

**DEPARTMENT OF COMPUTER APPLICATION  
TKM COLLEGE OF ENGINEERING  
KOLLAM – 691005**



**20MCA136 - NETWORKING & SYSTEM ADMINISTRATION  
LAB**

**PRACTICAL RECORD BOOK**

**Second Semester MCA**

**2020-2021**

**Submitted by:**

**NAME : MOHAMMED RASIK**

**ROLL NO : 20MCA223**

**DEPARTMENT OF COMPUTER APPLICATION**  
**TKM COLLEGE OF ENGINEERING**  
**KOLLAM – 691005**



**Certificate**

This is a bonafide record of the work done by MOHAMMED RASIK in the Second Semester in NETWORKING & SYSTEM ADMINISTRATION LAB Course (20MCA136) towards the partial fulfillment of the degree of Master of Computer Applications during the academic year 2020-2021.

Staff Member in-charge

Examiner

.....

.....

## INDEX

	<b>Contents</b>	<b>Date</b>	<b>Page No.</b>
	<b>Cycle 1</b>		
Exp 1	Familiarisation to computer hardware	28/04/21	1
Exp 2	Specification of desktop and server computers	28/04/21	10
Exp 3	Installation of linux on virtual box in windows 10	30/04/21	11
Exp 4	Installation of windows on virtual box in ubuntu	05/05/21	19
Exp 5	Installation of windows operating system on a bare machine	21/05/21	22
Exp 6	Installation of linux over windows operating system	02/06/21	33
Exp 7	Installation of server system	07/05/21	42
	<b>Cycle 2</b>		
Exp 8	Study on command line text editors	02/06/21	45
Exp 9	Basic Linux Commands	02/06/21	47
Exp 10	Familiarisation with Vi Editor	04/06/21	79
Exp 11	Familiarisation with EMACS editor	09/06/21	102
Exp 12	Linux File System	16/06/21	112
	<b>Cycle 3</b>		
Exp 13	Familiarisation to linux shell and shell scripting	18/06/21	120
Exp 14	Linux shell scripting problems	06/08/21	131
	<b>Cycle 4</b>		
Exp 15	Installation of LAMP	13/08/21	142
Exp 16	Installation of Laravel	1/09/21	150
	<b>Cycle 5</b>		
Exp 17	Familiarisation to command Line tool for networking	03/09/21	154
Exp 18	Familiarisation of Static and Dynamic IP	03/09/21	171
Exp 19	Concept of subnets and CIDR address scheme	05/09/21	173
Exp 20	Concept of subnet mask	05/09/21	176
Exp 21	Setting up of firewall on LAN	07/09/21	178
Exp 22	Wireshark and TCP Dump	08/09/21	180
Exp 23	Analyse Packets using wireshark	08/09/21	196
	<b>Cycle 6</b>		
Exp 24	Familiarisation to hypervisors and Virtual machines	15/09/21	200
Exp 25	Familiarisation to Xen or KVM	15/09/21	202
Exp 26	Installation of software from source code	19/09/21	207
Exp 27	Deploy Linux virtual machine using Ansible Playbook	19/09/21	211

## **EXPERIMENT - 1**

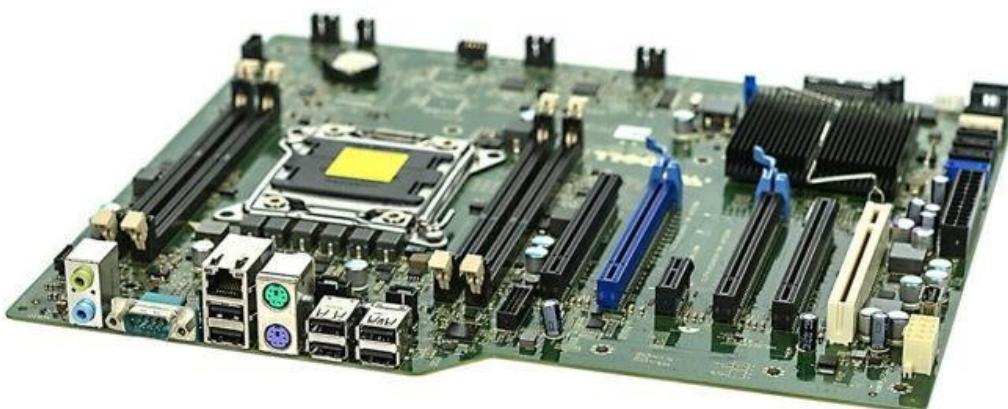
### **INTRODUCTION TO COMPUTER HARDWARES**

Computer hardware refers to the physical/tangible components that a computer system requires to function. It integrates all the input, output and functional devices with a circuit board that operates within a PC or laptop; including the motherboard, graphics card, CPU (Central Processing Unit), ventilation fans, webcam, power supply, and so on.

The design and size of computers may differ when compared between laptops and desktops but even so, all the core components remain the same in both. Without hardware, there would be no way of running the essential software that makes computers so useful. Software is defined as the virtual programs that run on your computer; that is, operating system, internet browser, word-processing documents, etc.

### **IMPORTANT HARDWARE COMPONENTS**

#### **MOTHERBOARD**



A motherboard is the main printed circuit board(PCB) in general-purpose computers and other expandable systems. It also known as mainboard, main circuit board, system board, baseboard, planar board, logic board, and in short mobo. It holds and allows communication between many of the crucial electronic components of a system, such as the central processing unit (CPU) and

memory, and provides connectors for other peripherals. A motherboard usually contains significant sub-systems, such as the central processor, the chipset's input/output

and memory controllers, interface connectors, and other components integrated for general use.

Motherboard means specifically a PCB with expansion capabilities. As the name suggests, this board is often referred to as the "mother" of all components attached to it.

## **CENTRAL PROCESSING UNIT & PROCESSOR (CPU)**

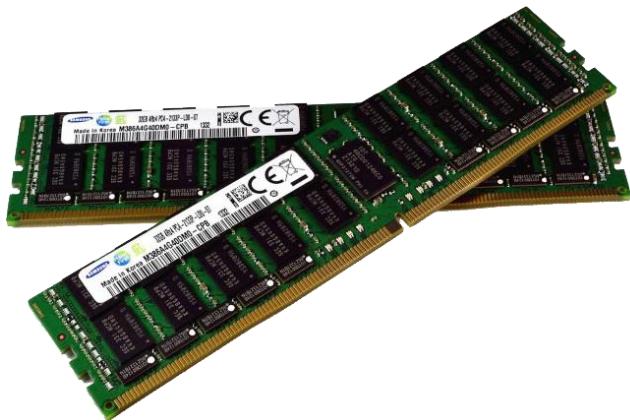


The computer's central processing unit (CPU) is the portion of a computer that retrieves and executes instructions. The CPU is essentially the brain of a CAD system. It consists of an arithmetic and logic unit (ALU), a control unit, and various registers. The CPU is often simply referred to as the processor. The ALU performs arithmetic operations , logic operations, and related operations, according to the program instructions.

The control unit controls all CPU operations, including ALU operations, the movement of data within the CPU, and the exchange of data and control signals across external interfaces (system bus). Registers are high-speed internal memory-storage units within the CPU.

The CPU Chip or Processor is responsible for processing all information from programs run by your computer. The 'clock speed', or the speed at which the processor processes information, is measured in gigahertz (GHz). This means that a processor advertising a high GHz rating will likely perform faster than a similarly specified processor of the same brand and age.

## RANDOM ACCESS MEMORY (RAM)



Random Access Memory, or RAM, is hardware found in the memory slots of the motherboard. The role of RAM is to temporarily store on-the-fly information created by programs and to do so in a way that makes this data immediately accessible. The tasks that require random memory could be; rendering images for graphic design, edited video or photographs, multi-tasking with multiple apps open (for example, running a game on one screen and chatting via Discord on the other).

## DAUGHTER CARD



A daughterboard (or daughter board , daughter card , or daughtercard ) is a circuit board that plugs into and extends the circuitry of another circuit board. The other circuit board may be the computer's main board (its motherboard ) or it may be another board or card that is already in the computer, often a sound card.

The daughter board is a computer hardware. It is also known as the piggyback board, riser card, daughter board, daughtercard or daughter card. A daughter board is a printed circuit board which is connected to the motherboard or expansion card. As compared to the motherboard, it is smaller in size. A daughter board does not act as an expansion card. An expansion card adds extra new

functions to the computer. But a daughter board that is connected to the motherboard adds or supports the main functions of the motherboard.

## **BUS SLOTS**

Bus slot or expansion port, an expansion slot is a connection or port inside a computer or motherboard or riser cards . 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.

Commonly founded are

- AMR - Modem, sound card.
- CNR - Modem, network card, sound card.
- EISA - SCSI, network card, video card.
- ISA - Network card, sound card, video card.
- PCI - Network card, SCSI, sound card, video card.
- PCI Express - Video card, modem, sound card, network card.
- VESA - Video card.

## **SMPS**



The working of SMPS is simply understood by knowing that the transistor used in LPS is used to control the voltage drop while the transistor in SMPS is used as a controlled switch.

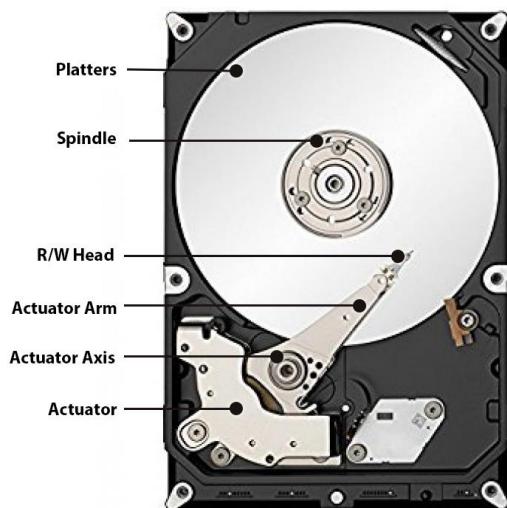
The disadvantages of LPS such as lower efficiency, the need for large value of capacitors to reduce ripples and heavy and costly transformers etc. are overcome by the implementation of Switched Mode Power Supplies.

## **READ-ONLY MEMORY (ROM)**

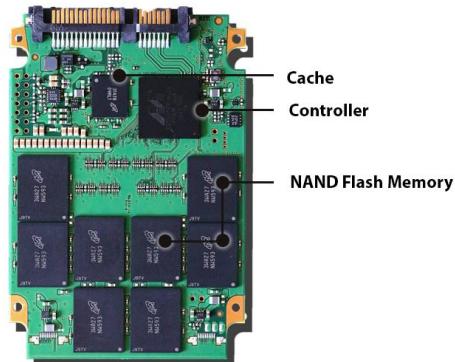
The ROM is also made of electronic microchips and is able to retain its content when power switches off. ROM is used for holding program instructions that can't be changed throughout the life of the computer because the content of ROM is impossible or very difficult to remove. For instance, ROM is used mainly for storing a boot program which is the instructions that the computer follows to perform self-diagnosis when it's switched on. This tells the computer how it will load the OS from secondary storage. ROM has different forms which include PROM (Programmable Read-Only Memory), EPROM (Erasable Programmable Read-Only Memory,) and EEPROM (Electrically Erasable Programmable Read-Only Memory).

## HARD DRIVE

**HDD**  
3.5"



**SSD**  
2.5"



The hard drive is a storage device responsible for storing permanent and temporary data. This data comes in many different forms, but is essentially anything saved or installed to a computer: for example, computer programs, family photos, operating system, word-processing documents, and so on. Find out more about hard drives and how they work. There are two different types of storage devices: the traditional hard disk drive (HDD) and the newer solid state drives (SSD). Hard disk drives work by writing binary data onto spinning magnetic disks called platters that rotate at high speeds, while a solid-state drive stores data by using static flash memory chips. Find out more about computer storage and how solid state drives work.

## GRAPHICS PROCESSING UNIT (GPU)



Especially important for 3D rendering, the GPU does exactly what its name suggests and processes huge batches of graphic data. You will find that your computer's graphics card has at least one GPU. As opposed to the basic on-board graphic capabilities that PC motherboards supply, dedicated graphics cards interface with the motherboard via an expansion slot to work almost exclusively on graphic rendering. This also means you can upgrade your graphics card if you want to get a bit more performance from your PC, not only this, but modern GPUs fulfil a broad computational workload beyond just rendering, making them an extension to the central processing unit.

## INTERFACING PORTS



A connection point that acts as interface between the computer and external devices like mouse, printer, modem, etc. is called port. Ports are of two types –

- **Internal port** – It connects the motherboard to internal devices like hard disk drive, CD drive, internal modem, etc.
- **External port** – It connects the motherboard to external devices like modem, mouse, printer, flash drives, etc.

Some of them are given below

## **SERIAL PORT**



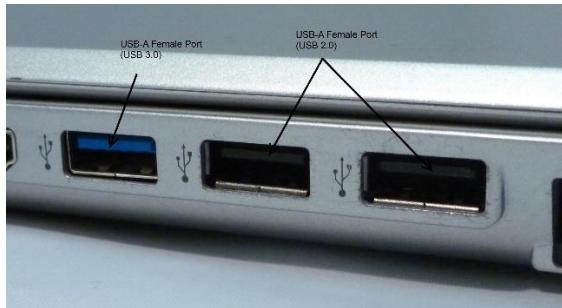
Serial ports transmit data sequentially one bit at a time. So they need only one wire to transmit 8 bits. However it also makes them slower. Serial ports are usually 9-pin or 25-pin male connectors. They are also known as COM (communication) ports or RS323C ports.

## **PARALLEL PORT**



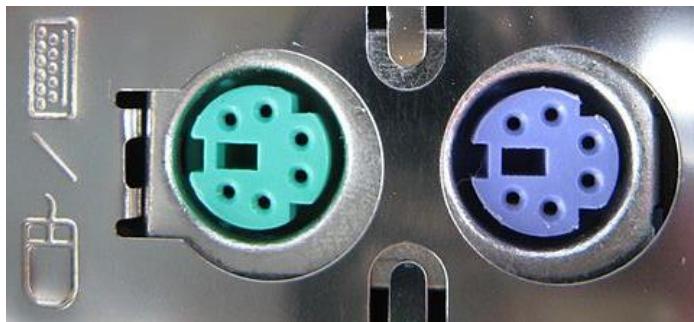
Parallel ports can send or receive 8 bits or 1 byte at a time. Parallel ports come in form of 25-pin female pins and are used to connect printer, scanner, external hard disk drive, etc.

## **USB PORT**



USB stands for Universal Serial Bus. It is the industry standard for short distance digital data connection. USB port is a standardized port to connect a variety of devices like printer, camera, keyboard, speaker, etc.

## **PS/2 PORTS**



PS/2 stands for Personal System/2. It is a female 6-pin port standard that connects to the male mini-DIN cable. PS/2 was introduced by IBM to connect mouse and keyboard to personal computers. This port is now mostly obsolete, though some systems compatible with IBM may have this port.

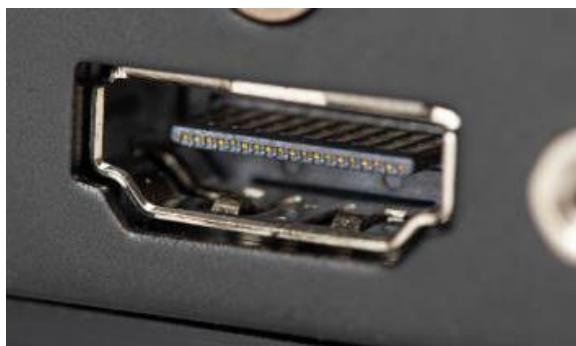
## **VGA PORT**



This port is commonly found in computers, projectors, and high definition TVs. It is a D-sub connector called DR-15 as it has 15 pins, which are arranged in 3 rows with five pins in each row. It was most often used to connect CPU with CRT monitors. Still, most of the LCD and LED monitors come with VGA ports. However, these ports don't assure high picture quality as VGA can carry only analogue video signals up to a resolution of 648X480.

As the demand and emphasis on video quality kept growing, the VGA ports were gradually replaced by more advanced ports that can assure high video quality such as HDMI and Display Ports.

## **HDMI PORT**



HDMI (High Definition Media Interface) is a digital interface developed to connect high definition devices such as digital cameras, gaming consoles, etc., to computers and TVs with HDMI ports. Besides this, it can carry uncompressed video and uncompressed or compressed audio signals. The advanced version of HDMI, such as 2.0, can transfer video signals of up to a resolution of 4096x2160.

## **EXPERIMENT - 2**

### **COMPUTER HARDWARE SPECIFICATIONS**

Computer hardware specifications are technical descriptions of the computer's components and capabilities.

- Processor speed, model and manufacturer. Processor speed is typically indicated in gigahertz (GHz). The higher the number, the faster the computer.
- Random Access Memory (RAM), This is typically indicated in gigabytes (GB). The more RAM in a computer the more it can do simultaneously.
- Hard disk (sometimes called ROM) space. This is typically indicated in gigabytes (GB) and refers generally to the amount of information (like documents, music and other data) your computer can hold.
- Other specifications might include network (ethernet or wi-fi) adapters or audio and video capabilities.

## **EXPERIMENT-3**

### **LINUX INSTALLATION USING VIRTUAL MACHINE ON WINDOWS 10**

#### **REQUIREMENTS**

- Good internet connection to download software and Linux ISO.
- Windows system with at least 12 GB of free space.
- Windows system with 4GB of RAM. (It can work with less RAM as well, but your system will start to lag while using Linux in the virtual machine.)
- Make sure to enable virtualization in the BIOS

#### **STEPS**

##### **Step 1: Download and install VirtualBox**

- Go to the website of Oracle VirtualBox and get the latest stable version from here:  
<https://www.virtualbox.org/wiki/Downloads>
- Just double-click on the downloaded .exe file and follow the instructions on the screen.
- It is like installing any regular software on Windows.



##### **Step 2: Download the Linux ISO**

- Next, you need to download the ISO file of the Linux distribution.
- You can get this image from the official website of the Linux distribution you are trying to use.
- I am using Ubuntu in this example, and you can download ISO images for Ubuntu from the link below:

### Creating a VirtualBox VM:

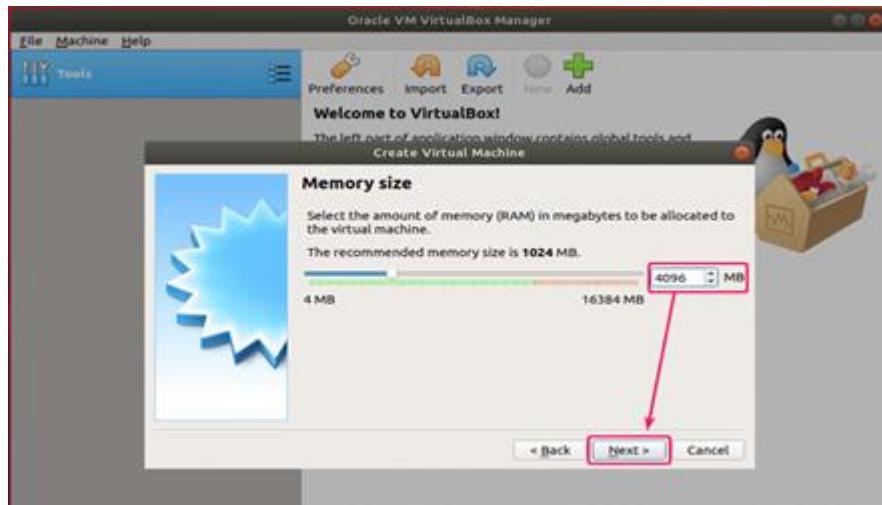
In this section, steps are shown how to create a VirtualBox VM for installing Ubuntu 20.04 LTS.

- Open VirtualBox.
- Then, click on Machine > New...
- Now, type in a name for the VM, select Linux from the Type dropdown menu, and Ubuntu (64-bit) from the Version dropdown menu. Then, click on Next >.



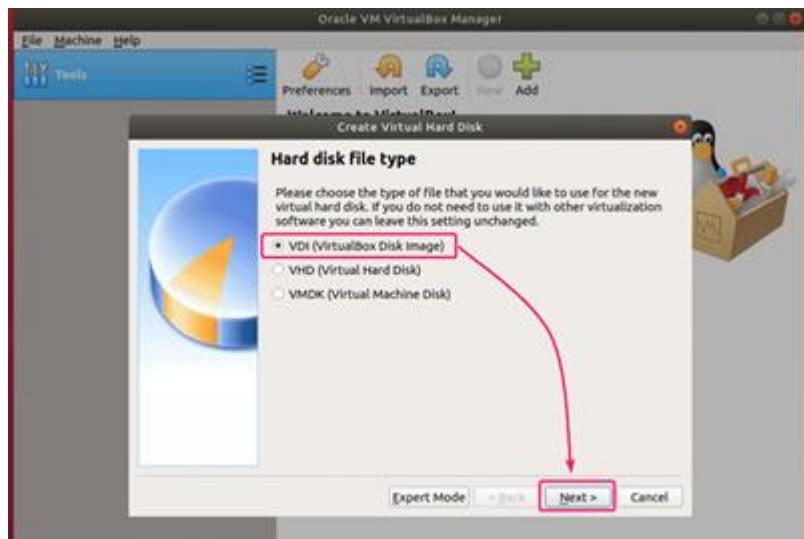
Now, you have to set the memory size for the VM.

- For Ubuntu Desktop 20.04 LTS, it should be at least 2048 MB (2 GB).
- For Ubuntu Server 20.04 LTS, it should be at least 512 MB.
- Once you're done, click on Next >.

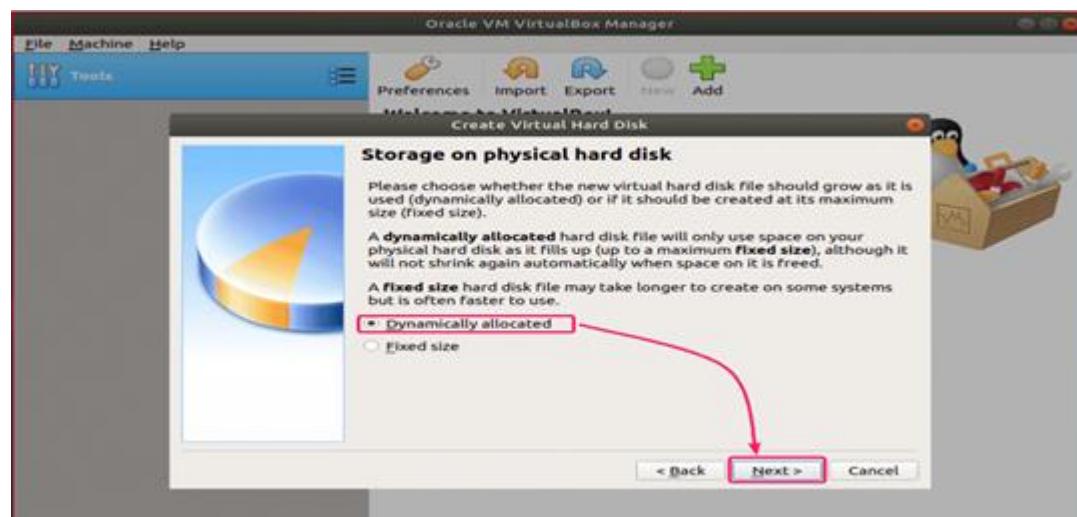


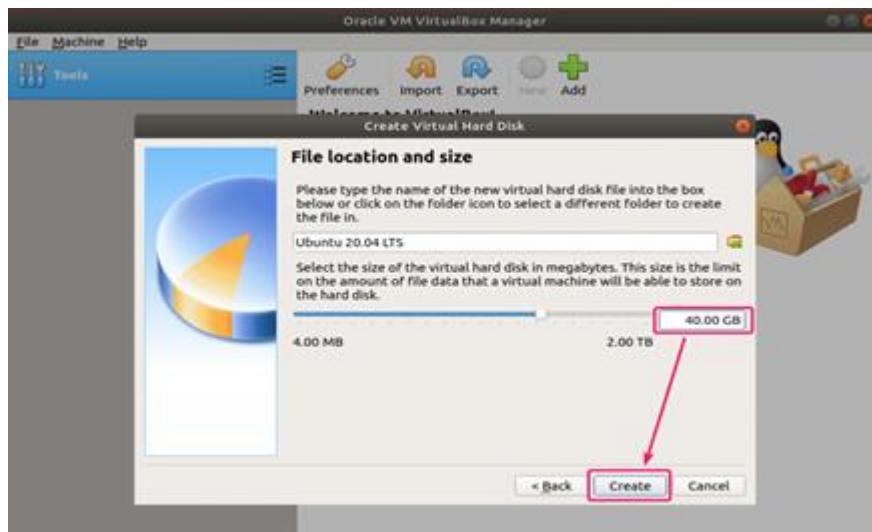
- Now, select Create a virtual hard disk now and click on Create . This serves as the hard disk of the virtual Linux system.
- It is where the virtual system will store its files.





- select Dynamically allocated and click on Next >.
- You can choose either the “Dynamically allocated” or the “Fixed size” option for creating the virtual hard disk.

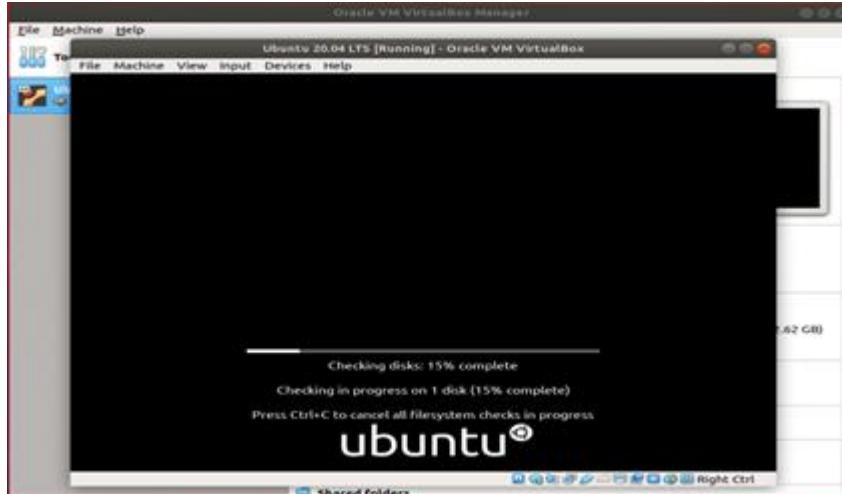




- set the virtual hard disk size for the VM.
- It should be at least 20.0 GB.
- Once you're done, click on Create

Starting the VM:

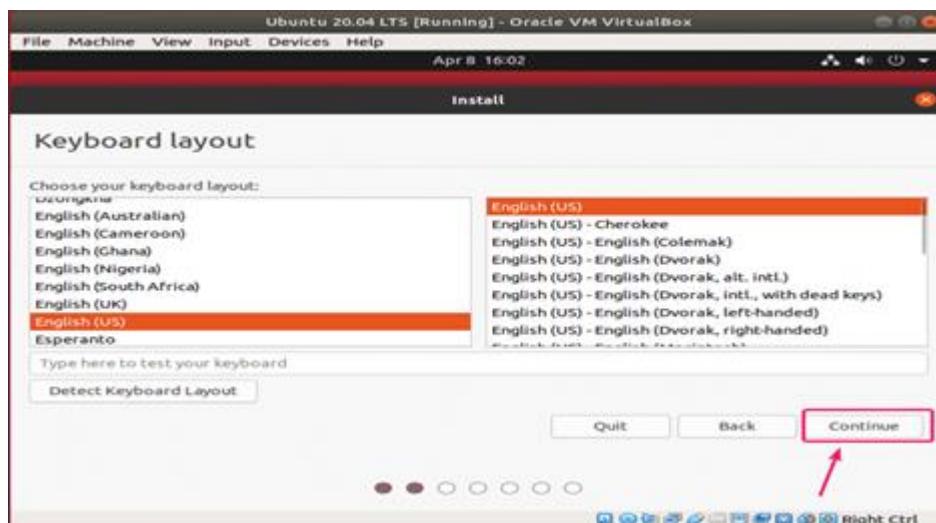
Now, select the VM and click on Start.



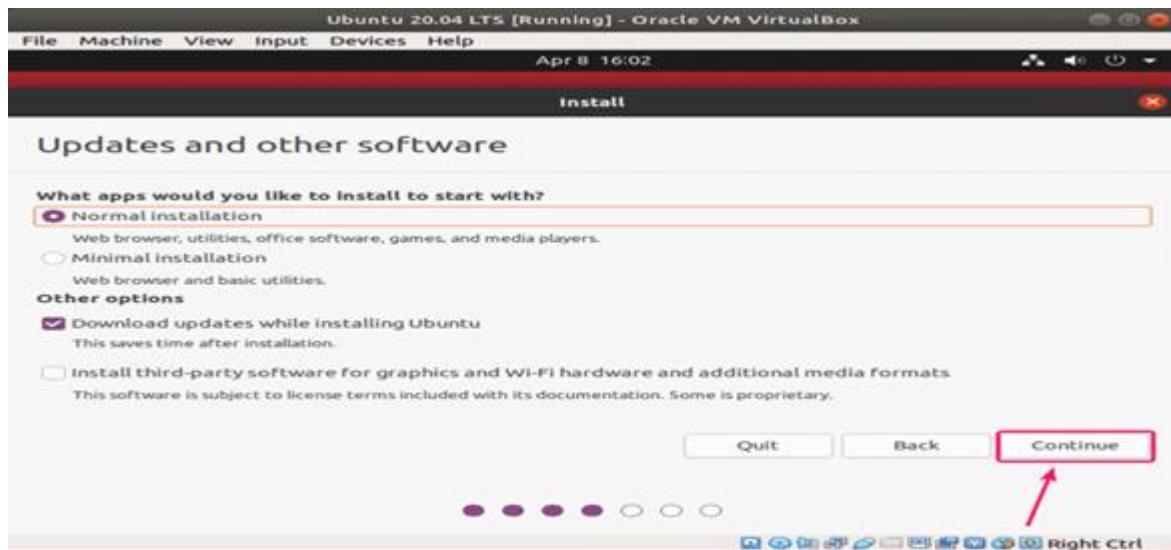
The VM should start and boot from the Ubuntu 20.04 LTS ISO image.



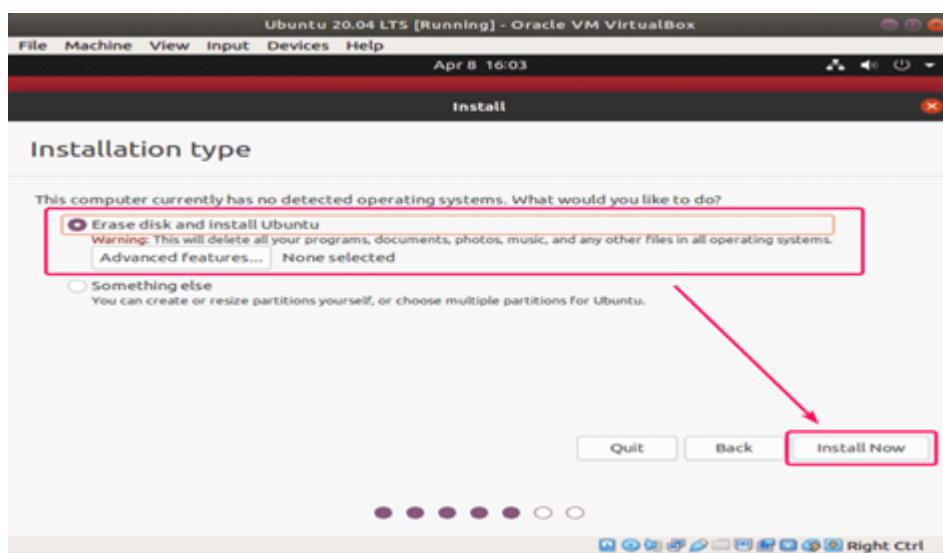
- From here, you can do a Normal installation or Minimal installation.
- Normal installation comes with all the apps as usual.
- Minimal installation comes with a limited number of apps. It saves a lot of disk spaces.
- If you have internet connection on your computer, you can check Download updates while installing Ubuntu to download all the necessary updates while installing Ubuntu on your computer.



Once you're done, click on Continue.



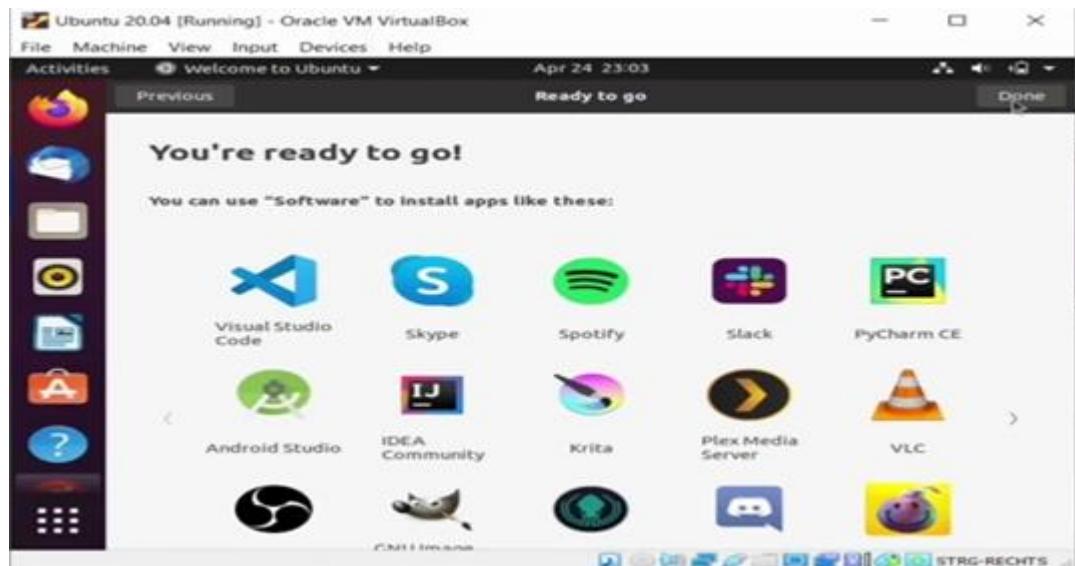
- As this is a VM, I won't go through the trouble of manually partitioning the hard drive.
- Just select Erase disk and install Ubuntu and click on Install Now.
- The Ubuntu installer will automatically create all the necessary partitions in your virtual hard drive.



Try to choose a password that you can remember.

You can also reset the password in Ubuntu if you forget it.

After creating an account , you can start using Ubuntu.



## EXPERIMENT-4

### INSTALLATION OF WINDOWS ON VIRTUAL BOX IN UBUNTU

#### **Step 1: Download Windows 10 ISO**

First and foremost, you need to download a Windows 10 ISO. You can download Windows 10 32-bit or 64-bit, depending on your system.

#### **Step 2: Install VirtualBox on Ubuntu**

Install VirtualBox on Ubuntu using the command `sudo apt install virtualbox`

```
File Edit View Search Terminal Help
rony@rony-HP-Laptop-14s-cr2xxx:~$ sudo apt install virtualbox
[sudo] password for rony:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  ffmpeg libatomic1:i386 libavresample4 libbsd0:i386 libdrm-amdgpu1:i386
  libdrm-intel1:i386 libdrm-nouveau2:i386 libdrm-radeon1:i386 libdrm2:i386
  libedit2:i386 libelf1:i386 libexpat1:i386 libffi7:i386 libglib1:i386
  libgl1-mesa-dri:i386 libgl1-mesa-glx:i386 libglapi-mesa:i386 libglvnd0:i386
```

#### **Step 3: Install Windows 10 in VirtualBox**

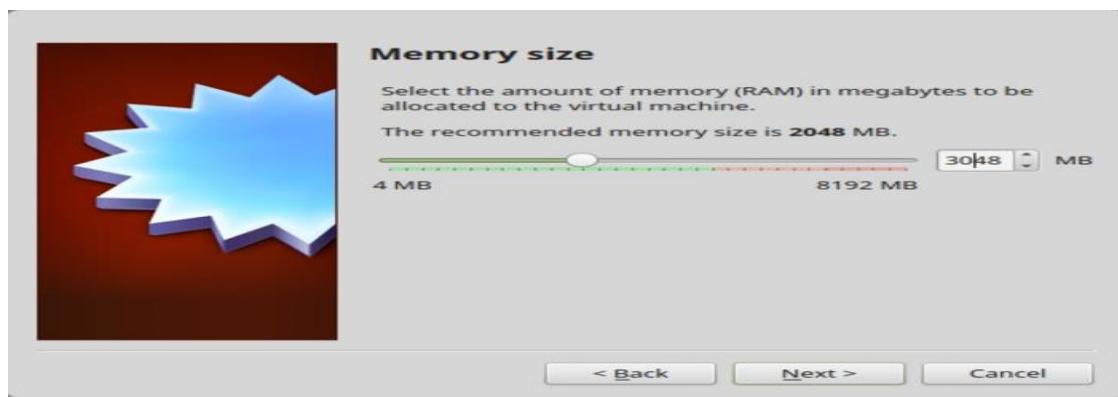
- Start VirtualBox. You should see a screen like the one below.



- Click on new and name the VM anything you like. Also select the operating system and version. In this case I've chosen Windows 10 and 64-bit.



- Choose the RAM size. The recommended RAM size for 64-bit Windows 10 is 2 GB, while for 32-bit Windows 10 it's 1 GB. But I suggest you have a little more than that for a smoother experience.



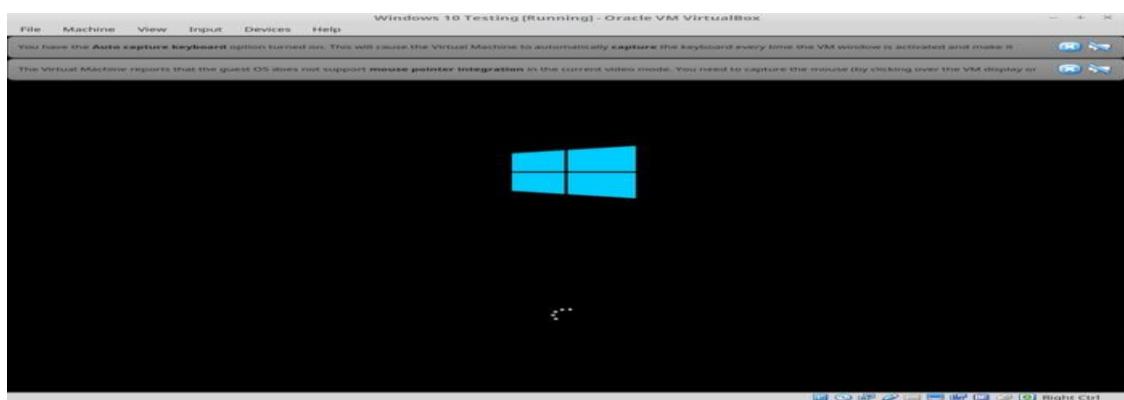
- Next is the size for the virtual machine. If you're crunched on space choose the recommended size, otherwise make it a little more than the recommended size.



- Select Create a virtual disk now, for the format, go ahead with the VDI format.



- Click on start to install Windows 10



## **EXPERIMENT-5**

### **INSTALLATION OF WINDOWS OPERATING SYSTEM ON A BARE MACHINE**

#### **INTRODUCTION**

Our Objective is to install Windows 10 Operating system to our PC/Machine via a bootable medium. We will be installing the OS from scratch assuming that we want to update/format/fix the OS in our system by making a fresh installation.

#### **Requirements :**

1. PC/Machine with at least 4GB RAM
2. An 8GB DVD or USB Flash drive.
3. Rufus software ([https://rufus.ie/en\\_US/](https://rufus.ie/en_US/)).
4. Required Windows 10 OS/OEM.
5. Strong stable internet connection.

#### **STEP 1 : ACQUIRING REQUIRED OS/OEM**

Before installing Windows 10, we must decide which OS version image we should install in our computer. Most users have the option to choose between the official Windows image from Microsoft or the official OEM Image from their Machine manufacturer like. HP, Dell, Asus and so on.

#### **Acquiring Official Windows 10 OS from Microsoft :**

1. Go to the Windows 10 official page and select the latest build of the os in dropdown menu and press ‘confirm’.[https://www.microsoft.com/en-us/software\\_download/windows10ISO](https://www.microsoft.com/en-us/software_download/windows10ISO).

**Download Windows 10 Disc Image (ISO File)**  
 Before updating, please refer to the Windows release information status for known issues to confirm your device is not impacted.

You've arrived here to this page because the operating system you're using hasn't support the Windows 10 media creation tool, and we want to make sure you can download Windows 10. To use the media creation tool visit the Microsoft Software Download Windows 10 page from a Windows 7, Windows 8.1 or Windows 10 device.

You can use this page to download a disc image (.ISO file) that can be used to install or embed Windows 10. This image can also be used to create deployment media using a USB flash drive or DVD.



[Before you begin](#)

Select edition

Windows 10 editions available for both Windows 10 Home and Windows 10 Pro:

Windows 10 Pro

[Confirm](#)

2. A new dropdown menu appears which allows us to choose the preferred language for our OS. Please choose the required language from the same and press ‘confirm’.

### Select the product language

You'll need to choose the same language when you install Windows. To see what language you're currently using, go to Time and language in PC settings or Region in Control Panel.

English International ▾

[Confirm](#)

3. Next we see a menu to download the 32-bit & 64-bit architecture versions of the OS for download. Please verify your hardware and download the required version.

### Downloads

Choose a link below to begin the download. If not sure which one to choose then refer to [FAQ](#).

#### Windows 10 English International

[32-bit Download](#)

[64-bit Download](#)

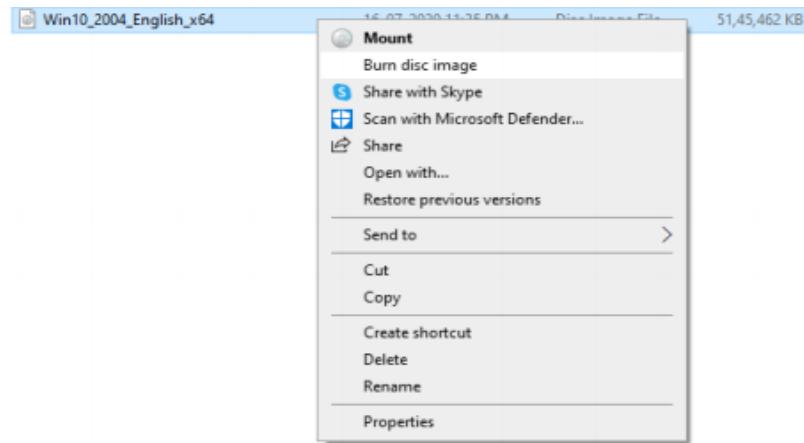
4. The download takes time as the image file is usually 5-6 GB in size. Once the download is complete, we will have file with extension (.iso).

### STEP 2: PREPARING BOOTABLE MEDIUM

To install the OS onto a machine, we need a bootable medium containing the installation files. The two common media are bootable Flash drive & bootable DVD. We must prepare either one of them in order to install the OS.

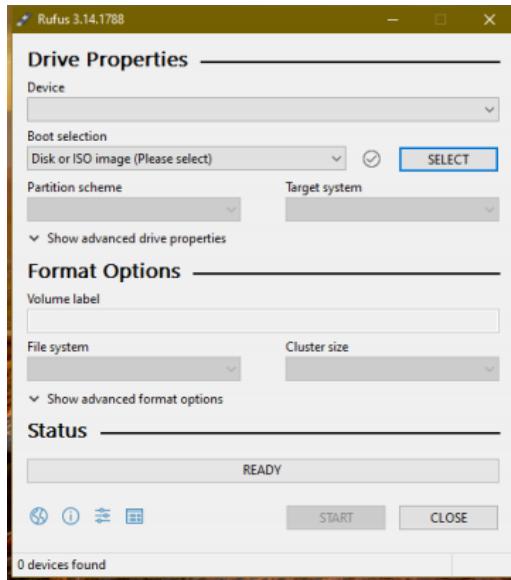
### Preparing Bootable DVD:

In order to create a bootable medium, we must burn the downloaded disk image of the OS (.iso) onto an empty DVD. Use either built-in functionality or 3<sup>rd</sup>-party software for the same.



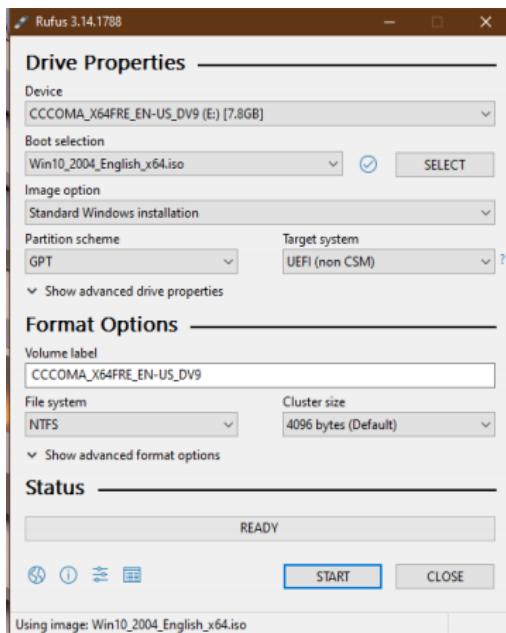
### Preparing Bootable Pen Drive:

1. Download the Rufus software from the official website.  
([https://rufus.ie/en\\_US/](https://rufus.ie/en_US/))&(<https://github.com/pbatard/rufus/releases/download/v3.14/rufus-3.14.exe>)



2. Open the Rufus software and the below UI opens.

3. Insert an empty flash drive with at least 8GB space into your computer and it will appear in the Rufus UI. Now, click the 'Select' button and choose the downloaded Windows OS Image (.iso) and click 'start' to initiate the process. Remember the flash drive will be formatted on conversion to bootable medium.

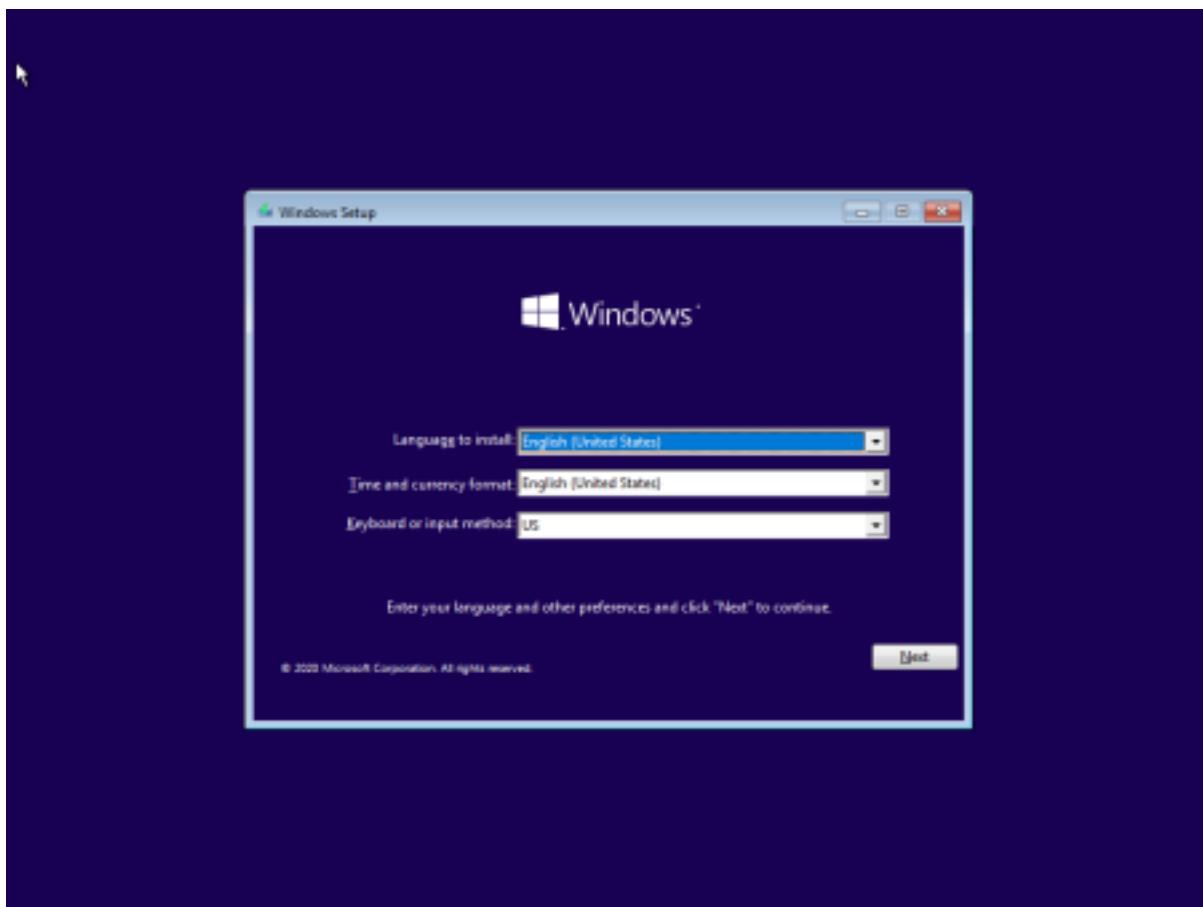


4. After the process finishes, the software will notify us the same and our flash drive is now successfully converted into a bootable medium.

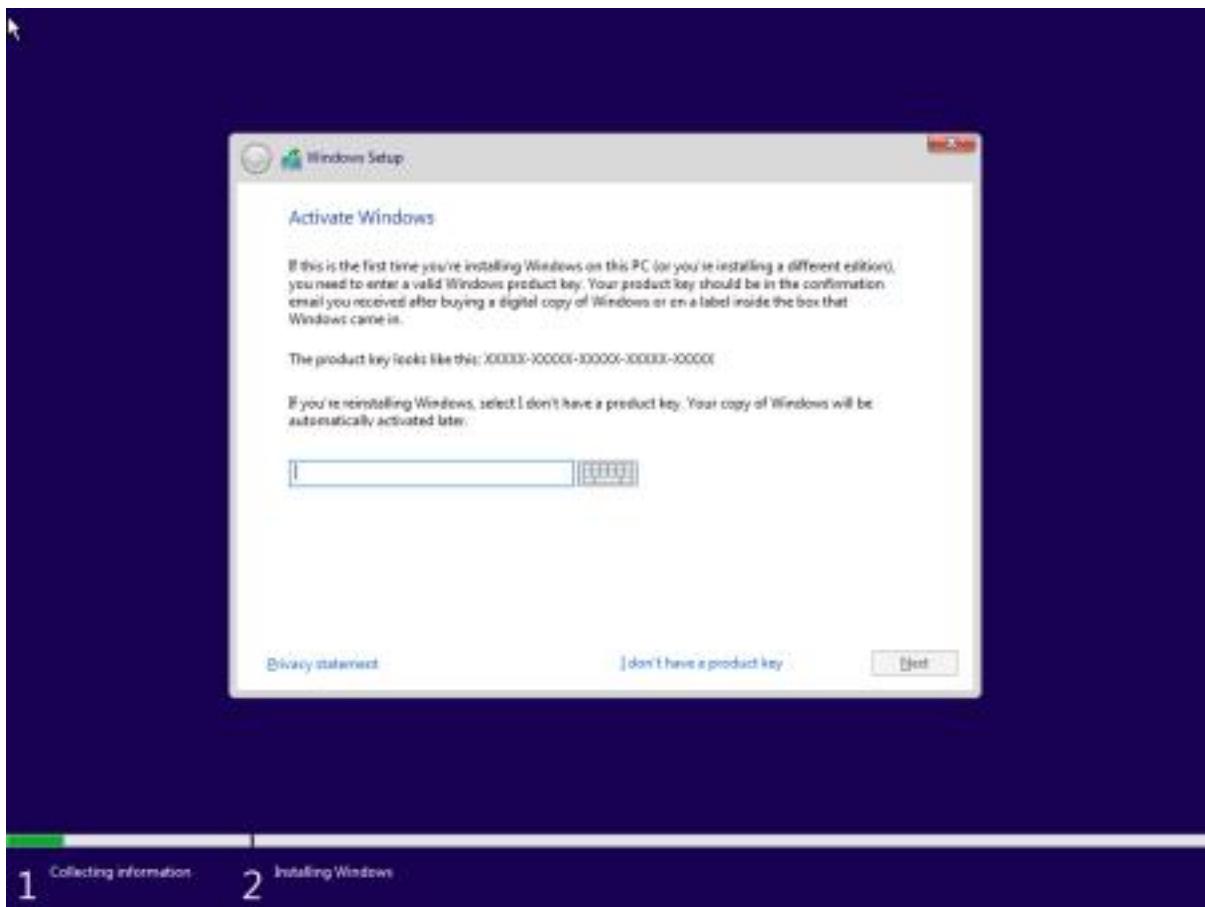
### STEP 3 : SYSTEM BOOTING & INSTALLATION

Once the bootable medium is ready, we can use it to boot the machine/PC and install the OS to the hard-disk.

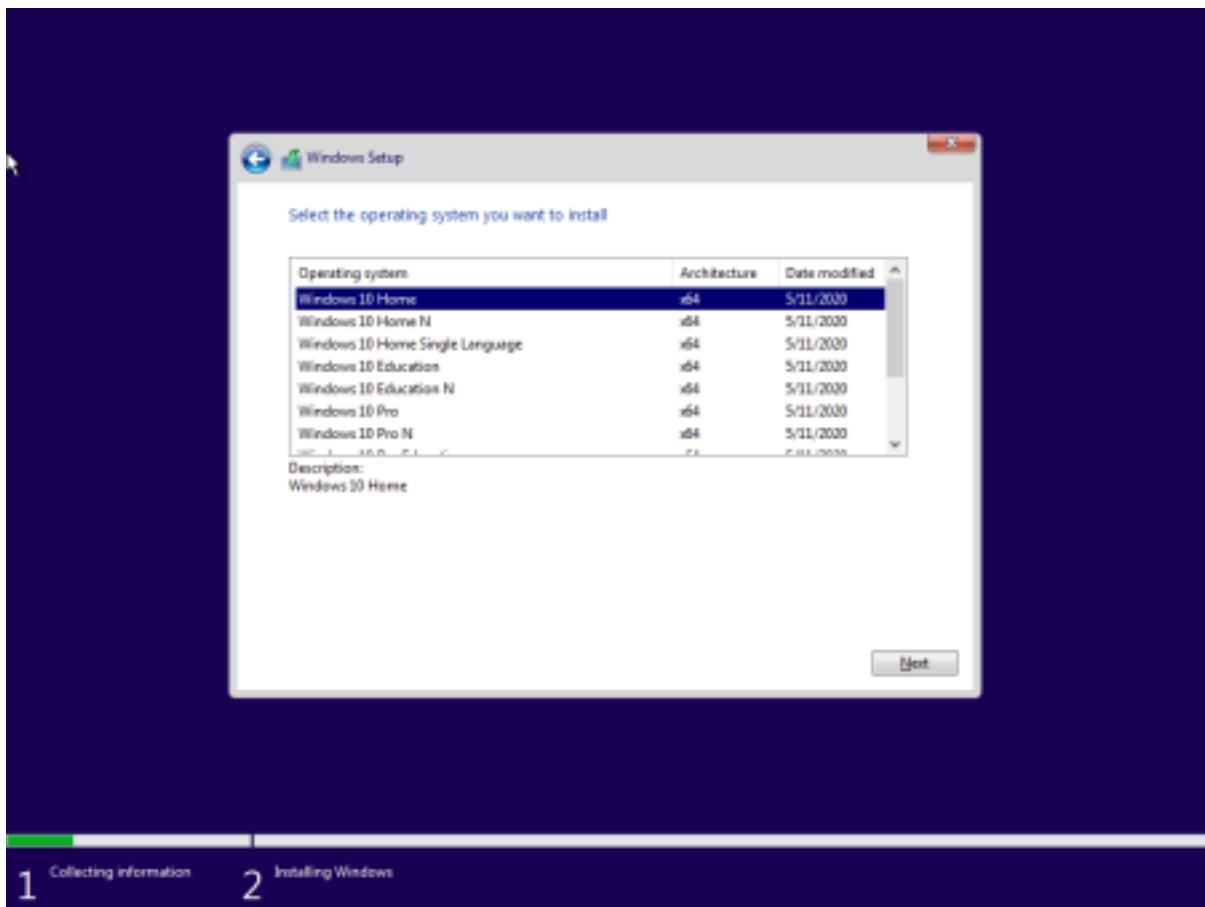
1. Insert the bootable DVD or flash drive into the computer and restart it.
2. Usually the BIOS will automatically boot using the bootable medium, otherwise manually boot it up on the boot menu.
3. Choose the required language and layout settings in the initial installation window and press next and install button.



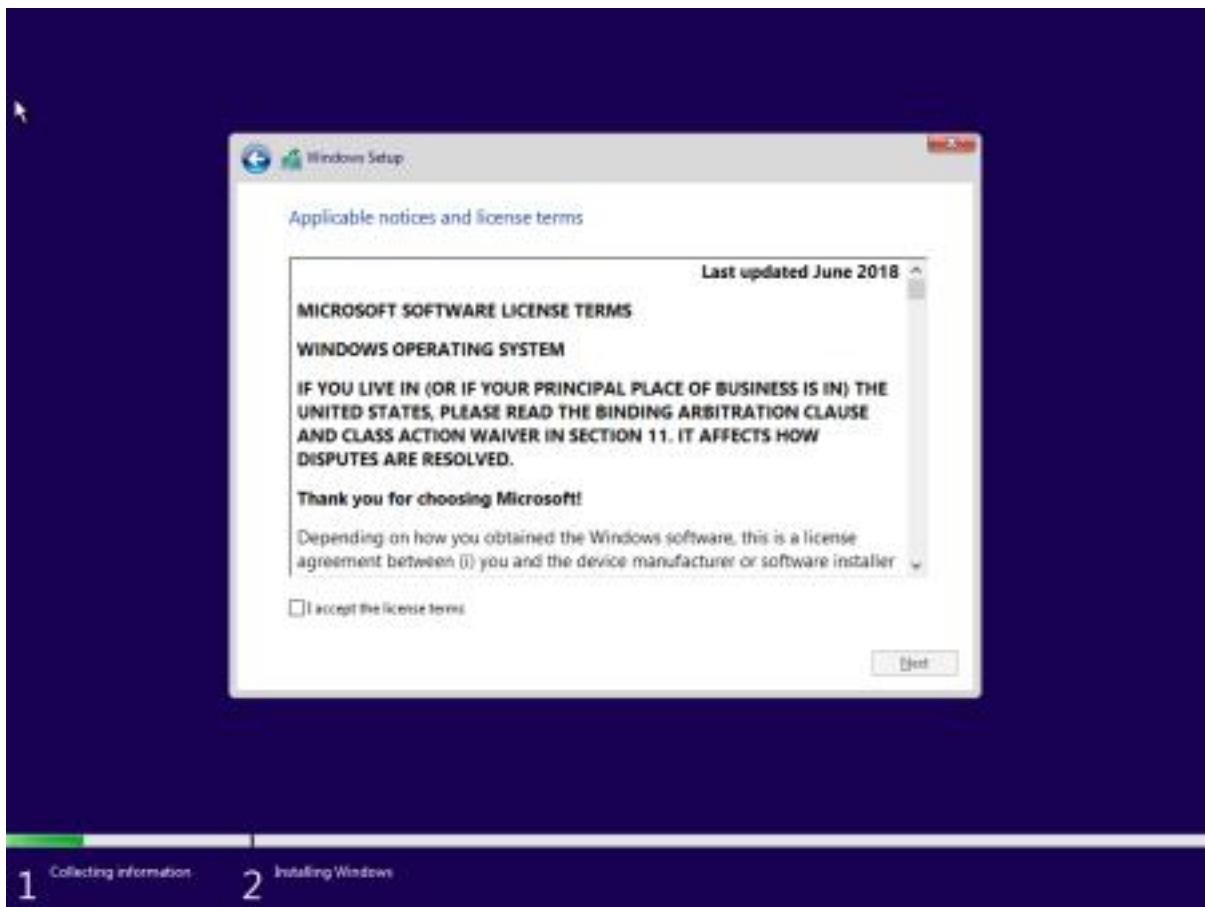
4. Now the installer will prompt you to input a valid License Key. If you do not have one, you can purchase and input it later.



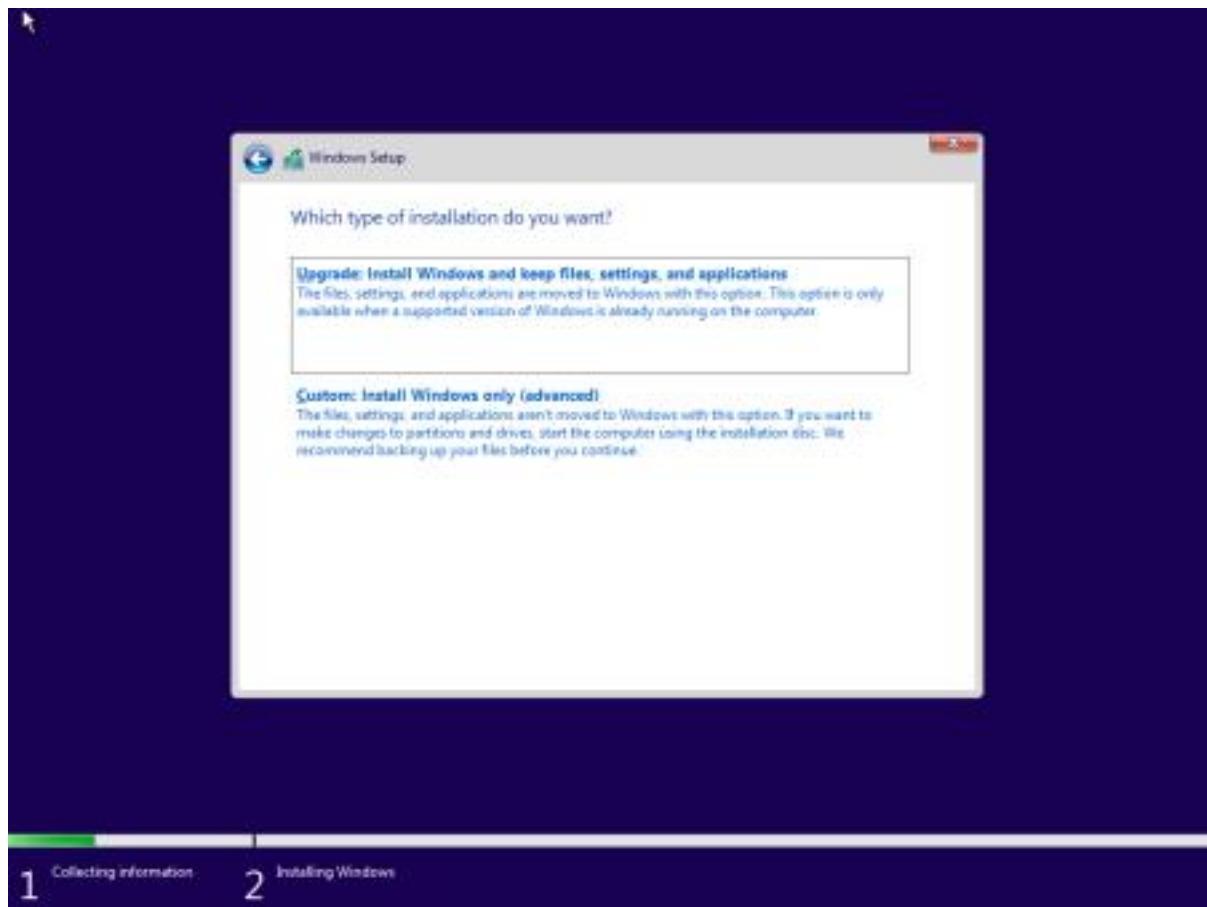
5. Next choose the version of Windows 10 you like to install. Make sure you purchase or have a matching License Key.



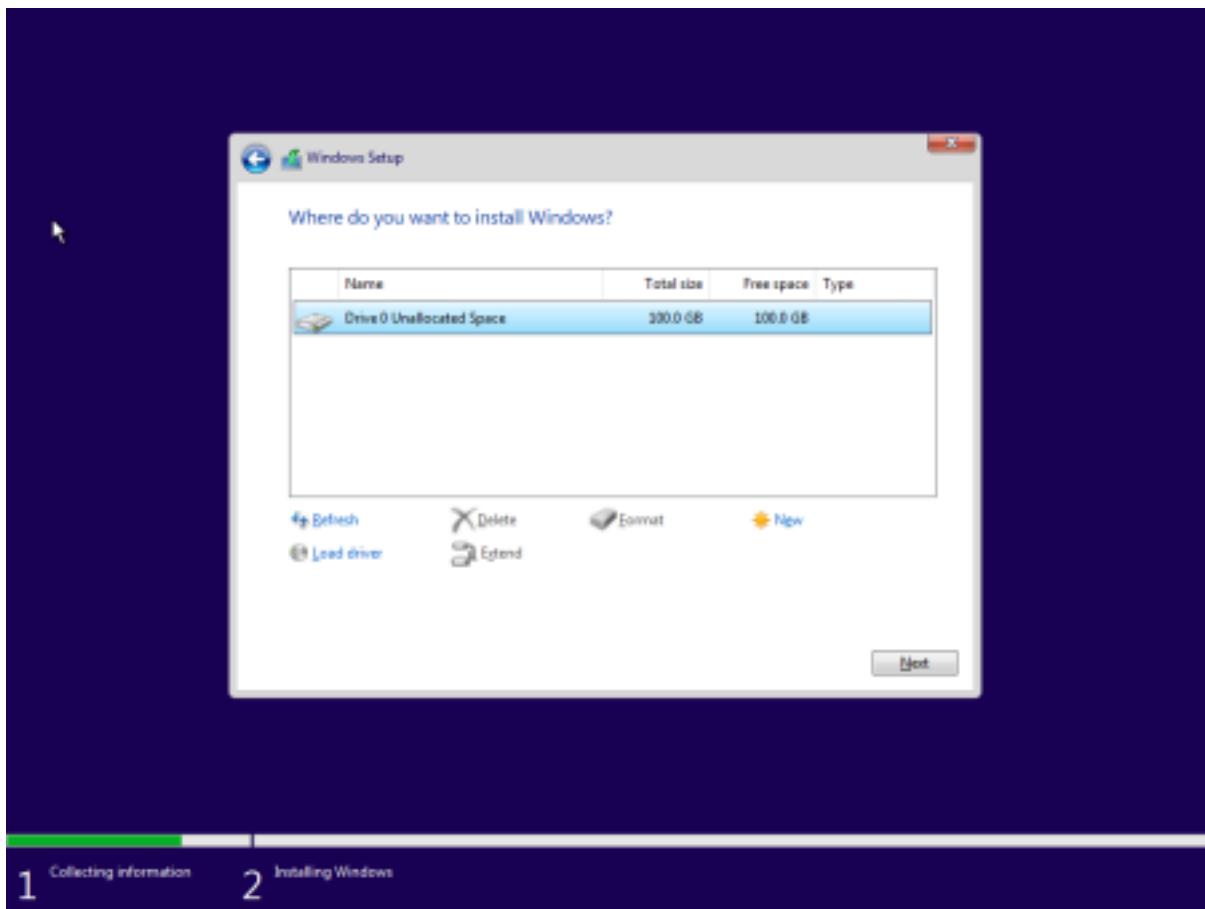
6. Accept the software license terms by ticking the checkbox and click next.



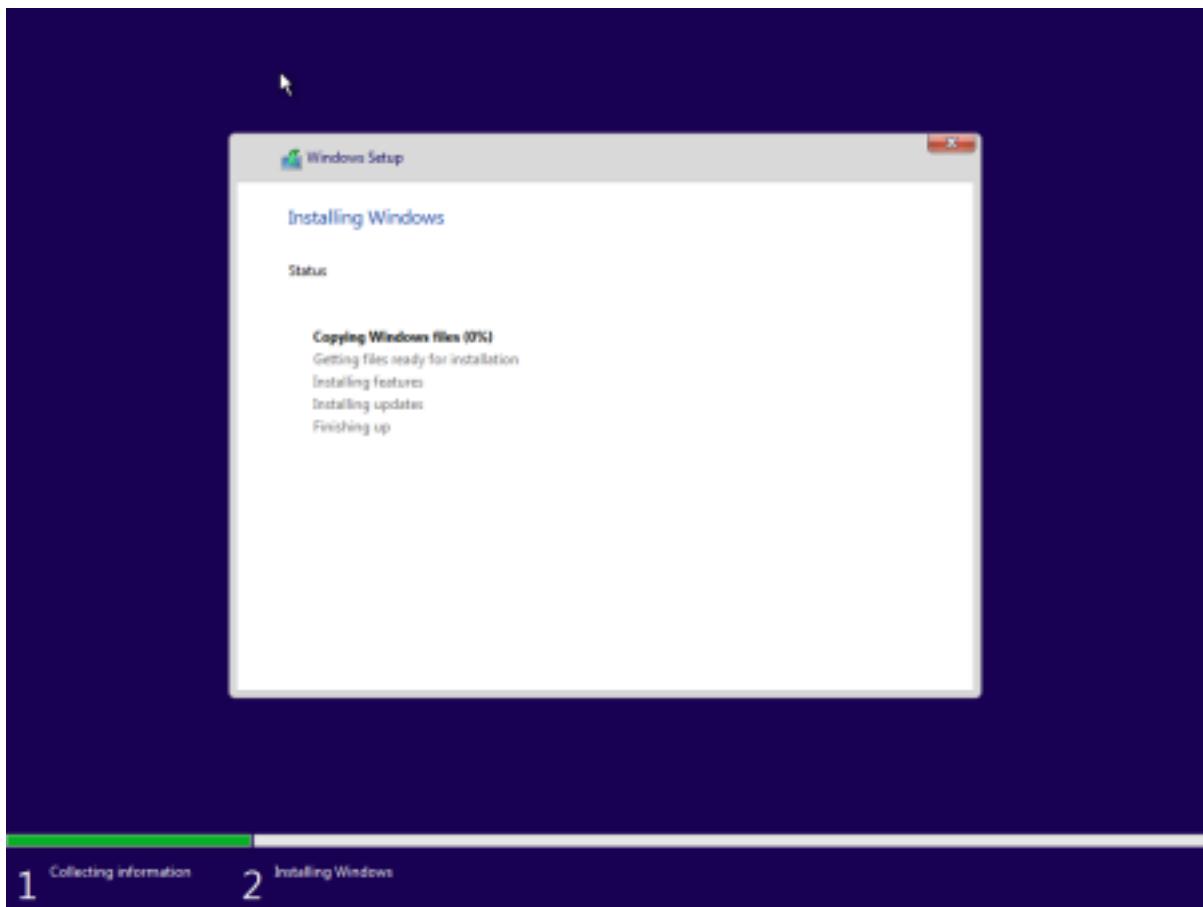
7. If you want to upgrade an existing version of Windows, select upgrade option, else select Custom Install for a fresh installation.



8. Next, inside the disk partition screen, you can create disk partitions and select the drive to install windows which will become the C:\ Drive.



9. On clicking next, the installer starts to install the Windows OS to the Hard Disk, please wait as it will take a few minutes to complete the same.



10. On complete installation, the PC will restart and move to a initial screen to configure your region, keyboard layout, time zone settings. Once they are setup, you will be introduced to the Windows 10 Desktop Screen.

11. Now, make sure you update the windows system immediately download and install updates to install necessary drivers and software patches to the system. In case of some proprietary hardware, the user may have to install the drivers manually.

12. Monitor your system performance on Task manager, if everything is working normally, your installation is complete and your system is ready for use.

## EXPERIMENT-6

### INSTALLATION OF LINUX OVER WINDOWS OPERATING SYSTEM

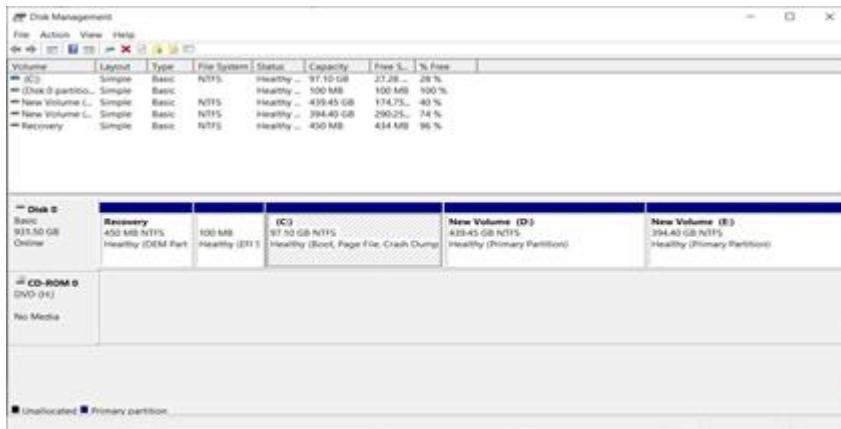
#### **REQUIREMENTS**

1. Pc with at least 2GB RAM
2. 8GB flash drive
3. Rufus
4. Linux ISO file

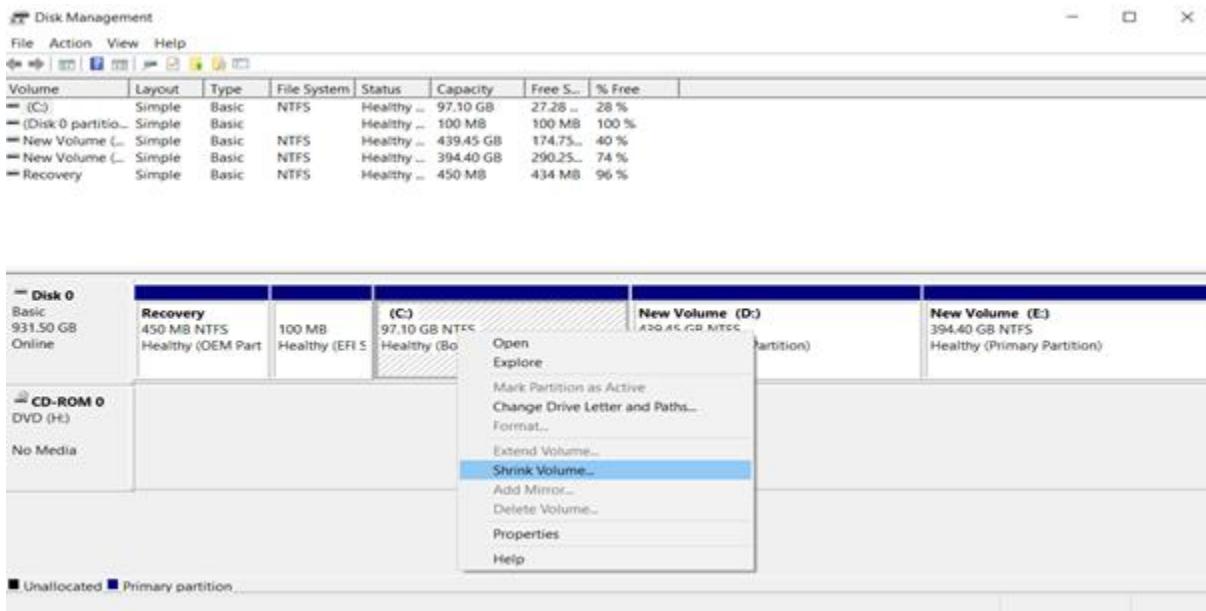
#### **Steps**

##### 1) Partition a Hard Drive in Windows 10

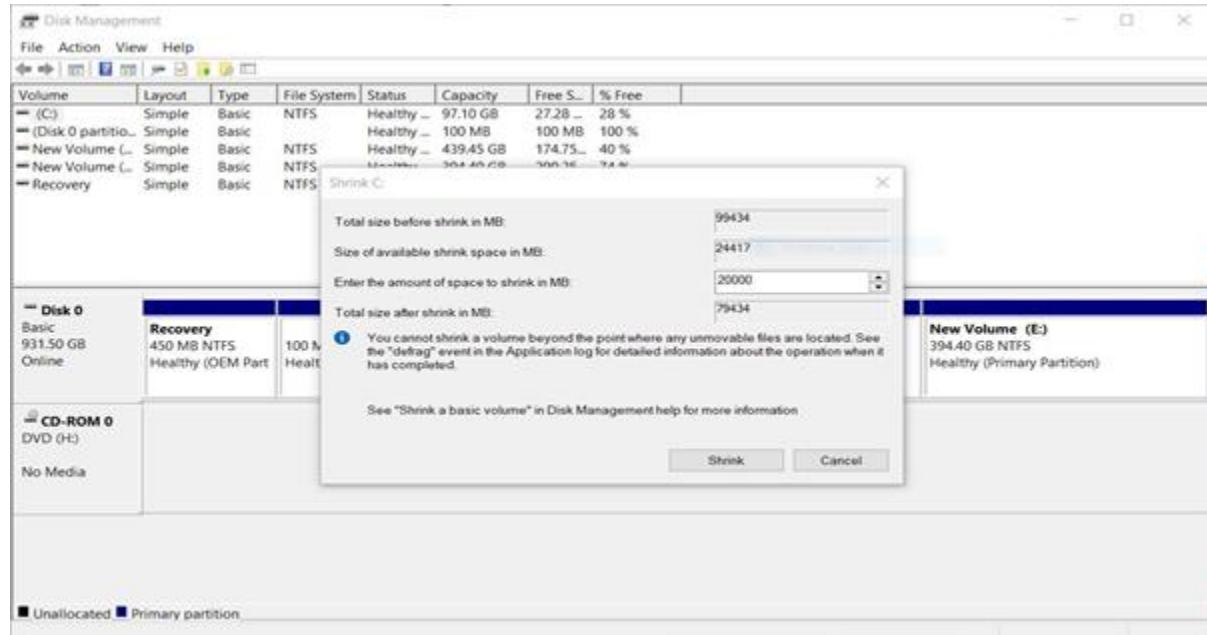
1. Open cmd
2. Type the command “diskmgmt.msc” and hit enter



3. Right-click on your main hard drive and select Shrink Volume.



4. choose how much you want to shrink your drive



## 2) Make a Linux Bootable USB

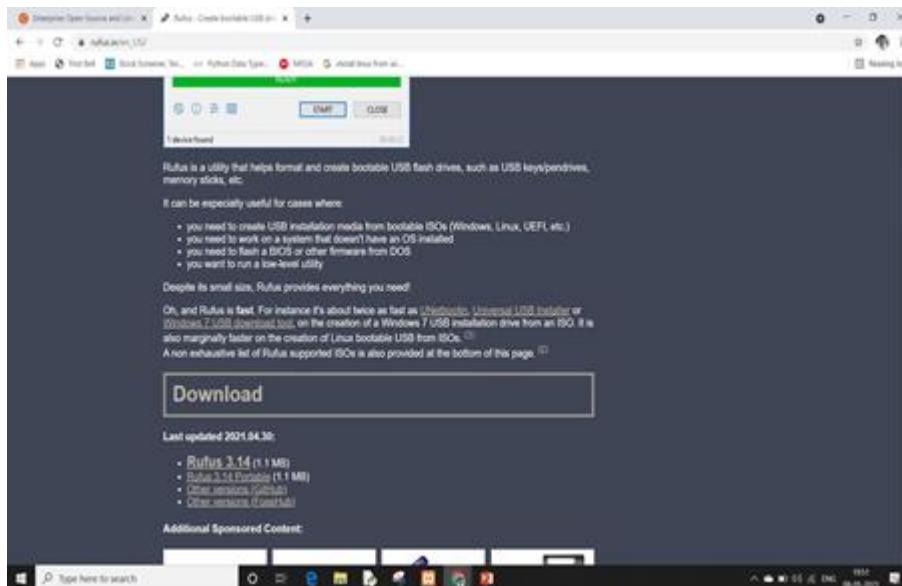
1. Download a Linux distro in ISO format.

<https://ubuntu.com/>

2. Insert the USB drive into your computer.

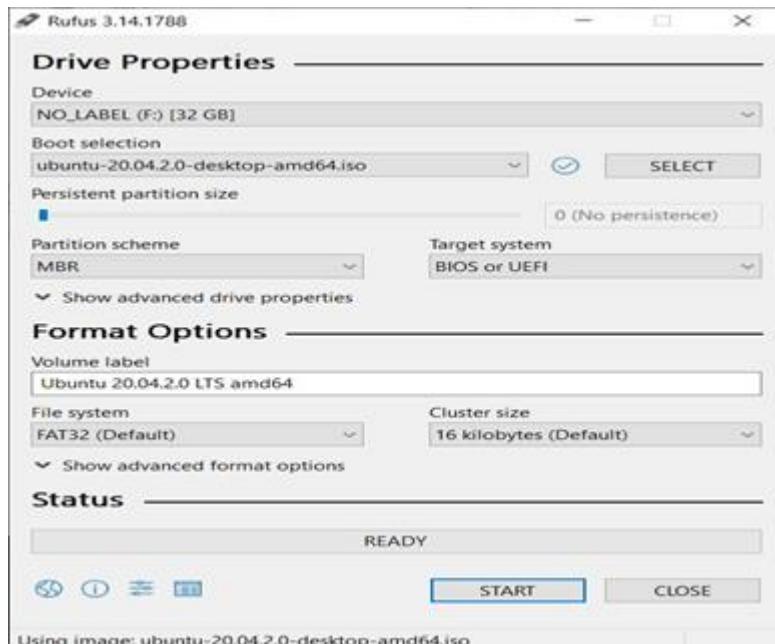
### 3. Download Rufus

[https://rufus.ie/en\\_US/](https://rufus.ie/en_US/)



### 4. Open Rufus and select your USB drive from the Device list

### 5. Under Boot Selection, click the Select button and choose the ISO file you downloaded

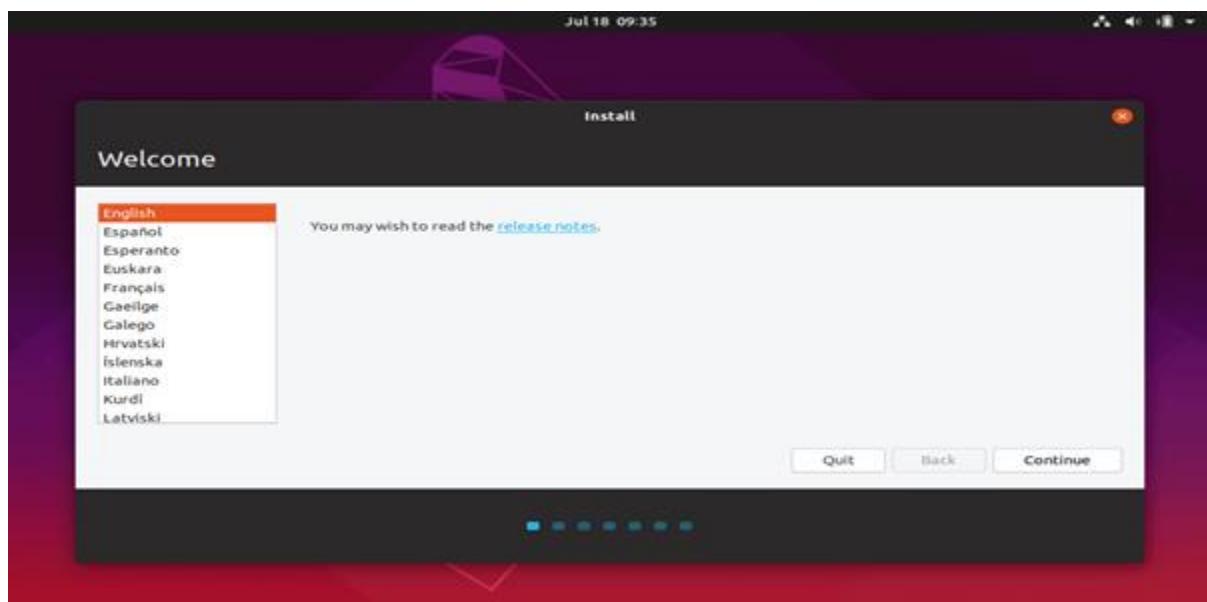


### 3) Install Linux from USB

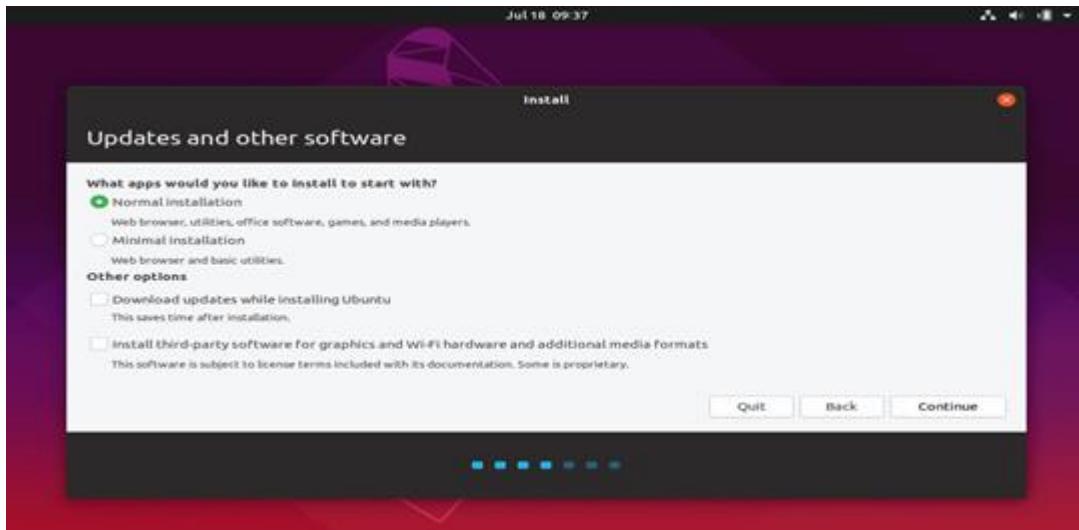
1. Place the USB stick, reboot the machine and instruct the UEFI to boot-up from the USB by pressing a special function key (usually F12, F10 or F2 depending on the vendor specifications).



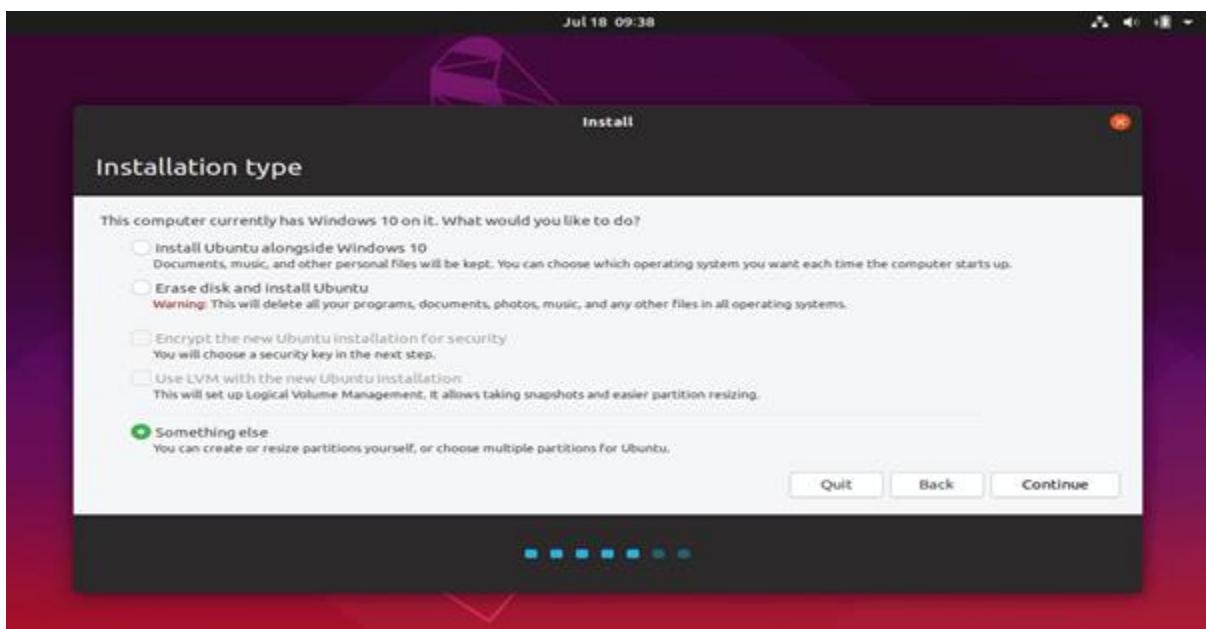
2. Choose the language you wish to perform the installation and click on the Continue



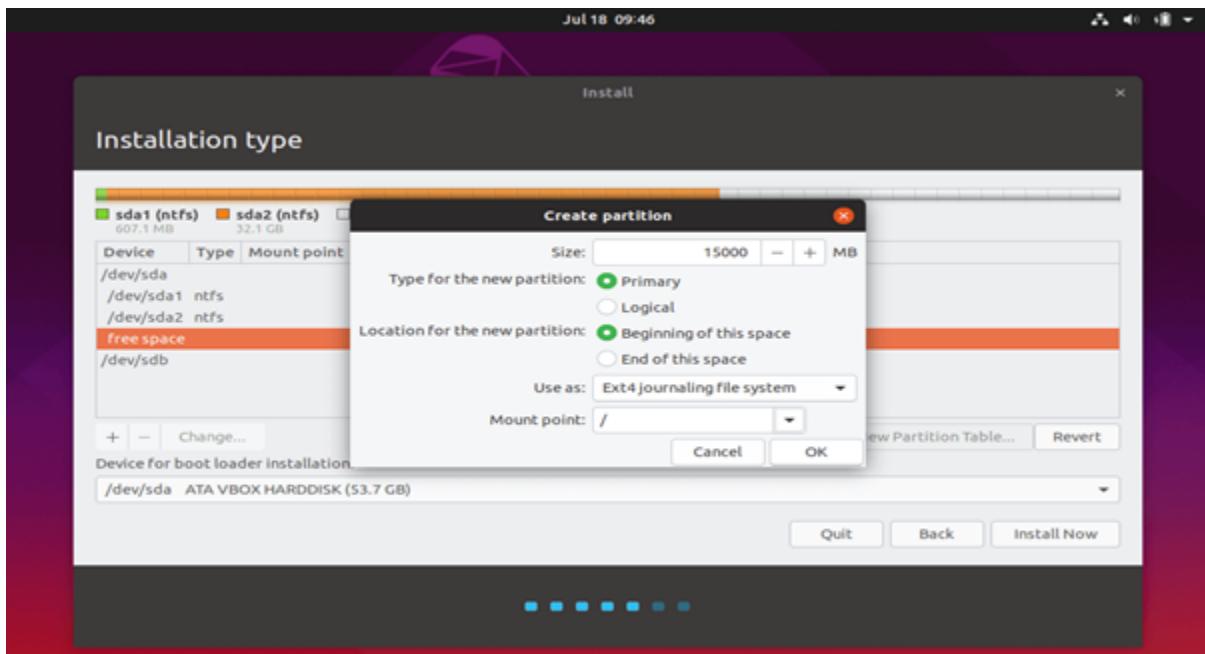
3. choose the first option “Normal Installation” and hit on the Continue button



4. check the Something else option and hit on the Continue button

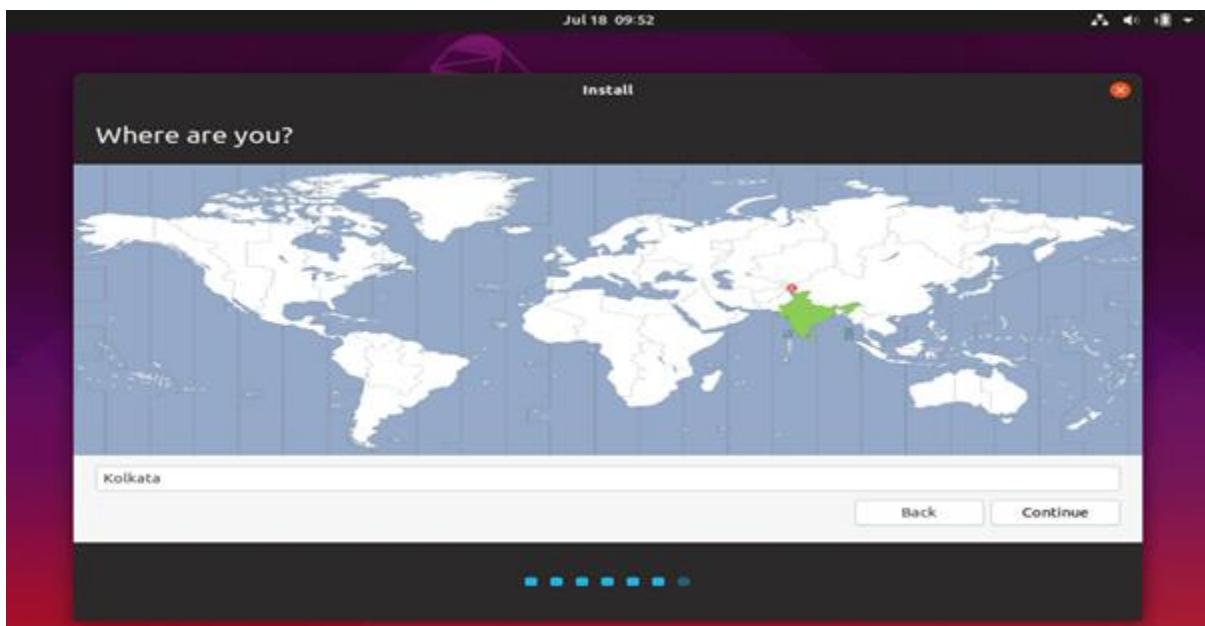


5. we'll create our custom partition layout for Ubuntu



6. hit the Install Now button in order to apply changes to disk and start the installation process.

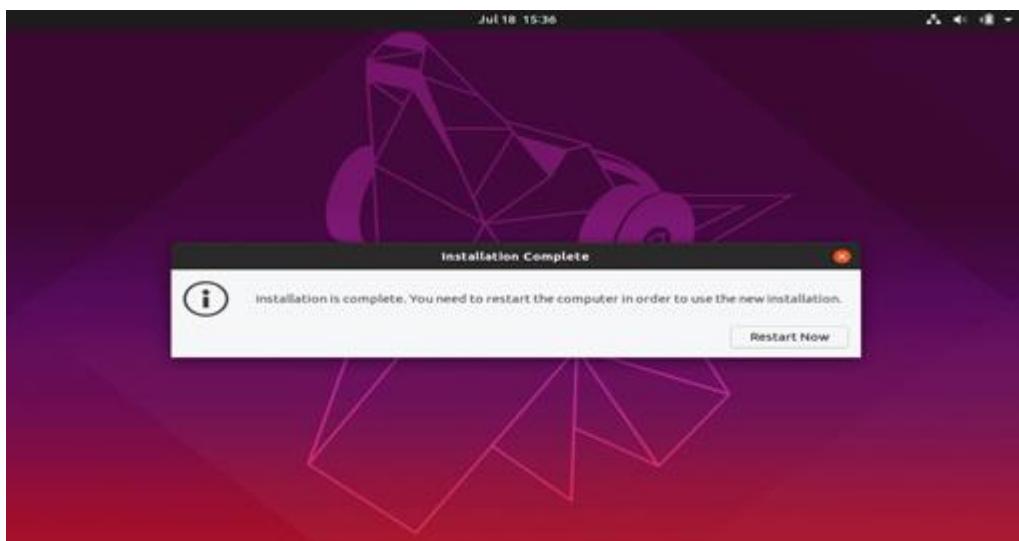
7. set your machine physical location by selecting a city nearby from the map. When done hit Continue to move ahead.



8. Pick up a username and password for your administrative sudo account, enter a descriptive name for your computer and hit Continue to finalize the installation.



9. After the installation process reaches its end hit on the Restart Now button in order to complete the installation.



10. Ubuntu has successfully installed on your system.

## INSTALLATION OF COMMON OPERATING SYSTEMS FOR DESKTOP

**Check the system requirements.** If you've decided that you want to install a new operating system, you'll first need to figure out which one you want to use. Operating systems have varying system requirements, so if you have an older computer, make sure that you can handle a newer operating system.

- Most Windows installations require at least 1 GB of RAM, and at least 15-20 GB of hard disk space. Also, your CPU needs to be powerful enough to run the OS you want to run. Make sure that your computer can accommodate this. If not, you may need to install an older operating system, such as Windows XP.
- Linux operating systems typically don't require as much space and computing power as Windows operating systems. The requirements vary depending on the distribution you choose (Ubuntu, Fedora, Mint, etc.).

**Determine your installation order.** If you are installing a Linux distribution that you want to run alongside Windows, you need to install Windows first and then Linux. This is because Windows has a very strict boot loader that needs to be in place before Linux is installed, otherwise Windows won't load.

**Boot from your installation disc.** Insert the installation disc into your optical drive, and reboot your computer. Normally a computer boots from the hard drive first, so you will need to adjust some settings in your BIOS in order to boot from the disc drive. You can enter the BIOS by hitting the designated Setup key during the boot process. The key will be displayed on the same screen as your manufacturer's logo.

Common Setup keys include F2, F10, F12, and Del/Delete. Once you are in the Setup menu, navigate to the Boot section. Set your DVD/CD drive as the first boot device. If you are installing from a USB drive, make sure that the drive is inserted and then select it as the first boot device. Once you've selected the correct drive, save your changes and exit Setup. Your computer will reboot.

**Wait for the Setup program to load.** No matter which operating system you choose, the setup program will need to copy some files to your computer before it can continue. This can take several minutes, depending on the speed of your computer's hardware. You will most likely need to choose some basic options, such as language and keyboard layout.

**Enter your product key.** If you are installing Windows 8, you will need to enter your product key before you can begin the installation. Older Windows versions will ask for the product key after installation is complete. Linux users will not need a product key unless it is a purchased version such as Red Hat.

**Choose your installation type.** Windows will give you the option of Upgrading or performing a Custom installation. Even if you are upgrading an older version of Windows, it is highly recommended that you choose Custom and start from scratch. This will minimize problems that may arise later from combining old settings and new ones.

If you are installing Linux, you will be given the option to install alongside your existing operating system (Windows), or to erase the disk and install Linux by itself. Choose the option that best meets your needs. If you choose to install alongside Windows, you will be given the option to choose how much hard disk space you want to designate for Linux.

**Format your partitions.** If you are installing Windows, you will need to choose which hard drive partition you want to install it on. Deleting partitions will wipe the data on the partition and return the space to the Unallocated section. Select the unallocated space and create a new partition. If you are installing Linux, the partition needs to be formatted in the Ext4 format.

**Set your Linux options.** Before installation begins, your Linux installer will ask you for your timezone, and you will need to create a username and password. You will use this to log in to your Linux distribution as well as authorize system changes. Windows users will fill out personal information after the installation is complete.

**Wait for the installation to complete.** Depending on the speed of your computer, this can take up to an hour to finish. Most installations are hands-off at this point. Your computer may reboot several times during the installation process.

**Create your Windows login.** Once your Windows installation is complete, you will need to create a username. You can also choose to create a password, though this is not necessary. After creating your login info, you will be asked for your product key.

**Install your drivers and programs.** Once the installation is complete, you will be taken to your new desktop. From here, you can begin installing your programs and make sure that your [drivers are installed and up to date](#). Make sure to install an [antivirus program](#) if you are going to be connecting to the internet.

## **EXPERIMENT-7**

### **HOW TO INSTALL A SERVER SYSTEM**

#### **Phase 1: Collecting Information**

In the first installation phase, the setup program asks for the preliminary information that it needs to begin the installation. A setup wizard prompts you for the following information:

- **Language:** Select your language, time-zone, and keyboard type.
- **Product Key:** Enter the 25-character product key that came with the installation media. If setup says you entered an invalid product key, double-check it carefully. You probably just typed the key incorrectly.
- **Operating System Type:** The setup program lets you select Windows Server 2008 Standard Edition or Core. Choose Standard Edition to install the full server operating system; choose Core if you want to install the new text-only version.
- **License Agreement:** The official license agreement is displayed. You have to agree to its terms in order to proceed.
- **Install Type:** Choose an Upgrade or Clean Install type.
- **Disk Location:** Choose the partition in which you want to install Windows.
- **Upgrade to NTFS:** If you want to upgrade a FAT32 system to NTFS, you'll need to say so now.

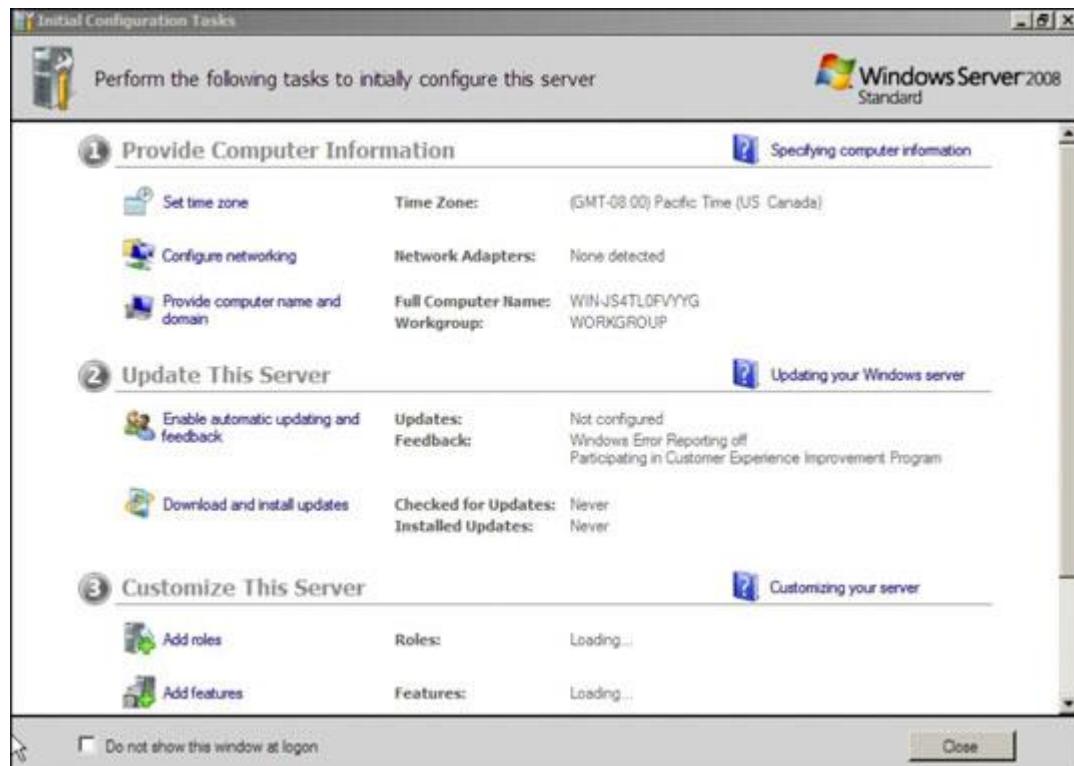
#### **Phase 2: Installing Windows**

In this phase, Windows setup begins the actual process of installing Windows. The following steps are performed in sequence:

1. Copying Files: Compressed versions of the installation files are copied to the server computer.
2. Expanding Files: The compressed installation files are expanded.
3. Installing Features: Windows server features are installed.
4. Installing Updates: The setup program checks Microsoft's website and downloads any critical updates to the operating system.
5. Completing Installation: When the updates are installed, the setup program reboots so it can complete the installation.

## Configuring Your Server

After you've installed Windows Server 2008, the computer automatically reboots, and you're presented with the Initial Configuration Tasks Wizard. This wizard guides you through the most important initial tasks for configuring your new server.



The following list describes the server configuration settings available from this wizard:

- **Set the Administrator Password:** The very first thing you should do after installing Windows is set a secure administrator password.
- **Set the Time Zone:** This is necessary only if the indicated time zone is incorrect.
- **Configure Networking:** The default network settings are usually appropriate, but you can use this option to change the defaults if you wish.
- **Provide Computer Name and Domain:** This option lets you change the server's computer name and join a domain.
- **Enable Automatic Updating:** Use this option if you want to let the server automatically check for operating system updates.
- **Download and Install Updates:** Use this option to check for critical operating system updates.
- **Add Roles:** This option launches the Add Roles Wizard, which lets you configure important roles for your server.
- **Add Features:** This option lets you add more operating system features.

- **Enable Remote Desktop:** Use this option to enable the Remote Desktop feature, which lets you administer this server from another computer.
- **Configure Windows Firewall:** If you want to use the built-in Windows firewall, this option lets you configure it.

## **EXPERIMENT-8**

### **STUDY ON COMMAND LINE TEXT EDITORS**

#### **Text editors**

- ❖ Program used to create and edit plain text files.
- ❖ The main purpose of the text editor is to create a file to be used by another program.
- ❖ For example:
  - HTML for a Web Browser
  - Source code that a compiler can process.

There are two types of text editors:

- Terminal Based editors  
E.g.: Vi, Pico, Nano, Emacs.
- GUI  
E.g.: Gedit, Kwrite

#### **The Emacs Editor**

Emacs is a very popular editor on Unix until recently. Emacs has so many available features like a calculator, calendar, email client. Commands in emacs are either control characters (hold down the <Ctrl> key while typing another character).

Syntax: emacs filename

#### **Pico Editors**

The UNIX Pico editor is a full screen editor which is very easy to use.

Syntax: pico filename

#### **Nano Editors**

The nano is improved open source of pico available for GNU/Linux.

Syntax: nano filename

#### **Vim Editor**

The vi command-line text editor is included with most versions of UNIX and Linux. Vim is an improved version of vi. The vi commands are now linked to the vim commands. There are three modes in vi editor:

1. Command mode: Use key combinations as commands instead of typing text.
2. Insert mode: Typed text is displayed on screen.
3. Extend mode: Used for more advanced commands, such as saving files, exiting vim, or searching

and replacing text.

## Command Mode

- Advanced Navigational Commands

Command	Description
Spacebar	Moves the cursor one space to the right.
Enter	Moves the cursor to the beginning of the next line.
^	Moves the cursor to the beginning of the current line.
\$	Moves the cursor to the end of the current line.
-	Moves the cursor to the beginning of the current line.
+	Moves the cursor to the beginning of the next line (same as the Enter key).

## Insert Mode

- Must be in insert mode to type text
- vim editor displays -- INSERT -- at the lower left to indicate insert mode
- **Switching modes**
  - i ->from command mode to insert mode
  - Esc ->from insert mode to command mode

## Cut, Yank, and Paste

- **Cut text**
  - Store text in a buffer. Cutting text is referred to as deleting in Vim
  - **Paste text**
  - Move from buffer to cursor location
- **Yank (copy) text**
  - Keep in current location and copy to buffer

## Undo Commands

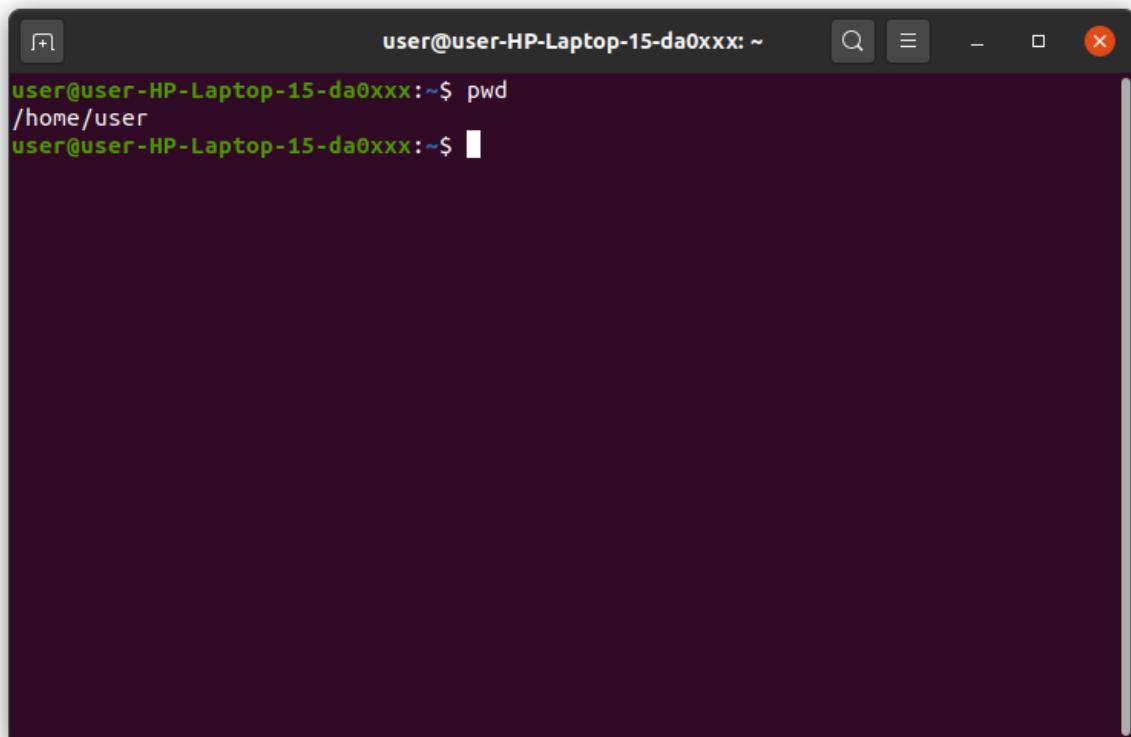
- Undo changes
  - Type u
  - Keep typing u until you have reached your oldest change

## **EXPERIMENT-9**

### **BASIC LINUX COMMANDS**

#### **1. pwd**

Use the `pwd` command to find out the path of the current working directory (folder) you're in. The command will return an absolute (full) path, which is basically a path of all the directories that start with a forward slash (/). An example of an absolute path is `/home/username`.



A screenshot of a terminal window titled "user@user-HP-Laptop-15-da0xxx: ~". The window has a dark background and light-colored text. It displays the command `user@user-HP-Laptop-15-da0xxx:~$ pwd` followed by the output `/home/user`. The terminal window includes standard Linux-style window controls at the top.

#### **2. cd**

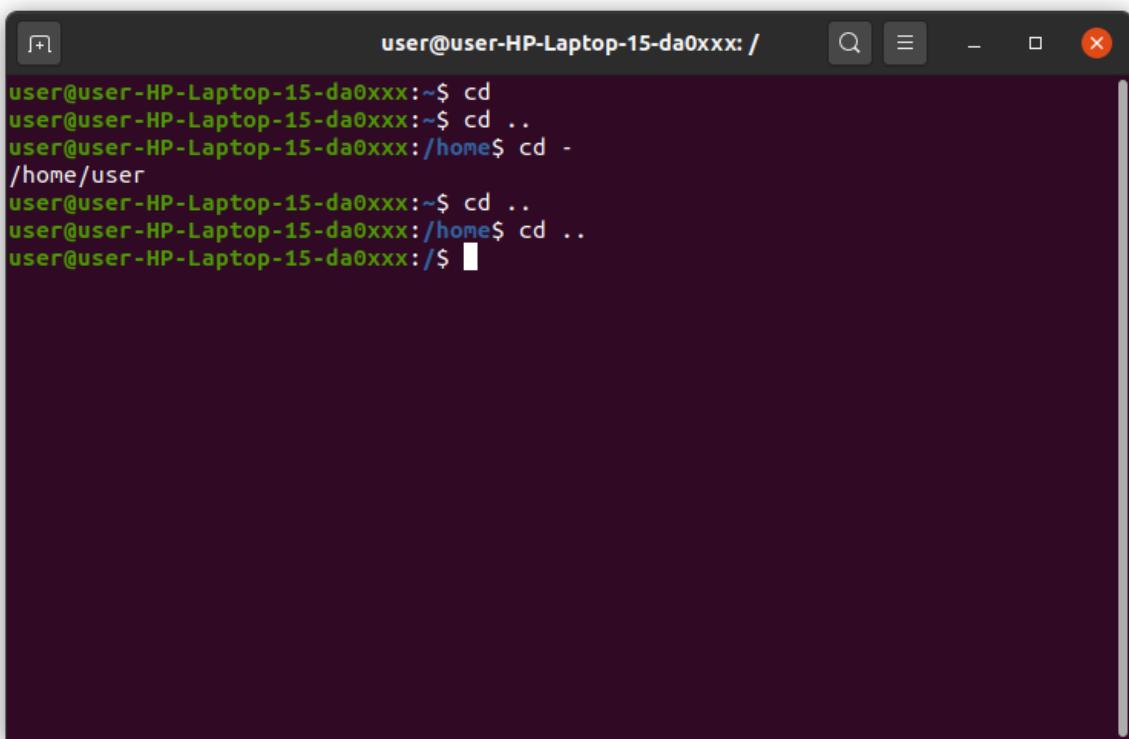
To navigate through the Linux files and directories, use the `cd .` It requires either the full path or the name of the directory, depending on the current working directory that you're in.

Let's say you're in `/home/username/Documents` and you want to go to `Photos`, a subdirectory of `Documents`. To do so, simply type the following command: `cd Photos`. Another scenario is if you want to switch to a completely new directory, for example, `/home/username/Movies`. In this case, you have to type `cd` followed by the directory's absolute path: `cd /home/username/Movies`.

There are some shortcuts to help you navigate quickly:

- cd .. (with two dots) to move one directory up
- cd to go straight to the home folder
- cd- (with a hyphen) to move to your previous directory

On a side note, Linux's shell is case sensitive. So, you have to type the name's directory exactly as it is.



The screenshot shows a terminal window with a dark background and light-colored text. The title bar reads "user@user-HP-Laptop-15-da0xxx: /". The terminal displays the following command history:

```
user@user-HP-Laptop-15-da0xxx:~$ cd
user@user-HP-Laptop-15-da0xxx:~$ cd ..
user@user-HP-Laptop-15-da0xxx:/home$ cd -
/home/user
user@user-HP-Laptop-15-da0xxx:~$ cd ..
user@user-HP-Laptop-15-da0xxx:/home$ cd ..
user@user-HP-Laptop-15-da0xxx:/$ █
```

### 3. ls

The ls command is used to view the contents of a directory. By default, this command will display the contents of your current working directory.

If you want to see the content of other directories, type ls and then the directory's path. For example, enter ls /home/username/Documents to view the content of Documents.

There are variations you can use with the ls command:

- ls -R will list all the files in the sub-directories as well
- ls -a will show the hidden files

- ls -al will list the files and directories with detailed information like the permissions, size, owner, etc.
- ls -t lists files sorted in the order of “last modified”
- -r option will reverse the natural sorting order. Usually used in combination with other switches such as ls -tr. This will reverse the time-wise listing.

```
user@user-HP-Laptop-15-da0xxx:~$ ls
bitstring.c  Desktop      GIT          Pictures    Templates
cn           Documents   Music        prims.c    Videos
cpgms        Downloads  '#newfile.txt#'  Public
```

```
user@user-HP-Laptop-15-da0xxx:~$ ls -al
total 160
drwxr-xr-x 20 user user 4096 Jun 11 02:08 .
drwxr-xr-x  3 root root 4096 Dec  1 2020 ..
-rw-r--r--  1 user user 10016 Jun 11 02:20 .bash_history
-rw-r--r--  1 user user  220 Dec  1 2020 .bash_logout
-rw-r--r--  1 user user 3771 Dec  1 2020 .bashrc
-rw-rw-r--  1 user user  458 Jan  3 00:59 bitstring.c
drwxr-xr-x 16 user user 4096 Jun 10 03:39 .cache
drwxrwxr-x  2 user user 4096 Jun 11 02:18 cn
drwx----- 16 user user 4096 Jun 10 03:39 .config
drwxrwxr-x  5 user user 4096 Mar 21 14:32 cpgms
drwxr-xr-x  2 user user 4096 Dec  2 2020 Desktop
drwxr-xr-x  2 user user 4096 Dec  2 2020 Documents
drwxr-xr-x  2 user user 4096 Jun 11 17:27 Downloads
drwx----- 3 user user 4096 Jun 10 03:47 .emacs.d
drwxrwxr-x  3 user user 4096 Dec 28 01:32 GIT
-rw-rw-r--  1 user user   58 Dec 25 21:00 .gitconfig
drwx----- 3 user user 4096 Jun 11 02:19 .gnupg
drwxr-xr-x  3 user user 4096 Dec  2 2020 .local
drwx----- 5 user user 4096 Dec  5 2020 .mozilla
drwxr-xr-x  2 user user 4096 Dec  2 2020 Music
-rw-rw-r--  1 user user  29 Jun 10 03:58 '#newfile.txt#'
drwxr-xr-x  4 user user 4096 Jun 11 17:28 Pictures
-rw-rw-r--  1 user user 1625 Mar 21 14:18 prims.c
-rw-r--r--  1 user user  807 Dec  1 2020 .profile
drwxr-xr-x  2 user user 4096 Dec  2 2020 Public
drwxr-xr-x  2 user user 4096 Dec 26 21:35 .ssh
-rw-r--r--  1 user user     0 Dec  2 2020 .sudo_as_admin_successful
-rw-----  1 user user 12288 Jun 11 02:08 .swp
-rw-----  1 user user 12288 Jun 10 04:34 .swo
-rw-----  1 user user 12288 Jun 10 04:13 .swp
drwxr-xr-x  2 user user 4096 Dec  2 2020 Templates
drwxr-xr-x  2 user user 4096 Dec  2 2020 Videos
-rw-----  1 user user  782 Jun 11 02:08 .viminfo
user@user-HP-Laptop-15-da0xxx:~$
```

```

user@user-HP-Laptop-15-da0xxx:~$ ls -R
.:
bitstring.c  Desktop   GIT          Pictures  Templates
cn           Documents  Music        prims.c   Videos
cpgms        Downloads '#newfile.txt#' Public

./cn:
a.1.png     'd$.2.png' dw.1.png      I.2.png   l.1.png   0.3.png
A.1.png     'd^.1.png' dw.2.png      I.3.png   l.2.png   x.1.png
a.2.png     D.1.png    emacs.png     i.png    o.1.png   X.1.png
A.2.png     'd^.2.png' 'emacs version.png' j.1.png   0.1.png   X.2.png
a.3.png     D.2.png    h.1.png      j.2.png   o.2.png   X.2.Z.png
A.3.png     dd.1.png   h.2.png      k.1.png   o.2.png
'd$.1.png'  dd.2.png   I.1.png      k.2.png   o.3.png

./cpgms:
a.c         dfs.c       kruskal.c    singldeletion.out
a.out       dfs.out     kruskal.out  singlell.c
bbits.c     dijkstra.c linkedlists  sing.out
bbits.out   disjoint.c llfinal.c    sma.c
bfs.c       disjoint.out llfinal.out  smallarray.c
bfs.out     display.c  mergesort.c  smallarray.out
binomial.c  display.out mergesort.out smallesst.c
binomial.out doublyll.c  mul.c      smallesst.out
bits.c      doublyll.out mul.out    sma.out
bitsfinal.c ds.c       oebrig.c    ssh.....
bitsfinal.out ds.out    oebrig.out  stackarray.c
bits.out    duply.c    prims.c    stackarray.out
Bitstring   duply.out   prims.out   stackll.c
bitstring.c evenodsum.c queuearray.c stackll.out
bitstringorg.c evenodsum.out queuearray.out stack.out
bitstringorg.out evenodsum.c queueell.c sum.c
bitstring.out evenodsum.out queueell.out sum.out
bridge course' great.c   rbtree.c    topological.c
bst.c       great.out  rbtree.out   tree.c
bst.out     infront.c  rectcircle.c ttt.c
btree.c     infront.out rectcircle.out reversearr.c
btree.out   infront.c  reversearr.c 'Untitled Document 1.c'
circular.c  infront.out reversearr.out

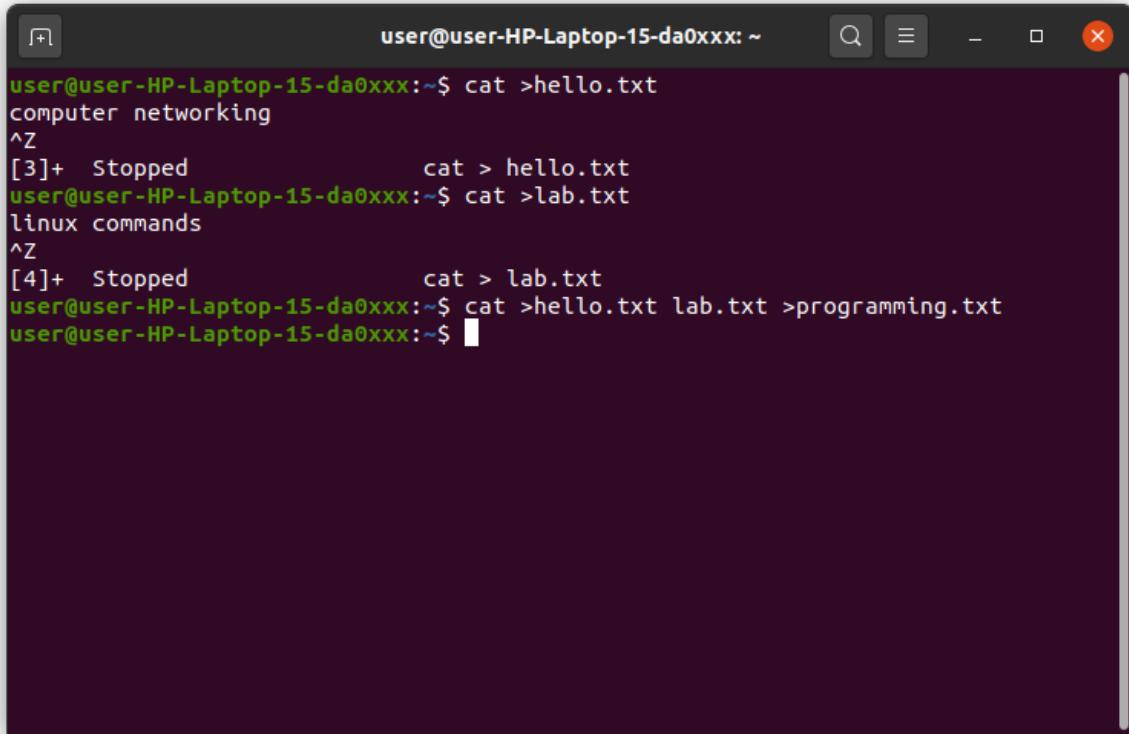
```

#### 4. cat

cat (short for concatenate) is one of the most frequently used commands in Linux. It is used to list the contents of a file on the standard output stdout . To run this command, type cat followed by the file's name and its extension. For instance: cat file.txt.

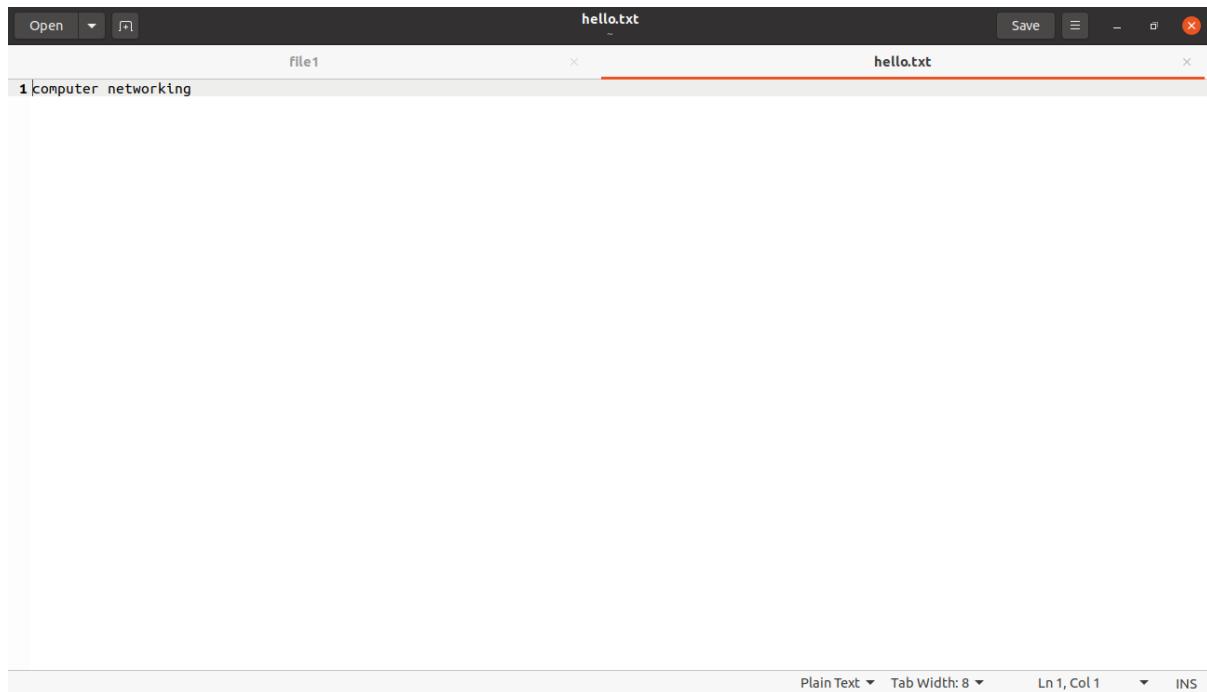
Here are other ways to use the cat command:

- cat > filename creates a new file
- cat filename1 filename2>filename3 joins two files (1 and 2) and stores the output of them in a new file (3)
- to convert a file to upper or lower case use, cat filename | tr a-z A-Z >output.txt



A screenshot of a terminal window titled "user@user-HP-Laptop-15-da0xxx: ~". The window shows several command-line operations:

```
user@user-HP-Laptop-15-da0xxx:~$ cat >hello.txt
computer networking
^Z
[3]+  Stopped                  cat > hello.txt
user@user-HP-Laptop-15-da0xxx:~$ cat >lab.txt
linux commands
^Z
[4]+  Stopped                  cat > lab.txt
user@user-HP-Laptop-15-da0xxx:~$ cat >hello.txt lab.txt >programming.txt
user@user-HP-Laptop-15-da0xxx:~$ █
```



The screenshot shows a terminal window with two tabs open. The top tab is titled "lab.txt" and contains the text "1 computer networking" and "2 linux commands". The bottom tab is titled "programming.txt" and also contains the same text. Both tabs have a "Plain Text" dropdown menu, a "Save" button, and other standard file operations buttons.

```
1 computer networking
2 linux commands
```

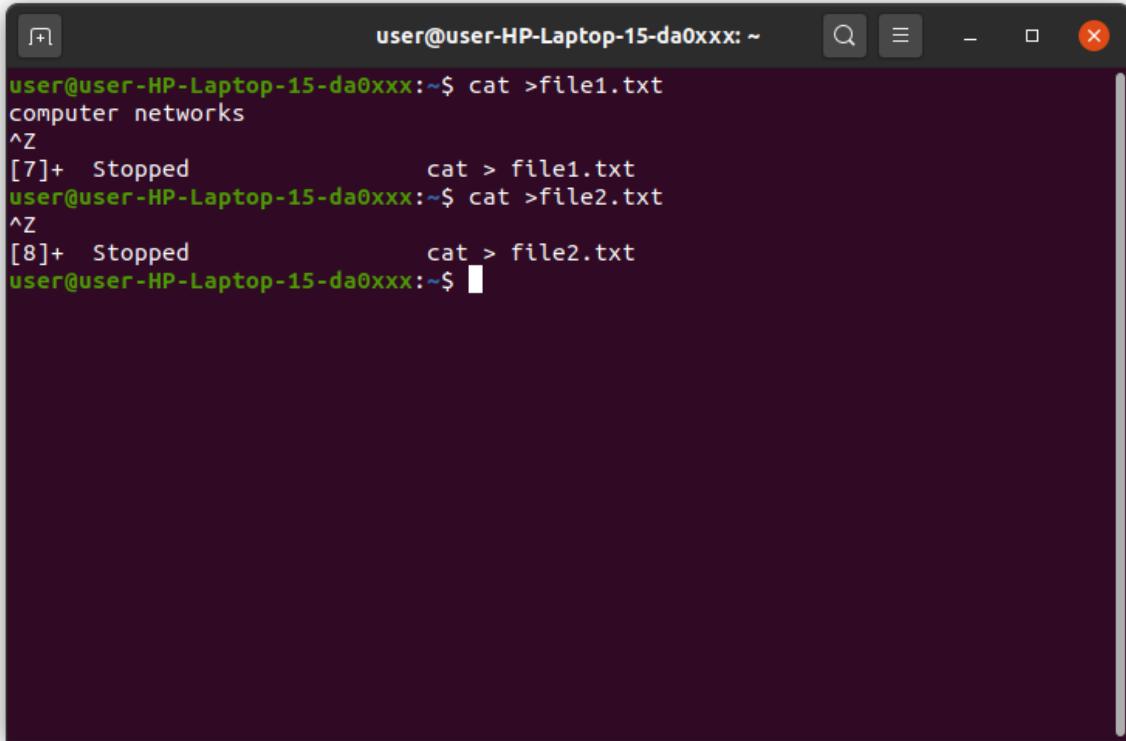
```
1 computer networking
2 linux commands
```

## 5. cp

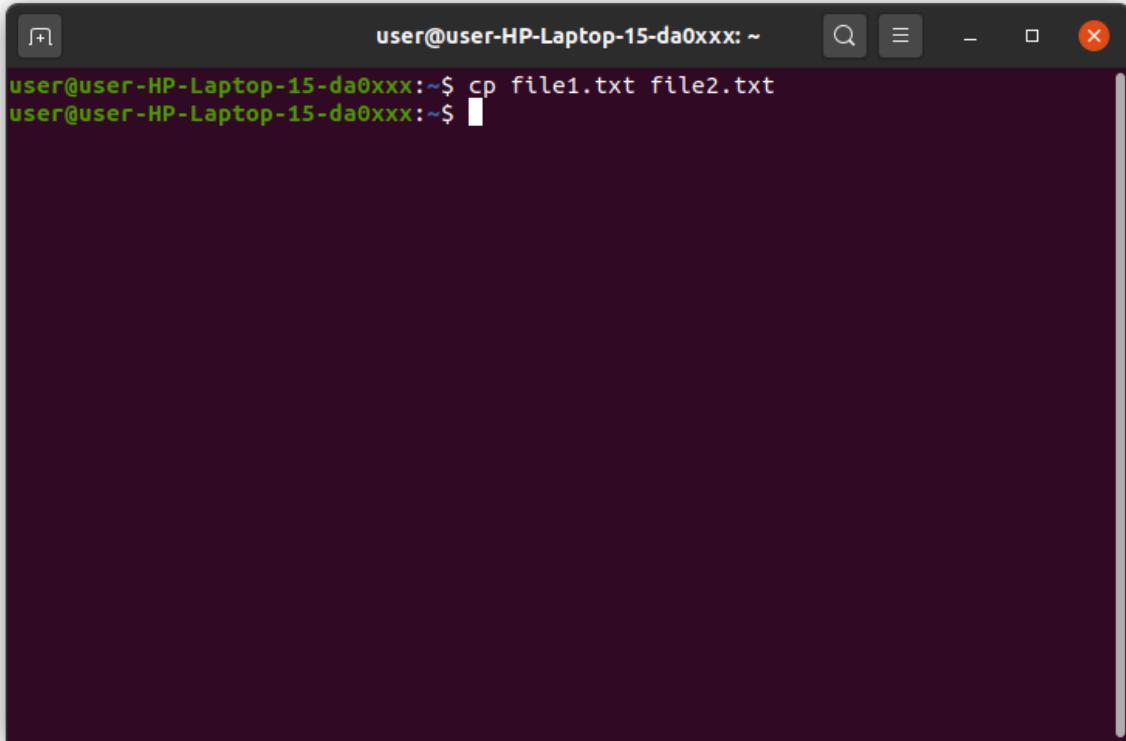
Use the cp command to copy files from the current directory to a different directory. For instance, the command `cp scenery.jpg /home/username/Pictures` would create a copy of `scenery.jpg` (from your current directory) into the `Pictures` directory.

- `cp -i` will ask for user's consent in case of a potential file overwrite.
- `cp -p` will preserve source files' mode, ownership and timestamp.
- `cp -r` will copy directories recursively.

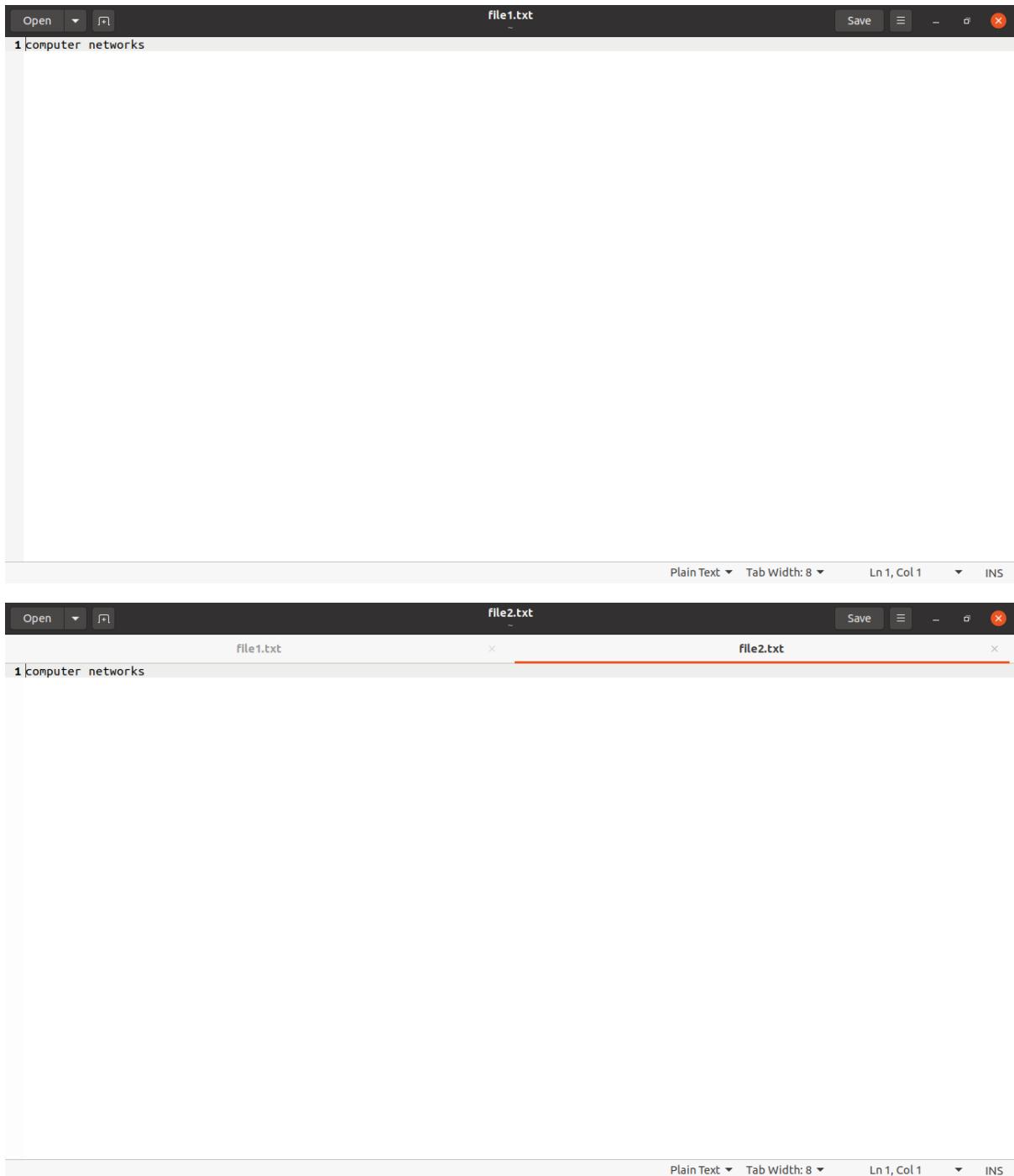
- cp -u copies files only if the destination file is not existing or the source file is newer than the destination file.



```
user@user-HP-Laptop-15-da0xxx:~$ cat >file1.txt
computer networks
^Z
[7]+  Stopped                  cat > file1.txt
user@user-HP-Laptop-15-da0xxx:~$ cat >file2.txt
^Z
[8]+  Stopped                  cat > file2.txt
user@user-HP-Laptop-15-da0xxx:~$
```



```
user@user-HP-Laptop-15-da0xxx:~$ cp file1.txt file2.txt
user@user-HP-Laptop-15-da0xxx:~$
```

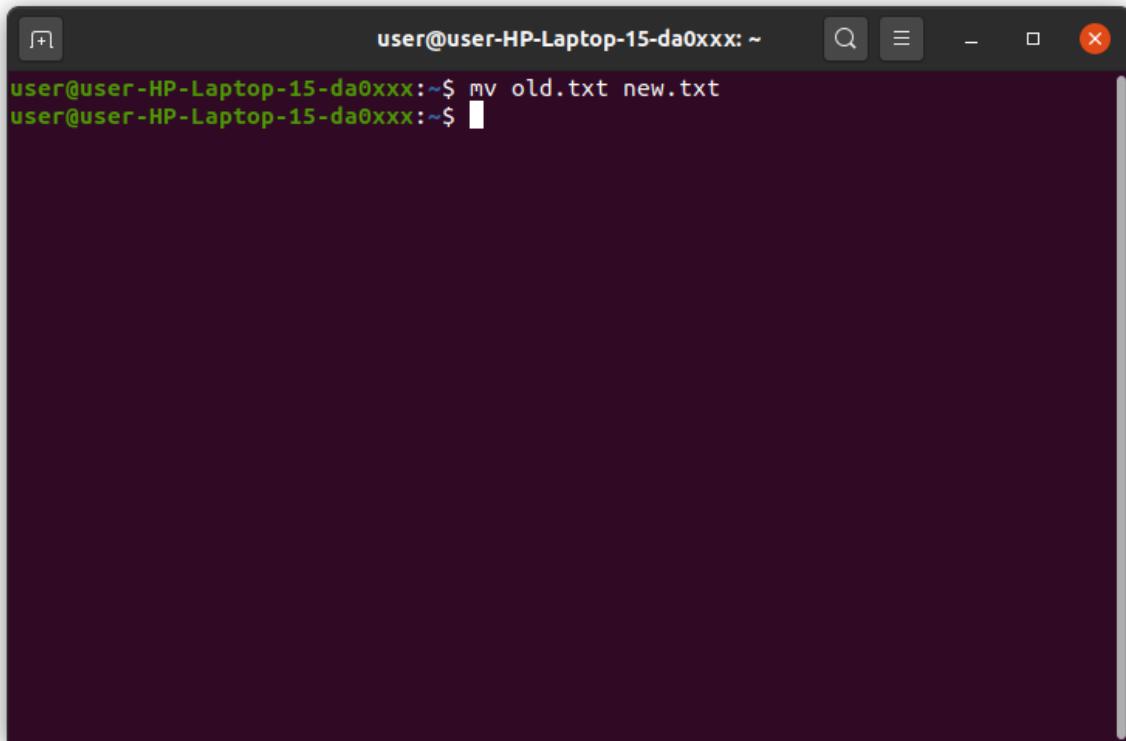


## 6. mv

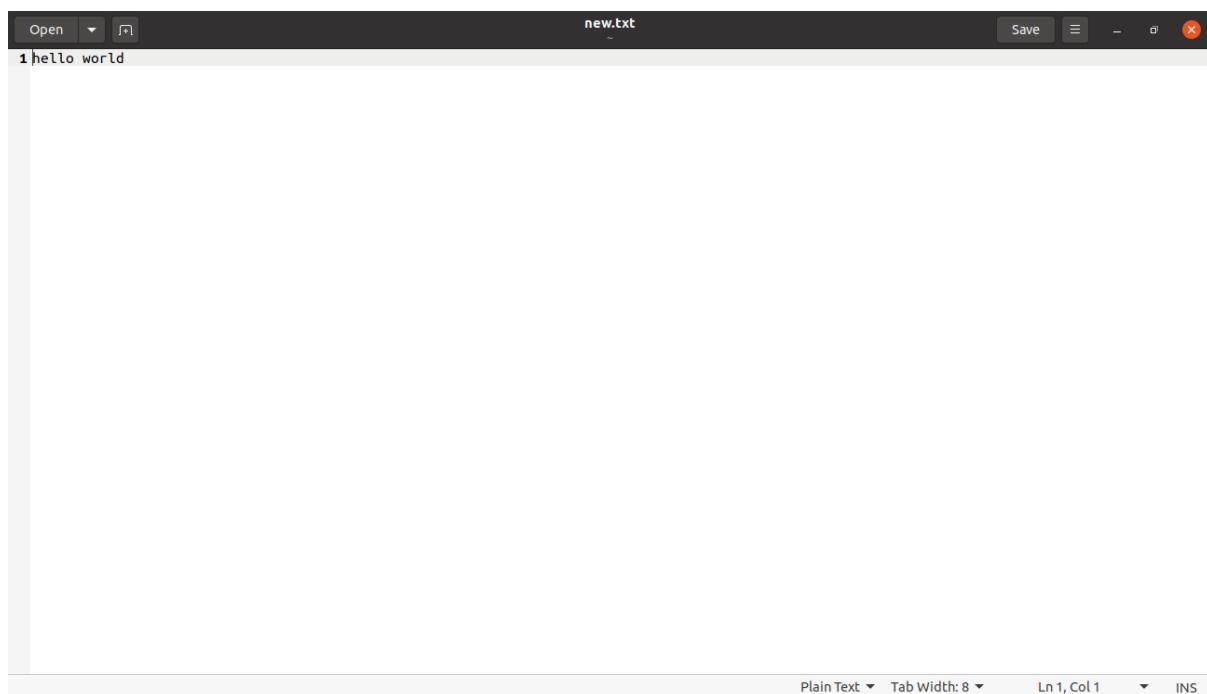
The primary use of the mv command is to move files, although it can also be used to rename files.

The arguments in mv are similar to the cp command. You need to type mv, the file's name, and the destination's directory. For example: mv file.txt /home/username/Documents.

To rename files, the Linux is mv oldname.ext newname.ext



A screenshot of a terminal window titled "user@user-HP-Laptop-15-da0xxx: ~". The window shows the command "mv old.txt new.txt" being typed at the prompt. The terminal has a dark background and standard light-colored text.



A screenshot of a text editor window titled "new.txt". The file contains the text "1 hello world". The editor has a dark interface with light-colored text. The status bar at the bottom shows "Plain Text" and "Tab Width: 8".

## 7. mkdir

Use mkdir command to make a new directory — if you type mkdir

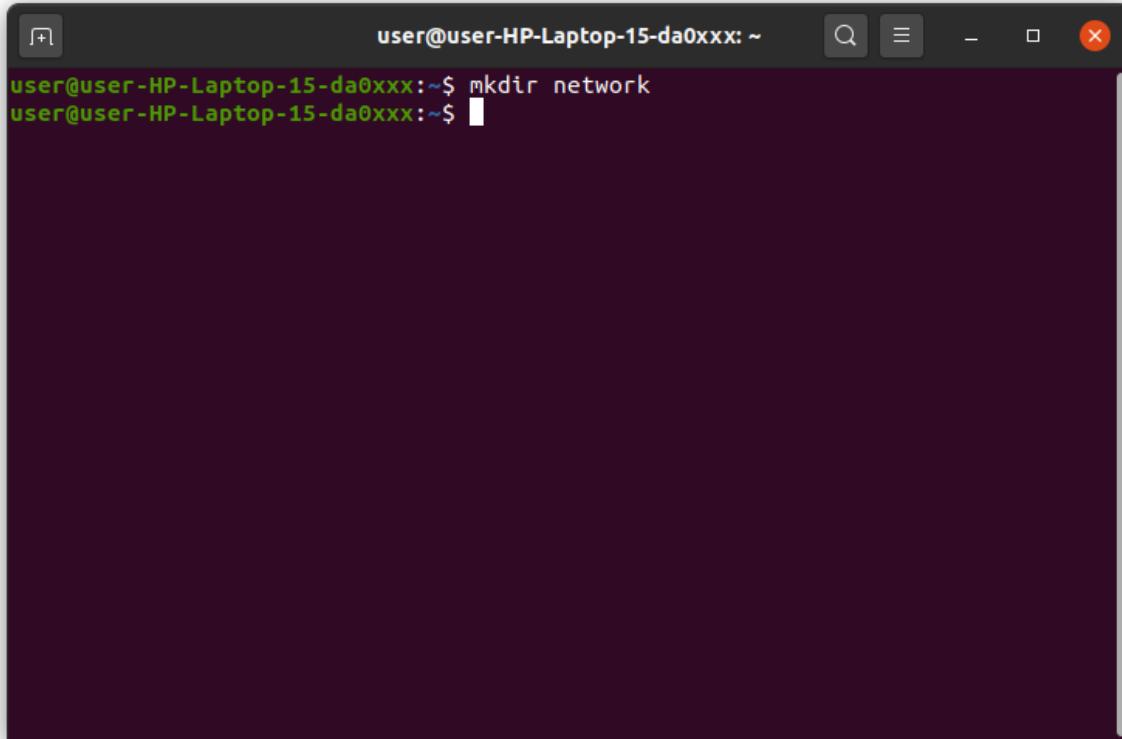
Music it will create a directory called Music.

There are extra mkdir commands as well:

- To generate a new directory inside another directory, use this Linux basic command

```
mkdir Music/Newfile
```

- use the p (parents) option to create a directory in between two existing directories. For example, mkdir -p Music/2020/Newfile will create the new “2020” file.

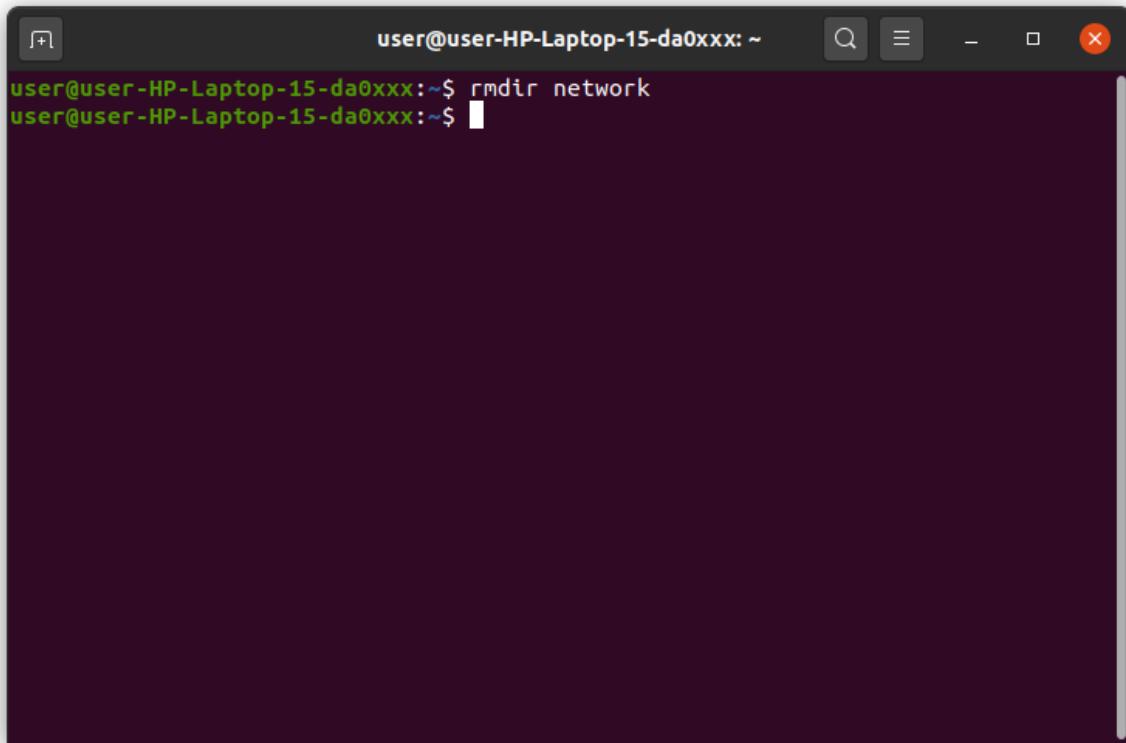


```
user@user-HP-Laptop-15-da0xxx:~$ mkdir network
user@user-HP-Laptop-15-da0xxx:~$
```

Name	Size	Location
network	0 items	

## 8. rmdir

If you need to delete a directory, use the rmdir command. However, rmdir only allows you to delete empty directories.

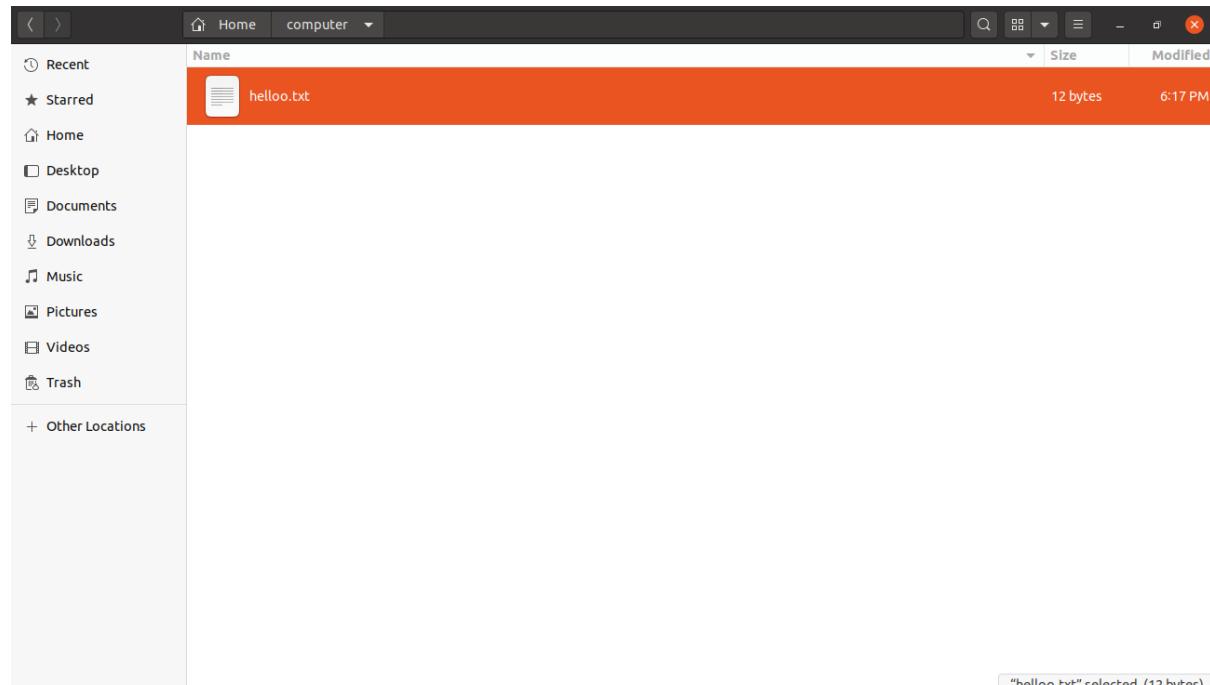


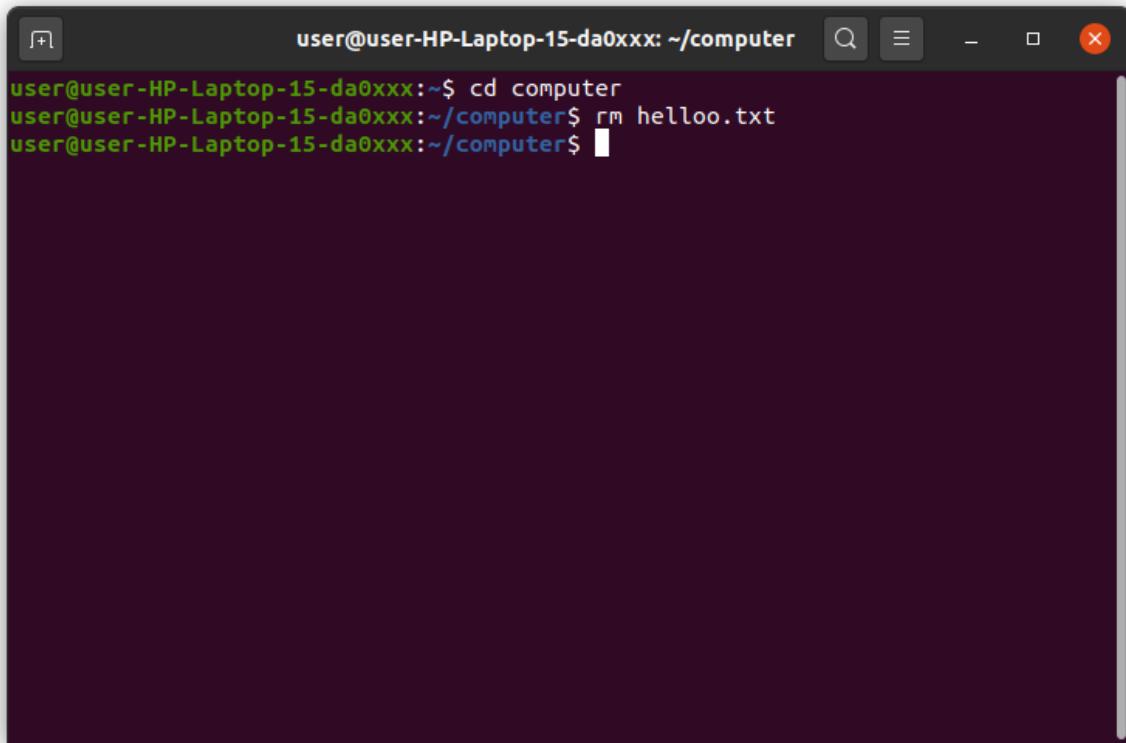
```
user@user-HP-Laptop-15-da0xxx:~$ rmdir network
user@user-HP-Laptop-15-da0xxx:~$
```

## 9. rm

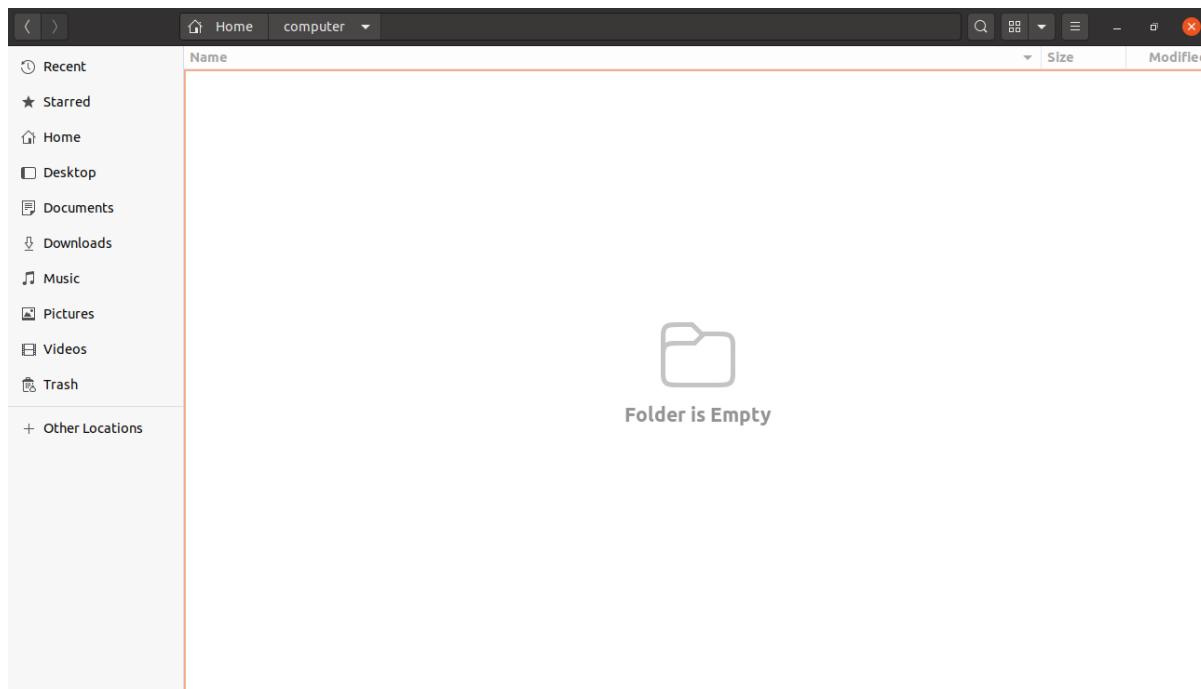
The rm command is used to delete directories and the contents within them. If you only want to delete the directory — as an alternative to rmdir — use rm -r.

Note: Be very careful with this command and double-check which directory you are in. This will delete everything and there is no undo.





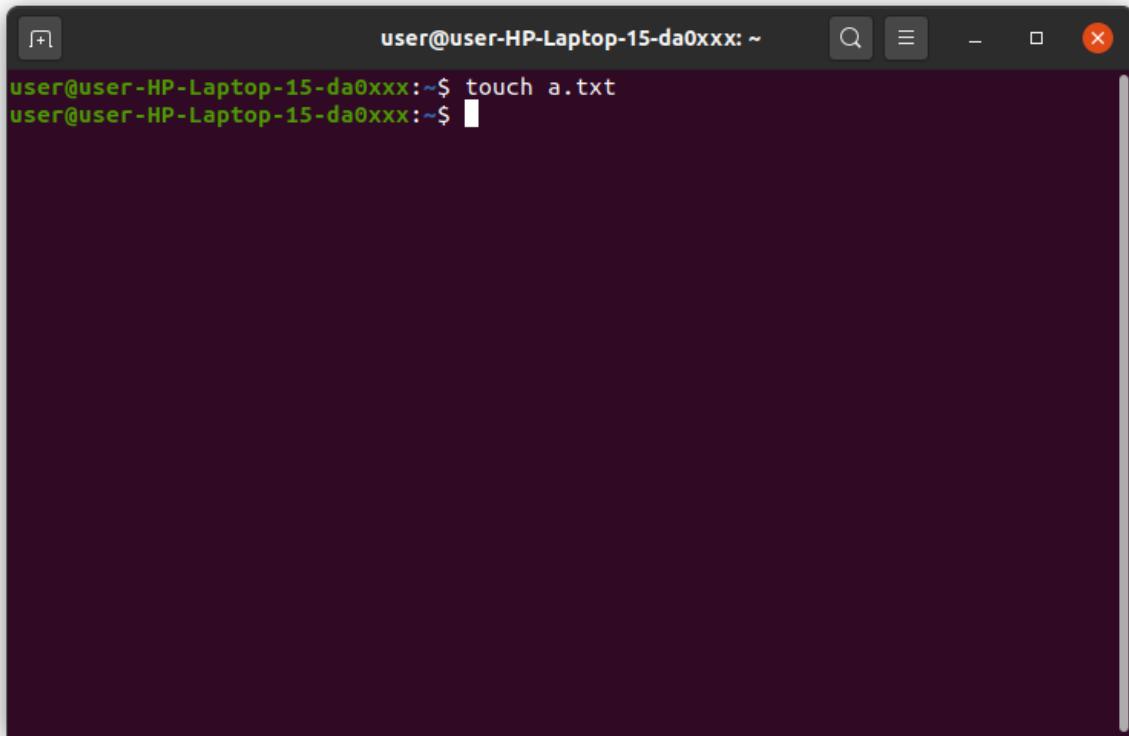
```
user@user-HP-Laptop-15-da0xxx:~/computer$ cd computer
user@user-HP-Laptop-15-da0xxx:~/computer$ rm heloo.txt
user@user-HP-Laptop-15-da0xxx:~/computer$
```



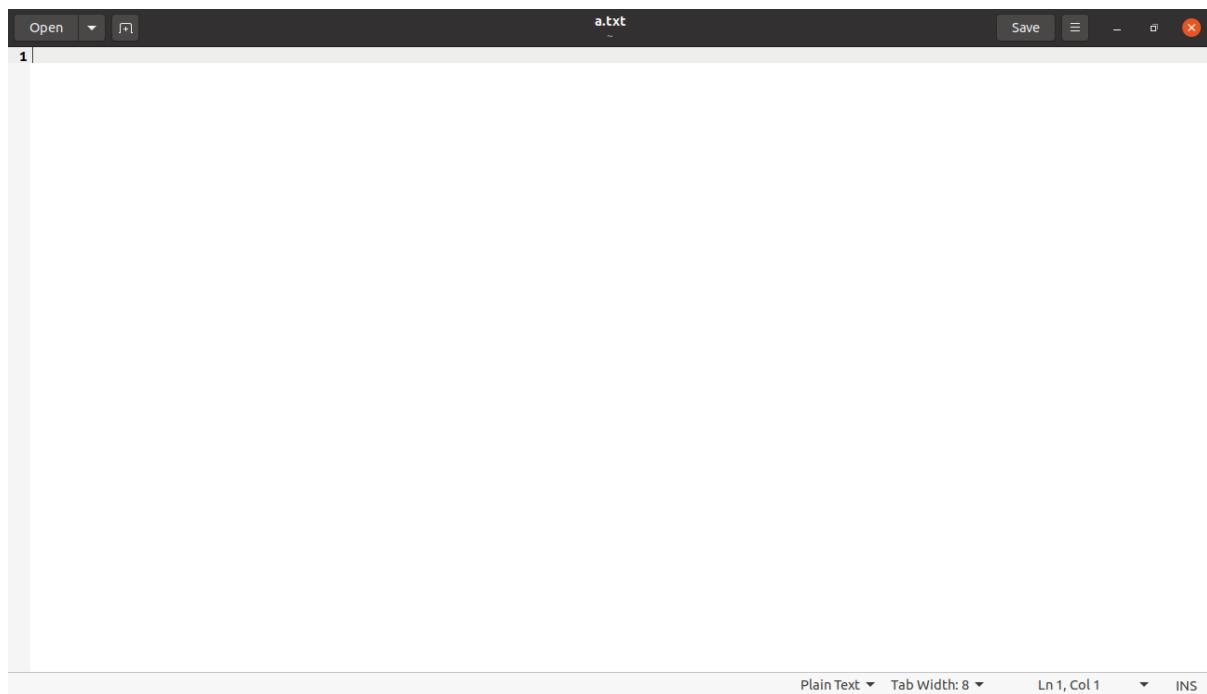
## 10. touch

The touch command allows you to create a blank new file through the Linux command line.

As an example, enter `touch /home/username/Documents/Web.html` to create an HTML file entitled Web under the Documents directory.



A screenshot of a terminal window titled "user@user-HP-Laptop-15-da0xxx: ~". The window has a dark background and light-colored text. It shows the command "touch a.txt" being typed at the prompt. The terminal window includes standard Linux-style icons for closing, minimizing, and maximizing.



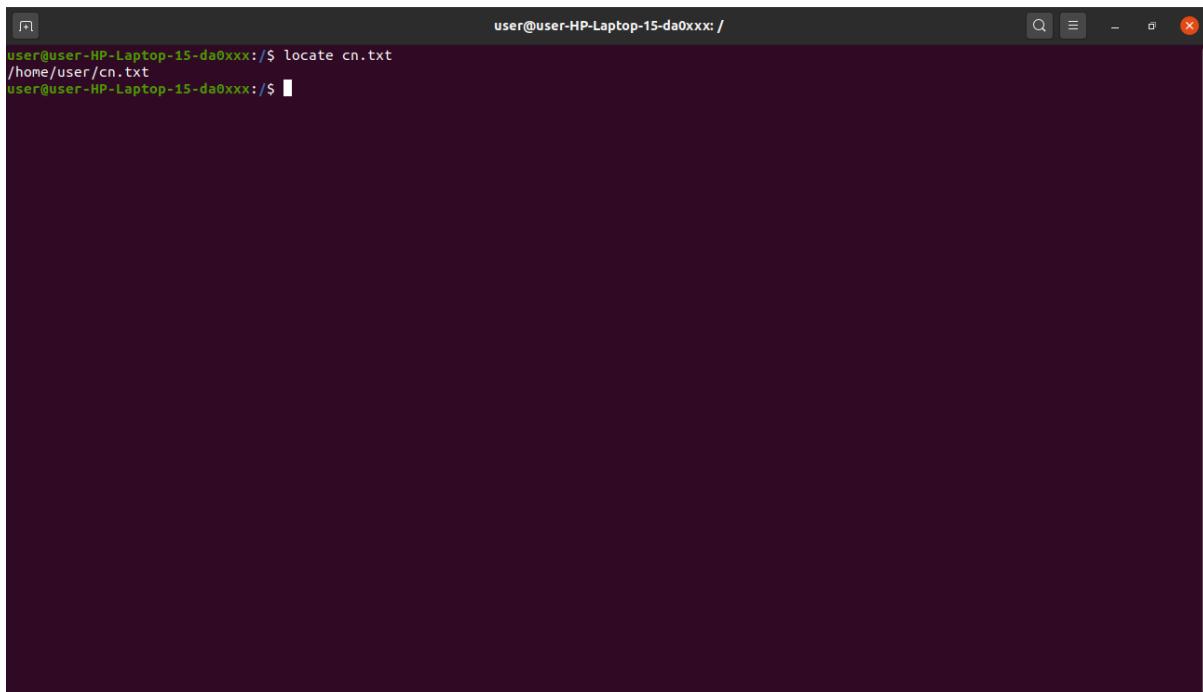
## 11. locate

You can use this command to locate a file, just like the search command in Windows.

What's more, using the `-i` argument along with this command will make it case-insensitive, so you can search for a file even if you don't remember its exact name.

To search for a file that contains two or more words, use an asterisk (\*). For example, locate

-i school\*note command will search for any file that contains the word “school” and “note”, whether it is uppercase or lowercase.



```
user@user-HP-Laptop-15-da0xxx:/$ locate cn.txt
/home/user/cn.txt
user@user-HP-Laptop-15-da0xxx:/$
```

A screenshot of a terminal window titled "user@user-HP-Laptop-15-da0xxx: /". The window shows the command "locate cn.txt" being run, and the output "/home/user/cn.txt" is displayed. The terminal has a dark background with light-colored text and standard window controls at the top.

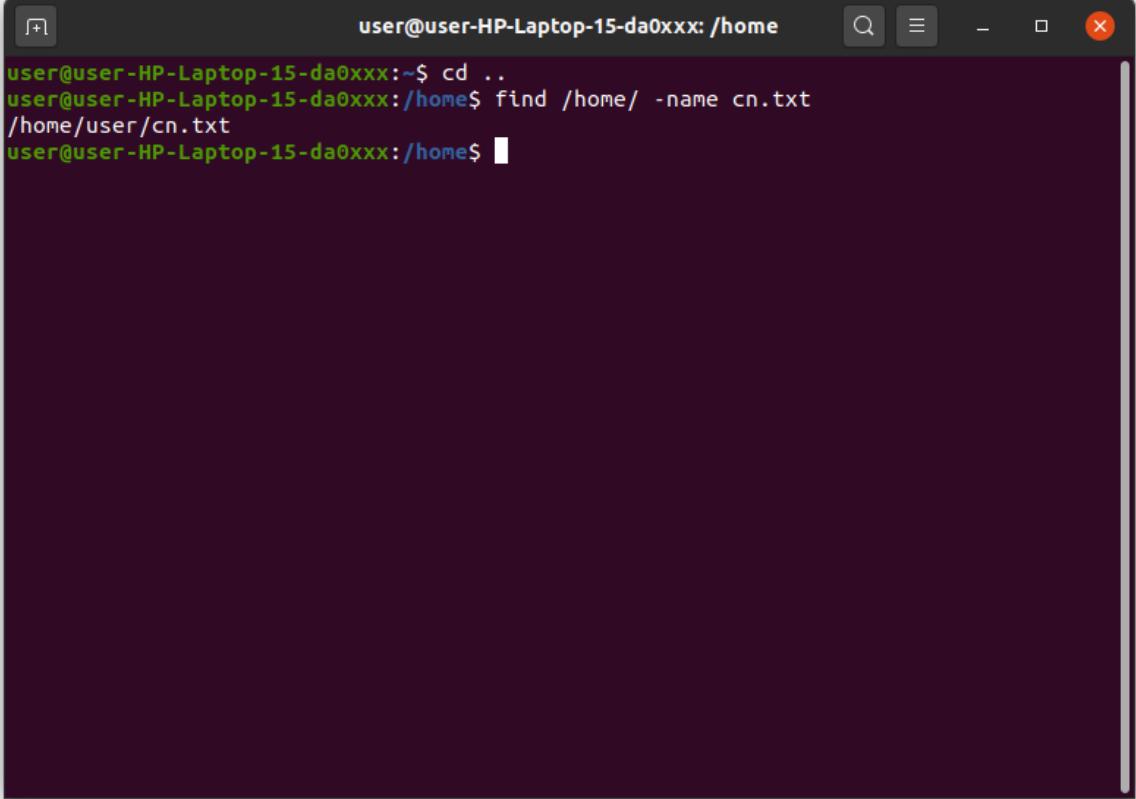
## 12. find

Similar to the locate command, using find also searches for files and directories. The difference is, you use the find command to locate files within a given directory.

As an example, find /home/ -name notes.txt command will search for a file called notes.txt within the home directory and its subdirectories.

Other variations when using the find are:

- To find files in the current directory use, find . -name notes.txt
- To look for directories use, / -type d -name notes. txt



The screenshot shows a terminal window with a dark background and light-colored text. The title bar reads "user@user-HP-Laptop-15-da0xxx: /home". The command entered was "find /home/ -name cn.txt", and the output was "/home/user/cn.txt". The terminal has a standard window interface with minimize, maximize, and close buttons at the top right.

### 13. grep

Another basic Linux command that is undoubtedly helpful for everyday use is grep. It lets you search through all the text in a given file.

To illustrate, grep blue notepad.txt will search for the word blue in the notepad file. Lines that contain the searched word will be displayed fully. You should refer to some grep tutorial

Useful for command line use as well. Usually output of a previous command is piped into the grep command. For example ls -l | grep “kernel”

```

user@user-HP-Laptop-15-da0xxx:~$ cat file1.txt
computer networks
user@user-HP-Laptop-15-da0xxx:~$ grep networks file1.txt
computer networks
user@user-HP-Laptop-15-da0xxx:~$ 

```

## 14. sudo

Short for “SuperUser Do”, this command enables you to perform tasks that require administrative or root permissions. You must have sufficient permissions to use this command.

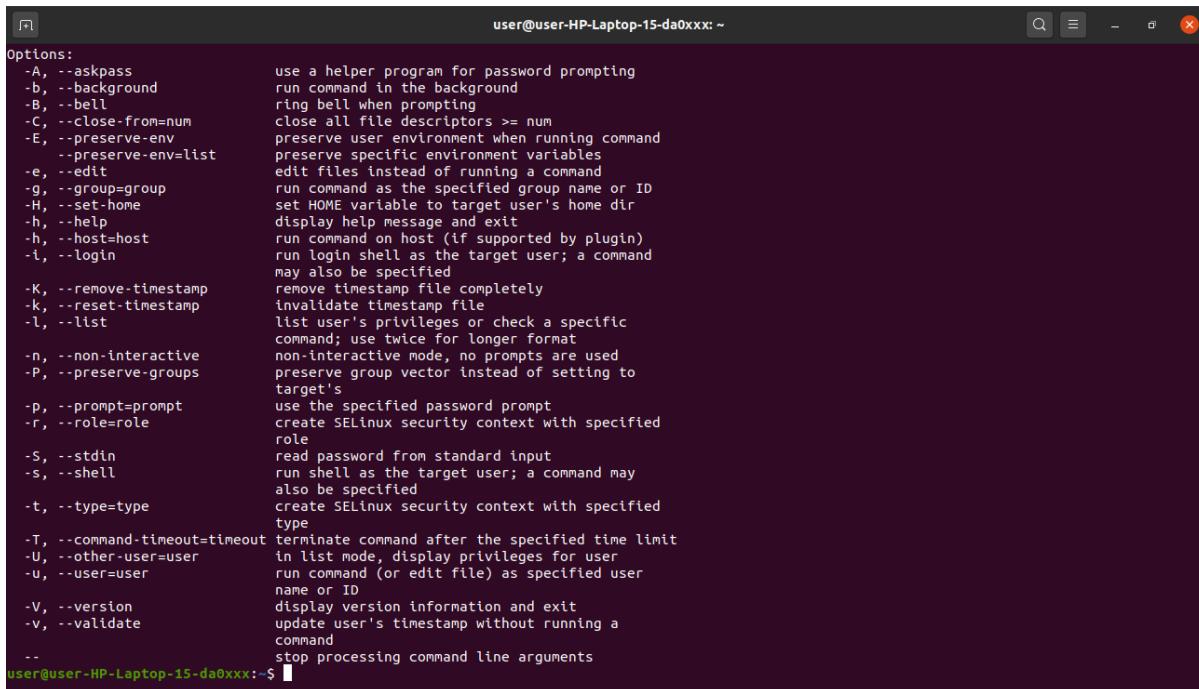
```

user@user-HP-Laptop-15-da0xxx:~$ sudo -h
sudo - execute a command as another user

usage: sudo -h | -K | -k | -V
usage: sudo -v [-AknS] [-g group] [-h host] [-p prompt] [-u user]
usage: sudo -l [-AknS] [-g group] [-h host] [-p prompt] [-U user] [-u user]
       [command]
usage: sudo [-AbEHknPSt] [-r role] [-t type] [-C num] [-g group] [-h host] [-p
           prompt] [-T timeout] [-u user] [VAR=value] [-i|-s] [<command>]
usage: sudo -e [-AknS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p
           prompt] [-T timeout] [-u user] file ...

Options:
  -A, --askpass          use a helper program for password prompting
  -b, --background        run command in the background
  -B, --bell              ring bell when prompting
  -C, --close-from=num   close all file descriptors >= num
  -E, --preserve-env     preserve user environment when running command
  --preserve-env=list    preserve specific environment variables
  -e, --edit              edit files instead of running a command
  -g, --group=group       run command as the specified group name or ID
  -H, --set-home          set HOME variable to target user's home dir
  -h, --help               display help message and exit
  -h, --host=host         run command on host (if supported by plugin)
  -h, --host-host         run command on host (if supported by plugin)
  -i, --login             run login shell as the target user; a command
                         may also be specified
  -K, --remove-timestamp remove timestamp file completely
  -k, --reset-timestamp  invalidate timestamp file
  -l, --list              list user's privileges or check a specific
                         command; use twice for longer format
  -n, --non-interactive   non-interactive mode, no prompts are used
  -P, --preserve-groups  preserve group vector instead of setting to
                         target's
  -p, --prompt=prompt    use the specified password prompt
  -r, --role=role         create SELinux security context with specified
                         role
  -S, --stdin            read password from standard input
  -s, --shell             run shell as the target user; a command may
                         also be specified

```



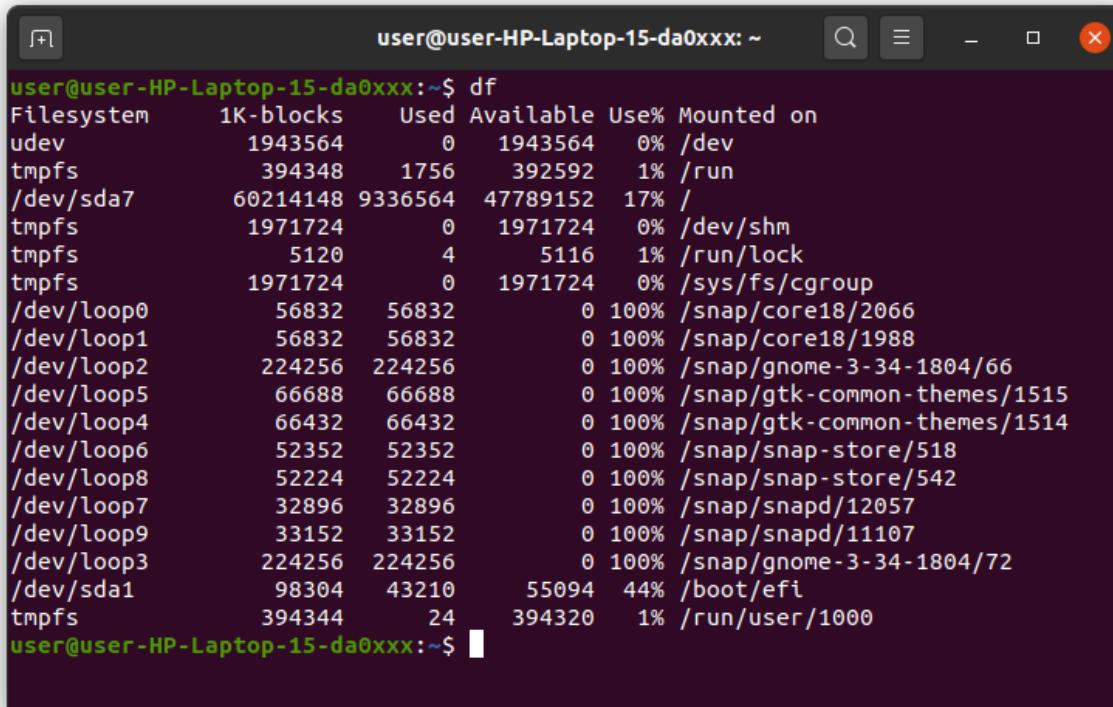
```

user@user-HP-Laptop-15-da0xxx: ~
Options:
-A, --askpass           use a helper program for password prompting
-B, --background        run command in the background
-B, --bell              ring bell when prompting
-C, --close-from=num   close all file descriptors >= num
-E, --preserve-env     preserve user environment when running command
--preserve-env=list    preserve specific environment variables
-e, --edit              edit files instead of running a command
-g, --group=group       run command as the specified group name or ID
-H, --set-home          set HOME variable to target user's home dir
-h, --help               display help message and exit
-h, --host=host         run command on host (if supported by plugin)
-i, --login              run login shell as the target user; a command
                         may also be specified
-K, --remove-timestamp  remove timestamp file completely
-k, --reset-timestamp   invalidate timestamp file
-l, --list               list user's privileges or check a specific
                         command; use twice for longer format
-n, --non-interactive   non-interactive mode, no prompts are used
-P, --preserve-groups   preserve group vector instead of setting to
                         target's
-p, --prompt=prompt     use the specified password prompt
-r, --role=role          create SELinux security context with specified
                         role
-S, --stdin             read password from standard input
-s, --shell              run shell as the target user; a command may
                         also be specified
-t, --type=type          create SELinux security context with specified
                         type
-T, --command-timeout=timeout  terminate command after the specified time limit
-U, --other-user=user    in list mode, display privileges for user
-U, --user=user          run command (or edit file) as specified user
                         name or ID
-V, --version            display version information and exit
-v, --validate           update user's timestamp without running a
                         command
--                          stop processing command line arguments
user@user-HP-Laptop-15-da0xxx: ~

```

## 15. df

Use df command to get a report on the system's disk space usage, shown in percentage and KBs. If you want to see the report in megabytes, type df -m.



Filesystem	1K-blocks	Used	Available	Use%	Mounted on
udev	1943564	0	1943564	0%	/dev
tmpfs	394348	1756	392592	1%	/run
/dev/sda7	60214148	9336564	47789152	17%	/
tmpfs	1971724	0	1971724	0%	/dev/shm
tmpfs	5120	4	5116	1%	/run/lock
tmpfs	1971724	0	1971724	0%	/sys/fs/cgroup
/dev/loop0	56832	56832	0	100%	/snap/core18/2066
/dev/loop1	56832	56832	0	100%	/snap/core18/1988
/dev/loop2	224256	224256	0	100%	/snap/gnome-3-34-1804/66
/dev/loop5	66688	66688	0	100%	/snap/gtk-common-themes/1515
/dev/loop4	66432	66432	0	100%	/snap/gtk-common-themes/1514
/dev/loop6	52352	52352	0	100%	/snap/snap-store/518
/dev/loop8	52224	52224	0	100%	/snap/snap-store/542
/dev/loop7	32896	32896	0	100%	/snap/snapd/12057
/dev/loop9	33152	33152	0	100%	/snap/snapd/11107
/dev/loop3	224256	224256	0	100%	/snap/gnome-3-34-1804/72
/dev/sda1	98304	43210	55094	44%	/boot/efi
tmpfs	394344	24	394320	1%	/run/user/1000

```

user@user-HP-Laptop-15-da0xxx:~$ df -m
Filesystem      1M-blocks  Used Available Use% Mounted on
udev              1899      0     1899   0% /dev
tmpfs             386       2     384   1% /run
/dev/sda7      58803  9118    46669  17% /
tmpfs             1926      0     1926   0% /dev/shm
tmpfs               5       1       5   1% /run/lock
tmpfs             1926      0     1926   0% /sys/fs/cgroup
/dev/loop0            56      56      0 100% /snap/core18/2066
/dev/loop1            56      56      0 100% /snap/core18/1988
/dev/loop2            219     219      0 100% /snap/gnome-3-34-1804/66
/dev/loop5            66      66      0 100% /snap/gtk-common-themes/1515
/dev/loop4            65      65      0 100% /snap/gtk-common-themes/1514
/dev/loop6            52      52      0 100% /snap/snap-store/518
/dev/loop8            51      51      0 100% /snap/snap-store/542
/dev/loop7            33      33      0 100% /snap/snapd/12057
/dev/loop9            33      33      0 100% /snap/snapd/11107
/dev/loop3            219     219      0 100% /snap/gnome-3-34-1804/72
/dev/sda1             96      43      54 44% /boot/efi
tmpfs             386       1     386   1% /run/user/1000
user@user-HP-Laptop-15-da0xxx:~$ 

```

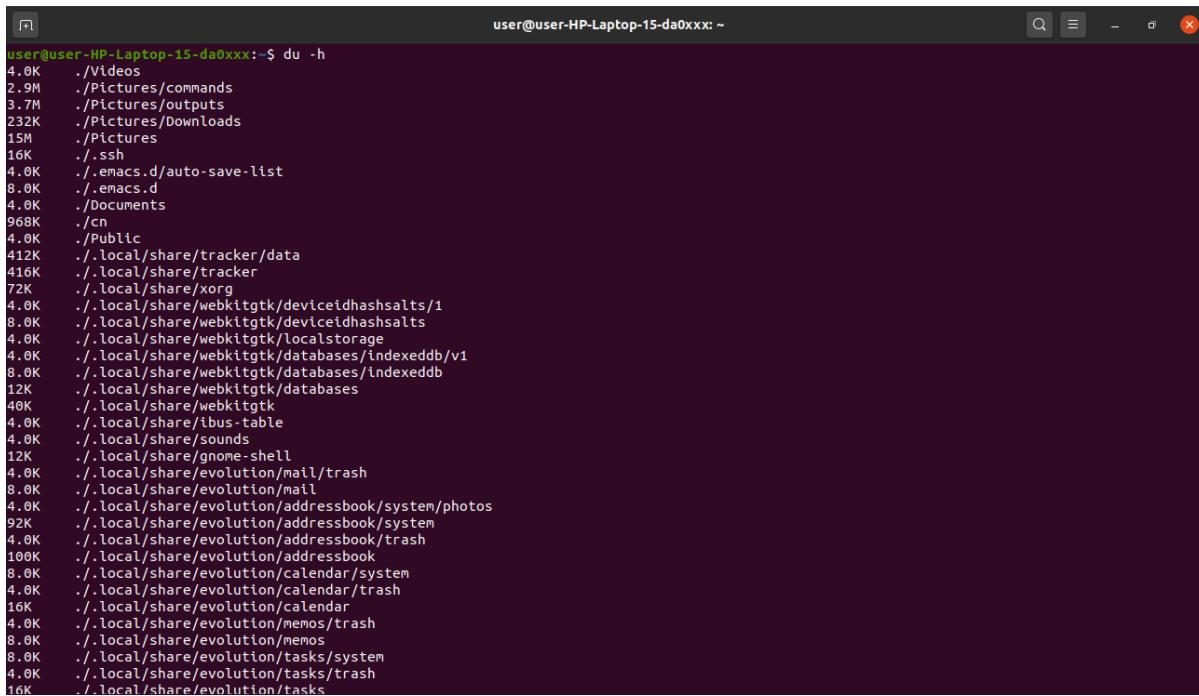
## 16. du

If you want to check how much space a file or a directory takes, the du (Disk Usage) command is the answer. However, the disk usage summary will show disk block numbers instead of the usual size format. If you want to see it in bytes, kilobytes, and megabytes, add the -h argument to the command line.

```

user@user-HP-Laptop-15-da0xxx:~$ du -h
4 ./Videos
1768 ./Pictures/commands
3736 ./Pictures/outputs
232 ./Pictures/downloads
13360 ./Pictures
16 ./ssh
4 ./emacs.d/auto-save-list
8 ./emacs.d
4 ./Documents
968 ./cn
4 ./Public
408 ./local/share/tracker/data
412 ./local/share/tracker
72 ./local/share/xorg
4 ./local/share/webkitgtk/deviceidhashsalts/1
8 ./local/share/webkitgtk/deviceidhashsalts
4 ./local/share/webkitgtk/localstorage
4 ./local/share/webkitgtk/databases/indexeddb/v1
8 ./local/share/webkitgtk/databases/indexeddb
12 ./local/share/webkitgtk/databases
40 ./local/share/webkitgtk
4 ./local/share/ibus-table
4 ./local/share/sounds
4 ./local/share/gnome-shell
4 ./local/share/evolution/mail/trash
8 ./local/share/evolution/mail
4 ./local/share/evolution/addressbook/system/photos
92 ./local/share/evolution/addressbook/system
4 ./local/share/evolution/addressbook/trash
100 ./local/share/evolution/addressbook
8 ./local/share/evolution/calendar/system
4 ./local/share/evolution/calendar/trash
16 ./local/share/evolution/calendar
4 ./local/share/evolution/memos/trash
8 ./local/share/evolution/memos
8 ./local/share/evolution/tasks/system
4 ./local/share/evolution/tasks/trash
4 ./local/share/evolution/tasks
16 ./local/share/evolution/tasks

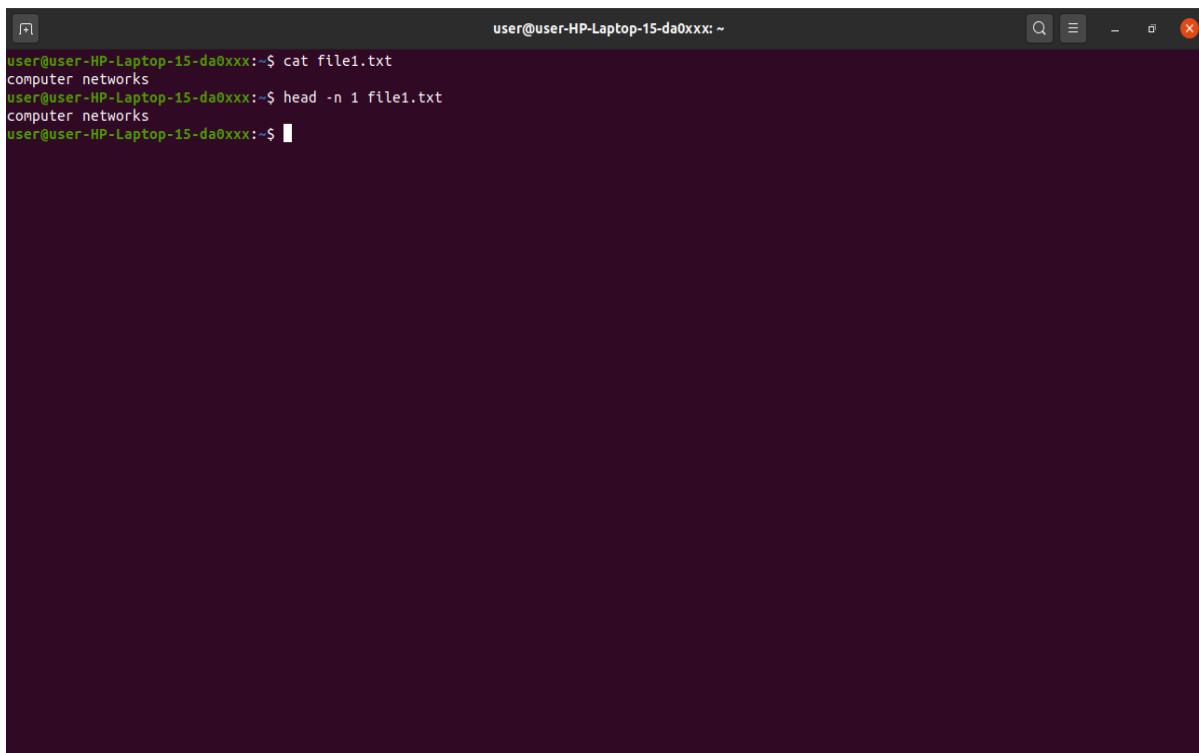
```



```
user@user-HP-Laptop-15-da0xxx:~$ du -h
4.0K ./Videos
2.9M ./Pictures/commands
3.7M ./Pictures/outputs
232K ./Pictures/Downloads
15M ./Pictures
16K ./ssh
4.0K ./emacs.d/auto-save-list
8.0K ./emacs.d
4.0K ./Documents
968K ./cn
4.0K ./Public
412K ./local/share/tracker/data
416K ./local/share/tracker
72K ./local/share/xorg
4.0K ./local/share/webkitgtk/deviceidhashsalts/1
8.0K ./local/share/webkitgtk/deviceidhashsalts
4.0K ./local/share/webkitgtk/localstorage
4.0K ./local/share/webkitgtk/databases/indexeddb/v1
8.0K ./local/share/webkitgtk/databases/indexeddb
12K ./local/share/webkitgtk/databases
40K ./local/share/webkitgtk
4.0K ./local/share/ibus-table
4.0K ./local/share/sounds
12K ./local/share/gnome-shell
4.0K ./local/share/evolution/mail/trash
8.0K ./local/share/evolution/mail
4.0K ./local/share/evolution/addressbook/system/photos
92K ./local/share/evolution/addressbook/system
4.0K ./local/share/evolution/addressbook/trash
100K ./local/share/evolution/addressbook
8.0K ./local/share/evolution/calendar/system
4.0K ./local/share/evolution/calendar/trash
16K ./local/share/evolution/calendar
4.0K ./local/share/evolution/memos/trash
8.0K ./local/share/evolution/memos
8.0K ./local/share/evolution/tasks/system
4.0K ./local/share/evolution/tasks/trash
16K ./local/share/evolution/tasks
```

## 17. head

The head command is used to view the first lines of any text file. By default, it will show the first ten lines, but you can change this number to your liking. For example, if you only want to show the first five lines, type head -n 5 filename.ext. ( Read the manual )

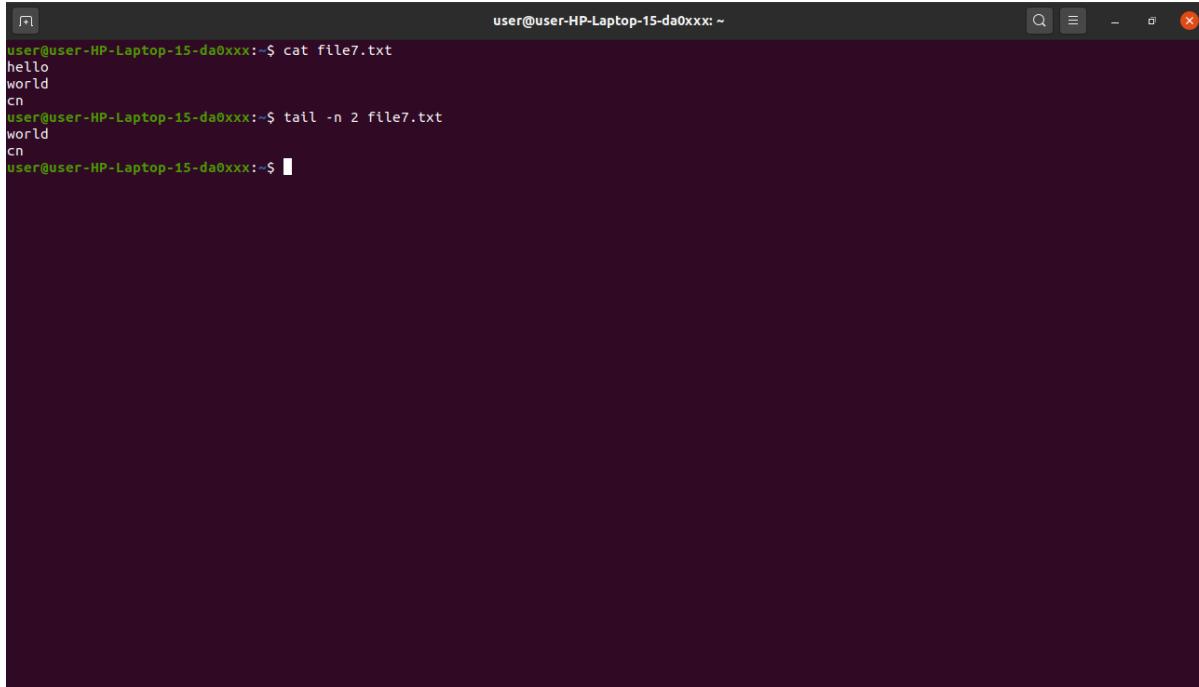


```
user@user-HP-Laptop-15-da0xxx:~$ cat file1.txt
computer networks
user@user-HP-Laptop-15-da0xxx:~$ head -n 1 file1.txt
computer networks
user@user-HP-Laptop-15-da0xxx:~$
```

## 18. tail

This one has a similar function to the head command, but instead of showing the first lines,

the tail command will display the last ten lines of a text file. For example, tail -n filename.ext.

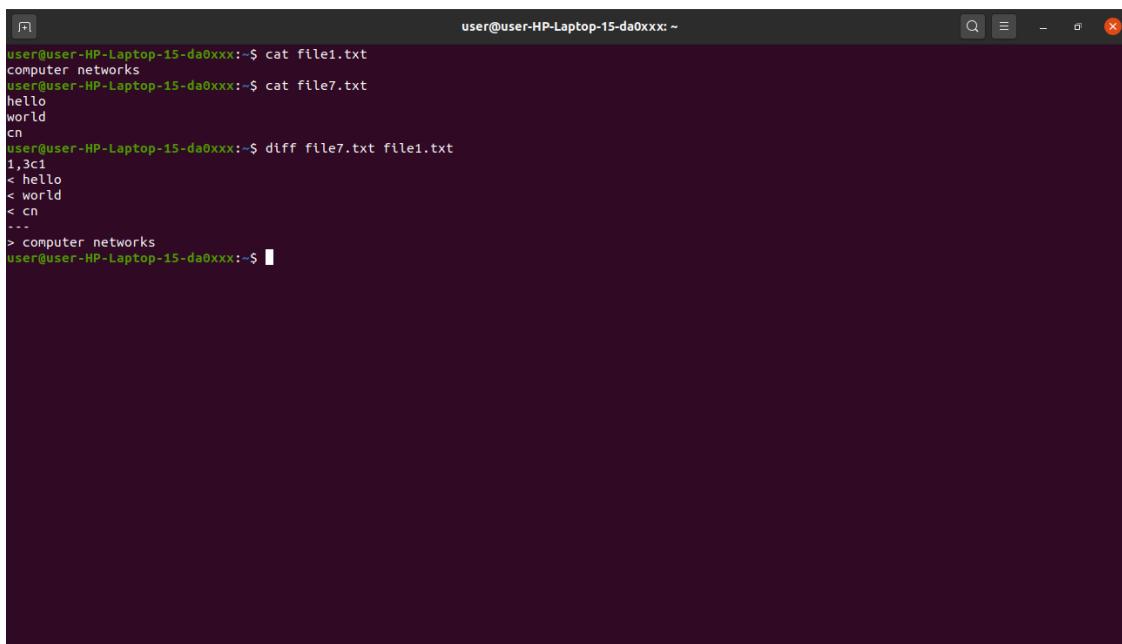


```
user@user-HP-Laptop-15-da0xxx:~$ cat file7.txt
hello
world
cn
user@user-HP-Laptop-15-da0xxx:~$ tail -n 2 file7.txt
world
cn
user@user-HP-Laptop-15-da0xxx:~$
```

## 19. diff

Short for difference, the diff command compares the contents of two files line by line. After analyzing the files, it will output the lines that do not match. Programmers often use this command when they need to make program alterations instead of rewriting the entire source code.

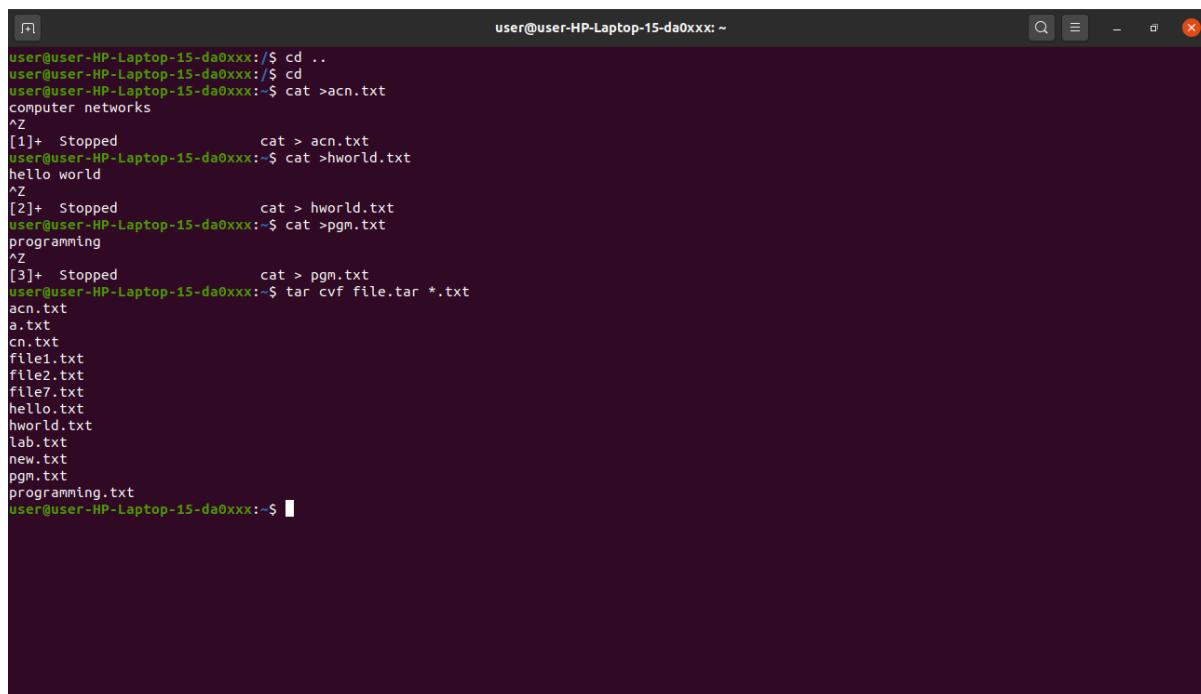
The simplest form of this command is diff file1.ext file2.ext



```
user@user-HP-Laptop-15-da0xxx:~$ cat file1.txt
computer networks
user@user-HP-Laptop-15-da0xxx:~$ cat file7.txt
hello
world
cn
user@user-HP-Laptop-15-da0xxx:~$ diff file7.txt file1.txt
1,3c1
< hello
< world
< cn
...
> computer networks
user@user-HP-Laptop-15-da0xxx:~$
```

## 20. tar

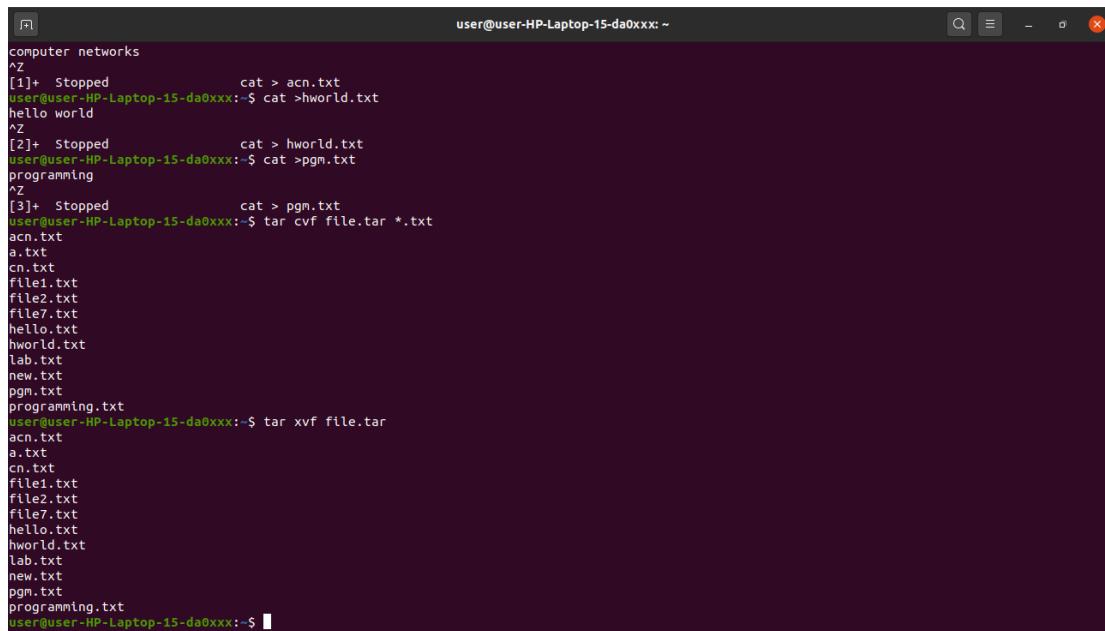
The tar command is the most used command to archive multiple files into a tarball — a common Linux file format that is similar to zip format, with compression being optional. This command is quite complex with a long list of functions such as adding new files into an existing archive, listing the content of an archive, extracting the content from an archive, and many more. Read some tutorial on net.



```

user@user-HP-Laptop-15-da0xxx:~/Desktop$ cd ..
user@user-HP-Laptop-15-da0xxx:~/Desktop$ cd ..
user@user-HP-Laptop-15-da0xxx:~$ cat >acn.txt
computer networks
^Z
[1]+  Stopped                  cat > acn.txt
user@user-HP-Laptop-15-da0xxx:~$ cat >hworld.txt
hello world
^Z
[2]+  Stopped                  cat > hworld.txt
user@user-HP-Laptop-15-da0xxx:~$ cat >pgm.txt
programming
^Z
[3]+  Stopped                  cat > pgm.txt
user@user-HP-Laptop-15-da0xxx:~$ tar cvf file.tar *.txt
acn.txt
a.txt
cn.txt
file1.txt
file2.txt
file7.txt
hello.txt
hworld.txt
lab.txt
new.txt
pgm.txt
programming.txt
user@user-HP-Laptop-15-da0xxx:~$ 

```



```

user@user-HP-Laptop-15-da0xxx:~/Desktop$ cd ..
user@user-HP-Laptop-15-da0xxx:~/Desktop$ cd ..
user@user-HP-Laptop-15-da0xxx:~$ cat >acn.txt
computer networks
^Z
[1]+  Stopped                  cat > acn.txt
user@user-HP-Laptop-15-da0xxx:~$ cat >hworld.txt
hello world
^Z
[2]+  Stopped                  cat > hworld.txt
user@user-HP-Laptop-15-da0xxx:~$ cat >pgm.txt
programming
^Z
[3]+  Stopped                  cat > pgm.txt
user@user-HP-Laptop-15-da0xxx:~$ tar cvf file.tar *.txt
acn.txt
a.txt
cn.txt
file1.txt
file2.txt
file7.txt
hello.txt
hworld.txt
lab.txt
new.txt
pgm.txt
programming.txt
user@user-HP-Laptop-15-da0xxx:~$ tar xvf file.tar
acn.txt
a.txt
cn.txt
file1.txt
file2.txt
file7.txt
hello.txt
hworld.txt
lab.txt
new.txt
pgm.txt
programming.txt
user@user-HP-Laptop-15-da0xxx:~$ 

```

## 21. chmod

chmod is another Linux command, used to change the read, write, and execute permissions of files and directories. Read about permissions and how to manipulate them .

```

user@user-HP-Laptop-15-da0xxx:~$ ls -l
total 80
-rw-rw-r-- 1 user user    0 Jun 12 01:13 a.txt
-rw-rw-r-- 1 user user 458 Jan  3 00:59 bitstring.c
drwxrwxr-x 2 user user 4096 Jun 11 02:18 cn
drwxrwxr-x 2 user user 4096 Jun 11 18:19 computer
drwxrwxr-x 5 user user 4096 Mar 21 14:32 cpgms
drwxr-xr-x 2 user user 4096 Dec  2  2020 Desktop
drwxr-xr-x 2 user user 4096 Dec  2  2020 Documents
-rw-rw-r-- 1 user user   18 Jun 11 17:59 file1.txt
-rw-rw-r-- 1 user user   18 Jun 11 18:04 file2.txt
-rw-rw-r-- 1 user user   15 Jun 12 01:50 file7.txt
drwxrwxr-x 3 user user 4096 Dec 28 01:32 GIT
-rw-rw-r-- 1 user user    0 Jun 11 17:54 hello.txt
-rw-rw-r-- 1 user user   15 Jun 11 17:54 lab.txt
drwxr-xr-x 2 user user 4096 Dec  2  2020 Music
-rw-rw-r-- 1 user user   29 Jun 10 03:58 '#newfile.txt#'
-rw-rw-r-- 1 user user   12 Jun 11 18:09 new.txt
drwxr-xr-x 5 user user 4096 Jun 12 01:35 Pictures
-rw-rw-r-- 1 user user 1625 Mar 21 14:18 prims.c
-rw-rw-r-- 1 user user    0 Jun 11 17:49 programming
-rw-rw-r-- 1 user user   15 Jun 11 17:54 programming.txt
drwxr-xr-x 2 user user 4096 Dec  2  2020 Public
drwxr-xr-x 2 user user 4096 Dec  2  2020 Templates
drwxr-xr-x 2 user user 4096 Dec  2  2020 Videos
user@user-HP-Laptop-15-da0xxx:~$ ls -l notes.txt
ls: cannot access 'notes.txt': No such file or directory
user@user-HP-Laptop-15-da0xxx:~$ ls -l file1.txt
-rw-rw-r-- 1 user user 18 Jun 11 17:59 file1.txt
user@user-HP-Laptop-15-da0xxx:~$ chmod u=rw,og=r file1.txt
user@user-HP-Laptop-15-da0xxx:~$ ls -l file1.txt
-rw-r--r-- 1 user user 18 Jun 11 17:59 file1.txt
user@user-HP-Laptop-15-da0xxx:~$ 

```

## 22. chown

In Linux, all files are owned by a specific user. The chown command enables you to change or transfer the ownership of a file to the specified username. For instance, chown linuxuser2 file.ext will make linuxuser2 as the owner of the file.ext.

```

user@user-HP-Laptop-15-da0xxx:~$ cat >cn.txt
computer
networks
programming
lab
hello
world
^Z
[1]+  Stopped                  cat > cn.txt
user@user-HP-Laptop-15-da0xxx:~$ cat cn.txt
computer
networks
programming
lab
hello
world
user@user-HP-Laptop-15-da0xxx:~$ ls
a.txt      cn.txt    Desktop    file2.txt  hello.txt  '#newfile.txt#'  prims.c      Public
bitstring.c  computer  Documents  file7.txt  lab.txt    new.txt    programming  Templates
cn          cpgms    file1.txt  GIT        Music     Pictures   programming.txt  Videos
user@user-HP-Laptop-15-da0xxx:~$ ls -l cn.txt
-rw-rw-r-- 1 user user 46 Jun 13 17:45 cn.txt
user@user-HP-Laptop-15-da0xxx:~$ chown gowri cn.txt
chown: changing ownership of 'cn.txt': Operation not permitted
user@user-HP-Laptop-15-da0xxx:~$ 

```

## 23. ps

Ps command will display all current processes along with their process ids (PID) . Read manuals for various options.

```
user@user-HP-Laptop-15-da0xxx:~$ ps
  PID TTY      TIME CMD
 2742 pts/0    00:00:00 bash
 2996 pts/0    00:00:00 ps
user@user-HP-Laptop-15-da0xxx:~$
```

```
user@user-HP-Laptop-15-da0xxx:~$ ps -ux
USER   PID %CPU %MEM   VSZ RSS TTY      STAT START  TIME COMMAND
user   1449  0.1  0.2 19272 10628 ?        Ss 17:27  0:00 /lib/systemd/
user   1451  0.0  0.0 168984 3556 ?        S 17:27  0:00 (sd-pam)
user   1464  0.0  0.4 2277664 19652 ?       S=>l 17:27  0:00 /usr/bin/pulse
user   1466  0.0  0.6 520092 24460 ?       SNSl 17:27  0:00 /usr/libexec/
user   1468  0.1  0.1  8740  5816 ?        Ss 17:27  0:00 /usr/bin/dbus
user   1472  0.0  0.2 248804 7968 ?        SL 17:27  0:00 /usr/bin/gnom
user   1477  0.0  0.1 248316 7796 ?        Ssl 17:27  0:00 /usr/libexec/
user   1482  0.0  0.2 382056 8552 ?        SL 17:27  0:00 /usr/libexec/
user   1502  0.0  0.2 326040 11804 ?       Ssl 17:27  0:00 /usr/libexec/
user   1507  0.0  0.1 244500 6192 ?        Ssl 17:27  0:00 /usr/libexec/
user   1511  0.0  0.9 550416 36356 ?       SL 17:27  0:00 /usr/libexec/
user   1518  0.0  0.2 327272 11556 ?       SL 17:27  0:00 /usr/libexec/
user   1525  0.0  0.1 325352 7572 ?       Ssl 17:27  0:00 /usr/libexec/
user   1531  0.0  0.1 246596 6684 ?       Ssl 17:27  0:00 /usr/libexec/
user   1532  0.0  0.1 172628 6596 tty2    Ssl+ 17:27  0:00 /usr/lib/gdm3
user   1536  0.0  0.1 244320 6180 ?       Ssl 17:27  0:00 /usr/libexec/
user   1543  4.4  2.2 865960 88972 tty2    S+ 17:27  0:18 /usr/lib/xorg
user   1570  0.0  0.3 199404 15604 tty2    S+ 17:27  0:00 /usr/libexec/
user   1644  0.0  0.0 6032  456 ?         Ss 17:27  0:00 /usr/bin/ssh-
user   1661  0.2  0.2 323528 9596 ?       Ssl 17:27  0:00 /usr/bin/ibus
user   1668  0.0  0.2 248868 9056 ?       SL 17:27  0:00 /usr/libexec/
user   1669  0.1  1.8 728184 73536 ?      SL 17:27  0:00 /usr/libexec/
user   1672  0.4  1.5 686388 61652 ?      SL 17:27  0:01 /usr/libexec/
user   1678  0.0  0.7 208996 29996 ?      SL 17:27  0:00 /usr/libexec/
user   1681  0.0  0.2 248848 9016 ?       SL 17:27  0:00 /usr/libexec/
user   1686  0.0  0.2 309820 9488 ?       Ssl 17:27  0:00 /usr/libexec/
user   1695  0.0  0.1  7216  4288 ?        S 17:27  0:00 /usr/bin/dbus
user   1707  0.0  0.1 162812 7640 ?       SL 17:27  0:00 /usr/libexec/
user   1716  0.0  0.2 1012088 10572 ?      Ssl 17:27  0:00 /usr/libexec/
user   1725  0.0  0.1  98672  4328 ?       Ssl 17:27  0:00 /usr/libexec/
user   1727  0.0  0.1 466048 6480 ?       Ssl 17:27  0:00 /usr/libexec/
user   1733  0.0  0.4 568944 17520 ?      Ssl 17:27  0:00 /usr/libexec/
user   1738  0.0  0.1 244220 4932 ?       Ssl 17:27  0:00 /usr/libexec/
user   1748  0.0  0.2 175172  9124 ?       SL 17:27  0:00 /usr/libexec/
user   1752  0.1  0.8 506680 32264 ?      Ssl 17:27  0:00 /usr/libexec/
user   1762  0.0  0.1 156208  5732 ?       SL 17:27  0:00 /usr/libexec/
user   1789  5.5  6.3 4552988 248976 ?     S=>l 17:27  0:23 /usr/bin/gnom
```

## 24. Kill

If you have an unresponsive program, you can terminate it manually by using the kill command. It will send a certain signal to the misbehaving app and instructs the app to terminate itself.

There is a total of sixty-four signals that you can use, but people usually only use two signals:

- SIGTERM (15) — requests a program to stop running and gives it some time to save all of its progress. If you don't specify the signal when entering the kill command, this signal will be used.
- SIGKILL (9) — forces programs to stop immediately. Unsaved progress will be lost.

Besides knowing the signals, you also need to know the process identification number (PID) of the program you want to kill. If you don't know the PID, simply run the command ps ux.

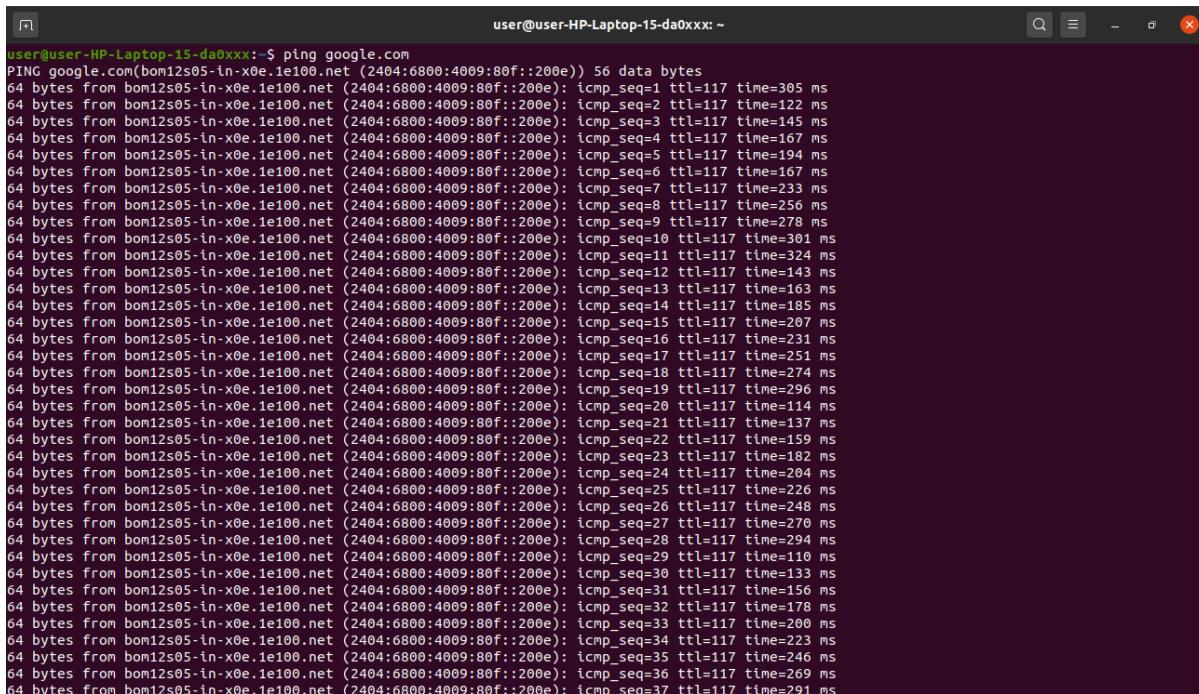
After knowing what signal you want to use and the PID of the program, enter the following syntax:

kill [signal option] PID.

You can issue kill -9 PID

## 25. ping

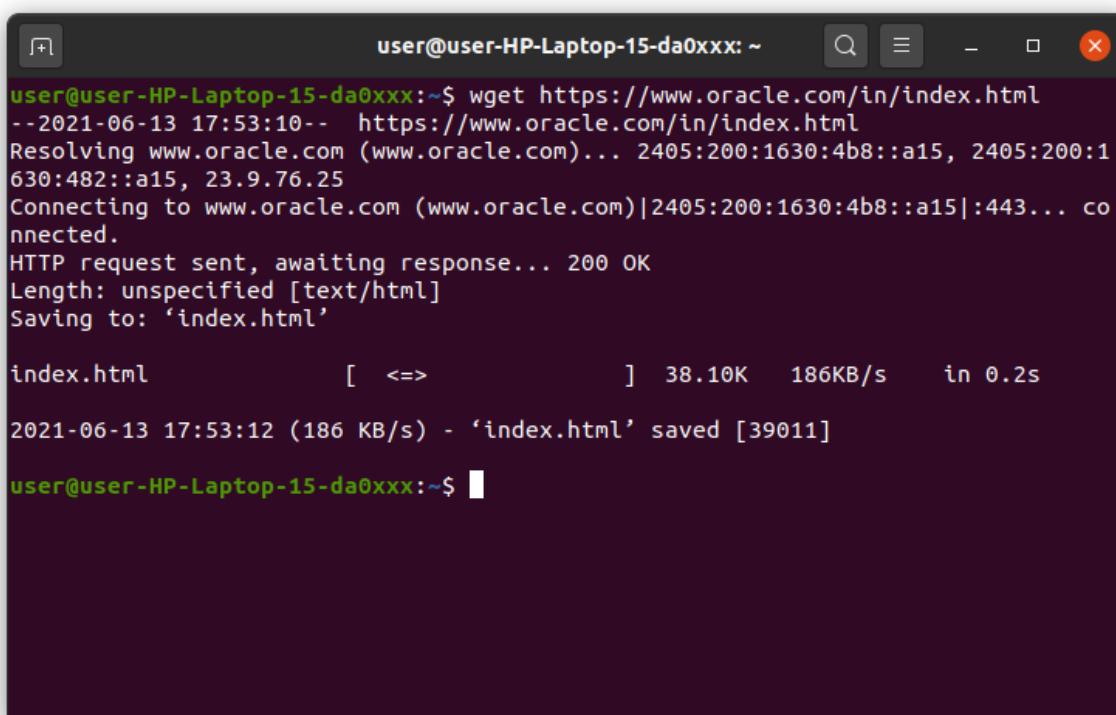
Use the ping command to check your connectivity status to a server. For example, by simply entering ping google.com, the command will check whether you're able to connect to Google and also measure the response time.



```
user@user-HP-Laptop-15-da0xxx:~$ ping google.com
PING google.com(bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e)) 56 data bytes
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=1 ttl=117 time=305 ms
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=2 ttl=117 time=122 ms
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=3 ttl=117 time=145 ms
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=4 ttl=117 time=167 ms
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=5 ttl=117 time=194 ms
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=6 ttl=117 time=167 ms
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=7 ttl=117 time=233 ms
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=8 ttl=117 time=256 ms
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=9 ttl=117 time=278 ms
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=10 ttl=117 time=301 ms
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=11 ttl=117 time=324 ms
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=12 ttl=117 time=143 ms
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=13 ttl=117 time=163 ms
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=14 ttl=117 time=185 ms
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=15 ttl=117 time=207 ms
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=16 ttl=117 time=231 ms
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=17 ttl=117 time=251 ms
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=18 ttl=117 time=274 ms
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=19 ttl=117 time=296 ms
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=20 ttl=117 time=114 ms
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=21 ttl=117 time=137 ms
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=22 ttl=117 time=159 ms
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=23 ttl=117 time=182 ms
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=24 ttl=117 time=204 ms
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=25 ttl=117 time=226 ms
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=26 ttl=117 time=248 ms
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=27 ttl=117 time=270 ms
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=28 ttl=117 time=294 ms
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=29 ttl=117 time=110 ms
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=30 ttl=117 time=133 ms
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=31 ttl=117 time=156 ms
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=32 ttl=117 time=178 ms
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=33 ttl=117 time=200 ms
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=34 ttl=117 time=223 ms
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=35 ttl=117 time=246 ms
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=36 ttl=117 time=269 ms
64 bytes from bom12s05-in-x0e.1e100.net (2404:6800:4009:80f::200e): icmp_seq=37 ttl=117 time=291 ms
```

## 26. wget

The Linux command line is super useful — you can even download files from the internet with the help of the wget command. To do so, simply type wget followed by the download link.



```
user@user-HP-Laptop-15-da0xxx:~$ wget https://www.oracle.com/in/index.html
--2021-06-13 17:53:10-- https://www.oracle.com/in/index.html
Resolving www.oracle.com (www.oracle.com)... 2405:200:1630:4b8::a15, 2405:200:1630:482::a15, 23.9.76.25
Connecting to www.oracle.com (www.oracle.com)|2405:200:1630:4b8::a15|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'index.html'

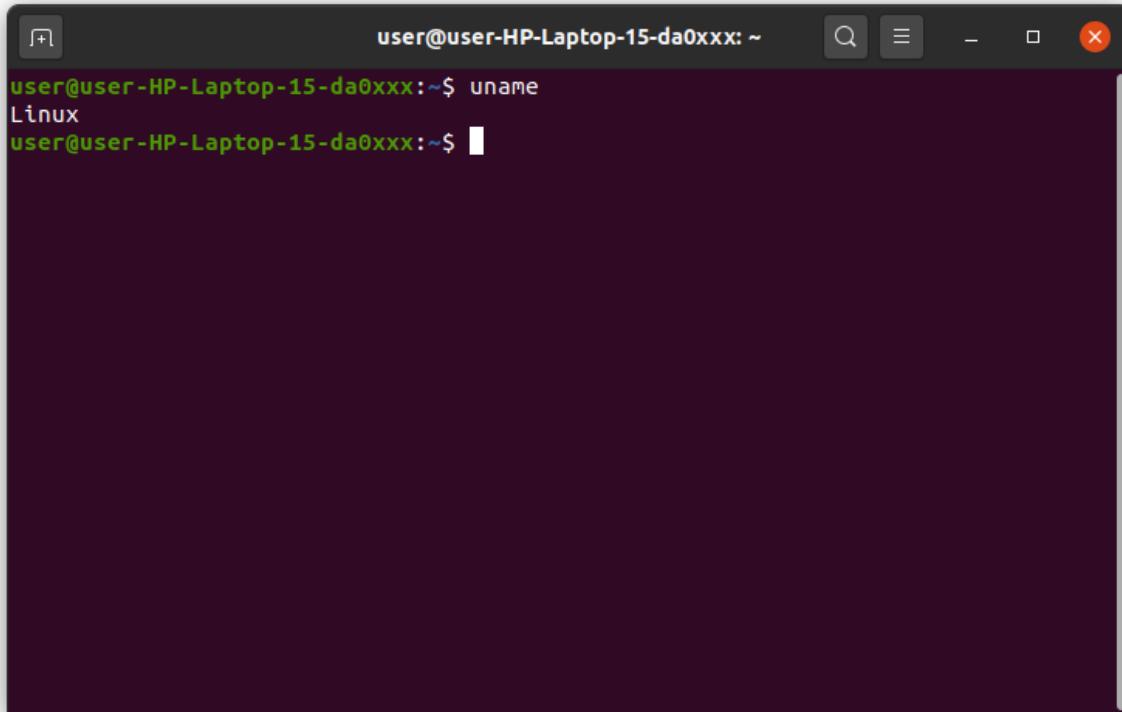
index.html      [ =>          ]  38.10K   186KB/s   in 0.2s

2021-06-13 17:53:12 (186 KB/s) - 'index.html' saved [39011]

user@user-HP-Laptop-15-da0xxx:~$
```

## 27. uname

The uname command, short for Unix Name, will print detailed information about your Linux system like the machine name, operating system, kernel, and so on.



A screenshot of a terminal window titled "user@user-HP-Laptop-15-da0xxx: ~". The window contains the command "uname" followed by the output "Linux". The terminal has a dark background and light-colored text. The title bar includes standard window controls (minimize, maximize, close) and a search icon.

```
user@user-HP-Laptop-15-da0xxx:~$ uname
Linux
user@user-HP-Laptop-15-da0xxx:~$
```

## 28. top

As a terminal equivalent to Task Manager in Windows, the top command will display a list of running processes and how much CPU each process uses. It's very useful to monitor system resource usage, especially knowing which process needs to be terminated because it consumes too many resources.

```

top - 17:55:37 up 29 min, 1 user, load average: 0.53, 0.28, 0.32
Tasks: 213 total, 2 running, 211 sleeping, 0 stopped, 0 zombie
%Cpu(s): 5.9 us, 4.1 sy, 0.0 ni, 89.0 id, 0.0 wa, 0.0 hi, 0.9 st, 0.0 st
MiB Mem : 3851.0 total, 1822.5 free, 1032.1 used, 996.4 buff/cache
MiB Swap: 2048.0 total, 2048.0 free, 0.0 used. 2420.1 avail Mem

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
1543 user 20 0 879916 90132 57736 R 11.1 2.3 0:56.43 Xorg
1789 user 20 0 4560456 255944 103488 S 11.1 6.5 1:05.16 gnome-shell
3825 user 20 0 971000 52368 39688 S 9.3 1.3 0:03.59 gnome-terminal-
963 rtkit 21 1 152916 3032 2792 S 3.7 0.1 0:00.06 rtkit-daemon
8 root 20 0 0 0 0 I 1.9 0.0 0:01.38 kworker/u8:0-i915
44 root 20 0 0 0 0 D 1.9 0.0 0:02.04 kworker/u8:1+events_unbound
4131 user 20 0 20640 3932 3176 R 1.9 0.1 0:00.04 top
1 root 20 0 167588 11792 8616 S 0.0 0.3 0:02.80 systemd
2 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kthreadd
3 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu_gp
4 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu_par_gp
6 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/0:0H-kblockd
9 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 mm_percpu_wq

```

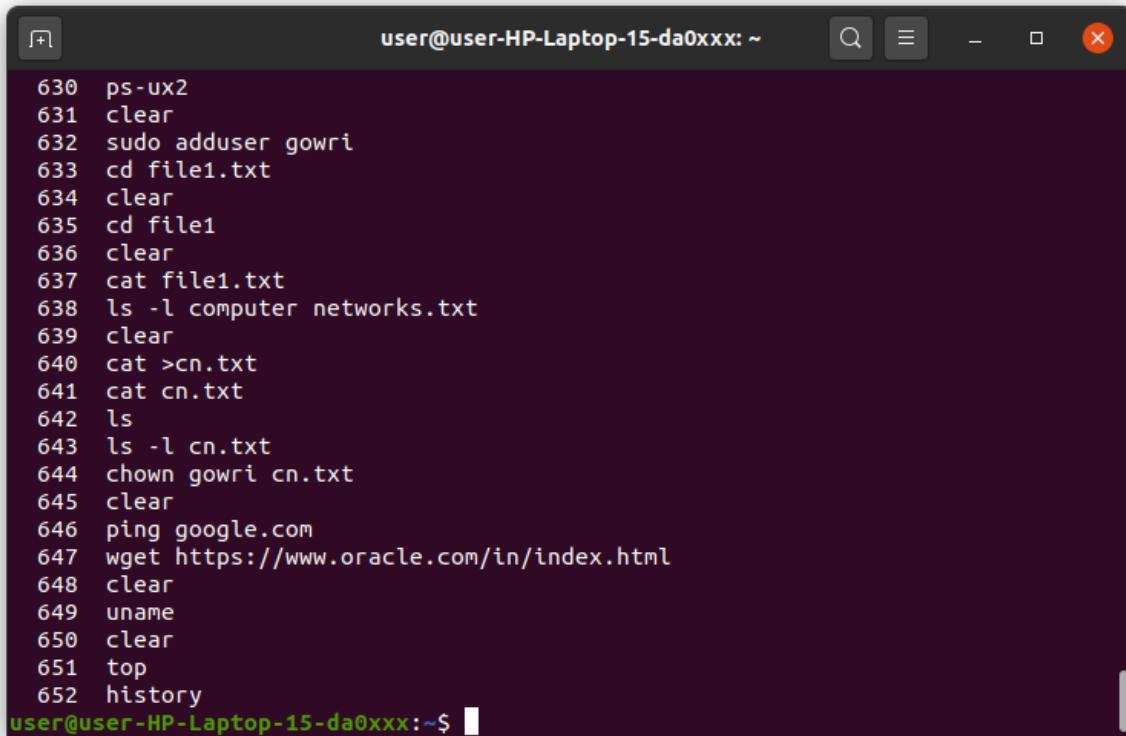
## 29. history

When you've been using Linux for a certain period of time, you'll quickly notice that you can run hundreds of commands every day. As such, running history command is particularly useful if you want to review the commands you've entered before.

```

user@user-HP-Laptop-15-da0xxx:~$ history
1 cd cpgms
2 gcc display.c -o display.out
3 sudo apt install gcc
4 gcc display.c -o display.out
5 sudo apt install gcc
6 sudo apt-get update
7 sudo apt install dovecot
8 sudo apt install gcc
9 cd cpgms
10 gcc display.c -o display.out
11 ./display.out
12 git --v
13 sudo apt-get install git
14 git --version
15 cd linkedlist
16 cd cpgms
17 gcc singlell.c -o singlell.out
18 gcc sing.c -o sing.out
19 cd cpgms
20 gcc insingle.c -o insingle.out
21 gcc singlell.c -o singlell.out
22 ./singlell.out
23 gcc singlell.c -o singlell.out

```



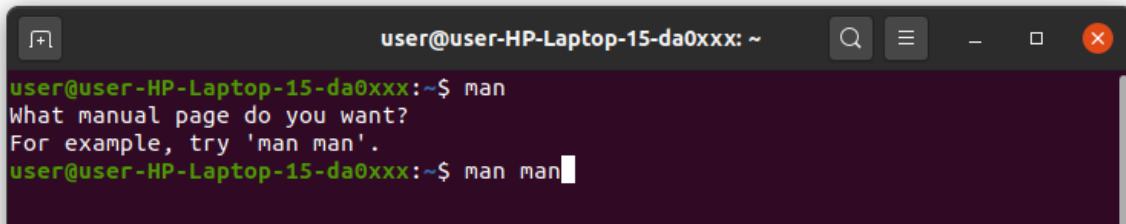
A screenshot of a terminal window titled "user@user-HP-Laptop-15-da0xxx: ~". The window shows a history of commands entered by the user, starting from line 630. The commands include various system utilities like ps, sudo, adduser, cd, clear, cat, ls, ping, wget, and top. The terminal has a dark background with light-colored text and standard window controls at the top.

```
630 ps-ux2
631 clear
632 sudo adduser gowri
633 cd file1.txt
634 clear
635 cd file1
636 clear
637 cat file1.txt
638 ls -l computer networks.txt
639 clear
640 cat >cn.txt
641 cat cn.txt
642 ls
643 ls -l cn.txt
644 chown gowri cn.txt
645 clear
646 ping google.com
647 wget https://www.oracle.com/in/index.html
648 clear
649 uname
650 clear
651 top
652 history
user@user-HP-Laptop-15-da0xxx:~$
```

## 30. man

Confused about the function of certain Linux commands? Don't worry, you can easily learn how to use them right from Linux's shell by using the `man` command. For instance, entering `man tail` will show the manual instruction of the `tail` command.

Use the command: `man man` to start learning about `man` utility.



A screenshot of a terminal window titled "user@user-HP-Laptop-15-da0xxx: ~". The user types `man` and is prompted with "What manual page do you want? For example, try 'man man'." The terminal has a dark background with light-colored text and standard window controls at the top.

```
user@user-HP-Laptop-15-da0xxx:~$ man
What manual page do you want?
For example, try 'man man'.
user@user-HP-Laptop-15-da0xxx:~$ man man
```

The screenshot shows a terminal window with the title bar "user@user-HP-Laptop-15-da0xxx: ~". The window contains the man(1) manual page for the "man" command. The page is divided into sections: NAME, SYNOPSIS, and DESCRIPTION. The SYNOPSIS section lists various command-line options. The DESCRIPTION section provides a detailed explanation of what "man" does and how it finds manual pages. A table below the description maps section numbers to types of pages. The bottom of the window shows a status bar with the message "Manual page man(1) line 1 (press h for help or q to quit)".

```

MAN(1)           Manual pager utils          MAN(1)

NAME
    man - an interface to the system reference manuals

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

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

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

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

    A manual page consists of several sections.

Manual page man(1) line 1 (press h for help or q to quit)

```

## 31. echo

This command is used to move some data into a file. For example, if you want to add the text, “Hello, my name is John” into a file called name.txt, you would type echo Hello, my name is John >> name.txt

The screenshot shows a terminal window with the title bar "user@user-HP-Laptop-15-da0xxx: ~". The window displays the command "echo My name is GangaKrishnanG" being run at the prompt. The output "My name is GangaKrishnanG" is shown in green text, indicating it was printed to the terminal.

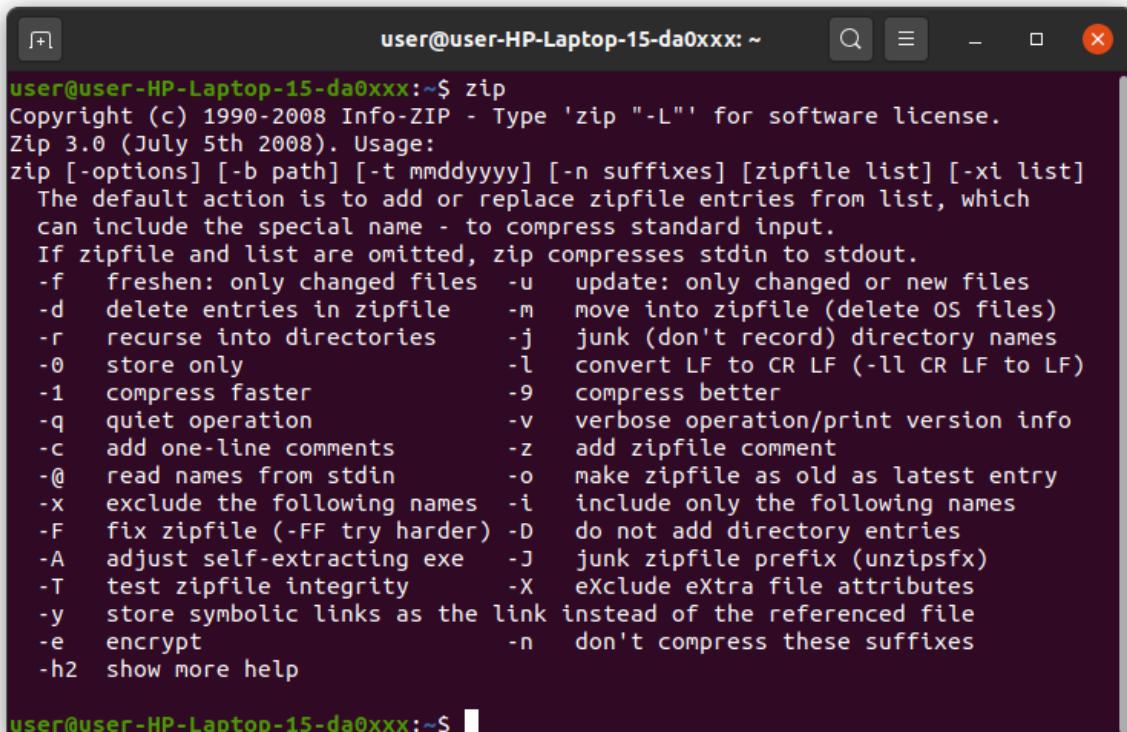
```

user@user-HP-Laptop-15-da0xxx:~$ echo My name is GangaKrishnanG
My name is GangaKrishnanG
user@user-HP-Laptop-15-da0xxx:~$ 

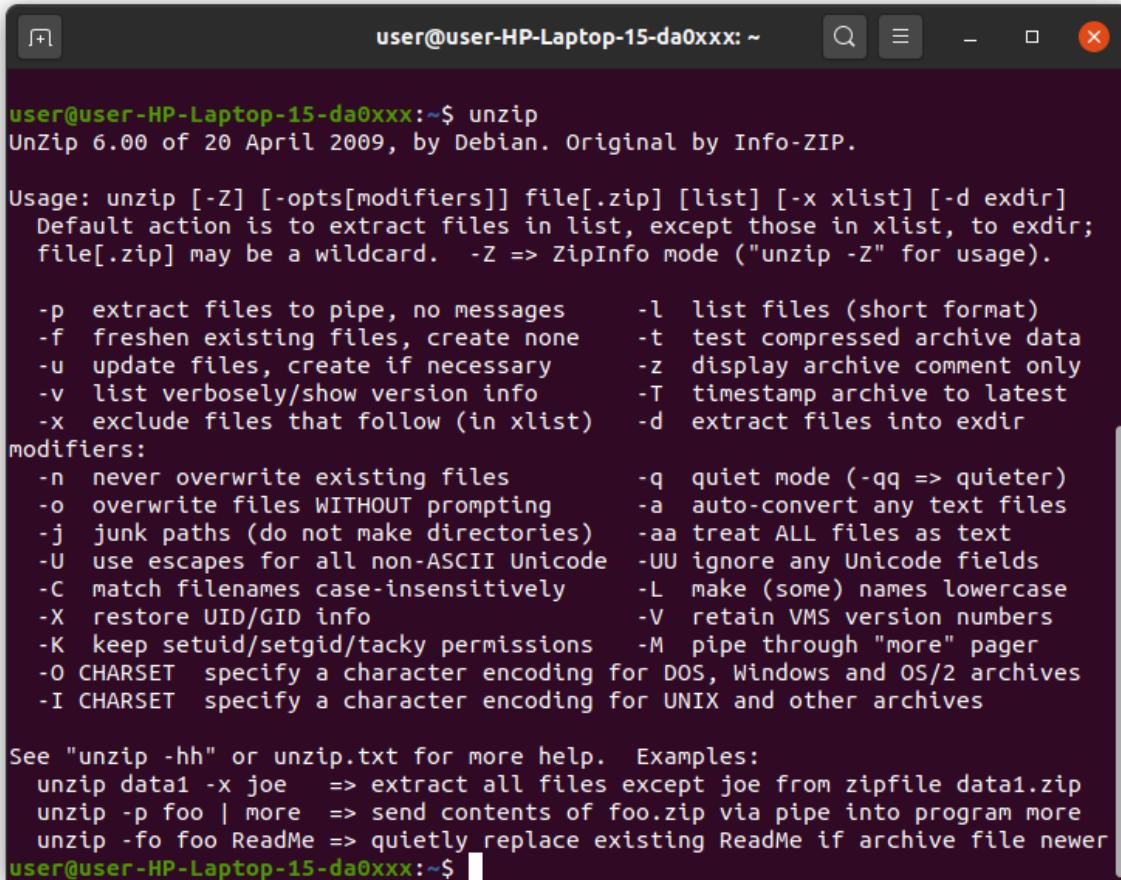
```

## 32. zip, unzip

Use the zip command to compress your files into a zip archive, and use the unzip command to extract the zipped files from a zip archive. ( This program should be installed , some distributions may not have them. You can also look at gzip and bzip commands)



```
user@user-HP-Laptop-15-da0xxx:~$ zip
Copyright (c) 1990-2008 Info-ZIP - Type 'zip "-L"' for software license.
Zip 3.0 (July 5th 2008). Usage:
zip [-options] [-b path] [-t mmddyyyy] [-n suffixes] [zipfile list] [-xi list]
  The default action is to add or replace zipfile entries from list, which
  can include the special name - to compress standard input.
  If zipfile and list are omitted, zip compresses stdin to stdout.
-f   freshen: only changed files -u   update: only changed or new files
-d   delete entries in zipfile -m   move into zipfile (delete OS files)
-r   recurse into directories -j   junk (don't record) directory names
-0   store only             -l   convert LF to CR LF (-ll CR LF to LF)
-1   compress faster        -9   compress better
-q   quiet operation         -v   verbose operation/print version info
-c   add one-line comments  -z   add zipfile comment
-@   read names from stdin -o   make zipfile as old as latest entry
-x   exclude the following names -i   include only the following names
-F   fix zipfile (-FF try harder) -D   do not add directory entries
-A   adjust self-extracting exe -J   junk zipfile prefix (unzipsfx)
-T   test zipfile integrity  -X   eXclude eXtra file attributes
-y   store symbolic links as the link instead of the referenced file
-e   encrypt                 -n   don't compress these suffixes
-h2  show more help
```



```

user@user-HP-Laptop-15-da0xxx:~$ unzip
UnZip 6.00 of 20 April 2009, by Debian. Original by Info-ZIP.

Usage: unzip [-Z] [-opts[modifiers]] file[.zip] [list] [-x xlist] [-d exdir]
  Default action is to extract files in list, except those in xlist, to exdir;
  file[.zip] may be a wildcard. -Z => ZipInfo mode ("unzip -Z" for usage).

-p  extract files to pipe, no messages      -l  list files (short format)
-f  freshen existing files, create none    -t  test compressed archive data
-u  update files, create if necessary     -z  display archive comment only
-v  list verbosely/show version info      -T  timestamp archive to latest
-x  exclude files that follow (in xlist)   -d  extract files into exdir

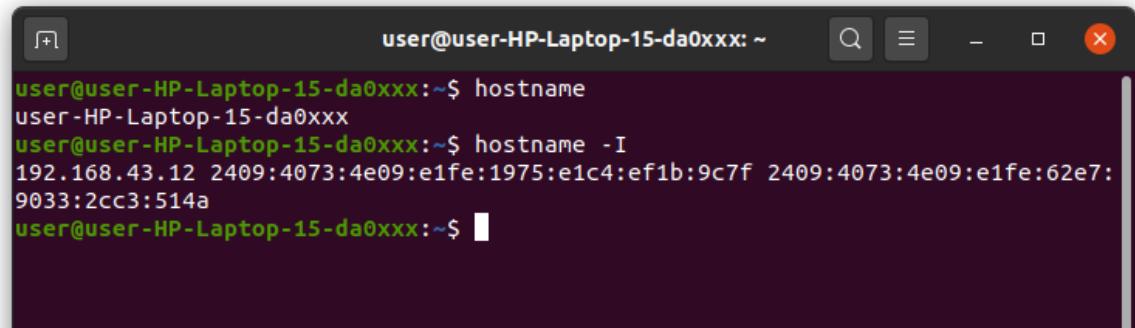
modifiers:
-n  never overwrite existing files        -q  quiet mode (-qq => quieter)
-o  overwrite files WITHOUT prompting     -a  auto-convert any text files
-j  junk paths (do not make directories)  -aa treat ALL files as text
-U  use escapes for all non-ASCII Unicode -UU ignore any Unicode fields
-C  match filenames case-insensitively   -L  make (some) names lowercase
-X  restore UID/GID info                 -V  retain VMS version numbers
-K  keep setuid/setgid/tacky permissions  -M  pipe through "more" pager
-O CHARSET specify a character encoding for DOS, Windows and OS/2 archives
-I CHARSET specify a character encoding for UNIX and other archives

See "unzip -hh" or unzip.txt for more help. Examples:
unzip data1 -x joe    => extract all files except joe from zipfile data1.zip
unzip -p foo | more   => send contents of foo.zip via pipe into program more
unzip -fo foo ReadMe => quietly replace existing ReadMe if archive file newer
user@user-HP-Laptop-15-da0xxx:~$ 

```

### 33. hostname

If you want to know the name of your host/network simply type hostname. Adding a -I to the end will display the IP address of your network.



```

user@user-HP-Laptop-15-da0xxx:~$ hostname
user-HP-Laptop-15-da0xxx
user@user-HP-Laptop-15-da0xxx:~$ hostname -I
192.168.43.12 2409:4073:4e09:e1fe:1975:e1c4:ef1b:9c7f 2409:4073:4e09:e1fe:62e7:
9033:2cc3:514a
user@user-HP-Laptop-15-da0xxx:~$ 

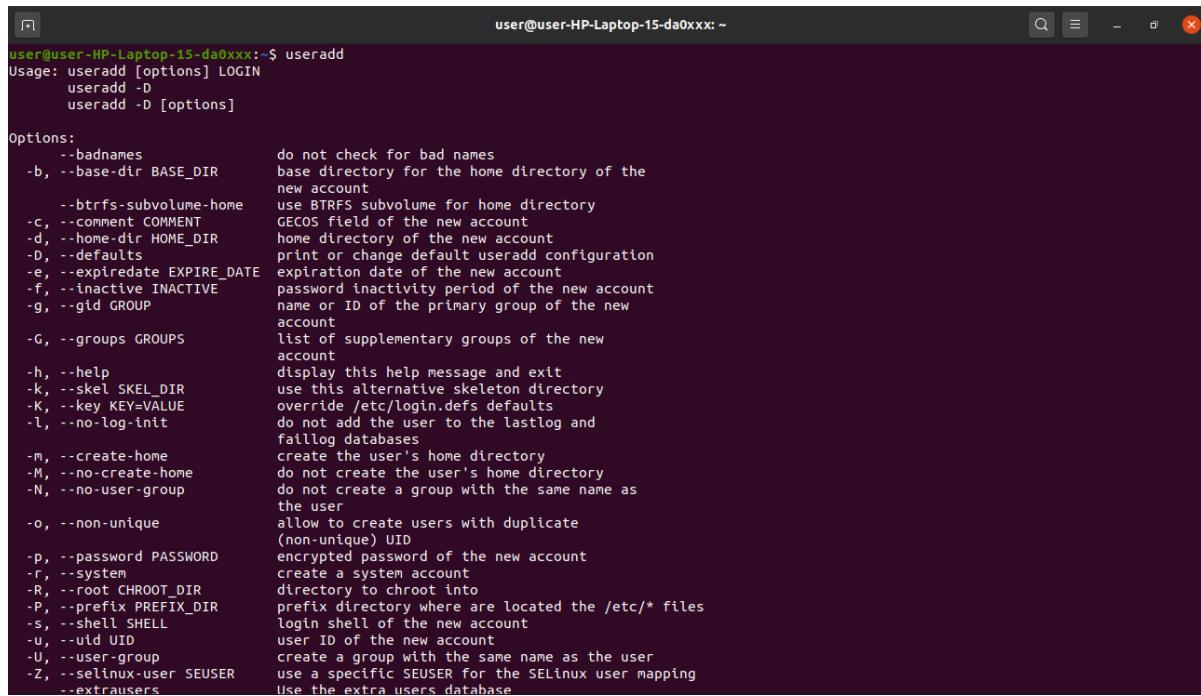
```

### 34. useradd, userdel

( This is available only to system admins) Since Linux is a multi-user system, this means more than one person can interact with the same system at the same time. useradd is used to create a new user, while passwd is adding a password to that user's account.

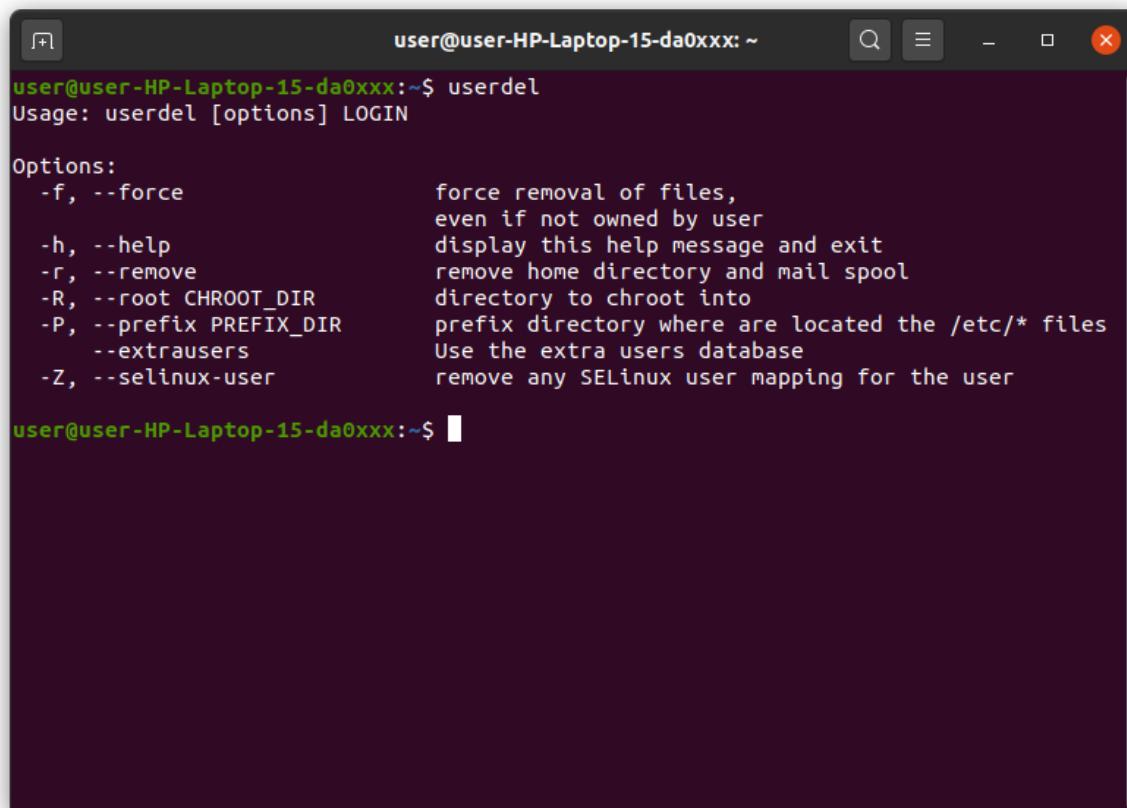
To add a new person named John type, useradd John and then to add his password

type, passwd 123456789.



```
user@user-HP-Laptop-15-da0xxx:~$ useradd
Usage: useradd [options] LOGIN
      useradd -D
      useradd -D [options]

Options:
  --badnames          do not check for bad names
  -b, --base-dir BASE_DIR    base directory for the home directory of the
                             new account
  --btrfs-subvolume-home   use BTRFS subvolume for home directory
  -c, --comment COMMENT    GECOS field of the new account
  -d, --home-dir HOME_DIR   home directory of the new account
  -D, --defaults          print or change default useradd configuration
  -e, --expiredate EXPIRE_DATE  expiration date of the new account
  -f, --inactive INACTIVE    password inactivity period of the new account
  -g, --gld GROUP          name or ID of the primary group of the new
                           account
  -G, --groups GROUPS      list of supplementary groups of the new
                           account
  -h, --help              display this help message and exit
  -k, --skel SKEL_DIR      use this alternative skeleton directory
  -K, --key KEY=VALUE      override /etc/login.defs defaults
  -l, --no-log-init       do not add the user to the lastlog and
                           faillog databases
  -m, --create-home       create the user's home directory
  -M, --no-create-home    do not create the user's home directory
  -N, --no-user-group     do not create a group with the same name as
                           the user
  -o, --non-unique        allow to create users with duplicate
                           (non-unique) UID
  -p, --password PASSWORD  encrypted password of the new account
  -r, --system             create a system account
  -R, --root CHROOT_DIR    directory to chroot into
  -P, --prefix PREFIX_DIR   prefix directory where are located the /etc/* files
  -s, --shell SHELL         login shell of the new account
  -U, --uid UID            user ID of the new account
  -U, --user-group         create a group with the same name as the user
  -Z, --selinux-user SEUSER  use a specific SEUSER for the SELinux user mapping
  --extrausers             Use the extra users database
```



```
user@user-HP-Laptop-15-da0xxx:~$ userdel
Usage: userdel [options] LOGIN

Options:
  -f, --force               force removal of files,
                             even if not owned by user
  -h, --help                display this help message and exit
  -r, --remove               remove home directory and mail spool
                             directory to chroot into
  -R, --root CHROOT_DIR     prefix directory where are located the /etc/* files
  -P, --prefix PREFIX_DIR    Use the extra users database
  --extrausers               remove any SELinux user mapping for the user

user@user-HP-Laptop-15-da0xxx:~$
```

## **EXPERIMENT-10**

### **FAMILIARISATION WITH VI EDITOR**

The VI editor is the most popular and classic text editor in the Linux family. Below, are some reasons which make it a widely used editor –

- 1) It is available in almost all Linux Distributions
- 2) It works the same across different platforms and Distributions
- 3) It is user-friendly. Hence, millions of Linux use it for their editing needs

Nowadays, there are advanced versions of the vi editor available, and the most popular one is **VIM** which is **Vi Improved**. Some of the other ones are Elvis, Nvi, Nano, and Vile.

#### **1. VI OPERATIONAL MODES**

- Vi Command Mode:
  - The vi editor opens in this mode, and it only understands commands.
  - In this mode, you can perform administrative tasks such as move the cursor and cut, copy, paste the text.
  - This mode also saves the changes you have made to the file.
  - Commands are case sensitive. The right letter case should be used.
- Vi Insert Mode:
  - This mode is for inserting text in the file.
  - You can switch to the Insert mode from the command mode **by pressing 'i' on the keyboard**
  - Once you are in Insert mode, any key would be taken as an input for the file on which you are currently working.
  - To return to the command mode and save the changes you have made you need to press the Esc key

## **2. STARTING VI**

To open a file with vi, type: vi ‘filename’.txt

- a. If the file does not exist, a screen will appear with just a cursor at the top followed by tildes (~) in the first column.
- b. If the file does exist, the first few lines of the file will appear.
- c. The status line at the bottom of the screen shows error messages and provides information and feedback, including the name of the file.

## **3. Vi EDITING COMMANDS**

- i - Insert at cursor (goes into insert mode)
- a - Write after cursor (goes into insert mode)
- A - Write at the end of line (goes into insert mode)
- ESC - Terminate insert mode
- u - Undo last change
- U - Undo all changes to the entire line
- o - Open a new line (goes into insert mode)
- dd - Delete line
- 3dd - Delete 3 lines.
- D - Delete contents of line after the cursor
- C - Delete contents of a line after the cursor and insert new text. Press ESC key to end insertion.
- dw - Delete word
- 4dw - Delete 4 words
- cw - Change word
- x - Delete character at the cursor
- r - Replace character
- R - Overwrite characters from cursor onward
- s - Substitute one character under cursor continue to insert
- S - Substitute entire line and begin to insert at the beginning of the line
- ~ - Change case of individual character

**Note:** You should be in the "**command mode**" to execute these commands. VI editor is **case-sensitive** so make sure you type the commands in the right letter-case.

Make sure you press the right command otherwise you will end up making undesirable changes to the file. You can also enter the insert mode by pressing a, A, o, as required.

#### Moving within a file :

- ★ k - Move cursor up
- ★ j - Move cursor down
- ★ h - Move cursor left
- ★ l - Move cursor right

You need to be in the command mode to move within a file. The default keys for navigation are mentioned below else; You can **also use the arrow keys on the keyboard**.

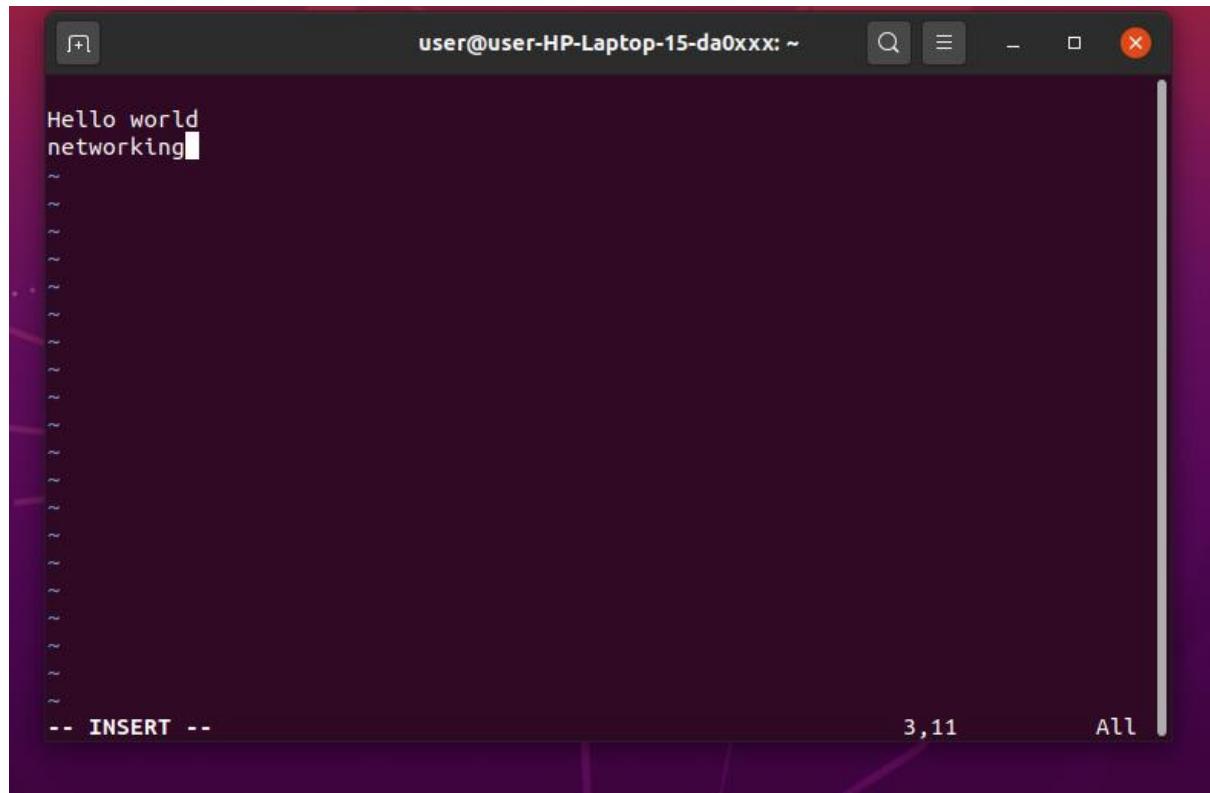
#### Saving and Closing the file

- ★ Shift+zz - Save the file and quit
- ★ :w - Save the file but keep it open
- ★ :q - Quit without saving
- ★ :wq - Save the file and quit

You should be in the **command mode to exit the editor and save changes** to the file.

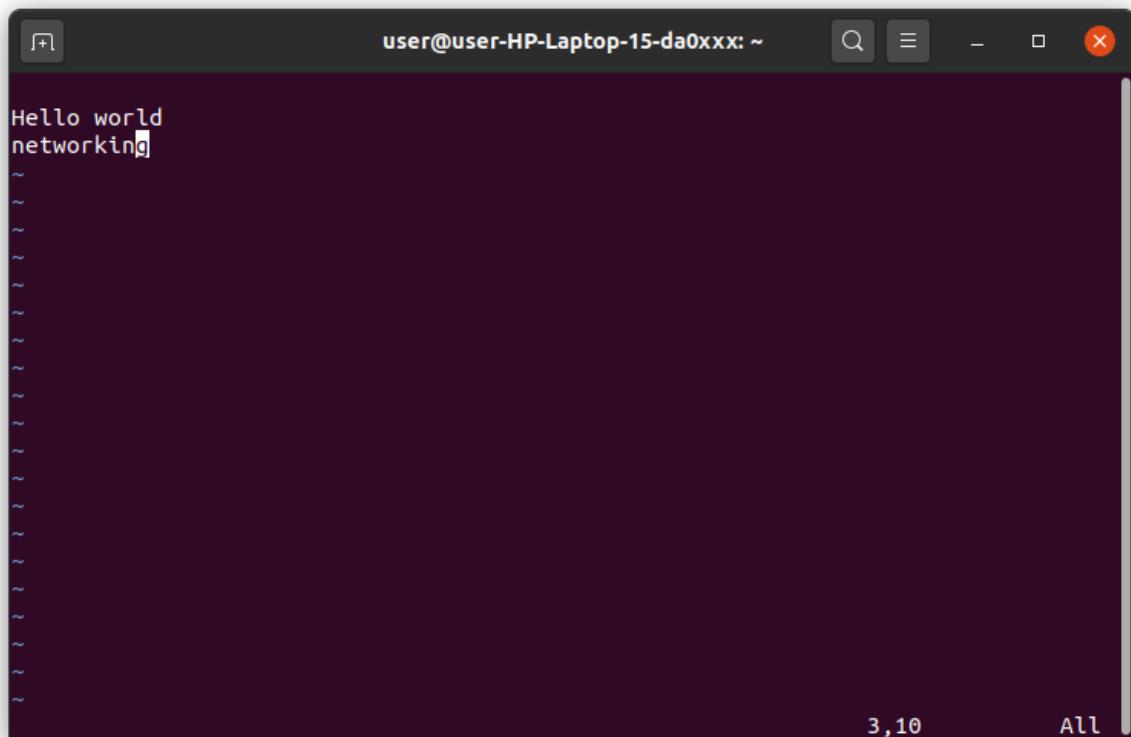
## RESULT

- **'i' command**



```
Hello world
networking
```

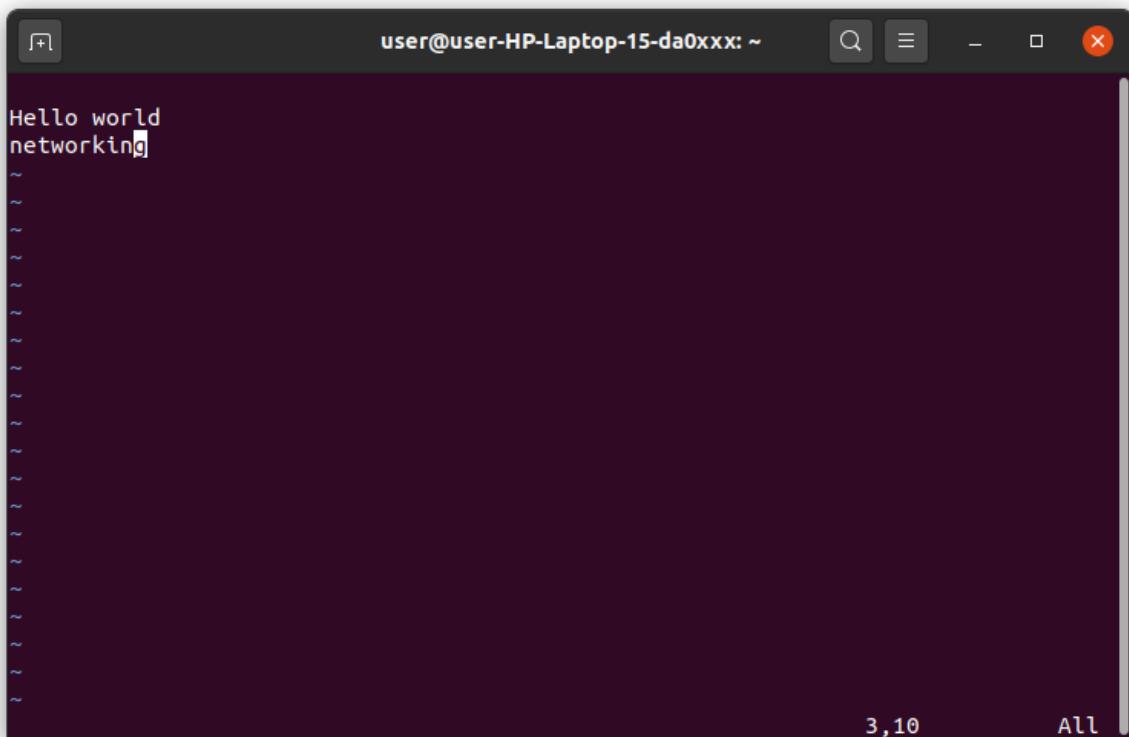
- **'k' command**: moves the cursor up one time.



```
Hello world
networking
```

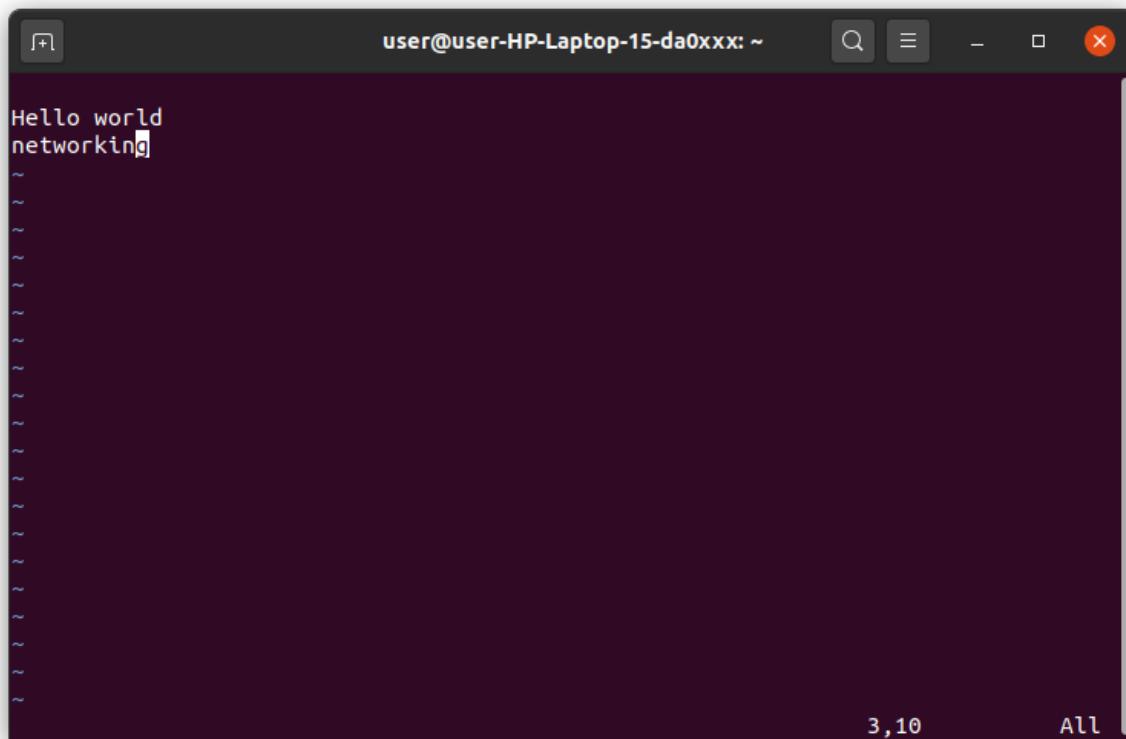
- ‘j’ command:moves the cursor down one line.

```
+ user@user-HP-Laptop-15-da0xxx: ~
Hello worl
networking
~
```

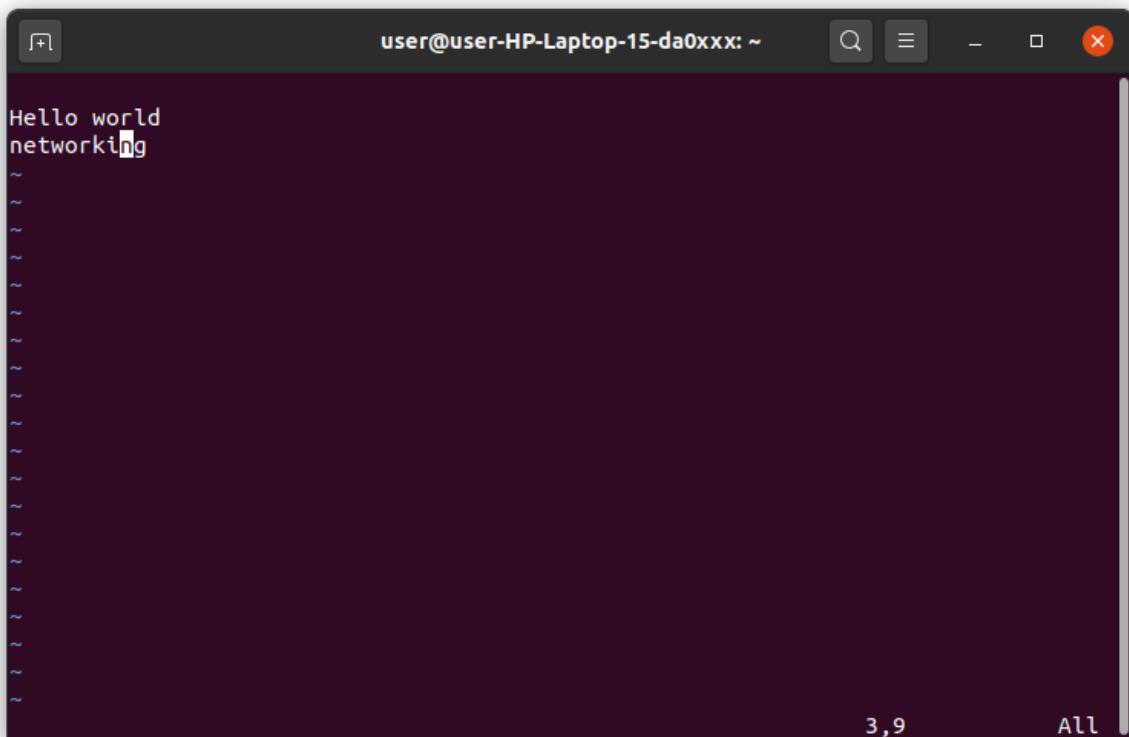


```
Hello world
networking
```

- **'h' command**: moves the cursor to the left one character position.

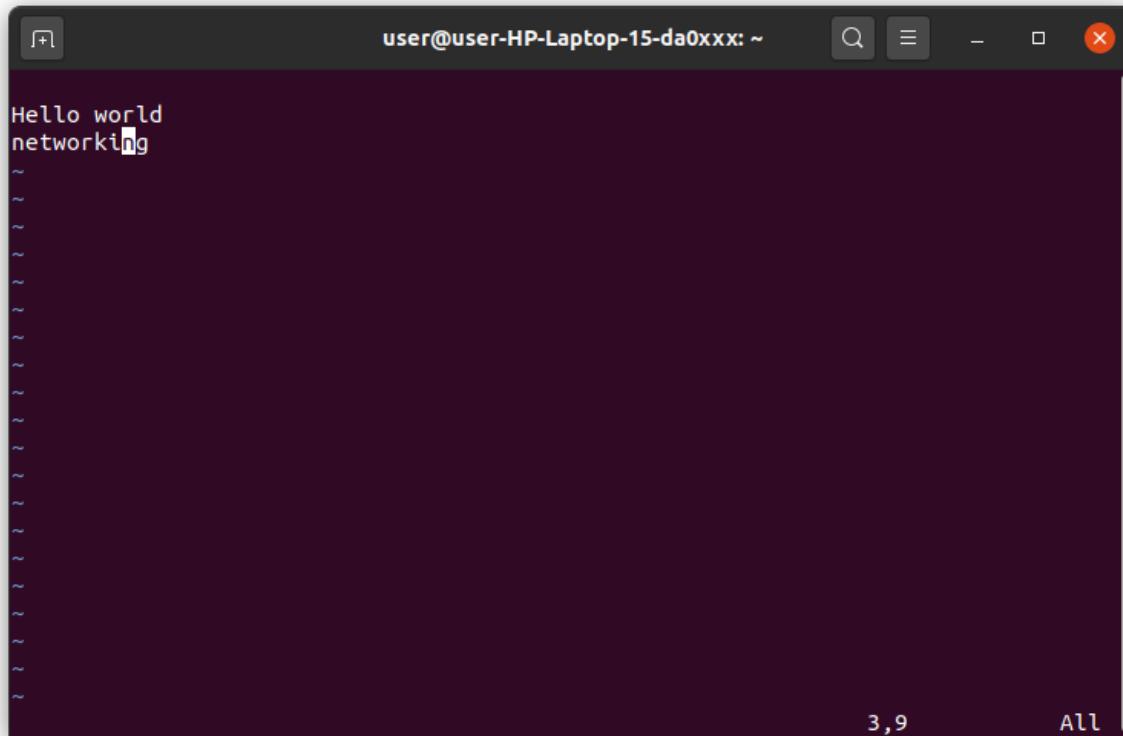


```
Hello world
networking
```

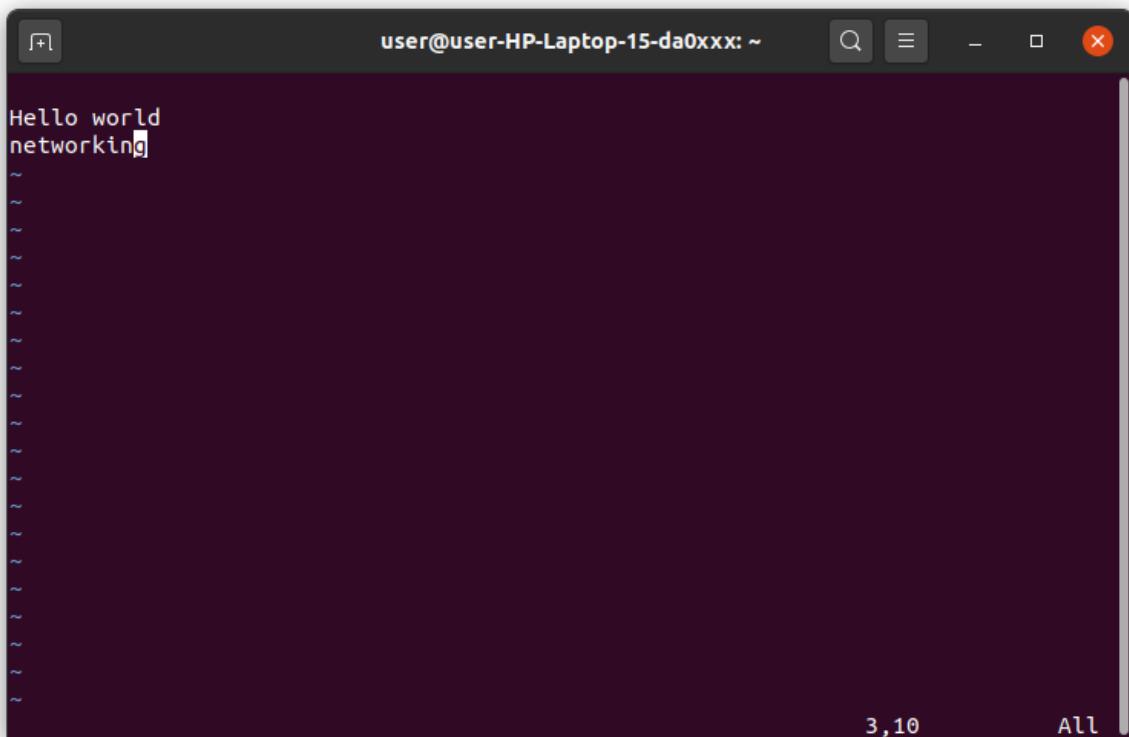


```
Hello world  
networking
```

- ‘l’**command**:moves the cursor to the right one character position.



```
Hello world  
networking
```

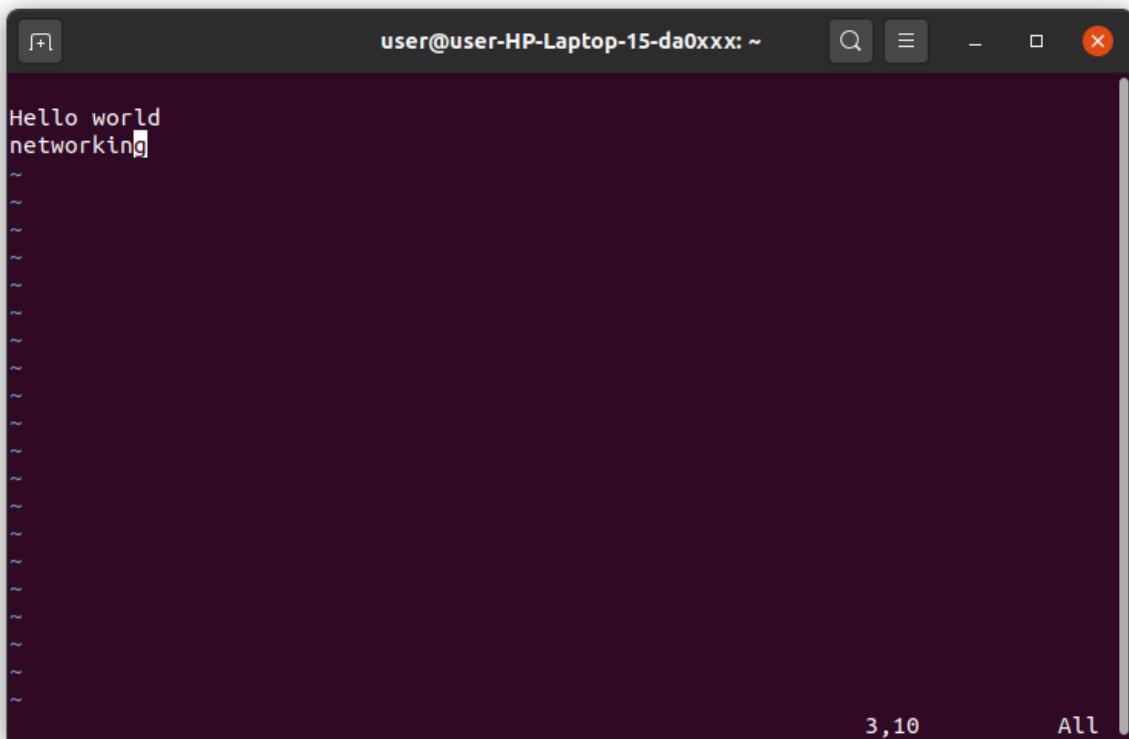


A screenshot of a terminal window titled "user@user-HP-Laptop-15-da0xxx: ~". The window has a dark background and a light-colored text area. In the text area, the following lines are visible:

```
Hello world
networking
```

The word "networking" is followed by a cursor. The status bar at the bottom right shows "3,10" and "All".

- **I** command: Insert text at the beginning of the current line.



A screenshot of a terminal window titled "user@user-HP-Laptop-15-da0xxx: ~". The window has a dark background and a light-colored text area. In the text area, the following lines are visible:

```
Hello world
networking
```

The word "networking" is followed by a cursor. The status bar at the bottom right shows "3,10" and "All".

A screenshot of a terminal window titled "user@user-HP-Laptop-15-da0xxx: ~". The window has a dark background and light-colored text. At the top, there are standard window controls (minimize, maximize, close) and a search bar. The main area of the terminal shows the text "Hello world" followed by several blank lines (~). In the bottom right corner, the status bar displays "3,1" and "All". The bottom of the terminal window has a footer bar with the text "-- INSERT --".

A screenshot of a terminal window titled "user@user-HP-Laptop-15-da0xxx: ~". The window has a dark background and light-colored text. The text "Hello world" and "Computer networking" are visible. The cursor is positioned between "Computer" and "networking". The status bar in the bottom right corner shows "3,10" and "All". The bottom of the terminal window has a footer bar with the text "-- INSERT --".

- **a** command: Insert text after the current cursor location.

A screenshot of a terminal window titled "user@user-HP-Laptop-15-da0xxx: ~". The window has a dark background and light-colored text. At the top, it shows the command "Hello world" followed by "Computer networking". Below this, there are approximately 20 blank lines. At the bottom left, it says "-- INSERT --". On the right side, there are status indicators: "3,19" and "All".

A screenshot of a terminal window titled "user@user-HP-Laptop-15-da0xxx: ~". The window has a dark background and light-colored text. It continues from the previous screenshot, showing the same "Hello world" and "Computer networking" lines, followed by another set of approximately 20 blank lines. At the bottom left, it says "-- INSERT --". On the right side, there are status indicators: "3,20" and "All".

- **'A' command**: Insert text at the end of the current line.

A screenshot of a terminal window titled "user@user-HP-Laptop-15-da0xxx: ~". The window shows a dark-themed vi editor interface. The text area contains the following content:

```
Hello world
Computer networking lab
```

The cursor is positioned at the end of the second line. The status bar at the bottom left shows "-- INSERT --". The status bar at the bottom right shows "3,24" and "All".

A screenshot of a terminal window titled "user@user-HP-Laptop-15-da0xxx: ~". The window shows a dark-themed vi editor interface. The text area contains the following content:

```
Hello world
Computer networking lab CN
```

A new line "CN" has been added after the word "lab". The cursor is positioned at the end of the second line. The status bar at the bottom left shows "-- INSERT --". The status bar at the bottom right shows "3,27" and "All".

- **'o' command:**Creates a new line for text entry below the cursor location.

Hello world  
Computer networking lab CN

-- INSERT -- 3,27 All

A screenshot of a terminal window titled "user@user-HP-Laptop-15-da0xxx: ~". The window has a dark background with light-colored text. It displays the message "Hello world" followed by "Computer networking lab CN". Below this, there are several tilde (~) characters. At the bottom left, it says "-- INSERT --". On the right side, there are status indicators: "3,27" and "All". The window has standard Linux-style window controls at the top.

Activities Terminal Jun 7 2008 codebind@codebind: ~

TKK College Of Engineering  
Kartikode

-- INSERT -- 2,9 All

A screenshot of a terminal window titled "codebind@codebind: ~". The window has a dark background with light-colored text. It shows a command history starting with "TKK College Of Engineering" and "Kartikode". At the bottom left, it says "-- INSERT --". On the right side, there are status indicators: "2,9" and "All". The window has standard Linux-style window controls at the top. To the left of the terminal, there is a vertical dock with icons for various applications like a browser, file manager, terminal, and system settings.

A screenshot of a terminal window titled "user@user-HP-Laptop-15-da0xxx: ~". The window shows a dark-themed vi editor interface. The text area contains the following content:

```
Hello world
Computer networking lab CN
[REDACTED LINE]
~
```

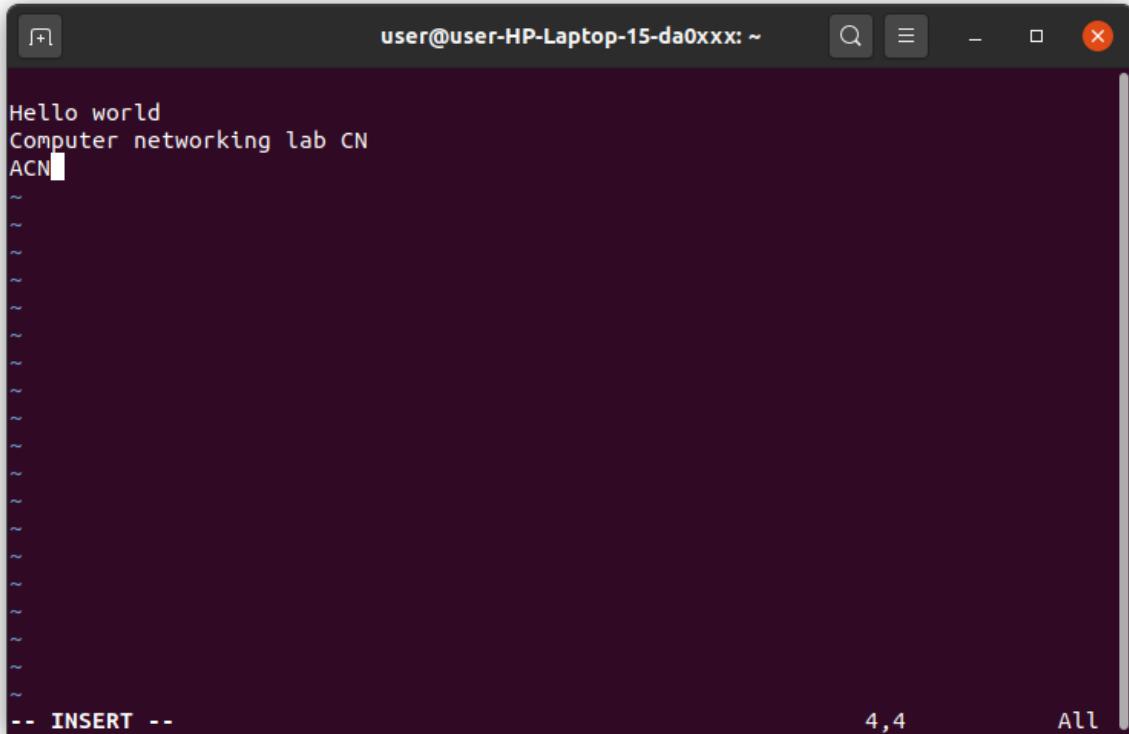
The cursor is positioned at the start of the line "[REDACTED LINE]". The status bar at the bottom left shows "-- INSERT --". The status bar at the bottom right shows "4,1" and "All".

A screenshot of a terminal window titled "user@user-HP-Laptop-15-da0xxx: ~". The window shows a dark-themed vi editor interface. The text area contains the following content:

```
Hello world
Computer networking lab CN
ACN[REDACTED LINE]
~
```

The cursor is positioned at the start of the line "[REDACTED LINE]". The status bar at the bottom left shows "-- INSERT --". The status bar at the bottom right shows "4,4" and "All".

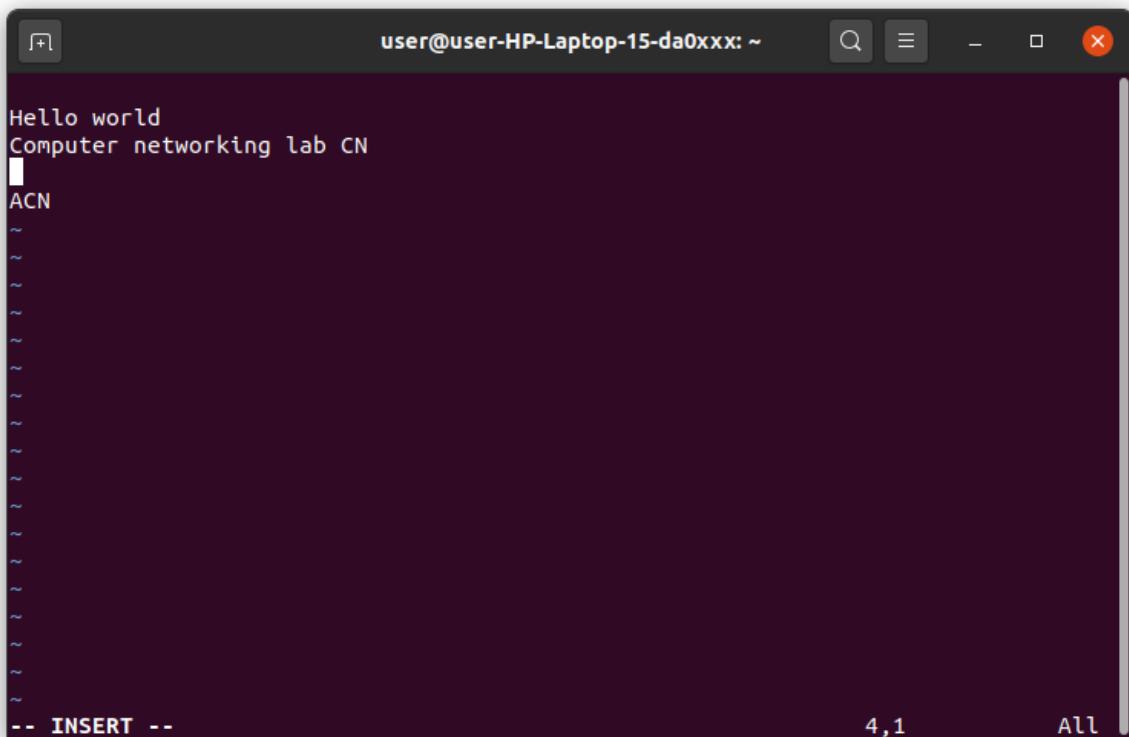
- **O'command:**Creates a new line for text entry above the cursor location.



A screenshot of a terminal window titled "user@user-HP-Laptop-15-da0xxx: ~". The window contains a dark-themed code editor. At the top, there is a status bar with icons for search, refresh, and close. The main area shows the following text:

```
Hello world
Computer networking lab CN
ACN
```

The word "ACN" is highlighted with a cursor. Below the text, there is a status bar with the message "-- INSERT --" and coordinates "4,4" and "All".



A screenshot of a terminal window titled "user@user-HP-Laptop-15-da0xxx: ~". The window contains a dark-themed code editor. At the top, there is a status bar with icons for search, refresh, and close. The main area shows the following text:

```
Hello world
Computer networking lab CN
ACN
```

The word "ACN" is highlighted with a cursor. Below the text, there is a status bar with the message "-- INSERT --" and coordinates "4,1" and "All".

```
Hello world
Computer networking lab CN
programming
ACN
~
```

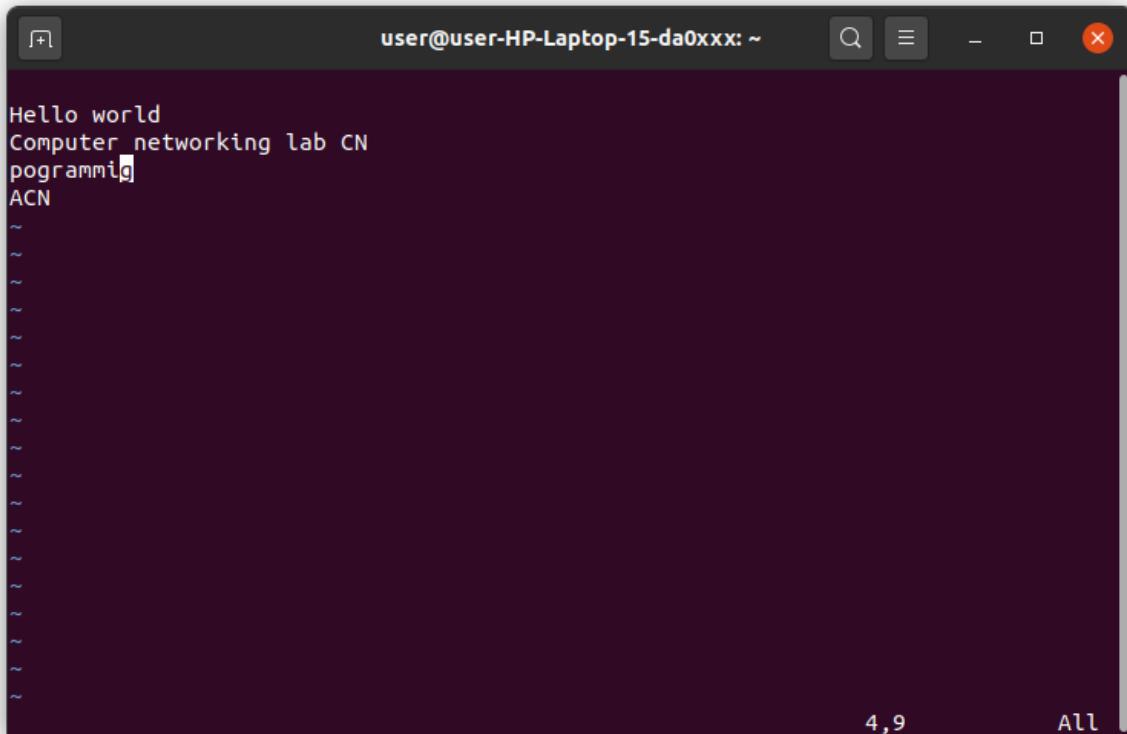
- **x command**: Deletes the character under the cursor location.

```
Hello world
Computer networking lab CN
programmin
ACN
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
4,9          All
```

- **X command:** Delete the character before the cursor location.

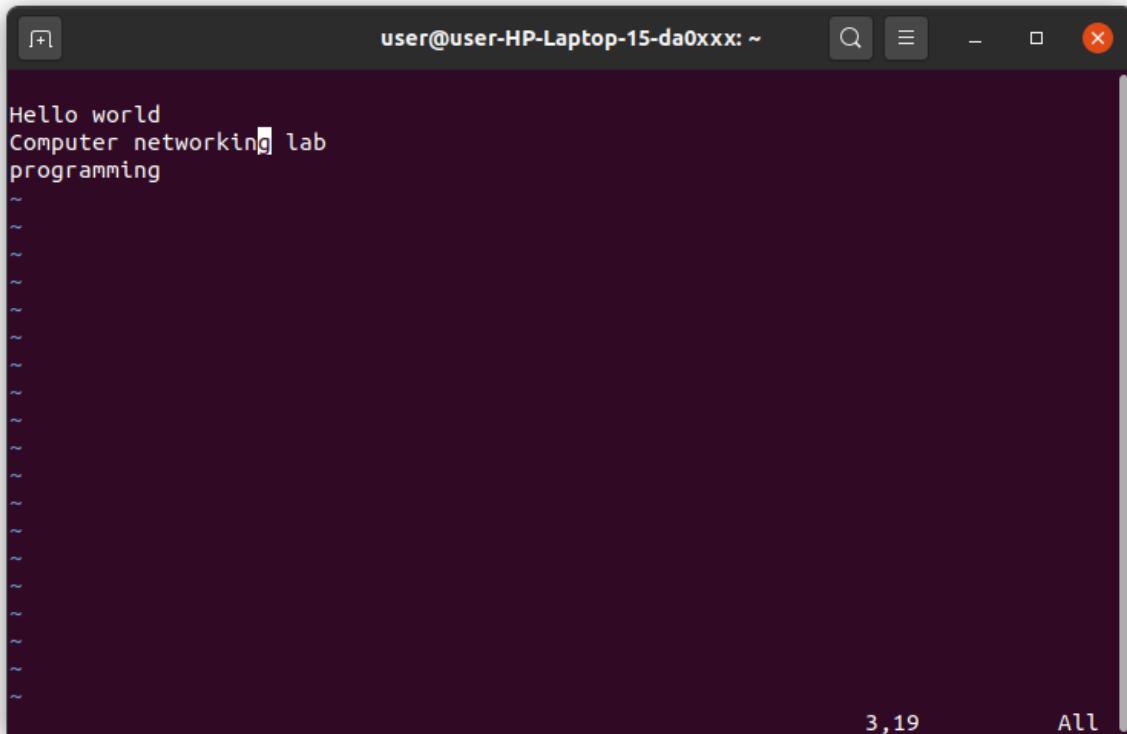
```
Hello world
Computer networking lab CN
programming
ACN
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
4,10          All
```

-- INSERT --

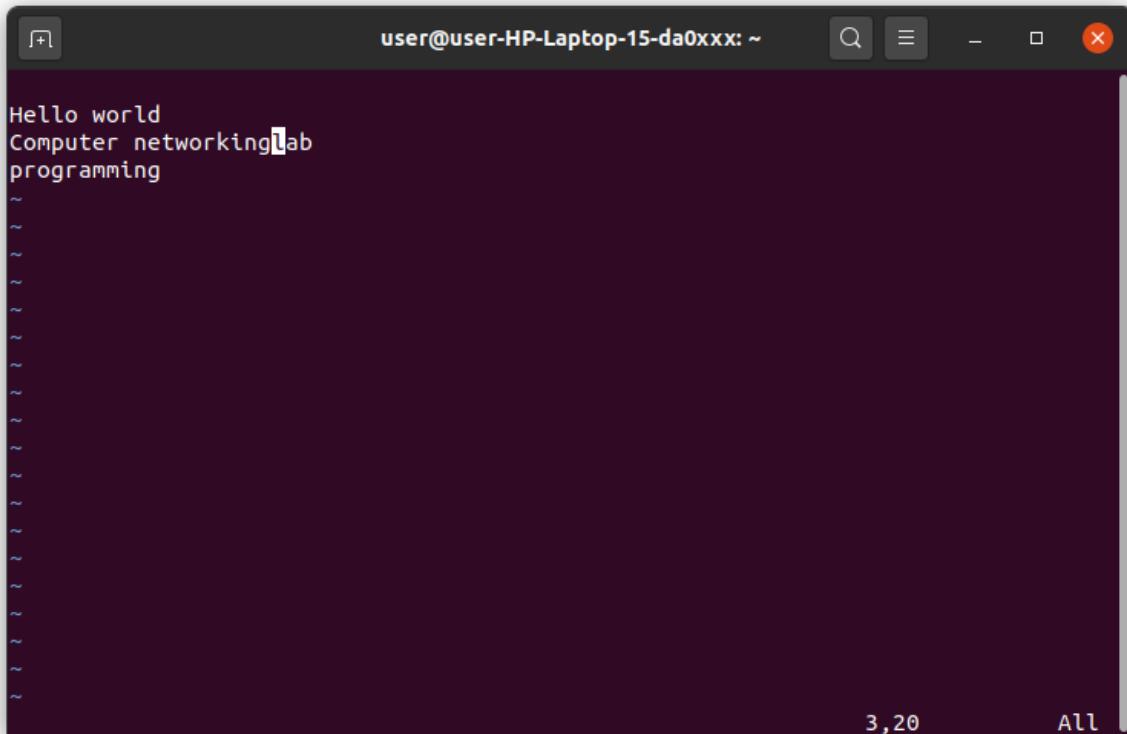


```
Hello world
Computer networking lab CN
programmid
ACN
~
```

- **‘dw’ command:** Deletes from the current cursor location to the next word.

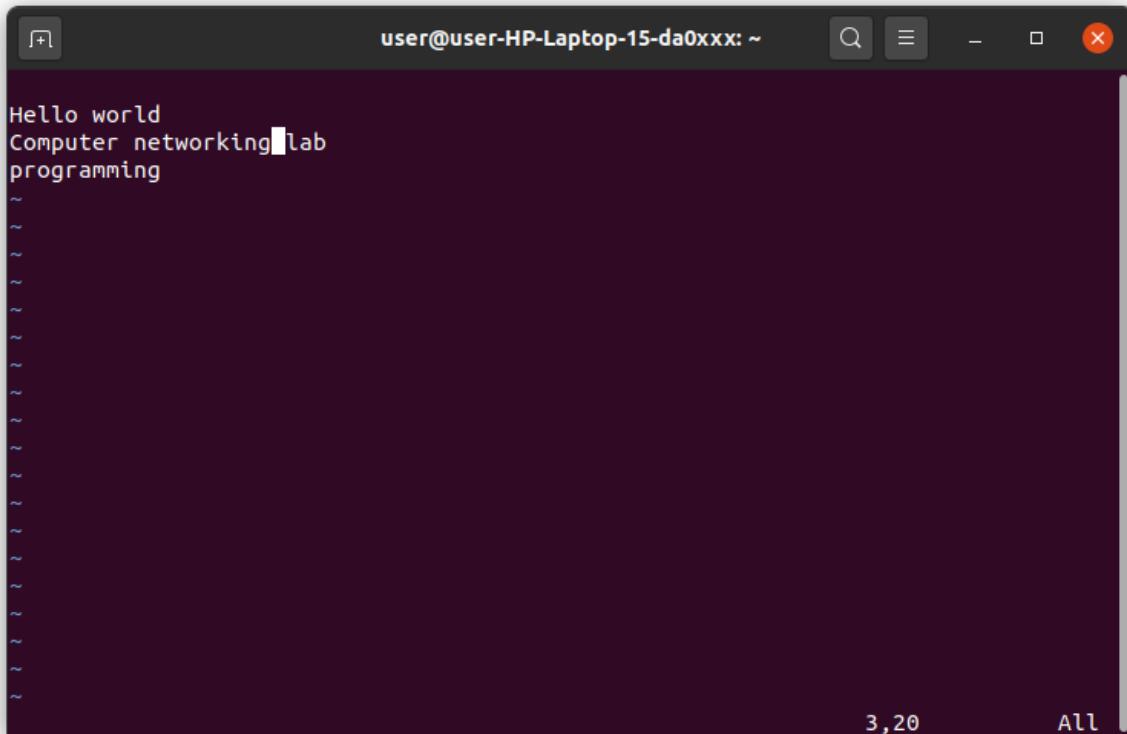


```
Hello world
Computer networkinglab lab
programming
~
```

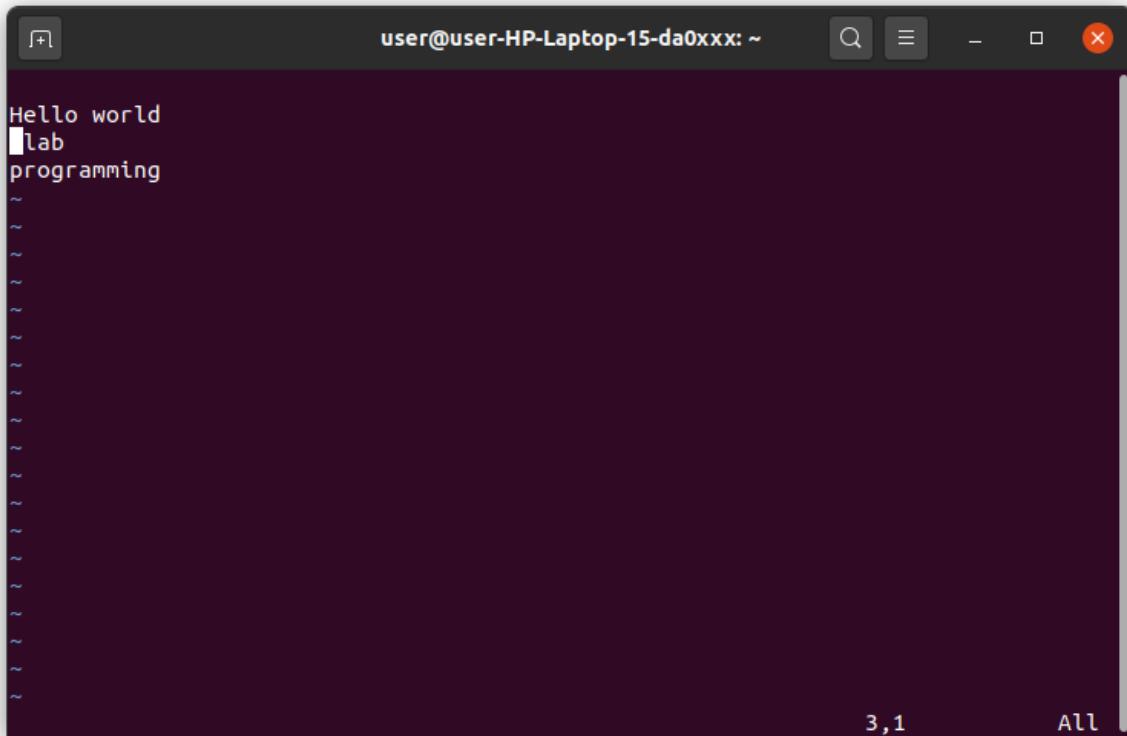


```
Hello world
Computer networkinglab
programming
~
```

- **d^** command: Deletes from the current cursor position to the beginning of the line.

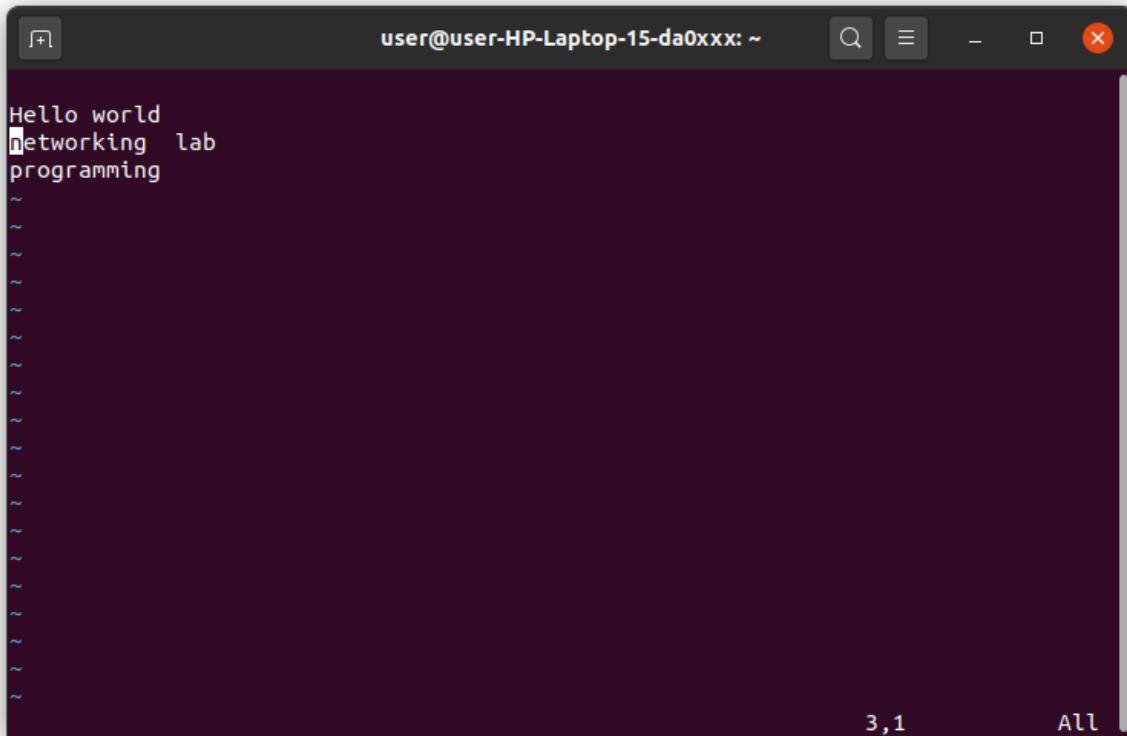


```
Hello world
Computer networkinglab
programming
~
```

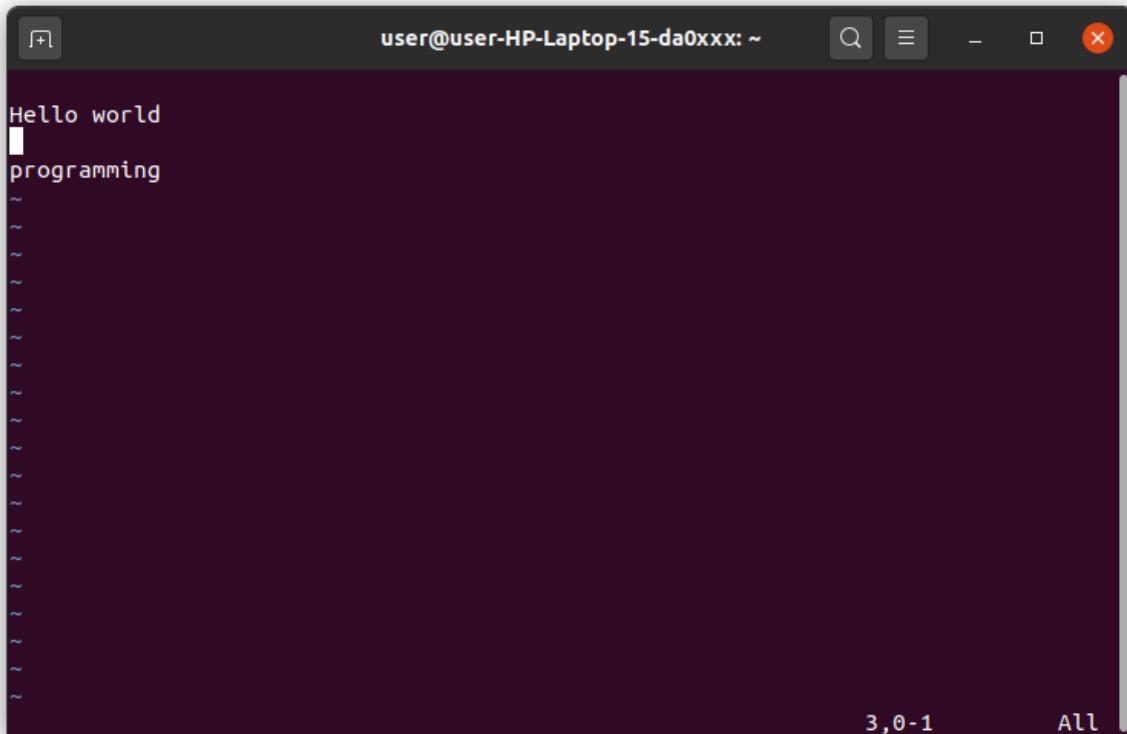


```
Hello world
lab
programming
~
```

- **d\$** command: Deletes from the current cursor position to the end of the line.

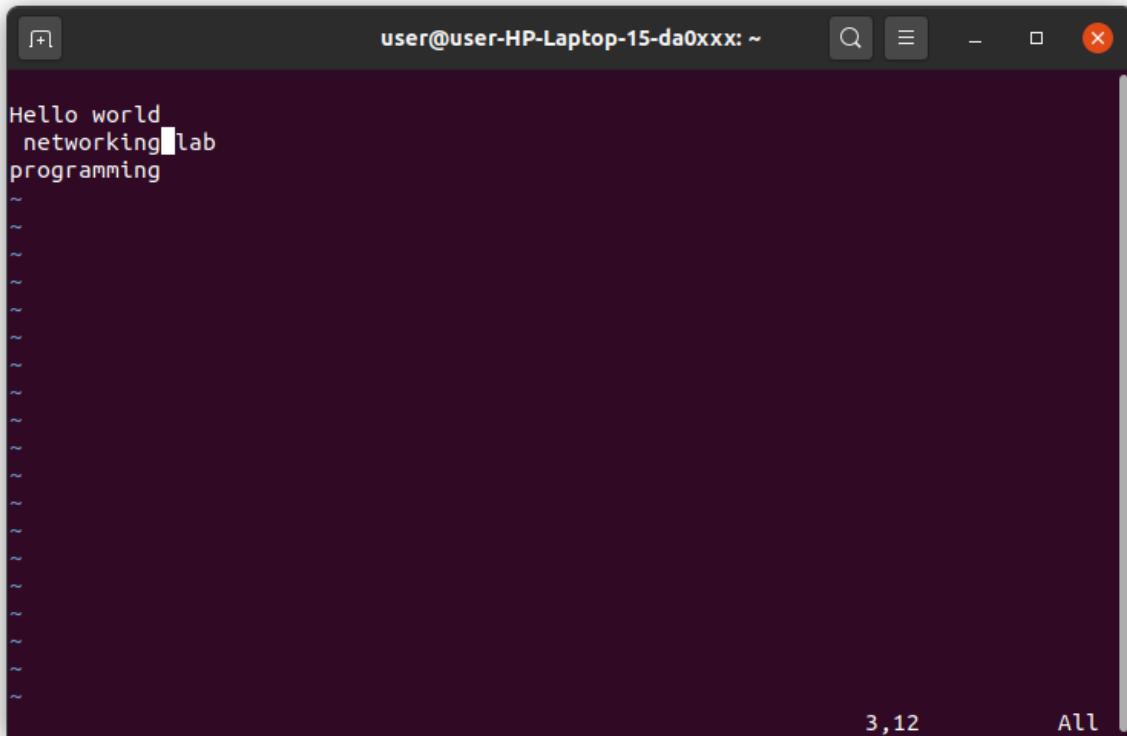


```
Hello world
Networking lab
programming
~
```

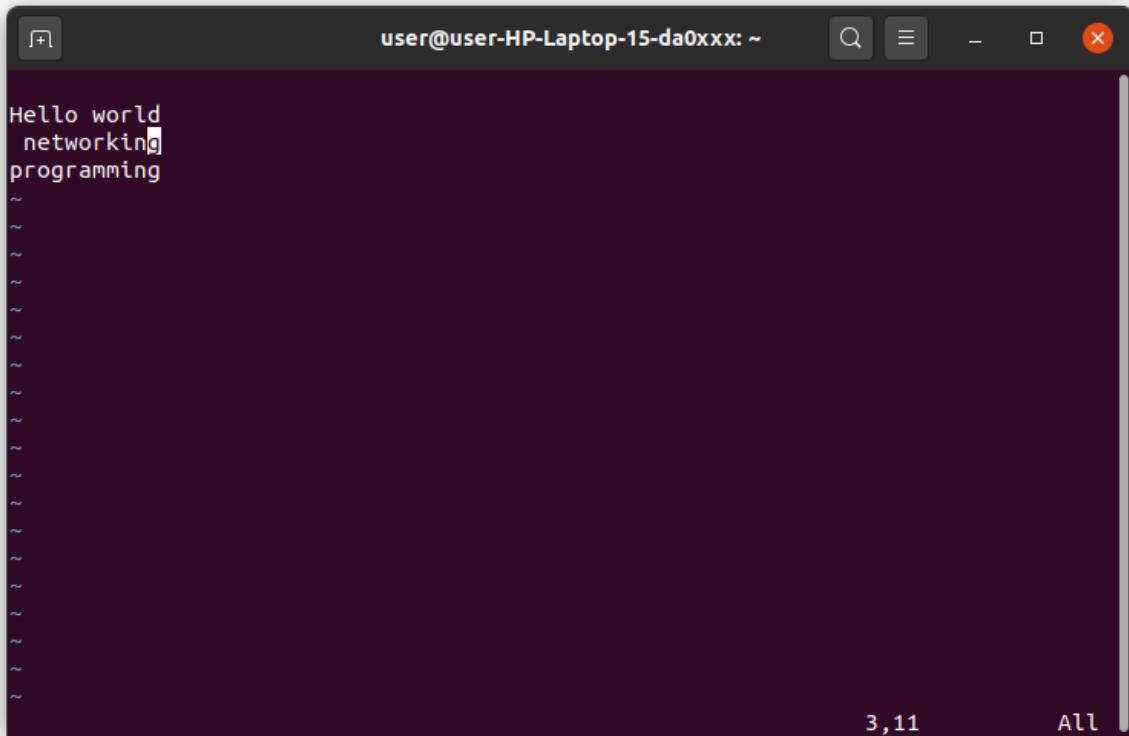


```
Hello world
programming
```

- **D** command: Deletes from the cursor position to the end of the current line.

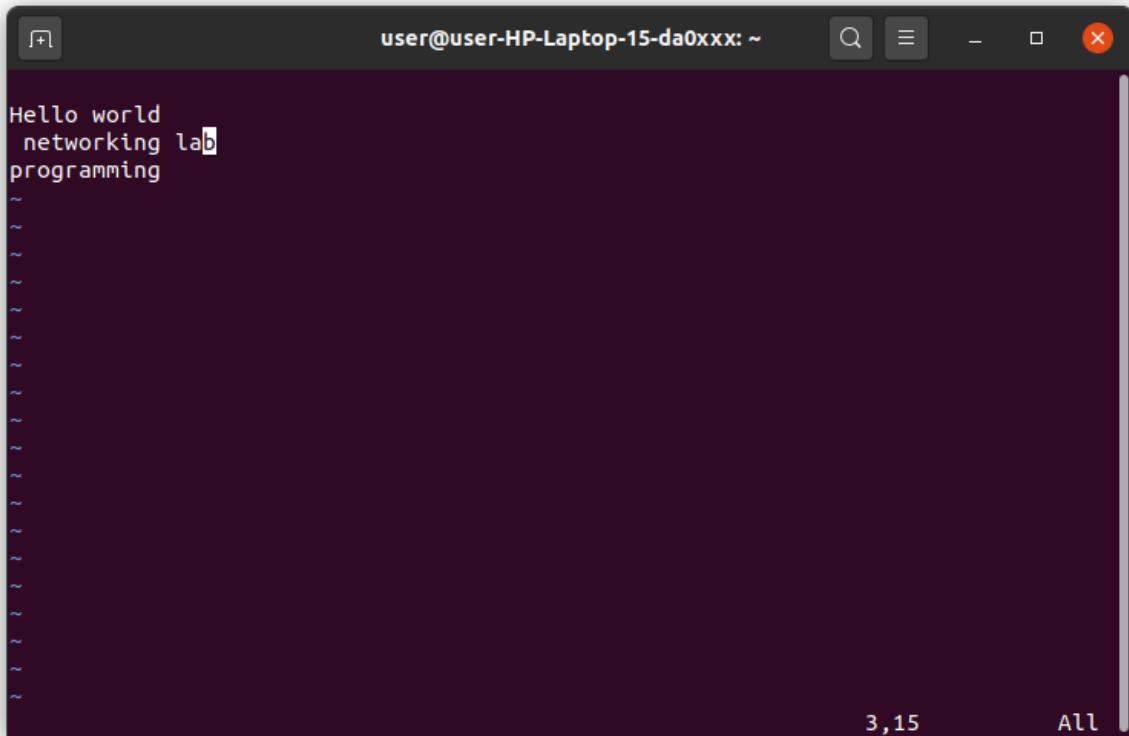


```
Hello world
networkinglab
programming
```



```
Hello world
networking
programming
~
```

- **‘dd’ command:** Deletes the line the cursor is on.



```
Hello world
networking lab
programming
~
```

A screenshot of a terminal window titled "user@user-HP-Laptop-15-da0xxx: ~". The window contains the text "Hello world" followed by "programming" on a new line. Below this, there are approximately 20 blank lines starting with a tilde (~). In the bottom right corner of the terminal window, the numbers "3,1" and "All" are displayed.

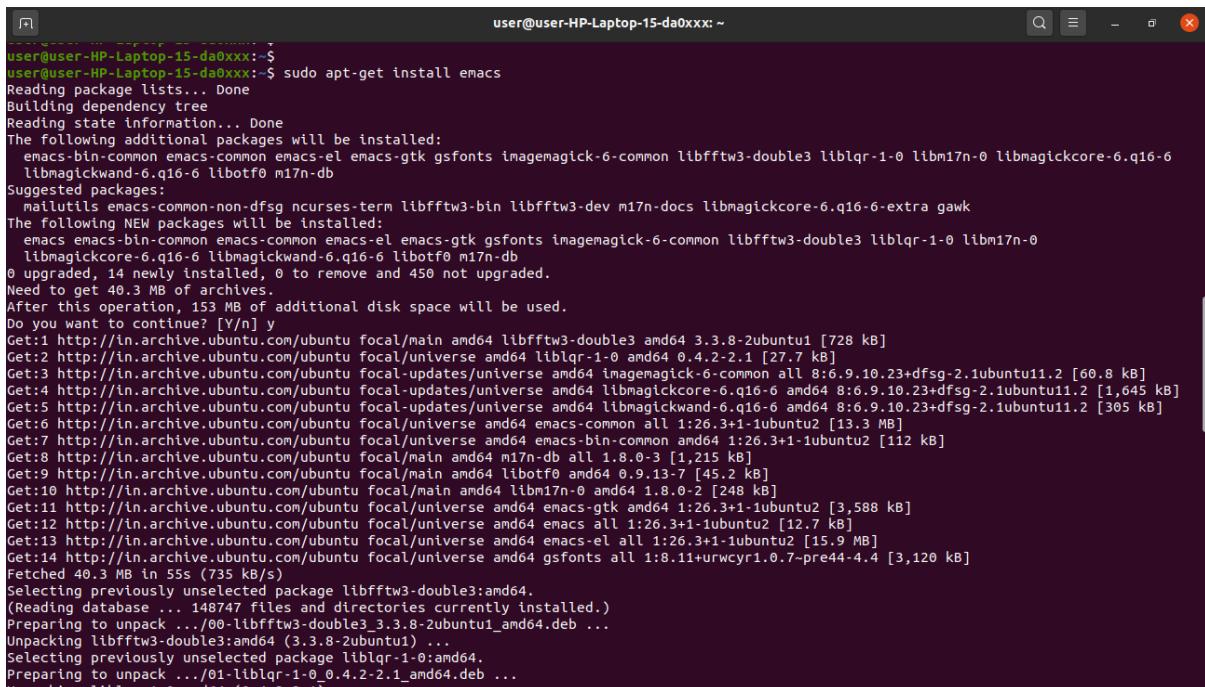
## EXPERIMENT-11

### FAMILIARISATION OF EMACS TEXT EDITOR

- very popular editor.
- most complex and flexible text editor than other text editors.
- freedom to the users to editing text files
- The emacs program can be running either in a terminal or in a GUI
- The main difference between text editors like vi, vim, nano, and the Emacs is that is faster, powerful, and simple in terms of usage because of its simple user interface.
- Emacs editor does not use an insert mode, and it is by default in editing mode, i.e., whatever you type will directly be written to the buffer, unless you manually enter command mode by using keyboard shortcuts.

### INSTALLING EMACS EDITOR ON UBUNTU

\$ sudo apt-get install emacs

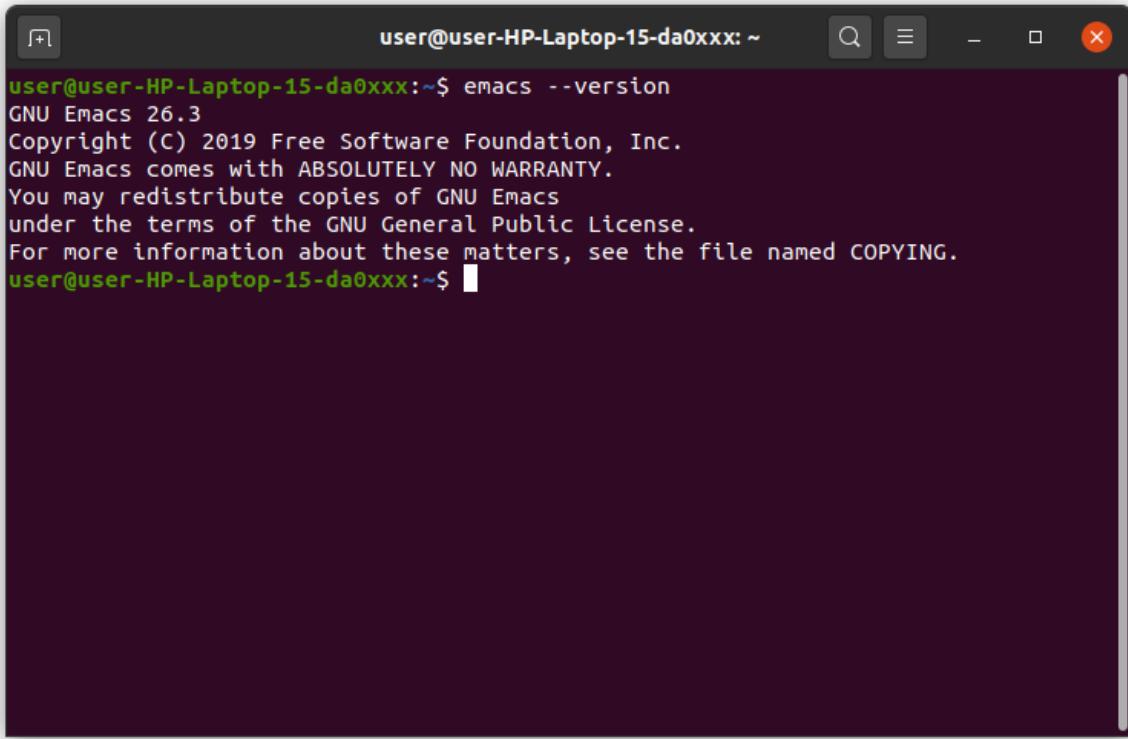


```

user@user-HP-Laptop-15-da0xxx:~$ sudo apt-get install emacs
[sudo] password for user: 
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  emacs-bin-common emacs-common emacs-el emacs-gtk gsfonts imagemagick-6-common libfftw3-double3 liblqr-1-0 libm17n-0 libmagickcore-6.q16-6
  libmagickwand-6.q16-6 libotf0 m17n-db
Suggested packages:
  mailutils emacs-common-non-dfsg ncurses-term libfftw3-bin libfftw3-dev m17n-docs libmagickcore-6.q16-6-extra gawk
The following NEW packages will be installed:
  emacs-bin-common emacs-common emacs-el emacs-gtk gsfonts imagemagick-6-common libfftw3-double3 liblqr-1-0 libm17n-0
  libmagickcore-6.q16-6 libmagickwand-6.q16-6 libotf0 m17n-db
0 upgraded, 14 newly installed, 0 to remove and 450 not upgraded.
Need to get 40.3 MB of archives.
After this operation, 153 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu focal amd64 libfftw3-double3 amd64 3.3.8-2ubuntu1 [728 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 liblqr-1-0 amd64 0.4.2-2.1 [27.7 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu focal-updates/universe amd64 imagemagick-6-common all 8:6.9.10.23+dfsg-2.1ubuntu11.2 [60.8 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu focal-updates/universe amd64 libmagickcore-6.q16-6 amd64 8:6.9.10.23+dfsg-2.1ubuntu11.2 [1,645 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu focal-updates/universe amd64 libmagickwand-6.q16-6 amd64 8:6.9.10.23+dfsg-2.1ubuntu11.2 [305 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 emacs-common all 1:26.3+1-1ubuntu2 [13.3 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 emacs-bin-common amd64 1:26.3+1-1ubuntu2 [112 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu focal/main amd64 m17n-db all 1:8.0-3 [1,215 kB]
Get:9 http://in.archive.ubuntu.com/ubuntu focal/main amd64 libotf0 amd64 0.9.13-7 [45.2 kB]
Get:10 http://in.archive.ubuntu.com/ubuntu focal/main amd64 libm17n-0 amd64 1:8.0-2 [248 kB]
Get:11 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 emacs-gtk amd64 1:26.3+1-1ubuntu2 [3,588 kB]
Get:12 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 emacs all 1:26.3+1-1ubuntu2 [12.7 kB]
Get:13 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 emacs-el all 1:26.3+1-1ubuntu2 [15.9 kB]
Get:14 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 gsfonts all 1:8.11+urwcyri.0.7-pre44-4.4 [3,120 kB]
Fetched 40.3 MB in 55s (735 kB/s)
Selecting previously unselected package libfftw3-double3:amd64.
(Reading database ... 148747 files and directories currently installed.)
Preparing to unpack .../00-libfftw3-double3_3.3.8-2ubuntu1_amd64.deb ...
Unpacking libfftw3-double3:amd64 (3.3.8-2ubuntu1) ...
Selecting previously unselected package liblqr-1-0:amd64.
Preparing to unpack .../01-liblqr-1-0_0.4.2-2.1_amd64.deb ...

```

\$ emac --version



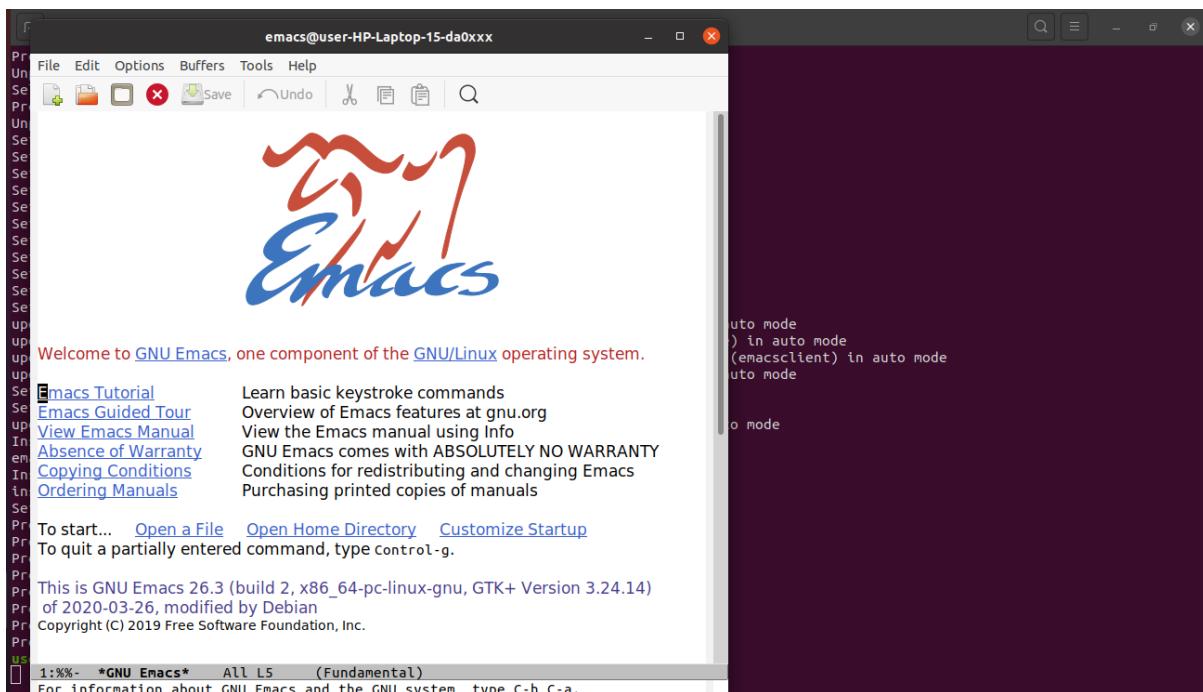
The screenshot shows a terminal window with a dark background and light-colored text. The title bar reads "user@user-HP-Laptop-15-da0xxx: ~". The command "emacs --version" is entered at the prompt, followed by its output:

```
user@user-HP-Laptop-15-da0xxx:~$ emacs --version
GNU Emacs 26.3
Copyright (C) 2019 Free Software Foundation, Inc.
GNU Emacs comes with ABSOLUTELY NO WARRANTY.
You may redistribute copies of GNU Emacs
under the terms of the GNU General Public License.
For more information about these matters, see the file named COPYING.
user@user-HP-Laptop-15-da0xxx:~$ █
```

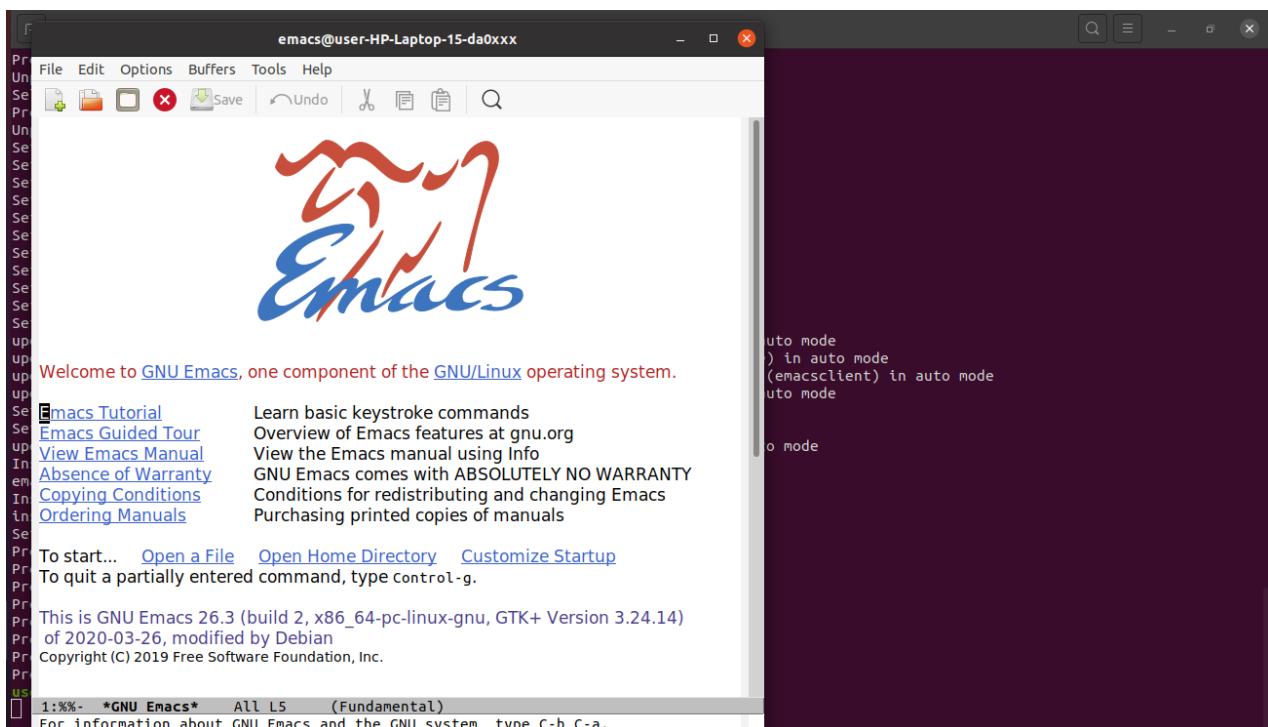
OPEN EMACS EDITOR (IN GUI WAY)

\$ emacs

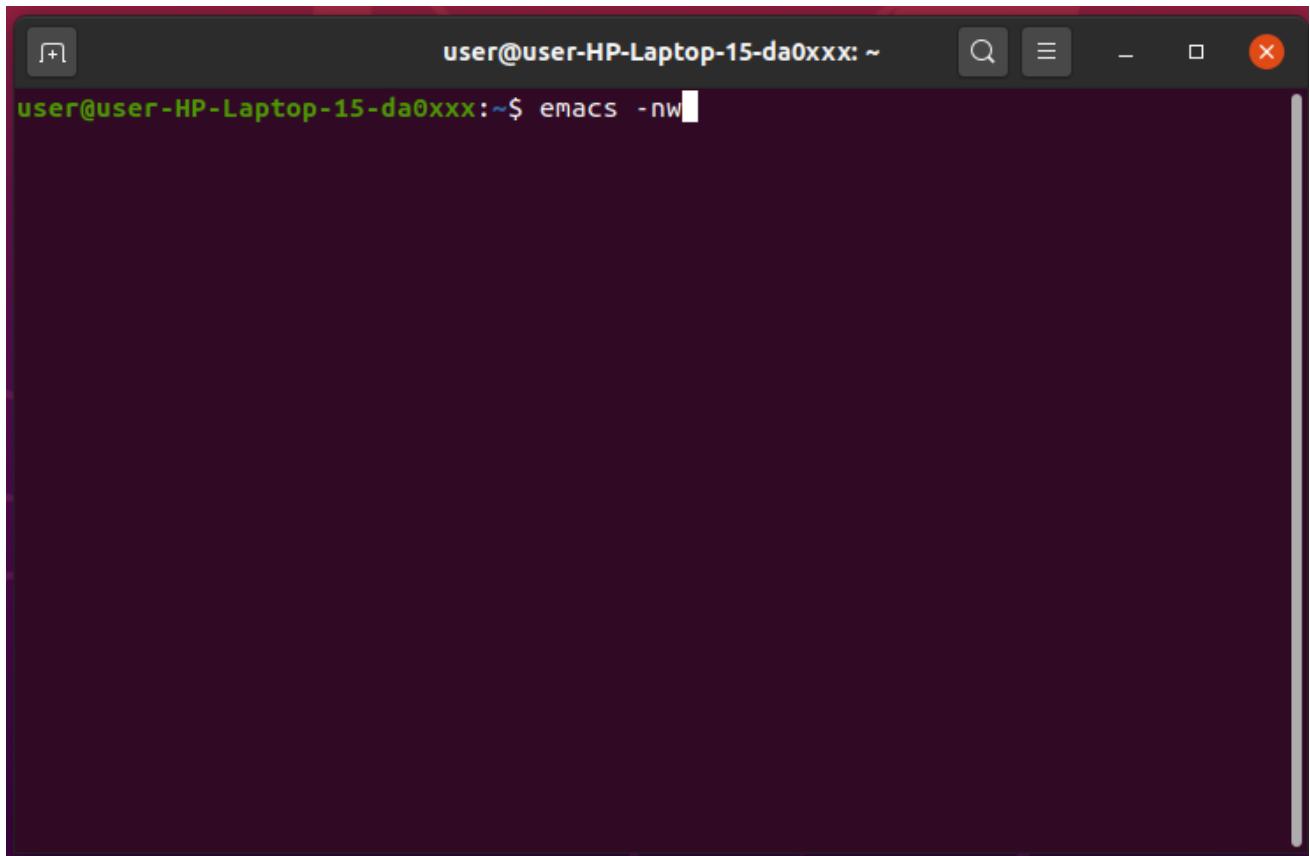
```
user@user-HP-Laptop-15-da0xxx: ~
Preparing to unpack .../12-emacs-el_1%3a26.3+1~ubuntu2_all.deb ...
Unpacking emacs-el (1:26.3+1~ubuntu2) ...
Selecting previously unselected package gsfots.
Preparing to unpack .../13-gsfots_1%3a8.11+urwcyri.0.7~pre44-4.4_all.deb ...
Unpacking gsfots (1:8.11+urwcyri.0.7~pre44-4.4) ...
Setting up libtbot0:amd64 (0.9.13-7) ...
Setting up imagemagick-6-common (8:6.9.10.23+dfsg-2.1ubuntu11.2) ...
Setting up m17n-db (1.8.0-3) ...
Setting up libm17n-0:amd64 (1.8.0-2) ...
Setting up gsfots (1:8.11+urwcyri.0.7~pre44-4.4) ...
Setting up libfftw3-double3:amd64 (3.3.8~2ubuntu1) ...
Setting up liblqr-1:0:amd64 (0.4.2-2.1) ...
Setting up emacs-common (1:26.3+1~ubuntu2) ...
Setting up libmagickcore-6.q16-6:amd64 (8:6.9.10.23+dfsg-2.1ubuntu11.2) ...
Setting up emacs-el (1:26.3+1~ubuntu2) ...
Setting up emacs-bin-common (1:26.3+1~ubuntu2) ...
update-alternatives: using /usr/bin/ctags.emacs to provide /usr/bin/ctags (ctags) in auto mode
update-alternatives: using /usr/bin/ebrowse.emacs to provide /usr/bin/ebrowse (ebrowse) in auto mode
update-alternatives: using /usr/bin/emacsclient.emacs to provide /usr/bin/emacsclient (emacsclient) in auto mode
update-alternatives: using /usr/bin/etags.emacs to provide /usr/bin/etags (etags) in auto mode
Setting up libmagickwand-6.q16-6:amd64 (8:6.9.10.23+dfsg-2.1ubuntu11.2) ...
Setting up emacs-gtk (1:26.3+1~ubuntu2) ...
update-alternatives: using /usr/bin/emacs-gtk to provide /usr/bin/emacs (emacs) in auto mode
Install emacsclient-common for emacs
emacsclient-common: Handling install of emacsclient flavor emacs
Install dictionaries-common for emacs
Install/dictionaries-common: Byte-compiling for emacsclient flavour emacs
Setting up emacs (1:26.3+1~ubuntu2) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu1) ...
Processing triggers for libc-bin (2.31-0ubuntu9) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for install-info (6.7.0.dfsg.2-5) ...
Processing triggers for fontconfig (2.13.1-2ubuntu3) ...
Processing triggers for desktop-file-utils (0.24-1ubuntu3) ...
user@user-HP-Laptop-15-da0xxx: ~$ emacs
```



## OR YOU CAN OPEN



## OPEN EMACS EDITOR (IN TERMINAL)



```
user@user-HP-Laptop-15-da0xxx: ~
File Edit Options Buffers Tools Help
Welcome to GNU Emacs, one component of the GNU/Linux operating system.

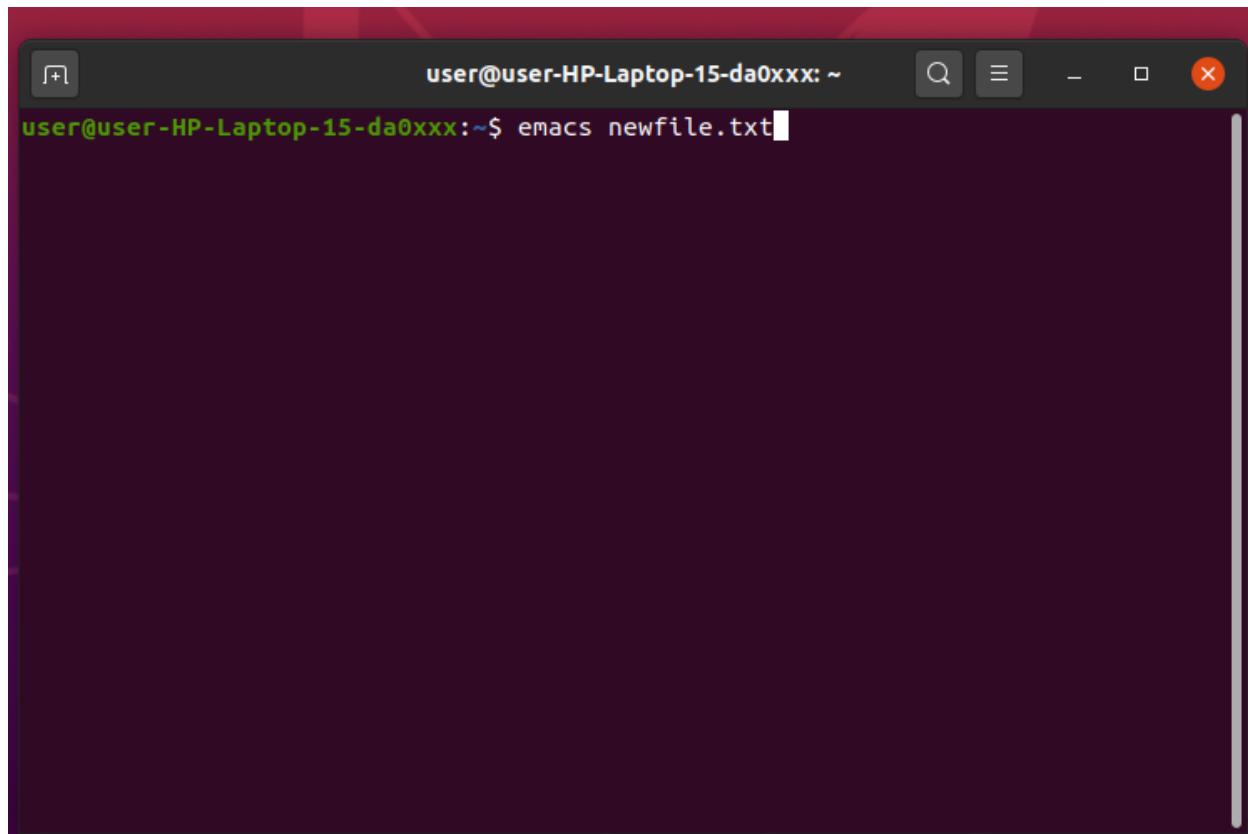
Get help      C-h (Hold down CTRL and press h)
Emacs manual  C-h r      Browse manuals  C-h i
Emacs tutorial  C-h t      Undo changes    C-x u
Buy manuals   C-h RET     Exit Emacs      C-x C-c
Activate menubar M-`      ( C-` means use the CTRL key.  M-` means use the Meta (or Alt) key.
If you have no Meta key, you may instead type ESC followed by the character.)
Useful tasks:
Visit New File          Open Home Directory
Customize Startup        Open *scratch* buffer

GNU Emacs 26.3 (build 2, x86_64-pc-linux-gnu, GTK+ Version 3.24.14)
of 2020-03-26, modified by Debian
Copyright (C) 2019 Free Software Foundation, Inc.

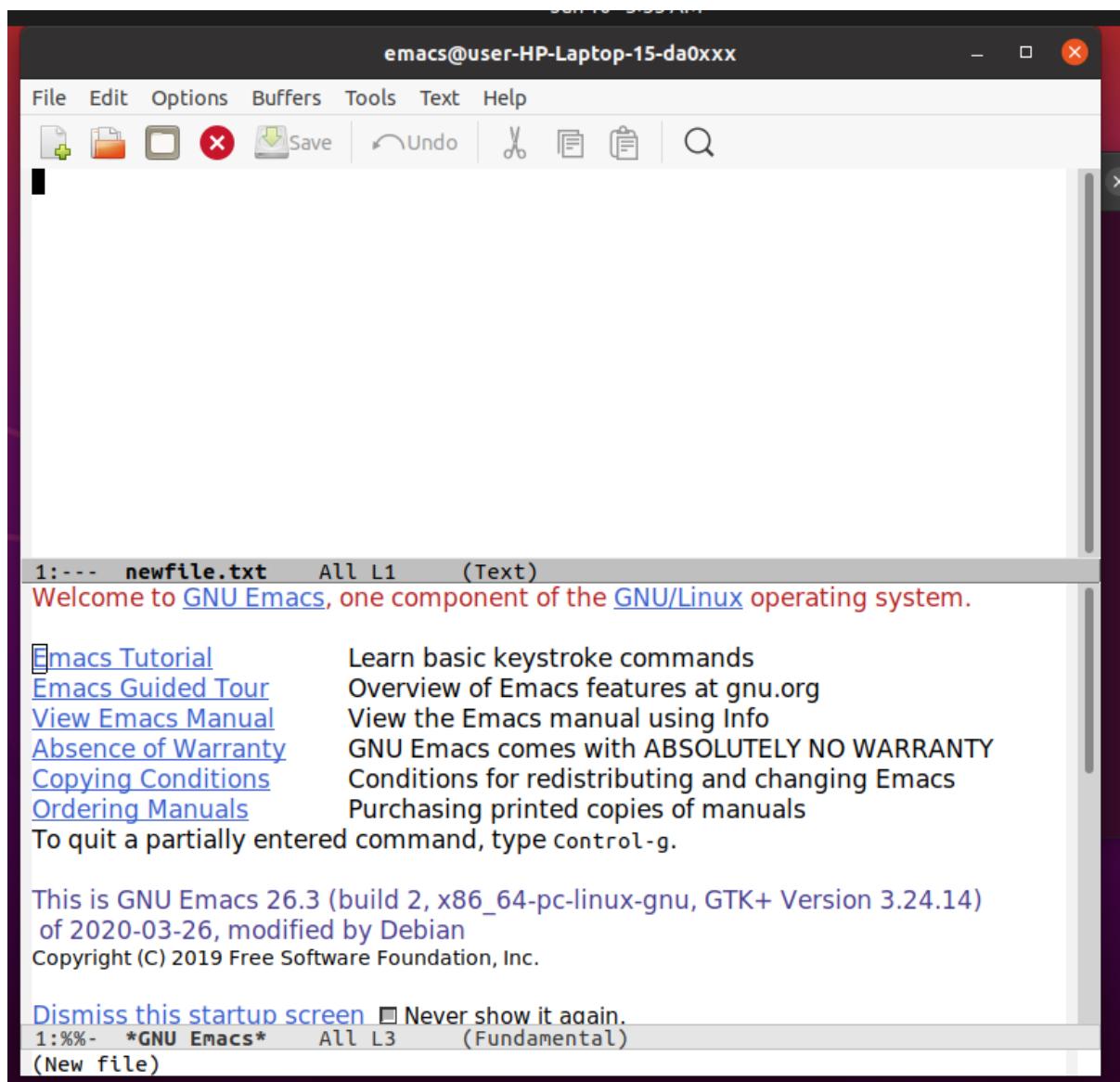
GNU Emacs comes with ABSOLUTELY NO WARRANTY; type C-h C-w for full details.
Emacs is Free Software--Free as in Freedom--so you can redistribute copies
of Emacs and modify it; type C-h C-c to see the conditions.
Type C-h C-o for information on getting the latest version.

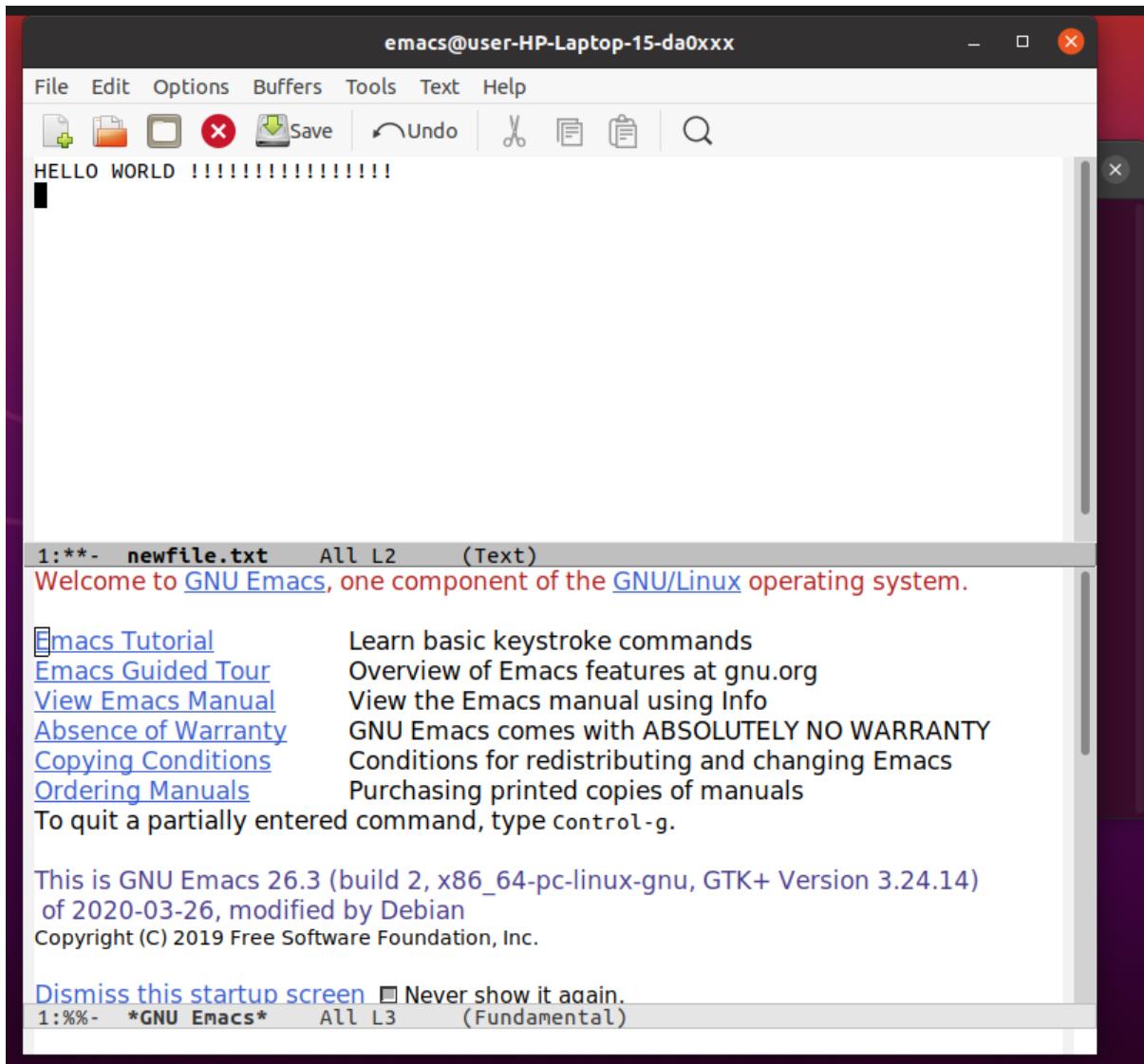
-111:%%--F1  *GNU Emacs*  All L1  (Fundamental) -----
For information about GNU Emacs and the GNU system, type C-h C-a.
```

## TO CREATE A NEW FILE OR EXISTING TEXT FILE



A screenshot of a terminal window titled "user@user-HP-Laptop-15-da0xxx: ~". The window has a dark background and light-colored text. In the title bar, there are icons for a plus sign, search, minimize, maximize, and close. The terminal prompt is "user@user-HP-Laptop-15-da0xxx:~\$". Below the prompt, the command "emacs newfile.txt" is being typed into the terminal. The cursor is positioned at the end of the command.





## SOME IMPORTANT COMMANDS

Commands	Actions
CTRL-X + CTRL-C	To exit emacs
CTRL-H	To access help documentation
CTRL-X + CTRL-S	To save the current file
CTRL-X + CTRL-W	To save as new file
CTRL-X + CTRL-U	To undo the last edit
CTRL-G	To cancel the current command
CTRL-D	To delete(Kill) the character at the cursor
ESC -D	To delete characters from the cursor to the end of the current line
CTRL-K	To delete characters from the cursor to the end of the current line
ESC -W	To copy region as kill
CTRL-Y	To paste into the buffer what had been deleted
CTRL-S	To search the string from the current cursor position
CTRL-R	To search the string above the current cursor position

## **SOME CURSOR MOVEMENT COMMAND**

Commands	Action
ESC-F	To move cursor forward
ESC-B	To move the cursor backward one word at a time
CTRL-A	To move the cursor to the beginning of the current line
CTRL-E	To move the cursor to the end of the current line
ESC-<	To move the cursor beginning of the buffer
ESC->	To move the cursor to the end of the buffer

## **PROGRAMMING IN EMACS**

- The emacs editor provides the environment to programmers develop programs such as c, java, HTML and shell script etc.
- After writing the program source code, it also compile,link,debug and execute itself in the terminal window to test it.

## **EXPERIMENT-12**

### **LINUX FILE SYSTEM**

A Linux file system is a structured collection of files on a disk drive or a partition.

A partition is a segment of memory and contains some specific data.

Linux file system is generally a built-in layer of a linux file system used to handle the data management of the storage.

The [Linux](#) file system contains the following sections:

1. The root directory (/)
2. A specific data storage format (EXT3, EXT4, BTRFS, XFS)
3. A partition or logical volume having a particular file system.

Some key features of linux file system are as following:

- Partition, Directories, and Drives
  - Case Sensitivity
  - File Extensions
  - Hidden files
- 1) Ext, Ext2, Ext3 and Ext4 file system
- The file system Ext stands for Extended File System
  - Ext file system is an older version, and is no longer used due to some limitations.
  - Ext2 is the first Linux file system that allows managing two terabytes of data
  - Ext3 is developed through Ext2; it is an upgraded version of Ext2 and contains backward compatibility.
  - Ext4 file system is the faster file system among all the Ext file systems.
- 2) JFS File System
- JFS stands for Journaled File System
  - It is an alternative to the Ext file system.
  - It can also be used in place of Ext4, where stability is needed with few resources

### 3) ReiserFS File System

- ReiserFS is an alternative to the Ext3 file system
- It has improved performance and advanced features.

### 4) XFS File System

- XFS file system was considered as high-speed JFS, which is developed for parallel I/O processing.

### 5) Btrfs File System

- Btrfs stands for the B tree file system
- It is used for fault tolerance, repair system, fun administration, extensive storage configuration, and more

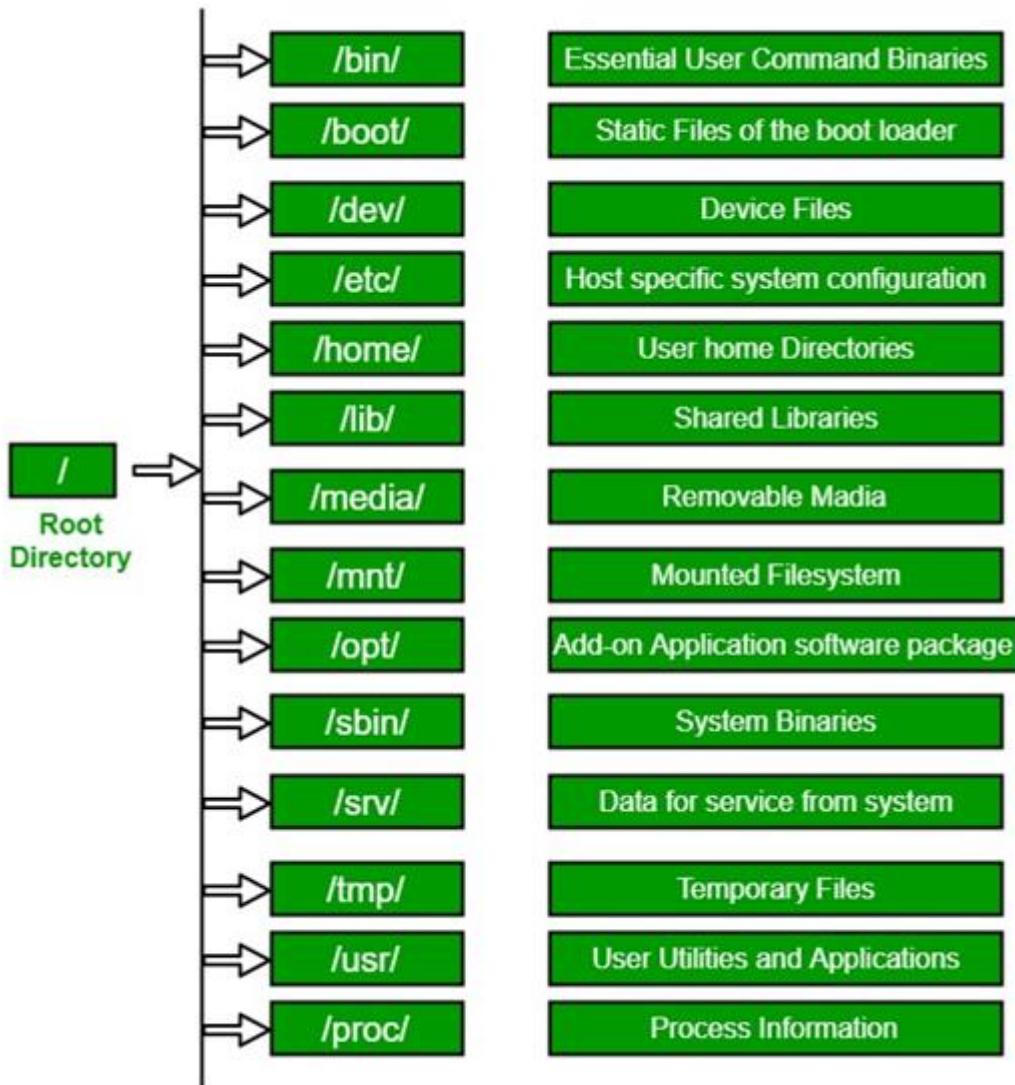
### 6) Swap File System

- The swap file system is used for memory paging in Linux operating system during the system hibernation
- A system that never goes in hibernate state is required to have swap space equal to its RAM size.

## **LINUX FILE SYSTEM HIERARCHY**

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

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



- ROOT DIRECTORY

All the directories in the Linux system comes under the root directory which is represented by a forward slash (/). Primary hierarchy root and root directory of the entire file system hierarchy.

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

/bin- The '/bin' directory contains user binaries, executable files, Linux commands that are used in single user model Contains binary executables

- Common linux commands you need to use in single-user modes are located under this directory.
- common commands that are used by all the users, like cat, cp, cd, ls, etc.

/sbin-'sbin' directory also contains executable files.

but unlike '/bin' it only contains system binaries which require root privilege to perform certain tasks and are helpful for system maintenance purpose.

e.g. fsck, root, init, ifconfig, etc.

```
razik@razik-VirtualBox:~$ ls /bin
['
aa-enabled
aa-exec
ab
aconnect
acpi_listen
add-apt-repository
addpart
alsabat
alsaloop
alsamixer
alsatplg
alsaucm
amidi
amixer
amuFormat.sh
apg
apgbfm
aplay
aplaymidi
apport-bug
apport-cli
apport-collect
mknode
mksquashfs
mktemp
mkzftree
mlabel
mmcli
mmd
mmount
mmove
monitor-sensor
more
mount
mountpoint
mousetweaks
mpartition
mrdr
mren
mscompress
msexpand
mshortname
mshowfat
mt
mt-gnu
```

/lib-The '/lib' directory contains shared libraries which are often used by the '/bin' and '/sbin' directories.

- /lib/modules: The '/lib/modules' stores kernel modules and has a directory for each installed kernel. Modules are meant to use extra hardware support without making a new kernel.
- /lib32 and /lib64: During compilation time of libraries you'll encounter through the directories named '/lib32' and '/lib64' which will clarify register size to be used. A 64-bit system may have compatibility for 32-bit binary.

It also contains kernel module. These filenames are identifiable as ld\* or lib\*.

```
razik@razik-VirtualBox:~$ ls /lib
accountsservice           libvmtools.so.0
apache2                   libvmtools.so.0.0.0
apg                        linux
apparmor                  linux-boot-probes
apt                        linux-sound-base
aspell                     locale
binfmt.d                  lsb
bluetooth                 man-db
bolt                       mecab
brltty                     memtest86+
cgi-bin                    mime
cnf-update-db              modprobe.d
command-not-found          modules
console-setup              modules-load.d
cpp                         mysql
crda                        netplan
cups                        networkd-dispatcher
dbus-1.0                   NetworkManager
debug                      nvidia
dpkg                        openssh
eject                      open-vm-tools
emacsclient-common          os-prober
environment.d              os-probes
```

/opt- term 'opt' is short for optional.

Its main purpose is to store optional application software packages.

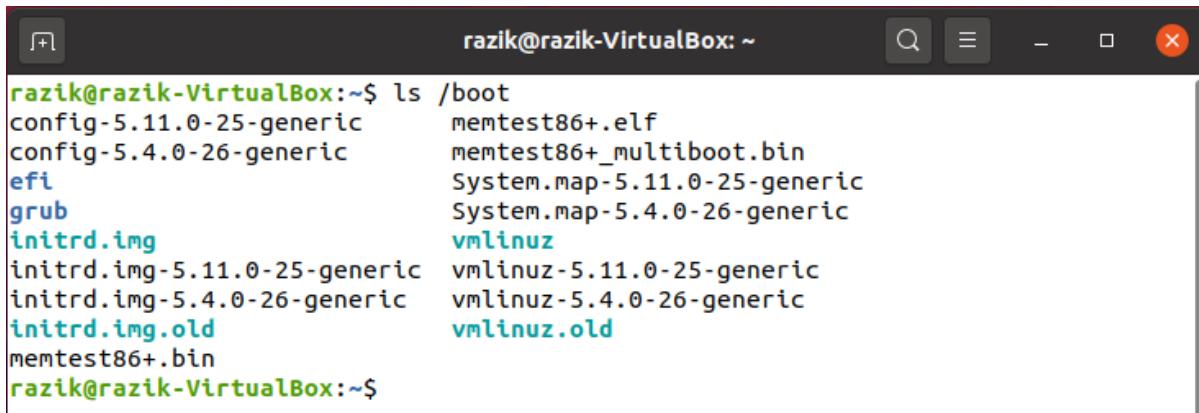
Add-on applications from individual vendors should be installed in '/opt'.

And so in some systems '/opt' is empty as they may not have any add-on application.

```
razik@razik-VirtualBox:~$ ls /opt
razik@razik-VirtualBox:~$
```

/boot-The '/boot' directory contains boot loader files which are essential to boot the system.

In other words, they only contain files which are needed for a basic Linux system to get up and going.

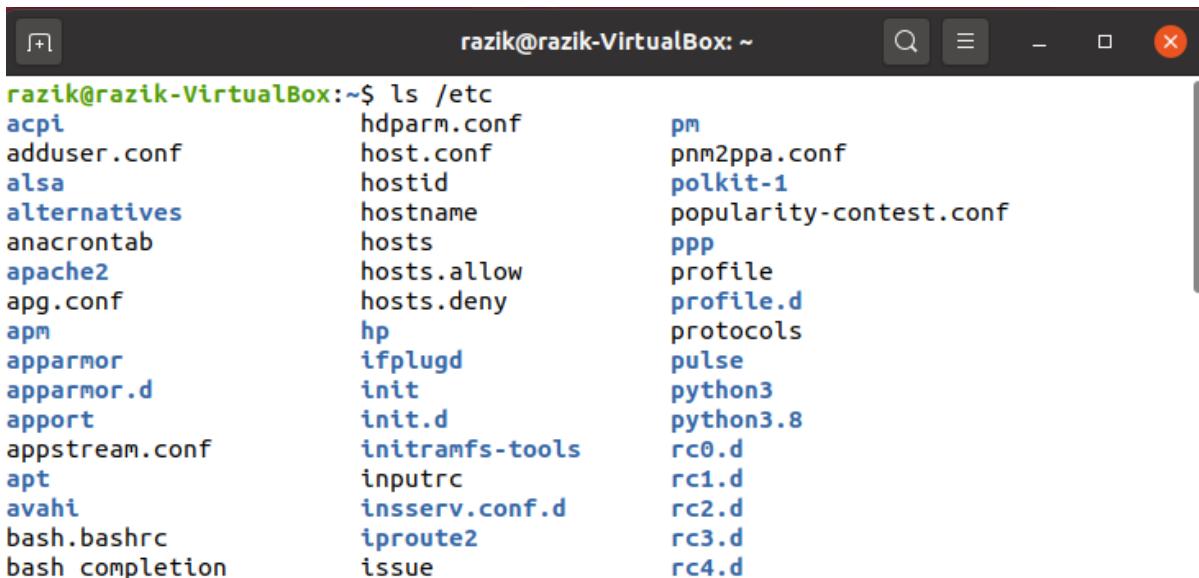


```
razik@razik-VirtualBox:~$ ls /boot
config-5.11.0-25-generic      memtest86+.elf
config-5.4.0-26-generic      memtest86+_multiboot.bin
efi                           System.map-5.11.0-25-generic
grub                          System.map-5.4.0-26-generic
initrd.img                     vmlinuz
initrd.img-5.11.0-25-generic  vmlinuz-5.11.0-25-generic
initrd.img-5.4.0-26-generic   vmlinuz-5.4.0-26-generic
initrd.img.old                 vmlinuz.old
memtest86+.bin
razik@razik-VirtualBox:~$
```

/etc-All the machine related configuration files are kept in '/etc'.

Almost everything related to the configuration of your system is placed here.

It also contain startup and shutdown shell script which is used to start and stop a program. All the files are static and text based and no binary files can be placed in this directory.



```
razik@razik-VirtualBox:~$ ls /etc
acpi                      hdparm.conf        pm
adduser.conf                host.conf          pm2ppa.conf
alsa                       hostid             polkit-1
alternatives               hostname           popularity-contest.conf
anacrontab                 hosts              ppp
apache2                    hosts.allow        profile
apg.conf                   hosts.deny        profile.d
apm                        hp                  protocols
apparmor                   ifplugd            pulse
apparmor.d                 init                python3
apport                     init.d              python3.8
appstream.conf             initramfs-tools  rc0.d
apt                        inputrc            rc1.d
avahi                      inserv.conf.d  rc2.d
bash.bashrc                iproute2           rc3.d
bash_completion            issue              rc4.d
razik@razik-VirtualBox:~$
```

/home-The '/home' directory stores users personnel files.

After the '/home' there is a directory which is generally named at the user's name like we have '/home/username'.

Inside this directory we have our sub-directories like Desktop, Downloads, Documents, pictures, etc.

```
razik@razik-VirtualBox:~$ ls /home
razik
razik@razik-VirtualBox:~$
```

/root- The '/root' directory is the home directory of the root user.

- '/root' directory is different from (/) root.

/srv-The term 'srv' is short for service.

- The '/srv' directory contains server specific data for services provided by the system like www, cvs, rysync, ftp, etc.

/media-The '/media' directory acts as a mount point for removable media devices such as CD-Rom, floppy, USB devices, etc.

- This is newly introduced directory and hence a system can run without this directory also.

/tmp-The term 'tmp' stands for temporary.

- Data stored in '/tmp' is temporary and may use either disk space or RAM.

```
razik@razik-VirtualBox:~$ ls /tmp
config-err-mig9Ch
ssh-a7ijvnd3Z8lr
systemd-private-a047548e9d0045f797f182738ab89682-apache2.service-GaMR7f
systemd-private-a047548e9d0045f797f182738ab89682-colord.service-yWj45i
systemd-private-a047548e9d0045f797f182738ab89682-ModemManager.service-mXsnVi
systemd-private-a047548e9d0045f797f182738ab89682-switcheroo-control.service-X0shPf
systemd-private-a047548e9d0045f797f182738ab89682-systemd-logind.service-BcuNUg
systemd-private-a047548e9d0045f797f182738ab89682-systemd-resolved.service-Zvh1bj
systemd-private-a047548e9d0045f797f182738ab89682-systemd-timesyncd.service-6EqdL
f
systemd-private-a047548e9d0045f797f182738ab89682-upower.service-u0Qmcg
tracker-extract-files.1000
tracker-extract-files.125
VMwareDnD
razik@razik-VirtualBox:~$
```

## FILE PERMISSIONS:

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

### 1.User

- A user is the owner of the file. By default, the person who created a file becomes its owner. Hence, a user is also sometimes called an owner.

## 2.Group

- A user- group can contain multiple users. All users belonging to a group will have the same Linux group permissions access to the file.

## 3.Other

- Any other user who has access to a file. This person has neither created the file, nor he belongs to a usergroup who could own the file.
- Every file and directory in your UNIX/Linux system has following 3 permissions defined for all the 3 owners

### 1.Read:

This permission give you the authority to open and read a file.

### 2.Write:

The write permission gives you the authority to modify the contents of a file. The write permission on a directory gives you the authority to add, remove and rename files stored in the directory

### 3.Execute:

If the execute permission is not set, you might still be able to see/modify the program code(provided read & write permissions are set), but not run it.

- chmod Linux command, used to change the read, write, and execute permissions of files and directories
- chmod +rwx filename to add permissions.
- chmod -rwx directory name to remove permissions.
- chmod +x filename to allow executable permissions.
- chmod -wx filename to take out write and executable permissions.

**EXPERIMENT-13****FAMILIARISATION TO LINUX SHELL AND 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.

**Kernel**

- The kernel is a computer program that is the core of a computer's operating system, with complete control over everything in the system.
- It manages following resources of the Linux system –
  - \* File management
  - \* Process management
  - \* I/O management
  - \* Memory management
  - \* Device management etc.

**Shell**

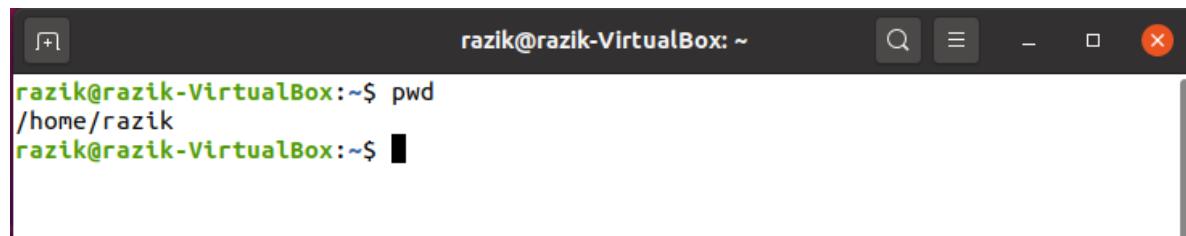
- A shell is a special user program which provides an interface to users to use operating system services.
- Shell accept human readable commands from the user and convert them into something which the kernel can understand.
- The shell gets started when the user logs in or starts the terminal.
- Shell is broadly classified into two categories:
  - \* Command Line Shell
  - \* Graphical shell
- The shell in the linux operation takes input in the form of commands, processes them and gives an output .
- When you are in the terminal the shell issues a command prompt.
- It is usually \$ where you can type the input which is then executed by hitting Enter key.

## Command Line Shell

Shell can be accessed by a user using a command line interface. A special program called Terminal in linux/macOS or Command Prompt in Windows OS is provided to type in the human readable commands such as “cat”, “ls” etc. and then it is being executed.

## Graphical Shells

Graphical shells provide means for manipulating programs based on graphical user interface (GUI), by allowing for operations such as opening, closing, moving and resizing windows, as well as switching focus between windows.



A screenshot of a terminal window titled "razik@razik-VirtualBox: ~". The window contains the command "razik@razik-VirtualBox:~\$ pwd" followed by the output "/home/razik". The terminal has a dark theme with light-colored text and standard window controls at the top right.

```
razik@razik-VirtualBox:~$ pwd
/home/razik
razik@razik-VirtualBox:~$
```

## **BASH (Bourne Again Shell) :**

- It is the most widely used shell in Linux systems.
- It is used as the default login shell in Linux systems
- It can also be installed on Windows OS.

## **Shell Scripting**

- Shell script is a series of command(s) stored in a plain text file.
- These files are called Shell Scripts or Shell Programs.
- Each shell script is saved with .sh file extension.
- eg: myscript.sh

## **Creating a shell script**

1. Creating a file using any text editor

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.sh

## Variables in Shell

- Variables store data in the form of character and number.
- In Linux (Shell), there are two types of variable:
  - System variables
    - Created and maintained by Linux itself.
      - This type of variable defined in CAPITAL LETTERS.
    - User defined variables (UDV)
      - Created and maintained by the user.
      - This type of variable is defined in lower letters.

## The read Statement

- Used to get input (data from user) from keyboard and store (data) to variables.
- Syntax:
  - read variable1, variable2,...variableN

## Conditional Statements

The if...else statements

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

Unix Shell supports following forms of if...else statement –

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

-used for making decisions in shell script.

-If the condition is true

-Then command1 is executed.

#### **Syntax:**

```
if condition
    then
        command1 if condition is true or if exit status of
        condition is 0
    fi
```

if condition - Example

```
1 #!/bin/sh
2 echo "enter the age"
3 read age
4 if [ $age -ge 18 ]
5 then
6   echo "eligible"
7 fi
```

Output :

```
enter the age
22
eligible
```

- if...else...fi

-If given condition is true

-Then command1 is executed

-Otherwise command2 is executed.

Syntax:

```

if condition

    then

        condition is zero (true - 0)

        execute all commands up to else statement

else

    if condition is not true then

        execute all commands up to fi

fi

```

if...else...fi -Example

```

1 #!/bin/sh
2 echo "enter the age"
3 read age
4 if [ $age -ge 18 ]
5 then
6   echo "eligible"
7 else
8   echo "Not eligible"
9 fi

```

Output :

```
enter the age
10
Not eligible
```

```
enter the age
50
eligible
```

## Loops in Shell Scripts

- Bash supports:

  1. for loop

## 2. while loop

Note that in each and every loop,

- First, the variable used in loop condition must be initialized, then execution of the loop begins.
- A test (condition) is made at the beginning of each iteration.
- The body of the loop ends with a statement that modifies the value of the test (condition) variable.

## 1. for Loop

Syntax:

```
for (( expr1; expr2; expr3 ))
do
repeat all statements between do and done until
expr2 is TRUE
done
```

for loop - Example

```
1 #!/bin/sh
2 echo "Enter the limit:"
3 read n
4 echo "....."
5
6 for ((i=1; i<=n; i++))
7 do
8 echo "$i"
9 done
```

Output :

```
Enter the limit:  
22  
.....  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19
```

## 2. while Loop

Syntax:

```
while [ condition ]
```

```
do
```

```
    command1
```

```
    command2
```

```
    ..
```

```
    ....
```

```
done
```

While loop - Example

```

1 #!/bin/sh
2 a=1
3 while [ $a -le 15 ]
4 do
5   echo "$a"
6   a=$((a+2))
7 done

```

Output :

```

1
3
5
7
9
11
13
15

```

## Functions

- Function is a series of instructions/commands.
- Function performs particular activity in shell.
- Code in a function is only executed when a function is ‘called’

Syntax:

```

function-name ( )
{
  Function body
}

```

Function - Example 1

```

1 #!/bin/sh
2 hello () {
3   echo "Hello world"
4 }
5 hello

```

Output :

```
Hello world
```

Function(using argument) - Example 2

```

1 #!/bin/sh
2 Addition () {
3 a=$((b+c))
4 echo "sum= $a"
5 }
6 echo "Enter the numbers to be add:"
7 read b
8 read c
9 Addition $b $c

```

Output :

```

Enter the numbers to be add:
7
14
sum= 21
Enter the numbers to be add:
9
-5
sum= 4

```

## Command Line Arguments

Command	Explanation
\$0	Represent the command or script.
\$1	Represent argument 1.
\$#	Represents the total number of arguments
\$*	Represents all arguments.
\$\$	Represents the PID of a running script.

## Example

```

1 #!/bin/sh
2 echo "File Name: $0"
3 echo "Single argument : Hi $1"
4 echo "Multiple arguments : Hloo $*"
5 echo "Multiple arguments : Hi $@"
6 echo "Total number of arguments : $#"
7 echo "Process ID $$"

```

## Output :

```

File Name: argum.sh
Single argument : Hi Paulo
Multiple arguments : Hloo Paulo Coelho Alchemist
Multiple arguments : Hi Paulo Coelho Alchemist
Total number of arguments : 3
Process ID 8741

```

**Mathematical Operators**

Mathematical Operator in Shell Script	Meaning	Normal Arithmetical/Mathematical Statements	But in Shell	
			For test statement with if command	For [ expr ] statement with if command
-eq	is equal to	5 == 6	if test 5 -eq 6	if [ 5 -eq 6 ]
-ne	is not equal to	5 != 6	if test 5 -ne 6	if [ 5 -ne 6 ]
-lt	is less than	5 < 6	if test 5 -lt 6	if [ 5 -lt 6 ]
-le	is less than or equal to	5 <= 6	if test 5 -le 6	if [ 5 -le 6 ]
-gt	is greater than	5 > 6	if test 5 -gt 6	if [ 5 -gt 6 ]
-ge	is greater than or equal to	5 >= 6	if test 5 -ge 6	if [ 5 -ge 6 ]

## String Operators

Operator	Meaning
string1 = string2	string1 is equal to string2
string1 != string2	string1 is NOT equal to string2
-n string1	string1 is NOT NULL or not defined
-z string1	string1 is NULL and does exist

**EXPERIMENT-14****LINUX SHELL SCRIPTING PROBLEMS**

1. Write a shell script to get current date, time, username and current working?

**Ans :**

```
echo "HELLO,$LOGNAME"
echo "CURRENT DATE IS $(date)"
echo "USER IS $(whoami)"
echo "CURRENT DIRECTORY $(pwd)"
```

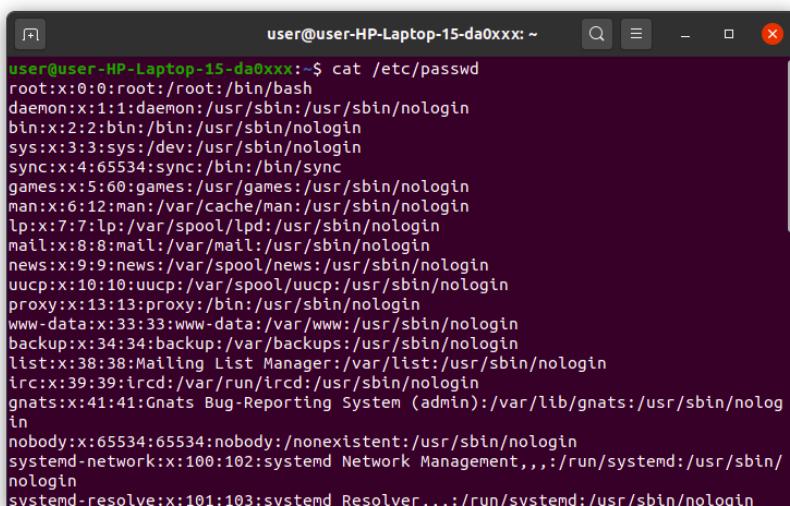
**OUTPUT:**

```
user@user-HP-Laptop-15-da0xxx:~/shellscriptingpgms$ bash pgm1.sh
HELLO,user
CURRENT DATE IS Friday 06 August 2021 06:12:32 PM IST
USER IS user
CURRENT DIRECTORY /home/user/shellscriptingpgms
user@user-HP-Laptop-15-da0xxx:~/shellscriptingpgms$
```

2. How to print the login names of all users on a system?

**Ans :**

```
cat /etc/passwd
```

**OUTPUT:**


```
user@user-HP-Laptop-15-da0xxx:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin/nologin
bin:x:2:2:bin:/bin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:0:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
```

3. How can we pass arguments to a script in Linux? And how to access these arguments from within the script?

**Ans :**

```
_____echo "First number : $1"
echo "Second number : $2"
echo "Third number : $3"
echo "Fourth number : $4"
echo "Fifth number : $5"
```

**OUTPUT:**

```
user@user-HP-Laptop-15-da0xxx:~/shells scriptingpgms$ bash pgm2.sh 100 200 300 400 500
First number : 100
Second number : 200
Third number : 300
Fourth number : 400
Fifth number : 500
user@user-HP-Laptop-15-da0xxx:~/shells scriptingpgms$
```

4. How to set an array in Linux?

**Ans :**

```
_____array=(hello i am ganga)
echo ${array[@]}
echo ${array[*]}
echo ${array[@]:0}
echo ${array[*]:0}
echo ${array[0]}
echo ${array[1]}
echo ${array[2]}
echo ${array[3]}
```

**OUTPUT:**

```
user@user-HP-Laptop-15-da0xxx:~/shells scriptingpgms$ bash pgm6.sh
hello i am ganga
hello
i
am
ganga
user@user-HP-Laptop-15-da0xxx:~/shells scriptingpgms$
```

5. How to check if a directory exists?

**Ans :**

```
if [ -d "/home/user/shells scriptingpgms" ]
then
    echo "/home/user/shells scriptingpgms exists."
else
    echo "Error: Directory /home/user/shells scriptingpgms does not exists."
fi
```

**OUTPUT :**

```
user@user-HP-Laptop-15-da0xxx:~/shells scriptingpgms$ bash pgm4.sh
/home/user/shells scriptingpgms exists.
user@user-HP-Laptop-15-da0xxx:~/shells scriptingpgms$
```

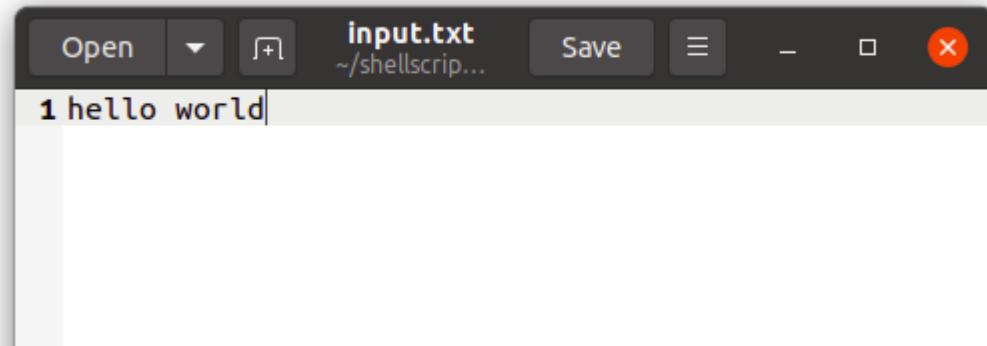
6. What is the difference between \$\* and \${@}?

**Ans:**

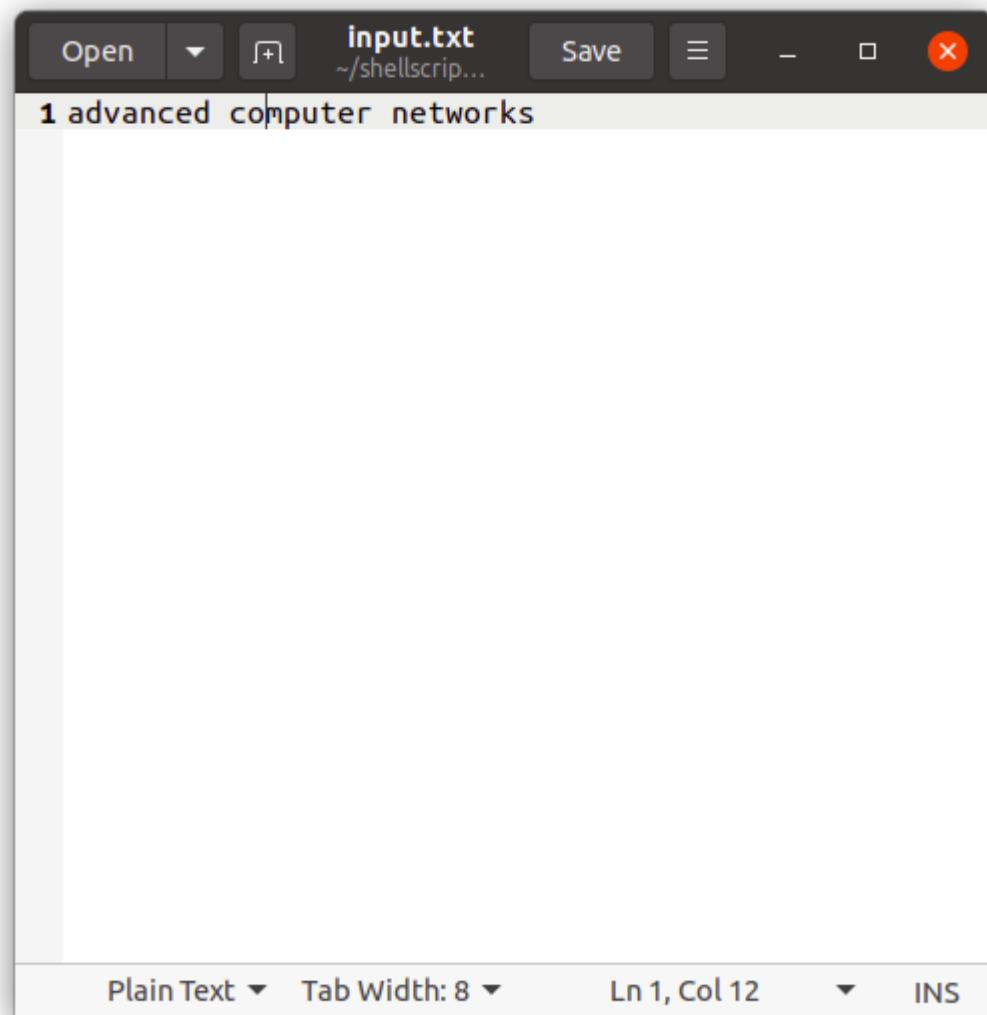
\_\_\_\_\_ \${@} treats each quoted arguments as separate arguments but \$\* considers the entire set of positional parameters as a single string.

7. Use the sed command to replace the content of the file?

**Ans:**



```
user@user-HP-Laptop-15-da0xxx:~/shells scriptingpgms$ sed -i 's/hello world/advanced computer networks/g' input.txt  
user@user-HP-Laptop-15-da0xxx:~/shells scriptingpgms$
```



Plain Text ▾ Tab Width: 8 ▾ Ln 1, Col 12 ▾ INS

8. Write a script to compare numbers?

Ans :

```
var1=100
var2=200
if [ $var2 -gt $var1 ]
then
echo "$var2 is greater than $var1"
fi
#second comparison
echo "_____"
if [ $var1 -gt 1000 ]
then
echo "$var1 is greater than 1000"
else
echo
echo "$var1 is less than 1000"
fi
```

**OUTPUT:**

```
user@user-HP-Laptop-15-da0xxx:~/shells scripting pgms$ bash pgm8.sh
200 is greater than 100
-----
100 is less than 1000
user@user-HP-Laptop-15-da0xxx:~/shells scripting pgms$
```

9. Write a shell script to check to see if the file “file\_path” exists. If it does exist, display “file\_path passwords are enabled.” Next, check to see if you can write to the file. If you can, display “You have permissions to edit “file\_path.”” If you cannot, display “You do NOT have permissions to edit“file\_path””?

**Ans :**

```
_____FILE="/home/user/shellscriptingpgms/pgm1"
if [ -e "$FILE" ]
then
echo "$FILEpasswords are enabled"
fi
if [ -x "$FILE" ]
then
echo "You have permission to execute $FILE"
else
echo "You have no permission to execute $FILE"
fi
```

**output :**

```
user@user-HP-Laptop-15-da0xxx:~/shellscriptingpgms$ bash pgm9.sh
You have no permission to execute /home/user/shellscriptingpgms/pgm1
user@user-HP-Laptop-15-da0xxx:~/shellscriptingpgms$
```

10. How to print all array indexes?

**Ans :**

```
_____arr=(10 20 30 40 50)
echo "index : value"
for((i=0;i<#${arr[@]};i++))
do
echo "$i : ${arr[$i]}"
done
```

**OUTPUT :**

```
user@user-HP-Laptop-15-da0xxx:~/shellscriptingpgms$ bash pgm10.sh
index : value
0      :    10
1      :    20
2      :    30
3      :    40
4      :    50
user@user-HP-Laptop-15-da0xxx:~/shellscriptingpgms$
```

11 . Write a shell script to display the last updated file or the newest file in a directory?

**Ans :**

```
_____ls -lrt | grep ^- | awk 'END{print $NF}'
```

**OUTPUT:**

```
user@user-HP-Laptop-15-da0xxx:~/shellscriptingpgms$ ls -lrt | grep ^- | awk '{print $NF}' | pgm10.png
user@user-HP-Laptop-15-da0xxx:~/shellscriptingpgms$
```

12. Write a shell script that adds an extension “.new” to all the files in directory.

**Ans :**

```
dir=$1
for file in '$1/*'
do
mv $file $file.new
done
```

13. Write a shell script to print a number in reverse order. It should support

- a. the following requirements.
- b. The script should accept the input from the command line.
- c. If you don't input any data, then display an error message to execute the script correctly

**Ans :**

```
echo "Enter a number to be reversed :"
read n
if [ $n == 0 ]
then
echo "Please provide the correct input in the below format."
echo "Usage: $0 number"
echo " This script will reverse the given number."
echo " For eg. $0 1234, will print 4321"
exit 1

else
rev=0
sd=0
while [ $n -gt 0 ]
do
```

```

sd=`expr $n % 10`
rev=`expr $rev \* 10 + $sd`
n=`expr $n / 10`
done
echo "Reverse number is $rev"
fi

```

**OUTPUT:**

```

user@user-HP-Laptop-15-da0xxx:~/shellspringpgms$ bash pgm13.sh
Enter a number to be reversed :
0
Please provide the correct input in the below format.
Usage: pgm13.sh number
      This script will reverse the given number.
      For eg. pgm13.sh 1234, will print 4321
user@user-HP-Laptop-15-da0xxx:~/shellspringpgms$ bash pgm13.sh
Enter a number to be reversed :
12345
Reverse number is 54321
user@user-HP-Laptop-15-da0xxx:~/shellspringpgms$ 

```

14. Write a shell script delete a file which has special characters in its file name.

**Ans:**

In Linux or Unix-like system, you may come across file names including the following special characters, white spaces, backslashes and more.

- 
- 
- ;
- &
- \$
- ?
- \*

Bash shell considers most of the above special characters as commands. Thus, the “rm” command may not be able to delete such files. The simplest way to delete files having special characters in its name is by using the inode number

15. Write a shell script to find out the unique words in a file and also count the occurrence of each of these words. We can say that the file under consideration contains many lines, and each line has multiple words.

**Ans:**

```

animal.txt
~/shellscrip...
pgm12.sh      x      animal.txt      x
1 Tiger
2 Lion
3 Deer
4 Bear
5 Elephant
6 Deer|
7

```

**OUTPUT:**

```

user@user-HP-Laptop-15-da0xxx:~/shellscripingpgms$ awk '{for(i=1;i<=NF;i++)a[$i]++;}END{for(i in a){print i, a[i];}}' animal.txt
Tiger 1
Lion 1
Elephant 1
Bear 1
Deer 2
user@user-HP-Laptop-15-da0xxx:~/shellscripingpgms$ 

```

16. Write a script to print the first 10 elements of Fibonacci series.

**Ans :**

```

N=10
a=0
b=1
echo "The Fibonacci series is :"

for (( i=1; i<=N; i++ ))
do
    echo -n "$a "
    fn=$((a + b))
    a=$b
    b=$fn
done

```

**OUTPUT:**

```
user@user-HP-Laptop-15-da0xxx:~/shells scriptingpgms$ bash pgm16.sh
The Fibonacci series is :
0 1 1 2 3 5 8 13 21 34 user@user-HP-Laptop-15-da0xxx:~/shells scriptingpgms$ █
```

17. Write a shell script to get the total count of the word “Linux” in all the “.txt” files and also across files present in subdirectories.

**Ans :**

```
$ find . -name *.txt -exec grep -c Linux '{}' \; | awk '{x+=$0;}END{print x}'
```

18. Write a shell script to validate password strength. Here are a few assumptions for the password string.

Length – minimum of 8 characters. Contain both alphabet and number.

Include both the small and capital case letters.

If the password doesn’t comply with any of the above conditions, then the script should report it as a <Weak Password>.

**Ans :**

```
echo "Enter your password"
read password
len="${#password}"

if test $len -ge 8 ; then
    echo "$password" | grep -q [0-9]
    if test $? -eq 0 ; then
        echo "$password" | grep -q [A-Z]
        if test $? -eq 0 ; then
            echo "$password" | grep -q [a-z]
            if test $? -eq 0 ; then
                echo "Strong Password"
            else
                echo "Weak Password -> Should include a lower case letter."
            fi
        else
            echo "Weak Password -> Should include a capital case letter."
        fi
    else
        echo "Weak Password -> Should use numbers in your password."
    fi
fi
```

```
        fi
else
    echo "Weak Password -> Password length should have at least 8 characters."
fi
```

## OUTPUT:

```
user@user-HP-Laptop-15-da0xxx:~/shellscriptingpgms$ bash pgm18.sh
Enter your password
God@17777
Strong Password
user@user-HP-Laptop-15-da0xxx:~/shellscriptingpgms$ bash pgm18.sh
Enter your password
ganga
Weak Password -> Password length should have at least 8 characters.
user@user-HP-Laptop-15-da0xxx:~/shellscriptingpgms$ bash pgm18.sh
Enter your password
gangaaaa
Weak Password -> Should use numbers in your password.
user@user-HP-Laptop-15-da0xxx:~/shellscriptingpgms$ bash pgm18.sh
Enter your password
ganga786
Weak Password -> Should include a capital case letter.
user@user-HP-Laptop-15-da0xxx:~/shellscriptingpgms$ bash pgm18.sh
Enter your password
Gangaaaa
Weak Password -> Should use numbers in your password.
```

## EXPERIMENT-15

### INSTALLATION OF LAMP ON UBUNTU

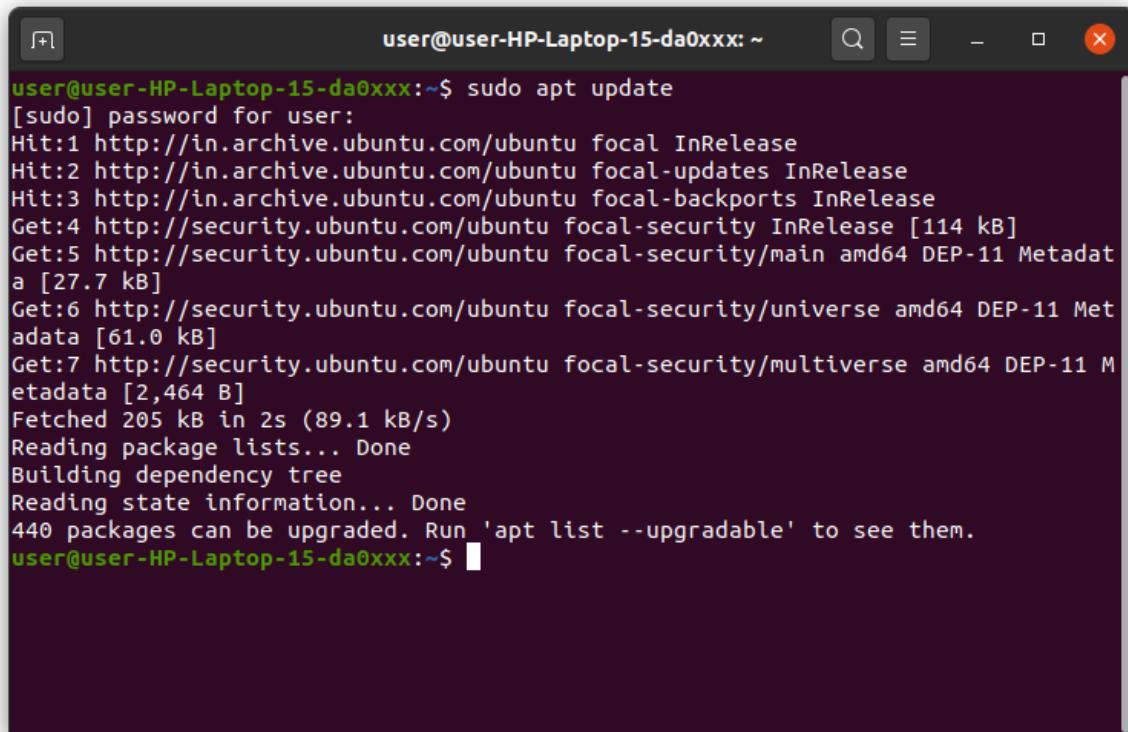
#### Introduction

A “LAMP” stack is a group of open-source software that is typically installed together to enable a server to host dynamic websites and web apps. This term is actually an acronym which represents the Linux operating system, with the Apache web server. The site data is stored in a MySQL database, and dynamic content is processed by PHP

#### Step 1 — Installing Apache and Updating the Firewall

First, make sure your apt cache is updated with:

```
$ sudo apt update
```



The screenshot shows a terminal window with a dark background and light-colored text. The title bar reads "user@user-HP-Laptop-15-da0xxx: ~". The command "sudo apt update" is entered at the prompt. The output shows the process of updating the apt cache from various Ubuntu repositories, including focal, focal-updates, focal-backports, and security channels. It includes details like file sizes and download speeds. The process is completed successfully with a message about 440 upgradeable packages.

```
user@user-HP-Laptop-15-da0xxx:~$ sudo apt update
[sudo] password for user:
Hit:1 http://in.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:5 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [27.7 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 Metadata [61.0 kB]
Get:7 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 DEP-11 Metadata [2,464 B]
Fetched 205 kB in 2s (89.1 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
440 packages can be upgraded. Run 'apt list --upgradable' to see them.
user@user-HP-Laptop-15-da0xxx:~$
```

Once the cache has been updated, you can install Apache with:

```
$ sudo apt install apache2
```

```

user@user-HP-Laptop-15-da0xxx:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0
0 upgraded, 9 newly installed, 0 to remove and 348 not upgraded.
Need to get 1,819 kB of archives.
After this operation, 7,938 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu focal/main amd64 libapr1 amd64 1.6.5-1
ubuntu1 [91.4 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu focal/main amd64 libaprutil1 amd64 1.6
.1-4ubuntu2 [84.7 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu focal/main amd64 libaprutil1-dbd-sqlit
e3 amd64 1.6.1-4ubuntu2 [10.5 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu focal/main amd64 libaprutil1-ldap amd6
4 1.6.1-4ubuntu2 [8,736 B]

```

Adjust the Firewall to Allow Web Traffic

**\$ sudo ufw app list**

```

user@user-HP-Laptop-15-da0xxx:~$ sudo ufw app list
Available applications:
  Apache
  Apache Full
  Apache Secure
  CUPS
user@user-HP-Laptop-15-da0xxx:~$ 

```

If you look at the Apache Full profile details, you'll see that it enables traffic to ports 80 and 443:

**\$ sudo ufw app info "Apache Full"**

```

user@user-HP-Laptop-15-da0xxx:~$ sudo ufw app info "Apache Full"
Profile: Apache Full
Title: Web Server (HTTP,HTTPS)
Description: Apache v2 is the next generation of the omnipresent Apache web
server.

Ports:
  80,443/tcp
user@user-HP-Laptop-15-da0xxx:~$ 

```

To allow incoming HTTP and HTTPS traffic for this server, run:

**sudo ufw allow "Apache Full"**

```
user@user-HP-Laptop-15-da0xxx:~$ sudo ufw allow "Apache Full"
Rules updated
Rules updated (v6)
user@user-HP-Laptop-15-da0xxx:~$
```



## Apache2 Ubuntu Default Page

### It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

### Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   '-- ports.conf
|-- mods-enabled
|   '-- *.load
|       '-- *.conf
|-- conf-enabled
|   '-- *.conf
|-- sites-enabled
|   '-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default configuration, `apache2` needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

### Document Roots

By default, Ubuntu does not allow access through the web browser to any file apart of those located in `/var/www`, `public_html` directories (when enabled) and `/usr/share` (for web applications). If your site is using a web document root located elsewhere (such as in `/srv`) you may need to whitelist your document root directory in `/etc/apache2/apache2.conf`.

The default Ubuntu document root is `/var/www/html`. You can make your own virtual hosts under `/var/www`. This is different to previous releases which provides better security out of the box.

### Reporting Problems

Please use the `ubuntu-bug` tool to report bugs in the Apache2 package with Ubuntu. However, check **existing bug reports** before reporting a new bug.

Please report bugs specific to modules (such as PHP and others) to respective packages, not to the web server itself.

## Step 2 — Installing MySQL

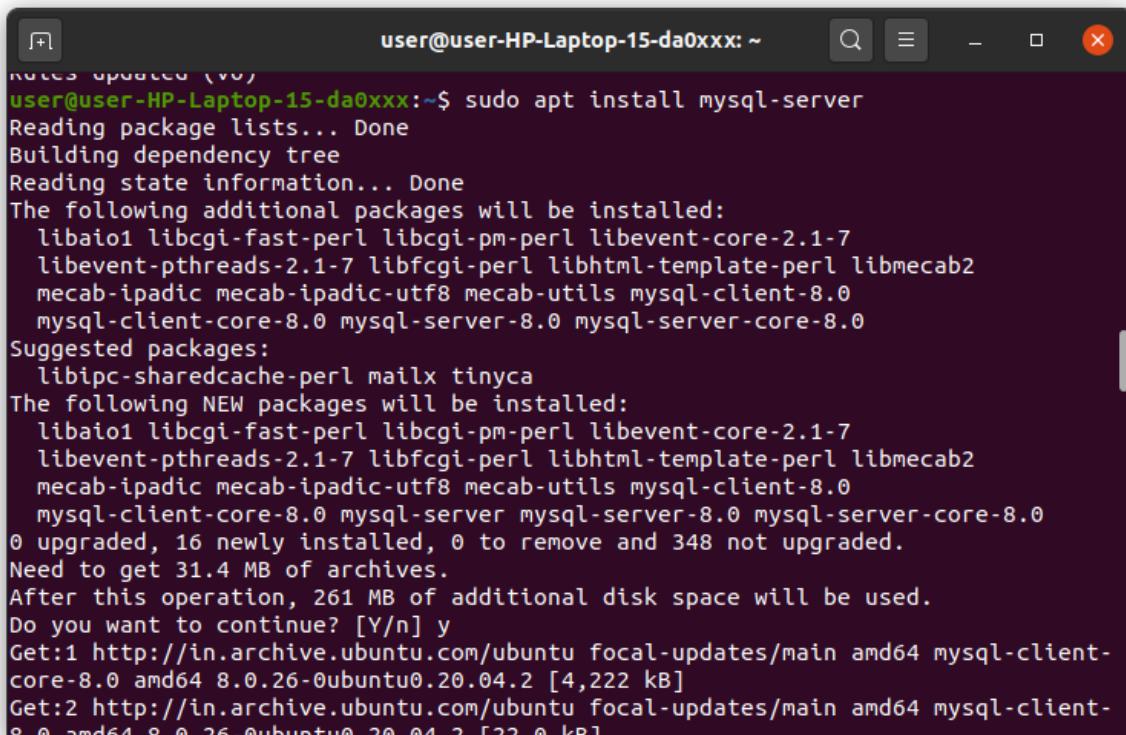
Now that you have your web server up and running, it is time to install MySQL.

MySQL is a database management system. Basically,

it will organize and provide access to databases where your site can store information.

Again, use apt to acquire and install this software:

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



The screenshot shows a terminal window titled "user@user-HP-Laptop-15-da0xxx: ~". The terminal displays the output of the command "sudo apt install mysql-server". The output includes package lists, dependency resolution, suggested packages, and download details. It shows 0 upgraded, 16 newly installed packages, and a total size of 261 MB. The user is prompted with "Do you want to continue? [Y/n] y".

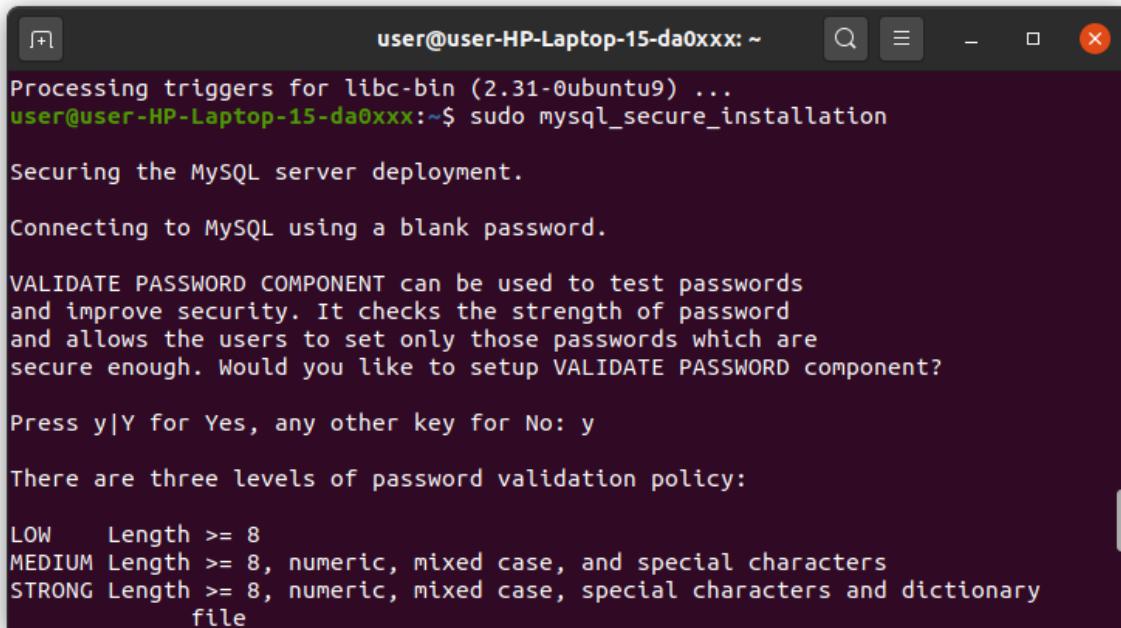
```
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libaio1 libcgi-fast-perl libcgi-pm-perl libevent-core-2.1-7
  libevent-pthreads-2.1-7 libfcgi-perl libhtml-template-perl libmecab2
  mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0
  mysql-client-core-8.0 mysql-server-8.0 mysql-server-core-8.0
Suggested packages:
  libipc-sharedcache-perl mailx tinyca
The following NEW packages will be installed:
  libaio1 libcgi-fast-perl libcgi-pm-perl libevent-core-2.1-7
  libevent-pthreads-2.1-7 libfcgi-perl libhtml-template-perl libmecab2
  mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0
  mysql-client-core-8.0 mysql-server mysql-server-8.0 mysql-server-core-8.0
0 upgraded, 16 newly installed, 0 to remove and 348 not upgraded.
Need to get 31.4 MB of archives.
After this operation, 261 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 mysql-client-core-8.0 amd64 8.0.26-0ubuntu0.20.04.2 [4,222 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 mysql-client-8.0 amd64 8.0.26-0ubuntu0.20.04.2 [4,222 kB]
```

When the installation is complete, run a simple security script that comes pre-installed with MySQL which

will remove some dangerous defaults and lock down access to your database system.

Start the interactive script by running:

```
$ sudo mysql_secure_installation
```



```

user@user-HP-Laptop-15-da0xxx:~$ sudo mysql_secure_installation

Processing triggers for libc-bin (2.31-0ubuntu9) ...
user@user-HP-Laptop-15-da0xxx:~$ sudo mysql_secure_installation

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: y

There are three levels of password validation policy:

LOW      Length >= 8
MEDIUM   Length >= 8, numeric, mixed case, and special characters
STRONG   Length >= 8, numeric, mixed case, special characters and dictionary
          file

```

This will ask if you want to configure the VALIDATE PASSWORD PLUGIN.

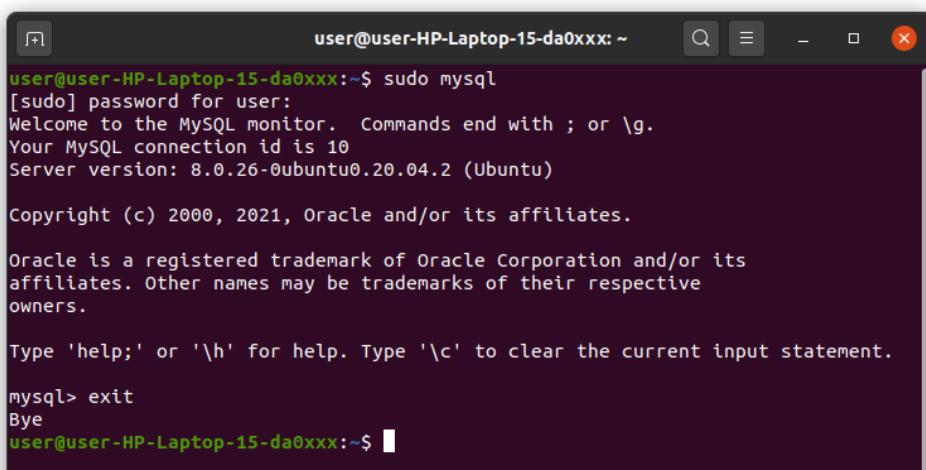
Answer Y for yes, or anything else to continue without enabling

When you're finished, test if you're able to log in to the MySQL console by typing:

**\$ sudo mysql**

To exit the MySQL console, type:

**\$ exit**



```

user@user-HP-Laptop-15-da0xxx:~$ sudo mysql
[sudo] password for user:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.26-0ubuntu0.20.04.2 (Ubuntu)

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

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

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

mysql> exit
Bye
user@user-HP-Laptop-15-da0xxx:~$ 

```

### Step 3 — Installing PHP

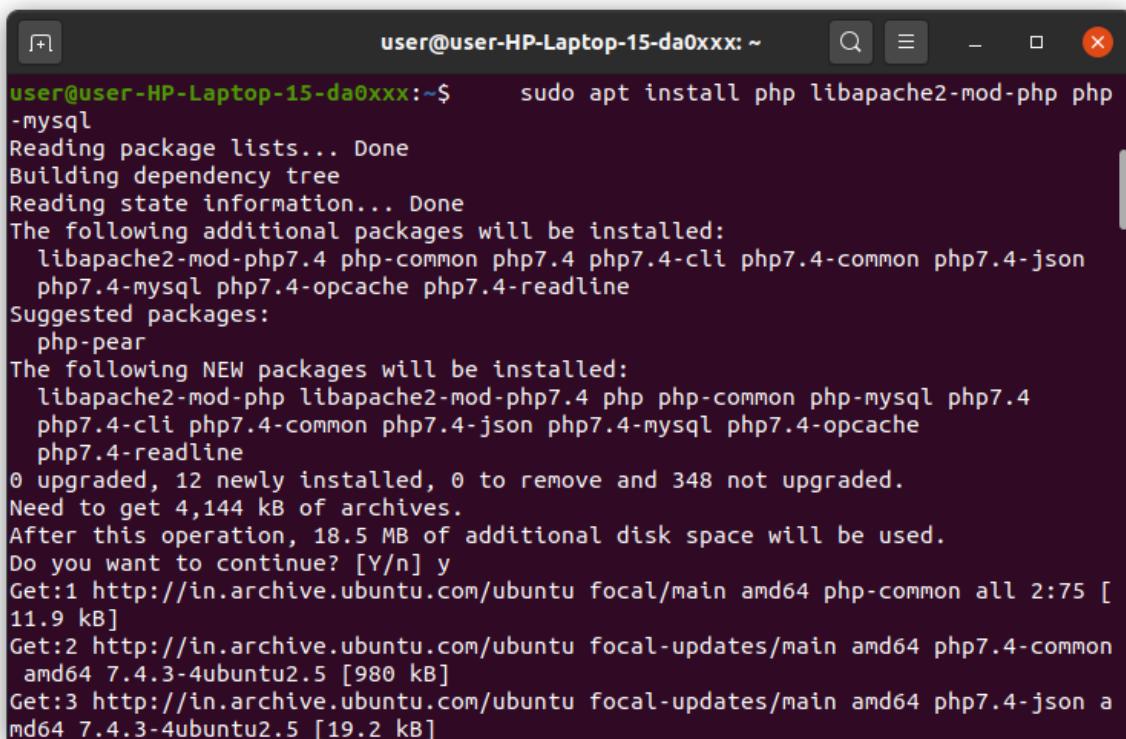
Once again, leverage the apt system to install PHP.

In addition to the php package, you'll also need libapache2-mod-php to integrate PHP into Apache,

and the php-mysql package to allow PHP to connect to MySQL databases.

Run the following command to install all three packages and their dependencies:

```
$ sudo apt install php libapache2-mod-php php-mysql
```



The screenshot shows a terminal window with a dark background and light-colored text. The title bar says "user@user-HP-Laptop-15-da0xxx: ~". The command entered is "sudo apt install php libapache2-mod-php php-mysql". The output shows the package lists being read, dependency trees being built, and state information being checked. It lists packages to be installed (libapache2-mod-php7.4, php-common, etc.) and suggested packages (php-pear). It also lists new packages to be installed (libapache2-mod-php, libapache2-mod-php7.4, etc.). The summary shows 0 upgraded, 12 newly installed, 0 to remove, and 348 not upgraded. It requires 4,144 KB of disk space and will use 18.5 MB after the operation. The user is prompted with "Do you want to continue? [Y/n] y". The final part of the output shows the download of three packages from the archive.ubuntu.com focal/main and focal-updates/main repositories.

This should install PHP without any problems. We'll test this in a moment.

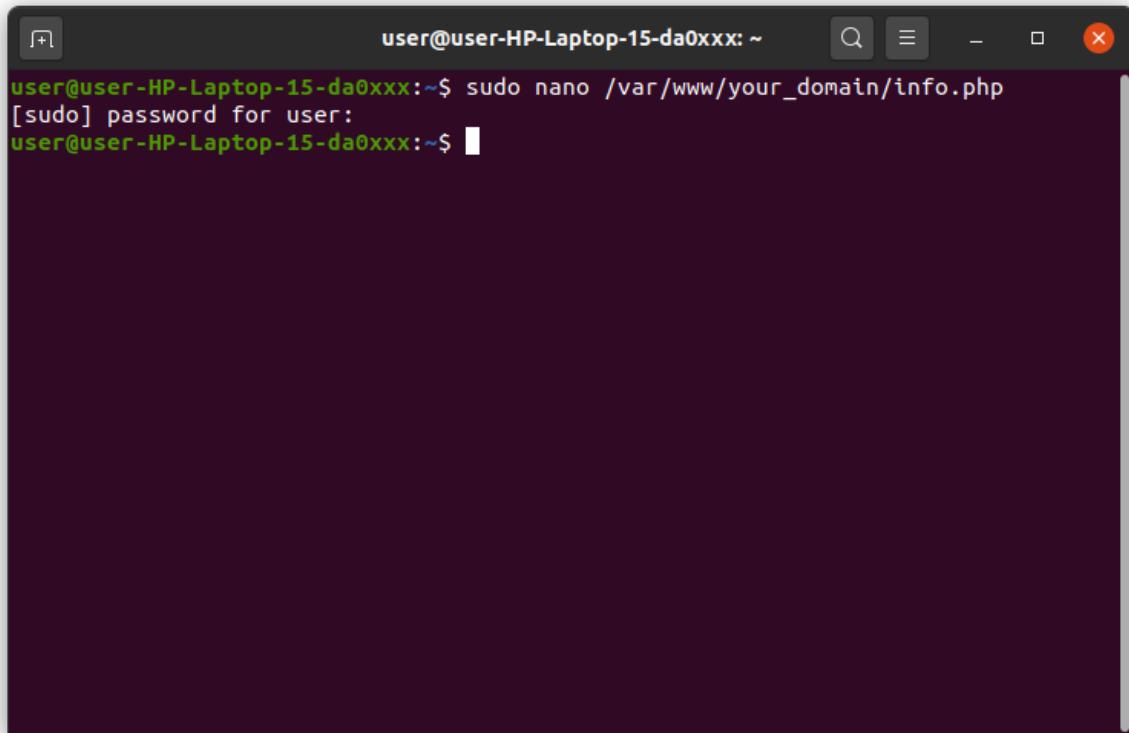
### Step 4 — Testing PHP Processing on your Web Server

In order to test that your system is properly configured for PHP, create a PHP script called info.php.

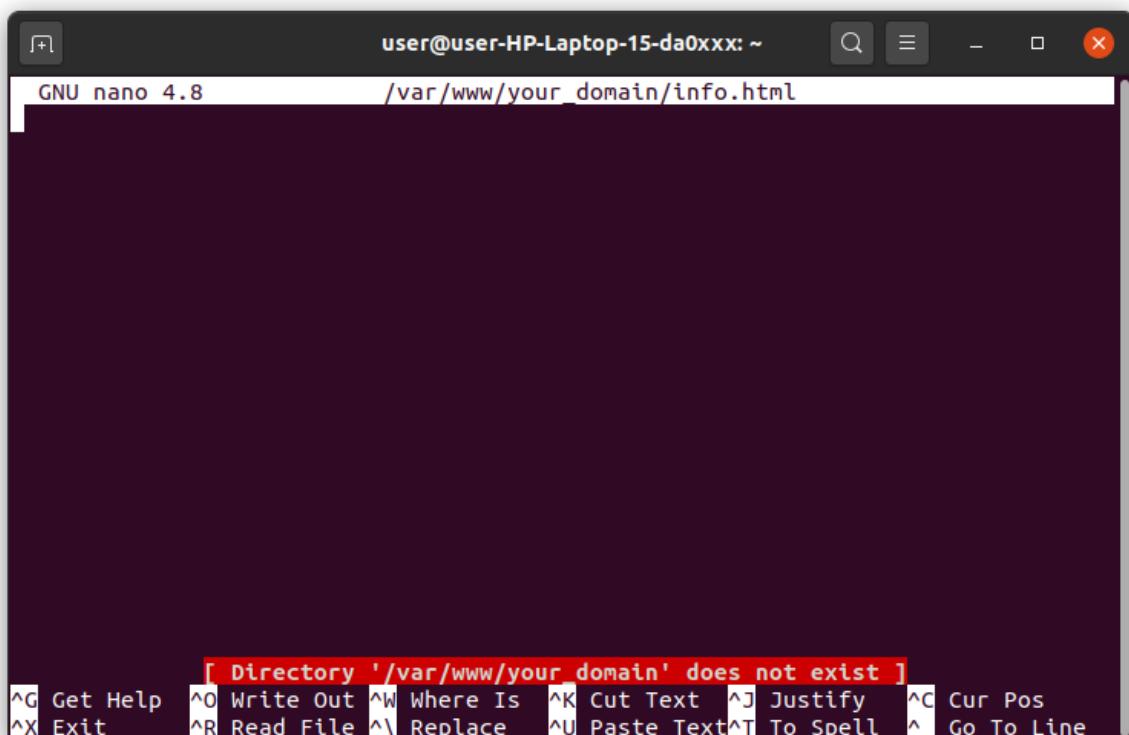
In order for Apache to find this file and serve it correctly, it must be saved to your web root directory.

Create the file at the web root you created in the previous step by running:

```
sudo nano /var/www/your_domain/info.php
```



```
user@user-HP-Laptop-15-da0xxx:~$ sudo nano /var/www/your_domain/info.php
[sudo] password for user:
user@user-HP-Laptop-15-da0xxx:~$
```



```
user@user-HP-Laptop-15-da0xxx:~$ sudo nano /var/www/your_domain/info.html
[ Directory '/var/www/your_domain' does not exist ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line
```

This will open a blank file. Add the following text, which is valid PHP code, inside the file:

```
<?php phpinfo(); ?>
```

The address you will want to visit is:

**http://your\_domain/info.php**

## **EXPERIMENT-16**

### **INSTALLATION OF LARAVEL**

#### **LARAVEL**

- Laravel is an open-source PHP framework, which is robust and easy to understand. It follows a model-view-controller design pattern. Laravel reuses the existing components of different frameworks which helps in creating a web application. The web application thus designed is more structured and pragmatic.
- Laravel offers a rich set of functionalities which incorporates the basic features of PHP frameworks like CodeIgniter, Yii and other programming languages like Ruby on Rails. Laravel has a very rich set of features which will boost the speed of web development.
- If you are familiar with Core PHP and Advanced PHP, Laravel will make your task easier. It saves a lot of time if you are planning to develop a website from scratch. Moreover, a website built in Laravel is secure and prevents several web attacks.

#### **ADVANTAGES OF LARAVEL**

- Laravel offers you the following advantages, when you are designing a web application based on it –
- The web application becomes more scalable, owing to the Laravel framework.
- Considerable time is saved in designing the web application, since Laravel reuses the components from other frameworks in developing web applications.
- It includes namespaces and interfaces, thus help to organize and manage resources.

#### **COMPOSER**

- Composer is a tool for dependency management in PHP. It allows you to declare the libraries your project depends on and it will manage (install/update) them for you.

### **INSTALLATION OF LARAVEL**

- Step 1 – Visit the following URL and download composer to install it on your system.

<https://getcomposer.org/download/>

Remember to set the path of the composer to the php.exe file in the php/xampp/C:

- Step 2 – After the Composer is installed, check the installation by typing the Composer command in the command prompt as shown in the following screenshot.

```

Command Prompt
Microsoft Windows [Version 10.0.19043.1110]
(c) Microsoft Corporation. All rights reserved.

C:\Users\GANGA>composer

Composer version 2.1.6 2021-08-19 17:11:08

Usage:
  command [options] [arguments]

Options:
  -h, --help           Display this help message
  -q, --quiet          Do not output any message
  -V, --version        Display this application version
  --ansi               Force ANSI output
  --no-ansi            Disable ANSI output
  -n, --no-interaction Do not ask any interactive question
  --profile            Display timing and memory usage information
  --no-plugins         Whether to disable plugins.
  -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.
  --no-cache           Prevent use of the cache
  -v|vv|vvv, --verbose Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and
  3 for debug

Available commands:
  
```

- Step 3 – Create a new directory anywhere in your system for your new Laravel project. After that, move to the path where you have created the new directory and type the following command there to install Laravel.

composer create-project --prefer-dist laravel/laravel folder\_name

But the latest version(currently 8.x) just requires you to type

composer create-project laravel/laravel folder\_name

```
C:\Users\GANGA>cd C:\xampp\htdocs
C:\xampp\htdocs> composer create-project laravel/laravel LARAVEL
Creating a "laravel/laravel" project at "./LARAVEL"
Installing laravel/laravel (v8.6.1)
- Downloading laravel/laravel (v8.6.1)
- Installing laravel/laravel (v8.6.1): Extracting archive
Created project in C:/xampp/htdocs/LARAVEL
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 110 installs, 0 updates, 0 removals
- Locking asmb9/stack-cors (v2.0.3)
- Locking brick/math (0.9.3)
- Locking dflydev/dot-access-data (v3.0.1)
- Locking doctrine/inflector (2.0.3)
- Locking doctrine/instantiator (1.4.0)
- Locking doctrine/lexer (1.2.1)
- Locking dragonmantank/cron-expression (v3.1.0)
- Locking egulias/email-validator (2.1.25)
- Locking facade/flare-client-php (1.8.1)
- Locking facade/ignition (2.12.0)
- Locking facade/ignition-contracts (1.0.2)
- Locking fakerphp/faker (v1.15.0)
- Locking filp/whoops (2.14.1)
- Locking fruitcake/laravel-cors (v2.0.4)
- Locking graham-campbell/result-type (v1.0.2)
- Locking guzzlehttp/guzzle (7.3.0)
- Locking guzzlehttp/promises (1.4.1)
```

- Step 4 – The above command will install Laravel in the current directory. Start the Laravel service by executing the following command.

`php artisan serve`

- Step 5 – After executing the above command, you will see a screen as shown below –

```
C:\xampp\htdocs\LARAVEL>php artisan serve
Starting Laravel development server: http://127.0.0.1:8000
[Sun Sep  5 17:10:23 2021] PHP 8.0.2 Development Server (http://127.0.0.1:8000) started
[Sun Sep  5 17:11:22 2021] 127.0.0.1:6289 Accepted
[Sun Sep  5 17:11:22 2021] 127.0.0.1:14095 Accepted
[Sun Sep  5 17:11:25 2021] 127.0.0.1:6289 Closing
[Sun Sep  5 17:11:26 2021] 127.0.0.1:14095 [200]: GET /favicon.ico
[Sun Sep  5 17:11:26 2021] 127.0.0.1:14095 Closing
```

- Step 6 – Copy the URL underlined in gray in the above screenshot and open that URL in the browser. If you see the following screen, it implies Laravel has been installed successfully.

The screenshot shows a web browser window with the URL `127.0.0.1:8000` in the address bar. The page content is the Laravel welcome screen. It includes the Laravel logo, a "Documentation" section with a brief description, a "Laracasts" section with a link to video tutorials, a "Laravel News" section with a brief description, and a "Vibrant Ecosystem" section with a brief description. At the bottom, there are links for "Shop" and "Sponsor", and the text "Laravel v8.58.0 (PHP v8.0.2)".

The screenshot shows a file manager interface with the URL `localhost/laravel/` in the address bar. The current directory is `/laravel`, showing its contents. The files listed include `.env`, `composer.json`, `package.json`, `phpunit.xml`, `public`, `resources`, `routes.php`, `storage`, `tests`, `vendor`, and `webpack.mix.js`. The files were last modified on 2021-09-05 at 17:04, except for `phpunit.xml` which was modified on 2021-08-24 at 08:59.

- Step 7 – The following screen indicates the laravel framework has been successfully installed in your device.

## **EXPERIMENT-17**

### **FAMILIARISATION TO COMMAND LINE TOOL FOR NETWORKING**

#### **NETWORK COMMANDS**

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

1. Ping
2. Hostname
- 3.netstat
4. If Config
5. Nslookup
6. Traceroute
- 7.Route

#### **PING**

- PACKET INTERNET GROPER
- Most widely used utility tool to troubleshoot
- It sends packets of information to the user-defined source. If the packets are received , the destination device sends packets back.
- Ping can be used for two purposes
  - Network connection can be established
  - Speed of the connection
- Ping is used to test a network host capacity to interact with another host. Just enter the command Ping, followed by the target host's name or IP address.

```
user@user-HP-Laptop-15-da0xxx:~$ ping google1.com
ping: google1.com: No address associated with hostname
user@user-HP-Laptop-15-da0xxx:~$ ping google.com
PING google.com(maa05s10-in-x0e.1e100.net (2404:6800:4007:808::200e)) 56 data bytes
64 bytes from maa05s10-in-x0e.1e100.net (2404:6800:4007:808::200e): icmp_seq=1 ttl=117 time=203 ms
64 bytes from maa05s10-in-x0e.1e100.net (2404:6800:4007:808::200e): icmp_seq=2 ttl=117 time=225 ms
64 bytes from maa05s10-in-x0e.1e100.net (2404:6800:4007:808::200e): icmp_seq=3 ttl=117 time=248 ms
64 bytes from maa05s10-in-x0e.1e100.net (2404:6800:4007:808::200e): icmp_seq=4 ttl=117 time=271 ms
64 bytes from maa05s10-in-x0e.1e100.net (2404:6800:4007:808::200e): icmp_seq=5 ttl=117 time=89.1 ms
64 bytes from maa05s10-in-x0e.1e100.net (2404:6800:4007:808::200e): icmp_seq=6 ttl=117 time=48.3 ms
64 bytes from maa05s10-in-x0e.1e100.net (2404:6800:4007:808::200e): icmp_seq=7 ttl=117 time=134 ms
64 bytes from maa05s10-in-x0e.1e100.net (2404:6800:4007:808::200e): icmp_seq=8 ttl=117 time=157 ms
64 bytes from maa05s10-in-x0e.1e100.net (2404:6800:4007:808::200e): icmp_seq=9 ttl=117 time=180 ms
64 bytes from maa05s10-in-x0e.1e100.net (2404:6800:4007:808::200e): icmp_seq=10
```

```
user@user-HP-Laptop-15-da0xxx:~$ ping 192.168.1.5
PING 192.168.1.5 (192.168.1.5) 56(84) bytes of data.
```

## HOSTNAME

- Hostname :displays the machine hostname
- Hostname -f :displays the fully qualified host and domain name
- Hostname -I :displays the ip address for the current machine

## NETSTAT

- Netstat command displays various network related information such as network connections, routing tables, interface statistics, multicast memberships etc.
- netstat -a : To show both listening and non-listening sockets
- netstat -at : To list all tcp ports.
- netstat -au : To list all udp ports.
- netstat -l : To list only the listening ports.
- netstat -lt : To list only the listening tcp ports
- netstat -lu : To list only the listening udp ports.
- netstat -lx : To list only the listening UNIX ports.
- netstat -s : To list the statistics for all ports

```

Processing triggers for man-db (2.9.1-1) ...
user@user-HP-Laptop-15-da0xxx:~$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp6      0      0 user-HP-Laptop-15:54678 2403:8940:3:1::f:http    TIME_WAIT
udp       0      0 user-HP-Laptop-1:bootpc _gateway:bootps          ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type      State           I-Node Path
unix    2      [ ]        DGRAM               38611  /run/user/1000/system
d/notify
unix    2      [ ]        DGRAM               33186  /run/wpa_supplicant/w
lo1
unix    5      [ ]        DGRAM               18532  /run/systemd/notify
unix    2      [ ]        DGRAM               18546  /run/systemd/journal/
syslog
unix   18      [ ]        DGRAM               18556  /run/systemd/journal/
dev-log
unix    7      [ ]        DGRAM               18560  /run/systemd/journal/
socket
unix    3      [ ]        STREAM   CONNECTED     44651
unix    3      [ ]        STREAM   CONNECTED     45557
unix    3      [ ]        STREAM   CONNECTED     46295
unix    3      [ ]        STREAM   CONNECTED     45306
unix    3      [ ]        STREAM   CONNECTED     40940

```

```
+ user@user-HP-Laptop-15-da0xxx:~$ netstat -at
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp     0      0 localhost:33060          0.0.0.0:*
tcp     0      0 localhost:mysql          0.0.0.0:*
tcp     0      0 localhost:domain        0.0.0.0:*
tcp     0      0 localhost:ipp           0.0.0.0:*
tcp6    0      0 [::]:http              [::]:*
tcp6    0      0 ip6-localhost:ipp       [::]:*
user@user-HP-Laptop-15-da0xxx:~$
```

```
+ user@user-HP-Laptop-15-da0xxx:~$ netstat -au
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp     0      0 localhost:domain        0.0.0.0:*
udp     0      0 user-HP-Laptop-1:bootpc _gateway:bootps
udp     0      0 0.0.0.0:43346          0.0.0.0:*
udp     0      0 0.0.0.0:631           0.0.0.0:*
udp     0      0 0.0.0.0:mdns          0.0.0.0:*
udp6    0      0 [::]:45475            [::]:*
udp6    0      0 [::]:mdns             [::]:*
user@user-HP-Laptop-15-da0xxx:~$
```

```
+ user@user-HP-Laptop-15-da0xxx:~$ netstat -l
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp     0      0 localhost:33060          0.0.0.0:*
tcp     0      0 localhost:mysql          0.0.0.0:*
tcp     0      0 localhost:domain        0.0.0.0:*
tcp     0      0 localhost:ipp           0.0.0.0:*
tcp6    0      0 [::]:http              [::]:*
tcp6    0      0 ip6-localhost:ipp       [::]:*
udp     0      0 localhost:domain        0.0.0.0:*
udp     0      0 0.0.0.0:43346          0.0.0.0:*
udp     0      0 0.0.0.0:631           0.0.0.0:*
udp     0      0 0.0.0.0:mdns          0.0.0.0:*
udp6    0      0 [::]:45475            [::]:*
udp6    0      0 [::]:mdns             [::]:*
raw6   0      0 [::]:ipv6-icmp         [::]:*                7
Active UNIX domain sockets (only servers)
Proto RefCnt Flags       Type      State      I-Node Path
unix  2      [ ACC ]     STREAM    LISTENING  42826  @/tmp/.ICE-unix/1765
unix  2      [ ACC ]     STREAM    LISTENING  38614  /run/user/1000/system
d/private
unix  2      [ ACC ]     STREAM    LISTENING  38619  /run/user/1000/bus
unix  2      [ ACC ]     STREAM    LISTENING  38620  /run/user/1000/gnupg/
S.dirmngr
unix  2      [ ACC ]     STREAM    LISTENING  38621  /run/user/1000/gnupg/
S.gpg-agent.browser
unix  2      [ ACC ]     STREAM    LISTENING  38622  /run/user/1000/gnupg/
```

```
user@user-HP-Laptop-15-da0xxx:~$ netstat -lt
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp     0      0 localhost:33060          0.0.0.0:*
tcp     0      0 localhost:mysql          0.0.0.0:*
tcp     0      0 localhost:domain        0.0.0.0:*
tcp     0      0 localhost:ipp           0.0.0.0:*
tcp6    0      0 [::]:http              [::]:*                LISTEN
tcp6    0      0 ip6-localhost:ipp       [::]:*                LISTEN
user@user-HP-Laptop-15-da0xxx:~$
```

```
user@user-HP-Laptop-15-da0xxx:~$ netstat -lu
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp     0      0 localhost:domain        0.0.0.0:*
udp     0      0 0.0.0.0:43346          0.0.0.0:*
udp     0      0 0.0.0.0:631           0.0.0.0:*
udp     0      0 0.0.0.0:mdns          0.0.0.0:*
udp6    0      0 [::]:45475            [::]:*                LISTEN
udp6    0      0 [::]:mdns            [::]:*                LISTEN
user@user-HP-Laptop-15-da0xxx:~$
```

If config

- The if config commands is used for displaying current network configuration information , setting up an ip address , netmask or broadcast address to an network interface , creating an alias for network interface , setting up hardware address and enable or disable network interface
- ifconfig –a :This option is used to display all the interfaces available, even if they are down.
- ifconfig -s : Display a short list, instead of details
- ifconfig interface up :This option is used to activate the driver for the given interface
- ifconfig interface down :This option is used to deactivate the driver for the given interface.

```
user@user-HP-Laptop-15-da0xxx:~$ ifconfig -a
eno1: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether 84:a9:3e:9f:3b:35 txqueuelen 1000 (Ethernet)
      RX packets 0 bytes 0 (0.0 B)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 0 bytes 0 (0.0 B)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
      inet6 ::1 prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
          RX packets 201 bytes 19612 (19.6 KB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 201 bytes 19612 (19.6 KB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlo1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.43.12 netmask 255.255.255.0 broadcast 192.168.43.255
      inet6 fe80::18ac:21dc:7318:8f81 prefixlen 64 scopeid 0x20<link>
        inet6 2409:4073:99:8044:cc07:b5e:a73f:469d prefixlen 64 scopeid 0x0<gl
obal>
    inet6 2409:4073:99:8044:4ca4:5d09:c50f:2442 prefixlen 64 scopeid 0x0<g
lobal>
      ether d0:c5:d3:78:e6:b7 txqueuelen 1000 (Ethernet)
        RX packets 18304 bytes 24942781 (24.9 MB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 9100 bytes 1030172 (1.0 MB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

user@user-HP-Laptop-15-da0xxx:~$
```

```
user@user-HP-Laptop-15-da0xxx:~$ ifconfig -s
Iface      MTU     RX-OK RX-ERR RX-DRP RX-OVR     TX-OK TX-ERR TX-DRP TX-OVR Flg
eno1      1500        0     0 0       0 0       0 0 0 0 BMU
lo        65536      210     0 0       210 0       0 0 0 0 LRU
wlo1      1500    18345     0 0       9149 0       0 0 0 0 BMRU
user@user-HP-Laptop-15-da0xxx:~$
```

## Ip config

- Displays all current TCP/IP network configuration values and refreshes Dynamic Host Configuration Protocol (DHCP) and Domain Name System (DNS) settings.
- Used without parameters, ipconfig displays Internet Protocol version 4 (IPv4) and IPv6 addresses, subnet mask, and default gateway for all adapters.

```
C:\Users\GANGA>ipconfig
Windows IP Configuration

Ethernet adapter Ethernet:
  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . :

Ethernet adapter VirtualBox Host-Only Network:
  Connection-specific DNS Suffix . :
  Link-local IPv6 Address . . . . . : fe80::947f:f324:b8f:f99%4
  IPv4 Address. . . . . : 192.168.56.1
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . :

Wireless LAN adapter Local Area Connection* 11:
  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . :

Wireless LAN adapter Local Area Connection* 12:
  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . :

Wireless LAN adapter WiFi:
```

```
C:\Users\GANGA>ipconfig -a
Error: unrecognized or incomplete command line.

USAGE:
  ipconfig [/allcompartments] [/? | /all |
    /renew [adapter] | /release [adapter] |
    /renew6 [adapter] | /release6 [adapter] |
    /flushdns | /displaydns | /registerdns |
    /showclassid adapter [classid] |
    /setclassid adapter [classid] |
    /setclassid6 adapter [classid] ]

where
  adapter      Connection name
              (wildcard characters * and ? allowed, see examples)

  Options:
    /?           Display this help message
    /all         Display full configuration information.
    /release    Release the IPv4 address for the specified adapter.
    /release6   Release the IPv6 address for the specified adapter.
    /renew      Renew the IPv4 address for the specified adapter.
    /renew6     Renew the IPv6 address for the specified adapter.
    /flushdns  Purges the DNS Resolver cache.
    /registerdns Refreshes all DHCP leases and re-registers DNS names
    /displaydns Display the contents of the DNS Resolver Cache.
    /showclassid Displays all the dhcp class IDs allowed for adapter.
```

## Nslookup

- Nslookup (stands for “Name Server Lookup”) is a useful command for getting information from a 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.

- nslookup google.com :nslookup followed by the domain name will display the “A Record” (IP Address) of the domain. Use this command to find the address record for a domain. It queries domain name servers and gets the details.
- nslookup 192.168.0.10 : Reverse DNS lookup we can also do the reverse DNS look-up by providing the IP Address as argument to nslookup.

command	Used for
<b>nslookup -type=any google.com</b>	Lookup for any record
<b>nslookup -type=soa redhat.com</b>	Lookup for an soa record(start of authority)
<b>nslookup -type=ns google.com</b>	Lookup for an ns record(Name Server)
<b>nslookup -type=mx google.com</b>	Lookup for an mx record(Mail Exchange)
<b>nslookup -type=txt google.com</b>	Lookup for an txt record

```
+ user@user-HP-Laptop-15-da0xxx:~$ nslookup google.com
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 216.58.200.142
Name:   google.com
Address: 2404:6800:4007:80f::200e

user@user-HP-Laptop-15-da0xxx:~$ nslookup 10.0.2.15
** server can't find 15.2.0.10.in-addr.arpa: NXDOMAIN

user@user-HP-Laptop-15-da0xxx:~$ 
```

```
+ user@user-HP-Laptop-15-da0xxx:~$ nslookup -type=any google.com
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 216.58.200.142
Name:   google.com
Address: 2404:6800:4007:80f::200e
google.com    mail exchanger = 20 alt1.aspmx.l.google.com.
google.com    mail exchanger = 10 aspmx.l.google.com.
google.com    mail exchanger = 30 alt2.aspmx.l.google.com.
google.com    mail exchanger = 50 alt4.aspmx.l.google.com.
google.com    mail exchanger = 40 alt3.aspmx.l.google.com.
google.com
          origin = ns1.google.com
          mail addr = dns-admin.google.com
          serial = 394833383
          refresh = 900
          retry = 900
          expire = 1800
```

```
user@user-HP-Laptop-15-da0xxx:~$ nslookup -type=soa redhat.com
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
redhat.com
    origin = a1-68.akam.net
    mail addr = noc.redhat.com
    serial = 2021090302
    refresh = 300
    retry = 180
    expire = 604800
    minimum = 14400

Authoritative answers can be found from:

user@user-HP-Laptop-15-da0xxx:~$
```

```
user@user-HP-Laptop-15-da0xxx:~$ nslookup -type=ns google.com
Server:      127.0.0.53
Address:     127.0.0.53#53

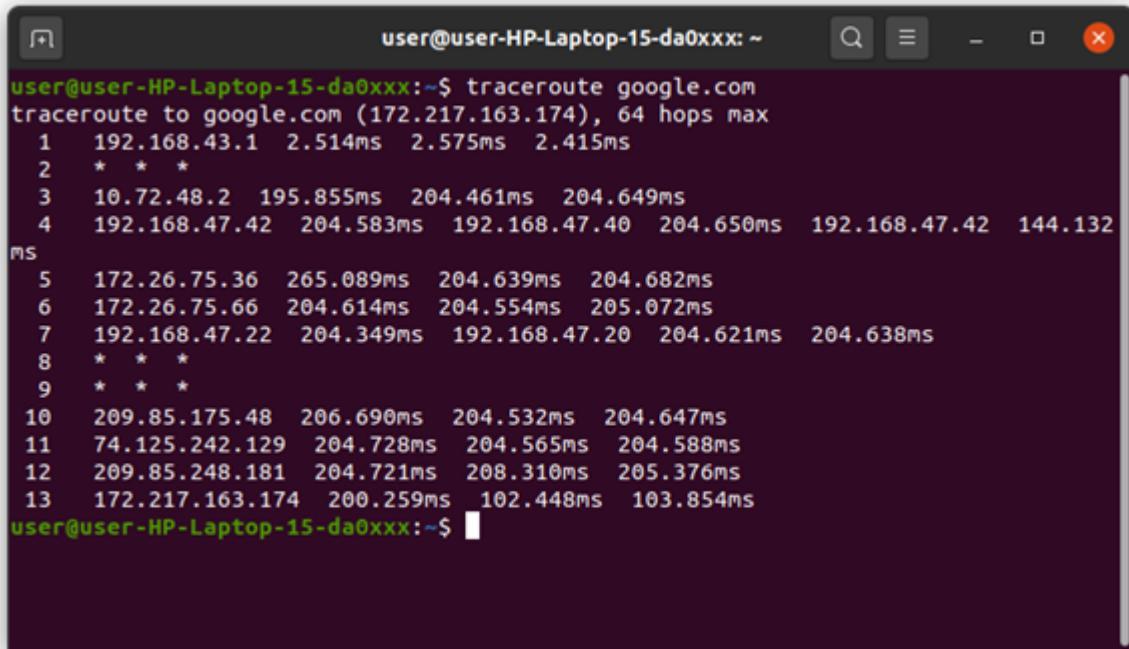
Non-authoritative answer:
google.com      nameserver = ns4.google.com.
google.com      nameserver = ns1.google.com.
google.com      nameserver = ns3.google.com.
google.com      nameserver = ns2.google.com.

Authoritative answers can be found from:

user@user-HP-Laptop-15-da0xxx:~$
```

## Traceroute

- Traceroute command prints the route that a packet takes to reach the host. This command is useful when we want to know about the route and about all the hops that a packet takes



```
user@user-HP-Laptop-15-da0xxx:~$ traceroute google.com
traceroute to google.com (172.217.163.174), 64 hops max
 1  192.168.43.1  2.514ms  2.575ms  2.415ms
 2  * * *
 3  10.72.48.2  195.855ms  204.461ms  204.649ms
 4  192.168.47.42  204.583ms  192.168.47.40  204.650ms  192.168.47.42  144.132
ms
 5  172.26.75.36  265.089ms  204.639ms  204.682ms
 6  172.26.75.66  204.614ms  204.554ms  205.072ms
 7  192.168.47.22  204.349ms  192.168.47.20  204.621ms  204.638ms
 8  * * *
 9  * * *
10  209.85.175.48  206.690ms  204.532ms  204.647ms
11  74.125.242.129  204.728ms  204.565ms  204.588ms
12  209.85.248.181  204.721ms  208.310ms  205.376ms
13  172.217.163.174  200.259ms  102.448ms  103.854ms
user@user-HP-Laptop-15-da0xxx:~$
```

command	Used for
\$ traceroute -g 192.168.43.45 google.com	Route the packet through gate
\$traceroute -m 5 google.com	Set the max number of hops for the packet to reach the destination. Default value is 30.
\$traceroute -n google.com	Do not resolve IP addresses to their domain names
\$traceroute -p 20292 google.com	Set the destination port to use. Default is 33434

## Route

- Route command in Linux is used when you want to work with the IP/kernel routing table. It is mainly used to set up static routes to specific hosts or networks via an interface. It is used for showing or updating the IP/kernel routing table.
- install net-tools by \$sudo apt-get install net-tools

```
user@user-HP-Laptop-15-da0xxx:~$ route
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref    Use Iface
default         _gateway       0.0.0.0        UG    600    0        0 wlo1
link-local      0.0.0.0        255.255.0.0   U     1000   0        0 wlo1
192.168.43.0   0.0.0.0        255.255.255.0 U     600    0        0 wlo1
user@user-HP-Laptop-15-da0xxx:~$
```

\$route	To display the IP/kernel routing table
\$route -n	To display routing table in full numeric form
\$sudo route add default gw 169.254.0.0	To add a default gateway.
\$route -Cn	To list kernel's routing cache information.
\$sudo route add -host 192.168.1.51 reject	. To reject routing to a particular host or network
\$ip route	To get details of the kernel/IP routing table using ip command
\$route del default	To delete the default gateway.
\$ip route show table local	To get the details of the local table with destination addresses assigned to the localhost.

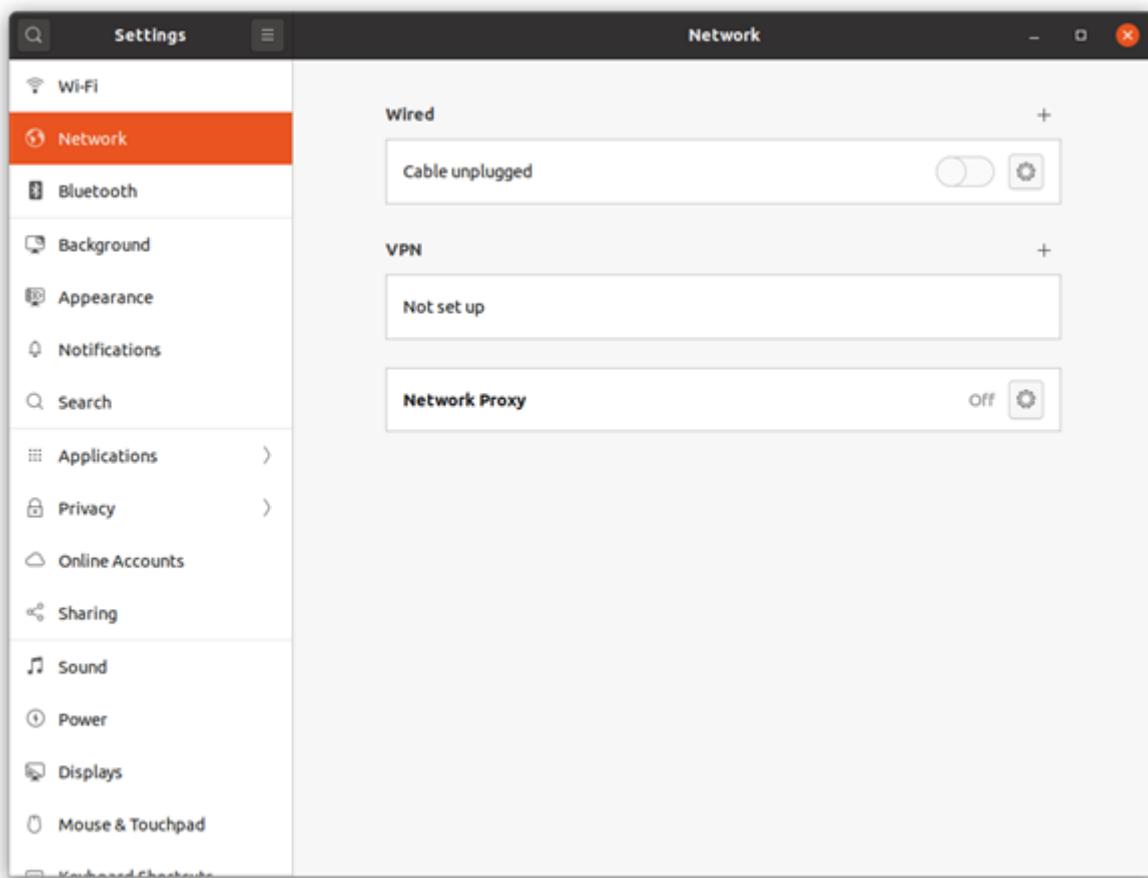
## STATIC AND DYNAMIC IP

- A fixed IP address is called static IP address , i.e. it never changes.

- It is required to set up an Ubuntu static IP address in order to access a device remotely and without losing a connection over the network.
- It is used to connect to an IP camera, home file server, game server, and many other devices.
- A static IP address is necessary only for servers and not for personal PCs

## STEPS FOR SETTING STATIC IP

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.



- Select manual and enter your desired IP address, netmask, gateway and DNS settings.
- Once ready click the Apply button.
- Turn OFF and ON switch to apply your new network static IP configuration settings.

- Click on the network settings icon once again to confirm your new static IP address settings.



## DYNAMIC IP

- A dynamic IP address as its name suggests is a temporary IP address assigned by a DHCP server for every new network.
- A dynamic IP address is used due to the shortage of IP addresses on IPV4.
- A single dynamic IP address can be used between many devices

*Configuring a dynamic ip address*

Step1: type the command in the terminal

```
sudo nano /etc/netplan/01-network-manager-all.yaml
```

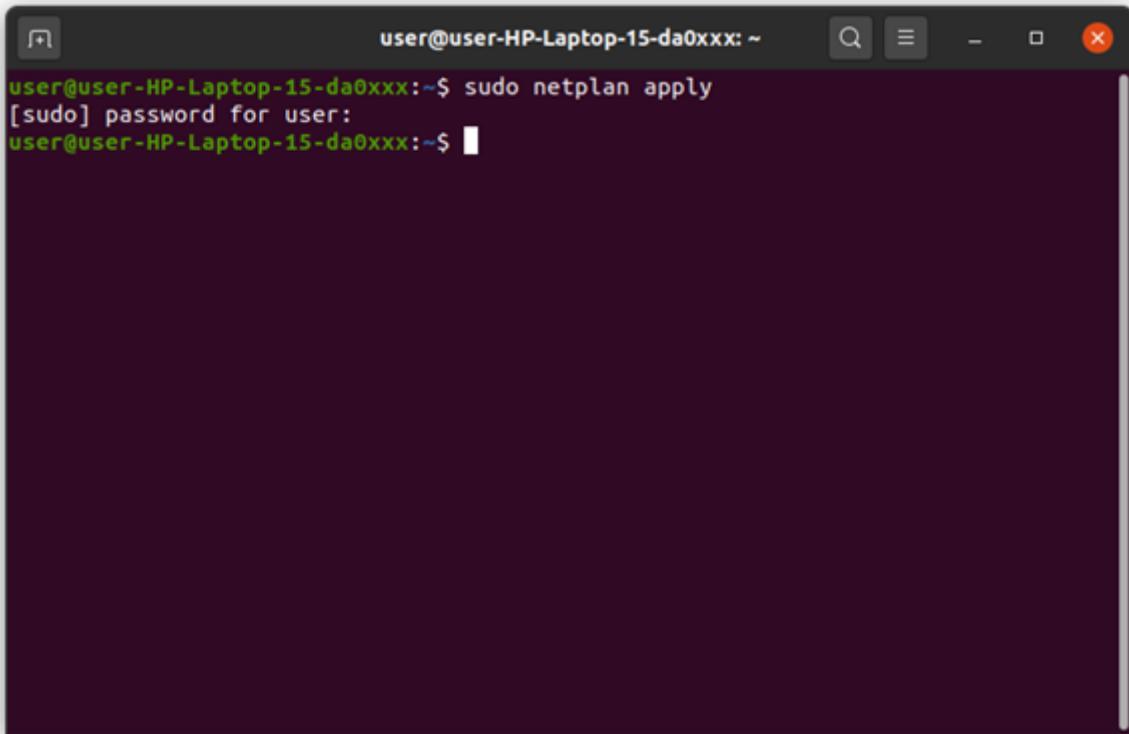
Step2:Now find the name of the network interface you want to configure and insert the following lines:

- dhcp4: yes
- dhcp6: yes

```
user@user-HP-Laptop-15-da0xxx: ~
GNU nano 4.8      /etc/netplan/01-network-manager-all.yaml
Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
```

[ Read 4 lines ]  
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos  
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^ ^ Go To Line

Step3:Apply the changes with sudo netplan apply command



A screenshot of a terminal window titled "user@user-HP-Laptop-15-da0xxx: ~". The window shows the command "sudo netplan apply" being run, followed by a password prompt "[sudo] password for user:". The terminal has a dark background and light-colored text.

## SETTING UP A FIREWALL FOR LAN

- The default firewall configuration tool for Ubuntu is ufw.
- Developed to ease iptables firewall configuration.
- ufw provides a user-friendly way to create an IPv4 or IPv6 host-based firewall.
- ufw by default is initially disabled.
- ufw is not intended to provide complete firewall functionality via its command interface, but instead provides an easy way to add or remove simple rules.
- It is currently mainly used for host-based firewalls.”
- To ENABLE ufw

Step1:check current firewall status

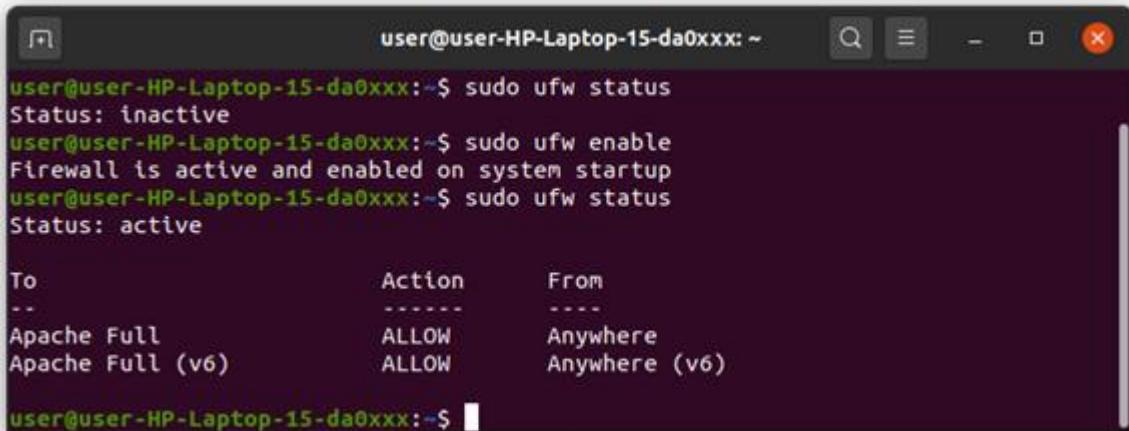
```
sudo ufw status
```

Step2:to enable firewall

```
sudo ufw enable
```

Step3:to check status

```
sudo ufw status
```



```
user@user-HP-Laptop-15-da0xxx:~$ sudo ufw status
Status: inactive
user@user-HP-Laptop-15-da0xxx:~$ sudo ufw enable
Firewall is active and enabled on system startup
user@user-HP-Laptop-15-da0xxx:~$ sudo ufw status
Status: active

To                         Action      From
--                         --          --
Apache Full                 ALLOW       Anywhere
Apache Full (v6)             ALLOW       Anywhere (v6)

user@user-HP-Laptop-15-da0xxx:~$
```

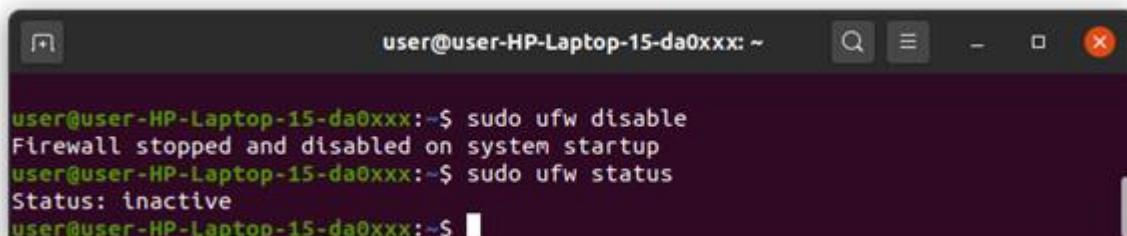
- To Disable ufw

Step1:To disable firewall

```
sudo ufw disable
```

Step2:to check status

```
sudo ufw status
```



```
user@user-HP-Laptop-15-da0xxx:~$ sudo ufw disable
Firewall stopped and disabled on system startup
user@user-HP-Laptop-15-da0xxx:~$ sudo ufw status
Status: inactive
user@user-HP-Laptop-15-da0xxx:~$
```

## **EXPERIMENT-18**

### **FAMILIARISATION OF STATIC AND DYNAMIC IP**

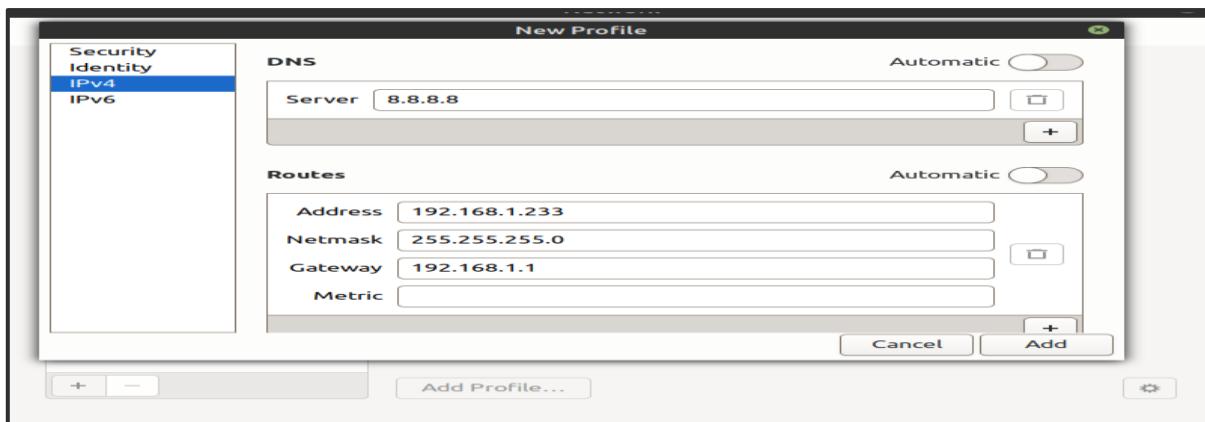
#### **STATIC IP:**

1. A fixed IP address is called static IP address , i.e. it never changes.
2. It is required to set up an Ubuntu static IP address in order to access a device remotely and without losing a connection over the network.
3. It is used to connect to an IP camera, home file server, game server, and many other devices.
4. A static IP address is necessary only for servers and not for personal PCs.

#### **STEPS FOR SETTING STATIC IP:**

1. 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.
2. Click on the settings icon to start IP address configuration Select IPv4 tab.
3. Select manual and enter your desired IP address, netmask, gateway and DNS settings.
4. Once ready click Apply button.
5. Turn OFF and ON switch to apply your new network static IP configuration settings.
6. Click on the network settings icon once again to confirm your new static IP address settings.





## DYNAMIC IP

1. A dynamic IP address as its name suggests is a temporary IP address assigned by a DHCP server for every new network.
2. A dynamic IP address is used due to the shortage of IP addresses on IPV4.
3. A single dynamic IP address can be used between many devices.

Configuring a dynamic ip address

**Step1:** type the command in the terminal

```
sudo nano /etc/netplan/01-network-manager-all.yaml
```

**Step2:** Now find the name of the network interface you want to configure and insert the following lines:

dhcp4: yes

dhcp6: yes

**Step3:** Apply the changes with sudo netplan apply command.

**EXPERIMENT-19****CONCEPT OF SUBNETS AND CIDR ADDRESS SCHEME****SUBNETS:**

The process of dividing a network into smaller network sections is called subnetting. This can be useful for many different purposes and helps isolate groups of hosts from each other to deal with them more easily. By default, each network has only one subnet, which contains all of the host addresses defined within. A netmask is basically a specification of the amount of address bits that are used for the network portion. A subnet mask is another netmask within used to further divide the network.

Each bit of the address that is considered significant for describing the network should be represented as a “1” in the netmask. For instance, the address we discussed above, 192.168.0.15 can be expressed like this, in binary:

1100 0000 - 1010 1000 - 0000 0000 - 0000 1111

As we described above, the network portion for class C addresses is the first 3 octets, or the first 24 bits. Since these are the significant bits that we want to preserve, the netmask would be:

1111 1111 - 1111 1111 - 1111 1111 - 0000 0000

This can be written in the normal IPv4 format as 255.255.255.0. Any bit that is a “0” in the binary representation of the netmask is considered part of the host portion of the address and can be variable. The bits that are “1” are static, however, for the network or subnetwork that is being discussed. We determine the network portion of the address by applying a bitwise AND operation to between the address and the netmask. A bitwise AND operation will save the networking portion of the address and discard the host portion. The result of this on our above example that represents our network is:

1100 0000 - 1010 1000 - 0000 0000 - 0000 0000

This can be expressed as 192.168.0.0. The host specification is then the difference between these

original value and the host portion. In our case, the host is 0000 1111 or 15. The idea of subnetting is to take a portion of the host space of an address, and use it as an additional networking specification to divide the address space again. For instance, a netmask of 255.255.255.0 as we saw above leaves us with 254 hosts in the network (you cannot end in 0 or 255 because these are reserved).

So, continuing with our example, the networking portion is:

1100 0000 - 1010 1000 - 0000 0000

The host portion is:

0000 1111

We can use the first bit of our host to designate a subnetwork. We can do this by adjusting the subnet mask from this:

1111 1111 - 1111 1111 - 1111 1111 - 0000 0000

To this:

1111 1111 - 1111 1111 - 1111 1111 - 1000 0000

In traditional IPv4 notation, this would be expressed as 192.168.0.128. What we have done here is to designate the first bit of the last octet as significant in addressing the network. This effectively produces two subnetworks. The first subnetwork is from 192.168.0.1 to 192.168.0.127. The second subnetwork contains the hosts 192.168.0.129 to 192.168.0.255.

## **CIDR NOTATION:**

A system called Classless Inter-Domain Routing, or CIDR, was developed as an alternative to traditional subnetting. For example, we could express the idea that the IP address 192.168.0.15 is associated with the netmask 255.255.255.0 by using the CIDR notation of 192.168.0.15/24. This means that the first 24 bits of the IP address given are considered significant for the network routing.

This allows us some interesting possibilities. We can use these to reference “supernets”. In this case, we mean a more inclusive address range that is not possible with a traditional subnet mask. For instance, in a class C network, like above, we could not combine the addresses from the networks 192.168.0.0 and 192.168.1.0 because the netmask for class C addresses is 255.255.255.0. However, using CIDR notation, we can combine these blocks by referencing this chunk as 192.168.0.0/23. This specifies that there are 23 bits used for the network portion that we are referring to. So the first network (192.168.0.0) could be represented like this in binary:

1100 0000 - 1010 1000 - 0000 0000 - 0000 0000

While the second network (192.168.1.0) would be like this:

1100 0000 - 1010 1000 - 0000 0001 - 0000 0000

The CIDR address we specified indicates that the first 23 bits are used for the network block we are referencing. This is equivalent to a netmask of 255.255.254.0, or:

1111 1111 - 1111 1111 - 1111 1110 - 0000 0000

As you can see, with this block the 24th bit can be either 0 or 1 and it will still match, because the network block only cares about the first 23 digits. CIDR allows us more control over addressing continuous blocks of IP addresses. This is much more useful than the subnetting we talked about originally.

## **EXPERIMENT-20**

### **CONCEPT OF SUBNET MASK**

The subnet mask is used by the TCP/IP protocol to determine whether a host is on the local subnet or on a remote network. In TCP/IP, the parts of the IP address that are used as the network and host addresses aren't fixed. Unless you have more information, the network and host addresses above can't be determined. This information is supplied in another 32-bit number called a subnet mask. The subnet mask is 255.255.255.0 in this example. It isn't obvious what this number means unless you know 255 in binary notation equals 11111111. So, the subnet mask is 11111111.11111111.11111111.00000000.

Lining up the IP address and the subnet mask together, the network, and host portions of the address can be separated:

11000000.10101000.01111011.10000100 - IP address (192.168.123.132)

11111111.11111111.11111111.00000000 - Subnet mask (255.255.255.0)

The first 24 bits (the number of ones in the subnet mask) are identified as the network address. The last 8 bits (the number of remaining zeros in the subnet mask) are identified as the host address. It gives you the following addresses:

11000000.10101000.01111011.00000000 - Network address (192.168.123.0)

00000000.00000000.00000000.10000100 - Host address (000.000.000.132)

So now you know, for this example using a 255.255.255.0 subnet mask, that the network ID is 192.168.123.0, and the host address is 0.0.0.132. When a packet arrives on the 192.168.123.0 subnet (from the local subnet or a remote network), and it has a destination address of 192.168.123.132, your computer will receive it from the network and process it. Almost all decimal subnet masks convert to binary numbers that are all ones on the left and all zeros on the right.

Some other common subnet masks are:

Decimal Binary 255.255.255.192 1111111.1111111.1111111.11000000 255.255.255.224

1111111.1111111.1111111.11100000

Internet RFC 1878 (available from InterNIC-Public Information Regarding Internet Domain Name Registration Services) describes the valid subnets and subnet masks that can be used on TCP/IP networks

## **EXPERIMENT-21**

### **SETTING UP OF FIREWALL ON LAN**

#### **ufw-uncomplicated firewall:**

- 1) The default firewall configuration tool for Ubuntu is ufw.
- 2) Developed to ease iptables firewall configuration.
- 3) ufw provides a user-friendly way to create an IPv4 or IPv6 host-based firewall.
- 4) ufw by default is initially disabled.
- 5) ufw is not intended to provide complete firewall functionality via its command interface, but instead provides an easy way to add or remove simple rules.
- 6) It is currently mainly used for host-based firewalls.

#### **To Enable uf:**

- **Step1:**check current firewall status

```
sudo ufw status
```

- **Step2:**to enable firewall

```
sudo ufw enable
```

- **Step3:**to check status

```
sudo ufw status
```

#### **To Disable ufw:**

- **Step1:**To disable firewall

```
sudo ufw disable
```

- **Step2:**to check status

```
sudo ufw status
```

## **EXPERIMENT-22**

### **WIRESHARK AND TCPDUMP**

#### **WHAT IS WIRESHARK?**

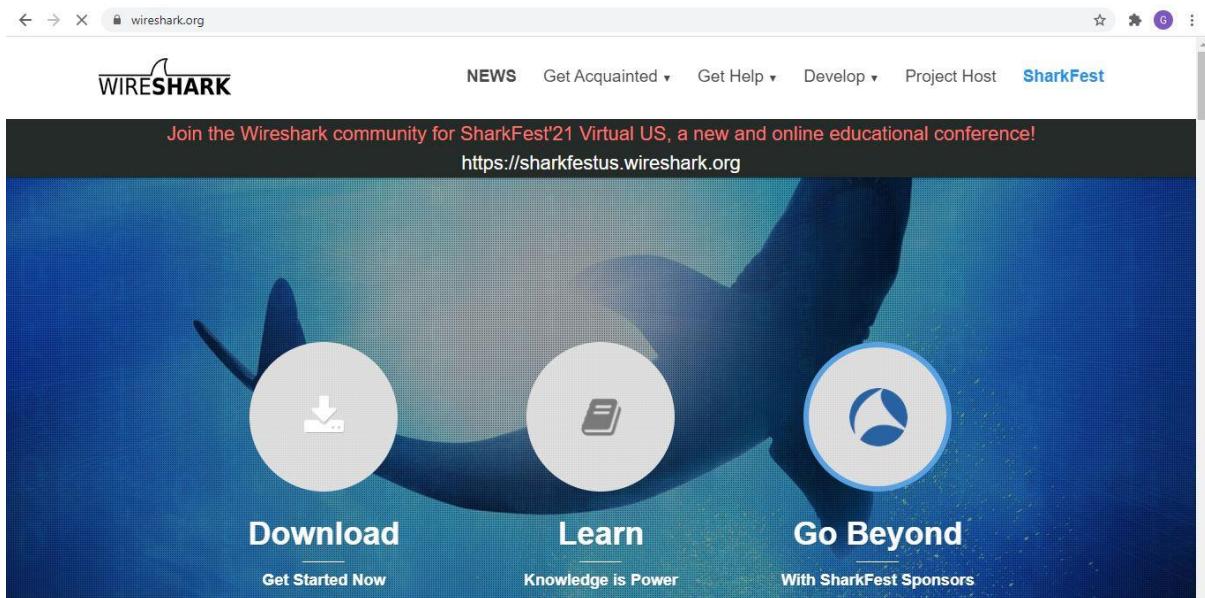
- Network packet protocol analyzer
- A network packet analyzer will try to capture network packets and try to display that packet data as detailed as possible.
- One of the best open source packet analyzers available today for UNIX and Windows

#### **WHERE IT USE ?**

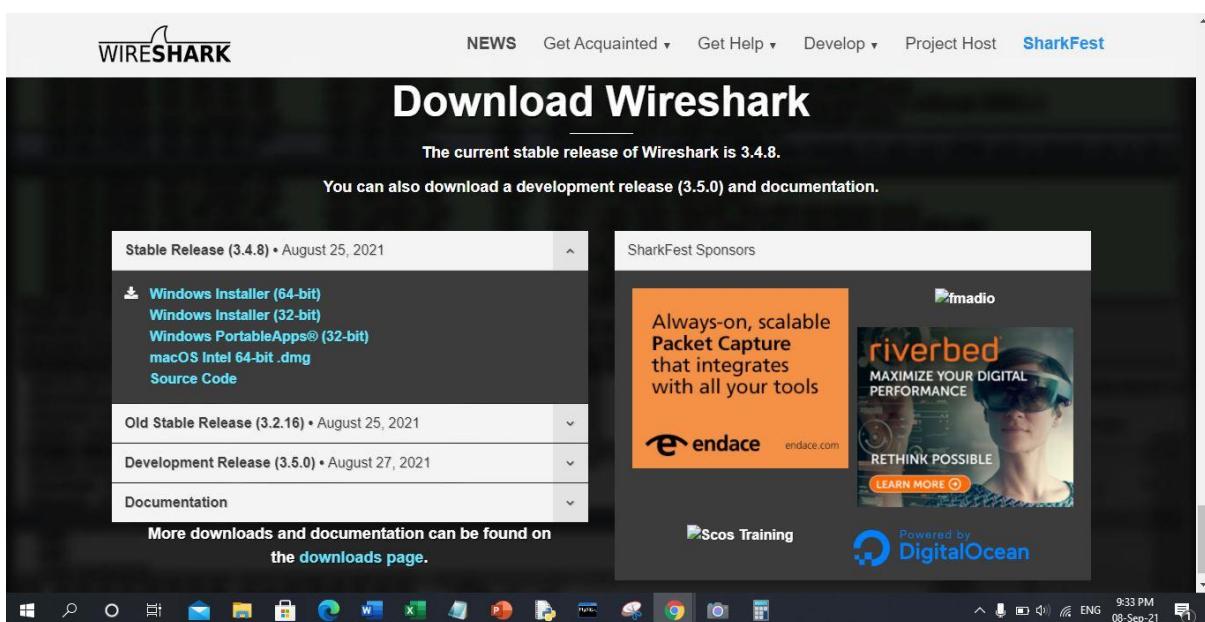
- Network administrators use it to troubleshoot network problems
- Network security engineers use it to examine security problems
- Developers use it to debug protocol implementations
- Testers use it to detect defects
- People use it to learn network protocol internals

#### **STEPS TO INSTALL WIRESHARK**

Step1: Go to [www.wireshark.org](http://www.wireshark.org) ->download 64-bit package

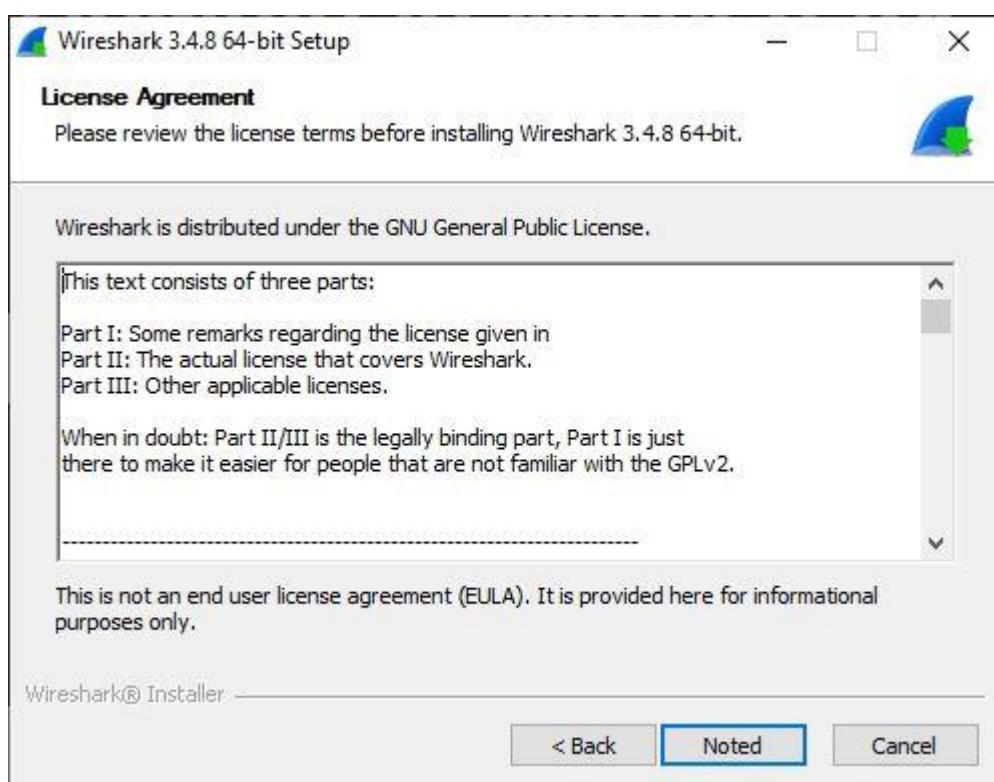
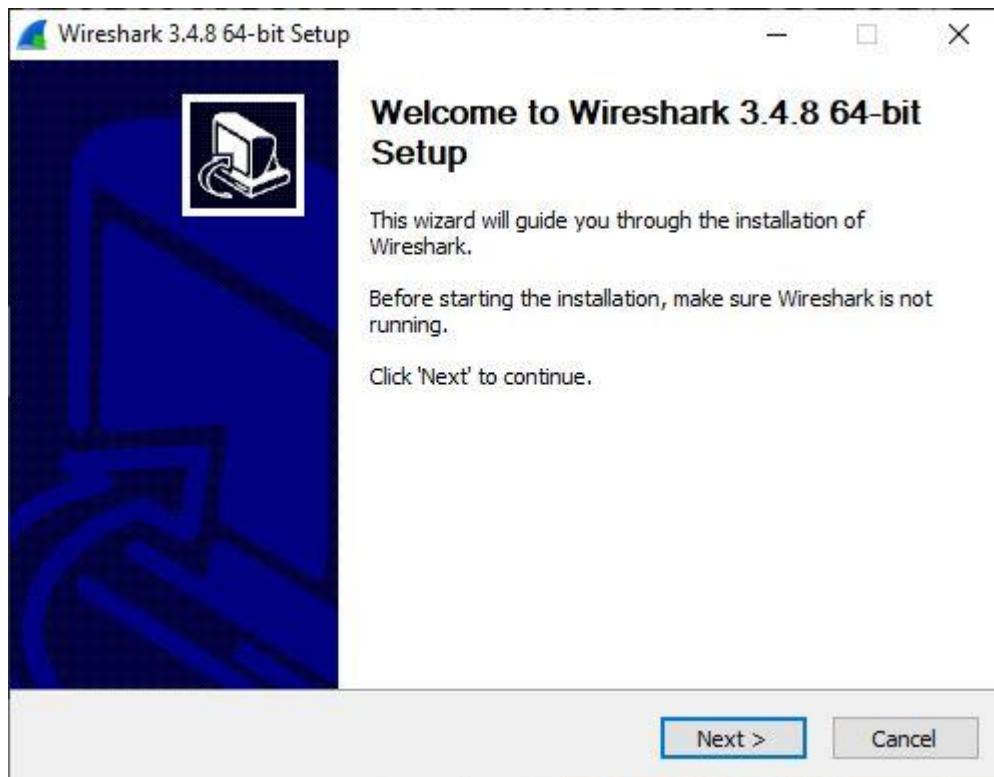


The screenshot shows the Wireshark homepage with a blue background featuring a shark. Three main sections are displayed: 'Download' (with a download icon), 'Learn' (with a book icon), and 'Go Beyond' (with a gear icon). Below each section are sub-headings: 'Get Started Now', 'Knowledge is Power', and 'With SharkFest Sponsors'. At the top, there's a banner for 'SharkFest' and a link to the conference website.

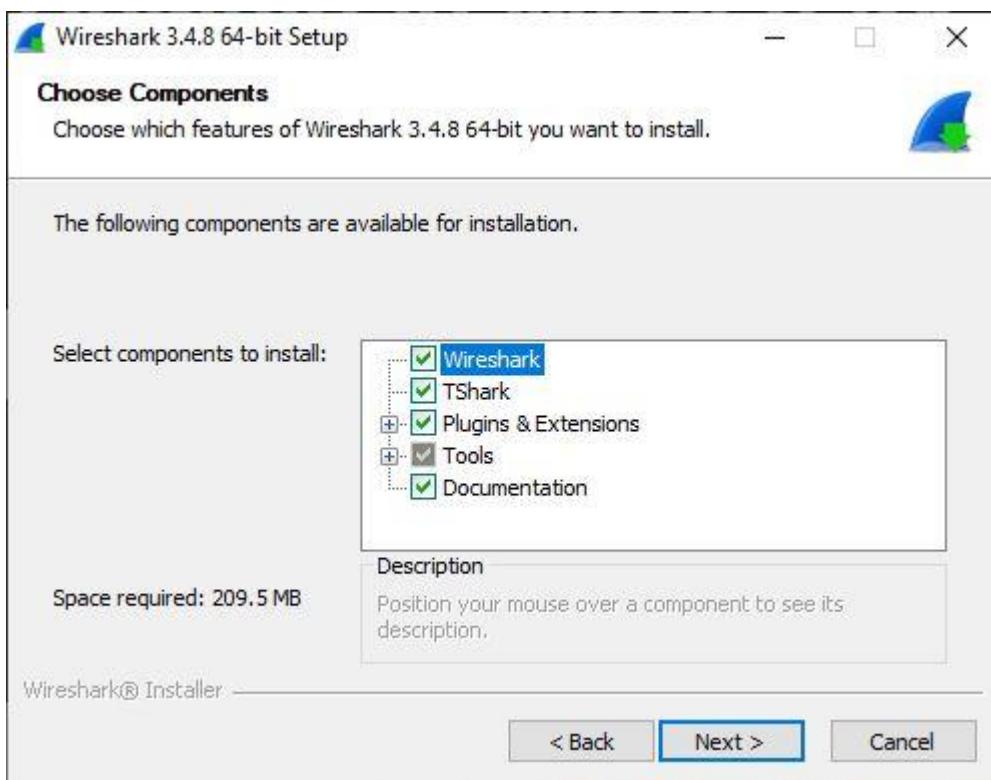
  


The screenshot shows the 'Download Wireshark' page. It features a large heading 'Download Wireshark' and a sub-heading 'The current stable release of Wireshark is 3.4.8.' Below this, it says 'You can also download a development release (3.5.0) and documentation.' On the left, there's a sidebar with download links for different platforms: 'Stable Release (3.4.8) • August 25, 2021' (Windows Installer (64-bit), Windows Installer (32-bit), Windows PortableApps® (32-bit), macOS Intel 64-bit .dmg, Source Code); 'Old Stable Release (3.2.16) • August 25, 2021'; 'Development Release (3.5.0) • August 27, 2021'; and 'Documentation'. A note at the bottom of the sidebar says 'More downloads and documentation can be found on the [downloads page](#)'. On the right, there's a 'SharkFest Sponsors' section with ads for fmadio, riverbed, endace, Scos Training, and DigitalOcean.

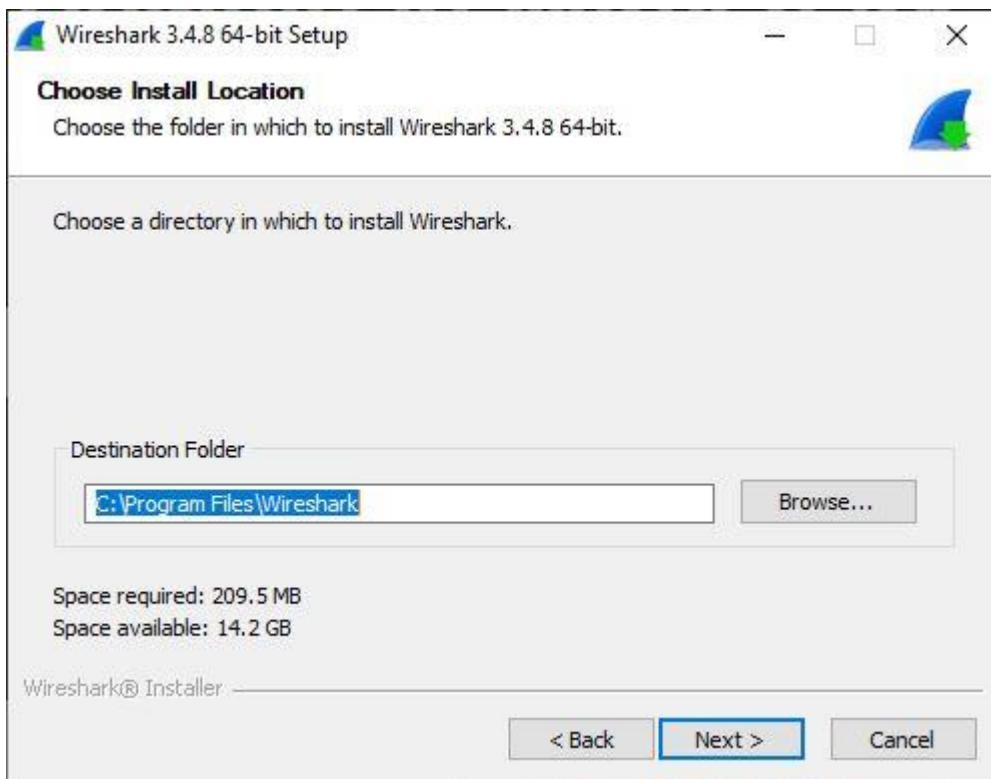
Step 2 : run application and click on noted



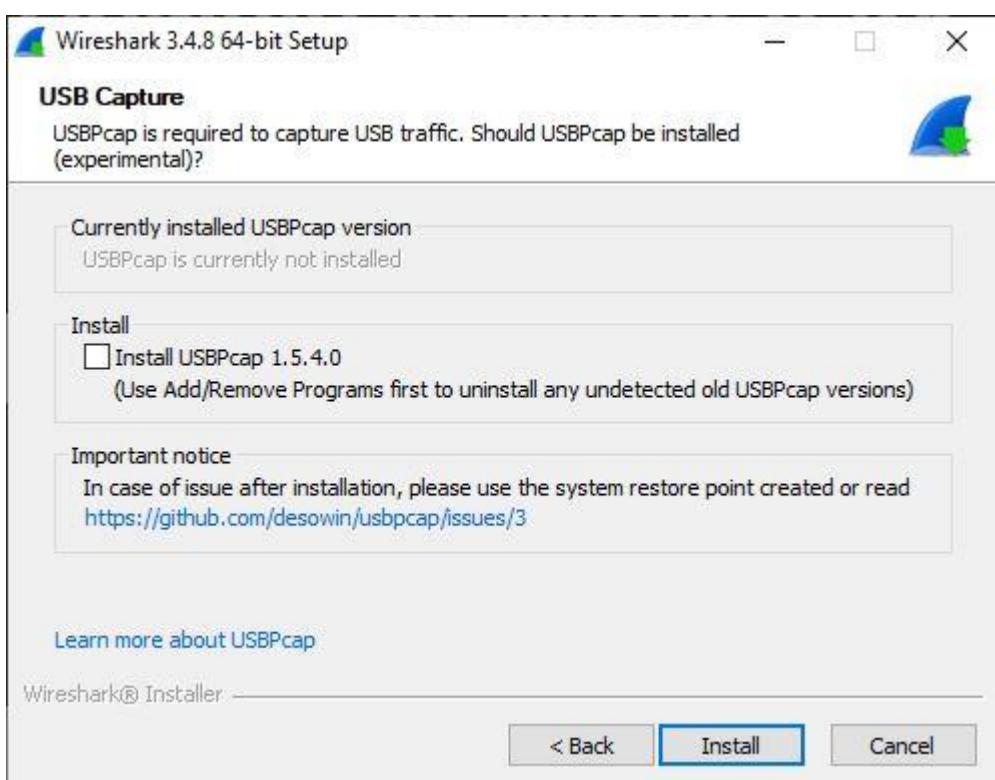
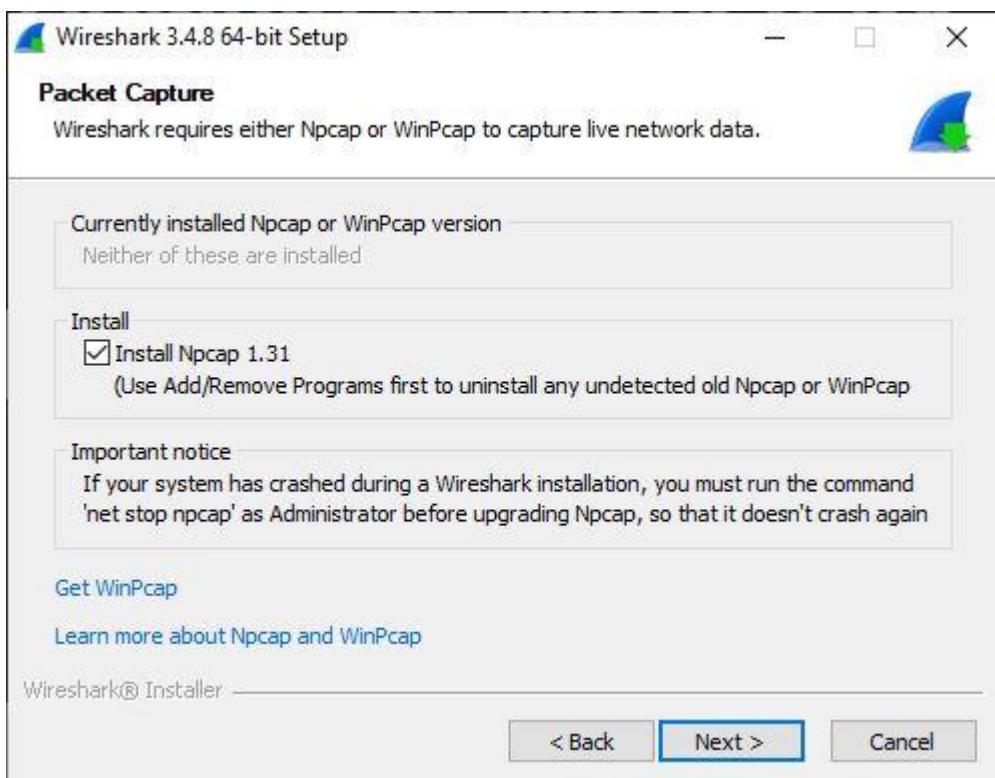
Step 3 :select components and click next

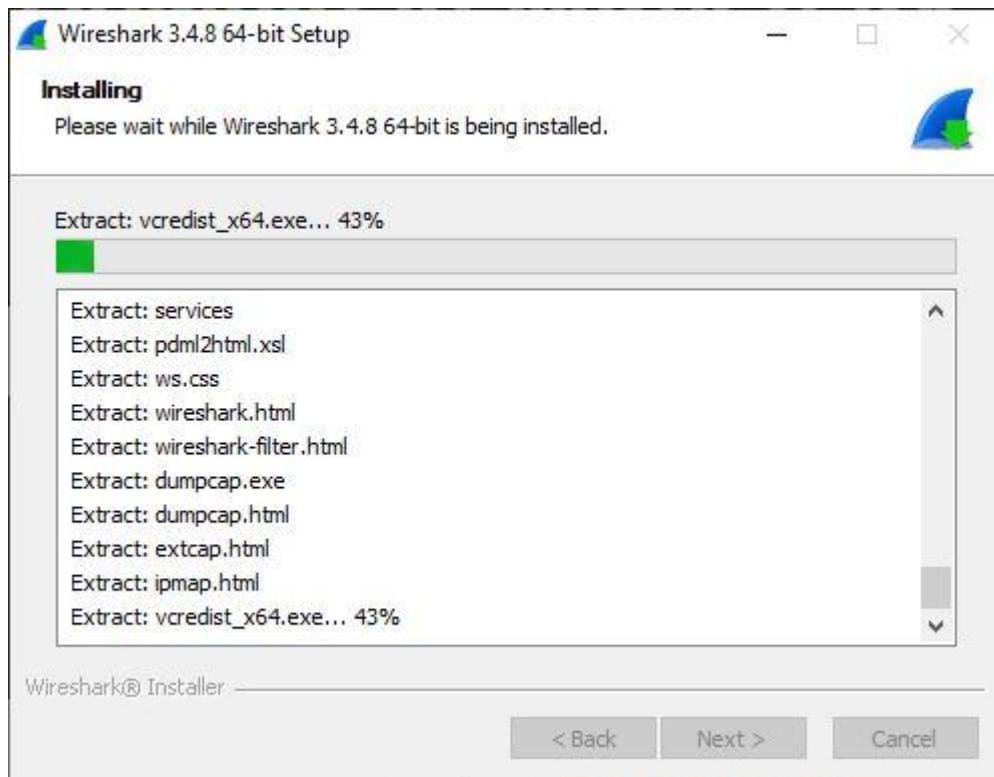


#### Step 4: choose default destination location

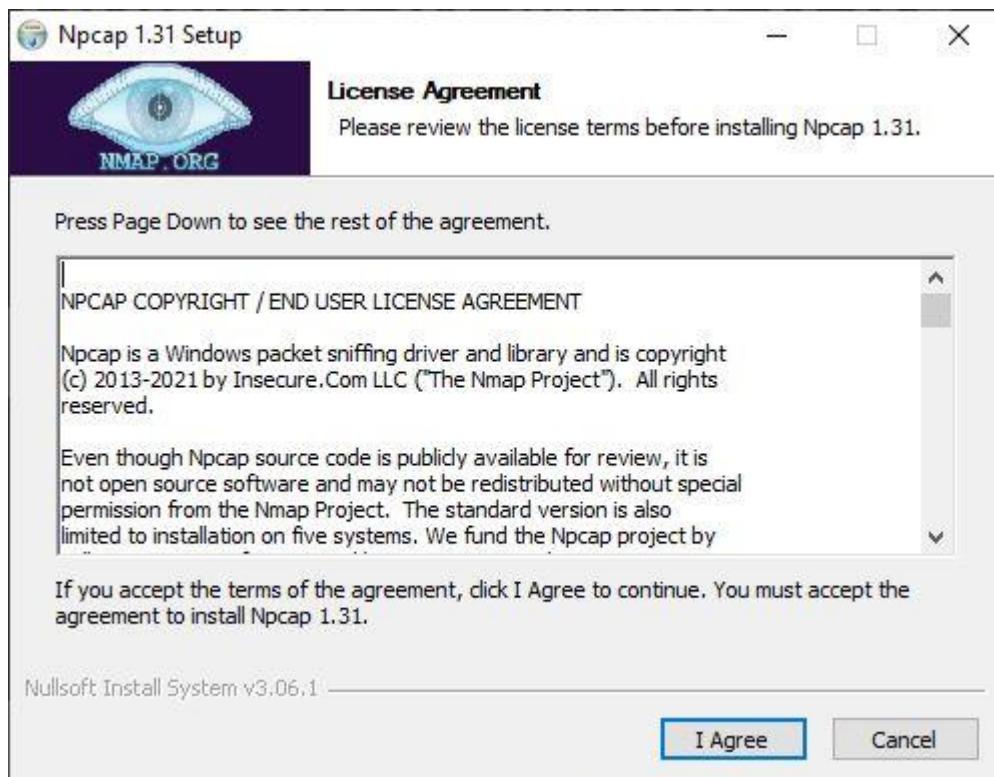


#### Step 5: installation start running

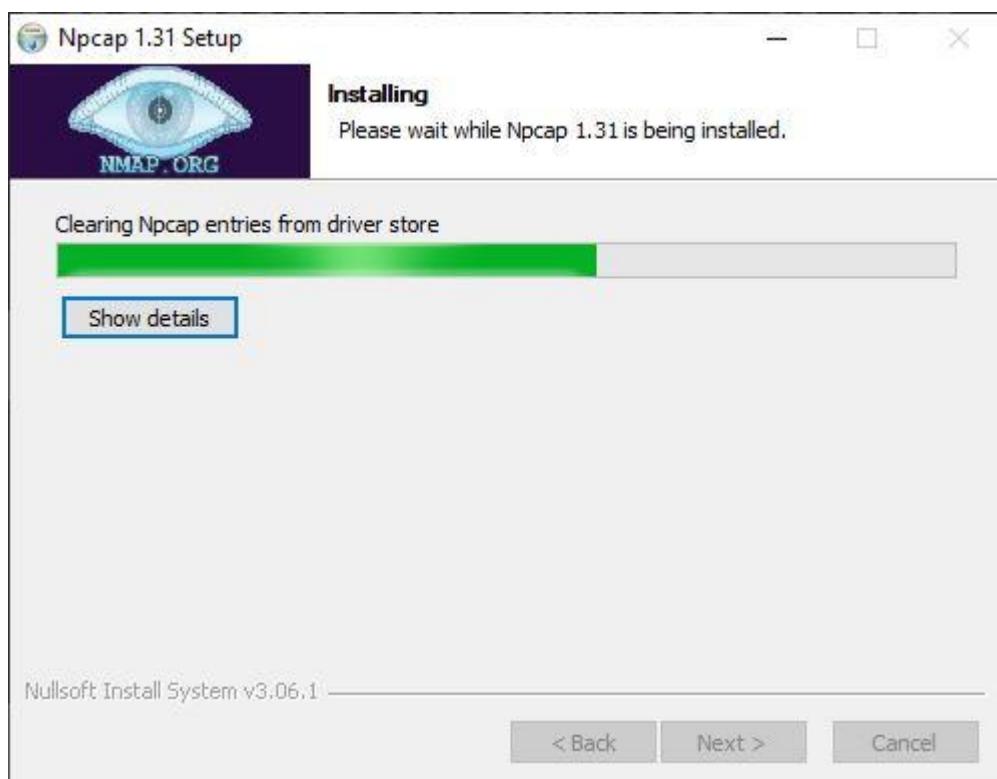
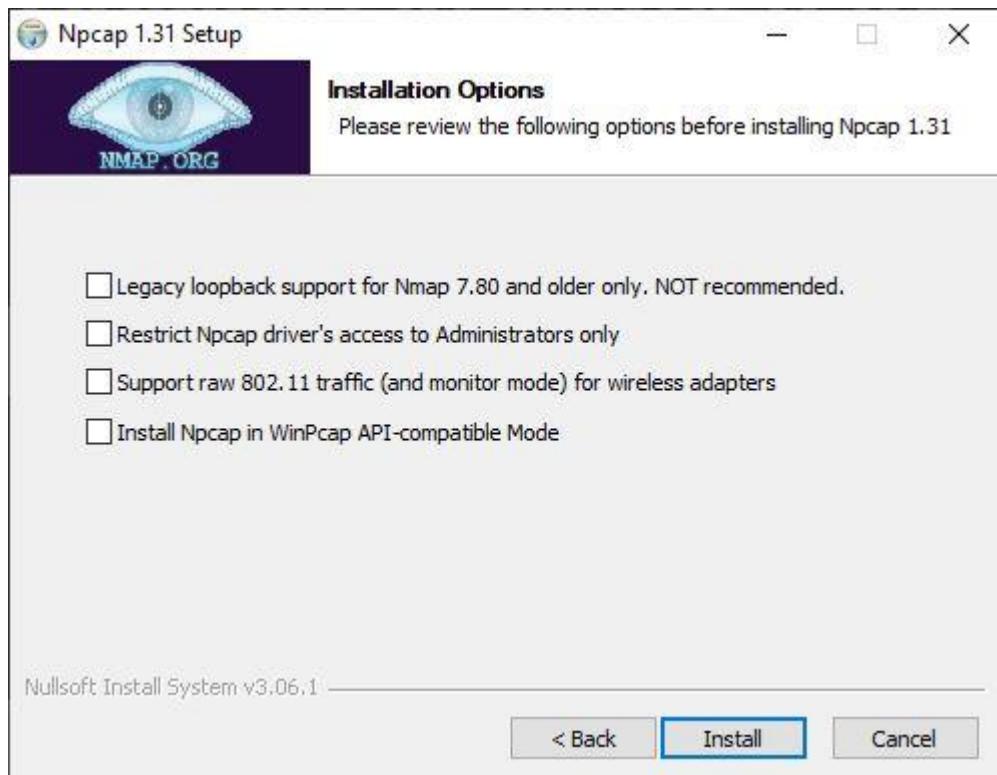


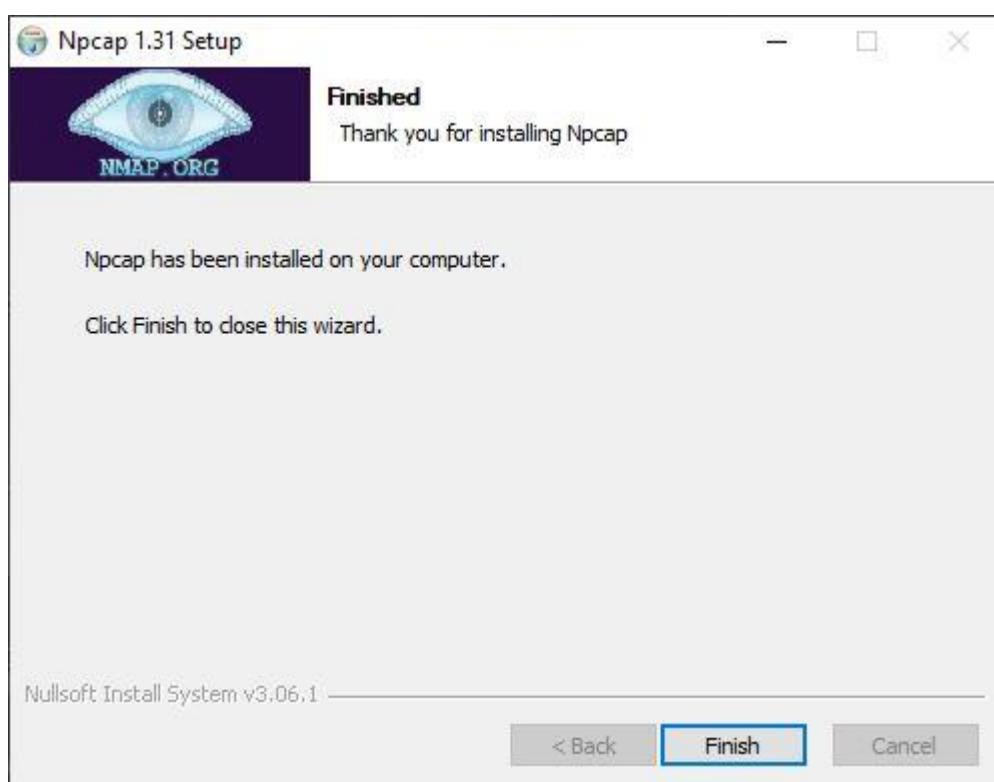
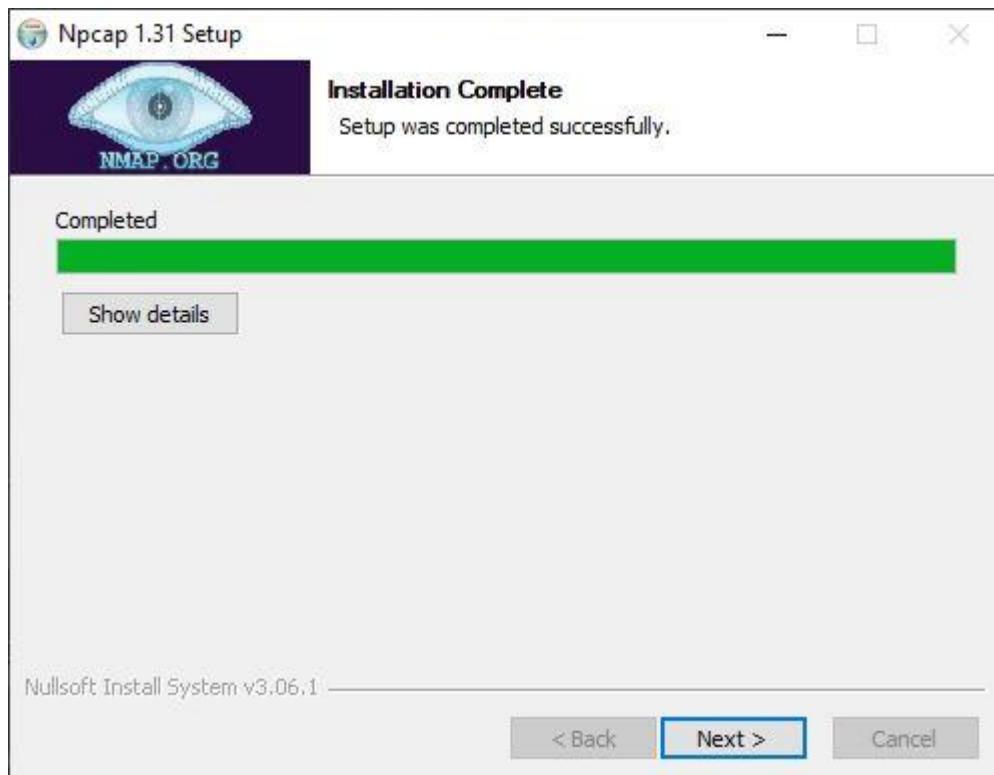


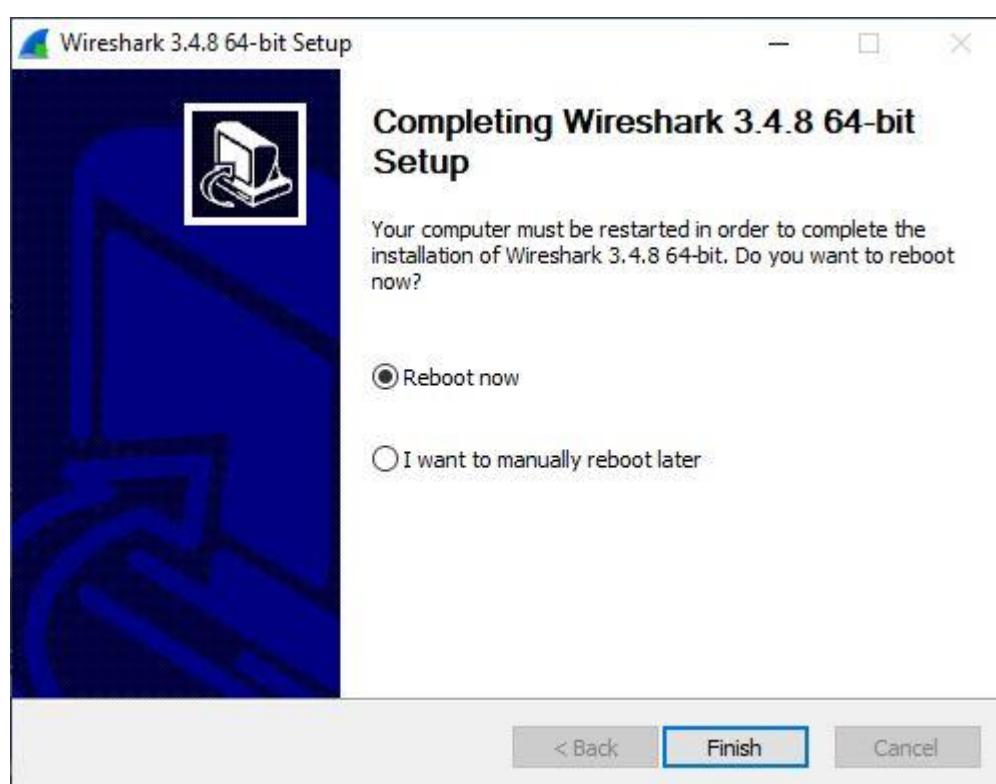
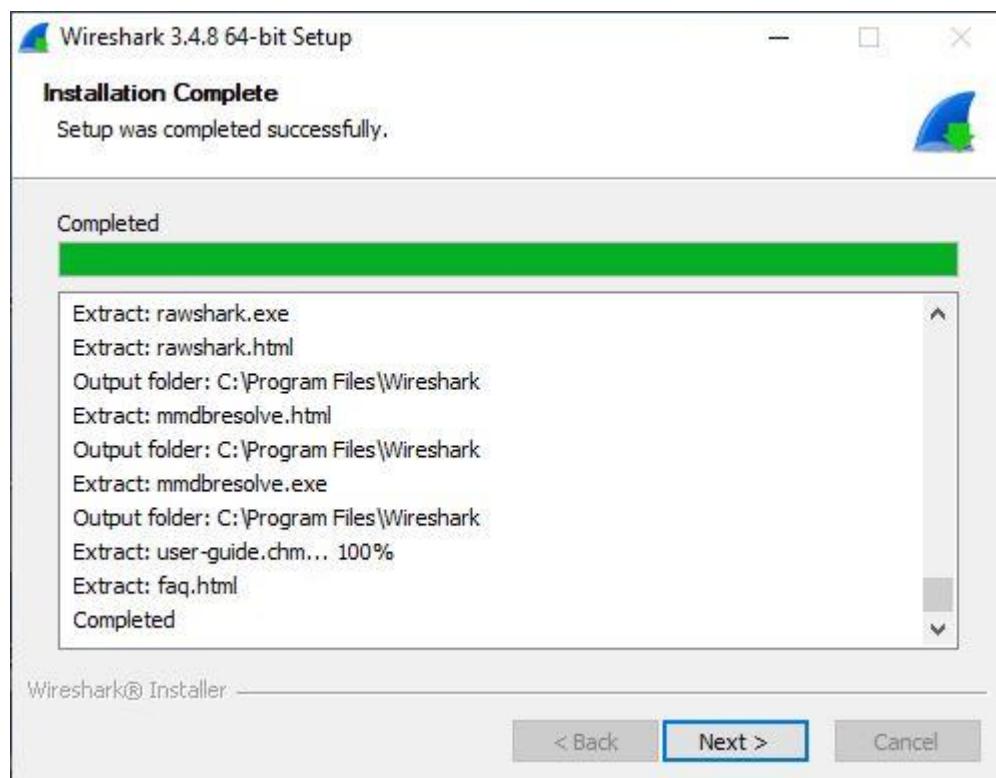
step 6 : click on I agree



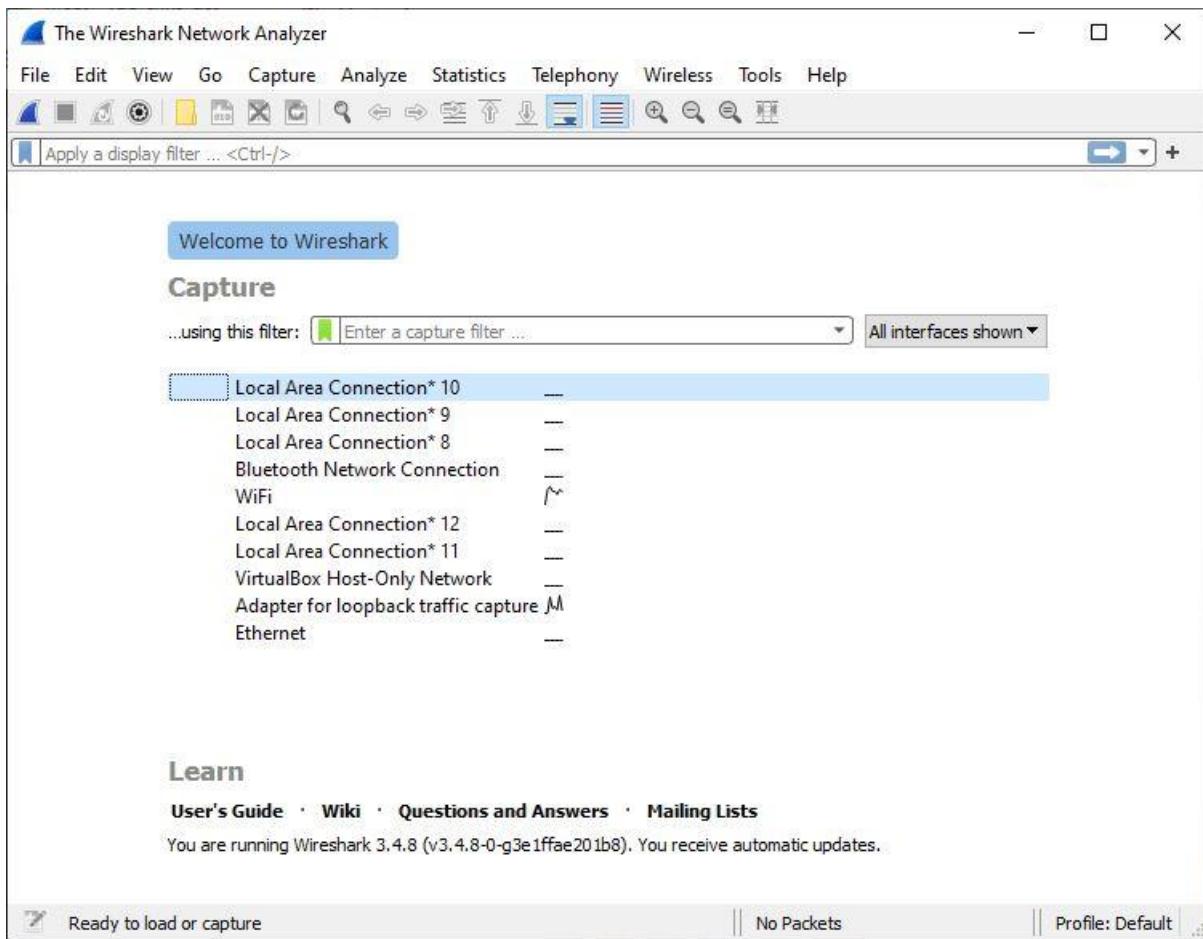
Step 7 : complete installation part and click finish





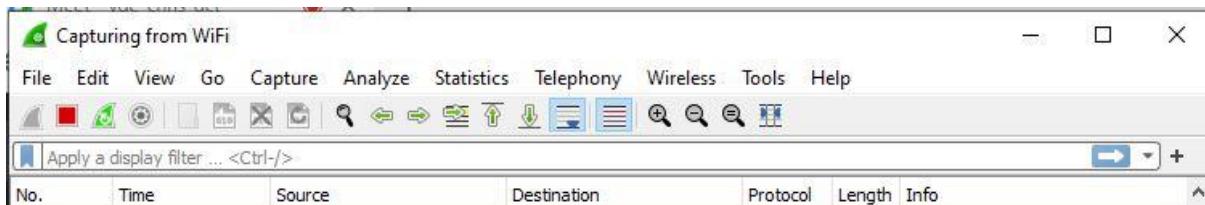
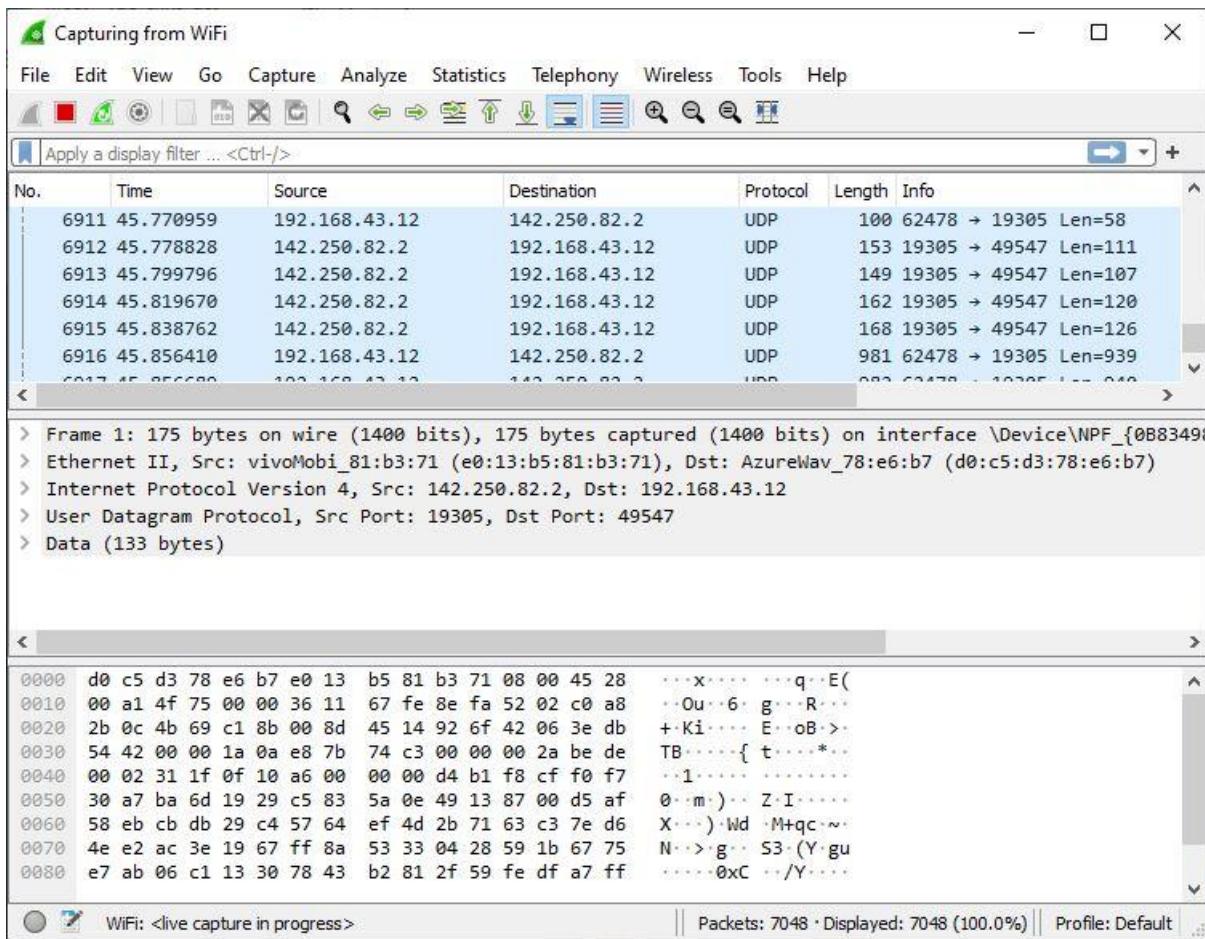


## HOMESCREEN



## HOW IT CAPTURE PACKETS?

- Wireshark captures packets and lets you examine their contents
- Select any interface to capture its packet



- No. shows the number of captured packet or index number.
- Time shows the time of capture
- Source shows the source ip of the packet or the packet is originally generated from which source ip.
- Destination shows the destination ip where the packet is going.
- Protocol shows which kind of protocol communication is held between the source and destination.
- Info shows the data payload in the packet

## FEATURES OF WIRESHARK

- Available for UNIX and Windows.
- Capture live packet data from a network interface.

- Import packets from text files containing hex dumps of packet data.
- Display packets with very detailed protocol information.
- Save packet data captured.
- Export some or all packets in a number of capture file formats.
- Filter packets on many criteria.
- Search for packets on many criteria.
- Colorize packet display based on filters.
- Create various statistics.

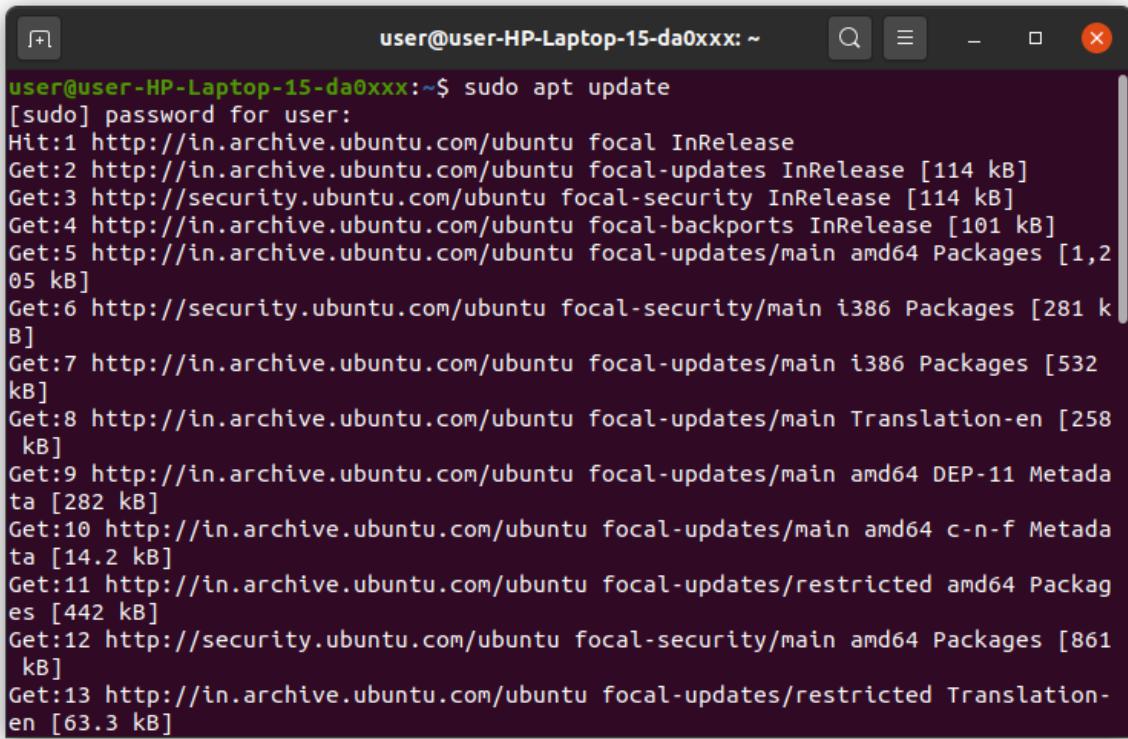
## **TCPDUMP**

- It is an ip utility tool used for real-time packet sniffing(Network).
- Command line program comes in built in a Unix based system.
- Programs like ethereal(Wireshark) provide an alternative to Tcpdump in GUI environment

## **STEPS TO INSTALL TCPDUMP**

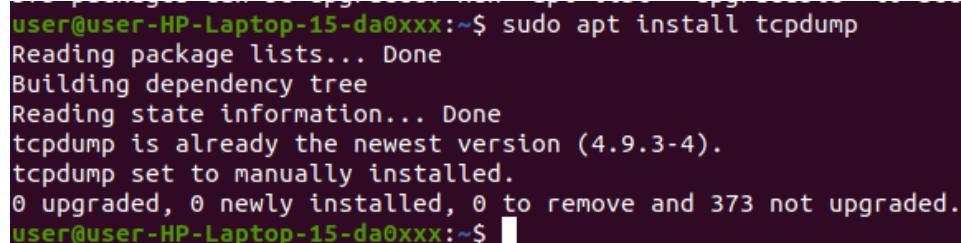
- Install tcpdump by entering the following commands in the terminal:

```
sudo apt update
```



```
user@user-HP-Laptop-15-da0xxx:~$ sudo apt update
[sudo] password for user:
Hit:1 http://in.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1,205 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security/main i386 Packages [281 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu focal-updates/main i386 Packages [532 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [258 kB]
Get:9 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metadata [282 kB]
Get:10 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metadata [14.2 kB]
Get:11 http://in.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [442 kB]
Get:12 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [861 kB]
Get:13 http://in.archive.ubuntu.com/ubuntu focal-updates/restricted Translation-en [63.3 kB]
```

sudo apt install tcpdump

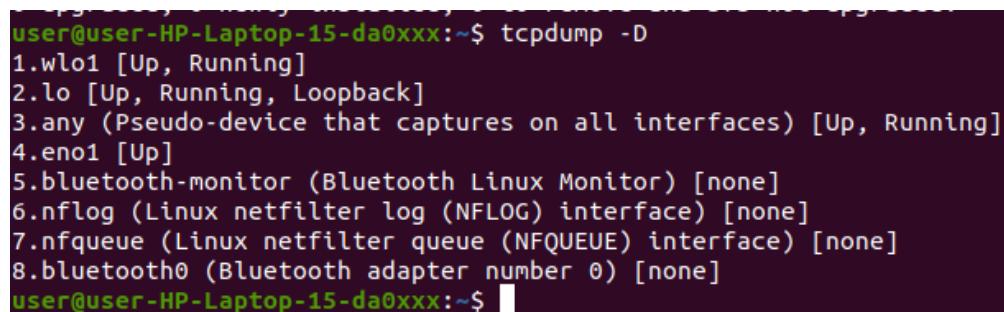


```
user@user-HP-Laptop-15-da0xxx:~$ sudo apt install tcpdump
Reading package lists... Done
Building dependency tree
Reading state information... Done
tcpdump is already the newest version (4.9.3-4).
tcpdump set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 373 not upgraded.
user@user-HP-Laptop-15-da0xxx:~$
```

## TCPDUMP COMMAND EXAMPLES

### 1. Display Available Interfaces

# tcpdump -D



```
user@user-HP-Laptop-15-da0xxx:~$ tcpdump -D
1.wlo1 [Up, Running]
2.lo [Up, Running, Loopback]
3.any (Pseudo-device that captures on all interfaces) [Up, Running]
4.eno1 [Up]
5.bluetooth-monitor (Bluetooth Linux Monitor) [none]
6.nflog (Linux netfilter log (NFLOG) interface) [none]
7.nfqueue (Linux netfilter queue (NFQUEUE) interface) [none]
8.bluetooth0 (Bluetooth adapter number 0) [none]
user@user-HP-Laptop-15-da0xxx:~$
```

### 2. Capture Packets from Specific Interface

# tcpdump -i any

- The command screen will scroll up until you interrupt and when we execute the tcpdump command it will captures from all the interfaces, however with -i switch only capture from the desired interface

### 3. Print Captured Packets in ASCII

- The below tcpdump command with the option -A displays the package in ASCII format. It is a character-encoding scheme format.

```
# tcpdump -A -i any
```

### 4. Capture Only N Number of Packets

- When you run the tcpdump command it will capture all the packets for the specified interface, until you hit the cancel button. But using -c option, you can capture a specified number of packets.

```
# tcpdump -c 5 -i any
```

### 5. Display Captured Packets in HEX and ASCII

- The following command with option -XX capture the data of each packet, including its link level header in HEX and ASCII format.

```
# tcpdump -XX -i any
```

### 6. Capture and Save Packets in a File

- As we said, that tcpdump has a feature to capture and save the file in a .Pcap format, to do this just execute the command with -w option.

```
# tcpdump -w 0001.Pcap -i any
```

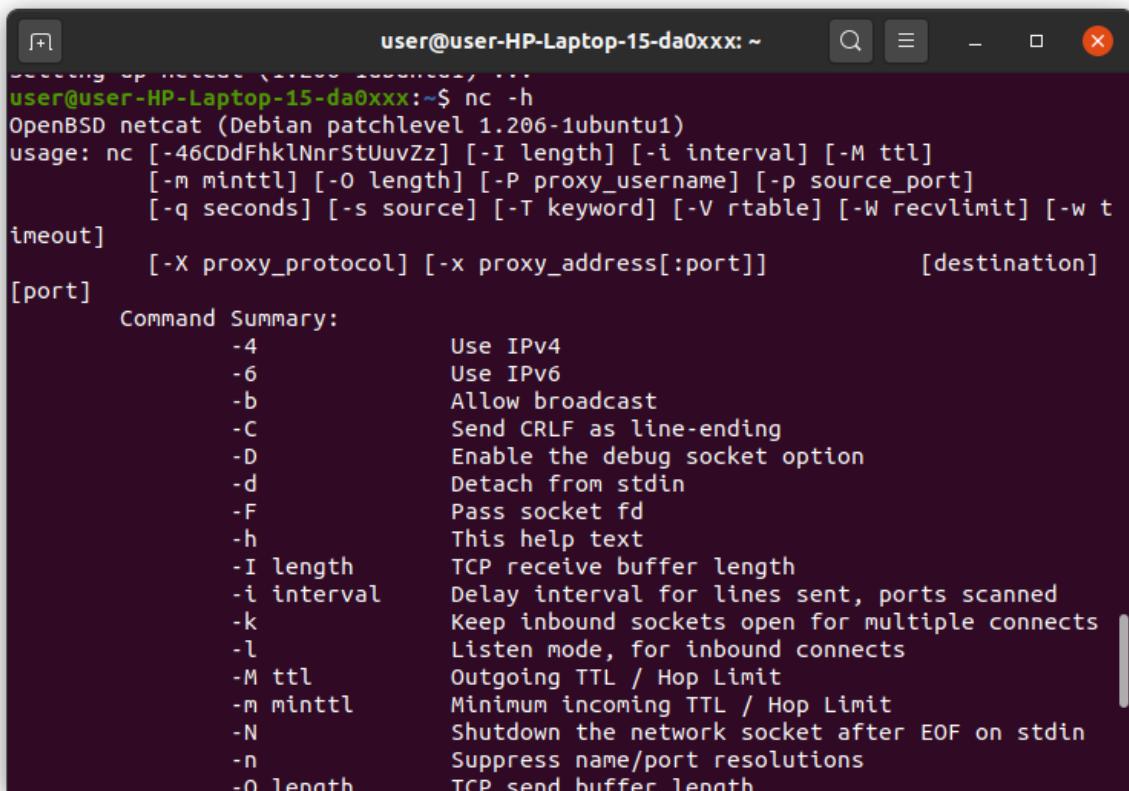
## **Testing network services with Netcat [nc]**

- Use the netcat command, nc, to access the service. If you don't have nc installed, type the following command on the command line

Step 1: \$ sudo apt-get install netcat

```
user@user-HP-Laptop-15-da0xxx:~$ sudo apt-get install netcat
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  netcat
0 upgraded, 1 newly installed, 0 to remove and 373 not upgraded.
Need to get 2,172 B of archives.
After this operation, 15.4 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 netcat all 1.206-1ubuntu1 [2,172 B]
Fetched 2,172 B in 1s (2,162 B/s)
Selecting previously unselected package netcat.
(Reading database ... 195152 files and directories currently installed.)
Preparing to unpack .../netcat_1.206-1ubuntu1_all.deb ...
Unpacking netcat (1.206-1ubuntu1) ...
Setting up netcat (1.206-1ubuntu1) ...
user@user-HP-Laptop-15-da0xxx:~$
```

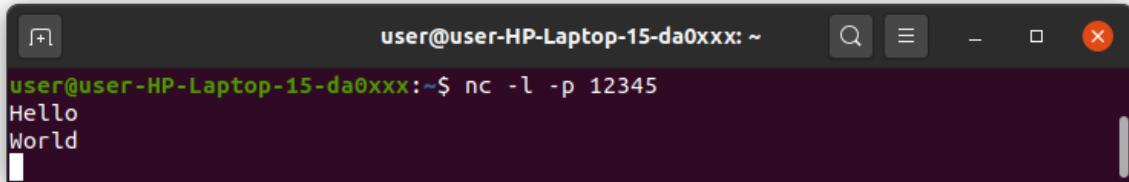
Step 2: After the installation is done type ‘nc -h’



```
user@user-HP-Laptop-15-da0xxx:~$ nc -h
OpenBSD netcat (Debian patchlevel 1.206-1ubuntu1)
usage: nc [-46CDdFhklNnrStUuvZz] [-I length] [-i interval] [-M ttl]
          [-m minttl] [-o length] [-P proxy_username] [-p source_port]
          [-q seconds] [-s source] [-T keyword] [-V rtable] [-W recvlimit] [-w t
imeout]
          [-X proxy_protocol] [-x proxy_address[:port]]           [destination]
[port]   Command Summary:
          -4             Use IPv4
          -6             Use IPv6
          -b             Allow broadcast
          -C             Send CRLF as line-ending
          -D             Enable the debug socket option
          -d             Detach from stdin
          -F             Pass socket fd
          -h             This help text
          -I length     TCP receive buffer length
          -i interval   Delay interval for lines sent, ports scanned
          -k             Keep inbound sockets open for multiple connects
          -l             Listen mode, for inbound connects
          -M ttl         Outgoing TTL / Hop Limit
          -m minttl      Minimum incoming TTL / Hop Limit
          -N             Shutdown the network socket after EOF on stdin
          -n             Suppress name/port resolutions
          -o length      TCP send buffer length
```

Step3: Set up the server using netcat in listening mode.

We will use port 12345 and will specify the port number with -p option.

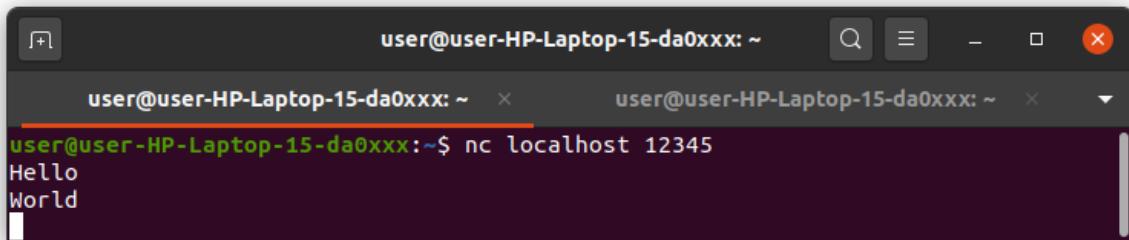


A terminal window titled "user@user-HP-Laptop-15-da0xxx: ~". The command "nc -l -p 12345" is run, and it outputs "Hello" and "World" on separate lines.

```
user@user-HP-Laptop-15-da0xxx:~$ nc -l -p 12345
Hello
World
```

#### Step 4:Creating the server with netcat

- The command 'nc hostname port' puts netcat in client mode and connects to the specified hostname on the specified port. Open a new terminal window and type 'nc localhost 12345'



Two terminal windows are shown. The left window has the command "nc localhost 12345" and outputs "Hello" and "World". The right window shows the netcat server running on port 12345.

```
user@user-HP-Laptop-15-da0xxx:~$ nc localhost 12345
Hello
World
```

#### Step 5 : Now that we are connected to the server we can start chatting

**EXPERIMENT-23****ANALYSE PACKETS USING WIRESHARK**

1. List 3 different protocols that appear in the protocol column in the unfiltered packet-listing window.  
Support your answer with an appropriate screenshot from your computer.

Ans: TCP, UDP, DNS, TLSV1.2, etc..

Capturing From wlo1						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000000	192.168.43.163	192.168.43.1	DNS	98	Standard query 0xf687 A locprod2-e1b-us-west-2.pr
2	0.05073789	192.168.43.163	192.168.43.1	DNS	98	Standard query 0x04f8 AAAA locprod2-e1b-us-west-2.pr
3	0.057182426	192.168.43.1	192.168.43.163	DNS	194	Standard query response 0xf687 A locprod2-e1b-us-west-2.pr
4	0.057309538	192.168.43.1	192.168.43.163	DNS	183	Standard query response 0x04f8 AAAA locprod2-e1b-us-west-2.pr
5	0.058580532	192.168.43.163	44.235.94.69	TCP	74	46594 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460
6	0.309396592	192.168.43.163	54.186.181.218	TCP	74	52282 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460
7	0.385512969	44.235.94.69	192.168.43.163	TCP	74	443 → 46594 [SYN, ACK] Seq=0 Ack=1 Win=26847 Len=0
8	0.385604684	192.168.43.163	44.235.94.69	TCP	66	46594 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0
9	0.387097536	192.168.43.163	44.235.94.69	TLSv1.2	583	Client Hello
10	0.712612618	54.186.181.218	192.168.43.163	TCP	74	443 → 52282 [SYN, ACK] Seq=0 Ack=1 Win=26847 Len=0
11	0.712672182	192.168.43.163	54.186.181.218	TCP	54	52282 → 443 [RST] Seq=1 Win=0 Len=0
12	0.731945836	44.235.94.69	192.168.43.163	TLSv1.2	1354	Server Hello
13	0.732014326	192.168.43.163	44.235.94.69	TCP	66	46594 → 443 [ACK] Seq=518 Ack=1289 Win=64128 Len=0
14	0.732143248	44.235.94.69	192.168.43.163	TCP	66	443 → 46594 [ACK] Seq=1 Ack=518 Win=27136 Len=0
15	0.732166511	192.168.43.163	44.235.94.69	TCP	66	[TCP Dup ACK 13#1] 46594 → 443 [ACK] Seq=518 Ack=518
16	0.733179615	44.235.94.69	192.168.43.163	TLSv1.2	2177	Certificate, Server Key Exchange, Server Hello Done
17	0.733207066	192.168.43.163	44.235.94.69	TCP	66	46594 → 443 [ACK] Seq=518 Ack=3400 Win=63872 Len=0
18	0.737824878	192.168.43.163	44.235.94.69	TLSv1.2	141	Client Key Exchange
19	0.737864719	192.168.43.163	44.235.94.69	TLSv1.2	72	Change Cipher Spec
20	0.7378984140	192.168.43.163	44.235.94.69	TLSv1.2	111	Encrypted Handshake Message
21	1.353322818	44.235.94.69	192.168.43.163	TLSv1.2	117	Change Cipher Spec, Encrypted Handshake Message
22	1.353408295	192.168.43.163	44.235.94.69	TCP	66	46594 → 443 [ACK] Seq=644 Ack=3451 Win=64128 Len=0
23	1.353599825	44.235.94.69	192.168.43.163	TCP	66	443 → 46594 [ACK] Seq=3400 Ack=644 Win=28160 Len=0
24	1.353624963	192.168.43.163	44.235.94.69	TCP	66	[TCP Dup ACK 22#1] 46594 → 443 [ACK] Seq=644 Ack=644
25	1.35372194	192.168.43.163	44.235.94.69	TLSv1.2	284	Application Data
26	1.354477905	192.168.43.163	44.235.94.69	TLSv1.2	173	Application Data
27	1.094106573	44.235.94.69	192.168.43.163	TLSv1.2	644	Application Data
28	1.094266621	192.168.43.163	44.235.94.69	TCP	66	46594 → 443 [ACK] Seq=968 Ack=4029 Win=64128 Len=0
29	1.094366699	44.235.94.69	192.168.43.163	TCP	66	443 → 46594 [ACK] Seq=2451 Ack=968 Win=29440 Len=0
30	1.094378436	192.168.43.163	44.235.94.69	TCP	66	[TCP Dup ACK 28#1] 46594 → 443 [ACK] Seq=968 Ack=968
31	3.913147626	138.199.14.85	192.168.43.163	TCP	66	80 → 41375 [ACK] Seq=1 Ack=1 Win=561 Len=0 TLSv1.2
32	3.9132094855	192.168.43.163	138.199.14.85	TCP	66	[TCP ACKed unseen segment] 41375 → 80 [ACK] Seq=1 Ack=1 Win=561 Len=0
33	5.266280315	192.168.43.163	138.199.14.85	TCP	66	[TCP Keep-Alive] [TCP ACKed unseen segment] 41375 → 80 [ACK] Seq=1 Ack=1 Win=561 Len=0
34	5.590243737	138.199.14.85	192.168.43.163	TCP	66	[TCP Previous segment not captured] 80 → 41375 [ACK] Seq=1 Ack=1 Win=561 Len=0
35	5.778254661	192.168.43.163	35.224.170.84	TCP	74	56360 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460
36	7.4344325874	172.217.174.238	192.168.43.163	UDP	88	443 → 51970 Len=46
37	7.451366056	192.168.43.163	172.217.174.238	UDP	75	51970 → 443 Len=33
38	10.648788511	192.168.43.163	34.122.121.32	TCP	74	51218 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460

2. How long did it take from when the HTTP GET message was sent until the HTTP OK reply was received? (By default, the value of the Time column in the packet-listing window is the amount of time, in seconds, since Wireshark tracing began. To display the Time field in time-of-day format, select the Wireshark View pull down menu, then select Time Display Format, then select Time-of-day.)

Capturing from wlo1						
No.	Time	Source	Destination	Protocol	Length	Info
1107	19.803607528	192.168.43.163	34.104.35.123	HTTP	427	GET /edged1/release2/chrome_component/acze3h5f67uhtnjsyv6pabz...
1199	20.545277344	192.168.43.163	34.104.35.123	HTTP	440	GET /edged1/release2/chrome_component/l3uy2ju206dj5ztu42t3vz4...
1203	20.611152381	34.104.35.123	192.168.43.163	HTTP	1434	[TCP Previous segment not captured] Continuation
1208	20.620517436	34.104.35.123	192.168.43.163	HTTP	1434	Continuation
1213	20.623559282	34.104.35.123	192.168.43.163	HTTP	1434	Continuation
1215	20.624783680	34.104.35.123	192.168.43.163	HTTP	1434	[TCP Previous segment not captured] Continuation
1219	20.625145492	34.104.35.123	192.168.43.163	HTTP	1434	[TCP Previous segment not captured] Continuation
1223	20.634587579	34.104.35.123	192.168.43.163	HTTP	763	Continuation
1226	20.666649819	192.168.43.163	34.104.35.123	HTTP	424	GET /edged1/release2/chrome_component/dp66roauucj16ol7ycwe24...
1262	20.762037849	34.104.35.123	192.168.43.163	HTTP	1718	HTTP/1.1 200 OK
1270	20.850177111	192.168.43.163	142.250.183.176	HTTP	413	GET /update-delta/jflookgnkcckhobaglndicnbbbonegd/2700/2698/...
1274	20.937810459	142.250.183.176	192.168.43.163	HTTP	1434	[TCP Previous segment not captured] Continuation
1278	20.939321691	142.250.183.176	192.168.43.163	HTTP	1434	Continuation
1280	20.940966147	142.250.183.176	192.168.43.163	HTTP	1008	Continuation
1288	20.999604211	192.168.43.163	142.250.183.176	HTTP	427	GET /update-delta/jamhcnnkikhinmdlakkakpbjbbcnf1c/96.0.4648...
1296	21.083363991	142.250.183.176	192.168.43.163	HTTP	2001	HTTP/1.1 200 OK
2502	182.158793428	192.168.43.163	34.122.121.32	HTTP	153	GET / HTTP/1.1
2504	182.816545064	34.122.121.32	192.168.43.163	HTTP	214	HTTP/1.1 204 No Content
2529	185.535354396	192.168.43.163	34.122.121.32	HTTP	153	GET / HTTP/1.1
2636	203.296503580	192.168.43.163	34.122.121.32	HTTP	153	GET / HTTP/1.1
2642	203.818573002	34.122.121.32	192.168.43.163	HTTP	214	HTTP/1.1 204 No Content
2848	219.604644130	192.168.43.163	34.122.121.32	HTTP	153	GET / HTTP/1.1
2850	220.122549407	34.122.121.32	192.168.43.163	HTTP	214	HTTP/1.1 204 No Content

The time interval between a particular HTTP GET message and HTTP OK message is

$$20.762037849\text{s} - 20.666049819\text{s} = 0.09598803\text{s}$$

3. What is the Internet address of the gaia.cs.umass.edu? What is the Internet address of your computer? Support your answer with an appropriate screenshot from your computer.

gaia.cs.umass.edu: 128.119.245.12

Local computer: 192.168.160.204

Capturing from wlo1						
No.	Time	Source	Destination	Protocol	Length	Info
3	0.000374544	192.168.160.204	128.119.245.12	HTTP	627	GET / HTTP/1.1
10	0.743777061	128.119.245.12	192.168.160.204	HTTP	306	HTTP/1.1 304 Not Modified
24	1.180087514	192.168.160.204	104.196.134.131	HTTP	466	GET /images/qupmember.gif HTTP/1.1
28	1.588929510	104.196.134.131	192.168.160.204	HTTP	458	HTTP/1.1 404 Not Found (text/html)

4. Print the two HTTP messages (GET and OK) referred to in question 2 above. To do so, select

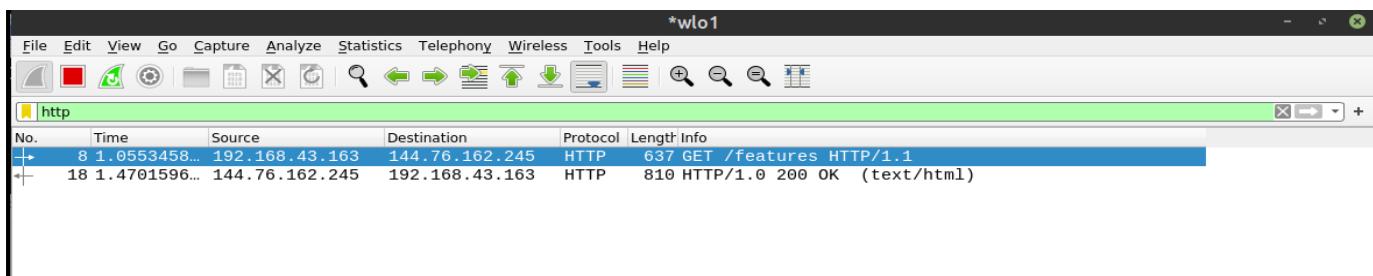
Print from the Wireshark File command menu, and select the “Selected Packet Only” and “Print as displayed” radial buttons, and then click OK

No.	Time	Source	Destination	Protocol	Length	Info
15	2.886090601	18.192.172.30	192.168.160.204	HTTP	117	HTTP/1.1 200 OK (text/html)
Frame 15: 117 bytes on wire (936 bits), 117 bytes captured (936 bits) on interface wlo1, id 0						
Ethernet II, Src: 46:04:d4:d7:47:d2 (46:04:d4:d7:47:d2), Dst: Chongqin_90:41:5f (18:47:3d:90:41:5f)						
Internet Protocol Version 4, Src: 18.192.172.30, Dst: 192.168.160.204						
Transmission Control Protocol, Src Port: 80, Dst Port: 56620, Seq: 2697, Ack: 518, Len: 51						
[3 Reassembled TCP Segments (2747 bytes): #11(1348), #13(1348), #15(51)]						
Hypertext Transfer Protocol						

No.	Time	Source	Destination	Protocol	Length	Info
15	2.886090601	18.192.172.30	192.168.160.204	HTTP	117	HTTP/1.1 200 OK (text/html)
Frame 15: 117 bytes on wire (936 bits), 117 bytes captured (936 bits) on interface wlo1, id 0						
Ethernet II, Src: 46:04:d4:d7:47:d2 (46:04:d4:d7:47:d2), Dst: Chongqin_90:41:5f (18:47:3d:90:41:5f)						
Internet Protocol Version 4, Src: 18.192.172.30, Dst: 192.168.160.204						
Transmission Control Protocol, Src Port: 80, Dst Port: 56620, Seq: 2697, Ack: 518, Len: 51						
[3 Reassembled TCP Segments (2747 bytes): #11(1348), #13(1348), #15(51)]						
Hypertext Transfer Protocol						

5. How many HTTP GET request messages did your browser send? Which packet number in the trace contains the GET message for the Bill or Rights? Which packet number in the trace contains the status code and phrase associated with the response to the HTTP GET request? What is the status code and phrase in the response?

Browser only sent 1 HTTP GET request to the server. The Packet that contained the GET message was packet number 8. The packet that contains the status code and phrase which the server sent in response to the GET message was packet number 18.



6. How many data-containing TCP segments were needed to carry the single HTTP response and the text of the Bill of Rights? Is there any HTTP header information in the transmitted data associated with TCP segmentation? For this question you may want to think about at what layer each protocol operates, and how the protocols at the different layers interoperate.

The data was sent in 3 TCP segments to the browser, then reassembled.

Wireshark screenshot showing network traffic for an HTTP session. The top pane displays a list of captured frames, and the bottom pane provides a detailed analysis of frame 23.

**Captured Frames:**

No.	Time	Source	Destination	Protocol	Length	Info
23	5.4785041...	192.168.160.204	18.192.172.30	HTTP	580	GET /artists.php HTTP/1.1
29	6.1515733...	18.192.172.30	192.168.160.204	HTTP	1322	HTTP/1.1 200 OK (text/html)
93	53.775755...	192.168.160.204	35.232.111.17	HTTP	153	GET / HTTP/1.1
96	54.510483...	35.232.111.17	192.168.160.204	HTTP	214	HTTP/1.1 204 No Content
23...	253.23293...	192.168.160.204	18.192.172.30	HTTP	574	GET /cart.php HTTP/1.1
23...	254.34332...	18.192.172.30	192.168.160.204	HTTP	1277	HTTP/1.1 200 OK (text/html)

**Detailed Analysis of Frame 23:**

- Frame 23: 580 bytes on wire (4640 bits), 580 bytes captured (4640 bits) on interface wlo1, id 0
- Ethernet II, Src: Chongqin\_90:41:5f (18:47:3d:90:41:5f), Dst: 46:04:d4:d7:47:d2 (46:04:d4:d7:47:d2)
- Internet Protocol Version 4, Src: 192.168.160.204, Dst: 18.192.172.30
- Transmission Control Protocol, Src Port: 56746, Dst Port: 80, Seq: 1, Ack: 1, Len: 514
  - Source Port: 56746
  - Destination Port: 80
  - [Stream index: 3]
  - [TCP Segment Len: 514]
  - Sequence number: 1 (relative sequence number)
  - Sequence number (raw): 1156019062
  - [Next sequence number: 515 (relative sequence number)]
  - Acknowledgment number: 1 (relative ack number)
  - Acknowledgment number (raw): 298910811
  - 1000 .... = Header Length: 32 bytes (8)
  - Flags: 0x018 (PSH, ACK)
  - Window size value: 502
  - [Calculated window size: 64256]
  - [Window size scaling factor: 128]
  - Checksum: 0xd176 [unverified]
  - [Checksum Status: Unverified]
  - Urgent pointer: 0
  - Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  - [SEQ/ACK analysis]
  - [Timestamps]

## **EXPERIMENT-24**

### **FAMILIARISATION TO HYPERVISORS AND VIRTUAL MACHINES**

#### **VIRTUAL MACHINE:**

A virtual machine is a virtual representation, or emulation, of a physical computer. They are often referred to as a guest while the physical machine they run on is referred to as the host. Virtualization makes it possible to create multiple virtual machines, each with their own operating system (OS) and applications, on a single physical machine. A VM cannot interact directly with a physical computer. Instead, it needs a lightweight software layer called a hypervisor to coordinate between it and the underlying physical hardware.

#### **HYPERVERISOR:**

Hypervisor is a software program that manages multiple operating systems (or multiple instances of the same operating system) on a single computer system. The hypervisor manages the system's processor, memory, and other resources to allocate what each operating system requires. Hypervisors are designed for a particular processor architecture and may also be called virtualization managers.

#### **HYPERVERISOR TYPES**

##### **Type 1: native (bare-metal) hypervisors**

The Hypervisor runs directly on the host's hardware to control the hardware and to manage guest operating systems.

E.g., Xen, VMWare ESXi, Microsoft Hyper-V

##### **Type 2: hosted hypervisors**

These hypervisors run on a conventional operating system just as other computer programs do.

Eg. VMWare Workstation, VirtualBox

**BENEFITS OF HYPERVISOR**

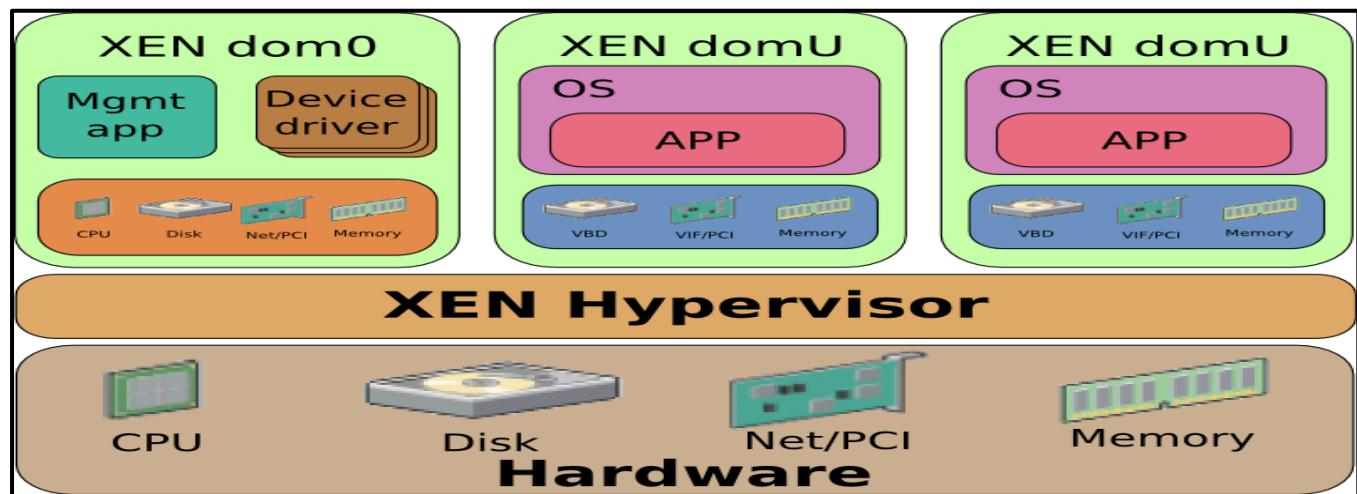
- 1) Speed
- 2) Efficiency
- 3) Flexibility
- 4) Portability

## EXPERIMENT-25

### FAMILIARISATION TO XEN OR KVM

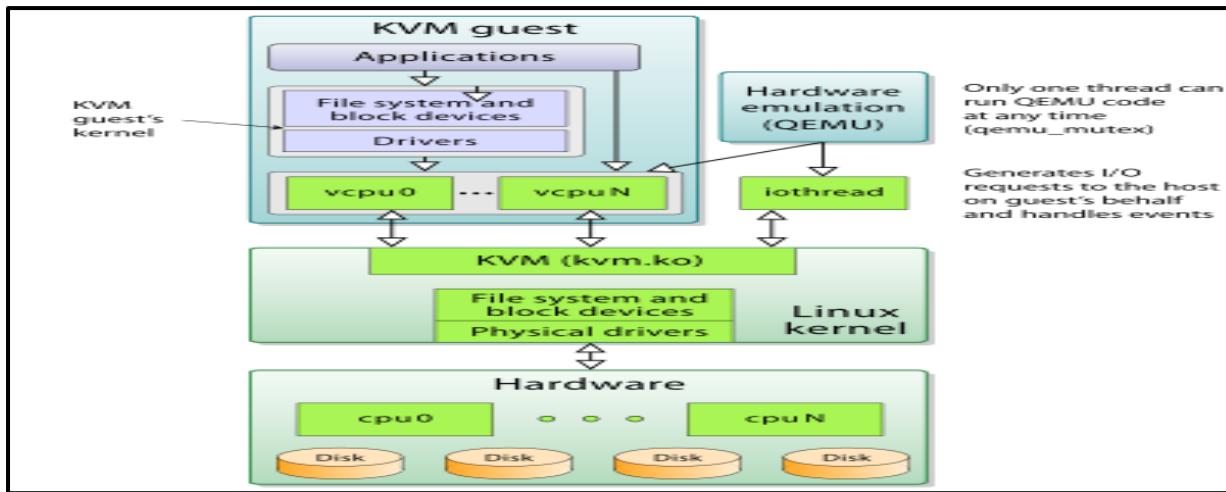
#### **WHAT IS XEN?**

Xen is an open-source paravirtualization technology that provides a platform for running multiple operating systems in parallel on one physical hardware resource.



#### KERNEL-BASED VIRTUAL MACHINE (KVM)

Kernel-based Virtual Machine (KVM) is an **open source virtualization** technology built into Linux. Specifically, KVM lets you turn Linux into a **hypervisor** that allows a host machine to run multiple, isolated virtual environments called guests or virtual machines (VMs). KVM converts Linux into a type-1 (bare-metal) hypervisor. All hypervisors need some operating system-level components—such as a memory manager, process scheduler, input/output (I/O) stack, device drivers, security manager, a network stack, and more—to run VMs.



## DOCKER

### Docker overview

- Docker is an open platform for developing, shipping, and running applications.
- Docker enables you to separate your applications from your infrastructure so you can deliver software quickly.
- With Docker, you can manage your infrastructure in the same ways you manage your applications.
- By taking advantage of Docker's methodologies for shipping, testing, and deploying code quickly, you can significantly reduce the delay between writing code and running it in production.

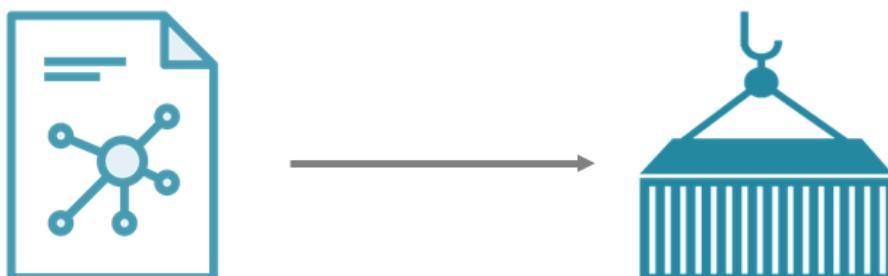
### What is a container?

- Docker provides the ability to package and run an application in a loosely isolated environment called a container.
- The isolation and security allow you to run many containers simultaneously on a given host.
- Containers are lightweight and contain everything needed to run the application, so you do not need to rely on what is currently installed on the host.
- You can easily share containers while you work, and be sure that everyone you share with gets the same container that works in the same way.

### What is a Docker image?

- A Docker image is a file used to execute code in a Docker container. Docker images act as a set of instructions to build a Docker container, like a template.
- Docker images also act as the starting point when using Docker. An image is comparable to a snapshot in virtual machine (VM) environments.

### The Role of Images and Containers



Docker Image

Docker Container

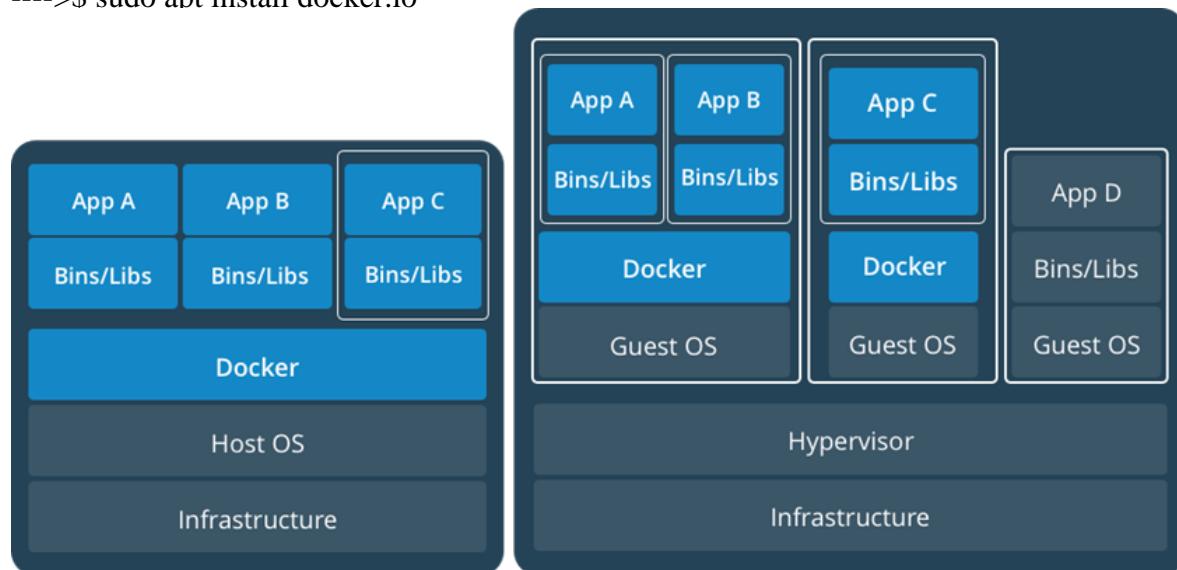
Example: Ubuntu with Node.js and Application Code

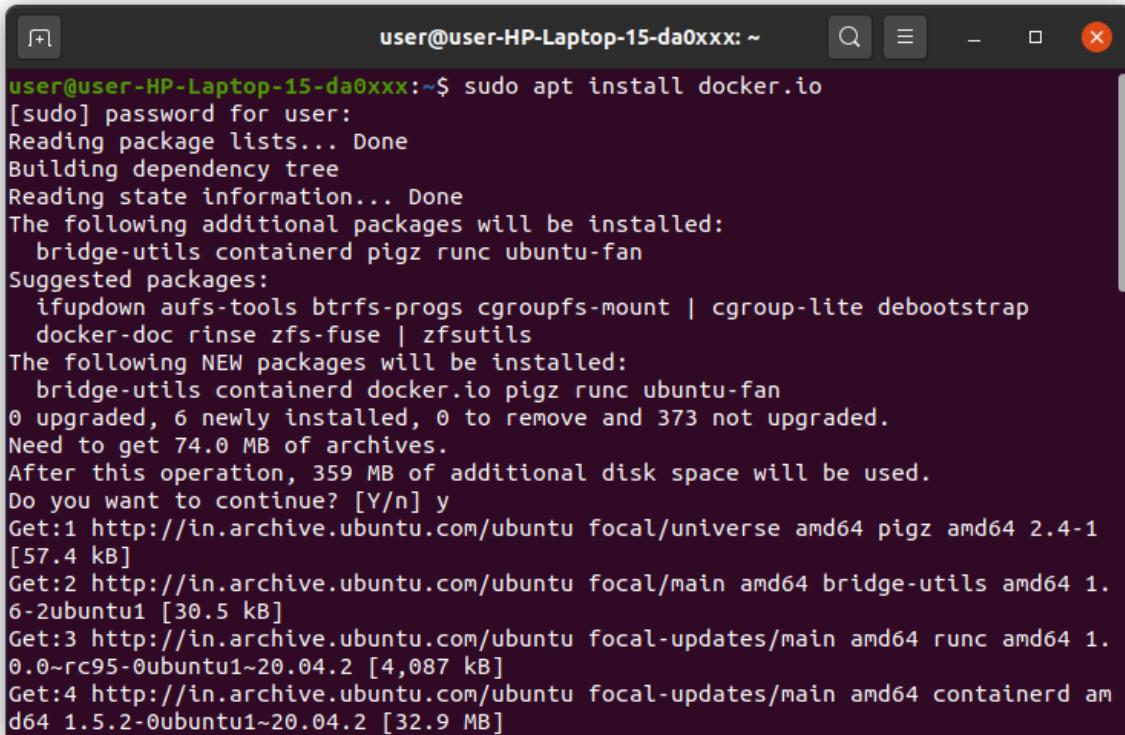
Created by using an image. Runs your application.

## Docker is better...

*Docker installation*

---->\$ sudo apt install docker.io

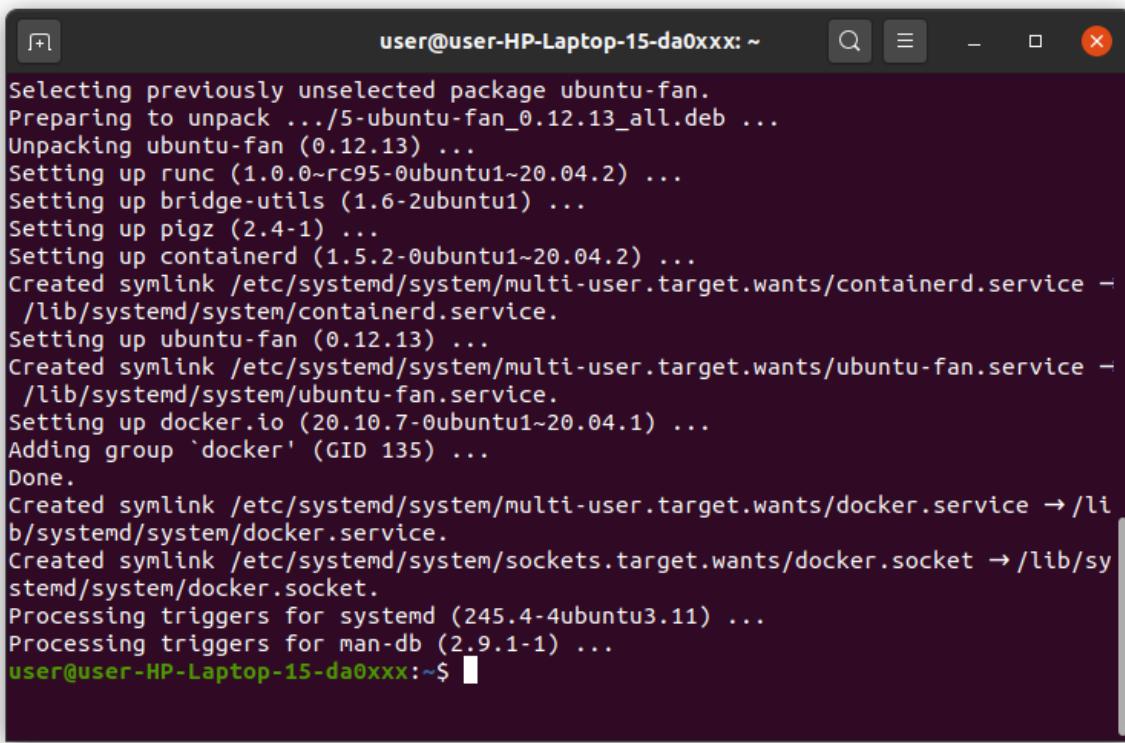




```

user@user-HP-Laptop-15-da0xxx:~$ sudo apt install docker.io
[sudo] password for user:
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 373 not upgraded.
Need to get 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
Get:1 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 pigz amd64 2.4-1
[57.4 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu focal/main amd64 bridge-utils amd64 1.
6-2ubuntu1 [30.5 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 runc amd64 1.
0.0~rc95-0ubuntu1~20.04.2 [4,087 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 containerd am
d64 1.5.2-0ubuntu1~20.04.2 [32.9 MB]

```



```

user@user-HP-Laptop-15-da0xxx:~$ 
Selecting previously unselected package ubuntu-fan.
Preparing to unpack .../5-ubuntu-fan_0.12.13_all.deb ...
Unpacking ubuntu-fan (0.12.13) ...
Setting up runc (1.0.0~rc95-0ubuntu1~20.04.2) ...
Setting up bridge-utils (1.6-2ubuntu1) ...
Setting up pigz (2.4-1) ...
Setting up containerd (1.5.2-0ubuntu1~20.04.2) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service →
 /lib/systemd/system/containerd.service.
Setting up ubuntu-fan (0.12.13) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service →
 /lib/systemd/system/ubuntu-fan.service.
Setting up docker.io (20.10.7-0ubuntu1~20.04.1) ...
Adding group `docker' (GID 135) ...
Done.
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /li
b/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/sy
stemd/system/docker.socket.
Processing triggers for systemd (245.4-4ubuntu3.11) ...
Processing triggers for man-db (2.9.1-1) ...
user@user-HP-Laptop-15-da0xxx:~$ 

```

### *Version check*

----->\$ docker --version

```
user@user-HP-Laptop-15-da0xxx:~$ docker --version
Docker version 20.10.7, build 20.10.7-0ubuntu1~20.04.1
user@user-HP-Laptop-15-da0xxx:~$
```

*Check whether it is running or not*

---->\$ sudo systemctl status checker

*If not active*

---->\$ sudo systemctl enable –now docker

*List all the images you have locally*

---->\$ sudo docker images

```
user@user-HP-Laptop-15-da0xxx:~$ sudo docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
user@user-HP-Laptop-15-da0xxx:~$
```

*Pull an image from the Docker registry*

---->\$ sudo docker pull <image\_name>:<tag>

The screenshot shows a terminal window with the title bar "user@user-HP-Laptop-15-da0xxx: ~". The command entered was "sudo docker pull debian". The output shows:

```
user@user-HP-Laptop-15-da0xxx:~$ sudo docker pull debian
Using default tag: latest
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post http://%2Fvar%2Frun%2Fdocker.sock/v1.24/images/create?fromImage=debian&tag=latest: dial unix /var/run/docker.sock: connect: permission denied
user@user-HP-Laptop-15-da0xxx:~$
```

*List all the running containers*

---->\$ sudo docker ps -a

```
user@user-HP-Laptop-15-da0xxx:~$ sudo docker ps -a
[sudo] password for user:
CONTAINER ID      IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES
user@user-HP-Laptop-15-da0xxx:~$
```

*Remove a container*

---->\$ sudo docker rm <container id>

## **EXPERIMENT-26**

### **INSTALLING SOFTWARE FROM SOURCE CODE**

- Source code software must be compiled and installed.
- Usually comes in a compressed archive, called a tarball with .tar or .tar.gz ending.
- Archive includes source, configure script, makefile, and install scripts.

#### *Package Managers*

- Automate the installation, removal, and management of the software applications.
- Only track software installed using the package manager.
- Similar to Add/Remove programs control panel in MS Windows

#### *--> Configure Script:*

- Inspects system for requirements and configures the “makefile” .

#### *----> Make:*

- Automates the compilation of programming source code for the target system.
- “makefile” defines the necessary steps to build the application.
- They are far from perfect.
- There is no central database to track applications installed with make.
- Removal of applications may or may not be supported by the make file.
- “makefile” contains installation parameters, variables, and setup instructions.
- “make” and “make install” commands are run to compile and install software.

#### *-----> Make Command:*

- Source code distributed as “gzipped tarballs”.
- After unpacking the code you must check the README file for specific install instructions.

```
$ configure  
$ make  
$ make install
```

#### **Installation Steps:**

Step 1: Open the Linux terminal and enter

*sudo apt update*

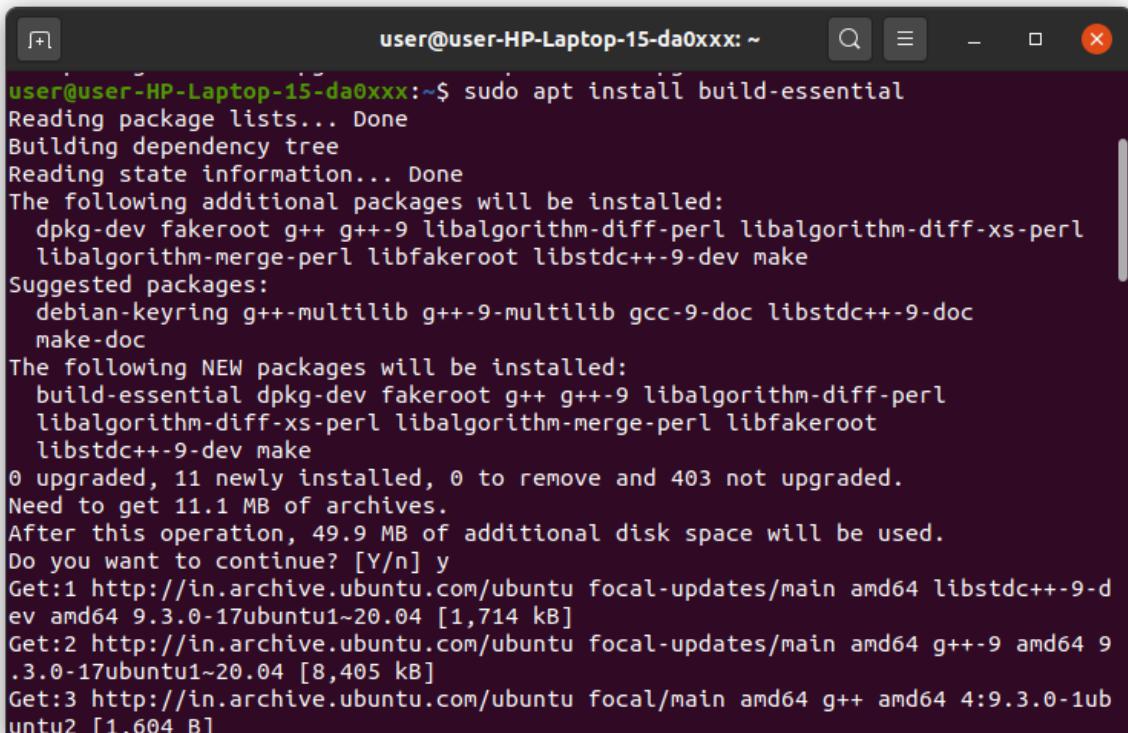
This command is used to install the latest versions of the packages currently installed on the user's system from the sources enumerated in /etc/apt/sources. The installed packages which have new packages available are retrieved and installed.

```
user@user-HP-Laptop-15-da0xxx:~$ sudo apt update
Hit:1 http://in.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://ppa.launchpad.net/ansible/ansible/ubuntu focal InRelease
Hit:4 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease
Get:5 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Fetched 114 kB in 3s (33.0 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
403 packages can be upgraded. Run 'apt list --upgradable' to see them.
user@user-HP-Laptop-15-da0xxx:~$
```

Step 2: Enter

*sudo apt install build-essential*

build-essential is called a meta-package. It in itself does not install anything. Instead, it is a link to several other packages that will be installed as dependencies. In the case of the build-essential meta-package, it will install everything required for compiling basic software written in C and C++..



```
user@user-HP-Laptop-15-da0xxx:~$ sudo apt install build-essential
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  dpkg-dev fakeroot g++ g++-9 libalgorithm-diff-perl libalgorithm-diff-xs-perl
  libalgorithm-merge-perl libfakeroot libstdc++-9-dev make
Suggested packages:
  debian-keyring g++-multilib g++-9-multilib gcc-9-doc libstdc++-9-doc
  make-doc
The following NEW packages will be installed:
  build-essential dpkg-dev fakeroot g++ g++-9 libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libfakeroot
  libstdc++-9-dev make
0 upgraded, 11 newly installed, 0 to remove and 403 not upgraded.
Need to get 11.1 MB of archives.
After this operation, 49.9 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 libstdc++-9-dev amd64 9.3.0-17ubuntu1~20.04 [1,714 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 g++-9 amd64 9.3.0-17ubuntu1~20.04 [8,405 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu focal/main amd64 g++ amd64 4:9.3.0-1ubuntu2 [1.604 kB]
```

Step 3: Enter

*cd /usr/local/src/*

The cd command, also known as chdir (change directory), is a command-line shell command used to change the current working directory in various operating systems.

```
user@user-HP-Laptop-15-da0xxx:~$ cd /usr/local/src
user@user-HP-Laptop-15-da0xxx:/usr/local/src$ sudo wget http://www.noip.com/client/linux/noip-duc-linux.tar.gz
--2021-09-21 16:59:32-- http://www.noip.com/client/linux/noip-duc-linux.tar.gz
Resolving www.noip.com (www.noip.com)... 8.23.224.107
Connecting to www.noip.com (www.noip.com)|8.23.224.107|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 134188 (131K) [application/x-gzip]
Saving to: 'noip-duc-linux.tar.gz'

noip-duc-linux.tar. 100%[=====] 131.04K 71.0KB/s in 1.8s

2021-09-21 16:59:35 (71.0 KB/s) - 'noip-duc-linux.tar.gz' saved [134188/134188]

user@user-HP-Laptop-15-da0xxx:/usr/local/src$
```

Step 4: Enter

```
sudo wget http://www.noip.com/client/linux/noip-duc-linux.tar.gz
```

The wget command is a command line utility for downloading files from the Internet. It supports downloading multiple files, downloading in the background, resuming downloads, limiting the bandwidth used for downloads and viewing headers.

```
user@user-HP-Laptop-15-da0xxx:~$ cd /usr/local/src
user@user-HP-Laptop-15-da0xxx:/usr/local/src$ sudo wget http://www.noip.com/client/linux/noip-duc-linux.tar.gz
--2021-09-21 16:59:32-- http://www.noip.com/client/linux/noip-duc-linux.tar.gz
Resolving www.noip.com (www.noip.com)... 8.23.224.107
Connecting to www.noip.com (www.noip.com)|8.23.224.107|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 134188 (131K) [application/x-gzip]
Saving to: 'noip-duc-linux.tar.gz'

noip-duc-linux.tar. 100%[=====] 131.04K 71.0KB/s in 1.8s

2021-09-21 16:59:35 (71.0 KB/s) - 'noip-duc-linux.tar.gz' saved [134188/134188]

user@user-HP-Laptop-15-da0xxx:/usr/local/src$
```

Step 5: Enter

```
sudo tar xf noip-duc-linux.tar.gz and cd noip-2.1.9-1/
```

The Linux ‘tar’ stands for tape archive, is used to create Archive and extract the Archive files. tar command in Linux is one of the important commands which provides archiving functionality in Linux. We can use Linux tar command to create compressed or uncompressed Archive files and also maintain and modify them.

```
user@user-HP-Laptop-15-da0xxx:/usr/local/src$ sudo tar xf noip-duc-linux.tar.gz
user@user-HP-Laptop-15-da0xxx:/usr/local/src$ ls
noip-2.1.9-1 noip-duc-linux.tar.gz
user@user-HP-Laptop-15-da0xxx:/usr/local/src$ cd noip-2.1.9-1
user@user-HP-Laptop-15-da0xxx:/usr/local/src/noip-2.1.9-1$
```

Step 6: Enter

```
sudo make install
```

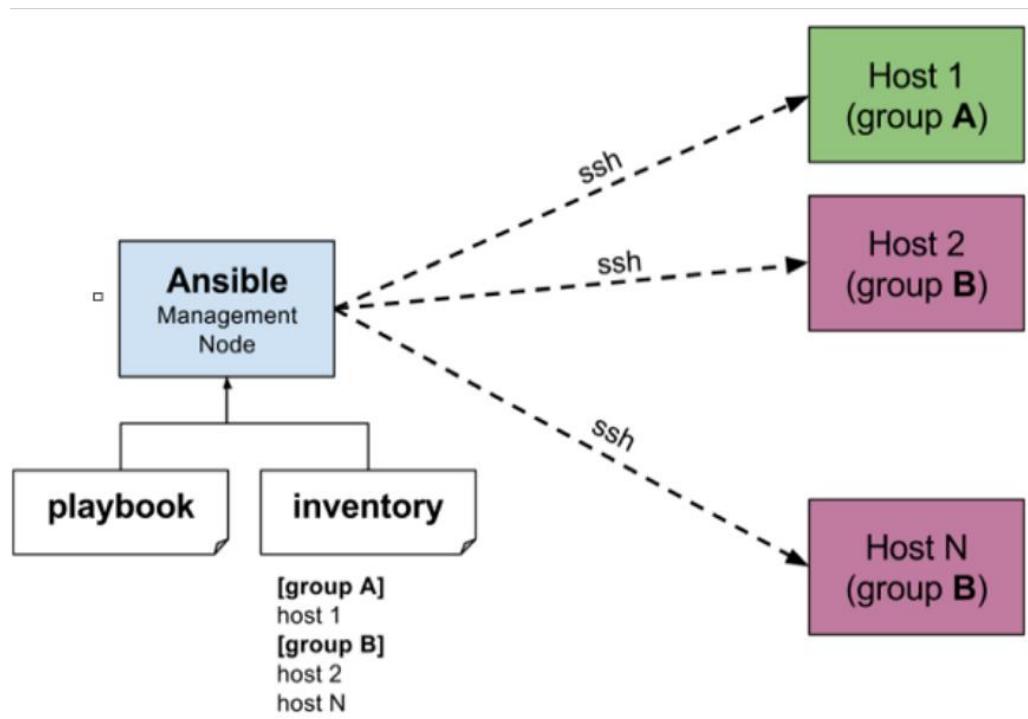
The make install command will copy the built program, and its libraries and documentation, to the correct locations.

**EXPERIMENT-27****DEPLOY LINUX VM USING ANSIBLE PLAYBOOK****What is Ansible ?**

- Ansible is a simple open source IT engine which automates application deployment, intra service orchestration, cloud provisioning and many other IT tools.
- Ansible is easy to deploy because it does not use any agents or custom security infrastructure.
- Ansible uses playbook to describe automation jobs, and playbook uses very simple language i.e. YAML (It's a human-readable data serialization language & is commonly used for configuration files, but could be used in many applications where data is being stored) which is very easy for humans to understand, read and write. Hence the advantage is that even the IT infrastructure support guys can read and understand the playbook and debug if needed (YAML – It is in human readable form).
- Ansible is designed for multi-tier deployment. Ansible does not manage one system at a time, it models IT infrastructure by describing all of your systems as interrelated. Ansible is completely agentless which means Ansible works by connecting your nodes through ssh(by default). But if you want another method for connection like Kerberos, Ansible gives that option to you.
- After connecting to your nodes, Ansible pushes small programs called “Ansible Modules”. Ansible runs that module on your nodes and removes them when finished. Ansible manages your inventory in simple text files (These are the hosts file). Ansible uses the hosts file where one can group the hosts and can control the actions on a specific group in the playbooks.

**How do Ansible playbooks work?**

- Ansible works by connecting to your nodes and pushing out small programs, called "Ansible modules" to them. Ansible then executes these modules (over SSH by default), and removes them when finished. Your library of modules can reside on any machine, and there are no servers, daemons, or databases required.



## Installation Process

- Ansible can be run from any machine with Python 2 (versions 2.6 or 2.7) or Python 3 (versions 3.5 and higher) installed.
- Ansible can be installed on control machines which have above mentioned requirements in different ways. You can install the latest release through Apt, yum, pkg, pip, OpenCSW, pacman,etc.

## Installation through Apt on Ubuntu Machine

- For installing Ansible you have to configure PPA on your machine. For this, you have to run the following line of code:

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

```
sudo apt update
```

```
sudo apt install ansible
```

## Setting up Inventory File

The *inventory file* contains information about the hosts you'll manage with Ansible. You can include anywhere from one to several hundred servers in your inventory file, and hosts can be organized into groups and subgroups. The inventory file is also often used to set variables that will be valid only for specific hosts or groups, in order to be used within playbooks and templates. Some variables can also affect the way a playbook is run, like the `ansible_python_interpreter` variable that we'll see in a moment.

To edit the contents of your default Ansible inventory, open the `/etc/ansible/hosts` file using your text editor of choice, on your Ansible Control Node:

```
$ sudo nano /etc/ansible/hosts
```

```
/etc/ansible/hosts

[servers]
server1 ansible_host=203.0.113.111
server2 ansible_host=203.0.113.112
server3 ansible_host=203.0.113.113

[all:vars]
ansible_python_interpreter=/usr/bin/python3
```

```
ansible-inventory --list -y
```

## Testing Connection

- After setting up the inventory file to include your servers, it's time to check if Ansible is able to connect to these servers and run commands via SSH.
- You can use the -u argument to specify the remote system user. When not provided, Ansible will try to connect as your current system user on the control node.

*ansible all -m ping -u root*

```
Output
server1 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
server2 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
server3 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
```

## PaaS (Platform-as-a-Service)

- Platform-as-a-service (PaaS) is a model of cloud service delivery where a third-party cloud service provider delivers some hardware and software tools, often those needed for application hosting or development, to customers over the internet. The key benefit of the PaaS model is that it enables users to access hardware and software that can be used to develop and run applications without having to purchase, install and maintain the infrastructure.

## Microsoft Azure

- Microsoft Azure, formerly known as Windows Azure, is Microsoft's public cloud computing platform. It provides a range of cloud services, including compute, analytics, storage and networking. Users can pick and choose from these services to develop and scale new applications, or run existing applications in the public cloud.
- The Azure platform aims to help businesses manage challenges and meet their organizational goals. It offers tools that support all industries -
  - including e-commerce, finance and a variety of Fortune 500 companies

-- and is compatible with open source technologies. This provides users with the flexibility to use their preferred tools and technologies. In addition, Azure offers 4 different forms of cloud computing: infrastructure as a service (IaaS), platform as a service (PaaS), software as a service (SaaS) and serverless.

## What is Microsoft Azure used for?

- Microsoft Azure consists of numerous service offerings, its use cases are extremely diverse. Running virtual machines or containers in the cloud is one of the most popular uses for Microsoft Azure. These compute resources can host infrastructure components, such as domain name system (DNS) servers; Windows Server services -- such as Internet Information Services (IIS); or third-party applications. Microsoft also supports the use of third-party operating systems, such as Linux.
- Azure is also commonly used as a platform for hosting databases in the cloud. Microsoft offers serverless relational databases such as Azure SQL and non-relational databases such as NoSQL.
- In addition, the platform is frequently used for backup and disaster recovery. Many organizations use Azure storage as an archive in order to meet their long-term Data retention requirements.

## AZURE SERVICES

- Microsoft sorts Azure cloud services into nearly two dozen categories, including:
- **Compute.** These services enable a user to deploy and manage VMs, containers and batch jobs, as well as support remote application access. Compute resources created within the Azure cloud can be configured with either public IP addresses or private IP addresses, depending on whether the resource needs to be accessible to the outside world.
- **Mobile.** These products help developers build cloud applications for mobile devices, providing notification services, support for back-end tasks, tools for building application program interfaces (APIs) and the ability to couple geospatial context with data.
- **Web.** These services support the development and deployment of web applications. They also offer features for search, content delivery, API management, notification and reporting.
- **Storage.** This category of services provides scalable cloud storage for structured and unstructured data. It also supports big data projects, persistent storage and archival storage.
- **Analytics.** These services provide distributed analytics and storage, as well as features for real-time analytics, big data analytics, data lakes, machine learning (ML), business intelligence (BI), internet of things (IoT) data streams and data warehousing.
- **Networking.** This group includes virtual networks, dedicated connections and gateways, as well as services for traffic management and diagnostics, load balancing, DNS hosting and network protection against distributed denial-of-service (DDoS) attacks.
- **Media and content delivery network (CDN).** These CDN services include on-demand streaming, digital rights protection, encoding and media playback and indexing.
- **Integration.** These are services for server backup, site recovery and connecting private and public clouds.
- **Identity.** These offerings ensure only authorized users can access Azure services and help protect encryption keys and other sensitive information in the cloud. Services include support for Azure Active Directory and multifactor authentication (MFA).
- **Internet of things.** These services help users capture, monitor and analyze IoT data from sensors and other devices. Services include notifications, analytics, monitoring and support for coding and execution.
- **DevOps.** This group provides project and collaboration tools, such as Azure DevOps -- formerly Visual Studio Team Services -- that facilitate DevOps software development processes. It also offers features for application diagnostics, DevOps tool integrations and test labs for build tests and experimentation.
- **Development.** These services help application developers share code, test applications and track potential issues. Azure supports a range of application programming languages, including JavaScript, Python, .NET and Node.js. Tools in this category also include support for Azure DevOps, software development kits (SDKs) and blockchain.

- **Security.** These products provide capabilities to identify and respond to cloud security threats, as well as manage encryption keys and other sensitive assets.
- **Artificial intelligence (AI) and machine learning.** This is a wide range of services that a developer can use to infuse artificial intelligence, machine learning and cognitive computing capabilities into applications and data sets.
- **Containers.** These services help an enterprise create, register, orchestrate and manage huge volumes of containers in the Azure cloud, using common platforms such as Docker and Kubernetes.
- **Databases.** This category includes Database as a Service (DBaaS) offerings for SQL and NoSQL, as well as other database instances -- such as Azure Cosmos DB and Azure Database for PostgreSQL. It also includes Azure SQL Data Warehouse support, caching and hybrid database integration and migration features. Azure SQL is the platform's flagship database service. It is a relational database that provides SQL functionality without the need for deploying a SQL server.
- **Migration.** This suite of tools helps an organization estimate workload Migration costs and perform the actual migration of workloads from local data centers to the Azure cloud.
- **Management and governance.** These services provide a range of backup, recovery, compliance, automation, scheduling and monitoring tools that can help a cloud administrator manage an Azure deployment.
- **Mixed reality.** These services are designed to help developers create content for the Windows Mixed Reality environment.
- **Blockchain.** The Azure Blockchain Service allows you to join a blockchain consortium or to create your own.

## VM DEPLOYMENT WITHOUT ANSIBLE

### HOME PAGE OF AZURE

The screenshot shows the Azure portal home page. At the top, there's a navigation bar with 'Azure services' and a 'Create a resource' button. Below it is a 'Recent resources' section listing five items: 'ansible-rg-test' (Resource group), 'aswin' (SSH key), 'myvm260' (Storage account), 'myNetworkSecurityGroup' (Network security group), and 'myResourceGroup' (Resource group). Further down are sections for 'Navigate' (Subscriptions, Resource groups, All resources, Dashboard) and 'Tools' (Microsoft Learn AI, Azure Monitor, Security Center, Cost Management).

Name	Type	Last Viewed
ansible-rg-test	Resource group	39 minutes ago
aswin	SSH key	2 days ago
myvm260	Storage account	2 days ago
myNetworkSecurityGroup	Network security group	2 days ago
myResourceGroup	Resource group	2 days ago

The screenshot shows the Azure Virtual Machines dashboard. At the top, there are several filter options: 'Location == all' (highlighted with a red arrow), 'Resource group == all', and 'Start with a preset configuration'. Below the filters, there are sorting columns for Name, Subscription, Resource group, Location, Status, Operating system, Size, Public IP address, and Disks. A large message in the center says 'No virtual machines to display' with a sub-instruction: 'Create a virtual machine that runs Linux or Windows. Select an image from the marketplace or use your own customized image.' At the bottom, there are links to 'Learn more about Windows virtual machines' and 'Learn more about Linux virtual machines'.

The screenshot shows the 'Create a virtual machine' wizard. On the left, it displays a 'Project details' section where 'Subscription' is set to 'Azure for Students' and 'Resource group' is set to '(New) Resource group'. On the right, the 'Instance details' section is shown, which includes fields for 'Virtual machine name' (empty), 'Region' (set to '(US) East US'), 'Availability options' (set to 'Availability zone'), 'Availability zone' (set to '1'), 'Image' (set to 'Ubuntu Server 20.04 LTS - Gen1'), 'Azure Spot instance' (unchecked), and 'Size' (set to 'Standard\_D2s\_v3 - 2 vcpus, 8 GiB memory (Loading price...)'). At the bottom, there are navigation buttons: 'Review + create' (highlighted with a red arrow), '< Previous', and 'Next : Disks >'.

## USING ANSIBLE

★ For Queries, go to:

<https://docs.microsoft.com/en-us/azure/developer/ansible/vm-configure?tabs=ansible>

★ To Configure LINUX VM Using Ansible Playbook ,

First we need an azure account, to get a free account:

<https://azure.microsoft.com/en-in/free/>

- Create a resource group
- Create a virtual network
- Create a public IP address
- Create a network security group
- Create a virtual network interface card
- Create a virtual machine

-Open The Azure Shell, we can directly run Ansible in Azure Cloud Shell, where Ansible is pre-installed.

-In case not installed,

```
# Update all packages that have available updates.
```

```
sudo yum update -y
```

```
# Install Python 3 and pip.
```

```
sudo yum install -y python3-pip
```

```
# Upgrade pip3.
```

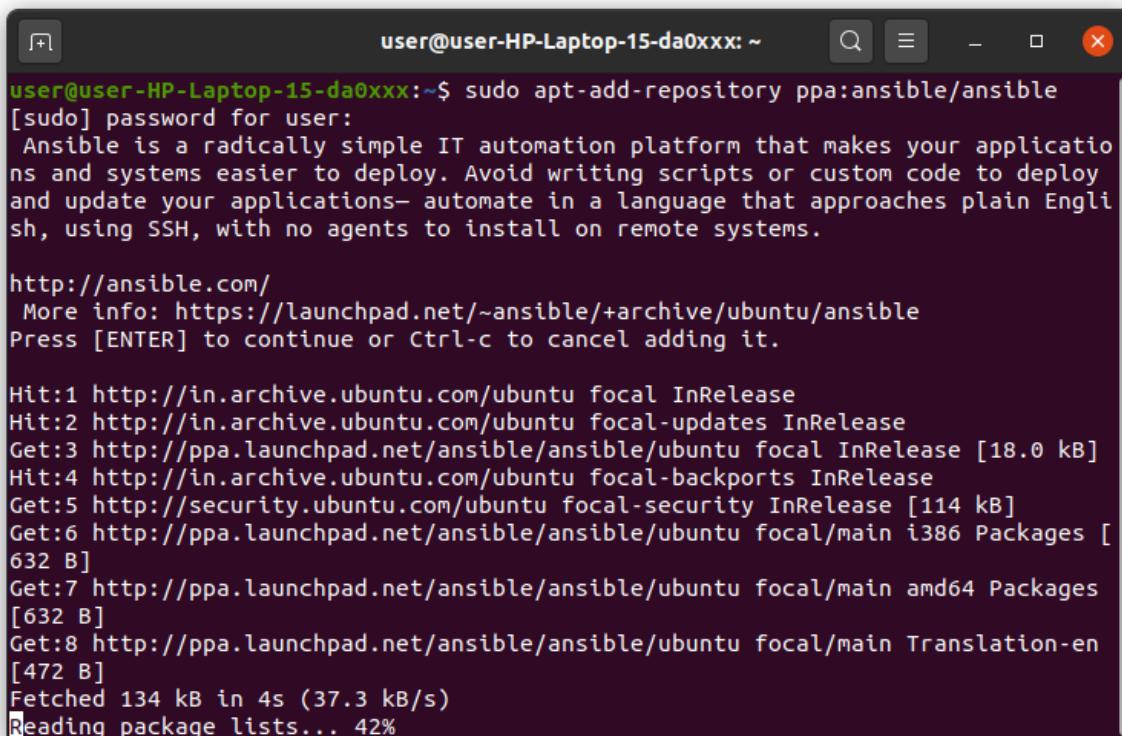
```
sudo pip3 install --upgrade pip
```

```
# Install Ansible.
```

```
pip3 install "ansible==2.9.17"
```

```
# Install Ansible azure_rm module for interacting with Azure.
```

```
pip3 install ansible[azure]
```



```
user@user-HP-Laptop-15-da0xxx:~$ sudo apt-add-repository ppa:ansible/ansible
[sudo] password for user:
Ansible is a radically simple IT automation platform that makes your applications and systems easier to deploy. Avoid writing scripts or custom code to deploy and update your applications— automate in a language that approaches plain English, using SSH, with no agents to install on remote systems.

http://ansible.com/
More info: https://launchpad.net/~ansible/+archive/ubuntu/ansible
Press [ENTER] to continue or Ctrl-c to cancel adding it.

Hit:1 http://in.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease
Get:3 http://ppa.launchpad.net/ansible/ansible/ubuntu focal InRelease [18.0 kB]
Hit:4 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease
Get:5 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:6 http://ppa.launchpad.net/ansible/ansible/ubuntu focal/main i386 Packages [632 B]
Get:7 http://ppa.launchpad.net/ansible/ansible/ubuntu focal/main amd64 Packages [632 B]
Get:8 http://ppa.launchpad.net/ansible/ansible/ubuntu focal/main Translation-en [472 B]
Fetched 134 kB in 4s (37.3 kB/s)
Reading package lists... 42%
```

## ASSIGNMENT

1. List 3 different protocols that appear in the protocol column in the unfiltered packet-listing window. Support your answer with an appropriate screenshot from your computer.

### SOLUTION

The following protocols appeared in the protocol column in the unfiltered packet listing window after downloading a webpage: TCP, UDP, HTTP, DNS.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	80.171.49.132	128.238.4.150	TCP	2509 > 9898 [SYN] Seq=0 Ack=0 win=16384 Len=0 MSS=1380
2	22.001637	128.238.4.150	128.238.2.38	DNS	Standard query A gaia.cs.umass.edu
3	22.231968	128.238.2.38	128.238.4.150	DNS	Standard query response A 128.119.245.12
4	22.231968	128.238.4.150	128.119.245.12	TCP	1310 > http [SYN] Seq=0 Ack=0 win=8192 Len=0 MSS=1460
5	22.412228	128.119.245.12	128.238.4.150	TCP	http > 1310 [SYN, ACK] Seq=0 Ack=1 win=5840 Len=0 MSS=1380
6	22.412228	128.238.4.150	128.119.245.12	TCP	1310 > http [ACK] Seq=1 Ack=1 win=8280 Len=0
7	22.412228	128.238.4.150	128.119.245.12	HTTP	GET /ethereal-labs/intro-ethereal-file1.html HTTP/1.1
8	22.682616	128.119.245.12	128.238.4.150	TCP	http > 1310 [ACK] Seq=1 Ack=425 win=6432 Len=0
9	22.752717	128.119.245.12	128.238.4.150	HTTP	HTTP/1.1 200 OK (text/html)
10	22.862876	128.238.4.150	128.119.245.12	TCP	1310 > http [ACK] Seq=425 Ack=393 win=7888 Len=0
11	32.687002	128.119.245.12	128.238.4.150	TCP	http > 1310 [FIN, ACK] Seq=393 Ack=425 win=6432 Len=0
12	32.687002	128.238.4.150	128.119.245.12	TCP	1310 > http [ACK] Seq=425 Ack=394 win=7888 Len=0
13	32.767117	128.238.4.150	128.119.245.12	TCP	1310 > http [RST, ACK] Seq=425 Ack=394 win=0 Len=0
14	114.22424	128.238.4.150	80.160.91.19	UDP	Source port: 3531 Destination port: 3531
15	114.53469	80.160.91.19	128.238.4.150	UDP	Source port: 3531 Destination port: 3531
16	114.54470	128.238.4.150	80.160.91.19	UDP	Source port: 3531 Destination port: 3531
17	114.55472	128.238.4.150	128.119.17.190	TCP	1311 > 3531 [SYN] Seq=0 Ack=0 win=8192 Len=0 MSS=1460
18	114.55472	128.238.4.150	128.119.17.190	UDP	Source port: 3531 Destination port: 3531
19	116.61768	128.238.4.150	128.119.17.190	UDP	Source port: 3531 Destination port: 3531
20	117.51898	128.238.4.150	128.119.17.190	TCP	1311 > 3531 [SYN] Seq=0 Ack=0 win=8192 Len=0 MSS=1460
21	123.52762	128.238.4.150	128.119.17.190	TCP	1311 > 3531 [SYN] Seq=0 Ack=0 win=8192 Len=0 MSS=1460
					.....

2. How long did it take from when the HTTP GET message was sent until the HTTP OK reply was received? (By default, the value of the Time column in the packet-listing window is the amount of time, in seconds, since Wireshark tracing began. To display the Time field in time-of-day format, select the Wireshark View pull down menu, then select Time Display Format, then select Time-of-day.)

### SOLUTION

If we look at the frame section of the GET request we see that the time the packet arrived is 11:43:13.422848000

```
▽ Frame 109 (492 bytes on wire, 492 bytes captured)
  Arrival Time: Sep 17, 2004 11:43:13.422848000
  Time delta from previous packet: 6.826032000 seconds
  Time since reference or first frame: 9.263432000 seconds
  Frame Number: 109
  Packet Length: 492 bytes
  Capture Length: 492 bytes
```

The same section for the HTTP OK shows an arrival time of

11:43:13.43960400

```

▽ Frame 110 (444 bytes on wire, 444 bytes captured)
  Arrival Time: Sep 17, 2004 11:43:13.439604000
  Time delta from previous packet: 0.016756000 seconds
  Time since reference or first frame: 9.280188000 seconds
  Frame Number: 110
  Packet Length: 444 bytes
  Capture Length: 444 bytes
  
```

The difference of these 2 times gives  $.43960400 - .426032000 = 0.013572$  seconds

- 3.** What is the Internet address of the gaia.cs.umass.edu? What is the Internet address of your computer? Support your answer with an appropriate screenshot from your computer.

### SOLUTION

If we look at the IP section of the GET request, the source and destination are shown

```

Source: 128.238.244.28 (128.238.244.28)
Destination: 128.119.245.12 (128.119.245.12)
The source is the local machine's address and the destination is the web server's public
My (local machine's) address = 128.238.244.28
IP address 128.119.245.12 = www-net.cs.umass.edu.
  
```

This can also be seen during the DNS query

```

▽ Queries
  ▷ gaia.cs.umass.edu: type A, class inet
▽ Answers
  ▷ gaia.cs.umass.edu: type A, class inet, addr 128.119.245.12
  
```

- 4.** Print the two HTTP messages (GET and OK) referred to in question 2 above. To do so, select Print from the Wireshark File command menu, and select the “Selected Packet Only” and “Print as displayed” radial buttons, and then click OK

### SOLUTION

Here is the information for the HTTP GET and OK packets:

#### HTTP GET:

Frame 4 (862 bytes on wire, 862 bytes captured)

Ethernet II, Src: Netgear\_61:8e:6d (00:09:5b:61:8e:6d), Dst: WestellT\_9f:92:b9  
(00:0f:db:9f:92:b9)

Internet Protocol, Src: 192.168.1.46 (192.168.1.46), Dst: 128.119.245.12  
(128.119.245.12)

Transmission Control Protocol, Src Port: 1474 (1474), Dst Port: http (80), Seq: 1,  
Ack: 1, Len: 808

Hypertext Transfer Protocol

GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1\r\n

Host: gaia.cs.umass.edu\r\n

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.4)

Gecko/20070515 Firefox/2.0.0.4\r\n

Accept:

text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,\*/\*;q=0.5\r\n

Accept-Language: en-us,en;q=0.5\r\n

Accept-Encoding: gzip,deflate\r\n

Accept-Charset: ISO-8859-1,utf-8;q=0.7,\*;q=0.7\r\n

Keep-Alive: 300\r\n

Connection: keep-alive\r\n

## HTTP OK:

Frame 6 (439 bytes on wire, 439 bytes captured)

Ethernet II, Src: WestellT\_9f:92:b9 (00:0f:db:9f:92:b9), Dst: Netgear\_61:8e:6d

(00:09:5b:61:8e:6d)

Internet Protocol, Src: 128.119.245.12 (128.119.245.12), Dst: 192.168.1.46

(192.168.1.46)

Transmission Control Protocol, Src Port: http (80), Dst Port: 1474 (1474), Seq: 1,

Ack: 809, Len: 385

Hypertext Transfer Protocol

HTTP/1.1 200 OK\r\n

Date: Thu, 07 Jun 2007 18:09:01 GMT\r\n

Server: Apache/2.0.52 (CentOS)\r\n

Last-Modified: Thu, 07 Jun 2007 18:08:01 GMT\r\n

ETag: "d6c69-50-cb94a240"\r\n

Accept-Ranges: bytes\r\n

Content-Length: 80

Keep-Alive: timeout=10, max=100\r\n

Connection: Keep-Alive\r\n

Content-Type: text/html; charset=ISO-8859-1\r\n

\r\n

Line-based text data: text/html

- 5.** How many HTTP GET request messages did your browser send? Which packet number in the trace contains the GET message for the Bill or Rights? Which packet number in the trace contains the status code and phrase associated with the response to the HTTP GET request? What is the status code and phrase in the response?

### SOLUTION

- My browser only sent 1 HTTP GET request to the server. The Packet that contained the GET message was packet number 4730
- The packet that contains the status code and phrase which the server sent in response to the GET message was packet number 4733
- The code and phrase in the response was 304 Not Modified

No.	Time	Source	Destination	Protocol	Length	Info
+ 4730	13:14:58.295897	192.168.43.104	128.119.245.12	HTTP	616	GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1
- 4733	13:14:59.036890	128.119.245.12	192.168.43.104	HTTP	295	HTTP/1.1 304 Not Modified

- 6.** How many data-containing TCP segments were needed to carry the single HTTP response and the text of the Bill of Rights? Is there any HTTP header information in the transmitted data associated with TCP segmentation? For this question you may want to think about at what layer each protocol operates, and how the protocols at the different layers interoperate.

### SOLUTION

The data was sent in 1 TCP segments to the browser

No.	Time	Source	Destination	Protocol	Length	Info
+ 4730	13:14:58.295897	192.168.43.104	128.119.245.12	HTTP	616	GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1
- 4733	13:14:59.036890	128.119.245.12	192.168.43.104	HTTP	295	HTTP/1.1 304 Not Modified

Frame 4730: 616 bytes on wire (4928 bits), 616 bytes captured (4928 bits) on interface 'Device\NPF\_{1115A191-1EEE-4E00-B416-6C4AB4E598AF}', id 0  
> Ethernet II, Src: Megawell\_7ca:87:73 (04:fc:77:7c:a8:73), Dst: 66:62:fc:8c:63:84 (66:62:fc:8c:63:84)  
> Internet Protocol Version 4, Src: 192.168.43.104, Dst: 128.119.245.12  
< Transmission Control Protocol, Src Port: 62457, Dst Port: 80, Seq: 1, Ack: 1, Len: 562  
Source Port: 62457  
Destination Port: 80  
[Stream Index: 60]  
[TCP Segment Len: 562]  
Sequence Number: 1 (relative sequence number)  
Sequence Number (raw): 788209447  
[Next Sequence Number: 563 (relative sequence number)]