FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)™

HORMIS NAGAR, MOOKKANNOOR

ANGAMALY-683577

'FOCUS ON EXCELLENCE'

NETWORKING & SYSTEM ADMINISTRATION LABORATORY RECORD

Name: NEEHA RAVINDRAN

Branch: MASTER OF COMPUTER APPLICATION

Semester: 2 Batch: B Roll No: 23

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)™

HORMIS NAGAR, MOOKKANNOOR, ANGAMALY-683577

'FOCUS ON EXCELLENCE'

Name: NEEHA RAVINDRAN

Branch: MASTER OF COMPUTER APPLICATION

Semester: 2 **Roll No: 23**

University Exam. Reg. No: FIT21MCA-2083

CERTIFICATE

This is to certify that this is a Bonafide record of the Practical work done and submitted to Kerala Technological University in partial fulfilment for the award of the Master of Computer Applications is a record of the original research work done by ***NEEHA RAVINDRAN*** *in the* ***NETWORKING & SYSTEM ADMINISTARION*** *Laboratory of the Federal Institute of Science and Technology during the academic year 2021-2022.*

Signature of Staff in Charge

Name:

Signature of H.O.D

Name:

Date of University practical examination .....

Signature of

Internal Examiner

Signature of

External Examiner

CONTENT

EXPERIMENT – 1

What is a computer?

A computer is an electronic device that manipulates information, or data. It has the ability to store, retrieve, and process data.

History of computers

In the early 1820s, it was designed by Charles Babbage who is known as "Father of Modern Computer". It was a mechanical computer which could perform simple calculations. It was a steam driven calculating machine designed to solve tables of numbers like logarithm tables Modern computers can perform generic sets of operations known as programs. These programs enable computers to perform a wide range of tasks. A computer system is a "complete" computer that includes the hardware, operating system (main software), and peripheral equipment needed and used for "full" operation. This term may also refer to a group of computers that are linked and function together, such as a computer network or computer cluster.

A broad range of industrial and consumer products use computers as control systems. Simple special-purpose devices like microwave ovens and remote controls are included, as are factory devices like industrial robots and computer-aided design, as well as general-purpose devices like personal computers and mobile devices like smartphones. Computers power the Internet, which links billions of other computers and users.

Early computers were meant to be used only for calculations. Simple manual instruments like the abacus have aided people in doing calculations since ancient times. Early in the Industrial Revolution, some mechanical devices were built to automate long tedious tasks, such as guiding patterns for looms. More sophisticated electrical machines did specialized analog calculations in the early 20th century. The first digital electronic calculating machines were developed during World War II.Thefirst semiconductor transistors in the late 1940s were followed by the silicon-based MOSFET (MOS transistor) and monolithic integrated circuit (IC) chip technologies in the late 1950s, leading to the microprocessor and the microcomputer revolution in the 1970s. The speed, power and versatility of computers have been increasing dramatically ever since then, with transistor counts increasing at a rapid pace (as predicted by Moore's law), leading to the Digital Revolution during the late 20th to early 21st centuries.

Components of Computer

1. Mother Board

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

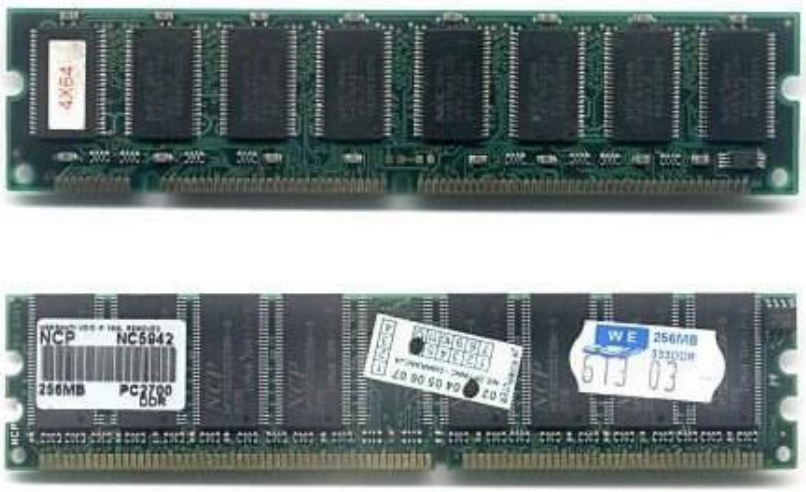
The motherboard is mounted inside the case and is securely attached via small screws through pre-drilled holes. Motherboard contains ports to connect all of the internal components. It provides a single socket for CPU, whereas for memory, normally one or more slots are available. Motherboards provide ports to attach the floppy drive, hard drive, and optical drives via ribbon cables. Motherboard carries fans and a special port designed for power supply.

There is a peripheral card slot in front of the motherboard using which video cards, sound cards, and other expansion cards can be connected to the motherboard.

On the left side, motherboards carry a number of ports to connect the monitor, printer, mouse, keyboard, speaker, and network cables. Motherboards also provide USB ports, which allow compatible devices to be connected in plug-in/plug-out fashion. For example, pen drive, digital cameras, etc.

2. Ram Modules

In computing, a memory module or RAM (random-access memory) stick is a printed circuit board on which memory integrated circuits are mounted. Memory modules permit easy installation and replacement in electronic systems, especially computers such as personal computers, workstations, and servers. The first memory modules were proprietary designs that were specific to a model of computer from a specific manufacturer.



3. Daughter cards

A daughterboard is type of circuit board that plugs in or is attached to the motherboard or similar expansion card to extend its features and services. A daughterboard complements the existing functionality of a motherboard or an expansion card. A daughterboard is also known as daughter card, piggyback board, riser card or mezzanine board. A daughterboard is connected directly to the motherboard. Unlike expansion cards, which connect with the motherboard using bus and other serial interfaces, daughterboards are usually directly embedded through soldering. Like a motherboard, a daughterboard has sockets, pins, plugs and connectors to be attached to other boards. Typically, daughterboards are released as a post-launch update to a motherboard or expansion card. For example, a MIDI daughterboard is used to add on the functionality of the sound card.

Functionalities of daughter board:

• It is known as the piggyback board, riser card, daughtercard etcetera.

• A daughter board is smaller than a motherboard and may have some slots like the motherboard.

• A daughter board is a printed circuit board which is connected to the motherboard or expansion card.

• Unlike expansion card, daughter boards are directly connected to the motherboard by soldering.

• Daughter boards do not provide new functions to the circuit like an expansion but they extend the circuitry of the circuit in which they are plugged into.

• Daughter boards are released by the vendors as an update of motherboard or expansion card.

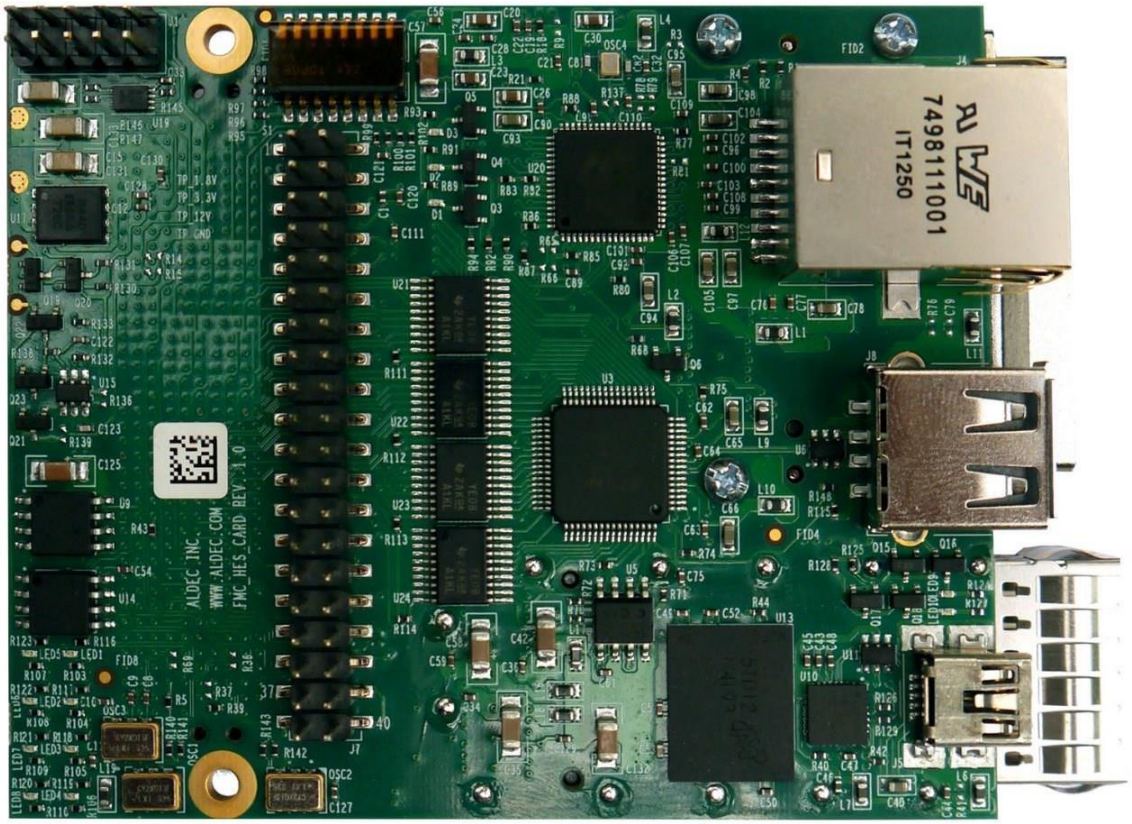
List of daughter cards

• Video Card: This is also referred to as the graphics adapter, display adapter or video adapter.

• Sound Card: To handle sound, to insert a microphone or connect a speaker this sound card is used.

• Network Interface Card: This is also referred as NIC. The computer can be connected to a network only with the use of this network interface card.

• Ethernet Card: Ethernet card is used to connect computers to computers. A cable is used to connect the Ethernet cards in each computer to make a network.


7498111001
IT1250
ALDEC INC.
WWW.ALDEC.COM
FMC_HES_CARD REV. 1.0

4. Bus slots

An expansion slot refers to any of the slots on a motherboard that can hold an expansion card to expand the computer's functionality, like a video card, network card, or sound card.The expansion card is plugged directly into the expansion port so that the motherboard has direct access to the hardware. However, since all computers have a limited number of expansion slots, it's important to open your computer and check what's available before you buy one.Some older systems require the use of a riser board to add additional expansion cards; however, modern computers not only usually have enough expansion slot options, but they also have features integrated directly into the motherboard, eliminating the need for so many expansion cards. There are three different types of expansion slots: PCI Express, PCI, and AGP.

PCI (Peripheral Component Interconnect) Slot: The PCI slot is the most common form of internal expansion for a PC.Some PCs have a mixture of PCI and PCI Express slots.

PCI express (PCIe) Slots: The best type of expansion slot to have in your PC is the PCI Express. The PCI Express type of expansion slot communicates with the motherboard, and therefore with the microprocessor, both quickly and efficiently.

AGP (Accelerated Graphics Port) Slot: This type of expansion slot was specifically designed to deal with graphics adapters. In fact, AGP stands for Accelerated Graphics Port. Older PCs may sport this expansion slot, but the best video cards use PCI Express.

PCI slot

AGP slot

5. SMPS

A switched-mode power supply (SMPS) is an electronic circuit that converts power using switching devices that are turned on and off at high frequencies, and storage components such as inductors or capacitors to supply power when the switching device is in its non-conduction state.

Switching power supplies have high efficiency and are widely used in a variety of electronic equipment, including computers and other sensitive equipment requiring stable and efficient power supply.

A switched-mode power supply is also known as a switch-mode power supply or switching-mode power supply.

Switched-mode power supplies are classified according to the type of input and output voltages. The four major categories are:

- •AC to DC
- •DC to DC
- •DC to AC
- •AC to AC

6. INTERNAL STORAGE DEVICES

Some storage devices are classed as 'internal' which means they are inside the computer case. Most computers have some form of internal storage. The most common type of internal storage is the hard disk. At the most basic level, internal storage is needed to hold the operating system so that the computer is able to access the input and output devices. It will also be used to store the applications software that you use and more than likely, the original copies of your data files. Internal storage allows the data and applications to be loaded very rapidly into memory, ready for use. The data can be accessed much faster than data which is stored on an external storage device. This is because internal storage devices are connected directly to the motherboard and its data bus whereas external devices are connected through a hardware interface such as USB, which means they are considerably slower to access. Internal storage also means that if the computer is moved around, it will still retain its most commonly used data. The main disadvantage of internal storage is that when the hard disk fails (and it will), all the data and applications may be lost. This can be avoided to some extent by using more than one hard disk within the machine. Each hard disk has a copy of all the data, so if one fails the other can carry on. This is called a RAID array. An alternative is to use external drives for backup.

7. Interfacing Ports

A Computer Port is an interface or a point of connection between the computer and its peripheral devices. Some of the common peripherals are mouse, keyboard, monitor or display unit, printer, speaker, flash drive etc. The main function of a computer port is to act as a point of attachment, where the cable from the peripheral can be plugged in and allows data to flow from and to the device.

Types of ports: -

Serial Port: used for external modems and older computer mouse.

Parallel Port: used for scanners and printers.

PS/2 Port: used for old computer keyboard and mouse.

Universal Serial Bus (or USB) Port: It can connect all kinds of external USB devices such as external hard disk, printer, scanner, mouse, keyboard, etc.

VGA Port: connects monitor to a computer's video card. It has 15 holes. Similar to the serial port connector. However, serial port connector has pins, VGA port has holes.

Power Connector: connects to the computer's power cable that plugs into a power bar or wall socket.

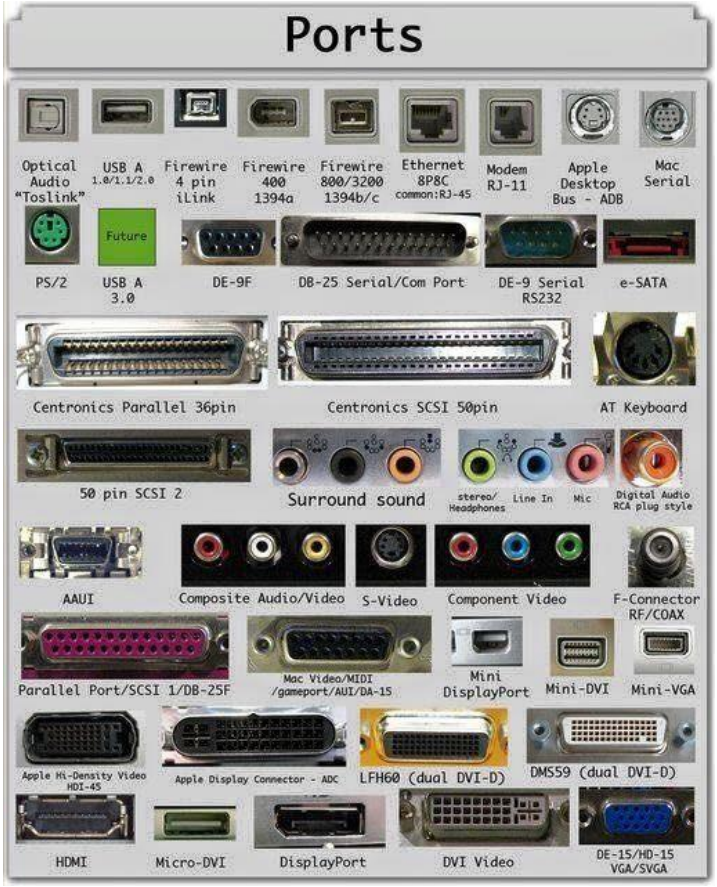
Modem Port: connects a PC's modem to the telephone network.

Ethernet Port: connects to a network and high speed Internet. Connects the network cable to a computer.

Game Port: connect a joystick to a PC. Now replaced by USB Digital Video Interface

DVI port: connects Flat panel LCD monitor to the computer's high-end video graphic cards.

Sockets: sockets connect the microphone and speakers to the sound card of the computer.

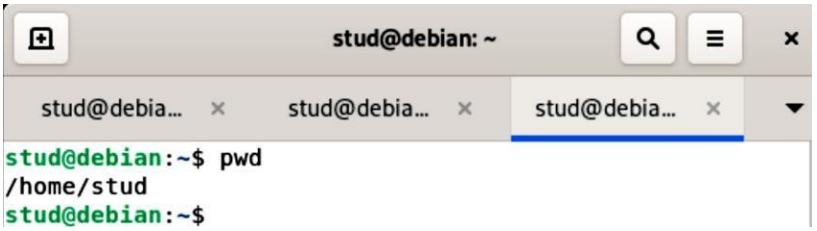

Ports
Optical Audio "Toslink"
USB A 1.0/1.1/2.0
Firewire 4 pin iLink
Firewire 400 1394a
Firewire 800/3200 1394b/c
Ethernet 8P8C common:RJ-45
Modem RJ-11
Apple Desktop Bus - ADB
Mac Serial
PS/2
Future
USB A 3.0
DE-9F
DB-25 Serial/Com Port
DE-9 Serial RS232
e-SATA
Centronics Parallel 36pin
Centronics SCSI 50pin
AT Keyboard
50 pin SCSI 2
Surround sound
stereo/ Headphones
Line In
Mic
Digital Audio RCA plug style
AAUI
Composite Audio/Video
S-Video
Component Video
F-Connector RF/COAX
Parallel Port/SCSI 1/DB-25F
Mac Video/MIDI /gameport/AUI/DA-15
Mini DisplayPort
Mini-DVI
Mini-VGA
Apple Hi-Density Video HDI-45
Apple Display Connector - ADC
LFH60 (dual DVI-D)
DMS59 (dual DVI-D)
HDMI
Micro-DVI
DisplayPort
DVI Video
DE-15/HD-15 VGA/SVGA

EXPERIMENT – 2

LINUX COMMANDS

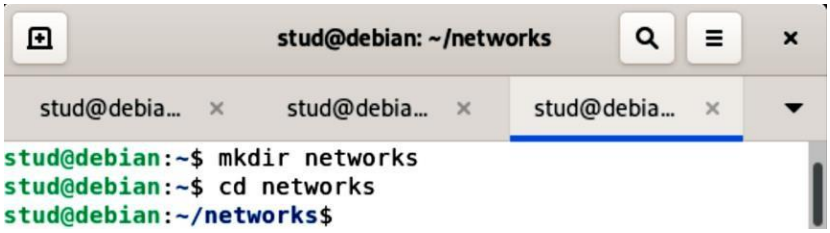
1. **pwd**

   The pwd command is used to display the location of the current working directory



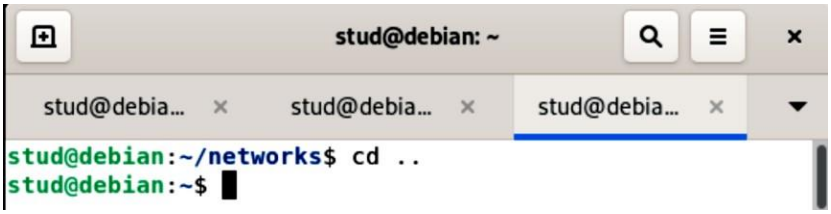
2. **mkdir**

   The mkdir command is used to create a new directory under any directory



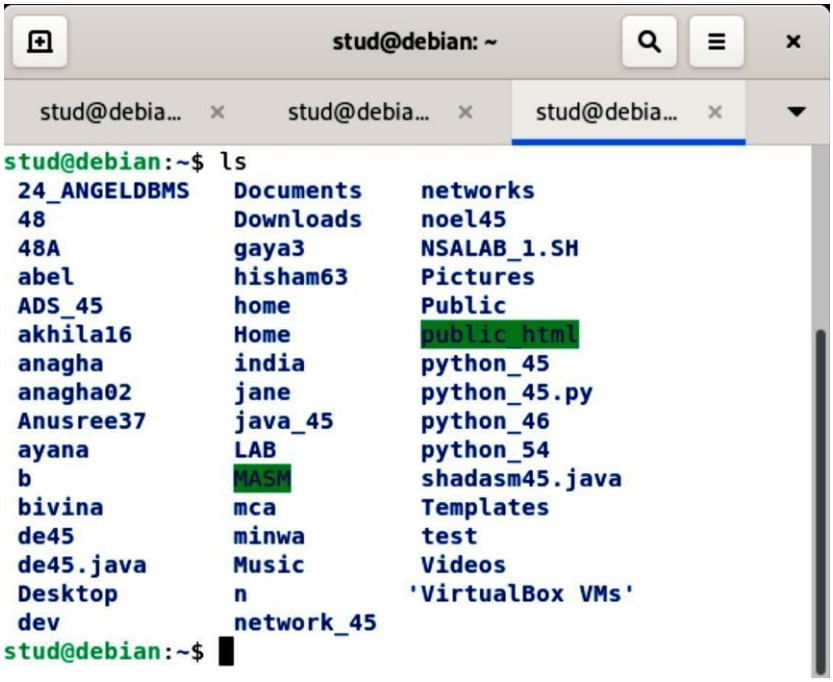
3. **cd**

   The cd command is used to change the current directory



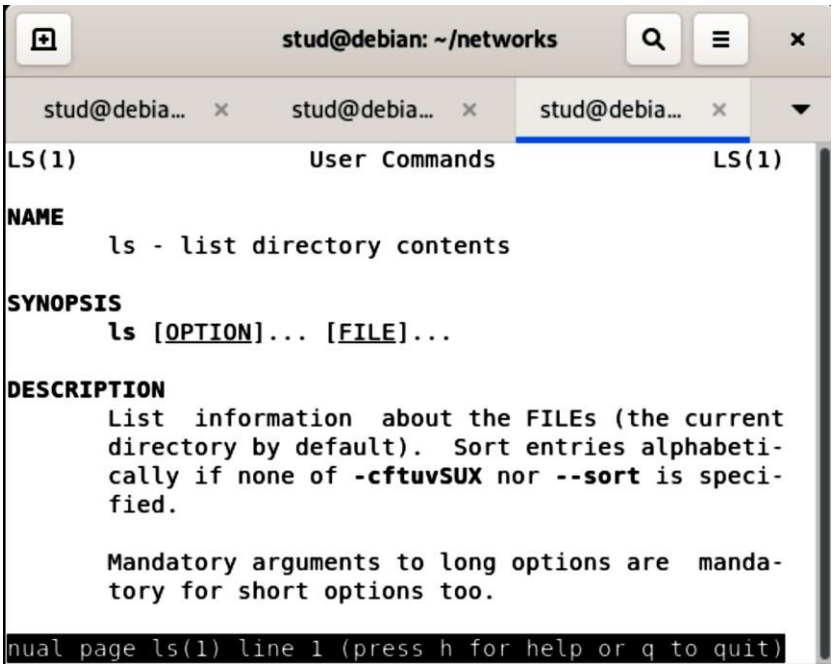
4. ls

The ls command is used to display a list of content of a directory.



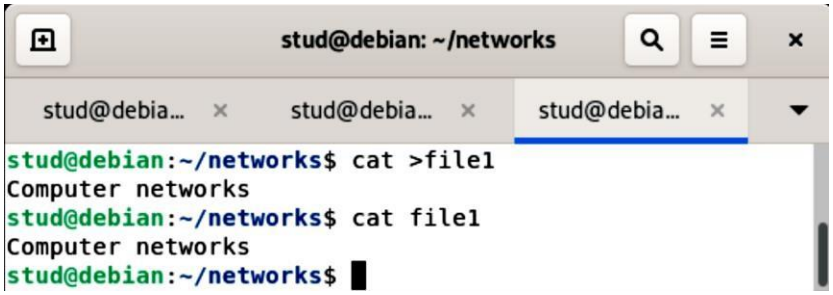
5. man

The man command is used to display the user manual of any command that we can run on the terminal.



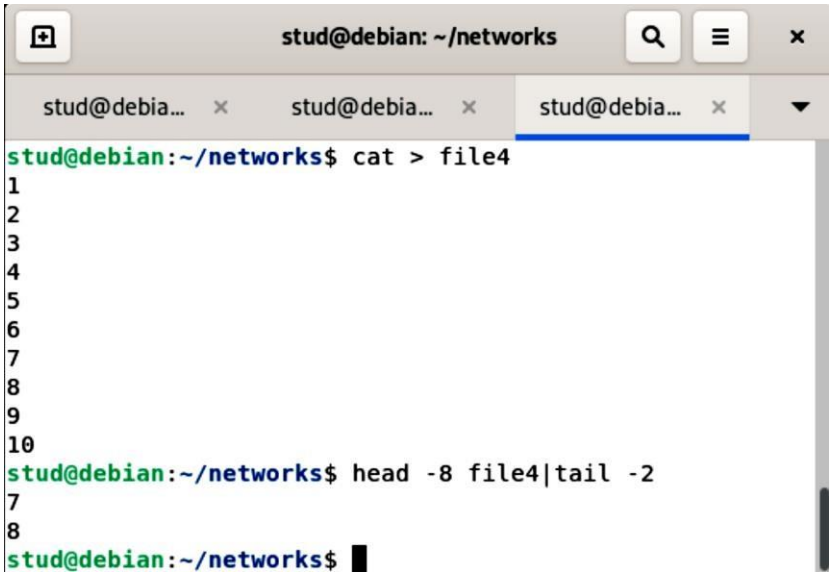
6. cat

The cat command is a multi-purpose utility in the Linux system. It can be used to create a file, display content of the file, copy the content of one file to another file, and more.



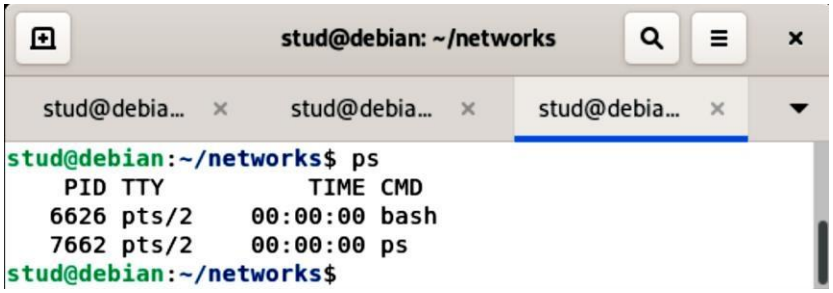
7. head and tail

The head command is used to display the content of a file. It displays the first 10 lines of a file. The tail command is similar to the head command. The difference between both commands is that it displays the last ten lines of the file content. It is useful for reading the error message.



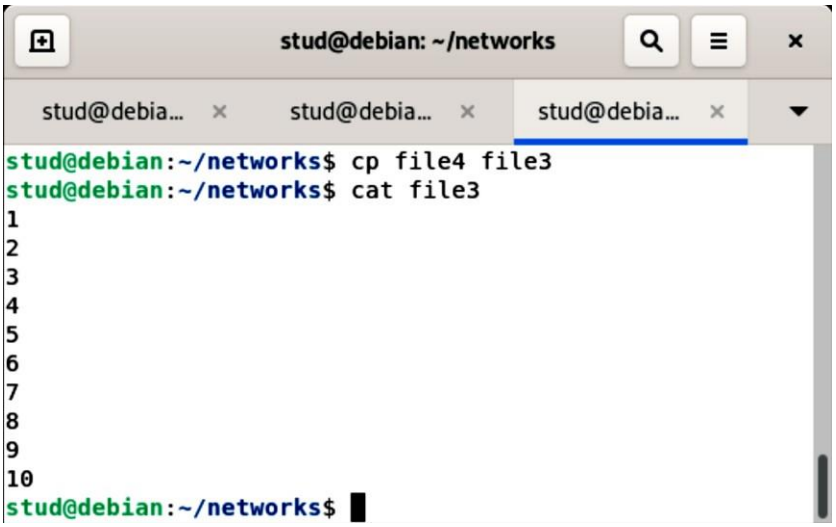
8. ps

The ps command is used to know the currently working process.



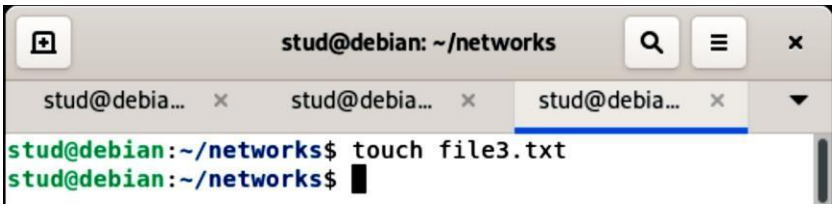
9. cp

The cp command is used to copy a file or directory.



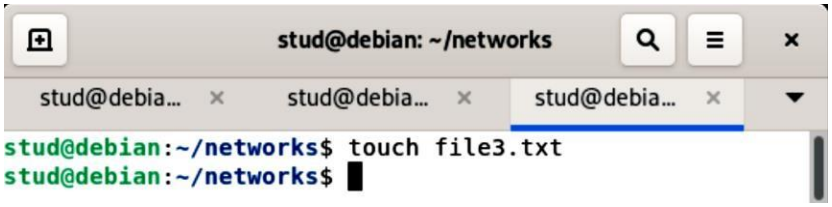
10. Touch

The touch command is used to create empty files. We can create multiple empty files by executing it once.



11. rm

The rm command is used to remove a file.

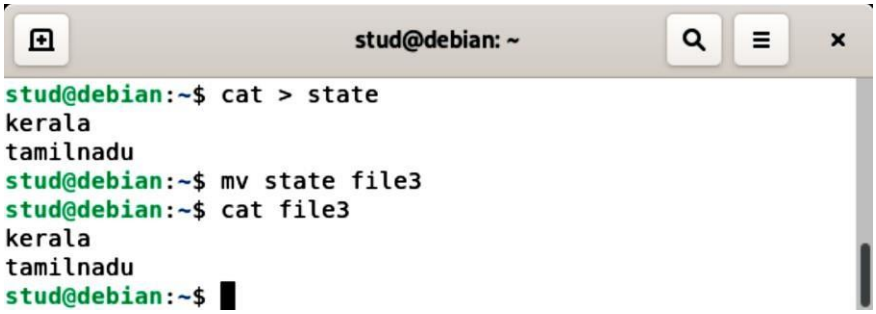


12. rmdir

The rmdir command is used to delete a directory.

13. rename

The rename command is used to rename files. It is useful for renaming a large group of files.



14. more

The more command is quite similar to the cat command, as it is used to display the file content in the same way that the cat command does.

15. less

The less command is similar to the more command. It also includes some extra features such as 'adjustment in width and height of the terminal.

16. sort

The more command is quite similar to the cat command, as it is used to display the file content in the same way that the cat command does.

17. wc

The wc command is used to count the lines, words, and characters in a file.

18. sed

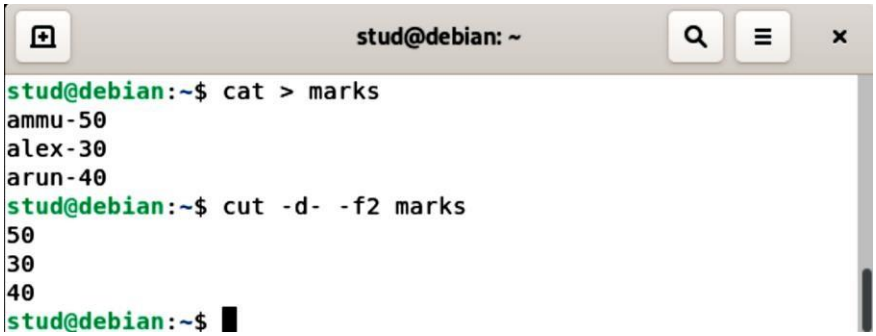
The sed command is used to edit files using a regular expression. It does not permanently edit files.

19. grep

The grep is useful for searching the content from a file. Generally, it is used with the pipe.

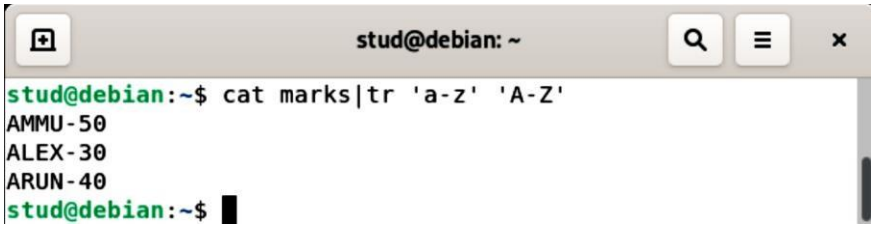
20. cut

The cut command is used to select a specific column of a file. The '-d' option is used as a delimiter, and it can be a space (' '), a slash (/), a hyphen (-), or anything else. And, the '-f' option is used to specify a column number.



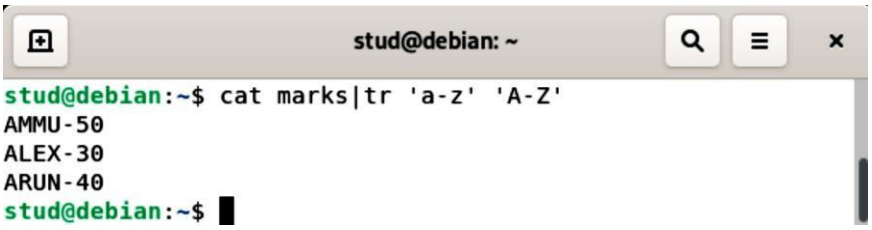
21. tr

The tr command is used to translate the file content like from lower case to upper case.



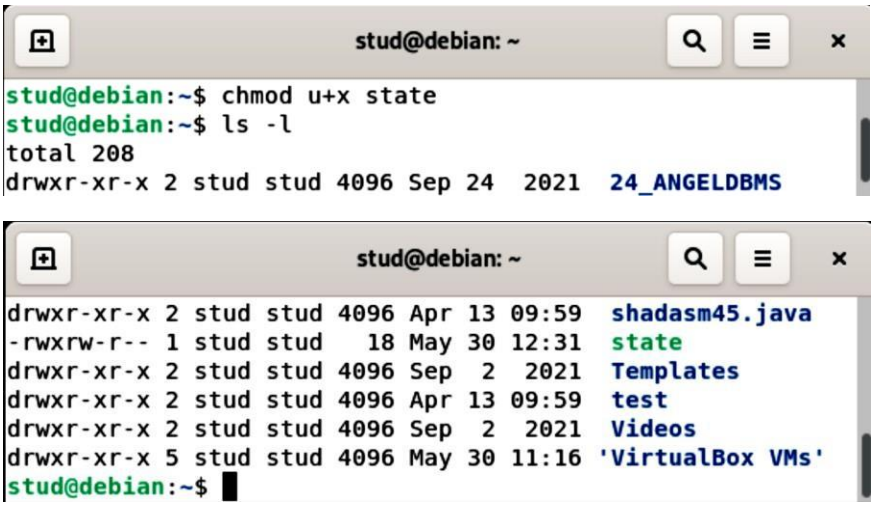
22. paste

Pasteis used to join files horizontally (parallel merging) by outputting lines consisting of lines from each file specified.



23. ch mod

chmod command is used to change the access permissions of files and directories.



EXPERIMENT – 3

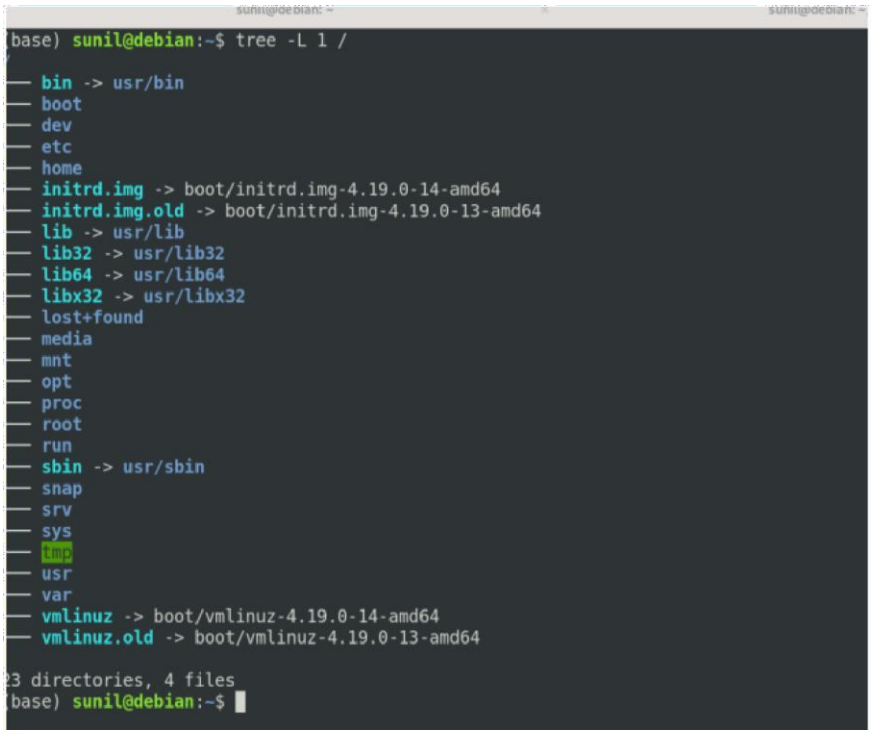
LINUX FILE SYSTEM

It makes sense to explore the Linux filesystem from a terminal window, In fact, that is the name of the first tool you'll install to help you on the way: tree. If you are using Ubuntu or Debian, you can do: sudo apt install tree Once installed, stay in your terminal window and run tree like this: $ tree /.

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

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

Instead, try this: tree -L 1 / And you should see a listing similar to what is shown in Figure 1.



The instruction above can be translated as “show me only the 1st Level of the directory tree starting at / (root)“. The -L option tells tree how many levels down you want to see. Most Linux distributions will show you the same or a very similar layout to what you can see in the image above. This means that even if you feel confused now, master this, and you will have a handle on most, if not all, Linux installations in the whole wide world.

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

Directories

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

/bin

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

/boot

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

/dev

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

/etc

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

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

/home

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

/lib

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

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

/media

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

/mnt

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

/opt

The /opt directory is often where software you compile (that is, you build yourself from source code and do not install from your distribution repositories) sometimes lands. Applications will end up in the /opt/bin directory and libraries in the /opt/lib directory. A slight digression: another place where applications and libraries end up in is /usr/local, When software gets installed here, there will also be /usr/local/bin and /usr/local/lib directories. What determines which software goes where is how the developers have configured the files that control the compilation and installation process.

/proc

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

/root

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

/run

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

/sbin

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

/usr

The /usr directory was where users' home directories were originally kept back in the early days of UNIX. However, now /home is where users kept their stuff as we saw above. These days, /usr contains a mish-mash of directories which in turn contain applications, libraries, documentation, wallpapers, icons and a long list of other stuff that need to be shared by applications and services. You will also find bin, sbin and lib directories in /usr. What is the difference with their roothanging cousins? Not much nowadays. Originally, the /bin directory (hanging off of root) would contain very basic commands, like ls, mv and rm; the kind of commands that would come pre-installed in all UNIX/Linux installations, the bare minimum to run and maintain a system. /usr/bin on the other hand would contain stuff the users would install and run to use the system as a work station, things like word processors, web browsers, and other apps. But many modern Linux distributions just put everything into /usr/bin and have /bin point to /usr/bin just in case erasing it completely would break something. So, while Debian, Ubuntu and Mint still keep /bin and /usr/bin (and /sbin and /usr/sbin) separate; others, like Arch and its derivatives just have one "real" directory for binaries, /usr/bin, and the rest or *bins are "fake" directories that point to /usr/bin.

/srv

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

/sys

/sys is another virtual directory like /proc and /dev and also contains information from devices connected to your computer. In some cases you can also manipulate those devices. I can, for example, change the brightness of the screen of my laptop by modifying the value

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

/tmp

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

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

/var

/var was originally given its name because its contents was deemed variable, in that it changed frequently. Today it is a bit of a misnomer because there are many other directories that also contain data that changes frequently, especially the virtual directories we saw above. Be that as it may, /var contains things like logs in the /var/log subdirectories. Logs are files that register events that happen on the system. If something fails in the kernel, it will be logged in a file in /var/log; if someone tries to break into your computer from outside, your firewall will also log the attempt here. It also contains spools for tasks. These "tasks" can be the jobs you send to a shared printer when you have to wait because another user is printing a long document, or mail that is waiting to be delivered to users on the system.

Your system may have some more directories we haven't mentioned above. In the screenshot, for example, there is a /snap directory. That's because the shot was captured on an Ubuntu system. Ubuntu has recently incorporated snap packages as a way of distributing software. The /snap directory contains all the files and the software installed from snaps.

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

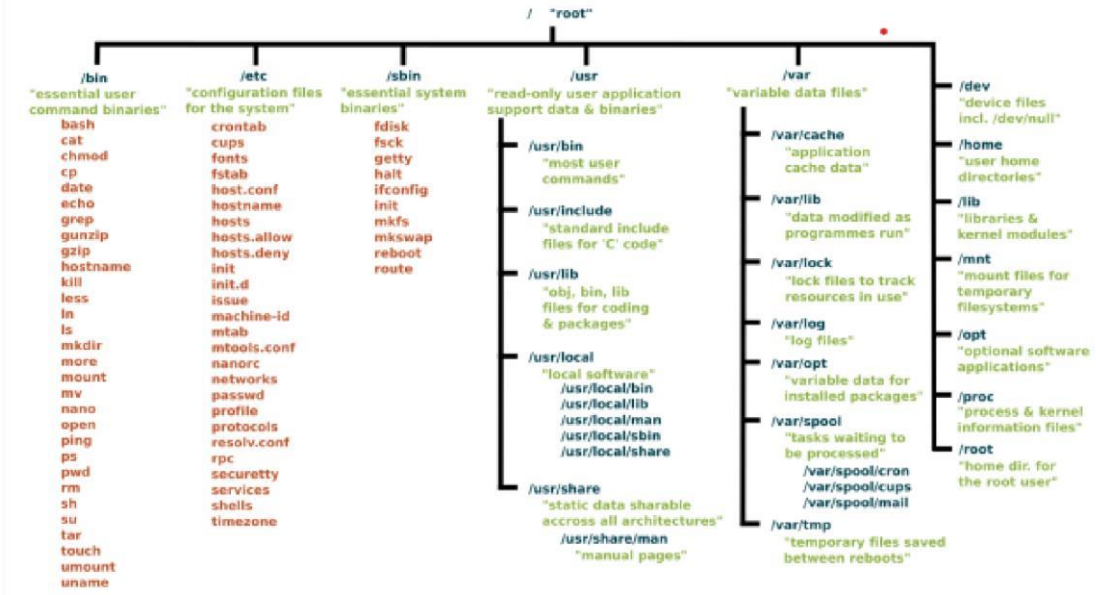


Figure 2: Standard Unix filesystem hierarchy.

To explore the filesystem yourself, use the cd command:

cd

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

pwd

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

cd ..

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

To see what a directory contains, use

ls

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

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

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

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

EXPERIMENT - 4

SHELL PROGRAMMING

Experiment no.1

Write a shell script to find the sum, the average and the product of the four integers entered.

Shell Script

```
echo "Enter four integers"
read a b c d
sum=`expr $a + $b + $c + $d`
echo "Sum      = $sum"
avg=`expr $sum / 4`
echo "Average = $avg"
pro=`expr $a \* $b \* $c \* $d`
echo "Product = $pro"
```

Output

Experiment no.2

a) Write a program to check whether a number entered is odd or even.

Shell Script

```
echo "enter a number"
read a
if [ `expr $a % 2` -eq 0 ]
then
echo "even number"
else
echo "odd number"
fi
```

b) Write a shell script to print given number in reverse order.

Shell Script

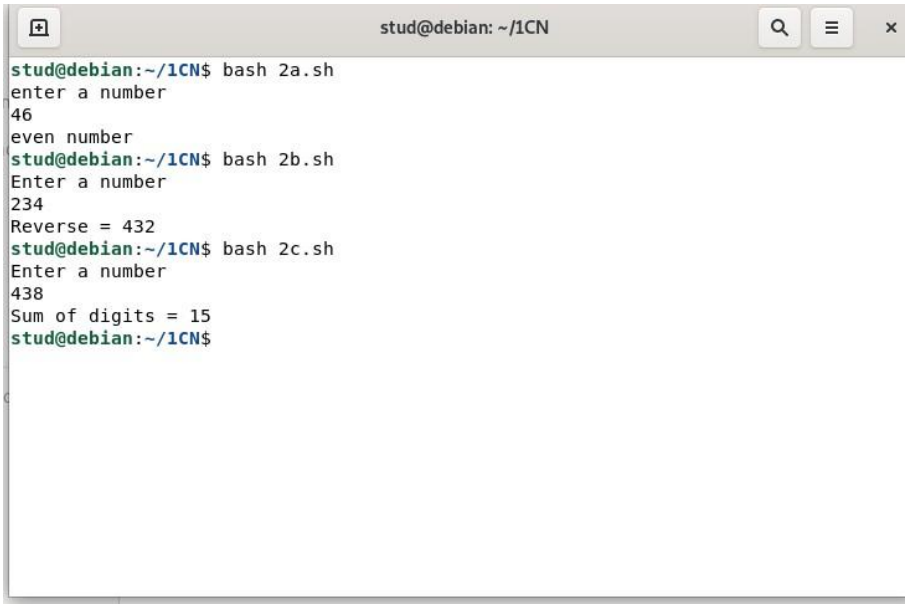
```
echo "Enter a number"
read s
r=0
d=0
while [ $s -gt 0 ]
do
d=$(($s % 10))
r=$(($r * 10 + $d))
s=$(($s /  10))
done
echo "Reverse = $r"
```

c) Write a shell script to print sum of all digits of a given number

Shell Script

```
echo "Enter a number"
read s
r=0
d=0
while [ $s -gt 0 ]
do
d=$(($s % 10))
r=$(($r + $d))
s=$(($s / 10))
done
echo "Sum of digits = $r"
```

Output



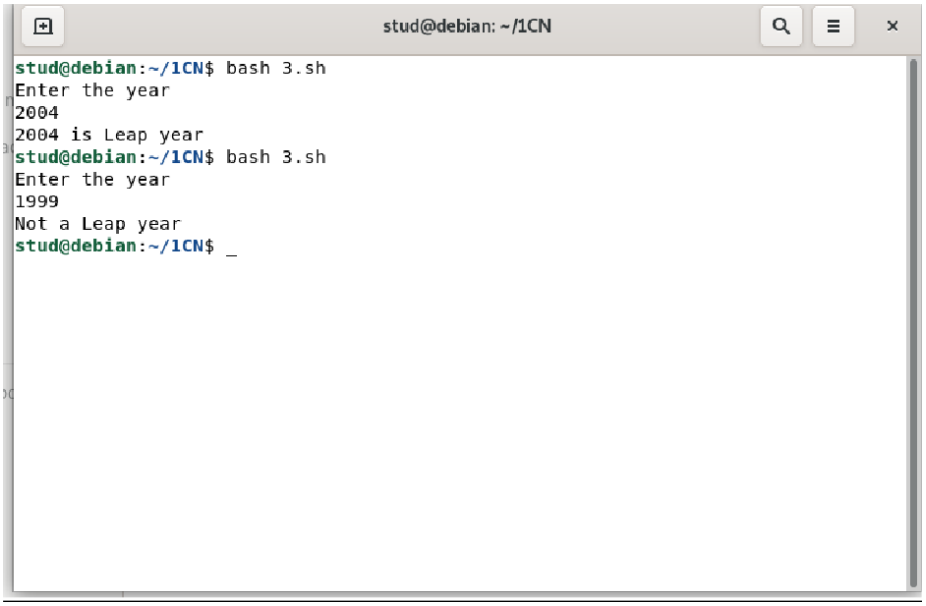
Experiment no.3

Write a shell script that accepts any year from the keyboard and determine whether the year is a leap year or not.

Shell Script

```
echo "Enter the year"
read y
if [ `expr $y % 4` -eq 0 ] && [ `expr $y % 100` -ne 0 ] || [ `expr $y % 400` -eq 0 ]
then
echo "$y is Leap year"
else
echo "Not a Leap year"
fi
```

Output



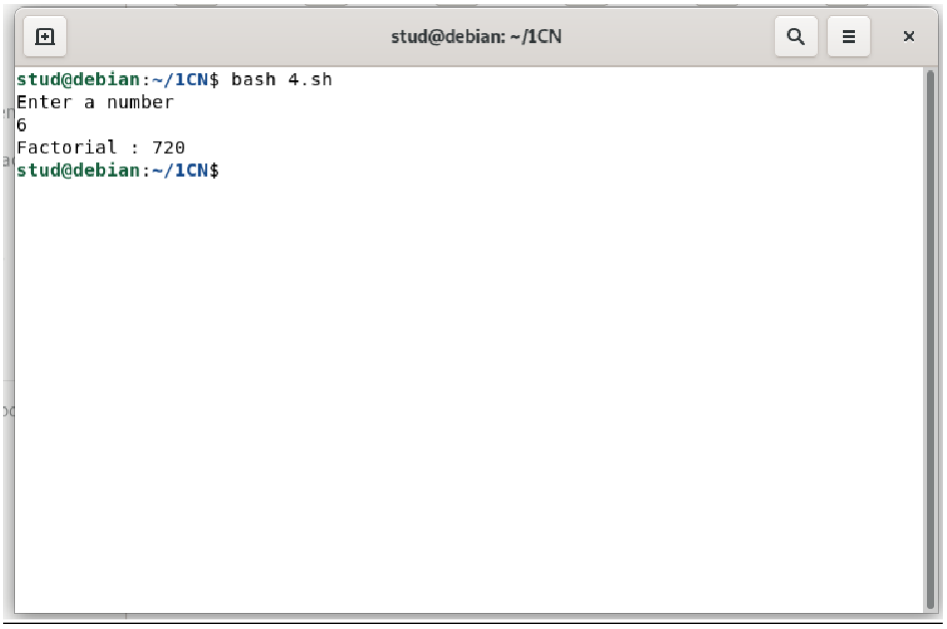
Experiment no.4

Write a shell script to find the factorial of a number

Shell Script

```
echo "Enter a number"
read n
i=1
f=1
while [ $i -le $n ]
do
f=$(( $f * $i ))
i=$(( $i + 1 ))
done
echo "Factorial : $f"
```

Output



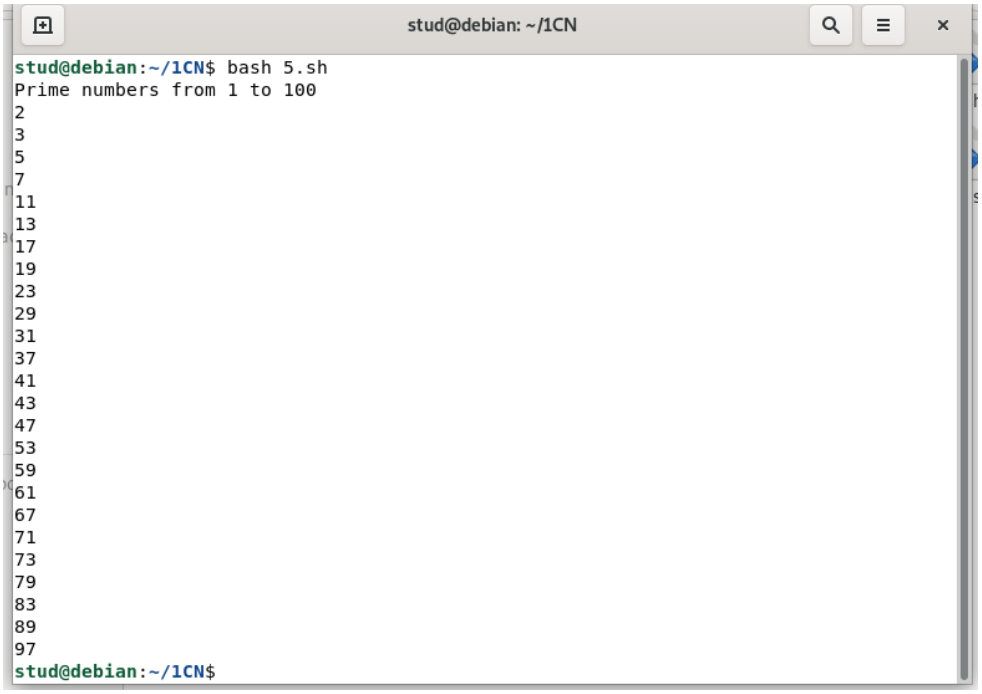
Experiment no.5

Write a program to print prime numbers from 1 to 100

Shell Script

```
echo "Prime numbers from 1 to 100"
for (( i=2; i<=100; i++ ))
do
   f=0
   for (( j=2; j<$i; j++ ))
   do
   x=`expr $i % $j`
        if [ $x -eq 0 ]
        then
             f=1
             break
        fi
   done
   if [ $f -eq 0 ]
   then
        echo $i
   fi
done
```

Output


stud@debian: ~/1CN
stud@debian:~/1CN$ bash 5.sh
Prime numbers from 1 to 100
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
53
59
61
67
71
73
79
83
89
97
stud@debian:~/1CN$

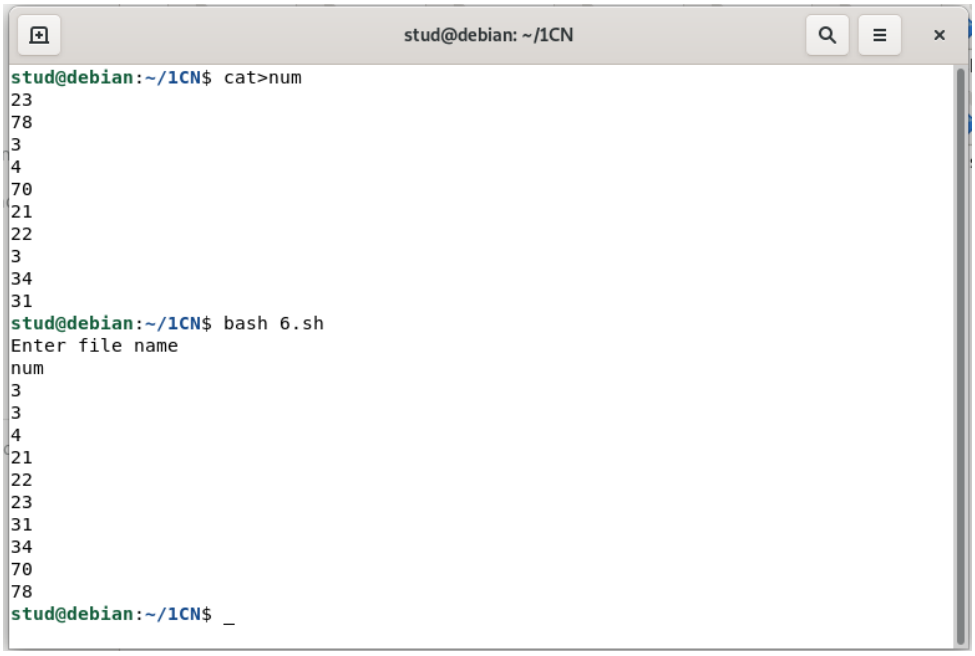
Experiment no.6

Write a script for Printing Numbers in ascending Order

Shell Script

```
echo "Enter file name"
read f
sort -n $f>f1
cat f1
```

Output



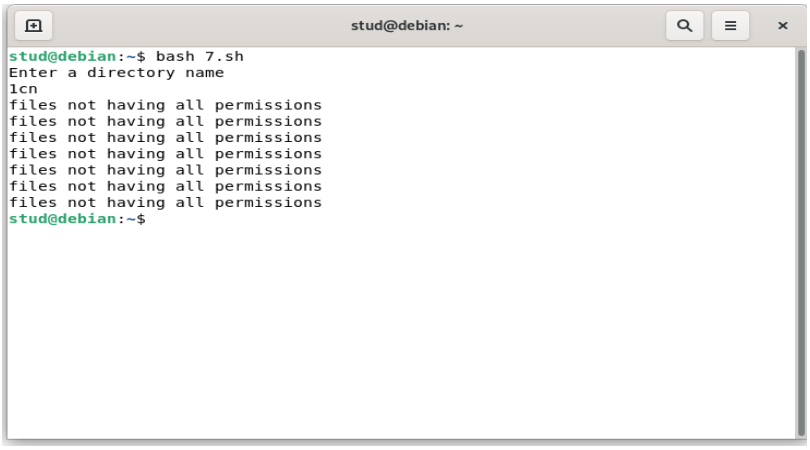
Experiment no.7

Write a shell script which displays a list of files in the current directory to which we have read write and execute permissions.

Shell Script

```
read dir
if [ -d $dir ]
then
cd $dir
ls > f
exec<f
while read line
do
if [ -f $line ]
then
if [ -r $line -a -w $line -a -x $line ]
then
echo "$line has all permissions"
else
echo "files not having all permissions"
fi
fi
done
fi
```

Output


stud@debian: ~
stud@debian:~$ bash 7.sh
Enter a directory name
1cn
files not having all permissions
files not having all permissions
files not having all permissions
files not having all permissions
files not having all permissions
files not having all permissions
files not having all permissions
stud@debian:~$

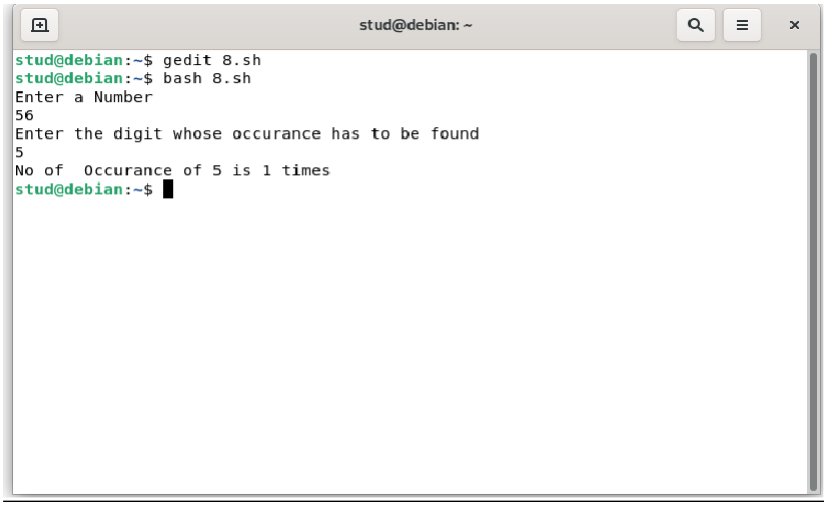
Experiment no.8

Write a shell script to find the number of occurrence of particular digit in an inputted number

Shell Script

```
c=0
echo "Enter a Number "
read n
x=$n
echo "Enter the digit whose occurance has to be found "
read a
while [ $n -ne 0 ]
do
   d=`expr $n % 10`
   if [ $d -eq $a ]
   then
         c=`expr $c + 1`
   fi
   n=`expr $n / 10 `
done
echo "No of  Occurance of $d is $c times"
```

Output



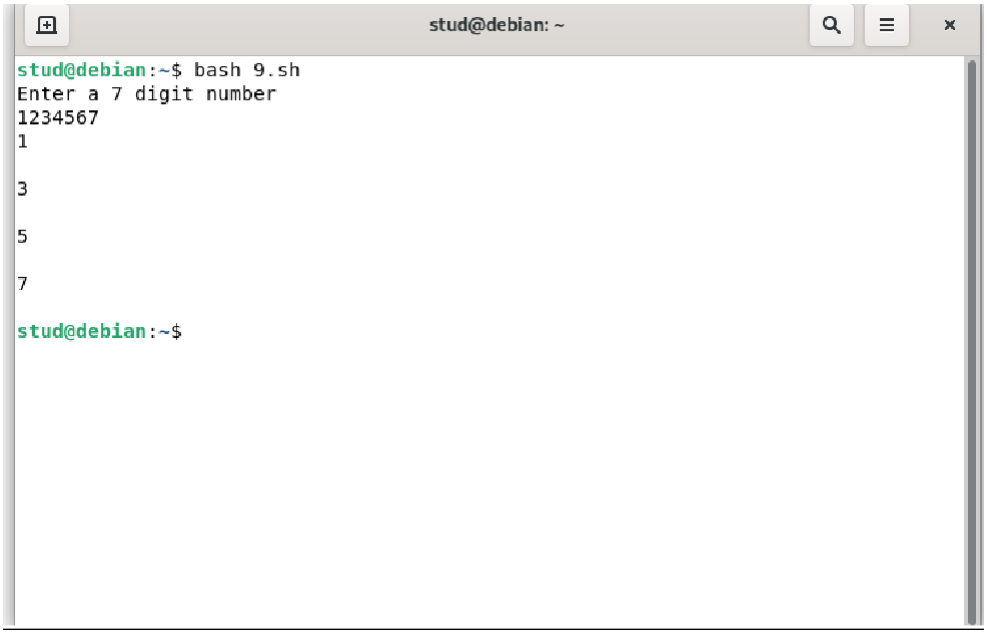
Experiment no.9

Write shell script to print alternate digits when a 7-digit number is passed as input

Shell Script

```
echo "Enter a 7 digit number"
read num
n=1
while [ $n -le 7 ]
do
   a= echo $num | cut -c $n
   echo $a
   n=`expr $n + 2`
done
```

Output



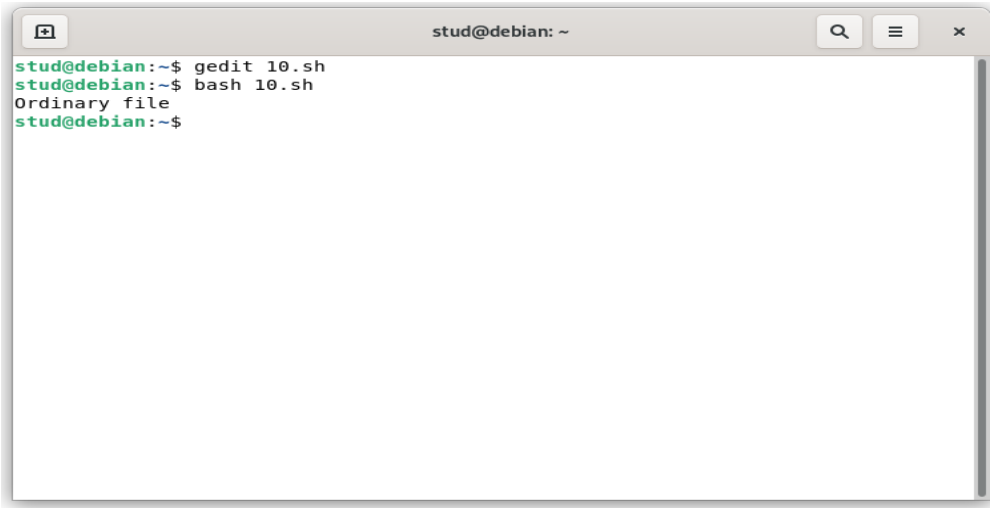
Experiment no.10

Write A shell script that takes a command –line argument and reports on whether it is directory, a file, or something else

Shell Script

```
if [ -f $1 ]
then
   echo "Ordinary file"
elif [ -d $1 ]
then
   echo "Directory file"
else
echo "Does not exists"
fi
```

Output



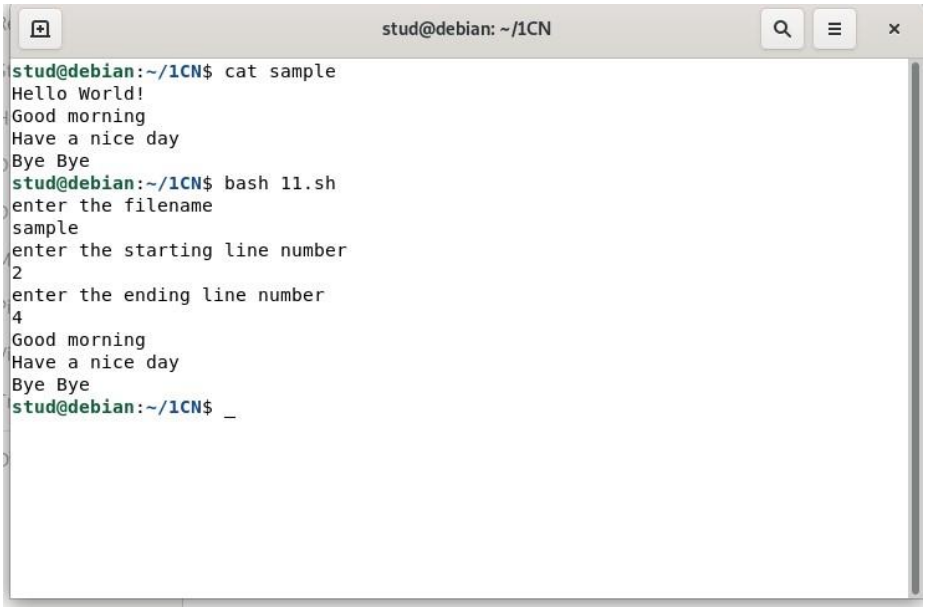
Experiment no.11

Write a shell script that accepts a file name starting and ending line numbers as arguments and displays all the lines between the given line numbers

Shell Script

```
echo "enter the filename"
read fname
echo "enter the starting line number"
read s
echo "enter the ending line number"
read n
sed -n $s,$n\p $fname |cat>new
cat new
```

Output



Experiment no.12

Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.

Shell Script

```
if [ $# -
lt 2 ]
then
      echo "not enough
      arguments"exit 1
fi

search_word=$1

for file_name in
"$@"do
      if [ "$file_name" =
      "$1" ]then
      conti
      nuefi
      echo $file_name

      echo "....."if [ ! -f $file_name ] then
       echo "File \"$file_name\" does not
       exist"exit 2
      fi

       sed "/$search_word/d" $file_name

done
```

Output

Experiment no.13

Write a shell script that computes the gross salary of an employee according to the following:

1) if basic salary is <1500 then HRA 10% of the basic and DA =90% of

the basic

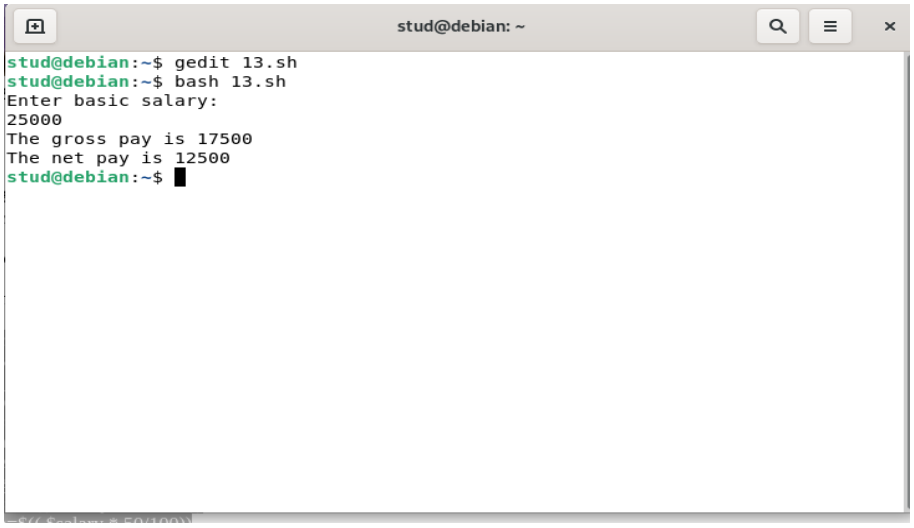
2) if basic salary is >1500 then HRA 500 and DA =98% of the basic

The basic salary is entered interactively through the key board

Shell Script

```
echo "Enter basic salary:"
read salary
if [ $salary -lt 15000 ]
then
hra=$(( $salary * 10/100 ))
da=$(( $salary * 40/100 ))
ded=$(( $salary * 10/100 ))
else
hra=$(( $salary * 20/100 ))
da=$(( $salary * 50/100))
ded=$(( $salary *20/100 ))
fi
gp=$(( $hra + $da ))
np=$(( $gp - $ded))
echo "The gross pay is $gp"
echo "The net pay is $np"
```

Output


stud@debian: ~
stud@debian:~$ gedit 13.sh
stud@debian:~$ bash 13.sh
Enter basic salary:
25000
The gross pay is 17500
The net pay is 12500
stud@debian:~$

EXPERIMENT – 5

Installing LAMP on Ubuntu

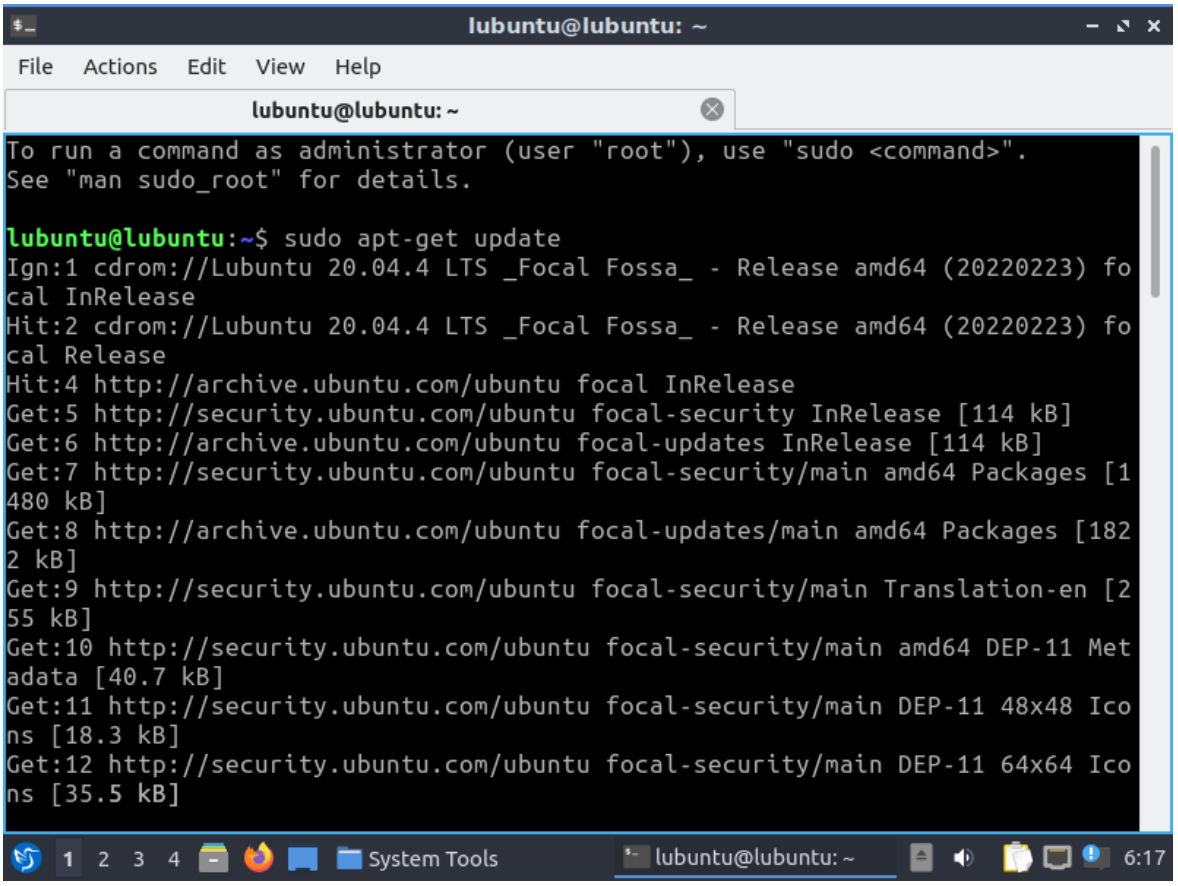
Step 1: Update Package Repository Cache

Before you begin:

1. Open the terminal either by using the **CTRL+ALT+T** keyboard shortcut or by searching forthe word *terminal* in **Ubuntu**

2. Make sure to update the package repository cache to ensure it installs the latest versions of the software. To do so, type in the following command:

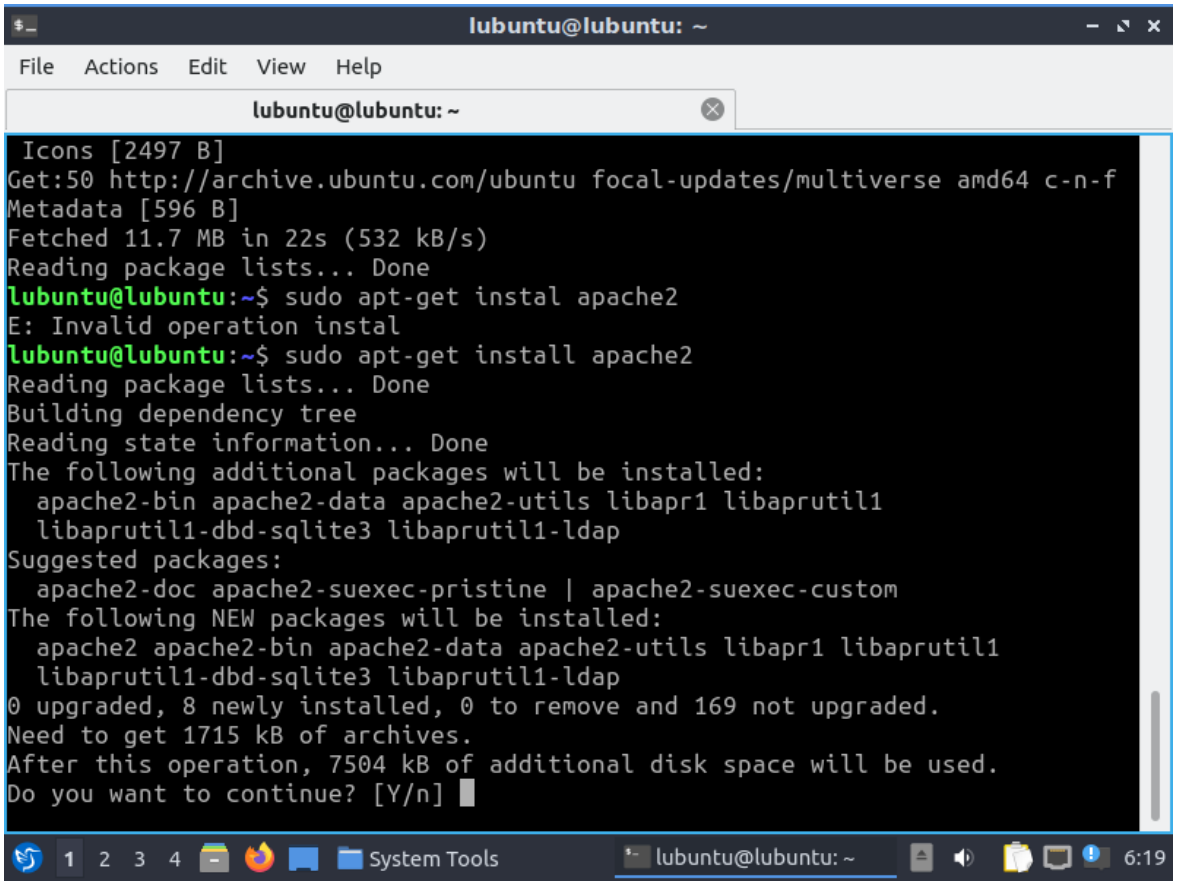
sudo apt-get update



Step 2: InstallApache

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

   sudo apt-get install apache2



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

Step 3: InstallPHP

1. To install PHP, run the following command:

$ sudo apt-get install php7.4

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

Step 4: RestartApache

After the php installation you must restart the Apache service. Enter the command:

$ sudo /etc/init.d/apache2 restart

Step 5: Test PHP Processing on Web Server

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

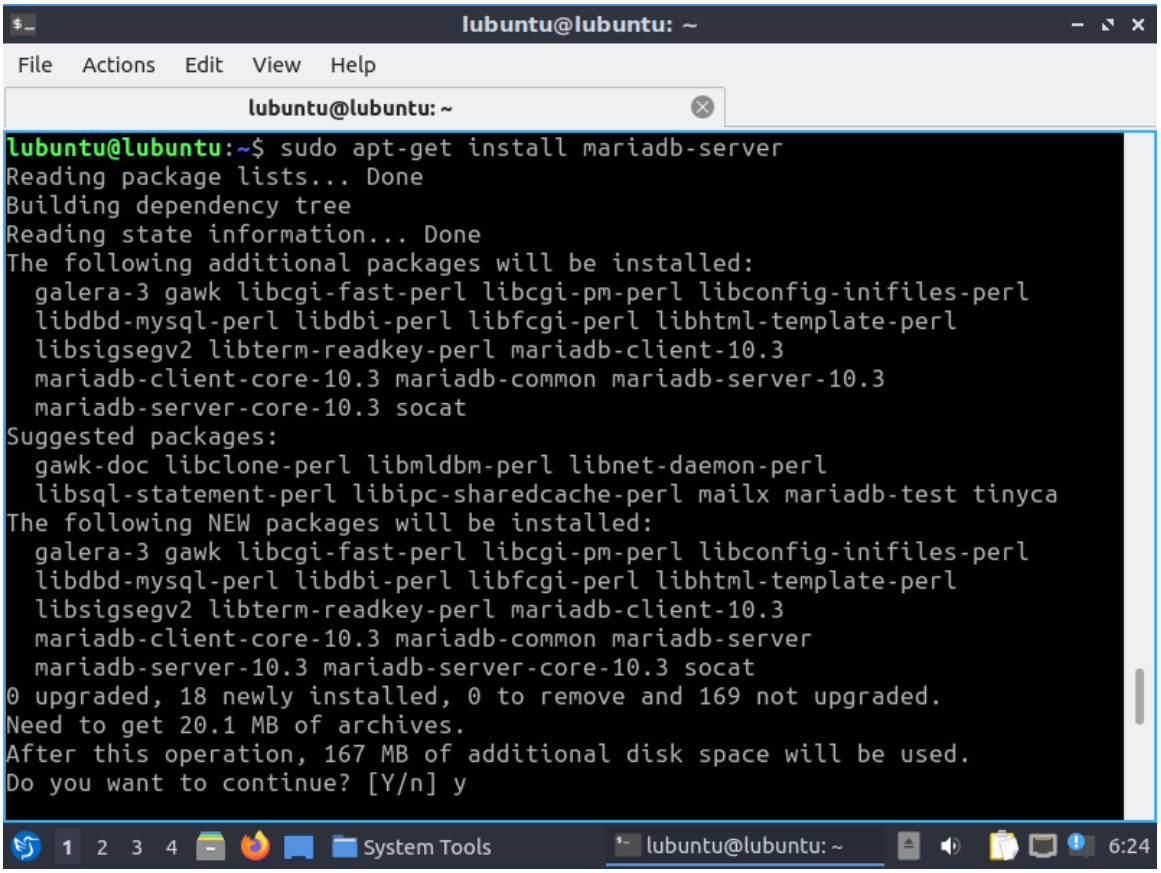
To create a file in that directory, type in the following command:

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

Step 6: Install mariadb server

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

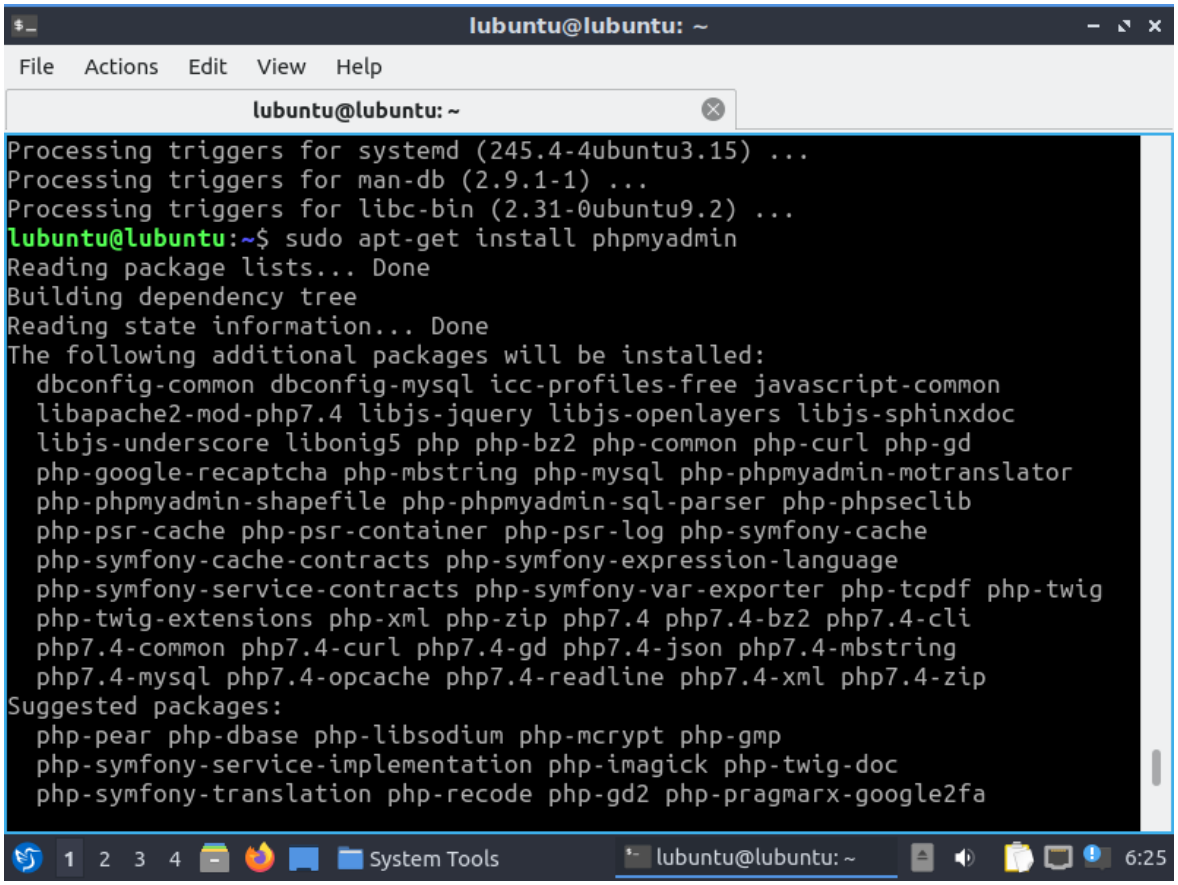
$ sudo apt-get install mariadb-server


lubuntu@lubuntu: ~
File Actions Edit View Help
lubuntu@lubuntu: ~
lubuntu@lubuntu:~$ sudo apt-get install mariadb-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
galera-3 gawk libcgi-fast-perl libcgi-pm-perl libconfig-inifiles-perl
libdbd-mysql-perl libdbi-perl libfcgi-perl libhtml-template-perl
libsigsegv2 libterm-readkey-perl mariadb-client-10.3
mariadb-client-core-10.3 mariadb-common mariadb-server-10.3
mariadb-server-core-10.3 socat
Suggested packages:
gawk-doc libclone-perl libmldbm-perl libnet-daemon-perl
libsql-statement-perl libipc-sharedcache-perl mailx mariadb-test tinyca
The following NEW packages will be installed:
galera-3 gawk libcgi-fast-perl libcgi-pm-perl libconfig-inifiles-perl
libdbd-mysql-perl libdbi-perl libfcgi-perl libhtml-template-perl
libsigsegv2 libterm-readkey-perl mariadb-client-10.3
mariadb-client-core-10.3 mariadb-common mariadb-server
mariadb-server-10.3 mariadb-server-core-10.3 socat
0 upgraded, 18 newly installed, 0 to remove and 169 not upgraded.
Need to get 20.1 MB of archives.
After this operation, 167 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
1 2 3 4 System Tools lubuntu@lubuntu: ~ 6:24

Step8:InstallPHPMyadmin

1. To install PHP Myadmin, run the followingcommand:

   $ sudo apt-get install phpMyAdmin



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

1. Then its ask what type of server, we have Apache2 is set by default that's what we want thenpress
   ok
2. Then a configuration prompt are open . here we're going to just choose yes and then it ask the input
   password for phpmyadmin
3. Then check it currect . go to the localhost/phpmyadmin.
   Here we can not found it so We have to actually edit the file
   php is located in Apache2 folder.
4. Enter the following command to edit the file
   $ sudo nano/etc/php7.4/apache2.php.ini

5. Then we need to uncomment an **extension=mysql.so.** find it the file just remove the Semicolon.

6. Then enter ctl+x to save

Step 9: RestartApache

After the php installation you must restart the Apache service. Enter the command:

$ sudo /etc/init.d/apache2 restart

Step9.1: Include phpMyAdmin in apache configuration

1. Enter the command:

   $ sudo nano/etc/apache2/apache2.conf

2. Type the following command to the nano editor

   Include /etc/phpmyadmin/apache.conf

3. Then enter ctl+x to save.

4. Then again restart the apache

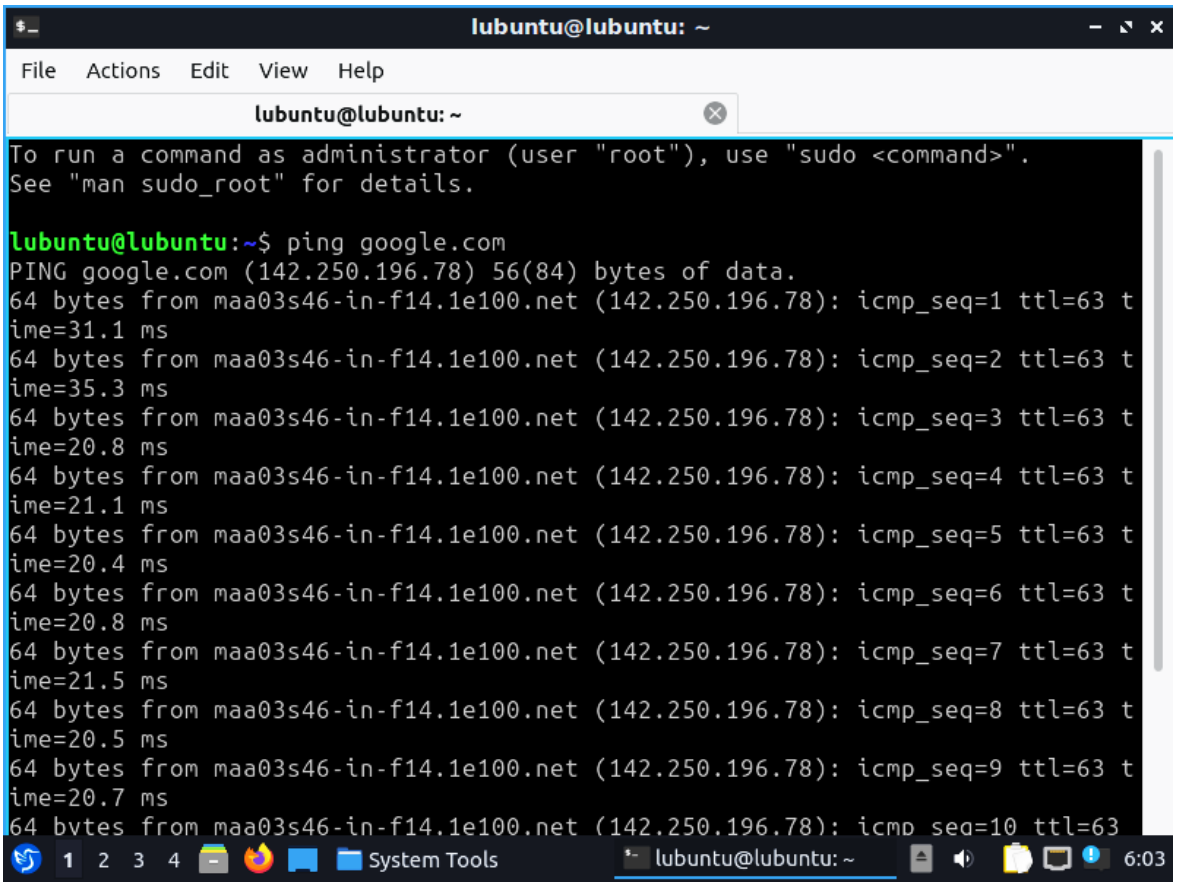
EXPERIMENT – 6

NETWORKING COMMANDS

PING COMMAND

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

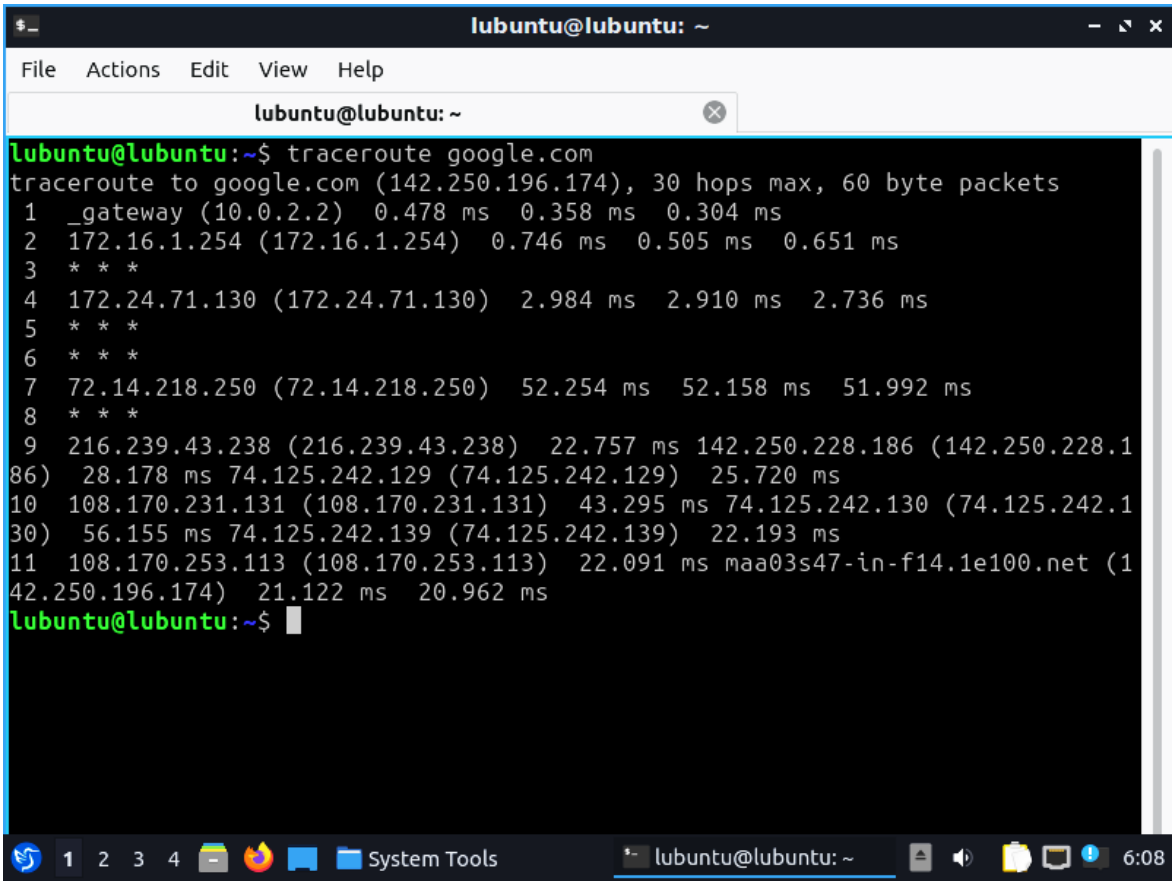
Syntax: ping [OPTIONS] DESTINATION



TRACEROUTE COMMAND

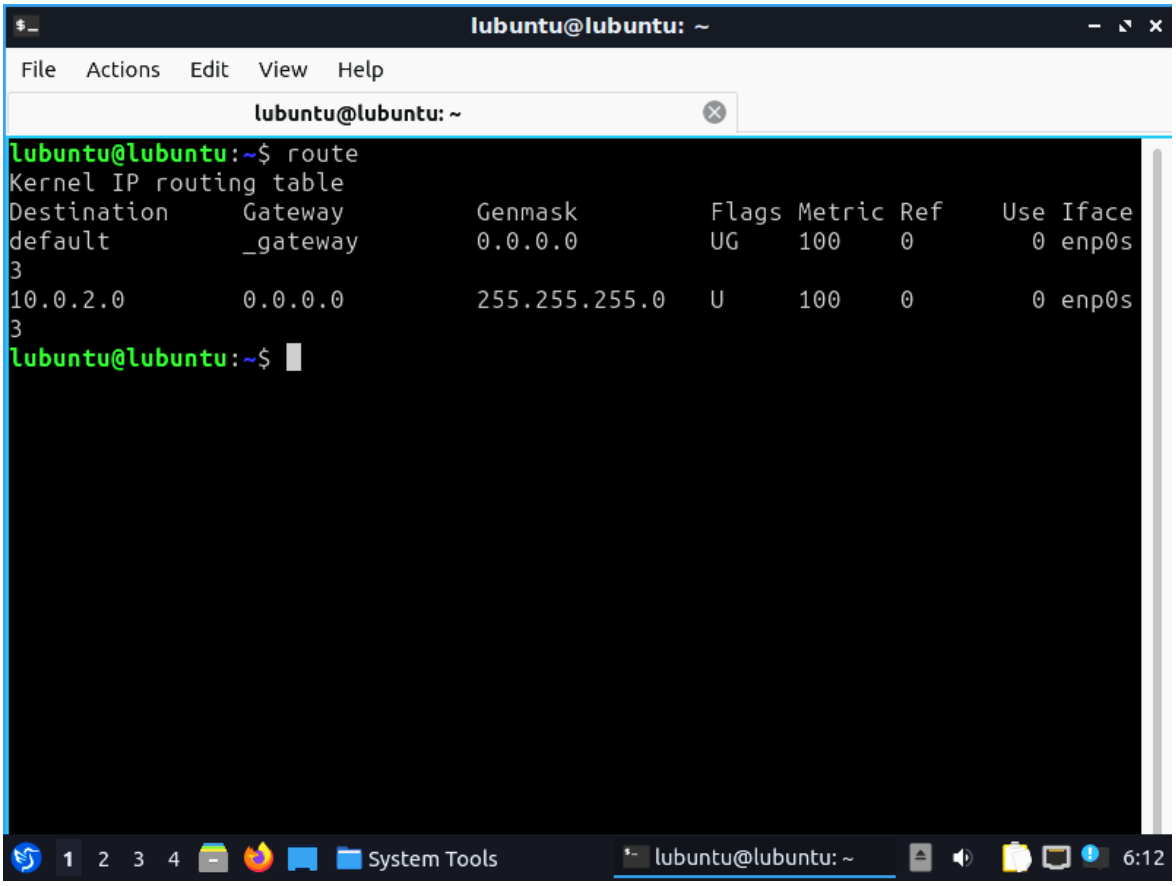
Traceroute command in Linux prints the route that a packet takes to reach the host. This command is useful when you want to know about the route and about all the hops that a packet takes. Below image depicts how traceroute command is used to reach the Google (172.217.26.206) host from the local machine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

Syntax: traceroute [options] host_Address [pathlength]



ROUTE COMMAND

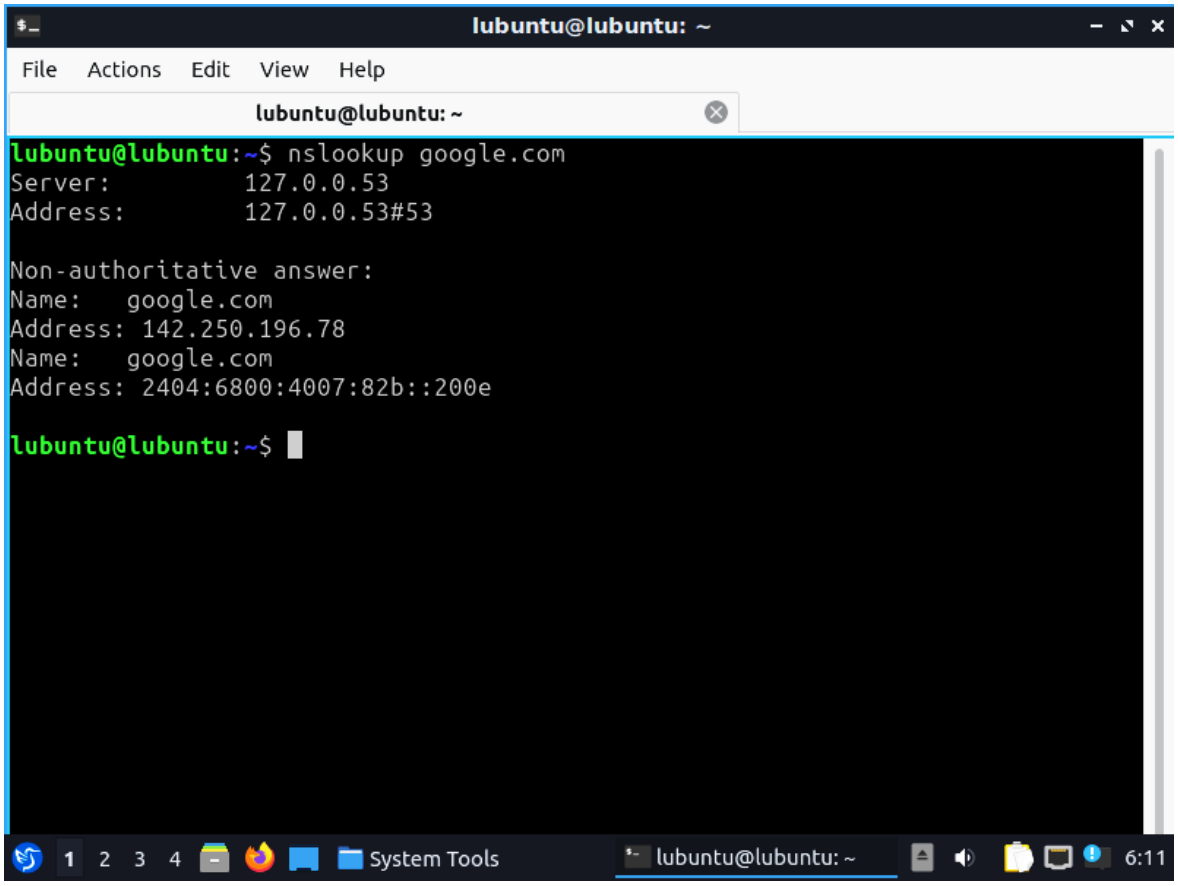
route command in Linux is used when you want to work with the IP/kernel routing table. It is mainly used to set up static routes to specific hosts or networks via an interface. It is used for showing or update the IP/kernel routing table.



NSLOOKUP COMMAND

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

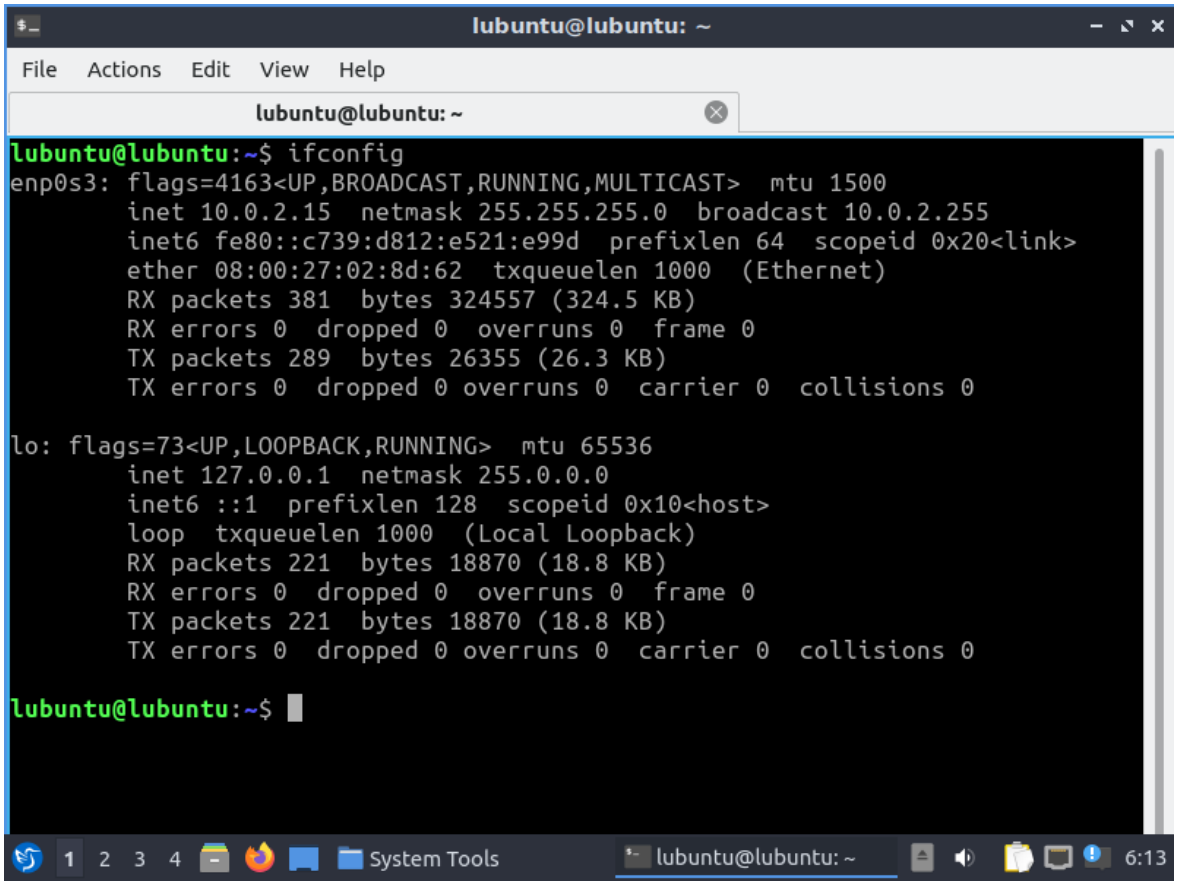
Syntax: nslookup [option]



IFCONFIG COMMAND

ifconfig(interface configuration) command is used to configure the kernel-resident network interfaces. It is used at the boot time to set up the interfaces as necessary. After that, it is usually used when needed during debugging or when you need system tuning. Also, this command is used to assign the IP address and netmask to an interface or to enable or disable a given interface.

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

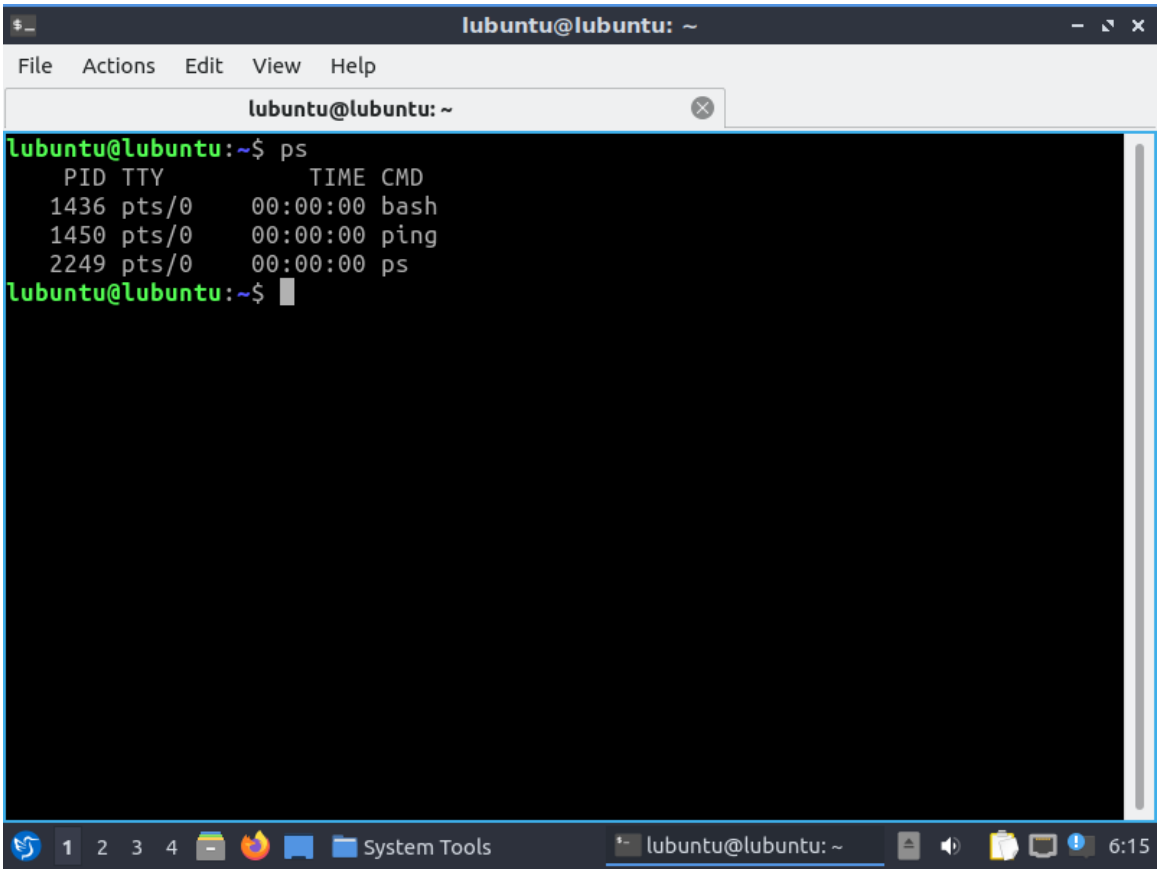


PS COMMAND

Linux provides us a utility called ps for viewing information related with the processes on a system which stands as abbreviation for “Process Status”. ps command is used to list the currently running processes and their PIDs along with some other information depends on different options. It reads the process information from the virtual files in /proc file-system. /proc contains virtual files, this is the reason it’s referred as a virtual file system.

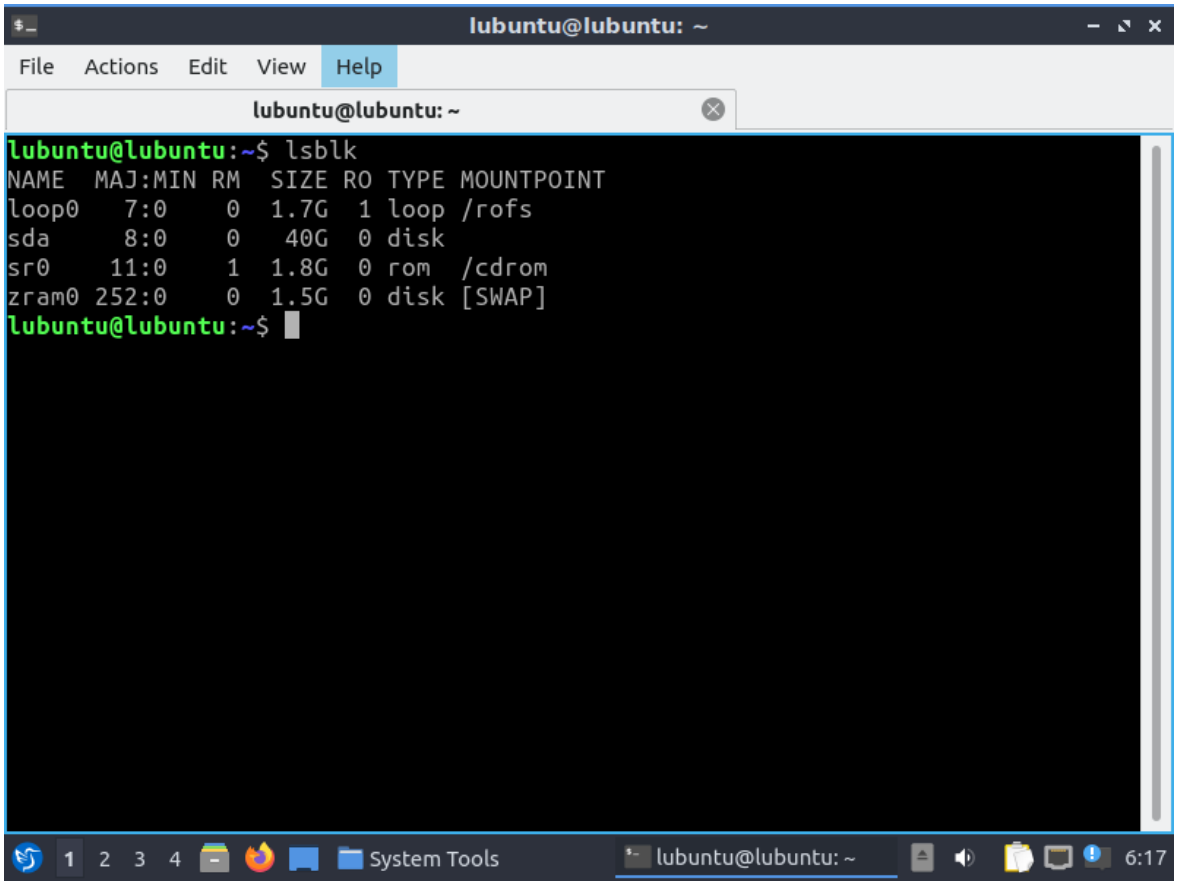
ps provides numerous options for manipulating the output according to our need.

Syntax: ps [options]



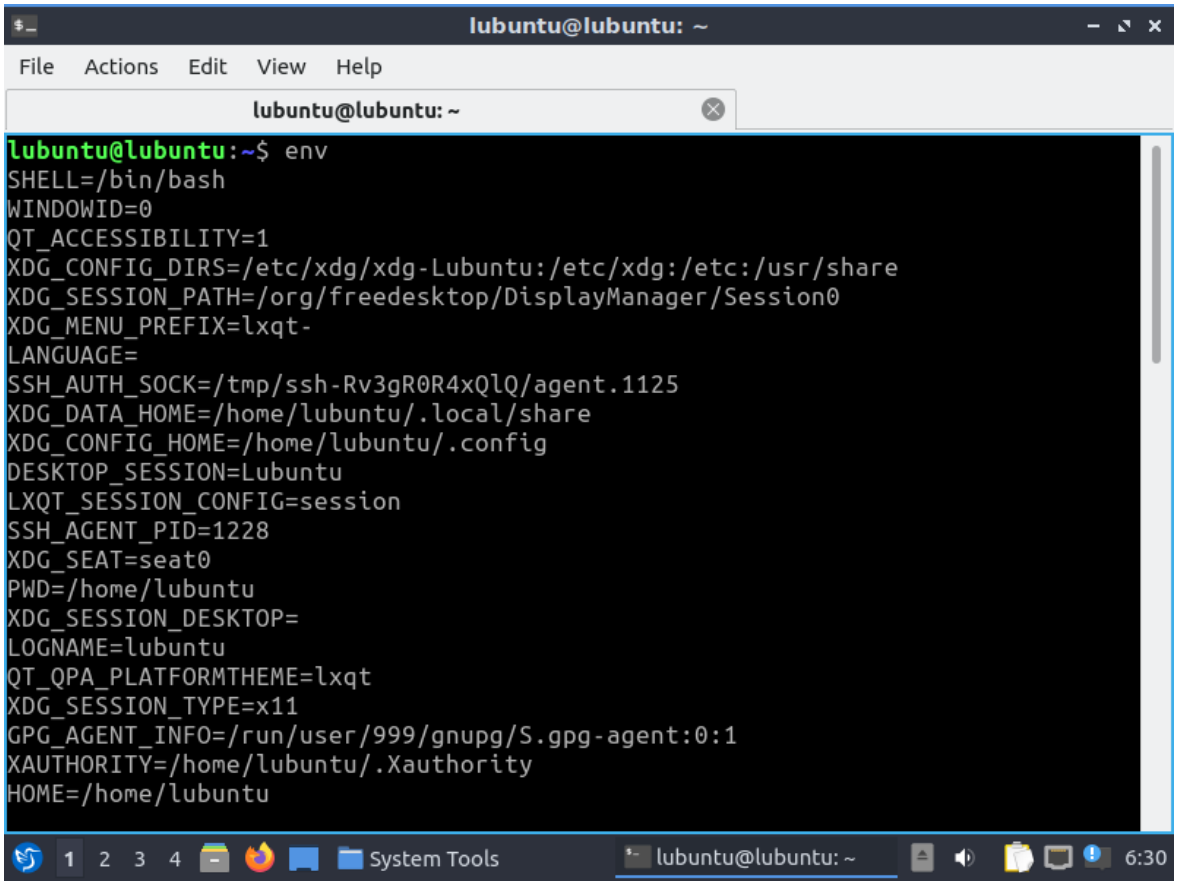
LSBLK COMMAND

Lsblk is used to display details about block devices and these block devices (Except ram disk) are basically those files that represent devices connected to the pc. It queries /sys virtual file system and udev db to obtain information that it displays. And it basically displays output in a tree-like structure. This command comes pre-installed with the util-Linux package.



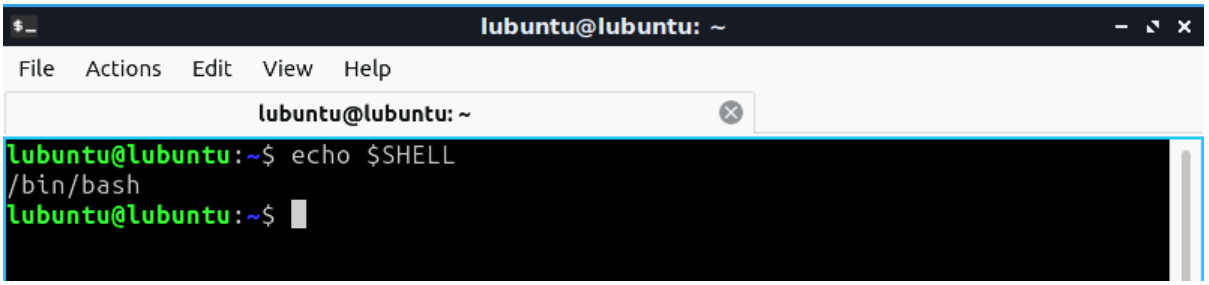
ENV COMMAND

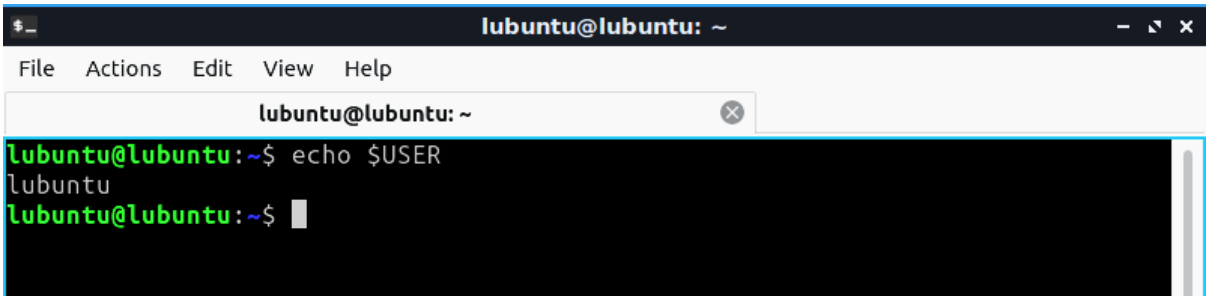
env-env is a shell command for Unix and Unix-like operating systems. It is used to either print a list of environment variables or run another utility in an altered environment without having to modify the currently existing environment. Using env, variables may be added or removed, and existing variables may be changed by assigning new values to them. In practice, env has another common use. It is often used by shell scripts to launch the correct interpreter. In this usage, the environment is typically not changed. If env is run without any options, it prints the variables of the current environment. Otherwise, env sets each NAME to VALUE and executes COMMAND.

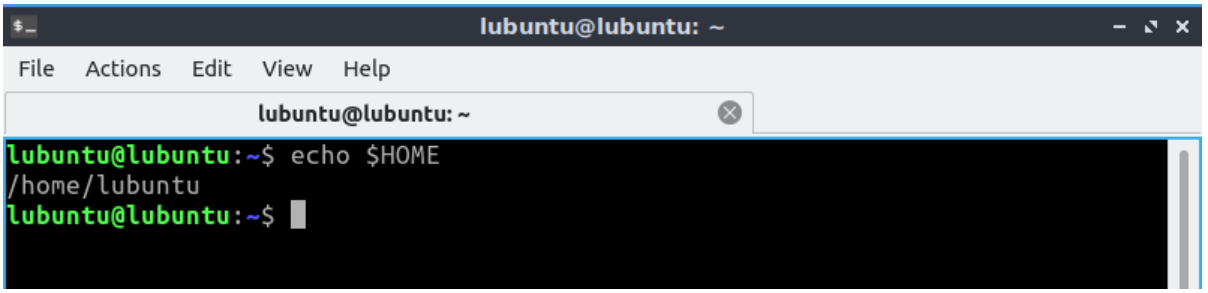


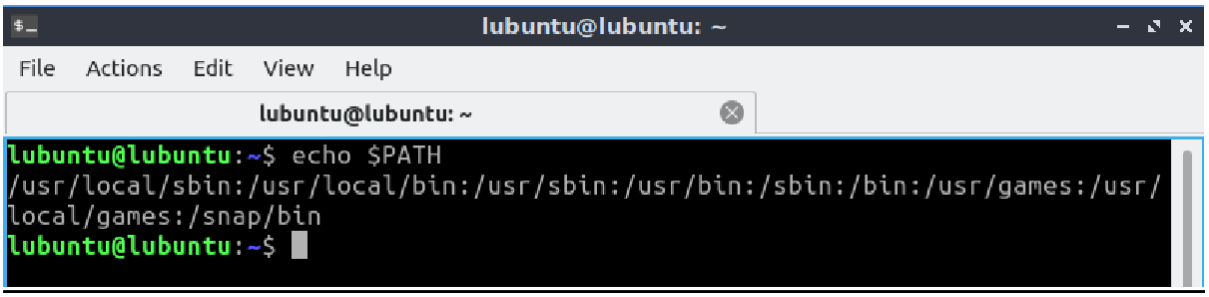
ECHO COMMAND

To get the current user name, shell, path, home.









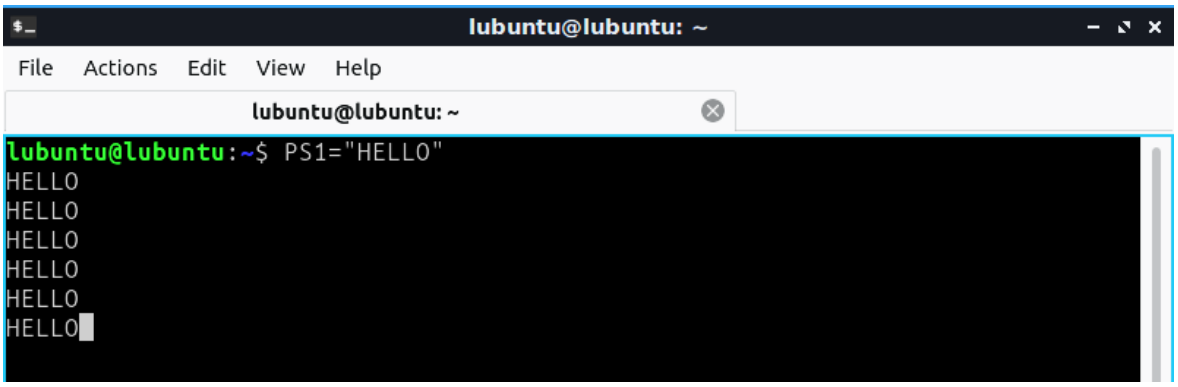
PS1 AND PS2 COMMANDS

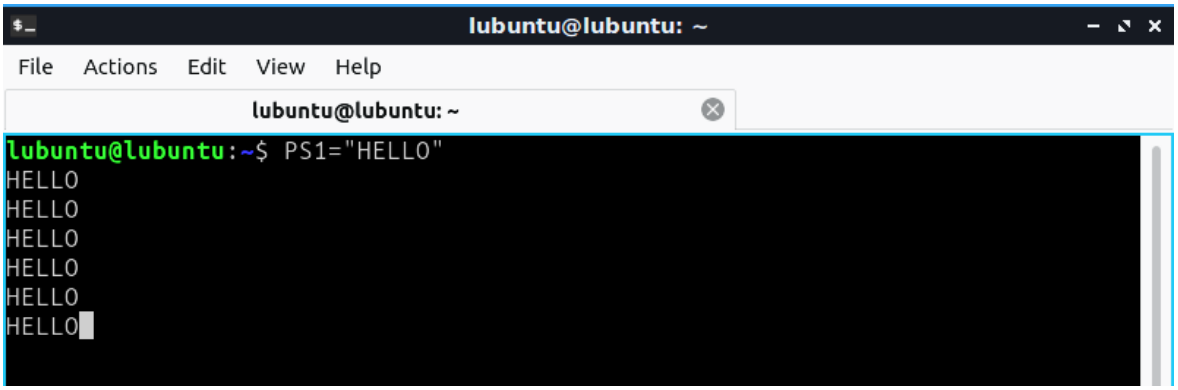
The BASH prompt contains four different values: PS1, PS2, PS3, and PS4. The PS stands for Prompt Statement.

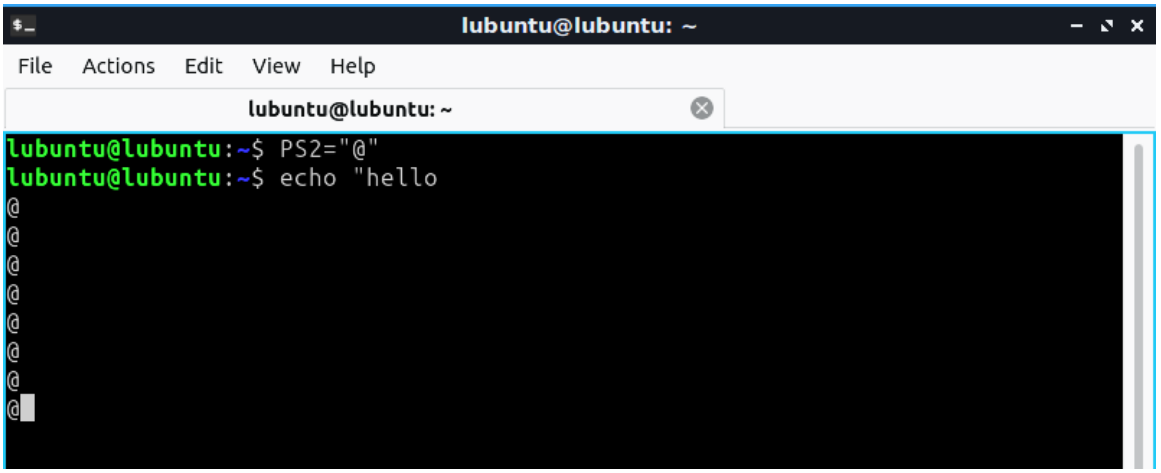
PS1 – This is the primary prompt display. This is where you set special characters or important information.

PS2 – This is the secondary prompt string. This is usually set as a divider between the prompt display and the text entry. It is also used to display when a long command is broken into sections with the \ sign.

You can change the BASH prompt temporarily by using the ps1 command.

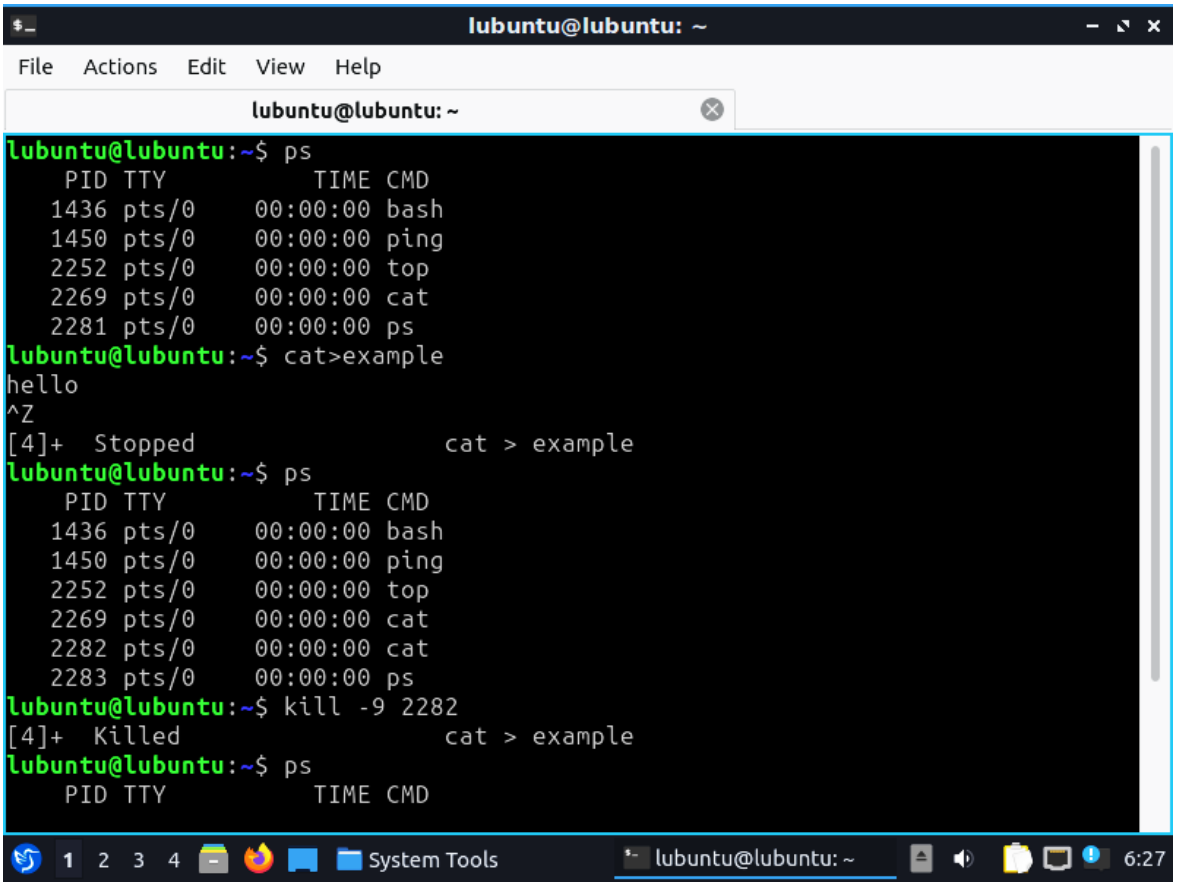


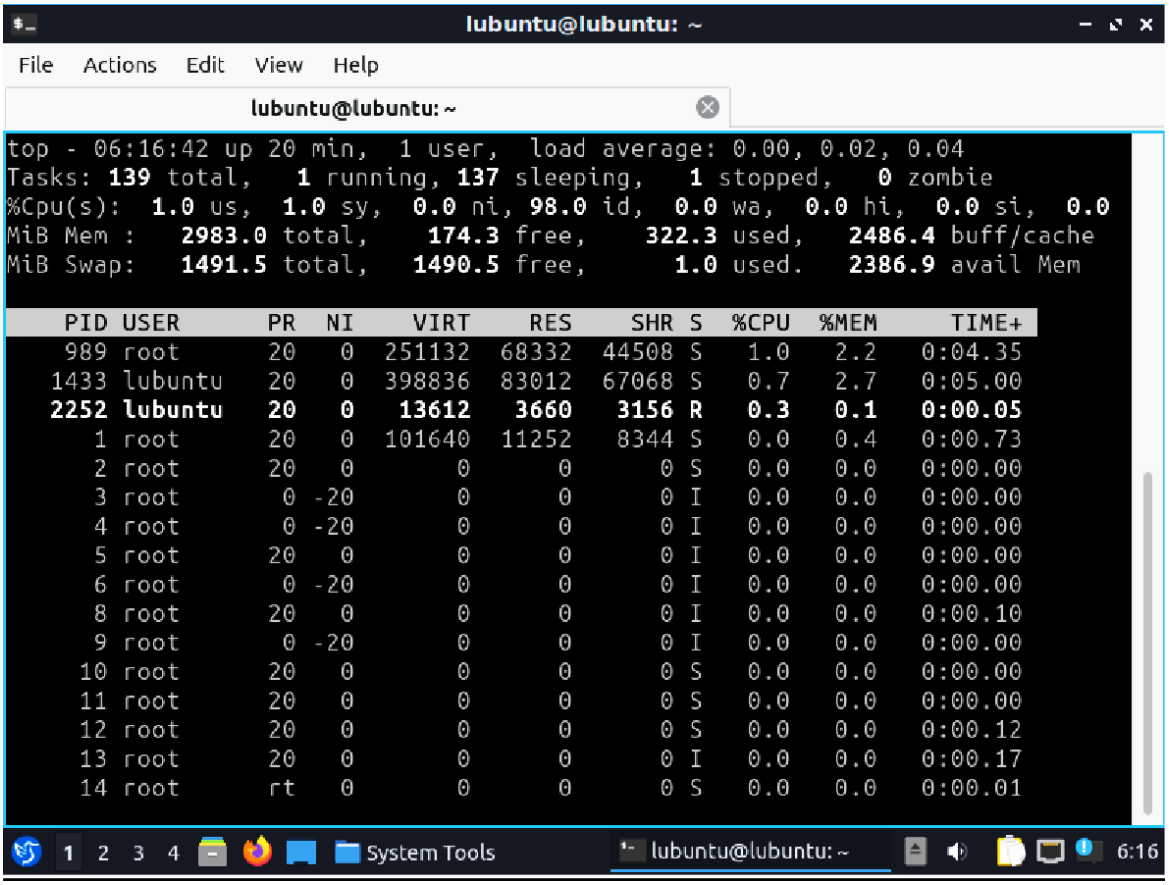




KILL COMMAND

The kill command is used to terminate (kill) a process in Linux. If a program becomes totally unresponsive, you might be forced to terminate it using this command. The kill command sends signals to running processes (identified by their PIDs) in order to request the termination of the process. Besides telling a process to end, signals can tell a process to reread configuration files, pause, continue if stopped, etc. Signals are represented using different numbers. If you don't specify the signal, the default value is 15, which causes the process to exit but allow it to close open files.




lubuntu@lubuntu: ~
File Actions Edit View Help
lubuntu@lubuntu: ~
top - 06:16:42 up 20 min, 1 user, load average: 0.00, 0.02, 0.04
Tasks: 139 total, 1 running, 137 sleeping, 1 stopped, 0 zombie
%Cpu(s): 1.0 us, 1.0 sy, 0.0 ni, 98.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0
MiB Mem : 2983.0 total, 174.3 free, 322.3 used, 2486.4 buff/cache
MiB Swap: 1491.5 total, 1490.5 free, 1.0 used. 2386.9 avail Mem
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+
989 root 20 0 251132 68332 44508 S 1.0 2.2 0:04.35
1433 lubuntu 20 0 398836 83012 67068 S 0.7 2.7 0:05.00
2252 lubuntu 20 0 13612 3660 3156 R 0.3 0.1 0:00.05
1 root 20 0 101640 11252 8344 S 0.0 0.4 0:00.73
2 root 20 0 0 0 0 S 0.0 0.0 0:00.00
3 root 0 -20 0 0 0 I 0.0 0.0 0:00.00
4 root 0 -20 0 0 0 I 0.0 0.0 0:00.00
5 root 20 0 0 0 0 I 0.0 0.0 0:00.00
6 root 0 -20 0 0 0 I 0.0 0.0 0:00.00
8 root 20 0 0 0 0 I 0.0 0.0 0:00.10
9 root 0 -20 0 0 0 I 0.0 0.0 0:00.00
10 root 20 0 0 0 0 S 0.0 0.0 0:00.00
11 root 20 0 0 0 0 S 0.0 0.0 0:00.00
12 root 20 0 0 0 0 S 0.0 0.0 0:00.12
13 root 20 0 0 0 0 I 0.0 0.0 0:00.17
14 root rt 0 0 0 0 S 0.0 0.0 0:00.01
1 2 3 4 System Tools lubuntu@lubuntu: ~ 6:16

EXPERIMENT – 7

INDRODUCTION TO VIRTUAL MACHINES

Creating a Virtual Machine

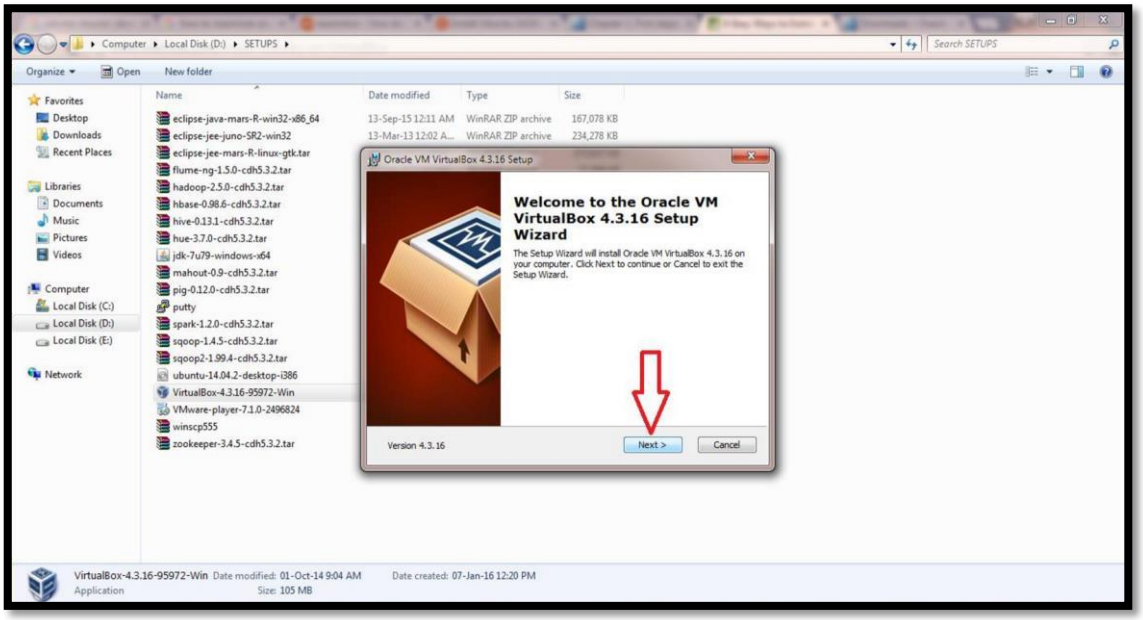
1. Install VirtualBox .

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

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

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

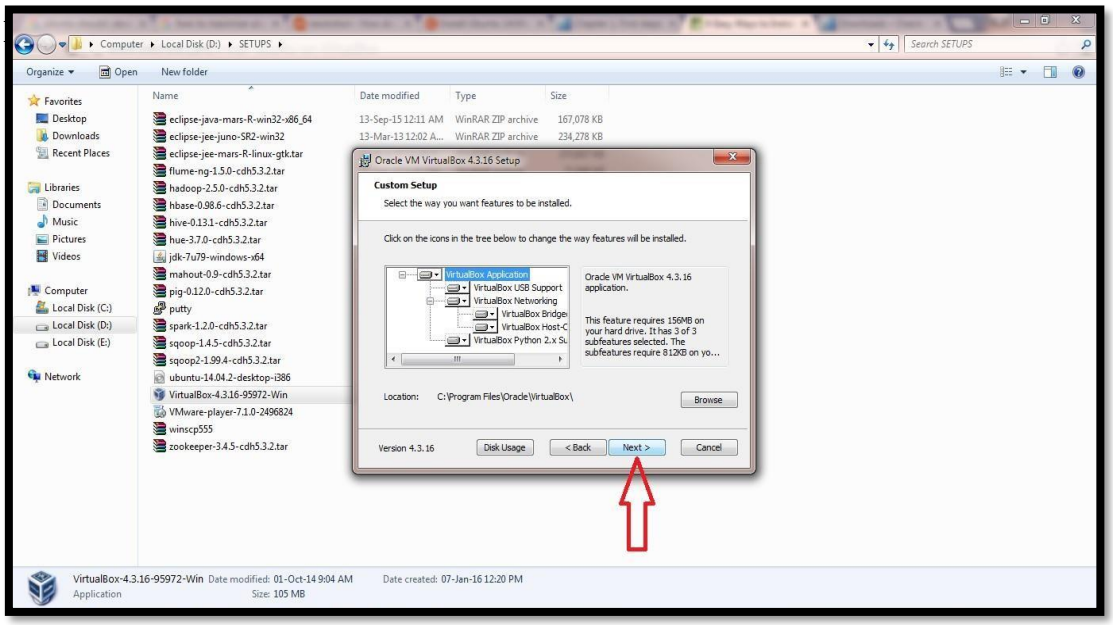
1.1. Click Next



To Install VirtualBox – Setup Wizard

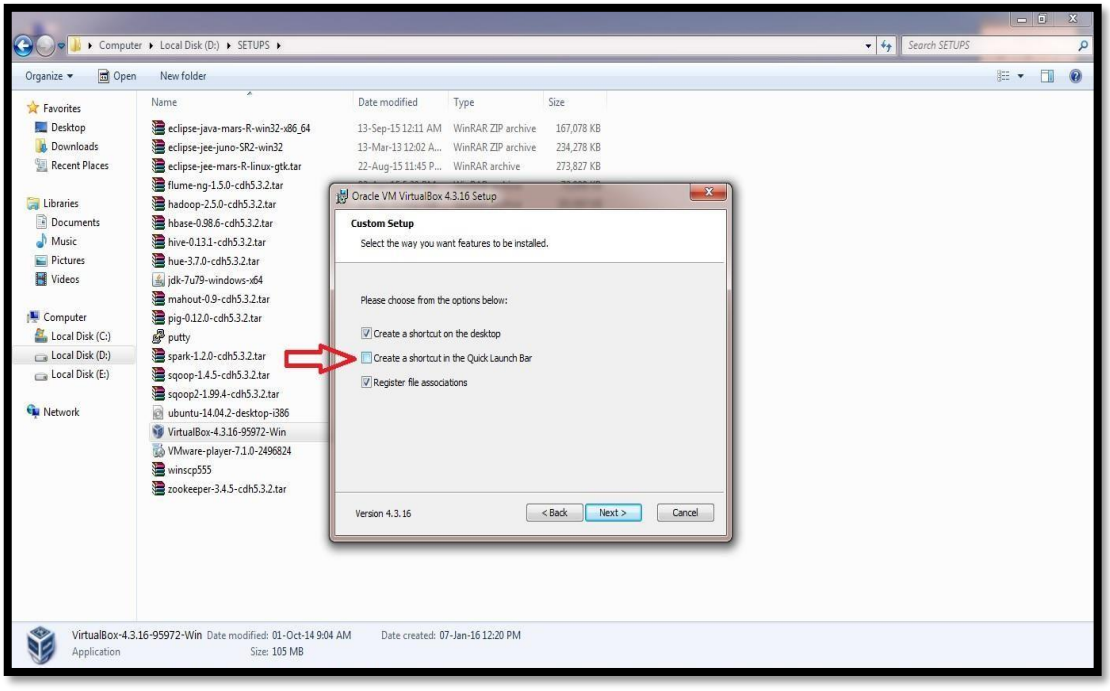
1.2. Click Next

To



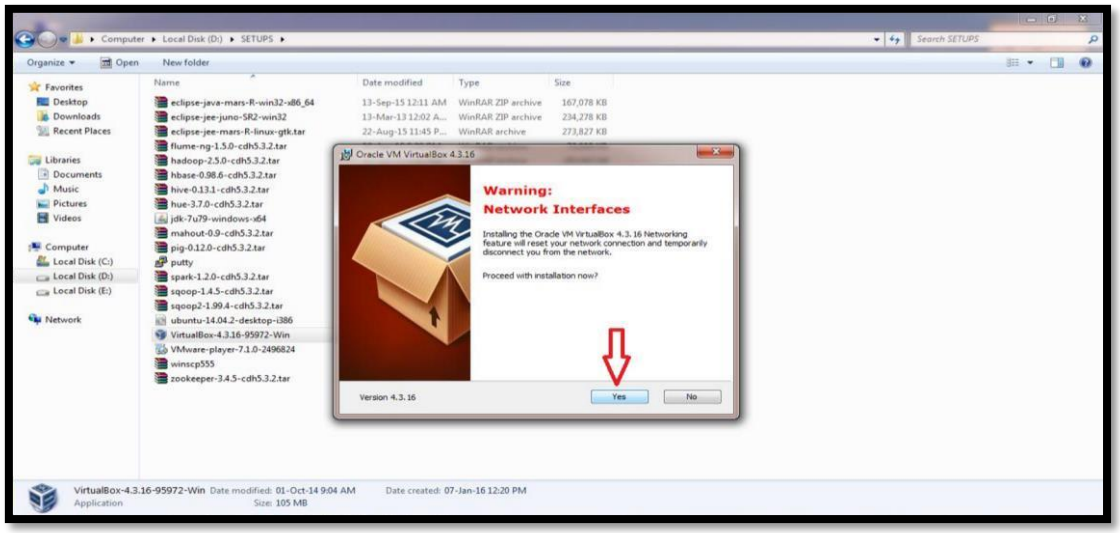
To Install VirtualBox – Custom Setup

1.3. Uncheck "Create a shortcut in the Quick Launch Bar" and click "Next"



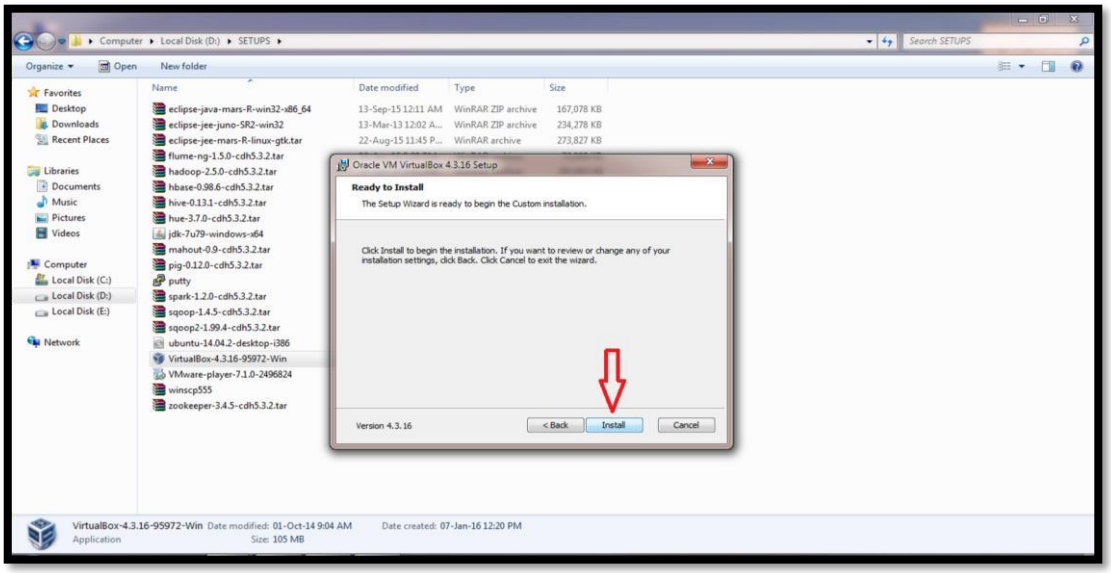
To Install VirtualBox – Features Selection

1.4. Click “Yes”



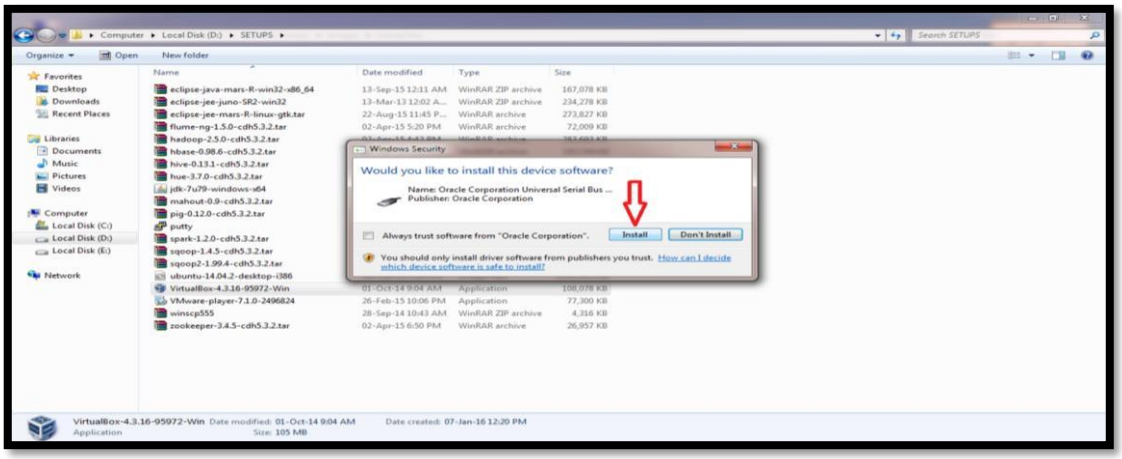
To Install VirtualBox – Network Interfaces Warning

1.5. Click “Install”



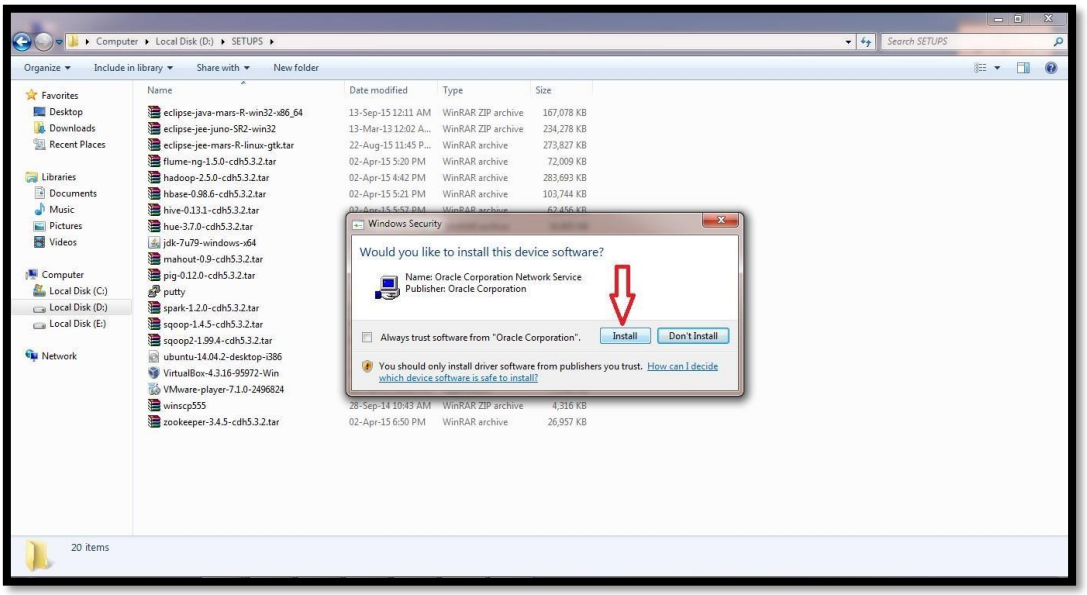
Installation of Oracle VM VirtualBox – Ready to Install

1.6. Click “Install”



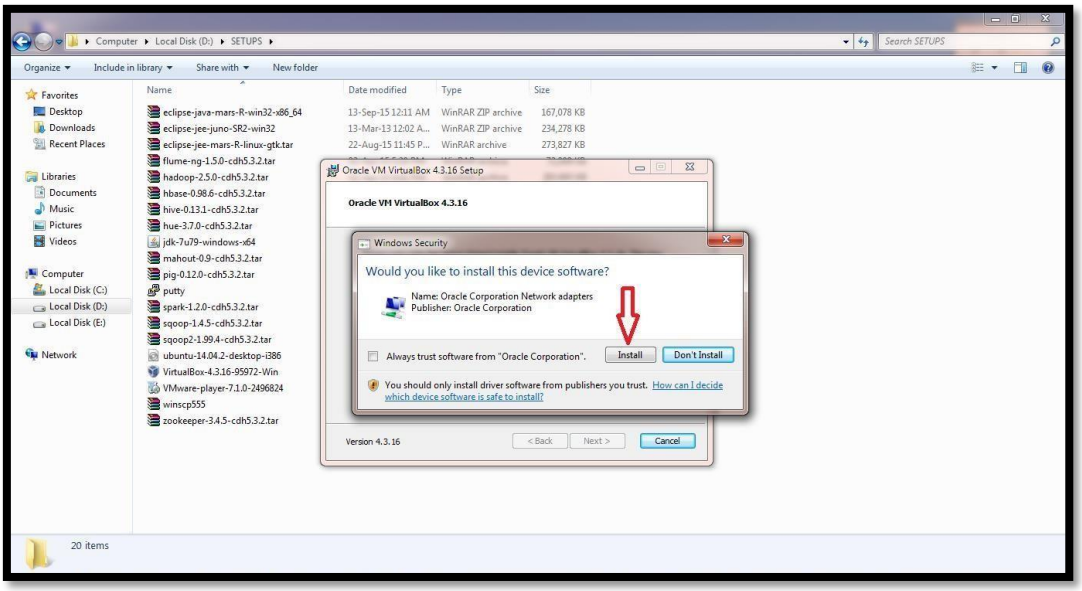
Installation of Oracle VM VirtualBox- Serial Bus Software Installation

1.7. Click “Install”



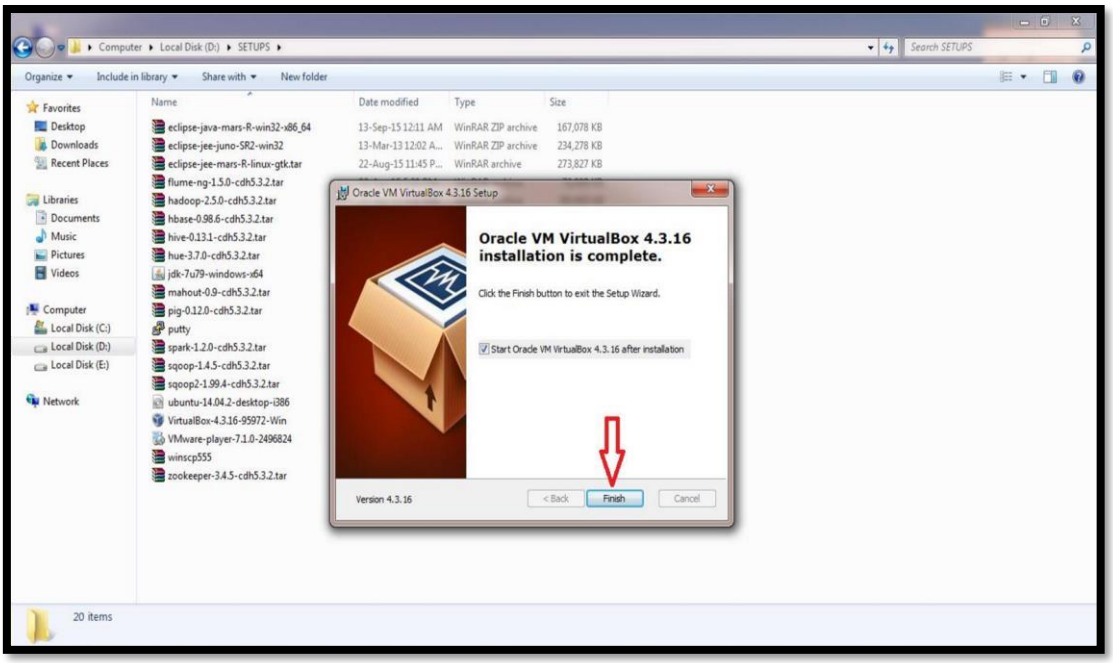
Installation of Oracle VM VirtualBox – Network Service Installation

1.8. Click “Install”



Installation of Oracle VM VirtualBox – Network Adapters Installation

1.9. Click “Finish”

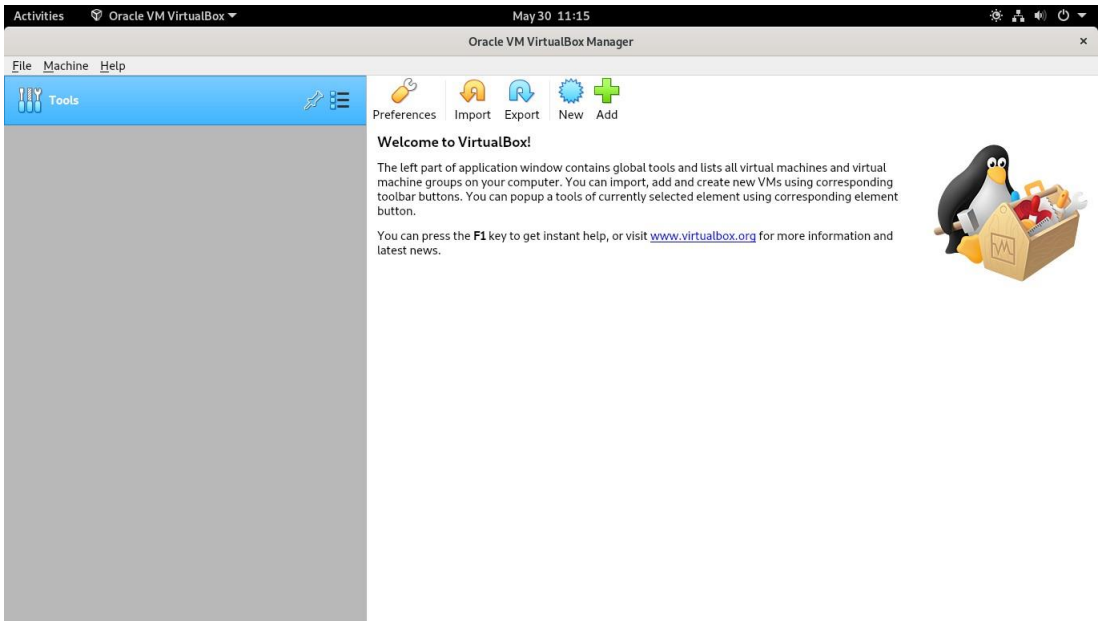


2. Open VirtualBox.

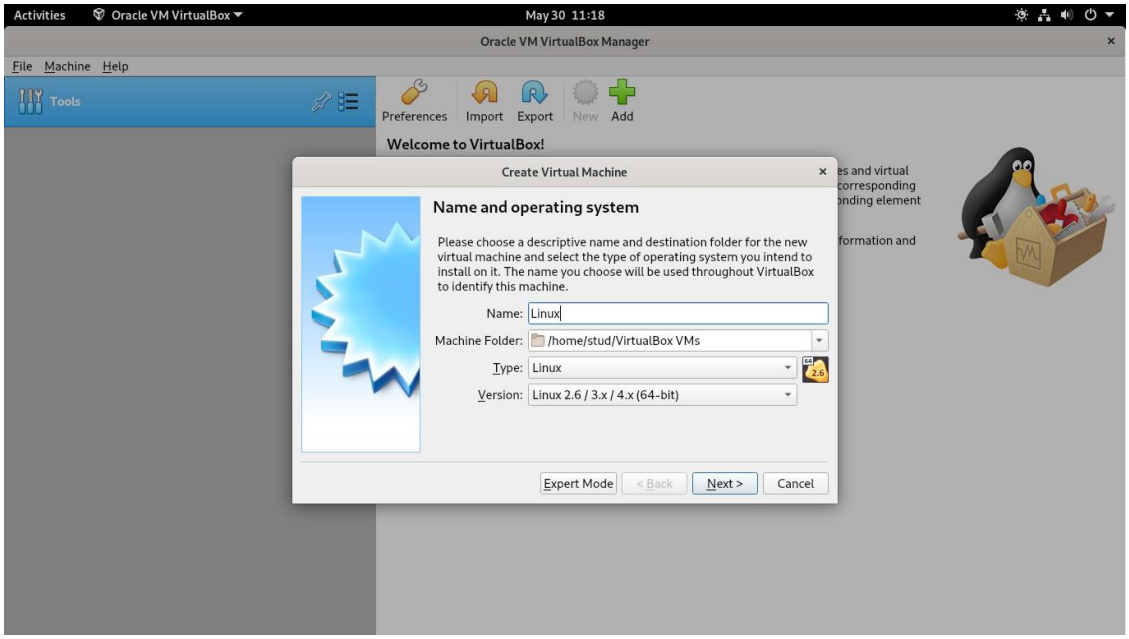
Double-click (or click once on a Mac) the VirtualBox app icon.

3. Click New.

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



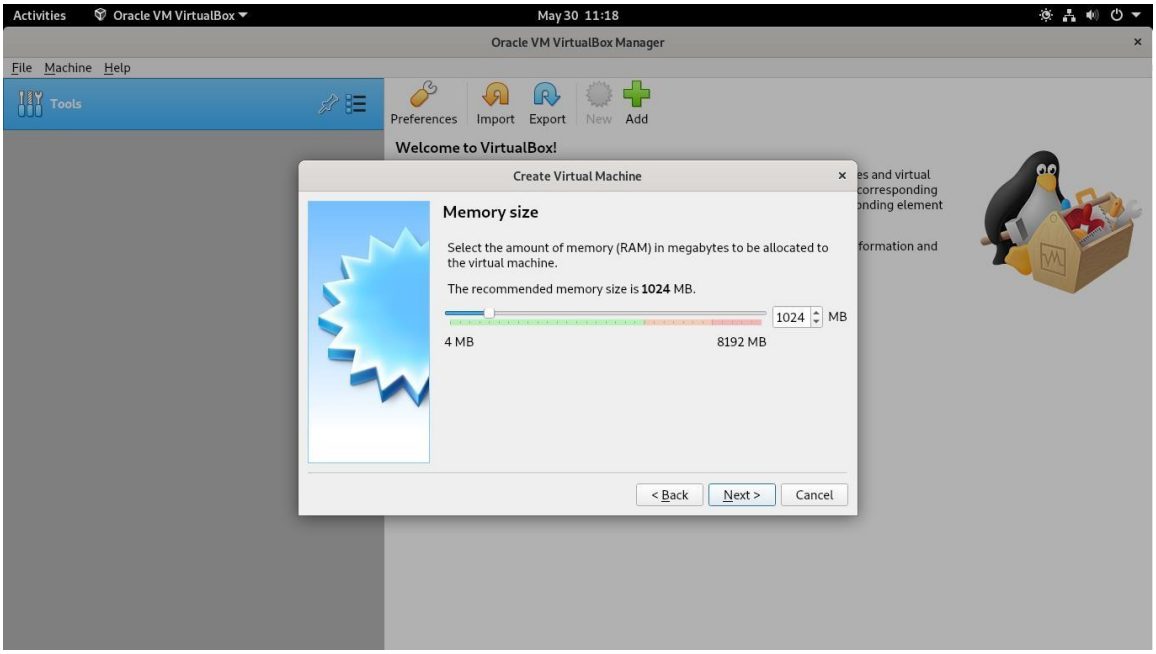
4. Enter a name for your virtual machine. Type whatever you want to name your virtual machine (e.g., Ubuntu) into the "Name" text field that's near the top of the pop-up menu.



For Operating System Type, select the OS that you want to install. Select the version of the operating system. **Click Next.** It's at the bottom of the menu.

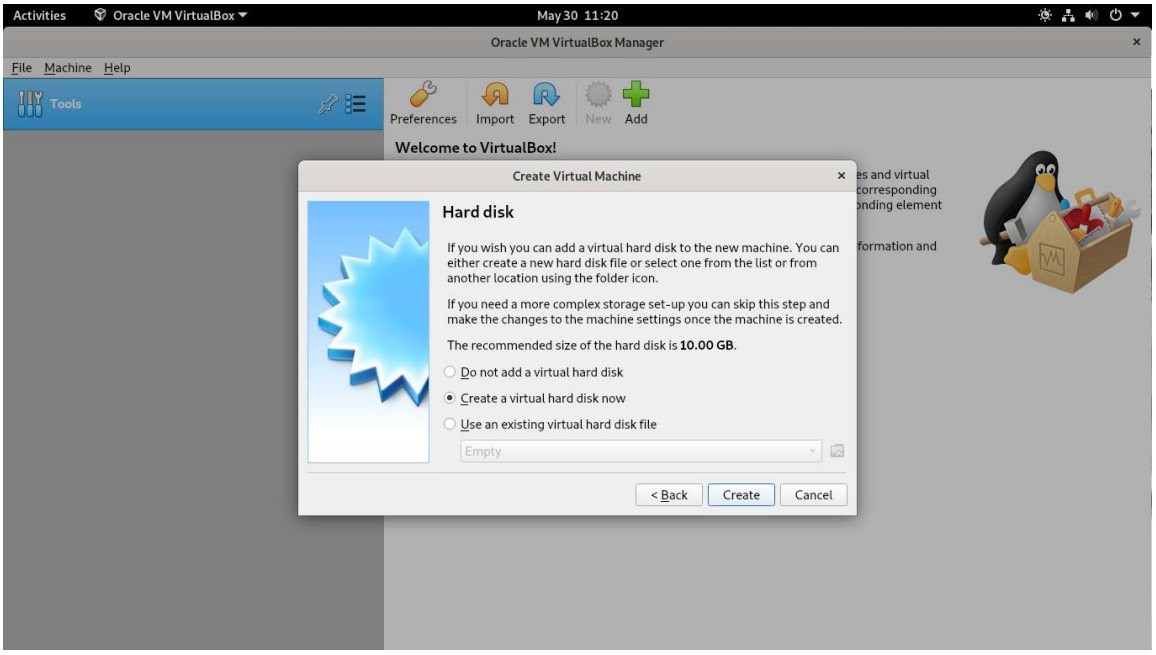
5. Select an amount of RAM to use. Click and drag the slider left or right to decrease or increase the amount of RAM that VirtualBox will have available for your virtual machine.

The ideal amount of RAM will automatically be selected when you get to this page. Make sure not to increase the RAM into the red section of the slider; try to keep the slider in the green.

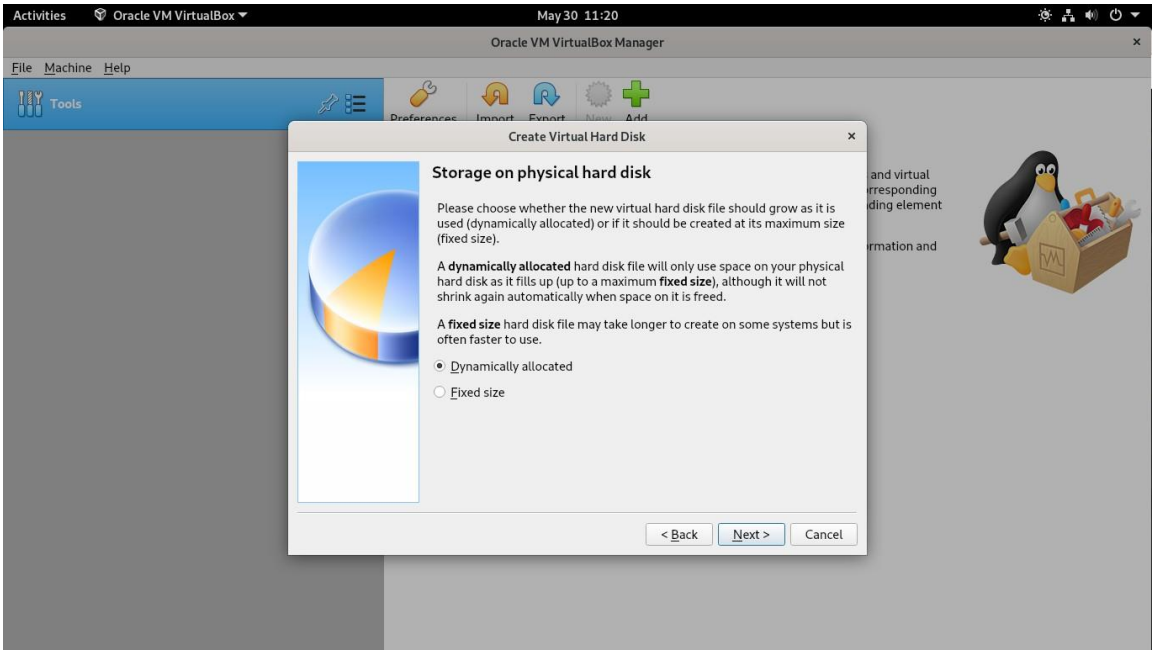


Click Next. It's at the bottom of the menu.

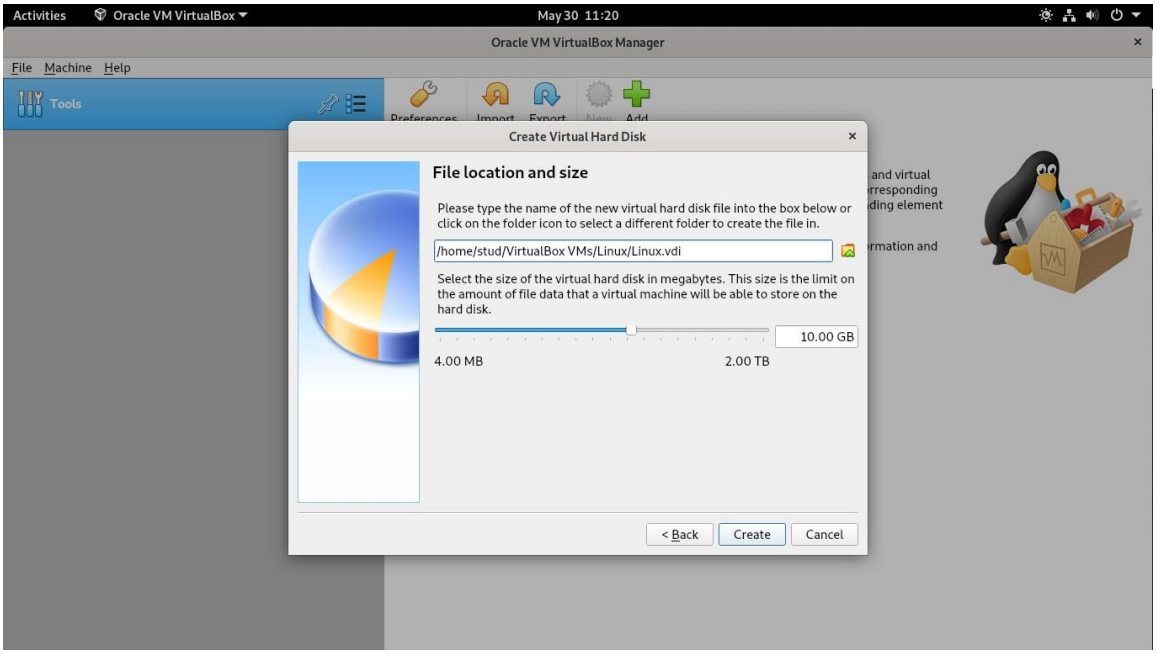
6. Create your virtual machine's virtual hard drive. The virtual hard drive is a section of your computer's hard drive space which will be used to store your virtual machine's files and programs:



7. Use "VDI" to create a virtual hard disk. Choose **"Dynamically allocated."**

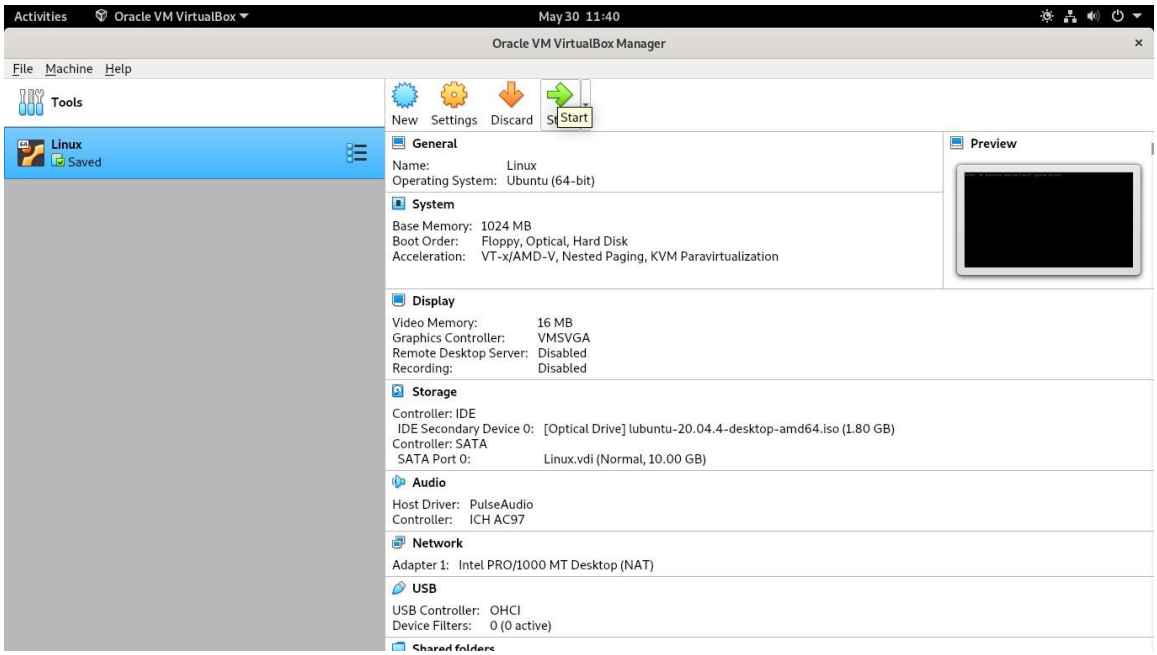


8. Allocate at Minimum 8 GB (recommended 10 or more).



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

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


Activities
Oracle VM VirtualBox
May 30 11:40
Oracle VM VirtualBox Manager
File Machine Help
Tools
Linux
Saved
New
Settings
Discard
Start
General
Name: Linux
Operating System: Ubuntu (64-bit)
System
Base Memory: 1024 MB
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, KVM Paravirtualization
Preview
Display
Video Memory: 16 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled
Storage
Controller: IDE
IDE Secondary Device 0: [Optical Drive] lubuntu-20.04.4-desktop-amd64.iso (1.80 GB)
Controller: SATA
SATA Port 0: Linux.vdi (Normal, 10.00 GB)
Audio
Host Driver: PulseAudio
Controller: ICH AC97
Network
Adapter 1: Intel PRO/1000 MT Desktop (NAT)
USB
USB Controller: OHCI
Device Filters: 0 (0 active)
Shared folders

EXPERIMENT No – 8

WIRESHARK

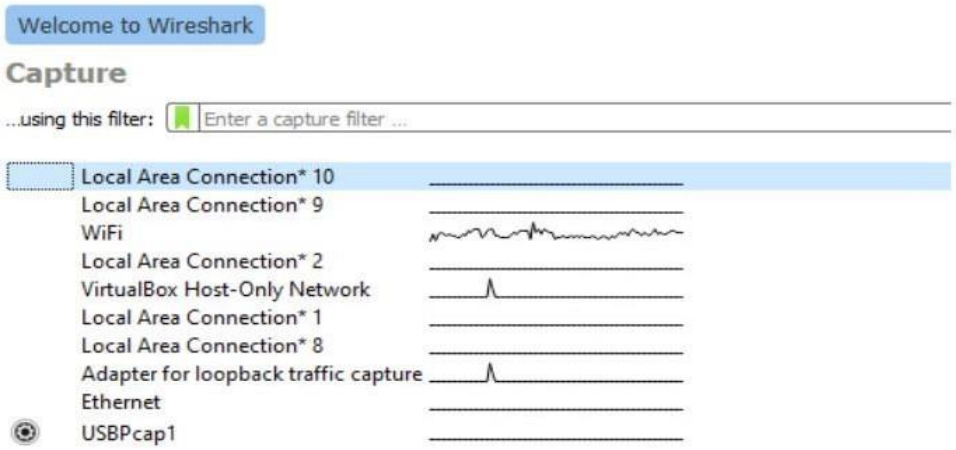
Wireshark is an open-source packet analyzer, which is used for education, analysis, software development, communication protocol development, and network troubleshooting. It is used to track the packets so that each one is filtered to meet our specific needs. It is commonly called as a sniffer, network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

It is also used by network security engineers to examine security problems. Wireshark is a data capturing program that "understands" the structure (encapsulation) of different networking protocols. It can parse and display the fields, along with their meanings as specified by different networking protocols. Wireshark usespcap to capture packets, so it can only capture packets on the types of networks that pcap supports.

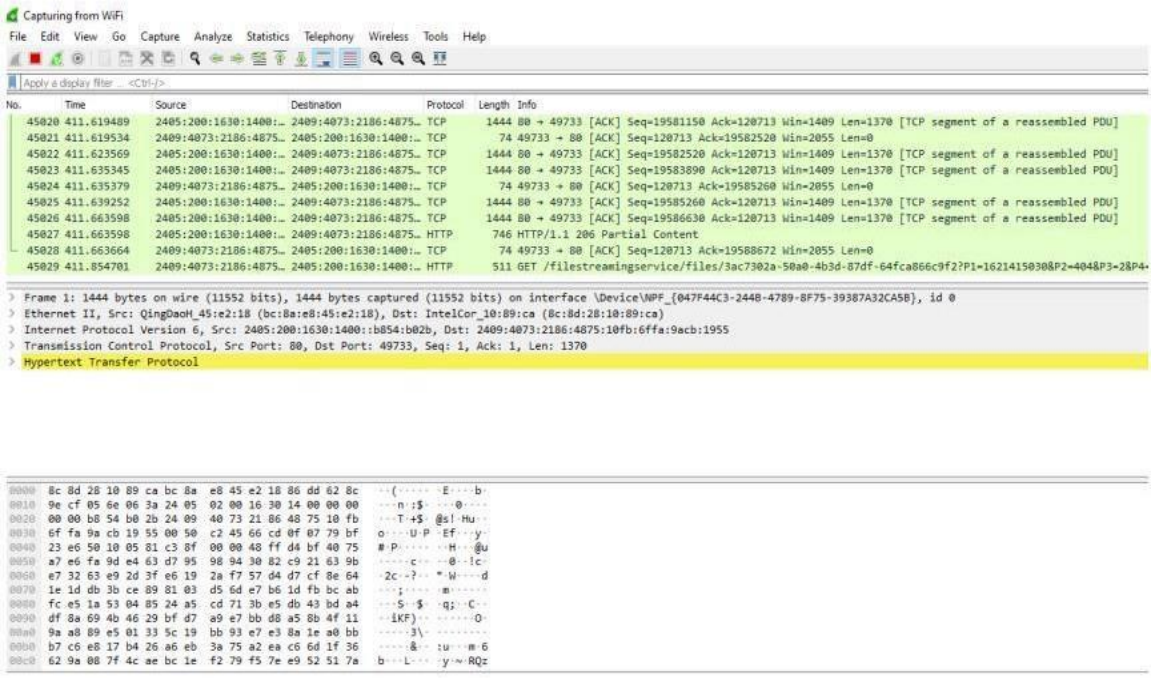
Installation of Wireshark Software

Below are the steps to install the Wireshark software on the computer:

- Open the web browser.
- Search for 'Download Wireshark.
- Select the Windows installer according to your system configuration, either 32-bt or 64-bit. Save the program and close the browser.
- Now, open the software, and follow the install instruction by accepting the license.The Wireshark is ready for use.
- On the network and Internet settings option, we can check the interface connected to our computer.
- By selecting the current interface, we can get the traffic traversing through that interface. The version used here is 3.0.3. This version will open as:

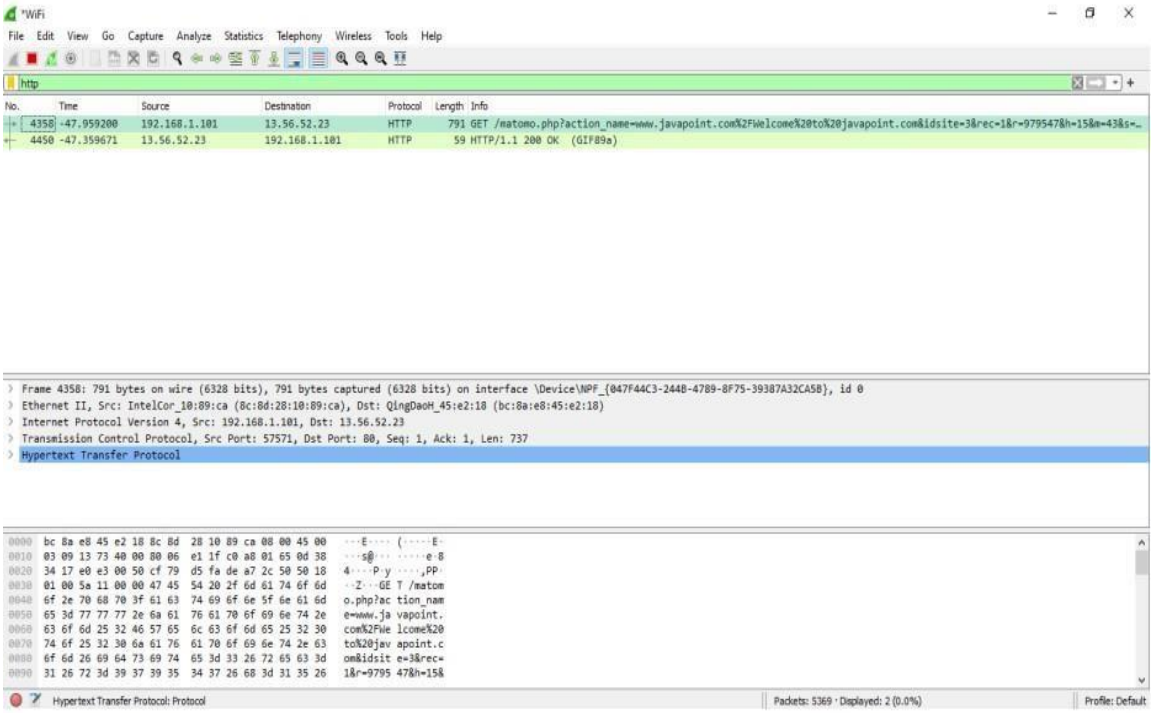


- The Wireshark software window is shown above, and all the processes on the network are carried within this screen only.
- The options given on the list are the Interface list options. The number of interface options will be present. Selection of any option will determine all the traffic.
- For example, from the above fig. select the Wi-Fi option. After this, a new window opens up, which will show all the current traffic on the network. Below is the image which tells us about the live capture of packets and our Wireshark will look like:



How to retrieve web pages using http and wireshark to capture the packets?

- Start up your web browser.
- Start up the Wireshark packet sniffer. Enter "http" in the display-filterspecification window, so that only captured HTTP messages will be displayed later in the packet-listing window. (We're onlyinterested in the HTTP protocol here, and don't want to see the clutter of all captured packets).
- Wait a bit morethan one minute (we'll see why shortly), and then begin wireshark packet capture.
- Enter the following URL in your web browser: http://www.javapoint.com
- Now stop the Wireshark packet capture.
- Find the lines in Wireshark that correspond to your browser retrieving the above URL. That is, the "GET message" (from your browser to the web server) and the response message from the server to your browser. The packet-contents window shows details of the selected message (in this case the HTTP OK message, which is highlighted in the packet-listing window).Recall that since the HTTP message was carried inside a TCP segment, which was carried inside an IP datagram, which was carried within an Ethernet frame, Wireshark displays the Frame, Ethernet, IP, and TCP packet information as well.
- Make sure the boxes at the far left of the Frame, Ethernet, IP and TCP information have a plus sign or a right-pointing triangle (which means there is hidden, undisplayed information), and the HTTP line has a minus sign or a down-pointing triangle (which means that all information about the HTTP message is displayed).



Wireshark display after http://javapoint.com has been retrieved by your browser.

EXPERIMENT No – 9

Laravel installation On Ubuntu with Apache

Step 1 – Install Apache Web Server

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

$ sudo apt update

After updating your package list install apache webserver

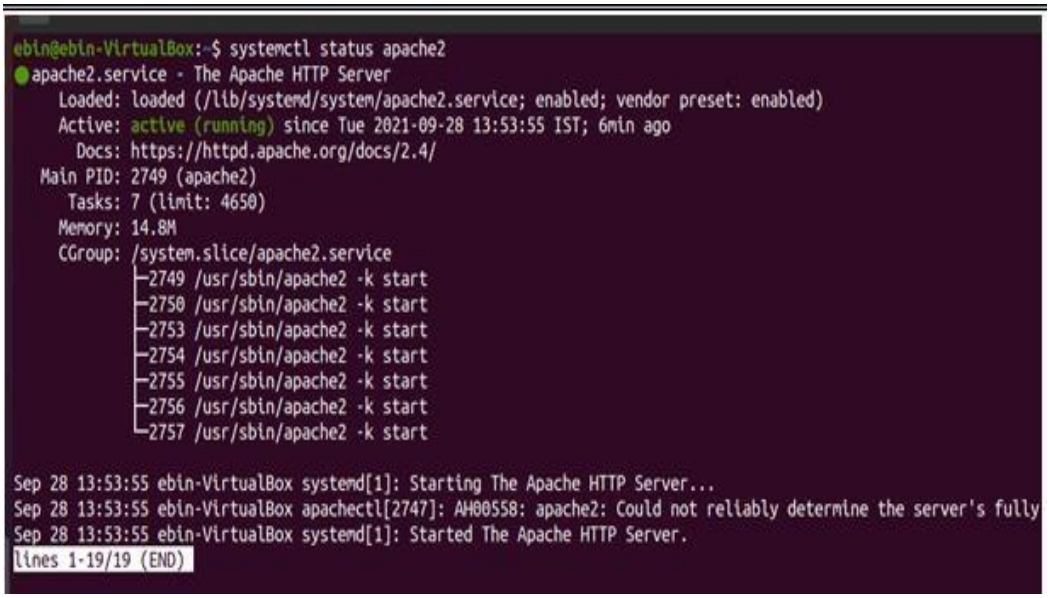
$ sudo apt install apache2
$ systemctl status apache2

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

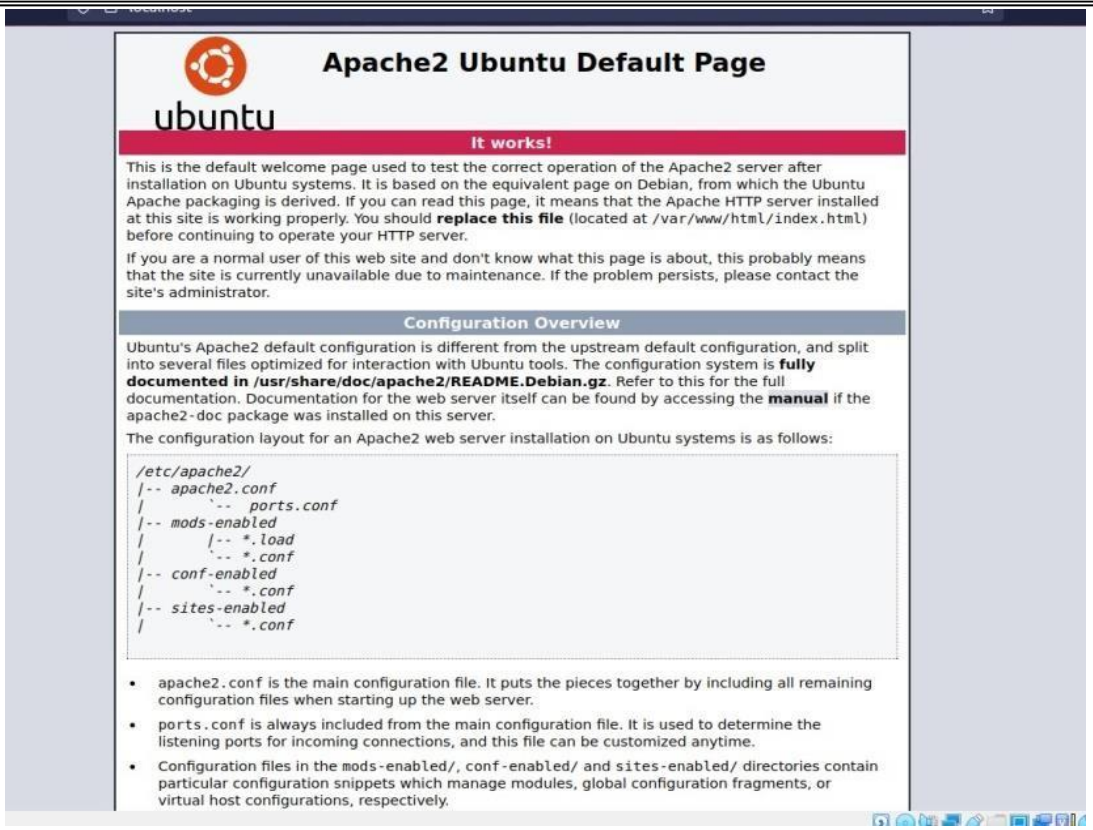
Now, check the status of apache server whether it is running or not.

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

$ systemctl start apache2
$ systemctl enable apache2



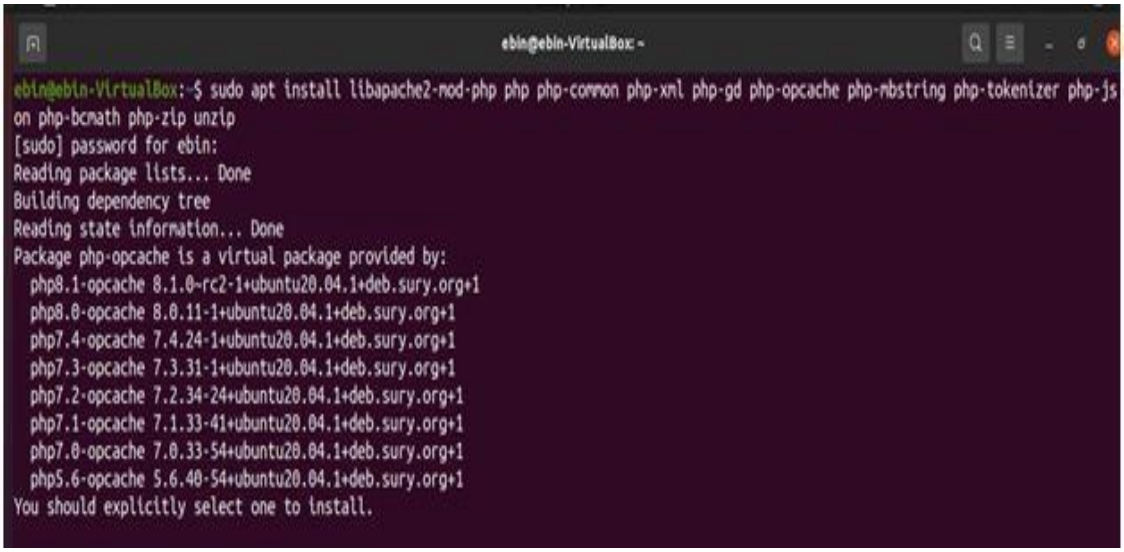
Open browser, goto localhost and check if default apache server page is available or not

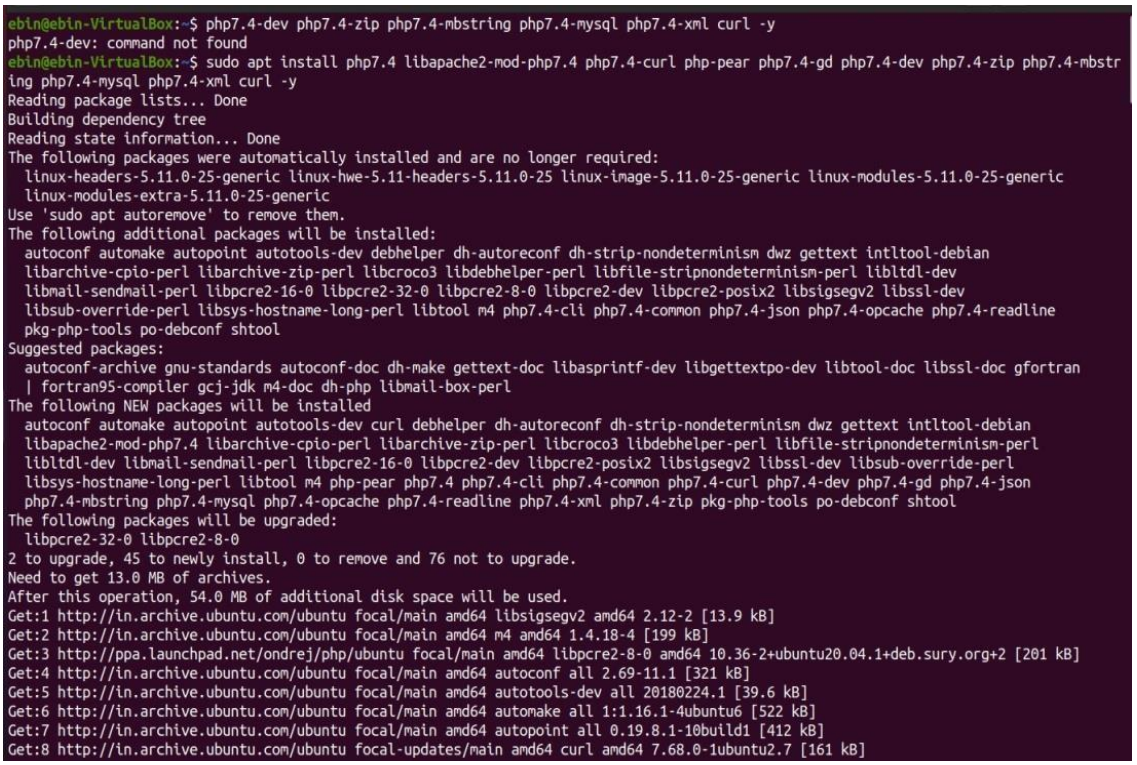


Step 2 – Install and Configure PHP 7.4

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

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





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

$ cd /etc/php/7.4/apache2
$ sudo nano php.ini

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

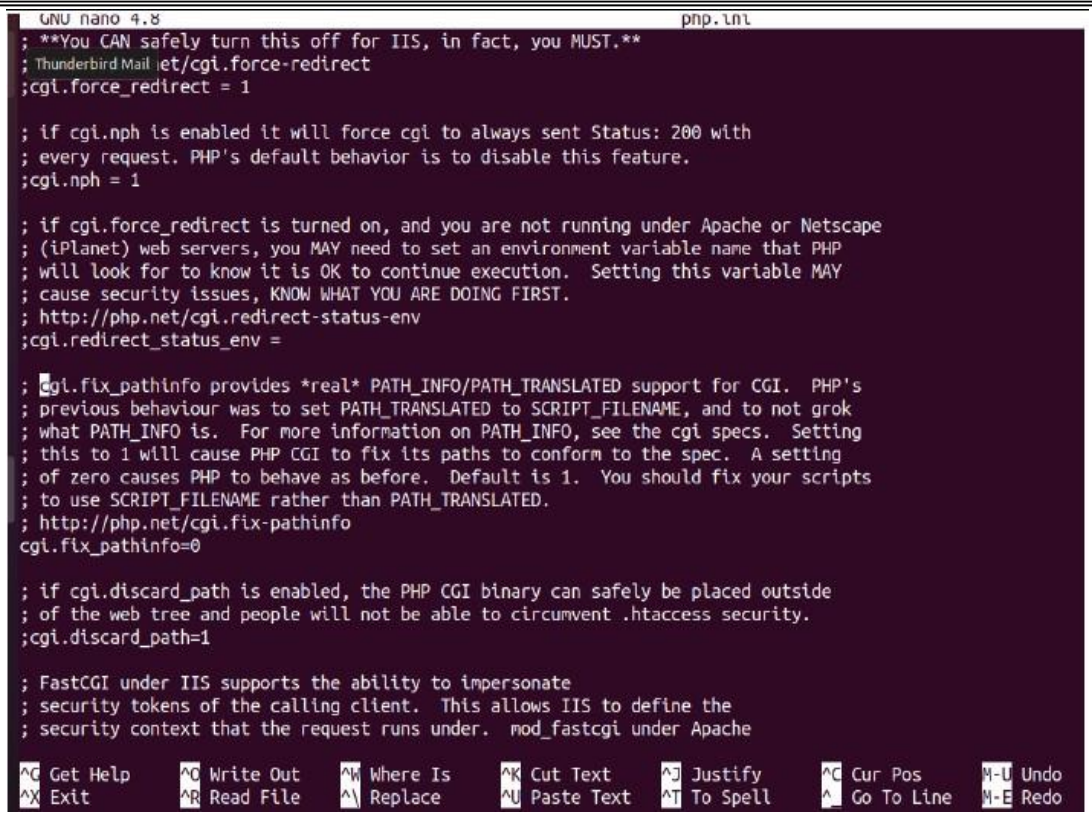
...

cgi.fix_pathinfo=0

...

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

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

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

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

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

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

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

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



Press Ctrl + x then y to Save and Exit.

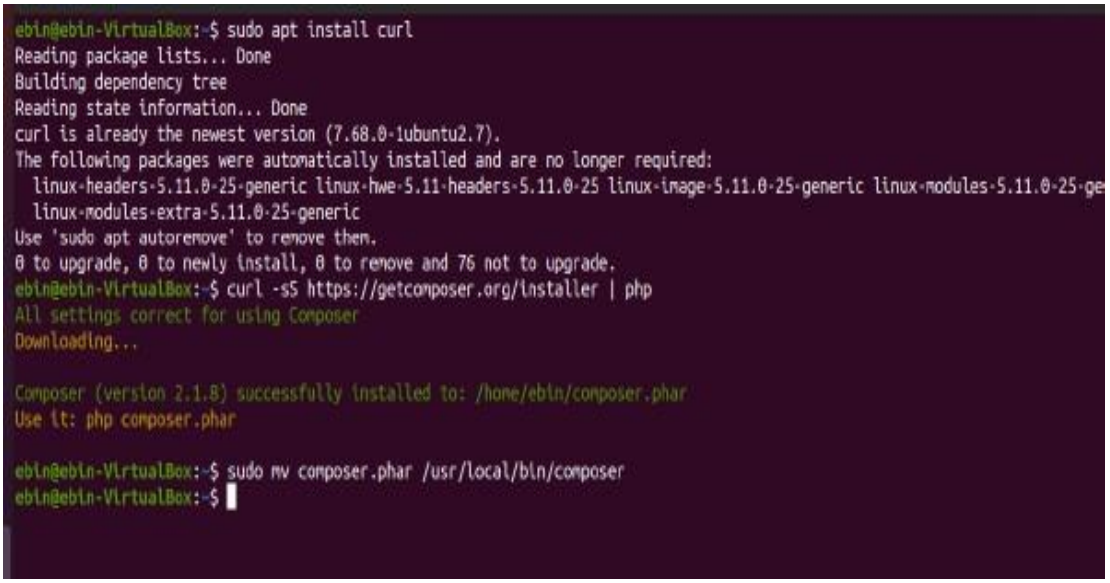
Now Restart The apache service.

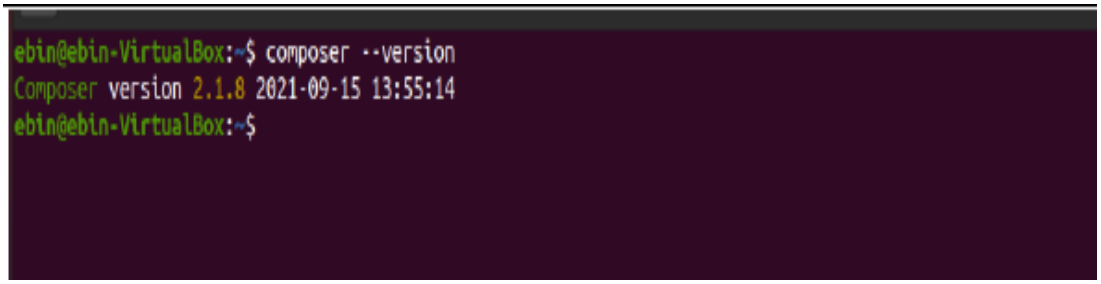
$ systemctl restart apache2

Step 3 – Install Composer PHP Packages Management

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

$ sudo apt install curl
$ curl -sS https://getcomposer.org/installer | php
$ sudo mv composer.phar /usr/local/bin/composer





Step 4 – Install Laravel 8.x on Ubuntu 20.04

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

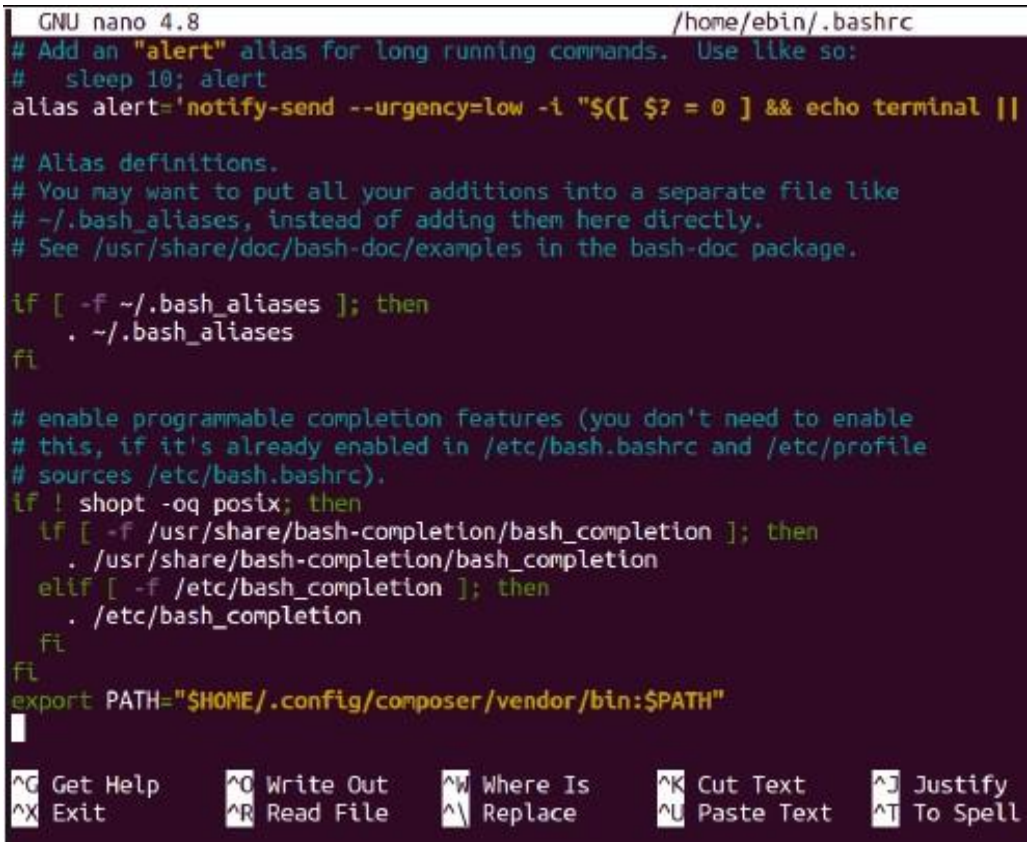
```
ebin@ebin-VirtualBox:~$ composer global require laravel/installer
Changed current directory to /home/ebin/.config/composer
Using version ^4.2 for laravel/installer
./composer.json has been created
Running composer update laravel/installer
Loading composer repositories with package information
Updating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

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

$ nano ~/.bashrc

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

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



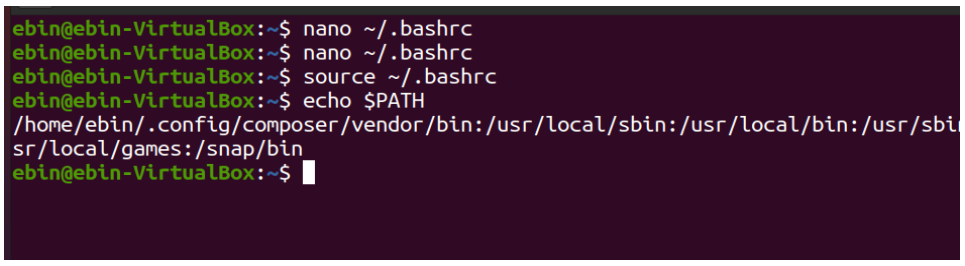
Press Ctrl + x then y to Save and Exit.

Now reload your bashrc configuration using the source command.

$ source ~/.bashrc

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

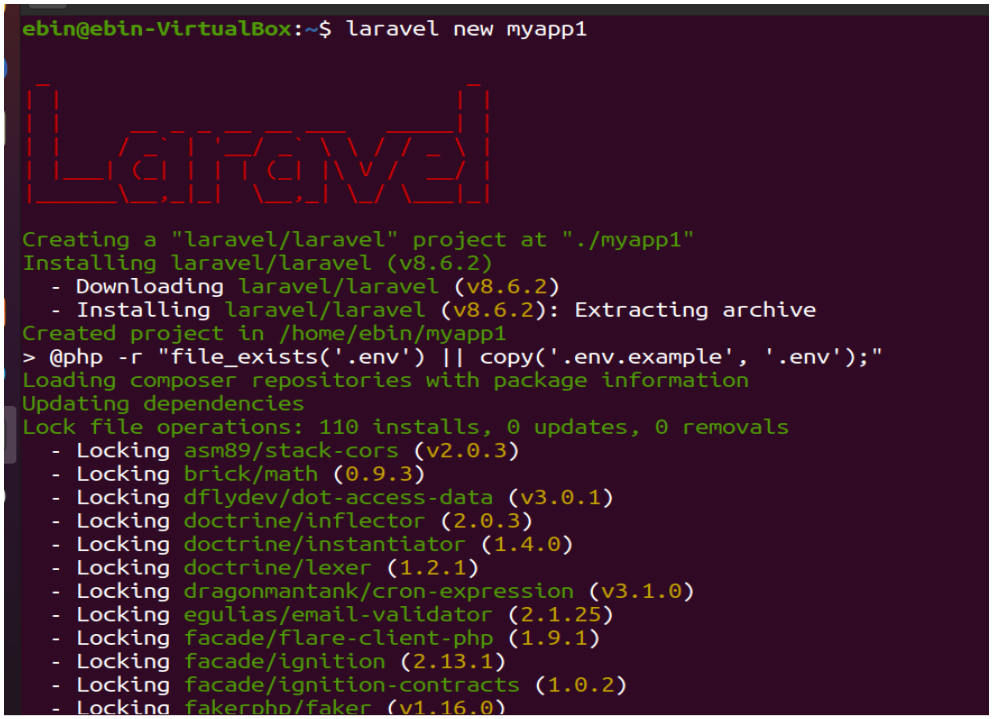
$ echo $PATH



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

$ laravel new myapp1

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



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

Step 5 – Finally Configure Apache for Laravel and test it

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

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

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

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

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

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

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

$ cd /etc/apache2/sites-available/
$ sudo nano myapp1.com.conf

And paste the following line inside the file.

```
<VirtualHost *:80>
ServerName myapp1.com

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

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

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

//All Red colored text must be changed as per your//

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

Now enable mod rewrite for apache2 just type

$ sudo a2enmod rewrite

Now enable your site, just type

$ sudo a2ensite myapp1.com.conf

Finally, Restart the apache service, type

$ systemctl restart apache2

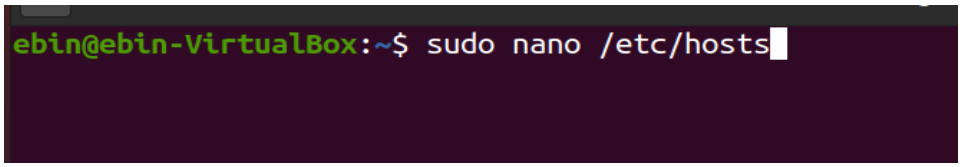
```
ebin@ebin-VirtualBox:~$ cd /etc/apache2/sites-available/
ebin@ebin-VirtualBox:/etc/apache2/sites-available$ sudo nano myapp1.com.conf
ebin@ebin-VirtualBox:/etc/apache2/sites-available$ sudo nano myapp1.com.conf
ebin@ebin-VirtualBox:/etc/apache2/sites-available$ sudo a2enmod rewrite
Enabling module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

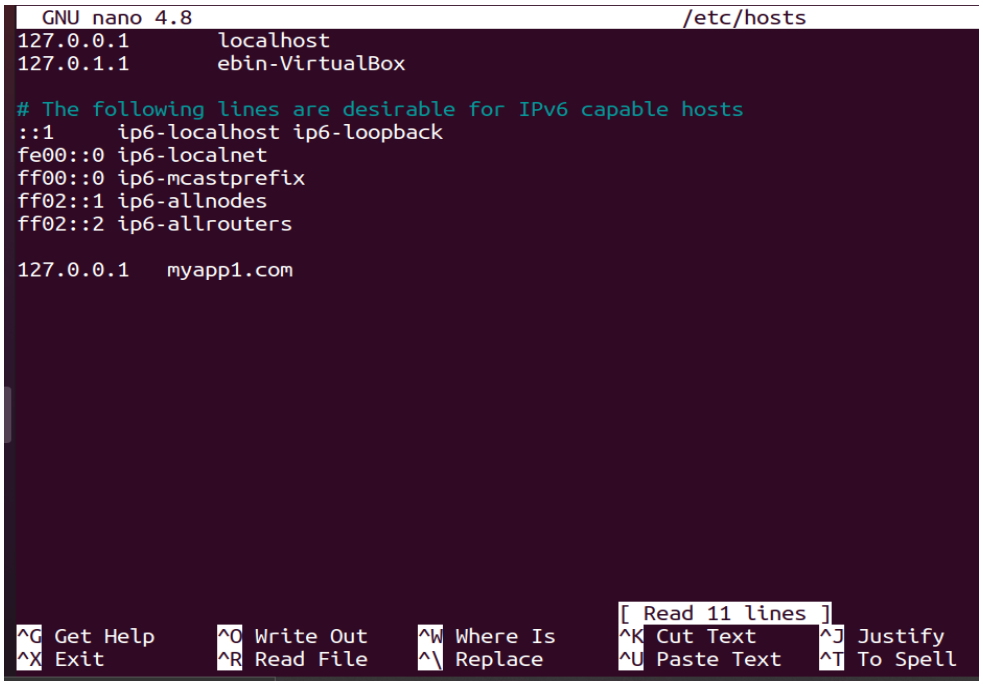
As you are in a local environment you need a local dns resolver for your site. Go ahead and edit /etc/hosts file, add a dns record for your site then save the file.

$ sudo nano /etc/hosts

...

127.0.0.1 myapp1.com





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

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