MASTER OF COMPUTER APPLICATIONS

PRACTICAL RECORD WORK

ON

20MCA136
NETWORKING AND SYSTEM ADMINISTRATION LAB

Submitted

By

MURSHID TP
(Reg. No. : VDA20MCA-2041)

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

OCTOBER - 2021

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

CERTIFICATE

Certified that this is a bonafide record of the practical work on the course 20MCA136 NETWORKING AND SYSTEM ADMINISTRATION LAB done by Mr.**MURSHID TP**(Reg.No:**VDA20MCA-2041)** Second Semester MCA student of Department of Computer Applications at College of Engineering Vatakara in the partial fulfillment for the award of the degree of Master of Computer Applications (MCA) of APJ Abdul Kalam Technological University (KTU).

(Mrs.Nidhina)
(Mrs.Anagha)
FACULTY-IN-CHARGE **HEAD OF THE DEPARTMENT**

CEV
4/10/2021

EXAMINERS:

INDEX

SL.NO	PROGRAMS	PAGE NO	REMARKS
1	INTRODUCTION TO COMPUTER HARDWARE	1	
2	VIM AND BASIC LINUX COMMANDS	34	
3	FILE SYSTEM HEIRARCHY AND FILE PERMISSIONS	53	
4	BASIC OF SHELL SCRIPTING	74	
5	LAMP INSTALLATION	85	
6	LARAVEL INSTALLATION	93	
7	INSTALL SOFTWARE FROM SOURCE CODE	102	
8	NETWORKING TOOLS AND CONFIGURATION	107	
9	TCPDUMP AND WIRESHARK	124	
10	HYPERVISORS AND DOCKER INSTALLATION	133	
11	ANSIBLE AND PLAYBOOK	142	

Experiment No : 1

Aim: Introduction to Computer hardware: Physical identification of major components of a computer system such as mother board, RAM modules, daughter cards, bus slots, SMPS, internal storage devices,interfacing ports. Specification of desktop and server class computers.
Installation of common operating systems for desktop and server use.

Introduction:

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

The term hardware refers to mechanical device that makes up computer. Computer hardware consists of interconnected electronic devices that we can use to control computer's operation, input and output. Examples of hardware are CPU, keyboard, mouse, hard disk, etc.

1.Motherboard

A motherboard provides connectivity between the hardware components of a computer, like the processor (CPU), memory (RAM), hard drive, and video card. There are multiple types of motherboards, designed to fit different types and sizes of computers.

Each type of motherboard is designed to work with specific types of processors and memory, so they don't work with every processor and type of memory. However, hard drives are mostly universal and work with the majority of motherboards, regardless of the type or brand.

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

NORTH BRIDGE : Alternatively referred to as the PAC (PCI/AGP Controller) and nb, the Northbridge is an integrated circuit responsible for communications between the CPU interface, AGP, and the memory.The northbridge is usually slightly larger than the southbridge, and is positioned closer to the CPU and memory.

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

SOUTH BRIDGE : The southbridge is an IC on the motherboard responsible for the hard drive controller, I/O controller and integrated hardware. Integrated hardware can include the sound card and video card if on the motherboard, USB, PCI, ISA, IDE, BIOS, and Ethernet.
Although the southbridge handles most of the I/O devices, less prominent input/output devices, such as a serial port, keyboard, and non-USB mouse are handled by the SIO (super input/output).

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

BIOS : Short for Basic Input/Output System, the BIOS is a ROM chip found on motherboards that allows you to access and set up your computer system at the most basic level.The BIOS includes instructions on how to load basic computer hardware. It also includes a test referred to as a POST (Power-On Self-Test) that helps verify the computer meets requirements to boot up properly. If the computer does not pass the POST, you hear a combination of beeps indicating what is malfunctioning in the computer.

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

Bootstrap Loader : Locate the operating system. If a capable operating system is located, the BIOS will pass control to it.

2. RAM (RANDOM ACCESS MEMORY)

Alternatively referred to as main memory, primary memory, or system memory, RAM (random-access memory) is a hardware device that allows information to be stored and retrieved on a computer. RAM is usually associated with DRAM, which is a type of memory module. Because data is accessed randomly instead of sequentially like it is on a CD or hard drive, access times are much faster. However, unlike ROM, RAM is a volatilememory and requires power to keep the data accessible. If the computer is turned off, all data contained in RAM is lost. Access time in RAM is independent of the address, that is, each storage location inside the memory is as easy to reach as other locations and takes the same amount of time. Data in the RAM can be accessed randomly but it is very expensive.

3.ROM (READ ONLY MEMORY)

Read-Only Memory (ROM), is a type of electronic storage that comes built in to a device during manufacturing. You'll find ROM chips in computers and many other types of electronic products; VCRs, game consoles, and car radios all use ROM to complete their functions smoothly. ROM chips come built into an external unit – like flash drives and other auxiliary memory devices – or installed into the device's hardware on a removable chip. Non-volatile memory like ROM remains viable even without a power supply.

4.Daughter board

A daughterboard is type of circuit board that plugs in or is attached to the motherboard or similar expansion card to extend its features and services. A daughterboard complements the existing functionality of a motherboard or an expansion card.

5.Bus Slots

Alternatively known as a expansion port, an expansion slot is a connection or port inside a computer on the motherboard or riser card. It provides an installation point for a hardware expansion card to be connected.
For example, if you wanted to install a new video card in the computer, you'd purchase a video expansion card and install that card into the compatible expansion slot.An expansion slot is a socket on the motherboard that is used to insert an expansion card (or circuit board), which provides additional features to a computer such as video, sound, advanced graphics, Ethernet or memory.

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

6.SMPS

SMPS stands for Switched-Mode Power Supply. It is an electronic power supply that uses a switching regulator to convert electrical power efficiently. It is also known as Switching Mode Power Supply. It is power supply unit (PSU) generally used in computers to convert the voltage into the computer acceptable range.

This device has the power handling electronic components that converts electrical power efficiently. Switched Mode Power Supply uses a great power conversion technique to reduce overall power loss.

7. Internal Storage Devices

Some storage devices are classed as 'internal' which means they are inside the computer case.Most computers have some form of internal storage. The most common type of internal storage is the hard disk.

At the most basic level, internal storage is needed to hold the operating system so that the computer is able to access the input and output devices.It will also be used to store the applications software that you use and more than likely, the original copies of your data files.

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

The main disadvantage of internal storage is that when the hard disk fails (and it will), all the data and applications may be lost.This can be avoided to some extent by using more than one hard disk within the machine. Each hard disk has a copy of all the data, so if one fails the other can carry on. This is called a RAID array. An alternative is to use external drives for backup.

8.Interfacing ports

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

Characteristics of Ports

A port has the following characteristics −

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

Serial Port

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

Parallel Port

•Used for scanners and printers

•Also called printer port

•25 pin model

•IEEE 1284-compliant Centronics port

PS/2 Port

•Used for old computer keyboard and mouse

•Also called mouse port

•Most of the old computers provide two PS/2 port, each for the mouse and keyboard

•IEEE 1284-compliant Centronics port

Universal Serial Bus (or USB) Port

•It can connect all kinds of external USB devices such as external hard disk, printer, scanner, mouse, keyboard, etc.

•It was introduced in 1997.

•Most of the computers provide two USB ports as minimum.

•Data travels at 12 megabits per seconds.

•USB compliant devices can get power from a USB port.

VGA Port

•Connects monitor to a computer's video card.

•It has 15 holes.

•Similar to the serial port connector. However, serial port connector has pins, VGA port has holes.

Power Connector

•Three-pronged plug.

•Connects to the computer's power cable that plugs into a power bar or wall socket.

Firewire Port

•Transfers large amount of data at very fast speed.

•Connects camcorders and video equipment to the computer.

•Data travels at 400 to 800 megabits per seconds.

•Invented by Apple.

•It has three variants: 4-Pin FireWire 400 connector, 6-Pin FireWire 400 connector, and 9-Pin FireWire 800 connector.

Modem Port

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

Ethernet Port

•Connects to a network and high speed Internet.

•Connects the network cable to a computer.

•This port resides on an Ethernet Card.

•Data travels at 10 megabits to 1000 megabits per seconds depending upon the network bandwidth.

Game Port

- Connect a joystick to a PC
- Now replaced by USB

Digital Video Interface, DVI port

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

System specifications

A system specification describes the operational and performance requirements of a system, such as a computer.System specifications help to define the operational and performance guidelines for a system.These are the necessary specifications our computer must have in order to use the software or hardware.

Factors affect the computer performance

In broad terms, the performance of a computer depends on four factors: the speed and architecture of its processor or "central processing unit" (CPU), how much random access memory (RAM) it has, its graphics system, and its internal hard drive speed and capacity. Also of importance to most users will be the specification of its Internet connection. Most computer users and in particular those working with a lot of photographs, music files or videos should also think about the most suitable storage devices they will need in order to keep and back up all of their valuable data.

Processor speed and architecture

The architecture of a processor is the most important factor to determine its performance, and refers to its basic design and complexity.The speed of a computer's processor chip (technically known as its "clock speed") in measured in gigahertz (GHz), with the fastest modern processors currently running at up to 4.7GHz. However, for most computing tasks, including web browsing, sending e-mails, word processing and spreadsheet work any processor running at 1GHz or more remains perfectly sufficient.

Cache is a form of very fast memory integrated into the processor chip, and used to store up instructions (work for the processor) so that it has to slow down as little as possible between tasks.For applications such as video editing, 3D graphics work and (for the majority of "power users") playing computer games, higher processor speeds is highly required. CPU performance is now determined by far more than raw speed alone. Intel made this very clear when it introduced its system of processor numbers. These provide an indication of a processor's "architecture", "cache" and "front side bus (FSB) speed" in addition to its clock speed.
Front side bus (FSB) speed is a measure of how fast a microprocessor communicates with the computer's main circuit board (or "motherboard") into which it is physically connected. Again, the higher the measure the better for overall performance, with FSB speeds currently ranging from 533MHz (still perfectly sufficient for the vast majority of applications) up to 1600 MHz.

The architecture of a processor is the most important factor to determine its performance, and refers to its basic design and complexity. Some processors are simply more sophisticated than others, with Intel (for example) producing "basic" processors called Celerons and Pentiums, as well as more powerful

processors under its "Core" processor family. The later include the Core 2, Core i3, Core i5 and Core i7, with the last of these being the most powerful.

In addition to clock speed and architecture, a processor's cache and front side bus (FSB) speed also determine a computer's overall power. Cache is a form of very fast memory integrated into the processor chip, and used to store up instructions (work for the processor) so that it has to slow down as little as possible between tasks. Cache is measured in megabytes (MB), with (for example) low-end Celeron processors having as little as 0.25MB of cache (256KB), and high-end Itaniums having up to 24MB. The simple message is, the more cache the better- though high levels of cache still come at a very significant price.

Random Access Memory (RAM)

The part of the computer in which information is stored temporarily when a program is being used.
RAM is measured in megabytes (MB) and gigabytes (GB), as detailed on the storage page.To a large extent, the more RAM a computer has the faster and more effectively it will operate. Computers with little RAM have to keep moving data to and from their hard disks in order to keep running. This tends to make them not just slow in general, but more annoyingly intermittently sluggish.
Just how much RAM a computer needs depends on the software it is required to run effectively. A computer running Windows XP will usually function quite happily with 1GB of RAM, whereas twice this amount (ie 2GB) is the realistic minimum for computers running Windows 7.

Graphics system

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

A computer's graphics system determines how well it can work with visual output. Graphics systems can either be integrated into a computer's motherboard, or plugged into the motherboard as a separate "video card". Graphics systems integrated into the motherboard (also known as "onboard graphics") are now quite powerful, and sufficient for handling the requirements of most software applications aside from games playing, 3D modeling, and some forms of video editing.

Any form of modern computer graphics system can now display high-resolution color images on a standard-sized display screen (ie any monitor up to about 19" in size). The more sophisticated graphics cards now determines how well a computer can handle the playback of high definition video, as well as the speed and quality at which 3D scenes can be rendered. Another key feature of separate graphics cards is that most of them now allow more than one display screen to be connected to a computer. Others also permit the recording of video.

Hard Drive Speed and Capacity

A part of the computer that is used for storing computer data and that contains one or more hard disks.

Today 40GB is an absolute minimum hard drive size for a new computer running Windows 7, with a far larger capacity being recommended in any situation where more than office software is going to be installed. Where a computer will frequently be used to edit video, a second internal hard disk dedicated only to video storage is highly recommended for stable operation. Indeed, for professional video editing using a program like Premiere Pro CS5, Adobe now recommend that a PC has at least three internal hard disks (one for the operating system and programs, one for video project files, and one for video media).

Two key factors that determine the speed of traditional, spinning hard disks

1. rotational velocity of the physical disk itself.
2. interface used to connect it to the computer's motherboard.

Three types of interface

1. Serial Advance Technology Attachment (SATA) the most modern and now pretty much the norm on new PCs

2. Integrated Device Electronics (IDE) (also known as UDMA)- a slower and older form of interface

3. Small Computer System Interface (SCSI)-the oldest but in it most modern variant is still the fastest disk interface standard.

Servers

In computing, a server is a piece of computer hardware or software (computer program) that provides functionality for other programs or devices, called "clients". This archetecture is called the client–server model. Servers can provide various functionalities, often called "services", such as sharing data or resources among multiple clients, or performing computation for a client. A single server can serve multiple clients, and a single client can use multiple servers. A client process may run on the same device or may connect over a network to a server on a different device.On a network, a server is known as **host**.The devices are connecting to the server known as **clients**.
A server is a software or hardware device that accepts and responds to requests made over a network. The device that makes the request, and receives a response from the server, is called a client. On the Internet, the term "server" commonly refers to the computer system that receives requests for a web files and sends those files to the client.

Servers manage network resources. For example, a user may set up a server to control access to a network, send/receive e-mail, manage print jobs, or host a website. They are also proficient at performing intense calculations. Some servers are committed to a specific task, often referred to as dedicated. However, many servers today are shared servers that take on the responsibility of e-mail, DNS, FTP, and even multiple websites in the case of a web server.

Server Working:

Every time you use the internet you are accessing a server. When you enter a URL into a browser your computer communicates with the server hosting that website and pulls the data onto your machine.

This is a simplistic view of how the process works

- You enter a URL and your web browser requests a web page
- The web browser requests a full URL for the site it wants to display
- This information is sent to the server
- The web server finds and builds all the data needed to display the site (this is why some sites load quicker than others)
- Your web browser receives the data and displays the website to you

Types of Servers

Web server

A web server offers web pages or other content to the web browser by loading the information from a disc and transfer files by using a network to the user's web browser. It is used by a computer or collection of computers to provide content to several users over the internet. This exchange was done with the help of HTTP communicating between the browser and the server.

It provides static content to a web browser by loading a file from a disc and transferring it across the network to the user's web browser. This exchange is intermediates by the browser and the server, communicating using HTTP.

Application server

It is an environment where applications are able to run, no matter which types of applications and what operation they perform. It is also known as a type of middleware and can be able to develop and run web-based applications. Generally, it is used to connect database servers and end-user. There are several types of application servers, as well as .NET Framework, Java, and PHP application servers.

Database server

It is a computer system that allows other systems to access and retrieve data from a database. These servers respond to several requests to the clients and run database applications. Databases can require extraordinary amounts of disk space and can be accessed by multiple clients at any given time. It is also used by many companies for storage purposes. It allows users to access the data with the help of running a query by using a query language specific to the database. For example, SQL is a structured query language, which allows executing a query to access the data. The most common types of database server software include DB2, Oracle, Microsoft SQL, and Informix.

File server

It is a computer on a network that is used to store and distribute files. It allows multiple users or clients to share files, which is stored on a server. Furthermore, it can improve performance by maximizing readability and writing speeds.

Email server

The email server provides the facility of sending emails to the receiver and to receive emails from the sender as well as stores all the details and messages of the users' account on the server. The email server is also called mail server transfer agent (MTA) or internet mailer.

Whenever you mail or send a message to someone, it first accesses to SMTP (Simple Mail Transfer Protocol) and then identifies the address of the person

whom you want to send mail and then mail goes to the recipient. In short, these types of servers act as a virtual post office.

STEP 1: Go to VirtualBox website to download the binary for your current operating system and install Virtual box.

Oracle VM VirtualBox 6.1.22
Warning:
Network Interfaces
Installing the Oracle VM VirtualBox 6.1.22 Networking feature will reset your network connection and temporarily disconnect you from the network.
Proceed with installation now?
Version 6.1.22
Yes
No

Oracle VM VirtualBox 6.1.22 Setup
Oracle VM VirtualBox 6.1.22 installation is complete.
Click the Finish button to exit the Setup Wizard.
Start Oracle VM VirtualBox 6.1.22 after installation
Version 6.1.22
< Back
Finish
Cancel

STEP 2: Create Virtual Machine

Click 'New' button to open a dialog.

Type a name for the new virtual machine.

The memory size depends on your host machine memory size. Note that VirtualBox will create a swap partition with the same amount space as base memeory you have entered here. So later when you are selecting the size of the virtual hard drive, make sure it is large enough since the hard drive will be splitted into root (/)and swap partitions. The root partition contains by default all your system files, program settings and documents.

Do the following steps

Oracle VM VirtualBox Manager
File Machine Help
Tools
Create Virtual Hard Disk
Storage on physical hard disk
Please choose whether the new virtual hard disk file should grow as it is used (dynamically allocated) or if it should be created at its maximum size (fixed size).
A dynamically allocated hard disk file will only use space on your physical hard disk as it fills up (up to a maximum fixed size), although it will not shrink again automatically when space on it is freed.
A fixed size hard disk file may take longer to create on some systems but is often faster to use.
Dynamically allocated
Fixed size
Next
Cancel

Oracle VM VirtualBox Manager
File Machine Help
Tools
Create Virtual Hard Disk
File location and size
Please type the name of the new virtual hard disk file into the box below or click on the folder icon to select a different folder to create the file in.
C:\Users\Murshid\VirtualBox VMs\ubuntu 20.04\ubuntu 20.04
Select the size of the virtual hard disk in megabytes. This size is the limit on the amount of file data that a virtual machine will be able to store on the hard disk.
20.05 GB
4.00 MB
2.00 TB
Create
Cancel

STEP 3: Download iso file of latest version of Ubuntu.

Download 20.04LTS from Ubuntu Desktop

STEP 4: Now the virtual machine is created. We are ready to install Ubuntu in this virtual machine.

Select your new virtual machine and click 'Settings' button. Click on 'Storage' category and then 'Empty' under Controller:IDE. Click "CD/DVD" icon on right hand side and select the ubuntu ISO file to mount.

Oracle VM VirtualBox Manager
File Machine Help
Tools
ubuntu 20.04
Powered Off
New Settings Discard Start
General
Name: ubuntu 20.04
Operating System: Ubuntu (64-bit)
System
Preview
ubuntu 20.04
ubuntu 20.04 - Settings
Please choose a virtual optical disk file
Murshid › Downloads ›
Search Downloads
Organize New folder
CEV
Personal (E:)
Photo
Screenshots
Creative Cloud File
OneDrive
Murshid
3D Objects
Desktop
Documents
Downloads
PDF
Telegram Desktop
Whatsapp
ubuntu-20.04.2.0-desktop-amd64
Type: Disc Image File
Size: 2.67 GB
Date modified: 20-05-2021 05:51 PM
File name:
All virtual optical disk files (*.dr
Open
Cancel
Type here to search
ENG
06:14 PM
20-05-2021

Oracle VM VirtualBox Manager
File Machine Help
Tools
ubuntu 20.04
Powered Off
New Settings Discard Start
General
Name: ubuntu 20.04
Operating System: Ubuntu (64-bit)
System
Preview
ubuntu 20.04
ubuntu 20.04 - Settings
General
System
Display
Storage
Audio
Network
Serial Ports
USB
Shared Folders
User Interface
Storage
Storage Devices
Controller: IDE
ubuntu-20.04.2.0-desktop...
Controller: SATA
ubuntu 20.04.vdi
Attributes
Optical Drive: IDE Secondary Devic
Live CD/DVD
Information
Type: Image
Size: 2.68 GB
Location: C:\Users\Murshid\Downloa...
Attached to: --
Invalid settings detected
OK
Cancel
Type here to search
ENG
06:15 PM
20-05-2021

STEP 5: Install Ubuntu 20.04

Back to Oracle VM VirtualBox Manager, click on the new Ubuntu virtual machine and hit 'Start' button. Now you shall see a 'Welcome' screen and do the following steps.

Ubunutu 20.04 LTS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
May 20 14:31
Install
Welcome
English
Español
Esperanto
Euskara
Français
Gaeilge
Galego
Hrvatski
Íslenska
Italiano
Kurdî
Latviski
Lietuviškai
Magyar
Nederlands
No localization (UTF-8)
Norsk bokmål
Try Ubuntu
Install Ubuntu
You can try Ubuntu without making any changes to your computer, directly from this CD.
Or if you're ready, you can install Ubuntu alongside (or instead of) your current operating system. This shouldn't take too long.
You may wish to read the release notes.

Ubunutu 20.04 LTS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
May 20 15:17
Install
Keyboard layout
Choose your keyboard layout:
English (Australian)
English (Cameroon)
English (Ghana)
English (Nigeria)
English (South Africa)
English (UK)
English (US)
English (US)
English (US) - Cherokee
English (US) - English (Colemak)
English (US) - English (Dvorak)
English (US) - English (Dvorak, alt. intl.)
English (US) - English (Dvorak, intl., with dead keys)
English (US) - English (Dvorak, left-handed)
English (US) - English (Dvorak, right-handed)
Type here to test your keyboard
Detect Keyboard Layout
Quit
Back
Continue

Ubunutu 20.04 LTS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
May 20 15:18
Install
Updates and other software
What apps would you like to install to start with?
Normal installation
Web browser, utilities, office software, games, and media players.
Minimal installation
Web browser and basic utilities.
Other options
Download updates while installing Ubuntu
This saves time after installation.
Install third-party software for graphics and Wi-Fi hardware and additional media formats
This software is subject to license terms included with its documentation. Some is proprietary.
Quit
Back
Continue
Right Ctrl
Type here to search
ENG
08:48 PM
20-05-2021

Ubunutu 20.04 LTS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
May 20 14:33
Install
Installation type
This computer currently has no detected operating systems. What would you like to do?
Erase disk and install Ubuntu
Warning: This will delete all your programs, documents, photos, music, and any other files in all operating systems.
Advanced Features...
None selected
Something else
You can create or resize partitions yourself, or choose multiple partitions for Ubuntu.
Quit
Back
Install Now
Right Ctrl
Type here to search
ENG
08:03 PM
20-05-2021

Murshid
Google Chrome
Oracle VM VirtualBox
null
New Microsoft ...
Recycle Bin
New Microsoft ...
Microsoft Edge
New Microsoft ...
Notepad
Ubunutu 20.04 LTS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
May 20 14:33
Install
Installation type
This computer currently has no detected operating systems. What would you like to do?
Write the changes to disks?
If you continue, the changes listed below will be written to the disks. Otherwise, you will be able to make further changes manually.
The partition tables of the following devices are changed:
SCSI3 (0,0,0) (sda)
The following partitions are going to be formatted:
partition #1 of SCSI3 (0,0,0) (sda) as
partition #5 of SCSI3 (0,0,0) (sda) as ext4
Go Back
Continue
Back
Install Now
Right Ctrl
Type here to search
ENG
08:03 PM
20-05-2021

Murshid
Google Chrome
Oracle VM VirtualBox
null
New Microsoft ...
Recycle Bin
New Microsoft ...
Microsoft Edge
New Microsoft ...
Notepad
Ubunutu 20.04 LTS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
May 20 14:34
Install
Where are you?
Kolkata
Back
Continue
Right Ctrl
Type here to search
ENG
08:04 PM
20-05-2021

Murshid
Google Chrome
Oracle VM VirtualBox
null
New Microsoft ...
Recycle Bin
New Microsoft ...
Microsoft Edge
New Microsoft ...
Notepad
Ubuntu 20.04 LTS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
May 20 20:05
Install
Who are you?
Your name: Murshid
Your computer's name: murshid-VirtualBox
The name it uses when it talks to other computers.
Pick a username: murshid
Choose a password:
Short password
Confirm your password:
Log in automatically
Require my password to log in
Back
Continue
Right Ctrl
Type here to search
ENG
08:05 PM
20-05-2021

Murshid
Google Chrome
Oracle VM VirtualBox
null
New Microsoft ...
Recycle Bin
New Microsoft ...
Microsoft Edge
New Microsoft ...
Notepad
Ubuntu 20.04 LTS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
May 20 20:06
Install
Welcome to Ubuntu
Fast and full of new features, the latest version of Ubuntu makes computing easier than ever. Here are just a few cool new things to look out for...
Copying files...
Skip
Right Ctrl
Type here to search
ENG
08:06 PM
20-05-2021

After installation is complete, click 'Restart Now' button.

Enter the password you have chosen and press 'Enter'.

The Ubuntu Desktop OS is ready

Experiment No : 2

Aim: Study of a terminal based text editor such as Vim or Emacs. (By the end of the course,
students are expected to acquire following skills in using the editor: cursor operations,
manipulate text, search for patterns, global search and replace) Basic Linux commands, familiarity with following commands/operations expected

Text Editor

A text editor is a type of computer program that edits plain text.Text editors are provided with operating systems and software development packages, and can be used to change files such as configuration files, documentation files and programming language source code.
A text editor is a computer program that lets a user enter, change, store, and usually print text(characters and numbers, each encoded by the computer and its input and output devices, arranged to have meaning to users or to other programs). Typically, a text editor provides an "empty" display screen (or "scrollable page") with a fixed-line length and visible line numbers.

A popular text editor in IBM's large or mainframe computers is called XEDIT. In UNIX systems, the two most commonly used text editors are Emacs and Vi . In personal computer systems, word processors are more common than text editors. However, there are variations of mainframe and UNIX text editors that are provided for use on personal computers. An example is KEDIT, which is basically XEDIT for Windows.

Vi Text Editor

A text editor, written by Bill Joy in 1976. Short for Visual Interface.The vi editor is elaborated as **vi**sual editor. It is installed in every Unix system. In other words, it is available in all Linux distros. It is user-friendly and

works same on different distros and platforms. It is a very powerful application. An improved version of vi editor is **vim**.

●It enables fast, simple, and effective text editing mostly based on simple key bindings.

●It provides fast and convenient moving around files and between files.

●One must learn a good number of commands to be proficient in vi.

●We'll be using vim - vi improved

●Written by Bram Moolenaar

●vi is just an alias to vim

VIM

Vim editor is one of the more popular text editors we use today. It is a clone of the Vi editor and is written by Bram Moolenaar. It is cross platform editor and available on most popular platforms like Windows, Linux, Mac and other UNIX variants.

Features of Vim

•Its memory footprint is very low

•It is command centric. You can perform complex text related task with few commands

•It is highly configurable and uses simple text file to store its configuration

•There are many plug-in available for Vim. Its functionality can be extended in great manner using these plug-in

•It supports multiple windows. Using this feature screen can be split into multiple windows

•Same as multiple windows, it also supports multiple buffers

•It supports multiple tabs which allows to work on multiple files

•It supports recording features which allows to record and play Vim commands in repeated manner

•Excellent for programming due to intelligent language detection.

•NOT a formatting tool ... plain text only.

VIM Installation

Open terminal application.

1. Update package database by typing the **sudo apt update** command
2. Install vim on Ubuntu Linux, type: **sudo apt install vim**
3. Verify vim installation by typing the **vim --version** command

Open Vim **“vim -v”** by this command in terminal

Open an existing file from vim,first go to the file directory and command

vim [your file]

Eg: vim Demo

<u>Vim has three modes</u>

➢ Command

➢ Input

➢ Last Line

1.Command Mode

When you start vim, you begin in Command Mode by default.Hitting ESCAPE will get you back to Command Mode from other modes.In this mode you can issue many vi commands, including commands like *insert*, *append*, and *delete*, and other search and navigation commands that let you move around your file.

2.Insert Mode

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

3.Lastline Mode

The last vi mode is known as *vi last line mode*. You can only get to last line mode from command mode, and you get into last line mode by pressing the **colon key(:)**.After pressing this key, you'll see a colon character appear at the beginning of the last line of your vi editor window, and your cursor will be moved to that position. This indicates that vi is ready for you to type in a "last line command".

Some useful commands for VIM

Cursor movement

h - move cursor left

j - move cursor down

k - move cursor upl - move cursor right

w - jump forwards to the start of a word

W - jump forwards to the start of a word

e - jump forwards to the end of a word

E - jump forwards to the end of a word

b - jump backwards to the start of a word

B - jump backwards to the start of a word

0 - jump to the start of the line

^ - jump to the first non-blank character of the line

$ - jump to the end of the line

Insert mode

i - insert before the cursor

I - insert at the beginning of the line

a - insert (append) after the cursor

A - insert (append) at the end of the line

o - append (open) a new line below the current line

O - append (open) a new line above the current line

ea - insert (append) at the end of the word

Esc - exit insert mode

Editing

r - replace a single character

J - join line below to the current onec

c - change (replace) entire line

cw - change (replace) to the end of the word

c$ - change (replace) to the end of the line

s - delete character and substitute text

S - delete line and substitute text (same as cc)

xp - transpose two letters (delete and paste)

u – undo

Ctrl + r – redo

Cut and paste

yy - yank (copy) a line

2yy - yank (copy) 2 lines

yw - yank (copy) word

y$ - yank (copy) to end of line

p - put (paste) the clipboard after cursor

P - put (paste) before cursor

dd - delete (cut) a line

2dd - delete (cut) 2 lines

dw - delete (cut) word

D - delete (cut) to the end of the line

d$ - delete (cut) to the end of the line

x - delete (cut) character

Exiting commands

:w - write (save) the file, but don't exit

:wq - write (save) and quit

:x - write (save) and quit

:q - quit (fails if there are unsaved changes)

:q! - quit and throw away unsaved changes

Basic LINUX Commands

man

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

Syntax: $ man [COMMAND NAME]

Example: **$ man ls**

ls: ls is a Linux shell command that lists directory contents of files and directories.

echo : echo command in linux is used to display line of text/string that are passed as an argument . This is a built in command that is mostly used in shell scripts and batch files to output status text to the screen or a file.

read : read command is used to obtain input from users. Understanding the “read” command is key to making your code more interactive. The “read” command is used to obtain inputted information from the user.

cd : cd command in linux known as change directory command. It is used to change current working directory.

mkdir : mkdir command in Linux allows the user to create directories (also referred to as folders in some operating systems). This command can create multiple directories at once as well as set the permissions for the directories.

pwd : pwd stands for **P**rint **W**orking **D**irectory. It prints the path of the working directory, starting from the root.

find : It seaches for files and directories in a directory heirarchy based on a user given expression and can perform user specified action on each matched file.

Syntax: **$ find [where to start searching from][exprssion determines what to find] [-options] [what to find]**

touch : It is used to create a file without any content. The file created using touch command is empty. This command can be used when the user doesn't have data to store at the time of file creation.

cat : Cat(concatenate) command is very frequently used in Linux. It reads data from the file and gives their content as output. It helps us to create, view, concatenate files.

Linux cat command usage

Option	Function
cat > [fileName]	To create a file.
cat [oldfile] > [newfile]	To copy content from older to new file.
cat [file1 file2 and so on] > [new file name]	To concatenate contents of multiple files into one.
cat -n/cat -b [fileName]	To display line numbers.
cat -e [fileName]	To display $ character at the end of each line.
cat [fileName] <<EOF	Used as page end marker.

locate : The locate command is used to search a file by file name. It is quite similar to find command; the difference is that it is a background process. It searches the file in the database, whereas the find command searches in the file system. It is faster than the find command. To find the file with the locates command, keep your database updated.

date and cal : The date command is used to display date, time, time zone, and more.

The **cal** command is used to display the current month's calendar with the current date highlighted.

ls -l : It will show the list in a long list format.It includes,

•number of links to the content

•owner of the content

•group owner of the content

•size of the content in bytes

•last modified date / time of the content

•file or directory name

whatis :command in Linux is used to get a one-line manual page descriptions. In Linux, each manual page has some sort of description within it. So this command search for the manual pages names and show the manual page description of the specified filename or argument.

whereis : command is used to find the location of source/binary file of a command and manuals sections for a specified file in Linux system.

```
user@murshid-tp: ~
user@murshid-tp:~$ whatis ls
ls (1)               - list directory contents
user@murshid-tp:~$ whatis read
read (2)             - read from a file descriptor
user@murshid-tp:~$ whatis echo
echo (1)             - display a line of text
user@murshid-tp:~$ whereis ls
ls: /usr/bin/ls /usr/share/man/man1/ls.1.gz
user@murshid-tp:~$ whereis read
read: /usr/share/man/man2/read.2.gz
user@murshid-tp:~$ whereis echo
echo: /usr/bin/echo /usr/share/man/man1/echo.1.gz
user@murshid-tp:~$
```

history : *history* command is used to view the previously executed command.

```
user@murshid-tp: ~
1456  cat Country
1457  clear
1458  ls
1459  rename capital Capital of state
1460  sudo app install rename
1461  sudo apt install rename
1462  clear
1463  id
1464  clear
1465  sort capital
1466  cat capital
1467  clear
1468  find . -name "*.pdf"
1469  find fee2.pdf
1470  locate fee2.pdf
1471  sudo apt install mlocate
1472  locate fee2.pdf
1473  exit
1474  cal
1475  time
1476  ls
1477  rename State s
1478  ping gmail.com
1479  clear
1480  ls
1481  date
1482  cal
1483  locate fee.pdf
1484  cd EXAM
1485  ls
1486  ls -l
1487  ls
1488  cd ..
1489  ls
1490  echo $SHELL
1491  history
1492  whatis ls
1493  whatis cd
1494  whatis read
1495  whatis -d ls
1496  clear
1497  whatis ls
1498  whatis read
1499  whatis echo
1500  whereis ls
1501  whereis read
1502  whereis echo
1503  clear
1504  history
user@murshid-tp:~$
```

alias : Linux 'alias' command replaces one string from the shell with another string. It is a shell built-in command. It converts a complicated command into a simpler command or in other words, it creates a shortcut by replacing it with the simpler one.

rm -i : prompt before every removal.

```
user@murshid-tp: ~/Desktop
user@murshid-tp:~/Desktop$ ls
 Country   EXAM  'My Projects'   Notes   Programs   State
user@murshid-tp:~/Desktop$ rm -i Country
rm: remove regular file 'Country'? y
user@murshid-tp:~/Desktop$ ls
 EXAM  'My Projects'   Notes   Programs   State
user@murshid-tp:~/Desktop$ alias s=ls
user@murshid-tp:~/Desktop$ s
 EXAM  'My Projects'   Notes   Programs   State
user@murshid-tp:~/Desktop$ ls
 EXAM  'My Projects'   Notes   Programs   State
user@murshid-tp:~/Desktop$
```

Sort : SORT command is used to sort a file, arranging the records in a particular order. By default, the sort command sorts file assuming the contents are ASCII. It sort alphabeticaly.

```
user@murshid-tp: ~/Desktop
user@murshid-tp:~/Desktop$ ls
 EXAM  'My Projects'   Notes   Programs   State
user@murshid-tp:~/Desktop$ cat State
kerala
tamilnadu
karnataka
goa
telengana
rajasthan
andrapradesh

user@murshid-tp:~/Desktop$ sort State

andrapradesh
goa
karnataka
kerala
rajasthan
tamilnadu
telengana
user@murshid-tp:~/Desktop$
```

<u>mv</u> : mv stands for **move**. mv is used to move one or more files or directories from one place to another in a file system.

<u>cp</u> : To copy one or more files to another location.

<u>rm</u> : rm command is used to remove objects such as files, directories, symbolic links and so on from the file system.

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

wc -l : Prints the number of lines in a file.
wc -w : prints the number of words in a file.
wc -c : Displays the count of bytes in a file.
wc -m : prints the count of characters from a file.
wc -L : prints only the length of the longest line in a file.

```
user@murshid-tp:~/Desktop$ ls
 demo.html   EXAM  'My Projects'   Notes   Programs
user@murshid-tp:~/Desktop$ cat demo.html
Hello
Welcome to Python
user@murshid-tp:~/Desktop$ wc -l demo.html
2 demo.html
user@murshid-tp:~/Desktop$ wc -w demo.html
4 demo.html
user@murshid-tp:~/Desktop$ wc -m demo.html
25 demo.html
user@murshid-tp:~/Desktop$ wc -c demo.html
25 demo.html
user@murshid-tp:~/Desktop$
```

cut : Linux cut command is useful for selecting a specific column of a file. It is used to cut a specific sections by byte position, character, and field and writes them to standard output. It cuts a line and extracts the text data. It is necessary to pass an argument with it; otherwise, it will throw an error message.

To cut a specific section, it is necessary to specify the delimiter. A delimiter will decide how the sections are separated in a text file. Delimiters can be a space (' '), a hyphen (-), a slash (/), or anything else. After '-f' option, the column number is mentioned.

-b, --bytes=LIST: It is used to cut a specific section by bytes.

-c, --characters=LIST: It is used to select the specified characters.

-d, --delimiter=DELIM: It is used to cut a specific section by a delimiter.

-f, --fields=LIST: It is used to select the specific fields. It also prints any line that does not contain any delimiter character, unless the -s option is specified.

-n: It is used to ignore any option.

paste : The paste command is useful for merging files together. The first line of each file is joined separated by a Tab character. It is possible to specify a different delimiter with the -d parameter.

```
user@murshid-tp: ~/Desktop
user@murshid-tp:~/Desktop$ cat demo
harry,25,16200
gill,46,17500
bill,45,20000
john,43,100000
barry,27,42000
paul,18,26500
user@murshid-tp:~/Desktop$ cut -d, -f 1,3 demo
harry,16200
gill,17500
bill,20000
john,100000
barry,42000
paul,26500
user@murshid-tp:~/Desktop$ cat state
kerala
tamilnadu
karnataka
telengana
user@murshid-tp:~/Desktop$ cat capital
Trivandrum
Chennai
Banglore
Hyderabad
user@murshid-tp:~/Desktop$ paste -d: state capital
kerala:Trivandrum
tamilnadu:Chennai
karnataka:Banglore
telengana:Hyderabad
user@murshid-tp:~/Desktop$
```

<u>head and tail</u> : The head command, as the name implies, print the top N number of data of the given input. By default, it prints the first 10 lines of the specified files. If more than one file name is provided then data from each file is preceded by its file name.The tail command, as the name implies, print the last N number of data of the given input. By default it prints the last 10 lines of the specified files. If more than one file name is provided then data from each file is precedes by its file name.

```
user@murshid-tp: ~/Desktop
user@murshid-tp:~/Desktop$ cat state
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chhattisgarh
Goa
Gujarat
Haryana
Himachal Pradesh
Jammu and Kashmir
Jharkhand
Karnataka
Kerala
Madhya Pradesh
Maharashtra
user@murshid-tp:~/Desktop$ head state
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chhattisgarh
Goa
Gujarat
Haryana
Himachal Pradesh
Jammu and Kashmir
user@murshid-tp:~/Desktop$ tail state
Goa
Gujarat
Haryana
Himachal Pradesh
Jammu and Kashmir
Jharkhand
Karnataka
Kerala
Madhya Pradesh
Maharashtra
user@murshid-tp:~/Desktop$
```

<u>expr and grep</u> : The **expr command** is used to evaluate a given expression and display its standard output. Each separated expression is considered as an argument. These expressions could be integer and string expressions, including regular expressions. If expressions are not passed properly, it will prevent the execution of the command.The grep filter searches a file for a particular pattern of characters, and displays all lines that contain that pattern. The pattern that is searched in the file is referred to as the regular expression.

<u>df,top,ps</u> : Linux df command is used to display the **disk space used in the file system**. The 'df' stands for **"disk filesystem**." It defines the number of blocks used, the number of blocks available, and the directory where the file system is mounted.

top command is used to show the Linux processes. It provides a dynamic real-time view of the running system. Usually, this command shows the summary information of the system and the list of processes or threads which are currently managed by the Linux Kernel.The ps command is used to view currently running processes on the system. It helps us to determine which process is doing what in our system, how much memory it is using, how much CPU space it occupies, user ID, command name, etc .The **ps** command may display different results for different systems because it displays information about the currently running process of a system.

Experiment No : 3

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

Linux Filesystem Hierarchy Standard (FHS)

Filesystem hierarchy standard describes directory structure and its content in Unix and Unix like operating system. It explains where files and directories should be located and what it should contain.

◆ In the FHS, all files and directories appear under the root directory /, even if they are stored on different physical or virtual devices.

◆ Some of these directories only exist on a particular system if certain subsystems, such as the X Window System, are installed.

◆ Most of these directories exist in all UNIX operating systems and are generally used in much the same way; however, the descriptions here are those used specifically for the FHS, and are not considered authoritative for platforms other than Linux.

Directory type	Types of files stored
Binary directories	Contains binary or compiled source code files, eg, /bin, /sbin, etc.
Configuration directories	Contains configuration files of the system, eg, /etc, /boot.
Data directories	Stores data files, eg, /home, /root, etc.
Memory directories	Stores device files which doesn't take up actual hard disk space, eg, /dev, /proc, /sys.
Usr (Unix System Resources)	Contains sharable, read only data, eg, /usr/bin, /usr/lib, etc.
var (variable directory)	Contains larger size data, eg, /var/log, /var/cache, etc.
Non standerd directories	Directories which do not come under standard FHS, eg, lost+found, /run, etc.

1. Root Directory

Everything on your Linux system is located under the / directory, known as the root directory. You can think of the / directory as being similar to the C:\ directory on Windows – but this isn't strictly true, as Linux doesn't have drive letters. While another partition would be located at D:\ on Windows, this other partition would appear in another folder under / on Linux.

◆ Every single file and directory starts from the root directory

◆ The only root user has the right to write under this directory

◆ /root is the root user's home directory, which is not the same as /

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

◆Contains binary executables

◆Common linux commands you need to use in single-user modes are located under this directory.

◆Commands used by all the users of the system are located here e.g. ps, ls, ping, grep, cp

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

•Kernel initrd, vmlinux, grub files are located under /boot

•Example: initrd.img-2.6.32-24-generic, vmlinuz-2.6.32-24-generic

4. /dev : Device files

Linux exposes devices as files, and the /dev directory contains a number of special files that represent devices. These are not actual files as we know them, but they appear as files – for example, /dev/sda represents the first SATA drive in the system. If you wanted to partition it, you could start a partition editor and tell it to edit /dev/sda.

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

◆ Contains configuration files required by all programs.

◆ This also contains startup and shutdown shell scripts used to start/stop individual programs.

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

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

- Home directories for all users to store their personal files.
- example: /home/kishlay, /home/kv.

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

- Library filenames are either ld* or lib*.so.*
- Example: ld-2.11.1.so, libncurses.so.5.7

8. /mnt : Temporary Mount Points

Historically speaking, the /mnt directory is where system administrators mounted temporary file systems while using them. For example, if you're mounting a Windows partition to perform some file recovery operations, you might mount it at /mnt/windows. However, you can mount other file systems anywhere on the system.

9. /opt : Optional Packages

The /opt directory contains subdirectories for optional software packages. It's commonly used by proprietary software that doesn't obey the standard file system hierarchy – for example, a proprietary program might dump its files in /opt/application when you install it.

10. <u>/sbin</u> : Essential system binaries, e.g., fsck, init, route.

◆Just like /bin, /sbin also contains binary executables.

◆The linux commands located under this directory are used typically by system administrator, for system maintenance purpose.

◆Example: iptables, reboot, fdisk, ifconfig, swapon

11. <u>/srv</u> : Service Data

The /srv directory contains “data for services provided by the system.” If you were using the Apache HTTP server to serve a website, you’d likely store your website’s files in a directory inside the /srv directory.

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

◆ Directory that contains temporary files created by system and users.

◆ Files under this directory are deleted when system is rebooted.

13./run : Application State Files

The /run directory is fairly new, and gives applications a standard place to store transient files they require like sockets and process IDs. These files can't be stored in /tmp because files in /tmp may be deleted.

14./usr – User Binaries & Read-Only Data

The /usr directory contains applications and files used by users, as opposed to applications and files used by the system. For example, non-essential applications are located inside the /usr/bin directory instead of the /bin directory and non-essential system administration binaries are located in the /usr/sbin directory instead of the /sbin directory. Libraries for each are located inside the /usr/lib directory. The /usr directory also contains other directories – for example, architecture-independent files like graphics are located in /usr/share.

The /usr/local directory is where locally compiled applications install to by default – this prevents them from mucking up the rest of the system.

15. /var – Variable Data Files

The /var directory is the writable counterpart to the /usr directory, which must be read-only in normal operation. Log files and everything else that would normally be written to /usr during normal operation are written to the /var directory. For example, you'll find log files in /var/log.

File permissions

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

When you execute an “ls” command, you are not given any information about the security of the files, because by default “ls” only lists the names of files. You can get more information by using an “option” with the “ls” command. All options start with a ‘-‘. For example, to execute “ls” with the “long listing” option, you would type ls -l

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

- **The file owner.**
- **The group members.**
- **Others.**

All the three owners (user owner, group, others) in the Linux system have three types of permissions defined. Nine characters denotes the three types of permissions.

1. **Read (r) :** The read permission allows you to open and read the content of a file. But you can't do any editing or modification in the file.

2. **Write (w) :** The write permission allows you to edit, remove or rename a file. For instance, if a file is present in a directory, and write permission is set on the file but not on the directory, then you can edit the content of the file but can't remove, or rename it.

3. **Execute (x):** In Unix type system, you can't run or execute a program unless execute permission is set.But in Windows, there is no such permission available.

To view the file permissions, use the **ls** command:

ls -l file_name

```
-rw-r--r-- 12 linuxize users 12.0K Apr  28 10:10 file_name
|[-][-][-]-   [------] [---]
| | | | |     |      |
| | | | |     |      +-----------> 7. Group
| | | | |     +------------------> 6. Owner
| | | | +-------------------------> 5. Alternate Access Method
| | | +---------------------------> 4. Others Permissions
| | +-------------------------------> 3. Group Permissions
| +----------------------------------> 2. Owner Permissions
+-------------------------------------> 1. File Type
```

Let's see file permissions in Linux with examples:

ls – l on terminal gives

Here, we have highlighted **'-rw-rw-r--'**and this weird looking code is the one that tells us about the Unix permissions given to the owner, user group and the world.

Here, the first '**-**' implies that we have selected a file.p>

Else, if it were a directory, **d** would have been shown.

The characters are pretty easy to remember.

r= read permission
w = write permission
x = execute permission
- = no permission

Let us look at it this way.

The first part of the code is **'rw-'**. This suggests that the owner 'Home' can:

•Read the file

•Write or edit the file

•He cannot execute the file since the execute bit is set to '-'.

By design, many Linux distributions like Fedora, CentOS, Ubuntu, etc. will add users to a group of the same group name as the user name. Thus, a user 'tom' is added to a group named 'tom'.

The second part is**'rw-'.**It for the user group 'Home' and group-members can:

- •Read the file
- •Write or edit the file

The third part is for the world which means any user. It says **'r--'.** This means the user can only:

- •Read the file

Changing file/directory permissions with 'chmod' command

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

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

chmod permissions filename
There are 2 ways to use the command -

1.**Absolute mode**

2.**Symbolic mode**

1.Absolute(Numeric) Mode

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

The table below gives numbers for all for permissions types.

Number	Permission Type	Symbol
0	No Permission	---
1	Execute	--x
2	Write	-w-
3	Execute + Write	-wx
4	Read	r--
5	Read + Execute	r-x
6	Read +Write	rw-
7	Read + Write +Execute	rwx

Example :

Checking current file permissions of file1 **: 664(- rw-rw-r--)**

```
user@murshid-tp:~/Desktop$ ls
 EXAM   file1  'My Projects'   Notes   Programs
user@murshid-tp:~/Desktop$ ls -l
total 16
drwxrwxr-x 4 user user 4096 Aug 12 20:38  EXAM
-rw-rw-r-- 1 user user    0 Aug 15 16:57  file1
drwxrwxr-x 5 user user 4096 Jun  5 19:52 'My Projects'
drwxrwxrwx 4 user user 4096 Jun 28 19:03  Notes
drwxrwxr-x 7 user user 4096 Aug 10 19:34  Programs
user@murshid-tp:~/Desktop$ ls -l file1
-rw-rw-r-- 1 user user 0 Aug 15 16:57 file1
user@murshid-tp:~/Desktop$
```

After change the file permission to **764(rwx-rw-r--)** by using the command:

chmod 764 file1

```
user@murshid-tp:~/Desktop$ ls
 EXAM   file1  'My Projects'   Notes   Programs
user@murshid-tp:~/Desktop$ ls -l file1
-rw-rw-r-- 1 user user 0 Aug 15 16:57 file1
user@murshid-tp:~/Desktop$ chmod 764 file1
user@murshid-tp:~/Desktop$ ls -l file1
-rwxrw-r-- 1 user user 0 Aug 15 16:57 file1
user@murshid-tp:~/Desktop$
```

'764' absolute code says the following:

- Owner can read, write and execute
- Usergroup can read and write
- World can only read

2.Symbolic Mode

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

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

The various owners are represented as -

User Denotations	
u	user/owner
g	group
o	other
a	all

Example :

Checking current file permissions of file1 : - rw-rw-r--

```
user@murshid-tp:~/Desktop$ ls
 EXAM   file1  'My Projects'   Notes   Programs
user@murshid-tp:~/Desktop$ ls -l
total 16
drwxrwxr-x 4 user user 4096 Aug 12 20:38  EXAM
-rw-rw-r-- 1 user user    0 Aug 15 16:57  file1
drwxrwxr-x 5 user user 4096 Jun  5 19:52 'My Projects'
drwxrwxrwx 4 user user 4096 Jun 28 19:03  Notes
drwxrwxr-x 7 user user 4096 Aug 10 19:34  Programs
user@murshid-tp:~/Desktop$ ls -l file1
-rw-rw-r-- 1 user user 0 Aug 15 16:57 file1
user@murshid-tp:~/Desktop$
```

Adding execute permission to owner :

```
user@murshid-tp:~/Desktop$ ls
 EXAM   file1  'My Projects'   Notes   Programs
user@murshid-tp:~/Desktop$ ls -l file1
-rw-rw-r-- 1 user user 0 Aug 15 16:57 file1
user@murshid-tp:~/Desktop$ chmod u+x file1
user@murshid-tp:~/Desktop$ ls -l file1
-rwxrw-r-- 1 user user 0 Aug 15 16:57 file1
user@murshid-tp:~/Desktop$
```

Removing wright permission to group :

```
user@murshid-tp:~/Desktop$ ls
 EXAM   file1  'My Projects'   Notes   Programs
user@murshid-tp:~/Desktop$ ls -l file1
-rw-rw-r-- 1 user user 0 Aug 15 16:57 file1
user@murshid-tp:~/Desktop$ chmod g-w file1
user@murshid-tp:~/Desktop$ ls -l file1
-rw-r--r-- 1 user user 0 Aug 15 16:57 file1
user@murshid-tp:~/Desktop$
```

Setting all permission to others:

```
user@murshid-tp:~/Desktop$ ls
 EXAM   file1  'My Projects'   Notes   Programs
user@murshid-tp:~/Desktop$ ls -l file1
-rw-rw-r-- 1 user user 0 Aug 15 16:57 file1
user@murshid-tp:~/Desktop$ chmod o=rwx file1
user@murshid-tp:~/Desktop$ ls -l file1
-rw-rw-rwx 1 user user 0 Aug 15 16:57 file1
user@murshid-tp:~/Desktop$
```

Changing Ownership and Group

For changing the ownership of a file/directory, you can use the following command:

```
chown user filename
```

In case you want to change the user as well as group for a file or directory use the command

```
chown user:group filename
```

Let's see this in action

Checking the current ownership:

```
user@murshid-tp:~/Desktop$ ls
 EXAM   file1  'My Projects'   Notes   Programs
user@murshid-tp:~/Desktop$ ls -l
total 16
drwxrwxr-x 4 user user 4096 Aug 12 20:38  EXAM
-rw-rw-r-- 1 user user    0 Aug 15 16:57  file1
drwxrwxr-x 5 user user 4096 Jun  5 19:52 'My Projects'
drwxrwxrwx 4 user user 4096 Jun 28 19:03  Notes
drwxrwxr-x 7 user user 4096 Aug 10 19:34  Programs
user@murshid-tp:~/Desktop$ ls -l file1
-rw-rw-r-- 1 user user 0 Aug 15 16:57 file1
user@murshid-tp:~/Desktop$
```

Change the file owner to musrhid,by using the command :

sudo chown murshid file1

```
user@murshid-tp:~/Desktop$ ls
 EXAM   file1  'My Projects'   Notes   Programs
user@murshid-tp:~/Desktop$ ls -l file1
-rw-rw-r-- 1 user user 0 Aug 15 16:57 file1
user@murshid-tp:~/Desktop$ sudo chown murshid file1
user@murshid-tp:~/Desktop$ ls -l file1
-rw-rw-r-- 1 murshid user 0 Aug 15 16:57 file1
user@murshid-tp:~/Desktop$
```

Changing user and group to root :

sudo chown root:root file1

```
user@murshid-tp:~/Desktop$ ls
 EXAM   file1  'My Projects'   Notes   Programs
user@murshid-tp:~/Desktop$ ls -l file1
-rw-rw-r-- 1 user user 0 Aug 15 16:57 file1
user@murshid-tp:~/Desktop$ sudo chown root:root file1
user@murshid-tp:~/Desktop$ ls -l file1
-rw-rw-r-- 1 root root 0 Aug 15 16:57 file1
user@murshid-tp:~/Desktop$
```

In case you want to change group-owner only, use the command

```
chgrp group_name filename
```

'**chgrp**' stands for change group.

Change the group owner to root :

sudo chgrp root file1

```
user@murshid-tp:~/Desktop$ ls
 EXAM   file1  'My Projects'   Notes   Programs
user@murshid-tp:~/Desktop$ ls -l file1
-rw-rw-r-- 1 user user 0 Aug 15 16:57 file1
user@murshid-tp:~/Desktop$ sudo chgrp root file1
user@murshid-tp:~/Desktop$ ls -l file1
-rw-rw-r-- 1 user root 0 Aug 15 16:57 file1
user@murshid-tp:~/Desktop$
```

The most important configuration files

As we mentioned before, most configuration files are stored in the /etc directory. Content can be viewed using the **cat** command, which sends text files to the standard output (usually your monitor). The syntax is straight forward:

cat file1 file2 ... fileN

In this section we try to give an overview of the most common configuration files. This is certainly not a complete list. Adding extra packages may also add extra configuration files in /etc. When reading the configuration files, you will find that they are usually quite well commented and self-explanatory. Some files also have man pages which contain extra documentation, such as **man *group***.

Table 3-3. Most common configuration files

File	Information/service
aliases	Mail aliases file for use with the Sendmail and Postfix mail server. Running a mail server on each and every system has long been common use in the UNIX world, and almost every Linux distribution still comes with a Sendmail package. In this file local user names are matched with real names as they occur in E-mail addresses, or with other local addresses.
apache	Config files for the Apache web server.
bashrc	The system-wide configuration file for the Bourne Again SHell. Defines functions and aliases for all users. Other shells may have their own system-wide config files, like cshrc.
crontab and the cron.* directories	Configuration of tasks that need to be executed periodically - backups, updates of the system databases, cleaning of the system, rotating logs etc.
default	Default options for certain commands, such as **useradd**.
filesystems	Known file systems: ext3, vfat, iso9660 etc.
fstab	Lists partitions and their *mount points*.
ftp*	Configuration of the ftp-server: who can connect, what parts of the system are accessible etc.
group	Configuration file for user groups. Use the shadow utilities **groupadd**, **groupmod** and **groupdel** to edit this file. Edit manually only if you really know what you are doing.
hosts	A list of machines that can be contacted using the network, but without the need for a domain name service. This has nothing to do with the system's network configuration, which is done in /etc/sysconfig.
inittab	Information for booting: mode, number of text consoles etc.
issue	Information about the distribution (release version and/or kernel

File	Information/service
	info).
ld.so.conf	Locations of library files.
lilo.conf, silo.conf, aboot.conf etc.	Boot information for the LInux LOader, the system for booting that is now gradually being replaced with GRUB.
logrotate.*	Rotation of the logs, a system preventing the collection of huge amounts of log files.
mail	Directory containing instructions for the behavior of the mail server.
modules.conf	Configuration of modules that enable special features (drivers).
motd	Message Of The Day: Shown to everyone who connects to the system (in text mode), may be used by the system admin to announce system services/maintenance etc.
mtab	Currently mounted file systems. It is advised to never edit this file.
nsswitch.conf	Order in which to contact the name resolvers when a process demands resolving of a host name.
pam.d	Configuration of authentication modules.
passwd	Lists local users. Use the shadow utilities **useradd**, **usermod** and **userdel** to edit this file. Edit manually only when you really know what you are doing.
printcap	Outdated but still frequently used printer configuration file. Don't edit this manually unless you really know what you are doing.
profile	System wide configuration of the shell environment: variables, default properties of new files, limitation of resources etc.
rc*	Directories defining active services for each run level.
resolv.conf	Order in which to contact DNS servers (Domain Name Servers only).
sendmail.cf	Main config file for the Sendmail server.
services	Connections accepted by this machine (open ports).
sndconfig or sound	Configuration of the sound card and sound events.
ssh	Directory containing the config files for secure shell client and server.
sysconfig	Directory containing the system configuration files: mouse, keyboard, network, desktop, system clock, power management etc. (specific to RedHat)
X11	Settings for the graphical server, X. RedHat uses XFree, which is reflected in the name of the main configuration file, XFree86Config. Also contains the general directions for the window managers available on the system, for example **gdm**, **fvwm**, **twm**, etc.
xinetd.* or inetd.conf	Configuration files for Internet services that are run from the system's (extended) Internet services daemon (servers that don't run an independent daemon).

Linux log files

Log files are the records that Linux stores for administrators to keep track and monitor important events about the server, kernel, services, and applications running on it. In this post, we'll go over the top Linux log files server administrators should monitor.

Log files are a set of records that Linux maintains for the administrators to keep track of important events. They contain messages about the server, including the kernel, services and applications running on it.

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

The log files generated in a Linux environment can typically be classified into four different categories:

- Application Logs
- Event Logs
- Service Logs
- System Logs

Common Linux log files names and usage

- /var/log/messages : General message and system related stuff
- /var/log/auth.log : Authenication logs
- /var/log/kern.log : Kernel logs
- /var/log/cron.log : Crond logs (cron job)
- /var/log/maillog : Mail server logs
- /var/log/qmail/ : Qmail log directory (more files inside this directory)
- /var/log/httpd/ : Apache access and error logs directory
- /var/log/lighttpd/ : Lighttpd access and error logs directory
- /var/log/nginx/ : Nginx access and error logs directory
- /var/log/apt/ : Apt/apt-get command history and logs directory
- /var/log/boot.log : System boot log
- /var/log/mysqld.log : MySQL database server log file
- /var/log/secure or /var/log/auth.log : Authentication log
- /var/log/utmp or /var/log/wtmp : Login records file
- /var/log/yum.log or /var/log/dnf.log: Yum/Dnf command log file.

Experiment No : 4

Aim : Shell scripting: study bash syntax, environment variables, variables, control constructs such as if, for and while, aliases and functions, accessing command line arguments passed to shell scripts. Study of startup scripts, login and logout scripts, familiarity with systemd and system 5 init scripts is expected.

Shell Scripting

Shell Scripting is an open-source computer program designed to be run by the Unix/Linux shell. Shell Scripting is a program to write a series of commands for the shell to execute. It can combine lengthy and repetitive sequences of commands into a single and simple script that can be stored and executed anytime which, reduces programming efforts.

Shell

Shell is a UNIX term for an interface between a user and an operating system service. Shell provides users with an interface and accepts human-readable commands into the system and executes those commands which can run automatically and give the program's output in a shell script.

An Operating is made of many components, but its two prime components are -

- Kernel
- Shell

A Kernel is at the nucleus of a computer. It makes the communication between the hardware and software possible. While the Kernel is the innermost part of an operating system, a shell is the outermost one.
A shell in a Linux operating system takes input from you in the form of commands, processes it, and then gives an output. It is the interface through which a user works on the programs, commands, and scripts. A shell is accessed by a terminal which runs it.
When you run the terminal, the Shell issues **a command prompt (usually $),** where you can type your input, which is then executed when you hit the Enter key. The output or the result is thereafter displayed on the terminal.
The Shell wraps around the delicate interior of an Operating system protecting it from accidental damage. Hence the name **Shell**.

Types of Shell

There are two main shells in Linux:

1. The **Bourne Shell**: The prompt for this shell is $ and its derivatives are listed below:

- POSIX shell also is known as sh
- Korn Shell also knew as sh
- **B**ourne **A**gain **SH**ell also knew as bash (most popular)

2. The C shell: The prompt for this shell is %, and its subcategories are:

- C shell also is known as csh
- Tops C shell also is known as tcsh

Why do we need shell scripts

There are many reasons to write shell scripts –

- To avoid repetitive work and automation
- System admins use shell scripting for routine backups
- System monitoring
- Adding new functionality to the shell etc.

Advantages of shell scripts

•The command and syntax are exactly the same as those directly entered in command line, so programmer do not need to switch to entirely different syntax

•Writing shell scripts are much quicker

•Quick start

•Interactive debugging etc.

Disadvantages of shell scripts

•Prone to costly errors, a single mistake can change the command which might be harmful

•Slow execution speed

•Design flaws within the language syntax or implementation

•Not well suited for large and complex task

•Provide minimal data structure unlike other scripting languages. etc

Steps in creating a Shell Script:

1. Create a file using a **vi** editor(or any other editor). Name script file with **extension .sh**
2. **Start** the script with **#! /bin/sh**
3. Write some code.
4. Save the script file as filename.sh
5. For **executing** the script type **bash filename**

Shell Variables

As discussed earlier, Variables store data in the form of characters and numbers. Similarly, Shell variables are used to store information and they can by the shell only.
For example, the following creates a shell variable and then prints it:

```
variable ="Hello"
echo $variable
```

Relational Operators

Operator	Description	Example
-eq	Checks if the value of two operands are equal or not; if yes, then the condition becomes true.	[$a -eq $b] is not true.
-ne	Checks if the value of two operands are equal or not; if values are not equal, then the condition becomes true.	[$a -ne $b] is true.
-gt	Checks if the value of left operand is greater than the value of right operand; if yes, then the condition becomes true.	[$a -gt $b] is not true.
-lt	Checks if the value of left operand is less than the value of right operand; if yes, then the condition becomes true.	[$a -lt $b] is true.
-ge	Checks if the value of left operand is greater than or equal to the value of right operand; if yes, then the condition becomes true.	[$a -ge $b] is not true.
-le	Checks if the value of left operand is less than or equal to the value of right operand; if yes, then the condition becomes true.	[$a -le $b] is true.

Example:

vim example.sh

```
#!/bin/sh

echo "Enter your name:"
read name
echo "Welocome" $name
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

```
user@murshid-tp:~/Desktop$ vim example.sh
user@murshid-tp:~/Desktop$ bash example.sh
Enter your name:
David
Welocome David
user@murshid-tp:~/Desktop$
```

Conditional Statements: There are total 5 conditional statements which can be used in bash programming

1.if statement

2.if-else statement

3.if..elif..else..fi statement (Else If ladder)

4.if..then..else..if..then..fi..fi..(Nested if)

5.switch statement

Their description with syntax is as follows:

<u>if statement</u>

This block will process if specified condition is true.

Syntax:

```
if [ expression ]
then
  statement
fi
```

<u>if-else statement</u>

If specified condition is not true in if part then else part will be execute.

Syntax

```
if [ expression ]
then
  statement1
else
  statement2
fi
```

<u>if..elif..else..fi statement (Else If ladder)</u>

To use multiple conditions in one if-else block, then elif keyword is used in shell. If expression1 is true then it executes statement 1 and 2, and this process continues. If none of the condition is true then it processes else part.

Syntax

```
if [ expression1 ]
then
  statement1
  statement2
  .
  .
```

```
elif [ expression2 ]
then
  statement3
  statement4
  .
  .
else
  statement5
fi
```

<u>if..then..else..if..then..fi..fi..(Nested if)</u>

Nested if-else block can be used when, one condition is satisfies then it again checks another condition. In the syntax, if expression1 is false then it processes else part, and again expression2 will be check.

Syntax:

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

<u>switch statement</u>

case statement works as a switch statement if specified value match with the pattern then it will execute a block of that particular pattern

When a match is found all of the associated statements until the double semicolon (;;) is executed.

A case will be terminated when the last command is executed.

If there is no match, the exit status of the case is zero.

Syntax:

```
case  in
  Pattern 1) Statement 1;;
  Pattern n) Statement n;;
esac
```

Example :

```
#!/bin/sh

echo "Enter a number:"
read n
if [ $((n % 2)) -eq 0 ]
then
        echo "No is even"
else
        echo "No is odd"
fi
~
~
~
~
~
~
~
~
~
~
```

```
user@murshid-tp:~/Desktop$ vim example.sh
user@murshid-tp:~/Desktop$ bash example.sh
Enter a number:
34
No is even
user@murshid-tp:~/Desktop$ bash example.sh
Enter a number:
23
No is odd
user@murshid-tp:~/Desktop$
```

Looping Statements in Shell Scripting: There are total 3 looping statements which can be used in bash programming

1.while statement

2.for statement

3.until statement

Their descriptions and syntax are as follows:

while statement

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

Syntax

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

for statement

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

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

Syntax

```
for var in word1 word2 ...wordn

do
   Statement to be executed
done
```

until statement

The until loop is executed as many as times the condition/command evaluates to false. The loop terminates when the condition/command becomes true.

Syntax

until command
do
  Statement to be executed until command is true
done

<u>Example :</u>

```
#!/bin/sh

for i in 0 1 2 3 4 5 6 7
do
        echo $i
done
~
~
~
~
~
~
~
~
~
~
```

```
user@murshid-tp:~/Desktop$ vim example.sh
user@murshid-tp:~/Desktop$ bash example.sh
0
1
2
3
4
5
6
7
user@murshid-tp:~/Desktop$
```

Functions in Shell script

Although aliases are quick and easy to implement, they are quite limited in their scope. You'll find as you're trying to chain commands together that you can't access arguments given at runtime very well, among other things. Aliases can also be quite slow at times because they are read after all functions.

There is an alternative to aliases that is more robust and can help you bridge the gap between bash aliases and full shell scripts. These are called shell functions. They work in almost the same way as aliases but are more programmatic and accept input in a standard way.

We won't go into extensive detail here, because these can be used in so many complex situations and bash is an entire scripting language, but we'll go over some basic examples.

For starters, there are two basic ways to declare a bash syntax. The first uses the function command and looks something like this:

```
function function_name {
   command1
   <^>command2</^>
}
```

The other syntax uses a set of parentheses which is more "C-like":

```
function_name () {
   command1
   command2
}
```

We can compress this second form into one line and separate the commands with semicolons. A semicolon *must* come after the last command too:

```
function_name () { command1; command2; }
```

<u>Example</u>

```
#!/bin/sh

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

Experiment No : 5

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

1.Update your package

2.Install Lamp Stack

3.Copy link address of WordPress.

4. Change the directory to root directory of Apache.

5. Download Wordpress

```
user@murshid-tp:/var/www/html$ sudo wget https://wordpress.org/latest.tar.gz
--2021-08-09 16:11:50--  https://wordpress.org/latest.tar.gz
Resolving wordpress.org (wordpress.org)... 198.143.164.252
Connecting to wordpress.org (wordpress.org)|198.143.164.252|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 15073609 (14M) [application/octet-stream]
Saving to: 'latest.tar.gz'

latest.tar.gz                       100%[=================================================================================>]  14.38M  4.24MB/s    in 6.2s

2021-08-09 16:11:59 (2.33 MB/s) - 'latest.tar.gz' saved [15073609/15073609]

user@murshid-tp:/var/www/html$
```

6. Unzip the copied file of Wordpress

```
user@murshid-tp:/var/www/html$ ls
index.html  latest.tar.gz
user@murshid-tp:/var/www/html$ sudo tar xvfz latest.tar.gz
```

7 . Check the rights on the current directory files.

8. Change owner of the WordPress file.

9 .Check the rights of the WordPress file changed.

10. Open mysql in terminal and create database wordpress

11. Open word press configuration in browser, then enter the database connection details and click submit.

12 . Click Run the install

13. Complete the information.

14. Finally successfully installed! Click login.Enter user name and password

15.Successfully enter into the WordPress dashboard.

EXPIRIMENT NO : 6

Aim: Installation and configuration of common software frame works such as Laravel. (Student should acquire the capability to install and configure a modern framework)

SETUP LARAVEL ON UBUNTU WITH APACHE

Step-1: Install Apache Web Server

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

- After updating your package list install apache webserver. So, go ahead and type sudo apt install apache2 then hit the enter key. Press y key to proceed. You can also setup laravel with Nginx instead of the apache web server.

```
user@murshid-tp:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
apache2 is already the newest version (2.4.41-4ubuntu3.4).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
user@murshid-tp:~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Sat 2021-08-28 10:56:27 IST; 5 days ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 1043 (apache2)
      Tasks: 6 (limit: 4548)
     Memory: 11.3M
     CGroup: /system.slice/apache2.service
             ├─ 1043 /usr/sbin/apache2 -k start
             ├─19643 /usr/sbin/apache2 -k start
             ├─19644 /usr/sbin/apache2 -k start
             ├─19645 /usr/sbin/apache2 -k start
             ├─19646 /usr/sbin/apache2 -k start
             └─19647 /usr/sbin/apache2 -k start

Aug 28 10:56:16 murshid-tp systemd[1]: Starting The Apache HTTP Server...
Aug 28 10:56:27 murshid-tp apachectl[942]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using ::1. Set the 'ServerName' directive globall
Aug 28 10:56:27 murshid-tp systemd[1]: Started The Apache HTTP Server.
Aug 31 09:15:02 murshid-tp systemd[1]: Reloading The Apache HTTP Server.
Aug 31 09:15:05 murshid-tp apachectl[19611]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 2409:4073:38d:2741:a797:d527:ea9c:e08e. S
Aug 31 09:15:05 murshid-tp systemd[1]: Reloaded The Apache HTTP Server.
lines 1-21/21 (END)
```

- Check Apache server is working

Step-2: Install and Configure PHP 7.4

- To install Laravel 8.x, at least you must have PHP >= 7.3 on your system. And by default, the official Ubuntu 20.04 repository provides PHP 7.4 packages. Install PHP 7.4 packages using the apt command
- You can check your PHP version using it.

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

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

```
  GNU nano 4.8                                                    php.ini
; properly in multithreaded servers, such as IIS or Zeus, and is automatically
; disabled on them.
; http://php.net/enable-dl
enable_dl = Off

; cgi.force_redirect is necessary to provide security running PHP as a CGI under
; most web servers.  Left undefined, PHP turns this on by default.  You can
; turn it off here AT YOUR OWN RISK
; **You CAN safely turn this off for IIS, in fact, you MUST.**
; http://php.net/cgi.force-redirect
;cgi.force_redirect = 1

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

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

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

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

; FastCGI under IIS supports the ability to impersonate
; security tokens of the calling client.  This allows IIS to define the
; security context that the request runs under.  mod_fastcgi under Apache
; does not currently support this feature (03/17/2002)
; Set to 1 if running under IIS.  Default is zero.
; http://php.net/fastcgi.impersonate
;fastcgi.impersonate = 1

; Disable logging through FastCGI connection. PHP's default behavior is to enable
; this feature.
Search [cgi.fix]:
^G Get Help      M-C Case Sens    M-B Backwards    ^P Older         ^T Go To Line
^C Cancel        M-R Regexp       ^R Replace       ^N Newer         M-J FullJstify
```

Press Ctrl + x then y to Save and Exit.

Now Restart The apache service.

```
user@murshid-tp:/etc/php/7.4/apache2$
user@murshid-tp:/etc/php/7.4/apache2$
user@murshid-tp:/etc/php/7.4/apache2$ systemctl restart apache2
```

Step-3: Install Composer PHP Packages Management

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

```
Terminal
user@murshid-tp: ~
user@murshid-tp:~$ systemctl restart apache2
user@murshid-tp:~$
user@murshid-tp:~$
user@murshid-tp:~$ curl -sS https://getcomposer.org/installer | php
All settings correct for using Composer
Downloading...

Composer (version 2.1.6) successfully installed to: /home/user/composer.phar
Use it: php composer.phar

user@murshid-tp:~$ mv composer.phar /usr/local/bin/composer
```

You can check your installed composer version by typing the composer –version.

```
user@murshid-tp: ~
user@murshid-tp:~$ mv composer.phar /usr/local/bin/composer
user@murshid-tp:~$ composer --version
Deprecation Notice: Required parameter $path follows optional parameter $schema in /usr/share/php/JsonSchema/Constraints/UndefinedConstraint.php:62
Deprecation Notice: Required parameter $path follows optional parameter $schema in /usr/share/php/JsonSchema/Constraints/UndefinedConstraint.php:108
Deprecation Notice: Method ReflectionParameter::getClass() is deprecated in /usr/share/php/Composer/Repository/RepositoryManager.php:130
Deprecation Notice: Method ReflectionParameter::getClass() is deprecated in /usr/share/php/Composer/Repository/RepositoryManager.php:130
Composer 1.10.1 2020-03-13 20:34:27
user@murshid-tp:~$
```

Step-4: Install Laravel 8.x on Ubuntu 20.04

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

```
user@murshid-tp: ~
user@murshid-tp:~$ composer global require laravel/installer
```

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

```
user@murshid-tp: ~
user@murshid-tp:~$ sudo nano ~/.bashrc
sudo: /etc/sudoers is world writable
sudo: no valid sudoers sources found, quitting
sudo: unable to initialize policy plugin
user@murshid-tp:~$ sudo apt install appache2
sudo: /etc/sudoers is world writable
sudo: no valid sudoers sources found, quitting
sudo: unable to initialize policy plugin
user@murshid-tp:~$ nano ~/.bashrc
user@murshid-tp:~$ source ~/.bashrc
user@murshid-tp:~$ echo $PATH
/home/user/.config/composer/vendor/bin:/home/user/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
user@murshid-tp:~$
```

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

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

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

Step-5: Finally Configure Apache for Laravel and test it

- Add your project directory to www-data group use the following command

```
user@murshid-tp: ~
user@murshid-tp:~$ sudo chgrp -R www-data /home/murshid/myapp1
```

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

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

```
user@murshid-tp: ~
user@murshid-tp:~$ sudo chmod -R 775 /home/murshid/myapp1/storage
```

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

```
user@murshid-tp:~$ cd /etc/apache2/sites-available/
user@murshid-tp:/etc/apache2/sites-available$ ls
000-default.conf  default-ssl.conf
user@murshid-tp:/etc/apache2/sites-available$ sudo nano myapp1.com.conf
```

And type the following line inside the file.

```
Terminal                                                                    Sep 5  5:54 PM
user@murshid-tp: /etc/apache2/sites-available
  GNU nano 4.8                      myapp1.com.conf                                Modified
<VirtualHost *:80>
    ServerName myapp1.com

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

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

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

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     M-U Undo       M-A Mark Text  M-] To Bracket  M-Q Previous   ^B Back
^X Exit        ^R Read File   ^\ Replace     ^U Paste Text  ^T To Spell    ^_ Go To Line  M-E Redo       M-6 Copy Text  ^Q Where Was    M-W Next       ^F Forward
```

Now enable mod rewrite for apache2 just type

Now enable your site, just type
Finally, Restart the apache service, type

Type the following 3 command

sudo a2enmod rewrite.
sudo a2ensite myapp1.com.conf.
systemctl restart apache2

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

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

Experiment No : 7

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

Step 1: Get The Server Ready

First change the user to super user by **su** command

As a best practice, make sure your packages are up to date:

apt-get update -y

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

apt-get install build-essential -y

```
root@murshid-tp: /home/user
user@murshid-tp:~$ su
Password:
root@murshid-tp:/home/user# apt-get update -y
Hit:1 http://in.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://dl.google.com/linux/chrome/deb stable InRelease
Hit:3 http://archive.canonical.com/ubuntu focal InRelease
Hit:4 http://ppa.launchpad.net/costales/yaru-colors-folder-color/ubuntu focal InRelease
Get:5 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Hit:6 https://dl.winehq.org/wine-builds/ubuntu focal InRelease
Hit:7 http://ppa.launchpad.net/deadsnakes/ppa/ubuntu focal InRelease
Get:8 http://packages.microsoft.com/repos/code stable InRelease [10.4 kB]
Get:9 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:10 http://in.archive.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Hit:11 http://ppa.launchpad.net/ondrej/php/ubuntu focal InRelease
Get:12 http://packages.microsoft.com/repos/code stable/main armhf Packages [45.1 kB]
Get:13 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1,175 kB]
Get:14 http://packages.microsoft.com/repos/code stable/main arm64 Packages [45.0 kB]
Get:15 http://packages.microsoft.com/repos/code stable/main amd64 Packages [44.2 kB]
Get:16 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metadata [282 kB]
Get:17 http://in.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [853 kB]
Get:18 http://in.archive.ubuntu.com/ubuntu focal-updates/universe amd64 DEP-11 Metadata [351 kB]
Get:19 http://in.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [24.6 kB]
Get:20 http://in.archive.ubuntu.com/ubuntu focal-updates/multiverse i386 Packages [8,256 B]
Get:21 http://in.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 DEP-11 Metadata [944 B]
Get:22 http://in.archive.ubuntu.com/ubuntu focal-backports/universe amd64 DEP-11 Metadata [10.3 kB]
Get:23 http://in.archive.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [27.6 kB]
Get:24 http://in.archive.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 Metadata [61.1 kB]
Get:25 http://in.archive.ubuntu.com/ubuntu focal-security/multiverse amd64 DEP-11 Metadata [2,464 B]
Fetched 3,270 kB in 4s (747 kB/s)
Reading package lists... Done
root@murshid-tp:/home/user# apt-get install build-essential -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
build-essential is already the newest version (12.8ubuntu1.1).
build-essential set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
6 not fully installed or removed.
After this operation, 0 B of additional disk space will be used.
Setting up php8.0-fpm (8.0.10-1+ubuntu20.04.1+deb.sury.org+1) ...
Error: The new file /usr/lib/php/8.0/php.ini-production does not exist!
dpkg: error processing package php8.0-fpm (--configure):
 installed php8.0-fpm package post-installation script subprocess returned error exit status 1
dpkg: dependency problems prevent configuration of php-fpm:
 php-fpm depends on php8.0-fpm; however:
  Package php8.0-fpm is not configured yet.
```

Step 2: Download Dependencies

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

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

```
root@murshid-tp: /home/user
root@murshid-tp:/home/user# apt install build-essential dh-autoreconf libcurl4-gnutls-dev libexpat1-dev gettext libz-dev libssl-dev -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'zlib1g-dev' instead of 'libz-dev'
dh-autoreconf is already the newest version (19).
dh-autoreconf set to manually installed.
gettext is already the newest version (0.19.8.1-10build1).
gettext set to manually installed.
libexpat1-dev is already the newest version (2.2.9-1build1).
libexpat1-dev set to manually installed.
build-essential is already the newest version (12.8ubuntu1.1).
libssl-dev is already the newest version (1.1.1f-1ubuntu2.8).
libssl-dev set to manually installed.
zlib1g-dev is already the newest version (1:1.2.11.dfsg-2ubuntu1.2).
zlib1g-dev set to manually installed.
Suggested packages:
  libcurl4-doc libgnutls28-dev libidn11-dev libkrb5-dev libldap2-dev librtmp-dev libssh2-1-dev
The following NEW packages will be installed:
  libcurl4-gnutls-dev
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
6 not fully installed or removed.
Need to get 319 kB of archives.
After this operation, 1,526 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 libcurl4-gnutls-dev amd64 7.68.0-1ubuntu2.6 [319 kB]
Fetched 319 kB in 2s (129 kB/s)
Selecting previously unselected package libcurl4-gnutls-dev:amd64.
(Reading database ... 215333 files and directories currently installed.)
Preparing to unpack .../libcurl4-gnutls-dev_7.68.0-1ubuntu2.6_amd64.deb ...
Unpacking libcurl4-gnutls-dev:amd64 (7.68.0-1ubuntu2.6) ...
Setting up libcurl4-gnutls-dev:amd64 (7.68.0-1ubuntu2.6) ...
Setting up php8.0-fpm (8.0.10-1+ubuntu20.04.1+deb.sury.org+1) ...
Error: The new file /usr/lib/php/8.0/php.ini-production does not exist!
dpkg: error processing package php8.0-fpm (--configure):
 installed php8.0-fpm package post-installation script subprocess returned error exit status 1
dpkg: dependency problems prevent configuration of php-fpm:
 php-fpm depends on php8.0-fpm; however:
  Package php8.0-fpm is not configured yet.

dpkg: error processing package php-fpm (--configure):
 dependency problems - leaving unconfigured
Setting up libapache2-mod-php8.0 (8.0.10-1+ubuntu20.04.1+deb.sury.org+1) ...
No apport report written because the error message indicates its a followup error from a previous failure.
```

Step 3: Download The Source Package

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

We download git **v2.23.0** by using the command

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

```
root@murshid-tp:/home/user# wget https://github.com/git/git/archive/v2.23.0.tar.gz
--2021-09-04 11:35:35--  https://github.com/git/git/archive/v2.23.0.tar.gz
Resolving github.com (github.com)... 13.234.176.102
Connecting to github.com (github.com)|13.234.176.102|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/git/git/tar.gz/v2.23.0 [following]
--2021-09-04 11:35:36--  https://codeload.github.com/git/git/tar.gz/v2.23.0
Resolving codeload.github.com (codeload.github.com)... 13.127.152.42
Connecting to codeload.github.com (codeload.github.com)|13.127.152.42|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/x-gzip]
Saving to: 'v2.23.0.tar.gz'

v2.23.0.tar.gz                                  [                          <=>

2021-09-04 11:35:43 (1.44 MB/s) - 'v2.23.0.tar.gz' saved [8647535]

root@murshid-tp:/home/user#
```

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

tar -xvzf v2.23.0.tar.gz
cd git-2.23.0/

```
root@murshid-tp: /home/user
root@murshid-tp:/home/user# tar -xvzf v2.23.0.tar.gz
git-2.23.0/
git-2.23.0/.clang-format
git-2.23.0/.editorconfig
git-2.23.0/.gitattributes
git-2.23.0/.github/
git-2.23.0/.github/CONTRIBUTING.md
git-2.23.0/.github/PULL_REQUEST_TEMPLATE.md
git-2.23.0/.gitignore
git-2.23.0/.gitmodules
git-2.23.0/.mailmap
git-2.23.0/.travis.yml
git-2.23.0/.tsan-suppressions
git-2.23.0/COPYING
git-2.23.0/Documentation/
git-2.23.0/Documentation/.gitattributes
git-2.23.0/Documentation/.gitignore
git-2.23.0/Documentation/CodingGuidelines
git-2.23.0/Documentation/Makefile
git-2.23.0/Documentation/MyFirstContribution.txt
git-2.23.0/Documentation/RelNotes/
git-2.23.0/Documentation/RelNotes/1.5.0.1.txt
git-2.23.0/Documentation/RelNotes/1.5.0.2.txt
git-2.23.0/Documentation/RelNotes/1.5.0.3.txt
git-2.23.0/Documentation/RelNotes/1.5.0.4.txt
git-2.23.0/Documentation/RelNotes/1.5.0.5.txt
git-2.23.0/Documentation/RelNotes/1.5.0.6.txt
git-2.23.0/Documentation/RelNotes/1.5.0.7.txt
git-2.23.0/Documentation/RelNotes/1.5.0.txt
git-2.23.0/Documentation/RelNotes/1.5.1.1.txt
git-2.23.0/Documentation/RelNotes/1.5.1.2.txt
git-2.23.0/Documentation/RelNotes/1.5.1.3.txt
git-2.23.0/Documentation/RelNotes/1.5.1.4.txt
git-2.23.0/Documentation/RelNotes/1.5.1.5.txt
git-2.23.0/Documentation/RelNotes/1.5.1.6.txt
git-2.23.0/Documentation/RelNotes/1.5.1.txt
git-2.23.0/Documentation/RelNotes/1.5.2.1.txt
git-2.23.0/Documentation/RelNotes/1.5.2.2.txt
git-2.23.0/Documentation/RelNotes/1.5.2.3.txt
git-2.23.0/Documentation/RelNotes/1.5.2.4.txt
git-2.23.0/Documentation/RelNotes/1.5.2.5.txt
git-2.23.0/Documentation/RelNotes/1.5.2.txt
git-2.23.0/Documentation/RelNotes/1.5.3.1.txt
git-2.23.0/Documentation/RelNotes/1.5.3.2.txt
git-2.23.0/Documentation/RelNotes/1.5.3.3.txt
git-2.23.0/Documentation/RelNotes/1.5.3.4.txt
git-2.23.0/Documentation/RelNotes/1.5.3.5.txt
git-2.23.0/Documentation/RelNotes/1.5.3.6.txt
git-2.23.0/Documentation/RelNotes/1.5.3.7.txt
git-2.23.0/Documentation/RelNotes/1.5.3.8.txt
```

```
git-2.23.0/xdiff-interface.c
git-2.23.0/xdiff-interface.h
git-2.23.0/xdiff/
git-2.23.0/xdiff/xdiff.h
git-2.23.0/xdiff/xdiffi.c
git-2.23.0/xdiff/xdiffi.h
git-2.23.0/xdiff/xemit.c
git-2.23.0/xdiff/xemit.h
git-2.23.0/xdiff/xhistogram.c
git-2.23.0/xdiff/xinclude.h
git-2.23.0/xdiff/xmacros.h
git-2.23.0/xdiff/xmerge.c
git-2.23.0/xdiff/xpatience.c
git-2.23.0/xdiff/xprepare.c
git-2.23.0/xdiff/xprepare.h
git-2.23.0/xdiff/xtypes.h
git-2.23.0/xdiff/xutils.c
git-2.23.0/xdiff/xutils.h
git-2.23.0/zlib.c
root@murshid-tp:/home/user# cd git-2.23.0/
root@murshid-tp:/home/user/git-2.23.0# make configure
GIT_VERSION = 2.23.0
    GEN configure
root@murshid-tp:/home/user/git-2.23.0#
```

Step 4: Install Git

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

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

./configure –prefix=/usr

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

After that, we'll build the source code:

make all

```
root@murshid-tp:/home/user/git-2.23.0# make all
    * new build flags
    CC fuzz-commit-graph.o
    CC fuzz-pack-headers.o
    CC fuzz-pack-idx.o
    CC credential-store.o
    * new link flags
    CC common-main.o
    CC abspath.o
    CC advice.o
    CC alias.o
    CC alloc.o
    CC apply.o
    CC archive.o
    CC archive-tar.o
    CC archive-zip.o
    CC argv-array.o
    * new prefix flags
    CC attr.o
    CC base85.o
    CC bisect.o
    CC blame.o
    CC blob.o
    CC branch.o
    CC bulk-checkin.o
    CC bundle.o
    CC cache-tree.o
    CC chdir-notify.o
    CC checkout.o
    CC color.o
    CC column.o
    CC combine-diff.o
    CC commit.o
    CC commit-graph.o
    CC commit-reach.o
    CC compat/obstack.o
    CC compat/terminal.o
    CC config.o
    CC connect.o
    CC connected.o
    CC convert.o
    CC copy.o
```

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

make install

```
root@murshid-tp: /home/user/git-2.23.0
root@murshid-tp:/home/user/git-2.23.0# make install
    SUBDIR git-gui
    SUBDIR gitk-git
    SUBDIR templates
install -d -m 755 '/usr/bin'
install -d -m 755 '/usr/libexec/git-core'
install   git-credential-store git-daemon git-fast-import git-http-backend git-imap-send git-sh-i18n--envsubst git-shell git-remote-testsvn git-http-fetch git-http-push git-credentia
l-cache git-credential-cache--daemon git-remote-http git-remote-https git-remote-ftp git-remote-ftps git-bisect git-difftool--helper git-filter-branch git-merge-octopus git-merge-one
-file git-merge-resolve git-mergetool git-quiltimport git-legacy-stash git-request-pull git-submodule git-web--browse git-add--interactive git-archimport git-cvsexportcommit git-cvsi
mport git-cvsserver git-send-email git-svn git-p4 git-instaweb '/usr/libexec/git-core'
install -m 644  git-mergetool--lib git-parse-remote git-rebase--preserve-merges git-sh-setup git-sh-i18n '/usr/libexec/git-core'
install git git-upload-pack git-receive-pack git-upload-archive git-shell git-cvsserver '/usr/bin'
make -C templates DESTDIR='' install
make[1]: Entering directory '/home/user/git-2.23.0/templates'
install -d -m 755 '/usr/share/git-core/templates'
(cd blt && tar cf - .) | \
(cd '/usr/share/git-core/templates' && umask 022 && tar xof -)
make[1]: Leaving directory '/home/user/git-2.23.0/templates'
install -d -m 755 '/usr/libexec/git-core/mergetools'
install -m 644 mergetools/* '/usr/libexec/git-core/mergetools'
install -d -m 755 '/usr/share/locale'
(cd po/build/locale && tar cf - .) | \
(cd '/usr/share/locale' && umask 022 && tar xof -)
install -d -m 755 '/usr/share/perl5'
(cd perl/build/lib && tar cf - .) | \
(cd '/usr/share/perl5' && umask 022 && tar xof -)
make -C gitweb install
make[1]: Entering directory '/home/user/git-2.23.0/gitweb'
make[2]: Entering directory '/home/user/git-2.23.0'
make[2]: 'GIT-VERSION-FILE' is up to date.
make[2]: Leaving directory '/home/user/git-2.23.0'
    GEN gitweb.cgi
    GEN static/gitweb.js
install -d -m 755 '/usr/share/gitweb'
install -m 755 gitweb.cgi '/usr/share/gitweb'
install -d -m 755 '/usr/share/gitweb/static'
install -m 644 static/gitweb.js static/gitweb.css static/git-logo.png static/git-favicon.png '/usr/share/gitweb/static'
make[1]: Leaving directory '/home/user/git-2.23.0/gitweb'
make -C gitk-git install
make[1]: Entering directory '/home/user/git-2.23.0/gitk-git'
install -d -m 755 '/usr/bin'
install -m 755 gitk-wish '/usr/bin'/gitk
install -d -m 755 '/usr/share/gitk/lib/msgs'
install -m 644 po/pt_br.msg '/usr/share/gitk/lib/msgs' &&  install -m 644 po/de.msg '/usr/share/gitk/lib/msgs' &&  install -m 644 po/ja.msg '/usr/share/gitk/lib/msgs' &&  install -m
644 po/ca.msg '/usr/share/gitk/lib/msgs' &&  install -m 644 po/sv.msg '/usr/share/gitk/lib/msgs' &&  install -m 644 po/it.msg '/usr/share/gitk/lib/msgs' &&  install -m 644 po/ru.msg
'/usr/share/gitk/lib/msgs' &&  install -m 644 po/pt_pt.msg '/usr/share/gitk/lib/msgs' &&  install -m 644 po/hu.msg '/usr/share/gitk/lib/msgs' &&  install -m 644 po/vi.msg '/usr/share
/gitk/lib/msgs' &&  install -m 644 po/bg.msg '/usr/share/gitk/lib/msgs' &&  install -m 644 po/es.msg '/usr/share/gitk/lib/msgs' &&  install -m 644 po/fr.msg '/usr/share/gitk/lib/msgs
' && true
make[1]: Leaving directory '/home/user/git-2.23.0/gitk-git'
make -C git-gui gitexecdir='/usr/libexec/git-core' install
```

That's it! The last thing to do is to verify that git is working:

git –version

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

Experiment No : 8

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

Ipv4 Networking:

The operating system consists of various built-in, command-line networking utilities that are used for network troubleshooting.

IP is part of an internet protocol suite, which also includes the transmission control protocol. Together, these two are known as TCP/IP. The internet protocol suite governs rules for packetizing, addressing, transmitting, routing, and receiving data over networks.

IP addressing is a logical means of assigning addresses to devices on a network. Each device connected to the internet requires a unique IP address.

Most networks that handle internet traffic are packet-switched. Small units of data, called packets, are routed through a network. A source host, like your computer, delivers these IP packets to a destination host, such as a server, based on IP addresses in packet headers. Packet-switching allows many users on a network to share the same data path.

An IP address has two parts—-one part identifies the host, such as a computer or other device. And the other part identifies the network it belongs to. TCP/IP uses a subnet mask to separate them.

* IP (version 4) addresses are 32-bit integers that can be expressed in hexadecimal notation. The more common format, known as dotted quad or dotted decimal, is x.x.x.x, where each x can be any value between 0 and 255. For example, 192.0.2.146 is a valid IPv4 address.

IPv4 still routes most of today's internet traffic. A 32-bit address space limits the number of unique hosts to 232, which is nearly 4.3 billion IPv4 addresses for the world to use (4,294,967,296, to be exact).

Network Commands:

ping: Ping command is typically used for checking the **network connectivity** from your system to an end device like a server or a printer and also of a website. This command is used while troubleshooting the entire network. So, when you enter a URL in your web browser, what you are actually doing is instructing your machine to connect to the website name. The website name is actually an alias for the IP address. So this command can be used in two ways:

1. It can be used to ping a network IP address.

2. It can used to ping a website or hostname directly.

Route: Using the route command displays or modifies the computer's routing table. For a typical computer that has a single network interface and is connected to a local area network (LAN) that has a router, the routing table is pretty simple and isn't often the source of network problems. Still, if you're having trouble accessing other computers or other networks, you can use the route command to make sure that a bad entry in the computer's routing table isn't the culprit.

For a computer with more than one interface and that's configured to work as a router, the routing table is often a major source of trouble. Setting up the routing table properly is a key part of configuring a router to work.

Syntax:

route [-f] [-p] [command [destination] [mask subnetmask] [gateway] [metric costmetric]]

This section explains each of the options that you can use with

the route command.

The -f option clears the routing tables of all gateway entries. If you use the -f option in conjunction with one of the commands, the tables are cleared before you run the command.

By default, routes are not preserved when you restart the system. Use the -p option with the add command to make a route persistent. Use the -p option with the print command to view the list of registered persistent routes.

Traceroute

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.

The first column corresponds to the hop count. The second column represents the address of that hop and after that, you see three space-separated time in

milliseconds. *traceroute* command sends three packets to the hop and each of the time refers to the time taken by the packet to reach the hop.

Syntax:

```
traceroute [options]  host_Address [pathlength]
```

Options:

- -4 Option: Use ip version 4 i.e. use Ipv4
- -6 Option: Use ip version 6 i.e. use Ipv6
- -F Option: Do not fragment packet.

Nslookup:

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

Syntax:

nslookup [option]

Setting up static ip address

Step 1 : Step 1 : List all the interfaces in the system.Use the ip address command to define a static IP address on an interface.

Step 2 : To view the content of Netplan network configuration file, run the following

command:
cat /etc/netplan/01-network-manager-all.yaml

Step 3 : Click on the top right network icon and select settings of the network interface you wish to configure to use a static IP address on Ubuntu.

Click on the settings icon to start IP address configuration.

Step 4 : Select IPv4 tab.Select manual and enter your desired IP address, netmask, gateway and DNS settings. Once ready click Apply button.

Step 5 : Turn OFF and ON switch to apply your new network static IP configuration settings.Run the command ip address and click on the network settings icon once again to confirm your new static IP address settings.

Static IP Addresses

A static IP address is an IP address that always stays the same. If you have a web server, FTP server, or other Internet resource that must have an address that cannot change, you can get a static IP address from your ISP. A static IP address is usually more expensive than a dynamic IP address, and some ISPs do not supply static IP addresses. You must configure a static IP address manually.

Dynamic IP Addresses

A dynamic IP address is an IP address that an ISP lets you use temporarily. If a dynamic address is not in use, it can be automatically assigned to a different device. Dynamic IP addresses are assigned using either DHCP or PPPoE.

Subnet

A subnet, or subnetwork, is a segmented piece of a larger network. More specifically, subnets are a logical partition of an IP network into multiple, smaller network segments. The Internet Protocol (IP) is the method for sending

data from one computer to another over the internet. Each computer, or host, on the internet has at least one IP address as a unique identifier.

CIDR

CIDR stands for **Classless Inter-Domain Routing**. It is an IP address assigning method that improves the efficiency of address distribution. It is also known as supernetting that replaces the older system based on classes A, B, and C networks. By using a single CIDR IP address many unique IP addresses can be designated. CIDR IP address is the same as the normal IP address except that it ends with a slash followed by a number.
172.200.0.0/16 It is called IP network prefix.

Characteristics of CIDR

It dynamically allocates the IP addresses by using CIDR blocks on the requirement of the user based on certain rules. The assignment of the CIDR block is handled by the Internet Assigned Number Authority (IANA). CIDR block consists of IP addresses and it consists of some rules:

•All IP addresses which are allocated to host must be continuous.

•The block size must be of power 2 and equal to the total number of IP addresses.

•The size of the block must be divisible by the first IP address of the block.

Subnet Mask

A subnet mask is a 32 bits address used to distinguish between a network address and a host address in IP address. A subnet mask identifies which part of an IP address is the network address and the host address. They are not shown inside the data packets traversing the Internet. They carry the destination IP address, which a router will match with a subnet.

iptables

iptables is a user-space utility program that allows a system administrator to configure the IP packet filter rules of the Linux kernel firewall, implemented as different Netfilter modules. The filters are organized in different tables, which contain chains of rules for how to treat network traffic packets. Different kernel modules and programs are currently used for different protocols; iptables applies to IPv4, ip6tables to IPv6, arptables to ARP, and ebtables to Ethernet frames.

Configure and Set Up a Firewall on Ubuntu

UFW stands for Uncomplicated Firewall which acts as an interface to IPTABLES that simplifies the process of the configuration of firewalls it will be a very hard for a beginners to learns and configure the firewall rules where we will secure the network from unknown users are machines. UFW works on the policies we configure as rules.

- For this, we needed a non-root user with root permission on the machine.

Installing the UFW (Firewall)

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

sudo apt-get install ufw –y

Enabling the UFW (Firewall)

Below is the command to enable the UFW –
sudo ufw enable

Enabling the Default Policies

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

sudo ufw default deny incoming
sudo ufw default allow outgoing

Enabling SSH Connections

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

sudo ufw allow ssh

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

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

Checking the UFW (Firewall) Status

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

sudo ufw status

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

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

HTTP port 80

sudo ufw allow 80

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

sudo ufw status

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

sudo ufw allow https

sudo ufw allow ftp

<u>Enabling to Allow Specific Range of Ports</u>

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

sudo ufw allow 500:800/tcp

Enable to Allow specific IP Addresses

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

sudo ufw allow from 192.168.100.1

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

sudo ufw allow from 192.168.100.1 to any port 8080

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

sudo ufw allow from 192.168.0.0/24

Deny the Connections or Rules

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

sudo ufw deny http

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

sudo ufw deny from 192.168.2.1

Deleting the Rules

We can delete the rules in two ways one with the actual rules and other with the rules numbers.

Actual Rules

The rules can be deleted using the actual rule which we allowed using the allow command. Below is the command to delete the HTTP rules from UFW.

sudo ufw allow http

sudo ufw delete allow http

Rules Number

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

sudo ufw status numbered

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

sudo ufw delete 14

Experiment No : 9

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

Tcpdump

Tcpdump is a type of packet analyzer software utility that monitors and logs TCP/IP traffic passing between a network and the computer on which it is executed.
Tcpdump is an open-source network utility that is freely available under the BSD license. Tcpdump works on the command line interface and provides descriptions of packet content in several formats, depending on the command used.

Tcpdump is primarily a network monitoring and management utility that captures and records TCP/IP data on the run time. Tcpdump is designed to provide statistics about the number of packets received and captured at the operating node for network performance analysis, debugging and diagnosing network bottlenecks and other network oriented tasks.

Because it is a command line utility, data retrieved through tcpdump can vary. For example, when used with -A operator, it prints out each packet in ASCII format. Tcpdump is supported by most Unix-based operating systems, such as Linux, Mac OSX and BSD. The Windows variant of tcpdump is known as WinDump.

Installing tcpdump

Working with tcpdump command

1.To capture the packets of current network interface

sudo tcpdump

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

```
user@murshid-tp:~$ sudo tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on wlo1, link-type EN10MB (Ethernet), capture size 262144 bytes
17:37:04.233576 IP murshid-tp.55627 > 239.255.255.250.1900: UDP, length 171
17:37:04.236167 IP murshid-tp.52478 > _gateway.domain: 6087+ PTR? 250.255.255.239.in-addr.arpa. (46)
17:37:05.235001 IP murshid-tp.55627 > 239.255.255.250.1900: UDP, length 171
17:37:05.236935 IP _gateway.domain > murshid-tp.52478: 6087 NXDomain 0/1/0 (103)
17:37:05.238190 IP murshid-tp.37423 > _gateway.domain: 29395+ PTR? 23.132.168.192.in-addr.arpa. (45)
17:37:05.242625 IP _gateway.domain > murshid-tp.37423: 29395 NXDomain 0/0/0 (45)
17:37:05.244293 IP murshid-tp.43052 > _gateway.domain: 3824+ PTR? 116.132.168.192.in-addr.arpa. (46)
17:37:05.691439 IP _gateway.domain > murshid-tp.43052: 3824 NXDomain* 0/1/0 (105)
```

2. To capture packets from a specific network interface

sudo tcpdump -i wlp2s0

```
user@murshid-tp:~$ sudo tcpdump -i wlp2s0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on wlp2s0, link-type EN10MB (Ethernet), capture size 262144 bytes
17:38:20.568952 IP6 murshid-tp.47764 > sb-in-f188.1e100.net.https: Flags [.], ack 1476248505, win 501, options [nop,nop,TS val 3644289533 ec
r 756747006], length 0
17:38:20.571233 IP murshid-tp.53351 > _gateway.domain: 38283+ PTR? c.b.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.c.0.3.0.0.4.0.0.8.6.4.0.4.2.ip6.arpa.
 (90)
17:38:20.831480 IP6 murshid-tp.53056 > whatsapp-cdn6-shv-02-maa2.fbcdn.net.https: Flags [P.], seq 801622269:801622308, ack 814465788, win 40
13, options [nop,nop,TS val 3693925603 ecr 3443487060], length 39
17:38:20.831579 IP6 murshid-tp.53056 > whatsapp-cdn6-shv-02-maa2.fbcdn.net.https: Flags [P.], seq 39:79, ack 1, win 4013, options [nop,nop,T
S val 3693925603 ecr 3443487060], length 40
17:38:20.833416 IP6 sb-in-f188.1e100.net.https > murshid-tp.47764: Flags [.], ack 1, win 265, options [nop,nop,TS val 756793829 ecr 36441518
24], length 0
17:38:20.833462 IP _gateway.domain > murshid-tp.53351: 38283 2/0/0 PTR sb-in-f188.1e100.net., PTR sb-in-xbc.1e100.net. (148)
17:38:20.835610 IP murshid-tp.54488 > _gateway.domain: 26003+ PTR? 1.e.0.f.6.2.d.7.c.2.1.1.8.b.2.b.9.c.1.7.b.9.e.4.3.7.0.4.9.0.4.2.ip6.arpa.
 (90)
17:38:20.838290 IP _gateway.domain > murshid-tp.54488: 26003 NXDomain 0/0/0 (90)
17:38:20.840115 IP murshid-tp.49709 > _gateway.domain: 24937+ PTR? 116.132.168.192.in-addr.arpa. (46)
17:38:20.842436 IP _gateway.domain > murshid-tp.49709: 24937 NXDomain 0/0/0 (46)
17:38:20.843871 IP murshid-tp.58907 > _gateway.domain: 51810+ PTR? 23.132.168.192.in-addr.arpa. (45)
17:38:20.846678 IP _gateway.domain > murshid-tp.58907: 51810 NXDomain 0/0/0 (45)
17:38:20.848701 IP murshid-tp.45911 > _gateway.domain: 33591+ PTR? 7.6.1.0.0.0.0.0.c.0.0.b.e.c.a.f.1.d.1.0.7.3.2.f.0.8.8.2.3.0.a.2.ip6.arpa.
 (90)
17:38:20.895946 IP6 whatsapp-cdn6-shv-02-maa2.fbcdn.net.https > murshid-tp.53056: Flags [.], ack 39, win 870, options [nop,nop,TS val 344350
9086 ecr 3693925603], length 0
17:38:20.896002 IP6 whatsapp-cdn6-shv-02-maa2.fbcdn.net.https > murshid-tp.53056: Flags [P.], seq 1:40, ack 39, win 870, options [nop,nop,TS
 val 3443509086 ecr 3693925603], length 39
17:38:20.896002 IP6 whatsapp-cdn6-shv-02-maa2.fbcdn.net.https > murshid-tp.53056: Flags [.], ack 79, win 870, options [nop,nop,TS val 344350
9096 ecr 3693925603], length 0
17:38:20.896028 IP6 murshid-tp.53056 > whatsapp-cdn6-shv-02-maa2.fbcdn.net.https: Flags [.], ack 40, win 4013, options [nop,nop,TS val 36939
25668 ecr 3443509086], length 0
17:38:20.899539 IP _gateway.domain > murshid-tp.45911: 33591 1/0/0 PTR whatsapp-cdn6-shv-02-maa2.fbcdn.net. (139)
17:38:21.305447 IP6 whatsapp-cdn6-shv-02-maa2.fbcdn.net.https > murshid-tp.53056: Flags [P.], seq 40:87, ack 79, win 870, options [nop,nop,T
S val 3443509305 ecr 3693925668], length 47
17:38:21.305518 IP6 murshid-tp.53056 > whatsapp-cdn6-shv-02-maa2.fbcdn.net.https: Flags [.], ack 87, win 4013, options [nop,nop,TS val 36939
26077 ecr 3443509305], length 0
```

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

3. To display all available interfaces

sudo tcpdump -D

Wireshark

Wireshark is a network protocol analyzer, or an application that captures packets from a network connection, such as from your computer to your home office or the internet. Packet is the name given to a discrete unit of data in a typical Ethernet network.

Wireshark is the most often-used packet sniffer in the world. Like any other packet sniffer, Wireshark does three things:

1. **Packet Capture:** Wireshark listens to a network connection in real time and then grabs entire streams of traffic – quite possibly tens of thousands of packets at a time.

2. **Filtering:** Wireshark is capable of slicing and dicing all of this random live data using filters. By applying a filter, you can obtain just the information you need to see.

3. **Visualization:** Wireshark, like any good packet sniffer, allows you to dive right into the very middle of a network packet. It also allows you to visualize entire conversations and network streams.

Uses of Wireshark

It is used by network security engineers to examine security problems.

1.It allows the users to watch all the traffic being passed over the network.

2.It is used by network engineers to troubleshoot network issues.

3.It also helps to troubleshoot latency issues and malicious activities on your network.

4.It can also analyze dropped packets.

5.It helps us to know how all the devices like laptop, mobile phones, desktop, switch, routers, etc., communicate in a local network or the rest of the world.

Wireshark installation

Step 1: Update APT

First, as always, update and upgrade your APT through the following command.

$ sudo apt update

```
user@murshid-tp:~$ sudo apt update
[sudo] password for user:
Hit:1 http://in.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://dl.google.com/linux/chrome/deb stable InRelease
Get:3 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Hit:4 https://dl.winehq.org/wine-builds/ubuntu focal InRelease
Get:5 http://packages.microsoft.com/repos/code stable InRelease [10.4 kB]
Hit:6 http://archive.canonical.com/ubuntu focal InRelease
Hit:7 http://ppa.launchpad.net/costales/yaru-colors-folder-color/ubuntu focal InRelease
Get:8 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Hit:9 http://ppa.launchpad.net/deadsnakes/ppa/ubuntu focal InRelease
Get:10 http://in.archive.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:11 http://packages.microsoft.com/repos/code stable/main arm64 Packages [49.3 kB]
Hit:12 http://ppa.launchpad.net/git-core/ppa/ubuntu focal InRelease
Get:13 http://packages.microsoft.com/repos/code stable/main amd64 Packages [48.3 kB]
Get:14 http://packages.microsoft.com/repos/code stable/main armhf Packages [49.1 kB]
Get:15 http://in.archive.ubuntu.com/ubuntu focal-updates/main i386 Packages [544 kB]
Hit:16 http://ppa.launchpad.net/ondrej/php/ubuntu focal InRelease
Get:17 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1,256 kB]
Get:18 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metadata [283 kB]
Get:19 http://in.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [864 kB]
Get:20 http://in.archive.ubuntu.com/ubuntu focal-updates/universe i386 Packages [641 kB]
Get:21 http://in.archive.ubuntu.com/ubuntu focal-updates/universe amd64 DEP-11 Metadata [361 kB]
Get:22 http://in.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 DEP-11 Metadata [944 B]
Get:23 http://in.archive.ubuntu.com/ubuntu focal-backports/universe amd64 DEP-11 Metadata [10.4 kB]
Get:24 http://in.archive.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [29.0 kB]
Get:25 http://in.archive.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 Metadata [62.4 kB]
Get:26 http://in.archive.ubuntu.com/ubuntu focal-security/multiverse amd64 DEP-11 Metadata [2,468 B]
Fetched 4,540 kB in 5s (937 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
1 package can be upgraded. Run 'apt list --upgradable' to see it.
user@murshid-tp:~$
```

Step 2: Download and Install Wireshark

Now that Wireshark's latest version has been added to the APT, you can download and install it with the following command.

$ sudo apt-get install wireshark

Step 3: Enable Root Privileges

When Wireshark installs on your system, you will be prompted by the following window. As Wireshark requires superuser/root privileges to operate, this option asks to enable or disable permissions for all every user on the system. Press the “Yes” button to allow other users, or press the “No” button to restrict other users from using Wireshark.

Step 4: (Optional) Reconfigure Permission Settings

If you have selected “No” in the above scenario, then you can change this selection again by executing the following command, which will reconfigure the Wireshark permission settings.

$ sudo dpkq-reconfigure wireshark-common

Step 5: Launch Wireshark

In the terminal window, type the following command to start the Wireshark application.

$ sudo wireshark

Netcat

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

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

Installing netcat on linux:

sudo apt-get install netcat

Port scanning:

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

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

Syntax:

nc -vz IP_address port

Connection timed out:

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

Connection succeeded

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

Syntax:

nc -vt IP Address Port

Closing the connection

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

Experiment No : 10

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

Hypervisor

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

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

•Space

•Energy

•Maintenance requirements

There are two main hypervisor types, referred to as "Type 1" (or "bare metal") and "Type 2" (or "hosted"). A **type 1 hypervisor** acts like a lightweight operating system and runs directly on the host's hardware, while a **type 2 hypervisor** runs as a software layer on an operating system, like other computer programs.

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

Benefits of hypervisors

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

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

Virtual mechine

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

Virtual machine technology is used for many use cases across on-premises and cloud environments. More recently, public cloud

services are using virtual machines to provide virtual application resources to multiple users at once, for even more cost efficient and flexible compute.

What are virtual machines used for?

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

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

Advantages of virtual machines

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

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

KVM

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

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

Kimchi is KVM's web-based virtualization management tool

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

OpenQRM allows managing and controlling various data centre infrastructures.

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

oVirt is an open source virtualization management tool for KVM

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

Xen

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

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

Docker Container

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

Container images become containers at runtime and in the case of Docker containers - images become containers when they run on Docker Engine. Available for both Linux and Windows-based applications, containerized software will always run the same, regardless of the infrastructure. Containers isolate software from its environment and ensure that it works uniformly despite differences for instance between development and staging.

Docker containers that run on Docker Engine:

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

Installing Docker:

1) Updating the local repository

sudo apt update

2) Installing Docker

sudo apt install docker.io

```
user@murshid-tp:~$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools btrfs-progs cgroupfs-mount | cgroup-lite debootstrap docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd docker.io pigz runc ubuntu-fan
0 upgraded, 6 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/74.0 MB of archives.
After this operation, 359 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

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

3) Checking docker installation

docker –version

```
user@murshid-tp:~$ docker --version
Docker version 20.10.7, build 20.10.7-0ubuntu1~20.04.1
user@murshid-tp:~$
```

4) Starting docker service

* Start the Docker service by running:

sudo systemctl start docker

* Then, enable it to run at startup:

sudo systemctl enable docker

*To check the status of the service, run:

sudo systemctl status docker

The output should verify Docker is **active (running)**.

Use Docker on Ubuntu 20.04

The basic syntax for docker commands is:

sudo docker [option] [command] [argument]

Working With Docker Images

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

To download a new Docker image, use the command:

docker pull [image_name]

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

docker search ubuntu

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

docker images

Working With Docker Containers

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

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

docker run [image_name]

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

docker run -it [image_name]

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

docker container ps

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

docker container ps -a

```
user@murshid-tp:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:9ade9cc2e26189a19c2e8854b9c8f1e14829b51c55a630ee675a5a9540ef6ccf
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

user@murshid-tp:~$
```

```
user@murshid-tp:~$ sudo docker images
REPOSITORY    TAG       IMAGE ID       CREATED       SIZE
hello-world   latest    feb5d9fea6a5   9 days ago    13.3kB
user@murshid-tp:~$
```

Experiment No : 11

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

Ansible

Ansible is a software tool that provides simple but powerful automation for cross-platform computer support. It is primarily intended for IT professionals, who use it for application deployment, updates on workstations and servers, cloud provisioning, configuration management, intra-service orchestration, and nearly anything a systems administrator does on a weekly or daily basis. Ansible doesn't depend on agent software and has no additional security infrastructure, so it's easy to deploy.

Because Ansible is all about automation, it requires instructions to accomplish each job. With everything written down in simple script form, it's easy to do version control. The practical result of this is a major contribution to the "infrastructure as code" movement in IT: the idea that the maintenance of server and client infrastructure can and should be treated the same as software development, with repositories of self-documenting, proven, and executable solutions capable of running an organization regardless of staff changes.

While Ansible may be at the forefront of automation, systems administration, and DevOps, it's also useful to everyday users. Ansible allows you to configure not just one computer, but potentially a whole network of computers at once, and using it requires no programming skills. Instructions written for Ansible are human-readable. Whether you're entirely new to computers or an expert, Ansible files are easy to understand.

How Ansible works

In Ansible, there are two categories of computers: the *control node* and *managed nodes*. The control node is a computer that runs Ansible. There must be at least one control node, although a backup control node may also exist. A managed node is any device being managed by the control node.

Ansible works by connecting to nodes (clients, servers, or whatever you're configuring) on a network, and then sending a small program called an Ansible *module* to that node. Ansible executes these modules over SSH and

removes them when finished. The only requirement for this interaction is that your Ansible control node has login access to the managed nodes. SSH keys are the most common way to provide access, but other forms of authentication are also supported.

What Ansible does

The term *Ansible modules* sounds complex, but most of the complexity is handled by Ansible and not the user. An Ansible module is written to be a model of the desired state of a system, meaning that each module defines what should be true on any given managed node. For instance, if a systems administrator decides that all workstations in an organization should have LibreOffice version X.Z installed, then it's up to Ansible's packaging module to determine whether each node has LibreOffice X.Z on it. Should Ansible find a managed node with LibreOffice X.Y installed, then it detects the operating system and runs the necessary routine to update it to LibreOffice X.Z. In this way, every workstation in an organization can be updated overnight with the software supported by the IT department.

Ansible Features

1. Configuration Management

Ansible is designed to be very simple, reliable, and consistent for configuration management. If you're already in IT, you can get up and running with it very quickly. Ansible configurations are simple data descriptions of infrastructure and are both readable by humans and parsable by machines. All you need to start managing systems is a password or an SSH (Secure Socket Shell, a network protocol) key. An example of how easy Ansible makes configuration management: If you want to install an updated version of a specific type of software on all the machines in your enterprise, all you have to do is write out all the IP addresses of the nodes (also called remote hosts) and write an Ansible playbook to install it on all the nodes, then run the playbook from your control machine.

2. Application Deployment

Ansible lets you quickly and easily deploy multitier apps. You won't need to write custom code to automate your systems; you list the tasks required to be done by writing a playbook, and Ansible will figure out how to get your

systems to the state you want them to be in. In other words, you won't have to configure the applications on every machine manually. When you run a playbook from your control machine, Ansible uses SSH to communicate with the remote hosts and run all the commands (tasks).

3. Orchestration

As the name suggests, orchestration involves bringing different elements into a beautifully run whole operation—similar to the way a musical conductor brings the notes produced by all the different instruments into a cohesive artistic work. For example, with application deployment, you need to manage not just the front-end and backend services but the databases, networks, storage, and so on. You also need to make sure that all the tasks are handled in the proper order. Ansible uses automated workflows, provisioning, and more to make orchestrating tasks easy. And once you've defined your infrastructure using the Ansible playbooks, you can use that same orchestration wherever you need to, thanks to the portability of Ansible playbooks.

4. Security and Compliance

As with application deployment, sitewide security policies (such as firewall rules or locking down users) can be implemented along with other automated processes. If you configure the security details on the control machine and run the associated playbook, all the remote hosts will automatically be updated with those details. That means you won't need to monitor each machine for security compliance continually manually. And for extra security, an admin's user ID and password aren't retrievable in plain text on Ansible.

5. Cloud Provisioning

The first step in automating your applications' life cycle is automating the provisioning of your infrastructure. With Ansible, you can provision cloud platforms, virtualized hosts, network devices, and bare-metal servers.

Ansible Terms:

•**Controller Machine**: The machine where Ansible is installed, responsible for running the provisioning on the servers you are managing.

•**Inventory**: An initialization file that contains information about the servers you are managing.

•**Playbook**: The entry point for Ansible provisioning, where the automation is defined through tasks using YAML format.

•**Task**: A block that defines a single procedure to be executed, e.g. Install a package.

•**Module**: A module typically abstracts a system task, like dealing with packages or creating and changing files. Ansible has a multitude of built-in modules, but you can also create custom ones.

•**Role**: A pre-defined way for organizing playbooks and other files in order to facilitate sharing and reusing portions of a provisioning.

•**Play**: A provisioning executed from start to finish is called a play. In simple words, execution of a playbook is called a play.

•**Facts**: Global variables containing information about the system, like network interfaces or operating system.

•**Handlers**: Used to trigger service status changes, like restarting or stopping a service.

Benefits of Ansible

●**Free**: Ansible is an open-source tool.

●**Very simple to set up and use**: No special coding skills are necessary to use Ansible's playbooks (more on playbooks later).

●**Powerful**: Ansible lets you model even highly complex IT workflows.

●**Flexible**: You can orchestrate the entire application environment no matter where it's deployed. You can also customize it based on your needs.

●**Agentless**: You don't need to install any other software or firewall ports on the client systems you want to automate. You also don't have to set up a separate management structure.

- **Efficient**: Because you don't need to install any extra software, there's more room for application resources on your server.

Ansible playbooks

Playbooks are the files where Ansible code is written. Playbooks are written in YAML format. YAML stands for Yet Another Markup Language. **Playbooks** are one of the core features of Ansible and tell Ansible what to execute. They are like a to-do list for Ansible that contains a list of tasks.

Playbooks contain the steps which the user wants to execute on a particular machine. Playbooks are run sequentially. Playbooks are the building blocks for all the use cases of Ansible.

Playbook Structure

Each playbook is an aggregation of one or more plays in it. Playbooks are structured using Plays. There can be more than one play inside a playbook.

The function of a play is to map a set of instructions defined against a particular host.

YAML is a strict typed language; so, extra care needs to be taken while writing the YAML files. There are different YAML editors but we will prefer to use a simple editor like notepad++. Just open notepad++ and copy and paste the below yaml and change the language to YAML (Language → YAML).

While modules provide the means of accomplishing a task, the way you use them is through an Ansible *playbook*. A playbook is a configuration file written in YAML that provides instructions for what needs to be done in order to bring a managed node into the desired state. Playbooks are meant to be simple, human-readable, and self-documenting. They are also *idempotent*, meaning that a playbook can be run on a system at any time without having a negative effect upon it. If a playbook is run on a system that's already properly configured and in its desired state, then that system should still be properly configured after a playbook runs.A playbook can be very simple, such as this one that installs, as a privileged user, the Apache HTTP server on any node in an IT department's *webservers* group:

```
---
- name: Apache server installed
  hosts: webservers
  become: yes
```

Playbooks can also be very complex, with conditionals and variables. However, because most of the real work is done by Ansible modules, playbooks remain brief, readable, and clear even though they can orchestrate entire networks of managed nodes.

The Different YAML Tags

Let us now go through the different YAML tags. The different tags are described below −

name

This tag specifies the name of the Ansible playbook. As in what this playbook will be doing. Any logical name can be given to the playbook.

hosts

This tag specifies the lists of hosts or host group against which we want to run the task. The hosts field/tag is mandatory. It tells Ansible on which hosts to run the listed tasks. The tasks can be run on the same machine or on a remote machine. One can run the tasks on multiple machines and hence hosts tag can have a group of hosts' entry as well.

vars

Vars tag lets you define the variables which you can use in your playbook. Usage is similar to variables in any programming language.

tasks

All playbooks should contain tasks or a list of tasks to be executed. Tasks are a list of actions one needs to perform. A tasks field contains the name of the task. This works as the help text for the user. It is not mandatory but proves useful in debugging the playbook. Each task internally links to a piece of code called a module. A module that should be executed, and arguments that are required for the module you want to execute.