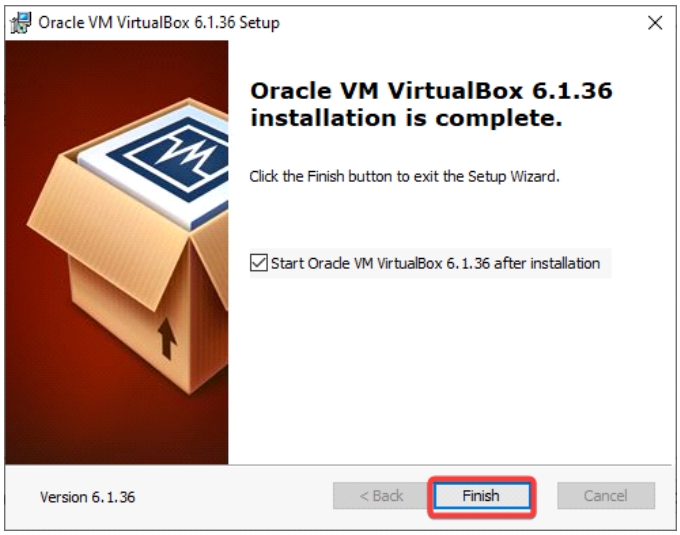
- On the next screen, you'll see a warning about networking. The setup process will install a virtual network adapter, which may cause your network connection to disconnect momentarily. Click Yes to continue.



7. Finally, you'll see a screen asking you to confirm the installation. Click Install to install VirtualBox on Windows 10.



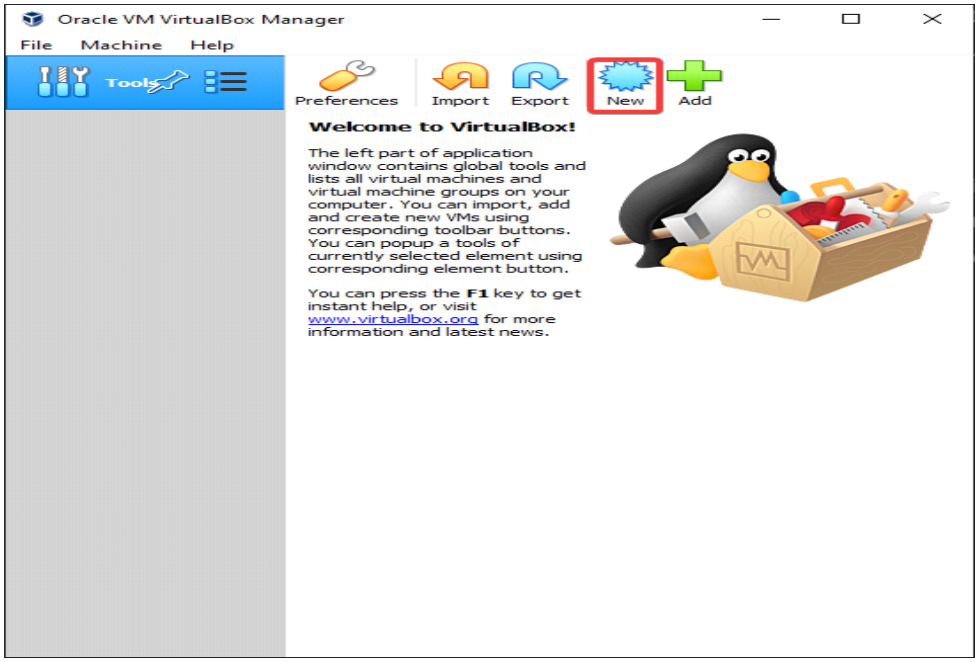
- The installation process takes several minutes, depending on your system speed. Click Finish to close the Wizard after the installation and start using VirtualBox.



Creating Your First Virtual Machine

After you install VirtualBox on Windows 10, your can create your first virtual machine

1. First, click the New button in the top-right corner of the VirtualBox window. This action brings up the Create Virtual Machine wizard. This Wizard lets you configure your new VM with the settings you want.



Click the New button

2. Configure the following on the Name and operating system page.

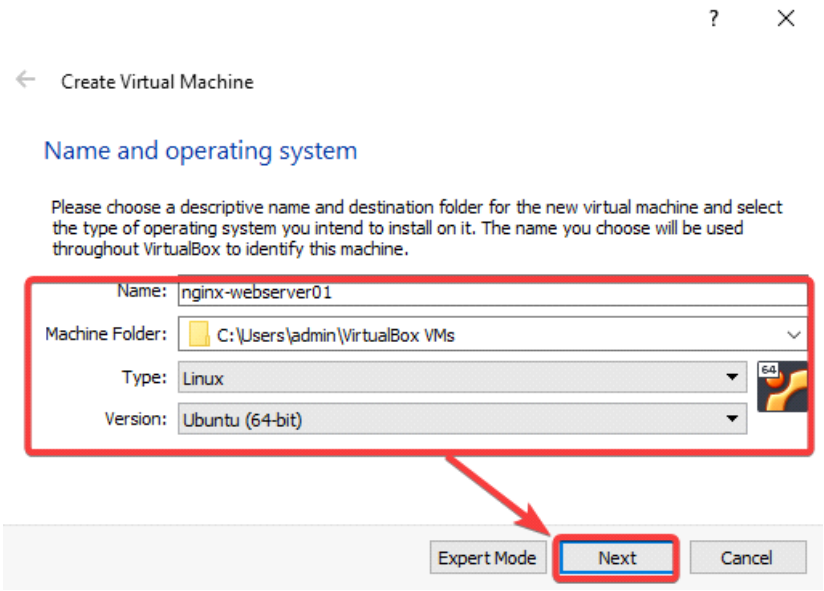
Name: Give your VM a name. This name can be anything you want.

Machine Folder: Specify the location to store your virtual machine files. You can leave the default location or click the Browse button to select a different one.

Type: VirtualBox offers many different types of OSes. Select the operating system you want to install from the drop-down menu. This tutorial selects Linux for the type.

Version: Select the version of the operating system you want to install from the drop-down menu. This tutorial goes with Ubuntu 20.04 LTS 64bit, so its version is Ubuntu (64-bit).

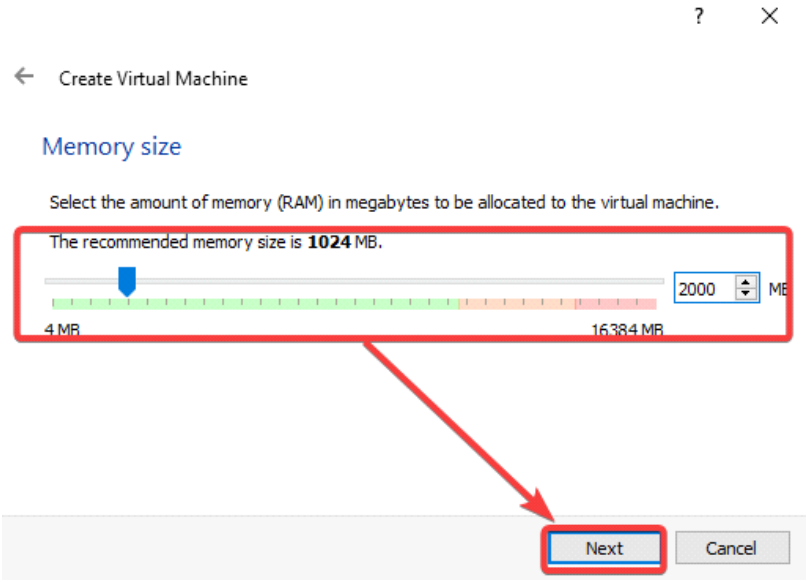
Click Next to continue.



3. On the Memory size page, set how much memory (RAM) your VM will have. You can choose between 4MB (this value is too low for any OS) to 16384MB. The default is 1024MB, but you can increase it.

Generally, 2000MB is a good starting point for a Linux VM, while 4096MB for a Windows VM.

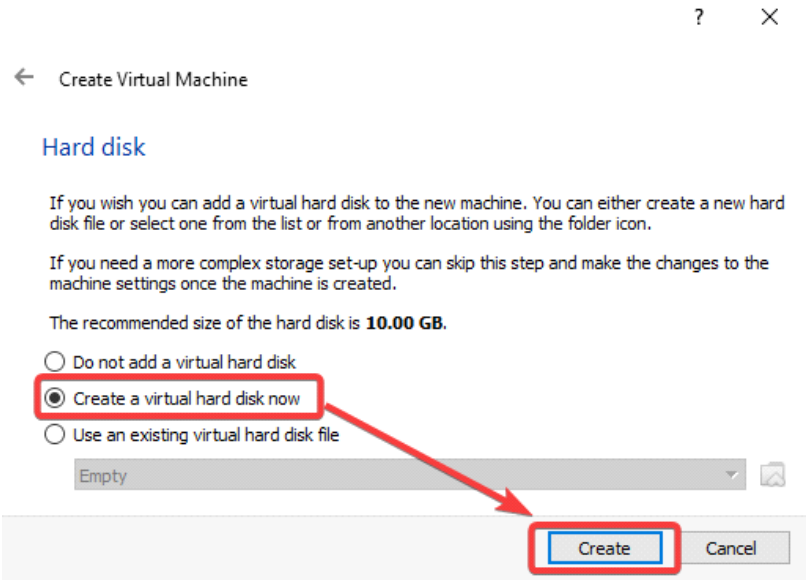
Click Next to continue.



Configure the memory size.

4. On the Hard disk screen, decide whether to create a new hard disk for your guest VM or use an existing one. This example chooses to create a new hard disk since.

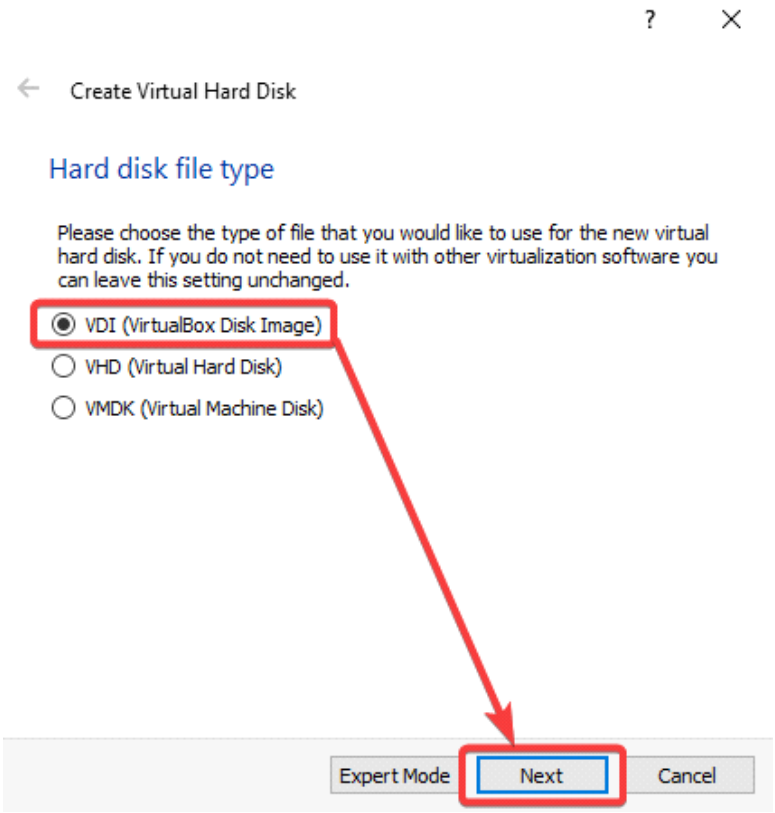
This virtual hard disk will be your guest VM's primary hard disk that holds all the data, such as the guest OS, apps, and other files.



Configure the hard disk

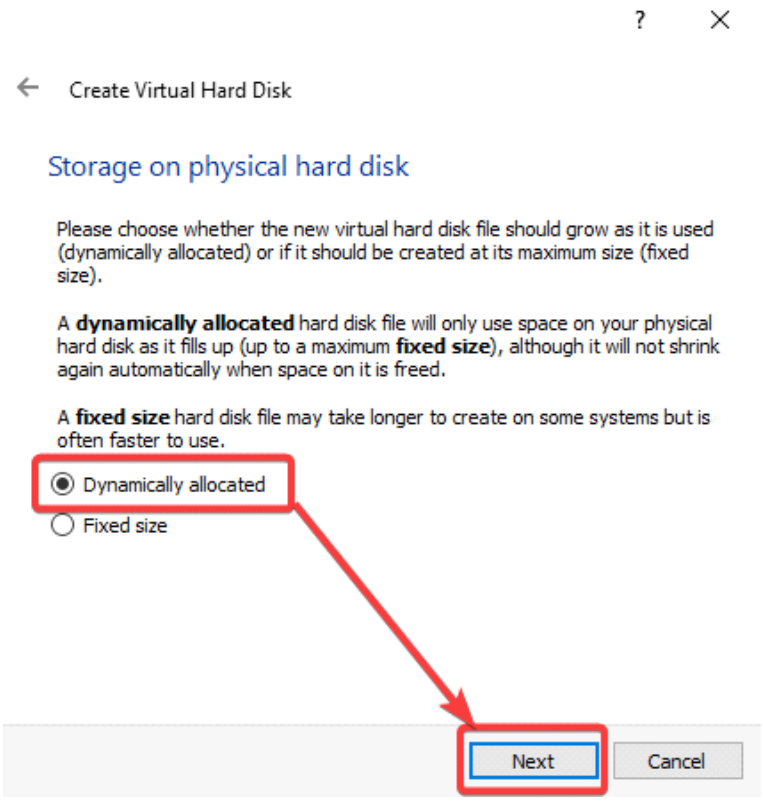
5. On the Hard disk file type page, select VDI (VirtualBox Disk Image) and click Next to continue. This file type is the default for VirtualBox.

You may choose the corresponding option if you plan to use the virtual hard disk with other virtualization solutions, like VMware or Hyper-V.



Selecting VDI (VirtualBox Disk Image).

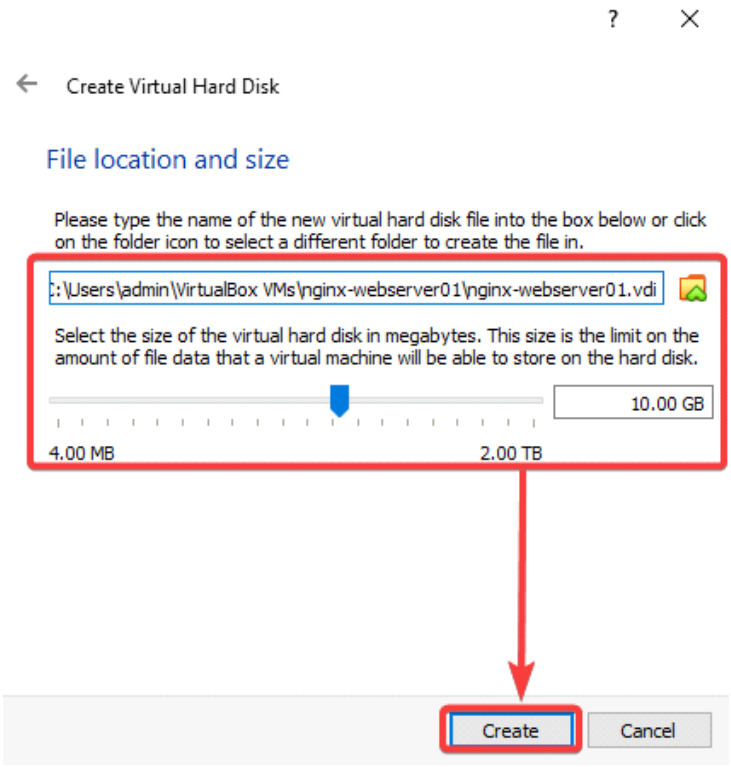
6. On the Storage on physical hard disk, select Dynamically allocated or Fixed size, depending on how you want to provision the virtual disk file size allocation. This example chooses the Dynamically allocated option. Click Next to continue.



Select Dynamically allocated

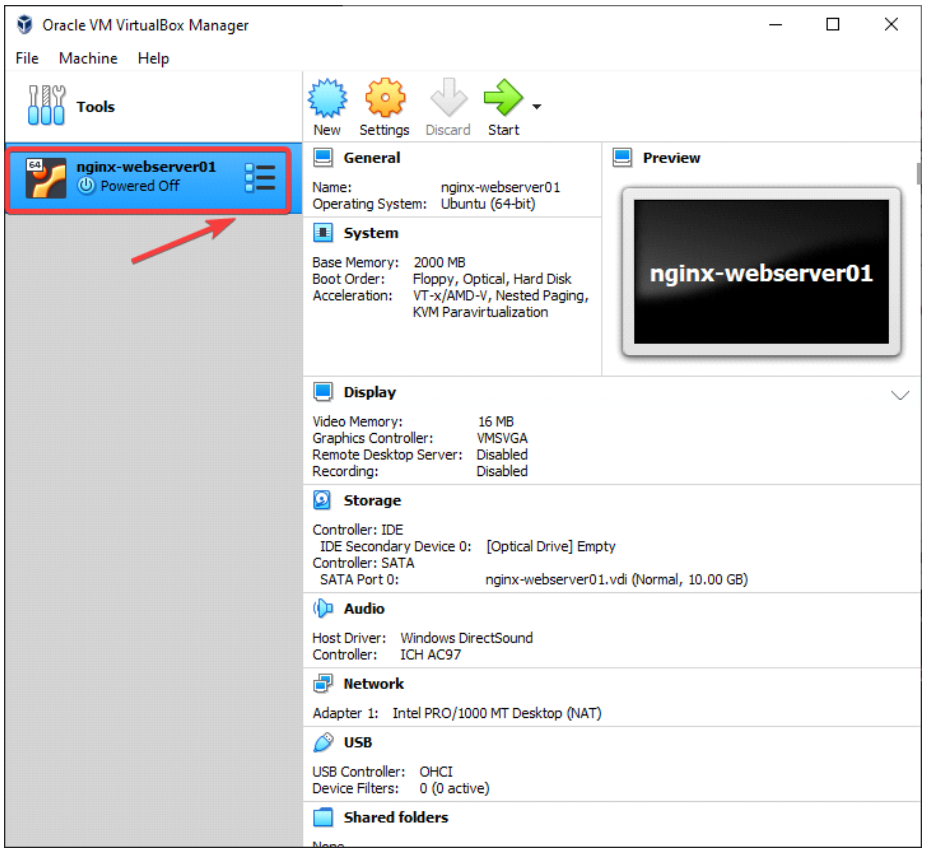
7. On the File location and size screen, specify the location and maximum size of the new virtual disk, and click Create.

A good rule of thumb is to allocate at least 10GB for a Linux VM and 40GB for a Windows VM.



Set the disk location and maximum size

You have a new virtual VM, as shown in the VirtualBox Manager window below.



Oracle VirtualBox Manager

Experiment 8

LAMP Stack Installation on UBUNTU

A LAMP (Linux, Apache, MySQL, PHP) stack is a common, free, and open-source web stack used for hosting web content in a Linux environment. Many consider it the platform of choice on which to develop and deploy high-performance web apps. For a web application to work, it has to include a server operating system, a web server, a database, and a programming language. Each layer of software is necessary for creating a database-driven and dynamic website.

Prerequisites

- Ubuntu 18.04 or later
- User with sudo privileges
- Access to a terminal/command line

How to Install LAMP in Ubuntu

LAMP is a collection of four components that make up a fully functional web development environment. The LAMP acronym contains the initials of the components' names:

- **Linux** Operating System
- **Apache** HTTP Server
- **MySQL** database management system
- **PHP** programming language (Pearl and Python are also sometimes used in the stack)

Follow the steps below to install each tool on your system.

Step 1: Install Apache

Apache HTTP Server is the web server running on top of Linux in the LAMP stack. The web server uses HTTP to process requests and transmit information through the internet.

Follow the procedure below to install Apache.

1. Before installing the first LAMP component, ensure the package list on the system is up to date. In the terminal, type:

sudo apt update

2. To install the Apache package, run the following command:

sudo apt install apache2 –y

```
marko@test-main:~$ sudo apt install apache2 -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following NEW packages will be installed:
  apache2
0 upgraded, 1 newly installed, 0 to remove and 11 not upgraded.
Need to get 95.6 kB of archives.
After this operation, 543 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 apache2 amd64 2.4.41
-4ubuntu3.12 [95.6 kB]
Fetched 95.6 kB in 1s (119 kB/s)
Selecting previously unselected package apache2.
(Reading database ... 174557 files and directories currently installed.)
Preparing to unpack .../apache2_2.4.41-4ubuntu3.12_amd64.deb ...
Unpacking apache2 (2.4.41-4ubuntu3.12) ...
Setting up apache2 (2.4.41-4ubuntu3.12) ...
Processing triggers for systemd (245.4-4ubuntu3.18) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for ufw (0.36-6ubuntu1) ...
marko@test-main:~$
```

3. Check if Apache installed correctly by checking the Apache service status:

sudo service apache2 status

The service shows as running in the output:

```
marko@test-main:~$ sudo service apache2 status
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enab>
     Active: active (running) since Thu 2022-10-13 05:26:21 EDT; 4min 53s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 15228 (apache2)
      Tasks: 6 (limit: 11573)
     Memory: 9.4M
     CGroup: /system.slice/apache2.service
             ├─15228 /usr/sbin/apache2 -k start
             ├─15230 /usr/sbin/apache2 -k start
             ├─15231 /usr/sbin/apache2 -k start
             ├─15233 /usr/sbin/apache2 -k start
             ├─15234 /usr/sbin/apache2 -k start
             └─15235 /usr/sbin/apache2 -k start

Oct 13 05:26:21 test-main systemd[1]: Starting The Apache HTTP Server...
Oct 13 05:26:21 test-main apachectl[15227]: AH00558: apache2: Could not reliably deter>
Oct 13 05:26:21 test-main systemd[1]: Started The Apache HTTP Server.
lines 1-18/18 (END)
```

Exit the status screen by pressing Ctrl + C on the keyboard.

4. Next, make sure that the UFW firewall contains the Apache profiles by typing in the following command:

sudo ufw app list

```
marko@test-main:~$ sudo ufw app list
Available applications:
  Apache
  Apache Full
  Apache Secure
  CUPS
marko@test-main:~$
```

5. Ensure the Apache Full profile allows the traffic on ports 80 and 443 by running the command:

sudo ufw app info "Apache Full"

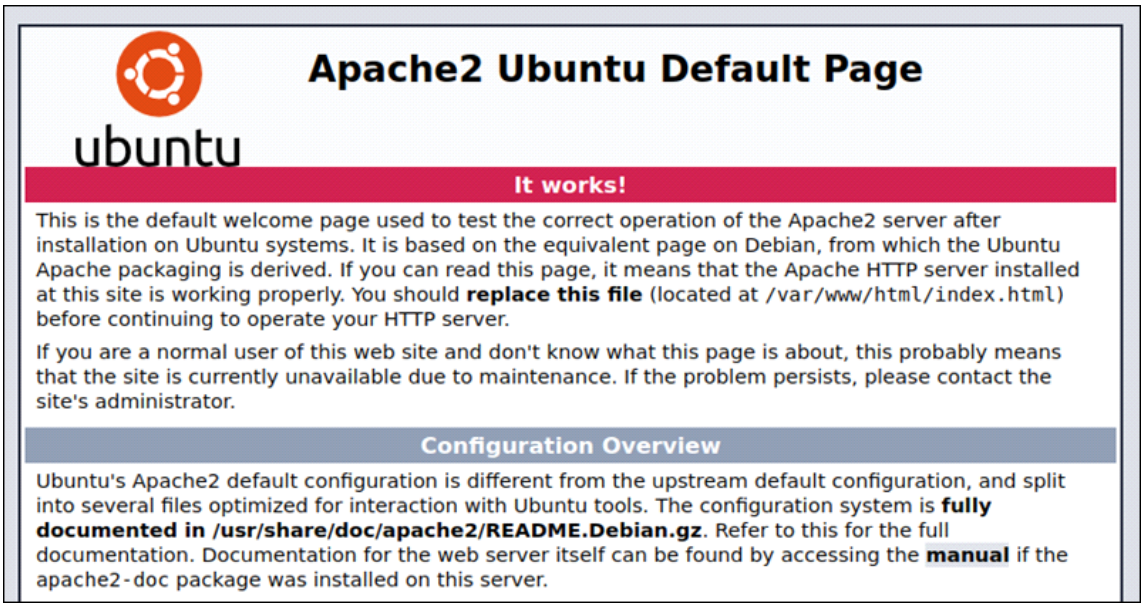
The output should look similar to the following example:

```
marko@test-main:~$ sudo ufw app info "Apache Full"
Profile: Apache Full
Title: Web Server (HTTP,HTTPS)
Description: Apache v2 is the next generation of the omnipresent Apache web
server.

Ports:
  80,443/tcp
marko@test-main:~$
```

6. To confirm that Apache is running, enter the IP address of your server in the address bar of an internet browser and press **ENTER**.

The test Apache web server page should display as below.



Step 2: Install MySQL and Create a Database

MySQL is a relational database management system for creating and maintaining dynamic enterprise-level databases. It is compatible with all major OS platforms, which makes it a good fit for web application development.

Install MySQL by typing the following command:

sudo apt install mysql-server –y

```
Setting up mysql-server-8.0 (8.0.30-0ubuntu0.20.04.2) ...
update-alternatives: using /etc/mysql/mysql.cnf to provide /etc/mysql/my.cnf (my.cnf) i
n auto mode
mysqld will log errors to /var/log/mysql/error.log
mysqld is running as pid 19563
Setting up mysql-server (8.0.30-0ubuntu0.20.04.2) ...
Processing triggers for systemd (245.4-4ubuntu3.18) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.9) ...
marko@test-main:~$
```

Step 3: Install PHP

Although other programming languages, such as Python and Pearl, also work well within LAMP, PHP is usually the final layer of the stack because it integrates well with MySQL. As a dynamically typed language, PHP embeds into HTML, improving the speed and reducing the complexity of web applications.

Install PHP by following the steps below.

1. Obtain the necessary PHP packages by typing:

sudo apt install php libapache2-mod-php php-mysql –y

```
Setting up php7.4-mysql (7.4.3-4ubuntu2.13) ...

Creating config file /etc/php/7.4/mods-available/mysqlnd.ini with new version

Creating config file /etc/php/7.4/mods-available/mysqli.ini with new version

Creating config file /etc/php/7.4/mods-available/pdo_mysql.ini with new version
Setting up libapache2-mod-php (2:7.4+75) ...
Setting up php-mysql (2:7.4+75) ...
Processing triggers for libapache2-mod-php7.4 (7.4.3-4ubuntu2.13) ...
Processing triggers for php7.4-cli (7.4.3-4ubuntu2.13) ...
marko@test-main:~$
```

2. Modify the way Apache serves files by opening the ***dir.conf*** file in a text editor with root privileges:

sudo nano /etc/apache2/mods-enabled/dir.conf

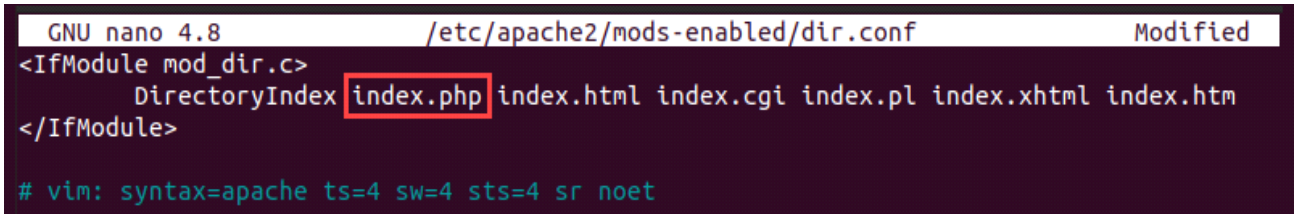
The configuration file looks like in the example below:

```
  GNU nano 4.8                 /etc/apache2/mods-enabled/dir.conf
<IfModule mod_dir.c>
        DirectoryIndex index.html index.cgi index.pl index.php index.xhtml index.htm
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

By default, Apache first looks for an ***index.html*** file card.

3. Edit the list so that the ***index.php*** file is in the first position:



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

Step 4 – Installing phpMyAdmin

phpMyAdmin provides a user friendly web interface to manage MySQL database server. You can install phpMyAdmin on Ubuntu 20.04 by executing the following command:

sudo apt install phpmyadmin

The installation process will prompt to select web server to configure. Select "Apache" as web server to run phpMyAdmin.

Next, this will prompt to create database for phpMyAdmin and prompt for the administrative user access details. Complete all steps to finish phpMyAdmin installation.

Step 5 – Manage Services

We have done with the installation of LAMP stack on Ubuntu 20.04 LTS system. The below commands will help you to start/stop or restart Apache and MySQL services running with systemd.

To restart Apache and MySQL services, type:

sudo systemctl restart apache2

sudo systemctl restart mysql

To start Apache and MySQL services, type:

sudo systemctl start apache2

sudo systemctl start mysql

To stop Apache and MySQL services, type:

sudo systemctl stop apache2

sudo systemctl stop mysql

Step 6: Test PHP Processing on Web Server

To test the new LAMP installation, create a basic PHP script and place it in the web root directory located at **/var/www/html/**, then check if the script is accessible via an internet browser. The steps below explain the procedure for performing this test.

1. Create a file in the web root directory by typing the following command:

sudo nano /var/www/html/info.php

2. Inside the file, type the PHP code:

<?php

phpinfo ();

?>

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

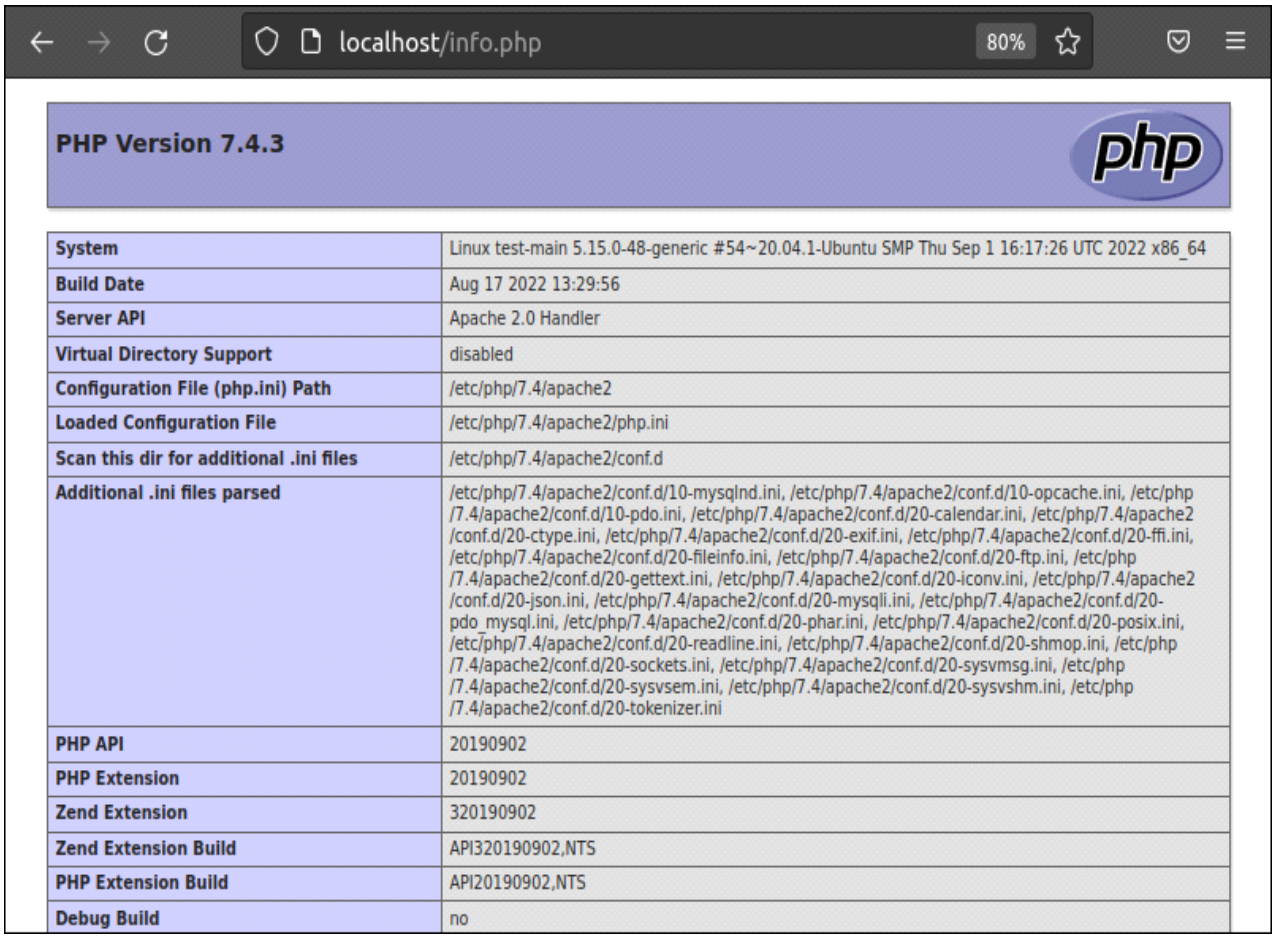
4. Open an internet browser and type the following address:

[server-ip-address]/info.php

Alternatively, type:

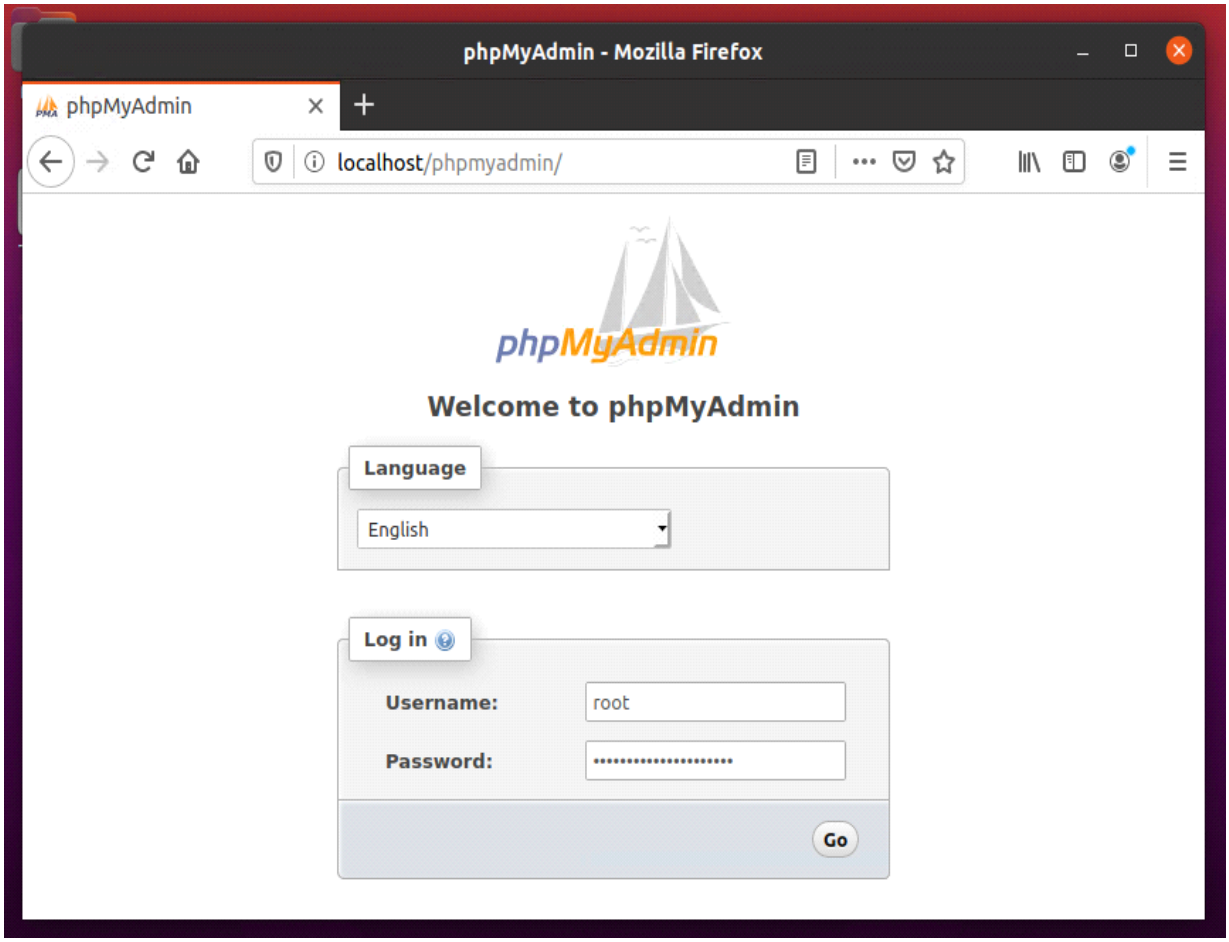
localhost/info.php

The output should display the details of the LAMP stack, as seen in the image below:



System	Linux test-main 5.15.0-48-generic #54~20.04.1-Ubuntu SMP Thu Sep 1 16:17:26 UTC 2022 x86_64
Build Date	Aug 17 2022 13:29:56
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.4/apache2
Loaded Configuration File	/etc/php/7.4/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.4/apache2/conf.d
Additional .ini files parsed	/etc/php/7.4/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.4/apache2/conf.d/10-opcache.ini, /etc/php/7.4/apache2/conf.d/10-pdo.ini, /etc/php/7.4/apache2/conf.d/20-calendar.ini, /etc/php/7.4/apache2/conf.d/20-ctype.ini, /etc/php/7.4/apache2/conf.d/20-exif.ini, /etc/php/7.4/apache2/conf.d/20-ffi.ini, /etc/php/7.4/apache2/conf.d/20-fileinfo.ini, /etc/php/7.4/apache2/conf.d/20-ftp.ini, /etc/php/7.4/apache2/conf.d/20-gettext.ini, /etc/php/7.4/apache2/conf.d/20-iconv.ini, /etc/php/7.4/apache2/conf.d/20-json.ini, /etc/php/7.4/apache2/conf.d/20-mysqli.ini, /etc/php/7.4/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.4/apache2/conf.d/20-phar.ini, /etc/php/7.4/apache2/conf.d/20-posix.ini, /etc/php/7.4/apache2/conf.d/20-readline.ini, /etc/php/7.4/apache2/conf.d/20-shmop.ini, /etc/php/7.4/apache2/conf.d/20-sockets.ini, /etc/php/7.4/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.4/apache2/conf.d/20-sysvsem.ini, /etc/php/7.4/apache2/conf.d/20-sysvshm.ini, /etc/php/7.4/apache2/conf.d/20-tokenizer.ini
PHP API	20190902
PHP Extension	20190902
Zend Extension	320190902
Zend Extension Build	API320190902,NTS
PHP Extension Build	API20190902,NTS
Debug Build	no

Also access the phpMyAdmin



Experiment 9

Laravel Installation on UBUNTU

Laravel is an open-source PHP web framework. It is mainly used for building PHP-based web applications. Laravel is suitable for both small-scale and enterprise-level application development. Its elegant syntax, advanced features, robust tools help simplify web application development. Laravel is highly scalable and has built-in support for distributed cache systems.

Step 1: Install Apache web server

Let's first install a webserver to host the Laravel application. You can either use Apache or Nginx web server. Here I am using an Apache web server.

To install apache2, type:

$ sudo apt install apache2

Once installed, Apache should be running. If it's not, for whatever reason, start it:

$ sudo systemctl start apache2

Then enable it to start on boot time.

$ sudo systemctl enable apache2

To verify the status of Apache, execute:

$ sudo systemctl status apache2

```
winnie@ubuntu20-04:~$
winnie@ubuntu20-04:~$ sudo systemctl status apache2
[sudo] password for winnie:
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2021-07-19 21:03:54 UTC; 4min 55s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 509 (apache2)
      Tasks: 7 (limit: 4713)
     Memory: 31.0M
     CGroup: /system.slice/apache2.service
             ├─ 509 /usr/sbin/apache2 -k start
             ├─ 560 /usr/sbin/apache2 -k start
```

Check the status of the Apache webserver

Step 2: Install PHP and additional PHP extensions

Laravel 8 requires PHP 7.3 or above. Thankfully, PHP 7.4 is available in Ubuntu repositories. So, install PHP and the following PHP extensions.

$ sudo apt install php libapache2-mod-php php-mbstring php-cli php-bcmath php-json php-xml php-zip php-pdo php-common php-tokenizer php-mysql

When the installation is complete, verify the PHP version.

$ php -v

```
winnie@ubuntu20-04:~$
winnie@ubuntu20-04:~$
winnie@ubuntu20-04:~$ php -v
PHP 7.4.3 (cli) (built: Jul  5 2021 15:13:35) ( NTS )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
    with Zend OPcache v7.4.3, Copyright (c), by Zend Technologies
winnie@ubuntu20-04:~$
```

Step 3: Create Database for Laravel Application

Next up, we will create a database for the Laravel application.

But first, we need to install a database server. Laravel supported database systems are MariaDB, MySQL, SQLite, Postgres, or SQL Server.

We will go with the MariaDB database engine.

$ sudo apt install mariadb-server

Once the database server is installed, log into the MariaDB prompt:

$ sudo mysql -u root -p

Once logged in create the database, database user, and grant all privileges to the database user.

CREATE DATABASE laravel_db;

CREATE USER 'laravel_user'@'localhost' IDENTIFIED BY 'secretpassword';

GRANT ALL ON laravel_db.* TO 'laravel_user'@'localhost';

FLUSH PRIVILEGES;

QUIT;

Step 4: Install Composer

Composer is a dependency package manager for PHP. It provides a framework for managing libraries and dependencies and required dependencies. To use Laravel, first install composer.

To download Composer, invoke the command shown.

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

This downloads the composer.phar file.

```
winnie@ubuntu20-04:~$
winnie@ubuntu20-04:~$ curl -sS https://getcomposer.org/installer | php
All settings correct for using Composer
Downloading...

Composer (version 2.1.3) successfully installed to: /home/winnie/composer.phar
Use it: php composer.phar

winnie@ubuntu20-04:~$
winnie@ubuntu20-04:~$ ls
composer.phar
winnie@ubuntu20-04:~$
```

Download Composer

Next, move the composer file to the /usr/local/bin path.

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

Assign execute permission:

$ sudo chmod +x /usr/local/bin/composer

Verify the Composer version installed:

$ composer –version

```
winnie@ubuntu20-04:~$
winnie@ubuntu20-04:~$
winnie@ubuntu20-04:~$ composer --version
Composer version 2.1.3 2021-06-09 16:31:20
winnie@ubuntu20-04:~$
winnie@ubuntu20-04:~$
winnie@ubuntu20-04:~$
```

Check composer version

Composer version 2.1.3 is installed.

Step 5: Install Laravel 8 on Ubuntu

With Composer installed, the next course of action is to install Laravel.

Navigate to the webroot directory, type:

$ cd /var/www/html

Now, install Laravel using the composer command, type:

$ sudo composer create-project laravel/laravel laravelapp

The command creates a new directory called laravelapp and installs all the files and directories for Laravel.

Change the ownership of the Laravel directory to the webserver user and also the permissions:

sudo chown -R www-data:www-data /var/www/html/laravelapp

sudo chmod -R 775 /var/www/html/laravelapp/storage

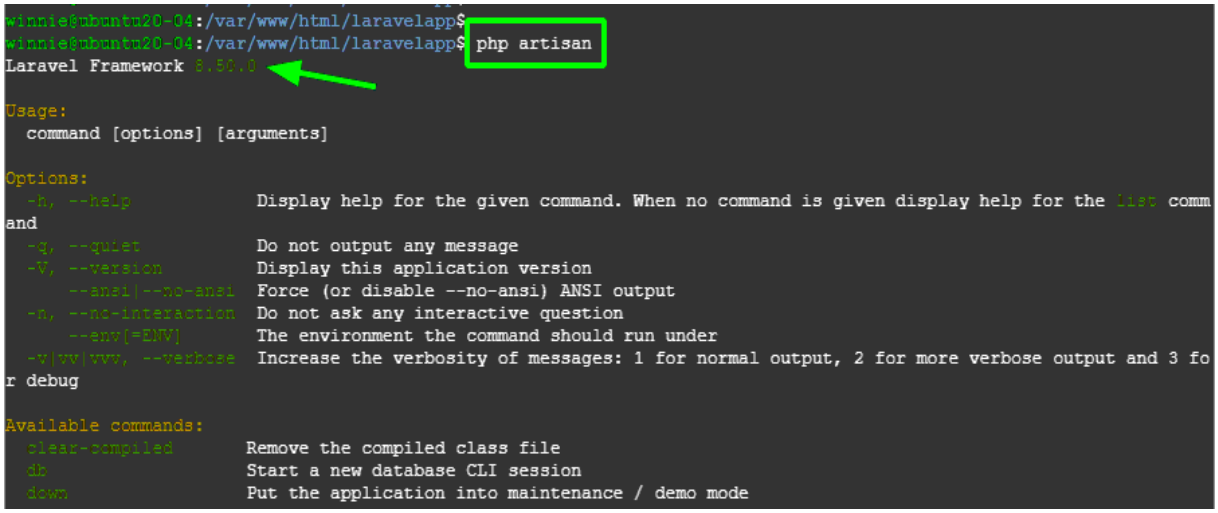
Feel free to replace laravelapp with a preferred directory name.

```
  - Installing phpunit/php-invoker (3.1.1): Extracting archive
  - Installing phpunit/php-file-iterator (3.0.5): Extracting archive
  - Installing theseer/tokenizer (1.2.0): Extracting archive
  - Installing sebastian/lines-of-code (1.0.3): Extracting archive
  - Installing sebastian/complexity (2.0.2): Extracting archive
  - Installing sebastian/code-unit-reverse-lookup (2.0.3): Extracting archive
  - Installing phpunit/php-code-coverage (9.2.6): Extracting archive
  - Installing doctrine/instantiator (1.4.0): Extracting archive
  - Installing phpspec/prophecy (1.13.0): Extracting archive
  - Installing phar-io/version (3.1.0): Extracting archive
  - Installing phar-io/manifest (2.0.1): Extracting archive
  - Installing myclabs/deep-copy (1.10.2): Extracting archive
  - Installing phpunit/phpunit (9.5.6): Extracting archive
74 package suggestions were added by new dependencies, use `composer suggest` to see details.
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
Discovered Package: facade/ignition
Discovered Package: fideloper/proxy
Discovered Package: fruitcake/laravel-cors
Discovered Package: laravel/sail
Discovered Package: laravel/tinker
Discovered Package: nesbot/carbon
Discovered Package: nunomaduro/collision
Package manifest generated successfully.
74 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
> @php artisan key:generate --ansi
Application key set successfully.
winnie@ubuntu20-04:/var/www/html$
```

Once the installation is done navigate to the installation directory and check the Laravel version.

$ cd laravelapp

$ php artisan



check Laravel version
Laravel Framework version 8.50.0 is installed.

Step 6: Configure Apache to serve Laravel site

Lastly, we need to set up the Apache webserver to host the Laravel site. For that to happen, we need to create a virtual host file.

$ sudo vim /etc/apache2/sites-available/laravel.conf

Next, past the content shown and replace the example.com ServerName directive with the FQDN or public IP of the server (Or private IP in case the server is on a LAN network).

<VirtualHost *:80>

ServerName example.com

ServerAdmin admin@example.com

DocumentRoot /var/www/html/laravelapp/public

<Directory /var/www/html/laravelapp>

AllowOverride All

</Directory>

ErrorLog ${APACHE_LOG_DIR}/error.log

CustomLog ${APACHE_LOG_DIR}/access.log combined

</VirtualHost>

Save the changes and exit the file. Next, enable the Laravel site and Apache rewrite module using these two commands.

$ sudo a2ensite laravel.conf

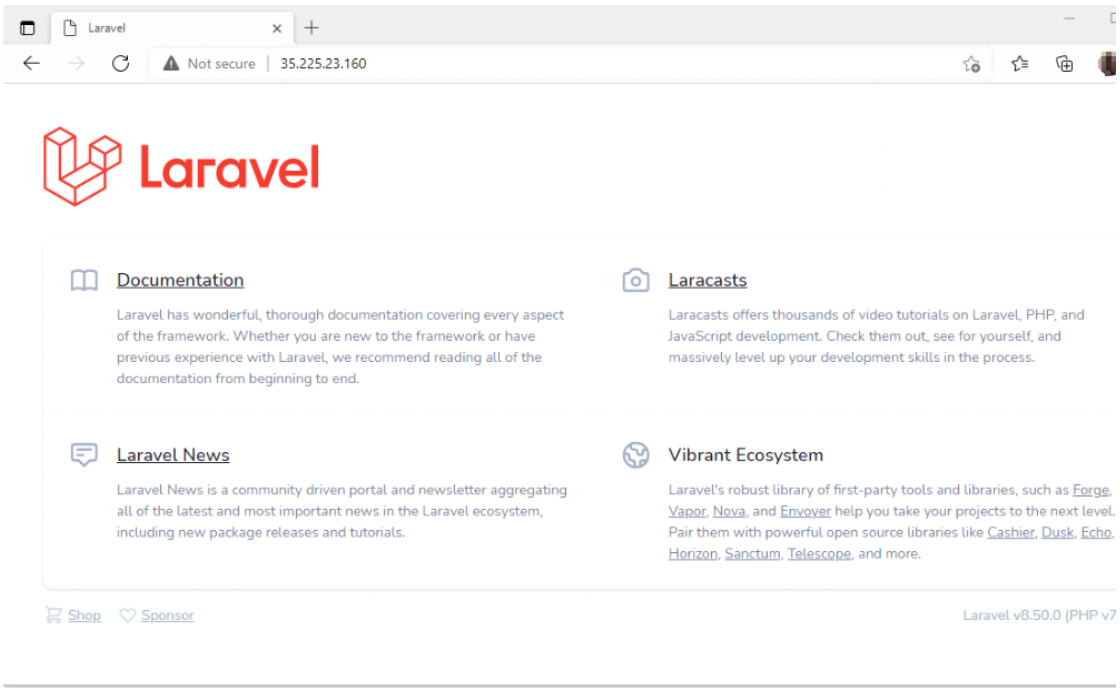
$ sudo a2enmod rewrite

To apply the changes, restart Apache.

$ sudo systemctl restart apache2

Step 7: Access Laravel from a browser

Finally, to access Laravel visit your server's FQDN or IP address. The default Laravel webpage will be displayed.



Default Laravel page

Experiment 10

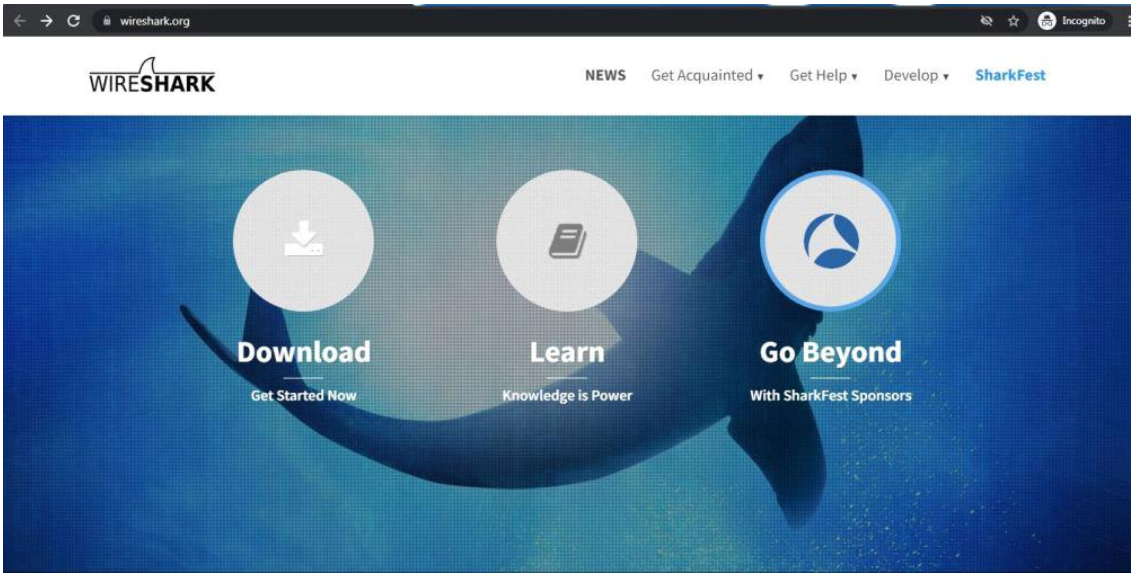
Wireshark Installation on Windows

Wireshark is software that is widely used in the analysis of data packets in a network. Wireshark is completely free and open source. This packet analyzer is used for a variety of purposes like troubleshooting networks, understanding communication between two systems, developing new protocols, etc. Wireshark is a cross-platform software, it can be run on Linux, windows, mac, and any other operating system.

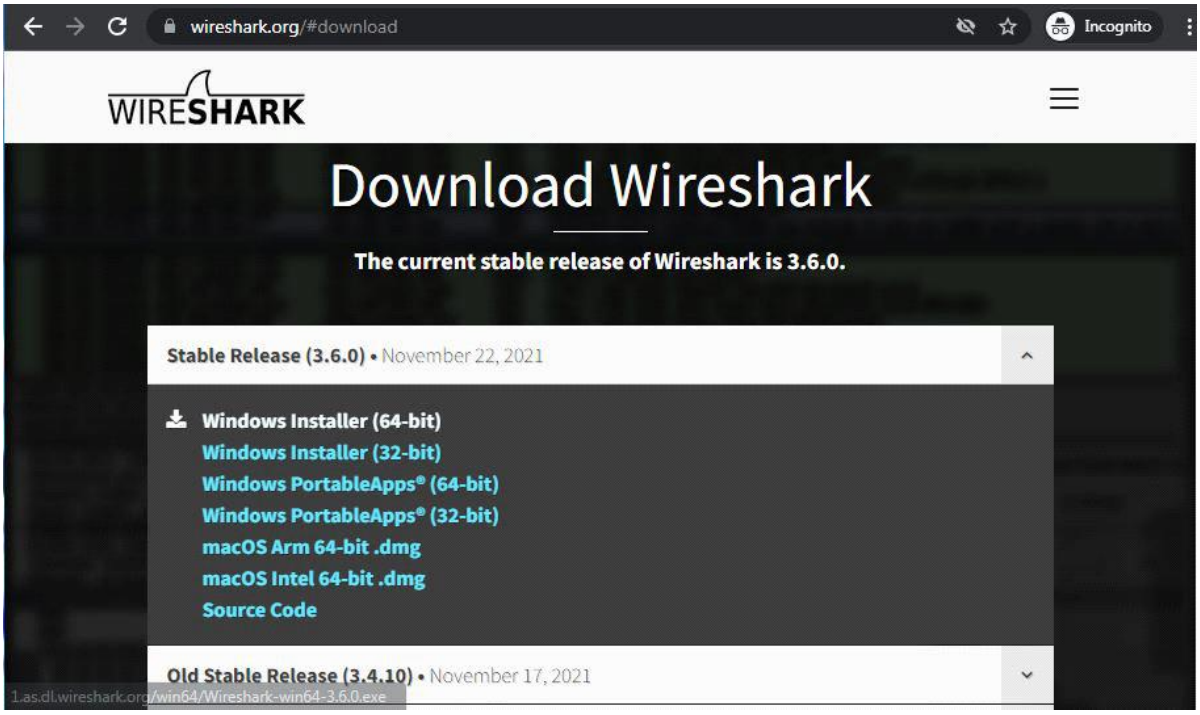
Installing Wireshark on Windows:

Follow the below steps to install Wireshark on Windows:

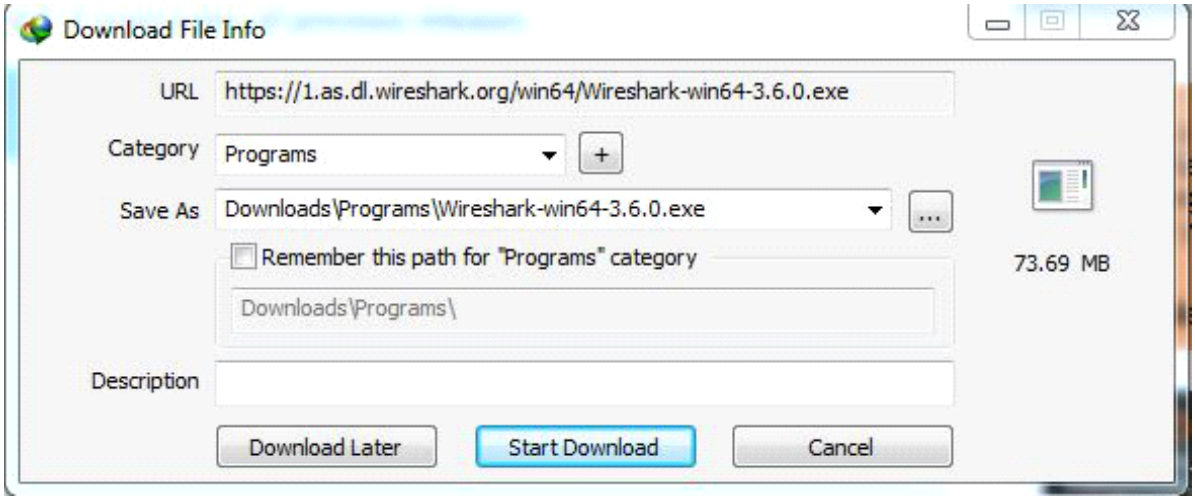
Step 1: Visit the official Wireshark website using any web browser.



Step 2: Click on Download, a new webpage will open with different installers of Wireshark.

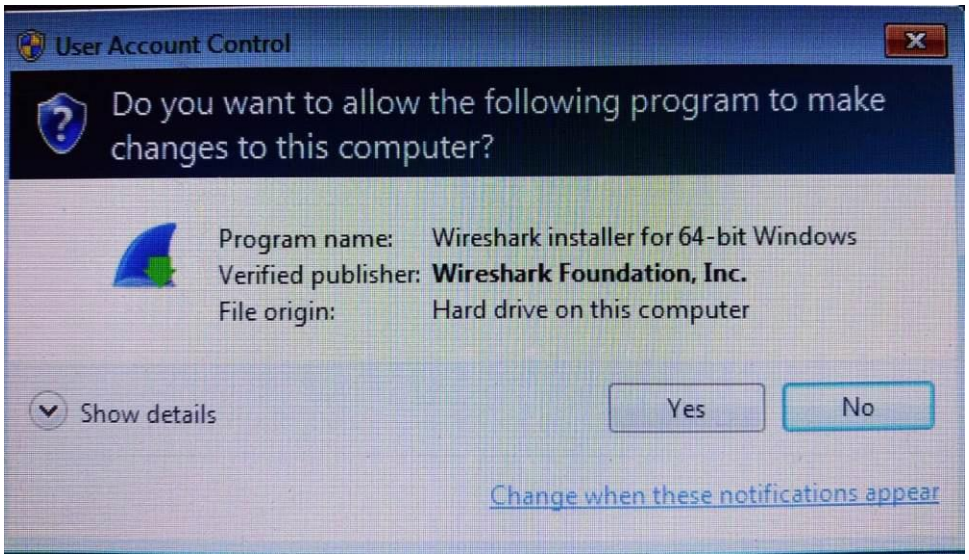


Step 3: Downloading of the executable file will start shortly. It is a small 73.69 MB file that will take some time.



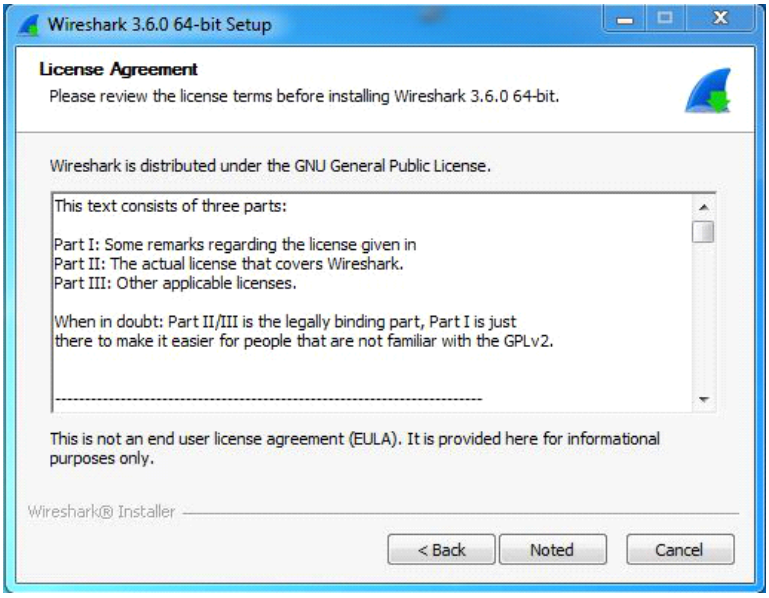
Step 4: Now check for the executable file in downloads in your system and run it.

Step 5: It will prompt confirmation to make changes to your system. Click on Yes.

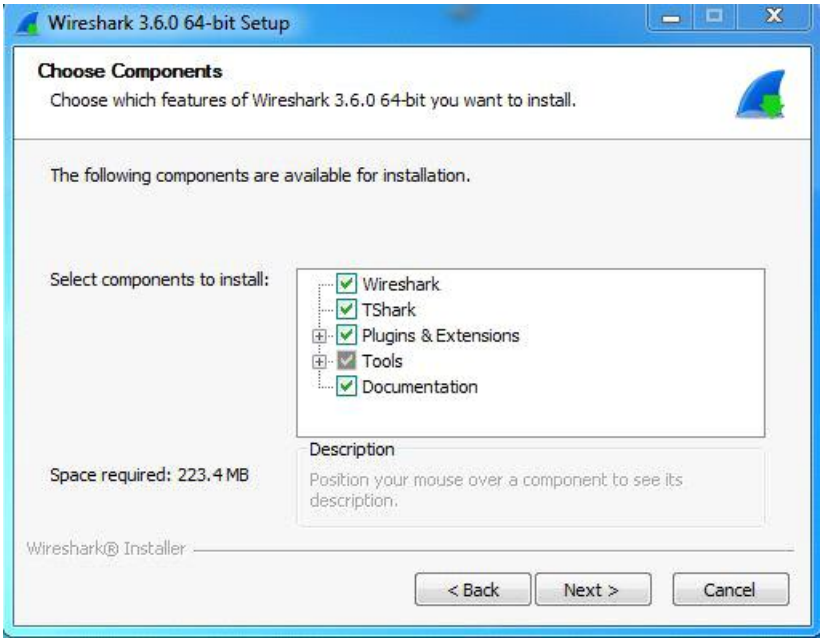


Step 6: Setup screen will appear, click on Next.

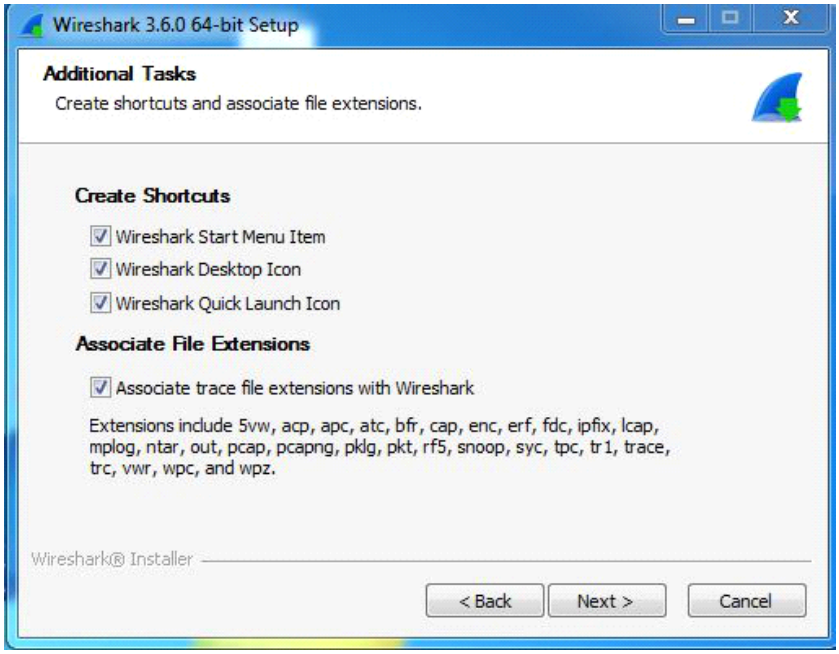
Step 7: The next screen will be of License Agreement, click on Noted.



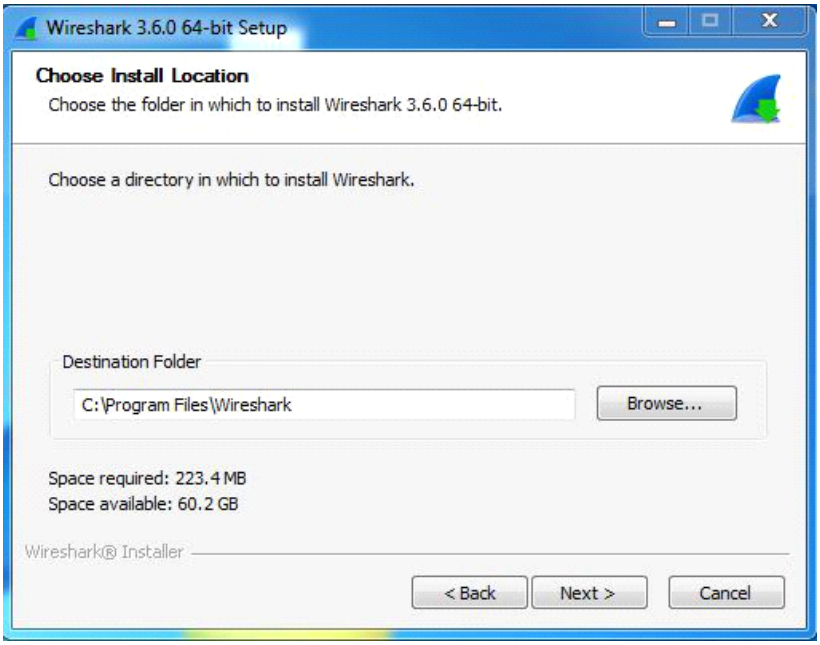
Step 8: This screen is for choosing components, all components are already marked so don't change anything just click on the Next button.



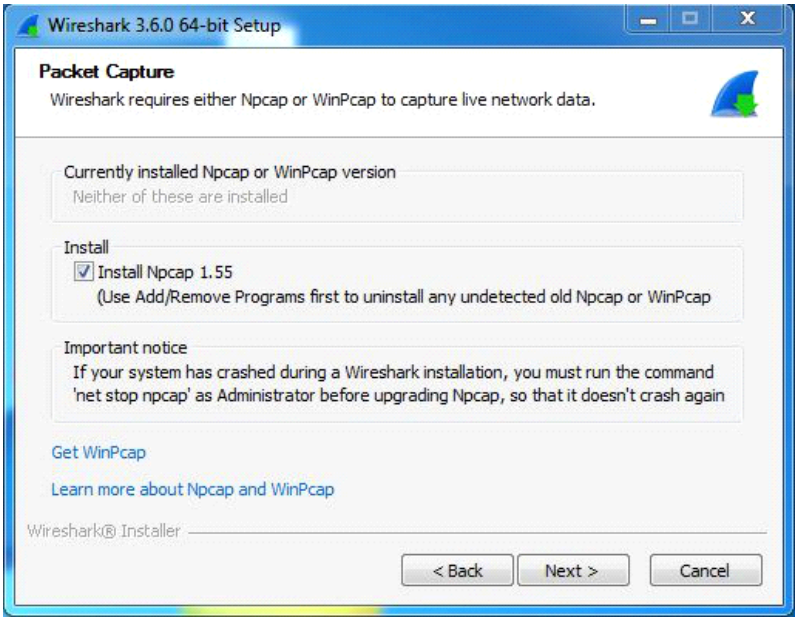
Step 9: This screen is of choosing shortcuts like start menu or desktop icon along with file extensions which can be intercepted by Wireshark, tick all boxes and click on Next button.



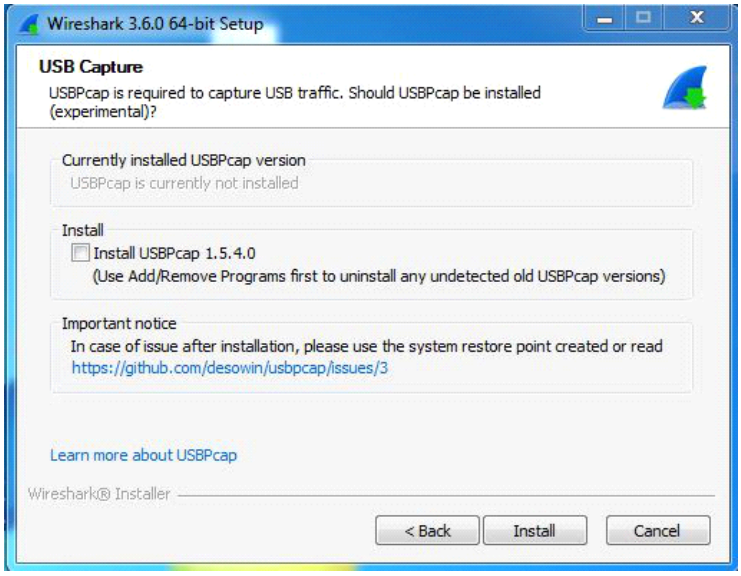
Step 10: The next screen will be of installing location so choose the drive which will have sufficient memory space for installation. It needed only a memory space of 223.4 MB.



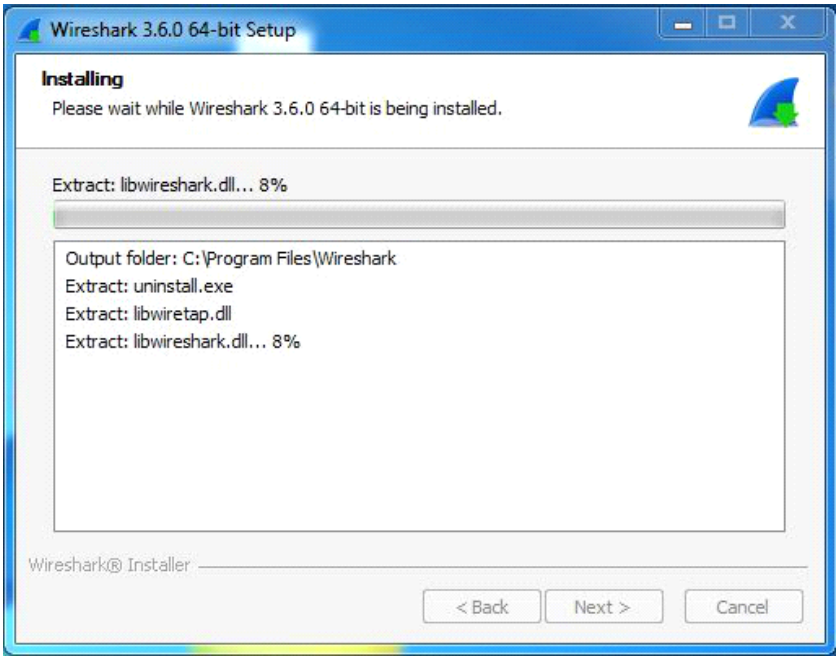
Step 11: Next screen has an option to install Npcap which is used with Wireshark to capture packets *pcap* means packet capture so the install option is already checked don't change anything and click the next button.



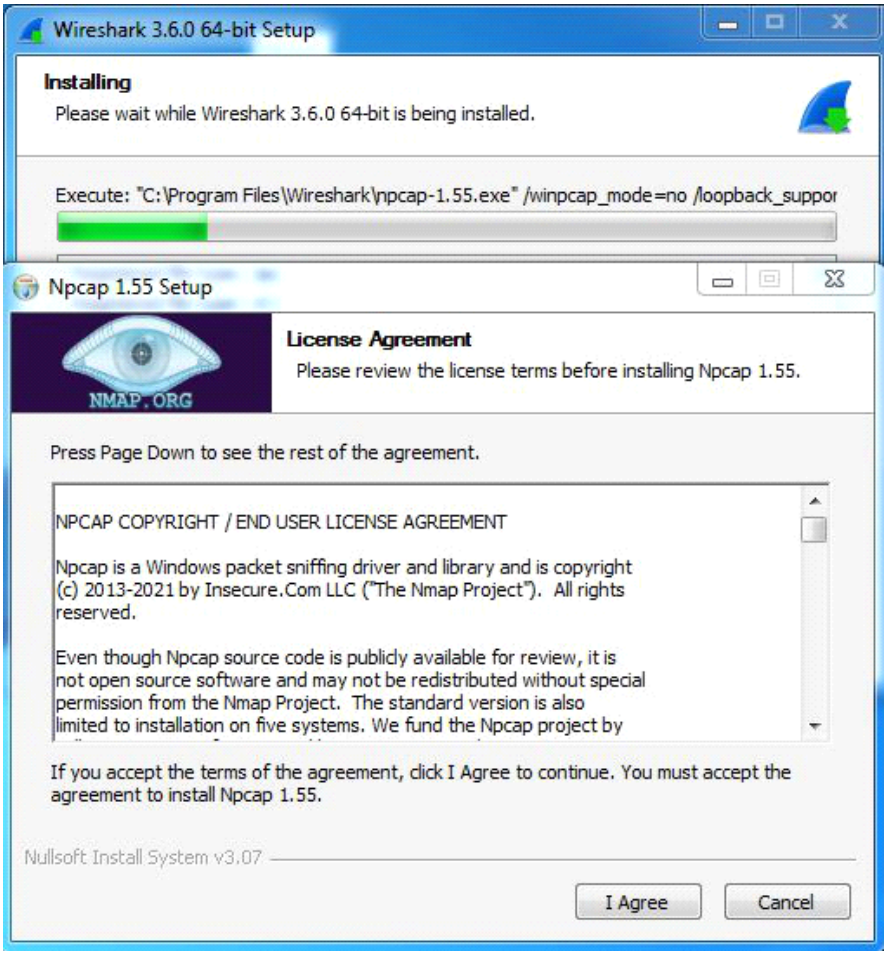
Step 12: Next screen is about USB network capturing so it is one's choice to use it or not, click on Install.



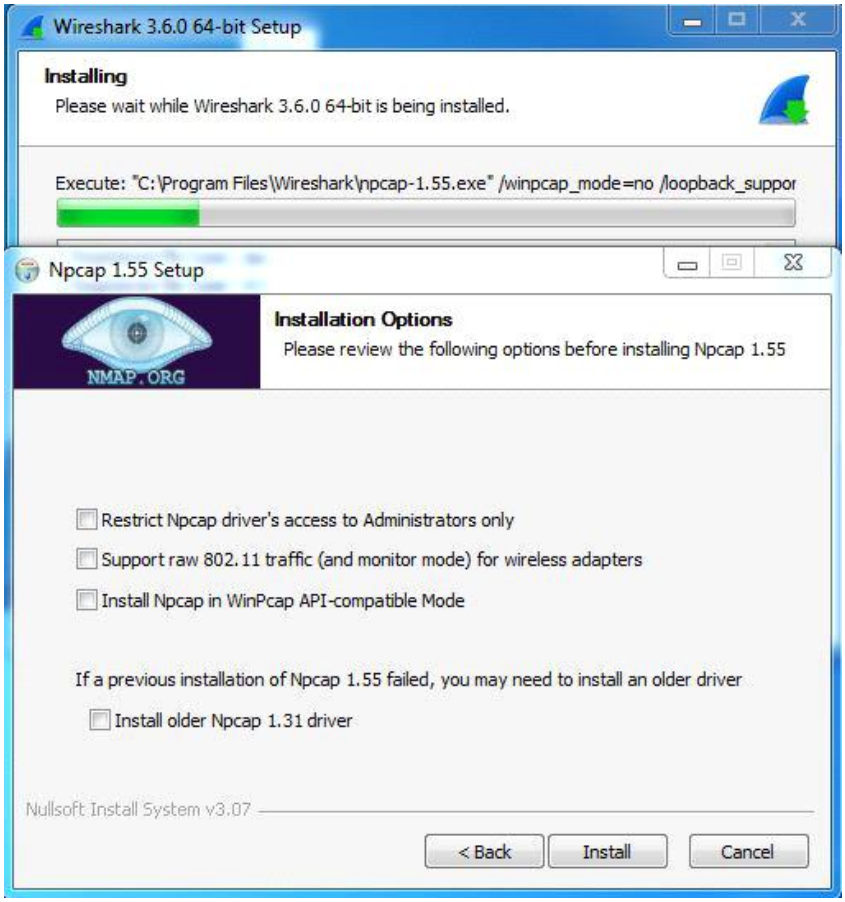
Step 13: After this installation process will start.



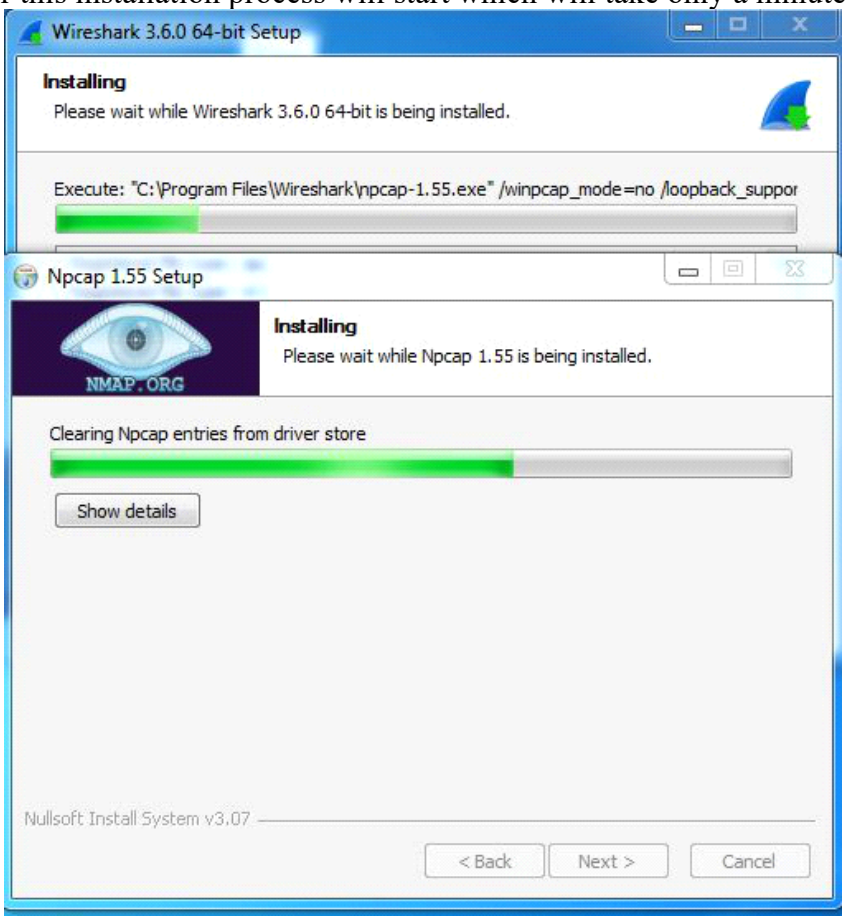
Step 14: This installation will prompt for Npcap installation as already checked so the license agreement of Npcap will appear to click on the *I Agree* button.



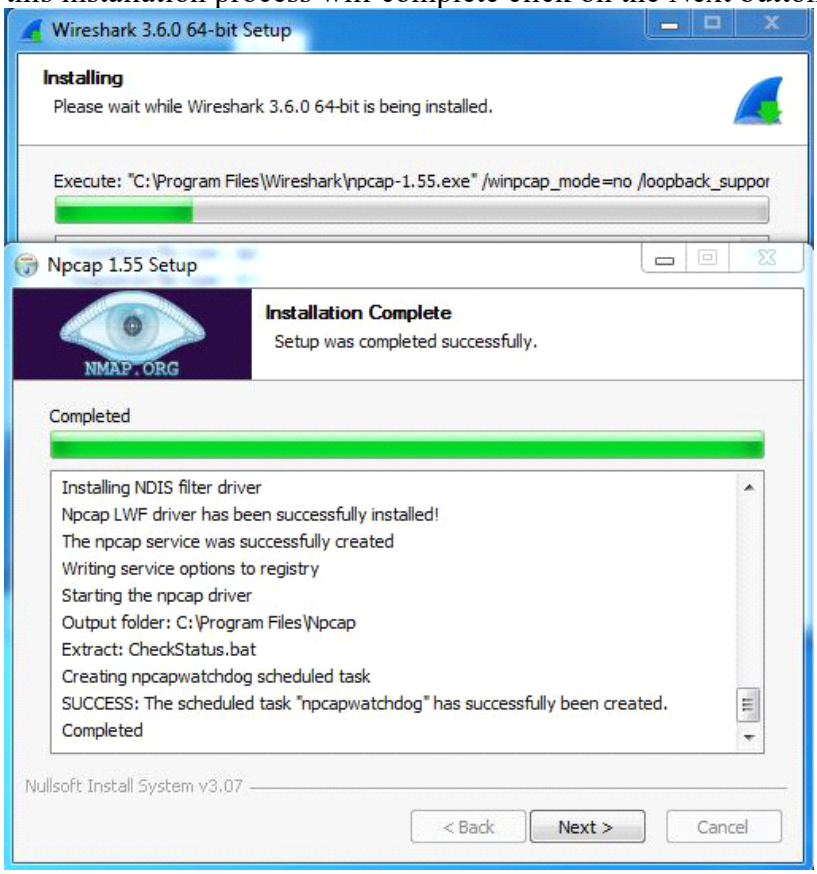
Step 15: Next screen is about different installing options of *npcap*, don't do anything click on Install.



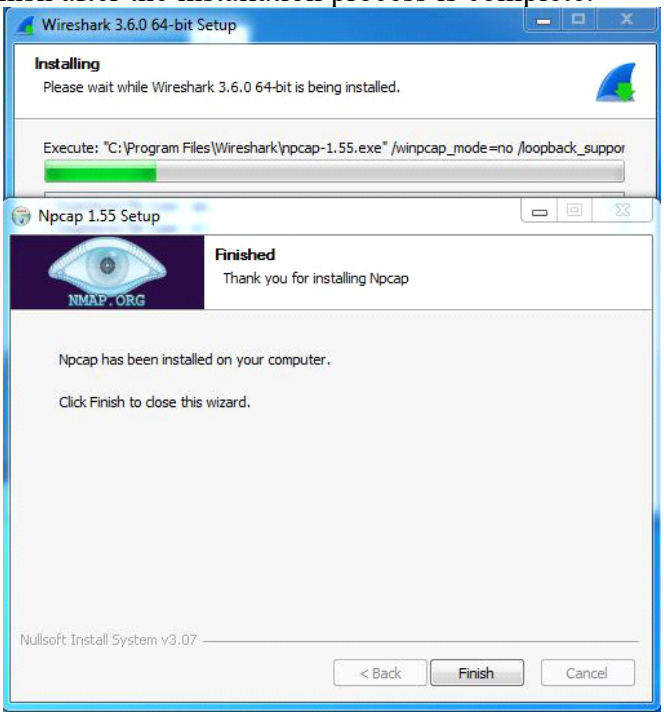
Step 16: After this installation process will start which will take only a minute.



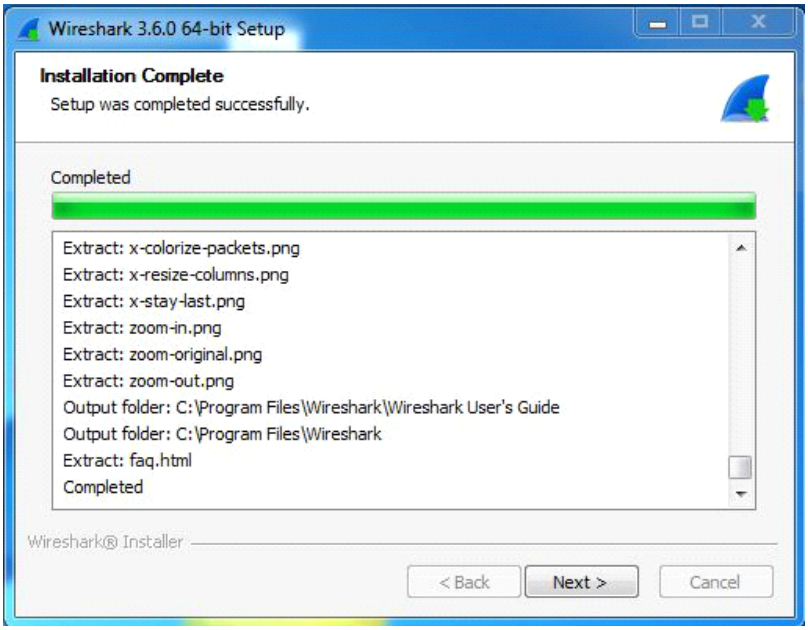
Step 17: After this installation process will complete click on the Next button.



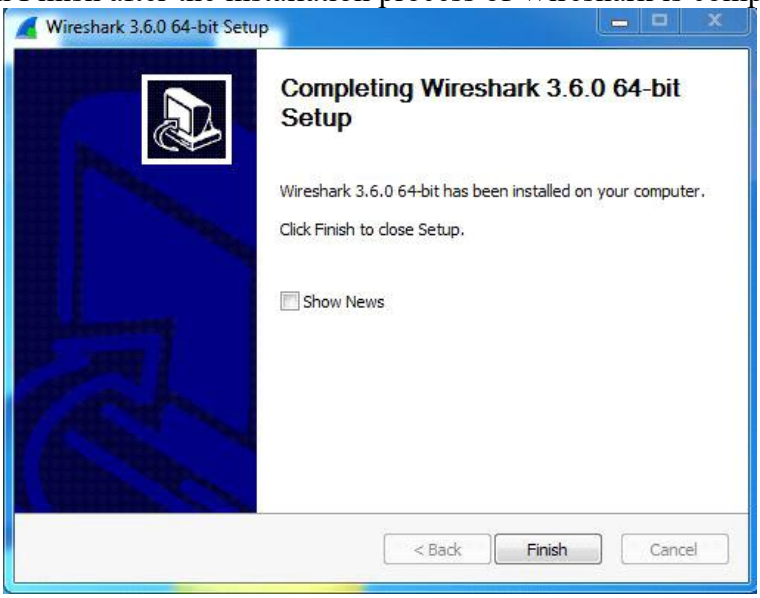
Step 18: Click on Finish after the installation process is complete.



Step 19: After this installation process of Wireshark will complete click on the Next button.

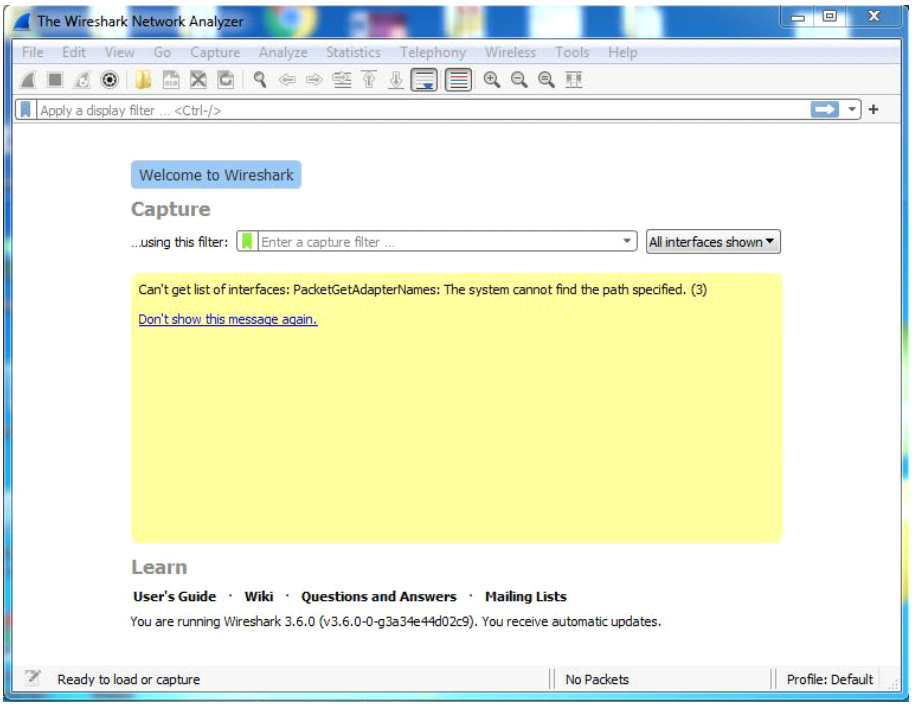


Step 20: Click on Finish after the installation process of Wireshark is complete.



Wireshark is successfully installed on the system and an icon is created on the desktop as shown below:

Now run the software and see the interface.



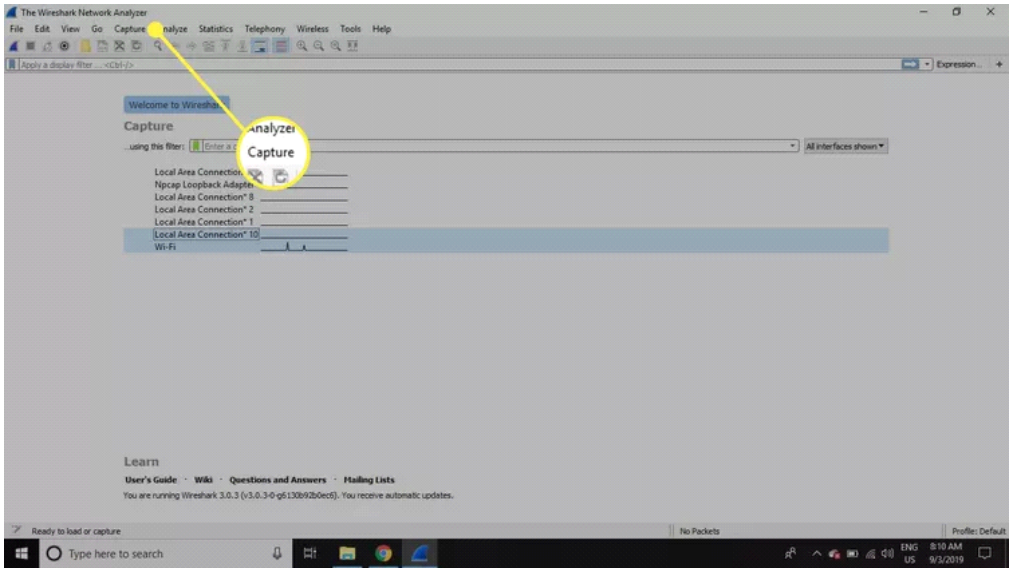
At this point, you have successfully installed Wireshark on your windows system.

How to Capture Data Packets With Wireshark

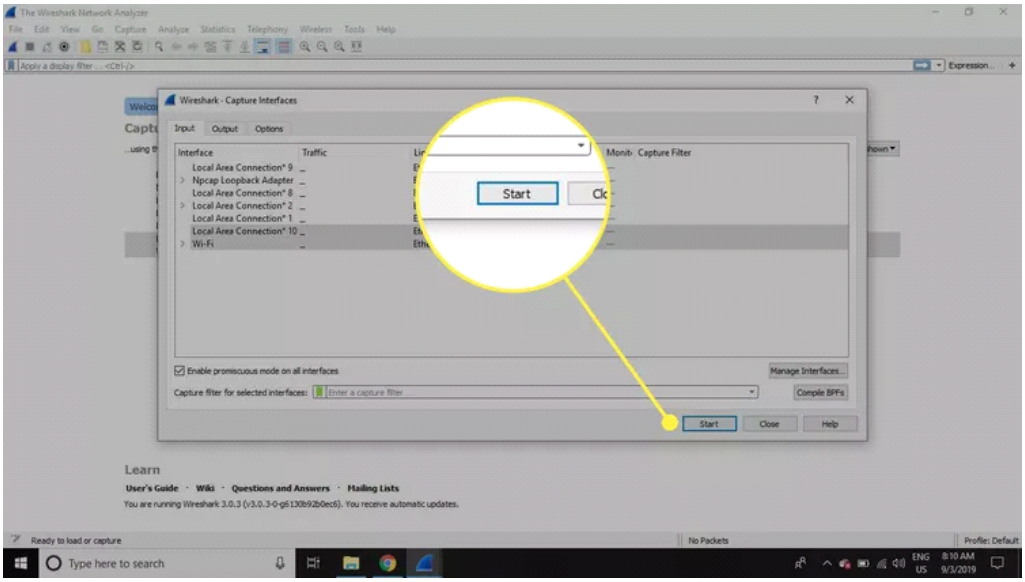
When you launch Wireshark, a welcome screen lists the available network connections on your current device. Displayed to the right of each is an EKG-style line graph that represents live traffic on that network.

To begin capturing packets with Wireshark:

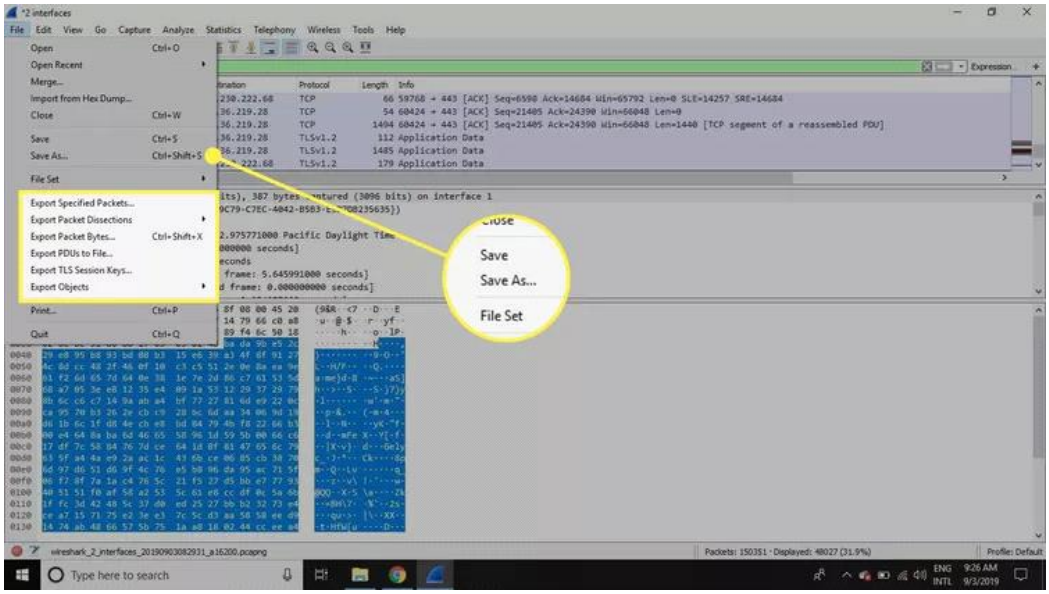
Select one or more of networks, go to the menu bar, then select Capture.



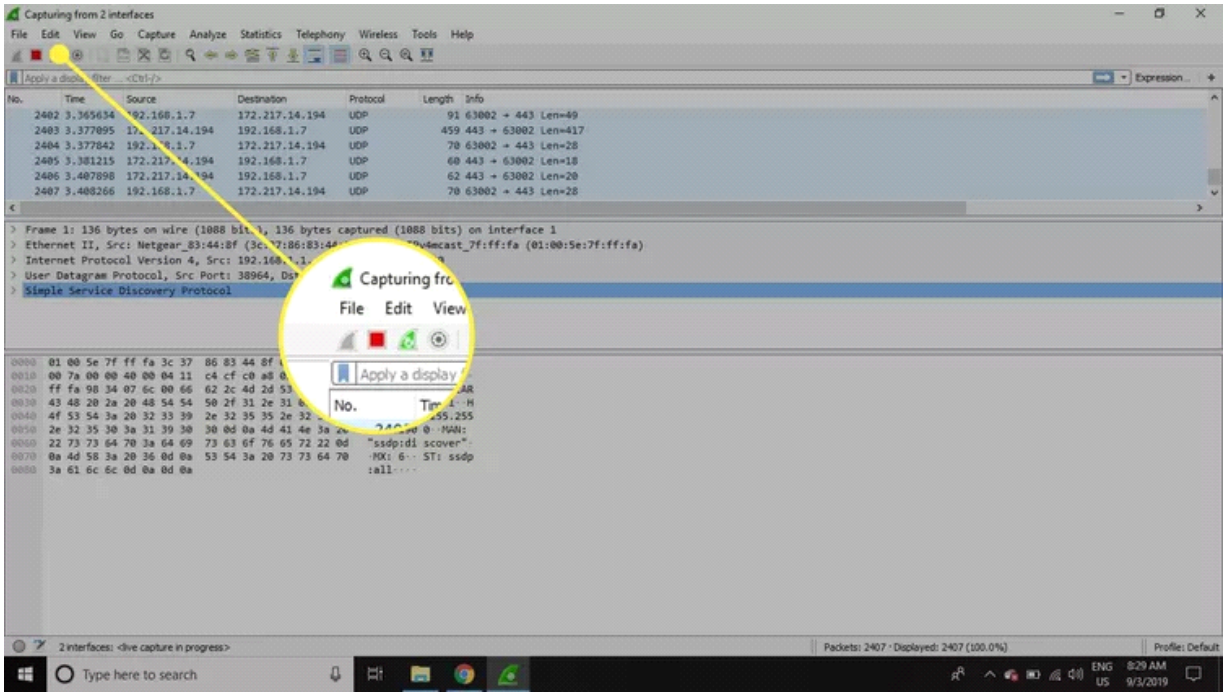
2. In the Wireshark Capture Interfaces window, select Start.



3. Select File > Save As or choose an Export option to record the capture.



4. To stop capturing, press Ctrl+E. Or, go to the Wireshark toolbar and select the red Stop button that's located next to the shark fin.



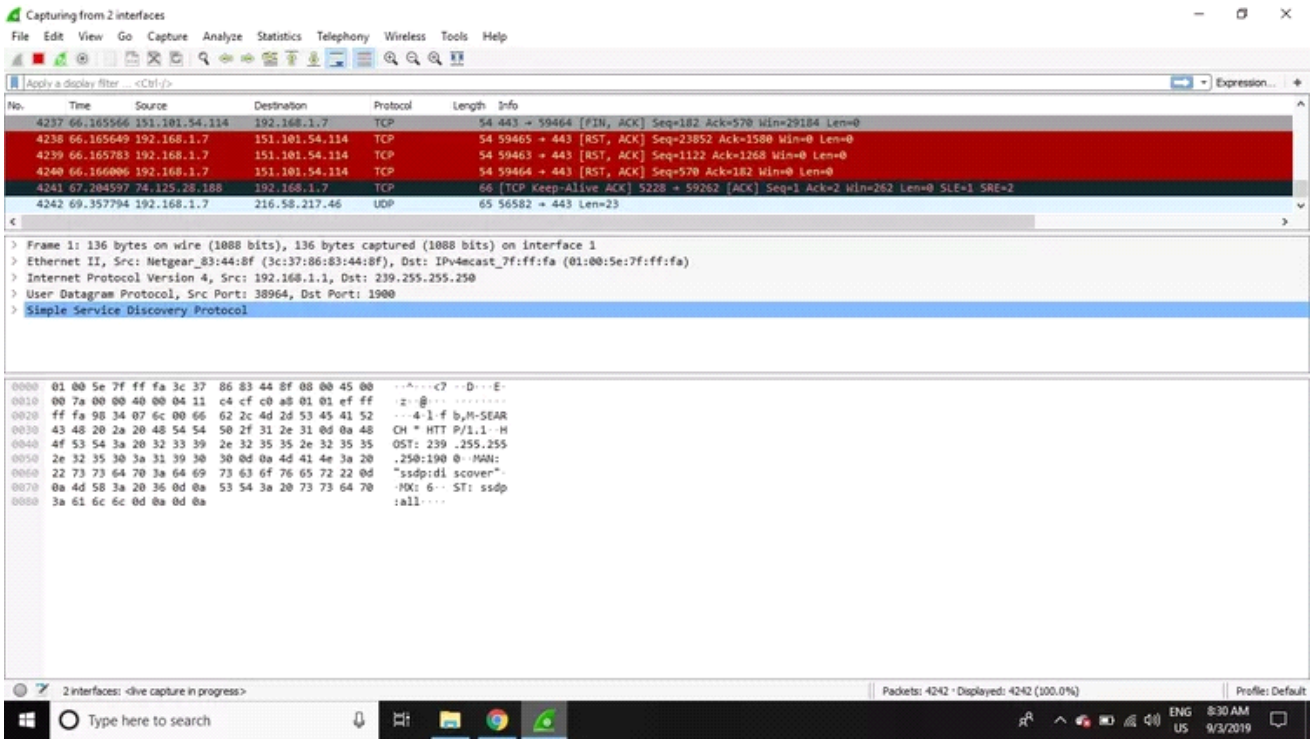
How to View and Analyze Packet Contents

The captured data interface contains three main sections:

The packet list pane (the top section)

The packet details pane (the middle section)

The packet bytes pane (the bottom section)


Capturing from 2 interfaces
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
No. Time Source Destination Protocol Length Info
4237 66.165566 151.101.54.114 192.168.1.7 TCP 54 443 → 59464 [FIN, ACK] Seq=182 Ack=570 Win=29184 Len=0
4238 66.165649 192.168.1.7 151.101.54.114 TCP 54 59465 → 443 [RST, ACK] Seq=23852 Ack=1580 Win=0 Len=0
4239 66.165783 192.168.1.7 151.101.54.114 TCP 54 59463 → 443 [RST, ACK] Seq=1122 Ack=1268 Win=0 Len=0
4240 66.166006 192.168.1.7 151.101.54.114 TCP 54 59464 → 443 [RST, ACK] Seq=570 Ack=182 Win=0 Len=0
4241 67.204597 74.125.28.188 192.168.1.7 TCP 66 [TCP Keep-Alive ACK] 5228 → 59262 [ACK] Seq=1 Ack=2 Win=262 Len=0 SLE=1 SRE=2
4242 69.357794 192.168.1.7 216.58.217.46 UDP 65 56582 → 443 Len=23
Frame 1: 136 bytes on wire (1088 bits), 136 bytes captured (1088 bits) on interface 1
Ethernet II, Src: Netgear_83:44:8f (3c:37:86:83:44:8f), Dst: IPv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)
Internet Protocol Version 4, Src: 192.168.1.1, Dst: 239.255.255.250
User Datagram Protocol, Src Port: 38964, Dst Port: 1900
Simple Service Discovery Protocol
0000 01 00 5e 7f ff fa 3c 37 86 83 44 8f 08 00 45 00 ··^···c7 ··D···E·
0010 00 7a 00 00 40 00 04 11 c4 cf c0 a8 01 01 ef ff ·z··@··· ····
0020 ff fa 98 34 07 6c 00 66 62 2c 4d 2d 53 45 41 52 ···4·l·f b,M-SEAR
0030 43 48 20 2a 20 48 54 54 50 2f 31 2e 31 0d 0a 48 CH * HTT P/1.1··H
0040 4f 53 54 3a 20 32 33 39 2e 32 35 35 2e 32 35 35 OST: 239 .255.255
0050 2e 32 35 30 3a 31 39 30 30 0d 0a 4d 41 4e 3a 20 .250:190 0··MAN:
0060 22 73 73 64 70 3a 64 69 73 63 6f 76 65 72 22 0d "ssdp:di scover"·
0070 0a 4d 58 3a 20 36 0d 0a 53 54 3a 20 73 73 64 70 ·MX: 6·· ST: ssdp
0080 3a 61 6c 6c 0d 0a 0d 0a :all····
2 interfaces: <live capture in progress>
Packets: 4242 · Displayed: 4242 (100.0%)
Profile: Default
Type here to search
ENG US
8:30 AM 9/3/2019