**1.Introduction to Computer Hardware**

Identify major components of a computer such as Mother board, Daughter cards, Bus slots, SMPS, Internal storage devices, Interfacing Ports.

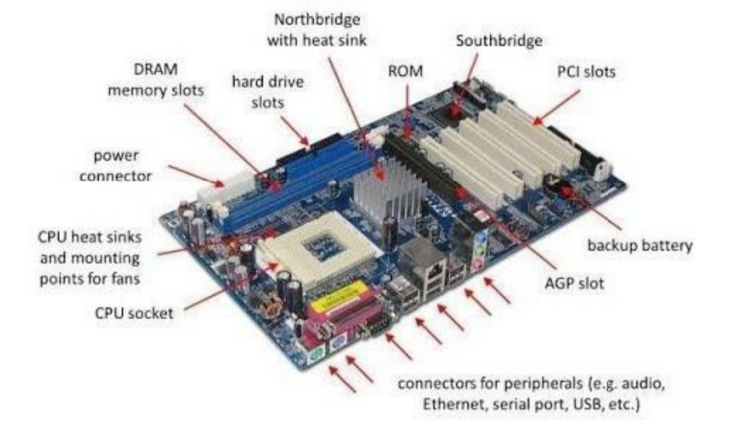
Physical identification of major components of a computer system such as motherboard, RAM modules, daughter cards, bus slots, SMPS, internal storage devices, interfacing ports. Specifications of desktop and server class computers. Installation of common operating systems for desktop and server use. (Students may be asked to formulate specifications for a computer to be used as a Desktop, Web server.)

**MOTHERBOARD**

A motherboard (also called mainboard, main circuit board, or mobo) is the main printed circuit board (PCB) in general-purpose computers and other expandable systems. It holds and allows communication between many of the crucial electronic components of a system, such as the central processing unit (CPU) and memory, and provides connectors for other peripherals.

Unlike a backplane, a motherboard usually contains significant sub-systems, such as the central processor, the chipset’s input/output and memory controllers, interface connectors, and other components integrated for general use.

Motherboard means specifically a PCB with expansion capabilities. As the name suggests, this board is often referred to as the "mother" of all components attached to it, which often include peripherals, interface cards, and daughterboards: sound cards, video cards, network cards, host bus adapters, TV tuner cards, IEEE 1394 cards, and a variety of other custom components.



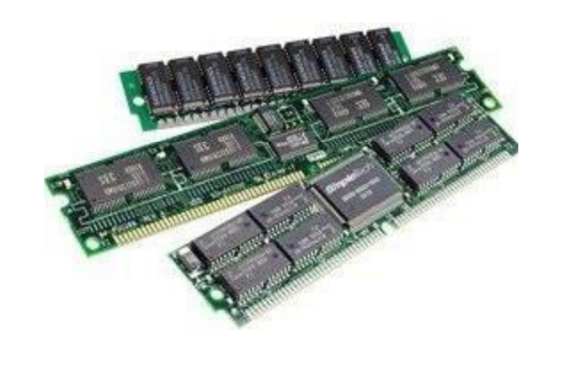
**RAM MODULES**

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. Later, memory modules were standardized by organizations such as JEDEC and could be used in any system designed to use them.

Types of memory modules include:

* TransFlash Memory Module
* SIMM, a single in-line memory module
* DIMM, dual in-line memory module
* Rambus memory modules (subset of DIMMs, normally referred to as RIMMs)
* SO-DIMM, small outline DIMM, a smaller version of the DIMM, used in laptops

Distinguishing characteristics of computer memory modules include voltage, capacity, speed (bit rate), and form factor. For economic reasons, the large (main) memories found in personal computers, workstations, and non-handheld game consoles (such as PlayStation and Xbox) normally consist of dynamic RAM (DRAM). Other parts of the computer, such as cache memories, normally use static RAM (SRAM). Small amounts of SRAM are sometimes used in the same package as DRAM. However, since SRAM has high leakage power and low density, die-stacked DRAM has recently been used for designing multi-megabyte-sized processor caches.

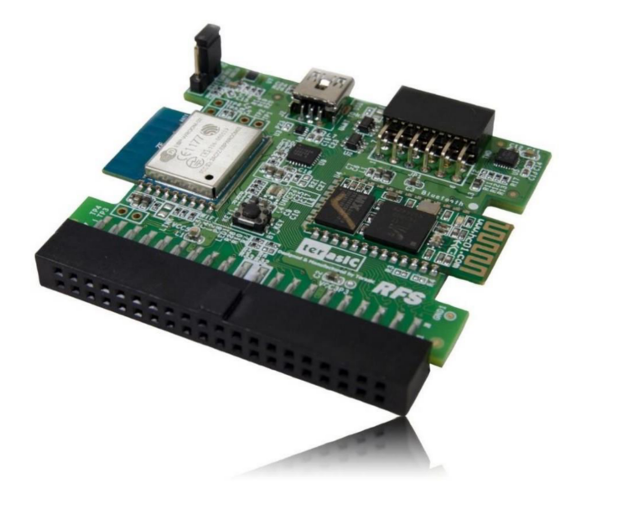


**DAUGHTER BOARD**

The daughter board is a computer hardware component. It is also known as the piggyback board, riser card, daughter board, daughter card, or daughter card. A daughter board is a printed circuit board that 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.

Daughter boards are directly connected to the motherboard. Expansion cards are connected to the motherboard using the bus and other serial interfaces, but daughter boards are directly connected to the board by soldering. As an update of the motherboard or expansion card, daughter boards are released to extend the features and services of the motherboard or expansion cards.



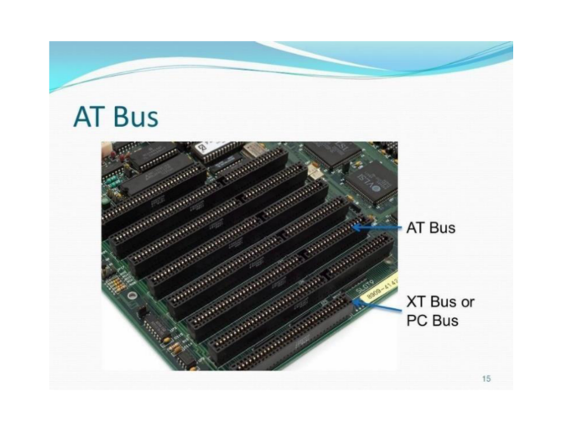
**BUS SLOTS**

Alternatively known as a bus slot or expansion port, an expansion slot is a connection or port inside a computer on the motherboard or riser card. It provides an installation point for a hardware expansion card to be connected, which provides additional features to a computer such as video, sound, advanced graphics, Ethernet, or memory.

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

Expansion cards can provide various functions including:

* Sound
* Modems
* Solid-state drive
* Network
* Power-on self-test
* Interface adapters
* Advanced multirate codec
* TV and radio tuning
* Basic input/output system (BIOS)
* Video processing
* Host adapting such as redundant array of independent disks or small computer system interface
* Expansion read-only memory (ROM)
* Security devices
* RAM memory



**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 a 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

A basic isolated AC to DC switched-mode power supply consists of:

* Input rectifier and filter
* Inverter consisting of switching devices such as MOSFETs
* Transformer
* Output rectifier and filter
* Feedback and control circuit



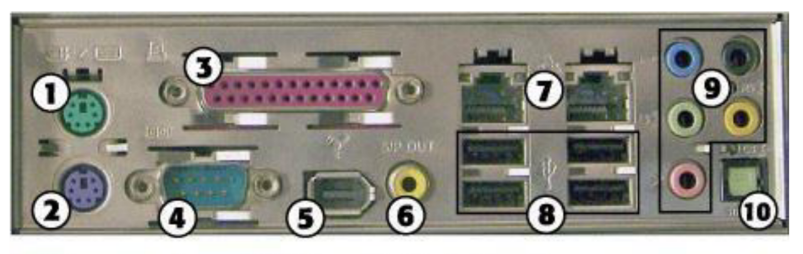
**INTERNAL STORAGE DEVICES**



A storage unit is a part of the computer system that is employed to store the information and instructions to be processed. A storage device is an integral part of the computer hardware that stores information/data to process the result of any computational work.

Without a storage device, a computer would not be able to run or even boot up. Or in other words, we can say that a storage device is hardware that is used for storing, porting, or extracting data files. It can also store information/data both temporarily and permanently. Computer storage is of two types:

* **Primary Storage Devices:** Also known as internal memory and main memory. This is a section of the CPU that holds program instructions, input data, and intermediate results. It is generally smaller in size. RAM (Random Access Memory) and ROM (Read Only Memory) are examples of primary storage.
* **Secondary Storage Devices:** Secondary storage is a memory that is stored externally to the computer. It is mainly used for permanent and long-term storage of programs and data. Examples include Hard Disk, CD, DVD, Pen/Flash drive, SSD, etc.

**INTERFACING PORTS**

1. PS/2 mouse port
2. PS/2 keyboard port
3. Parallel port
4. Serial port
5. IEEE 1394a port
6. SPDIF coaxial digital audio port
7. Ethernet ports
8. USB ports
9. 1/8-inch mini-jack audio ports
10. SPDIF optical digital audio port

A **port** is a physical docking point used to connect external devices to the computer. It acts as an interface between the computer and external devices like hard drives and printers.

**Characteristics of Ports:**

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

**2.Basic Linux Commands**

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

Basic Linux commands, familiarity with following commands/operations expected

1. man

2. ls, echo, read

3. more, less, cat,

4. cd, mkdir, pwd, find

5. mv, cp, rm ,tar

6. wc, cut, paste

7. head, tail, grep, expr

8. chmod, chown

9. Redirections & Piping

10. useradd, usermod, userdel, passwd

11. df,top, ps

12. ssh, scp, ssh-keygen, ssh-copy-id

**1. man :**by using this command you can easily learn how to use

mits@mits-H610M-H-V2-DDR4:~$ man ls

NAME

       ls - list directory contents

SYNOPSIS

       ls [OPTION]... [FILE]...

DESCRIPTION

       List  information  about the FILEs (the current directory by default).  Sort entries alphabetically if none of -cftuvSUX nor --sort

       is specified.

       Mandatory arguments to long options are mandatory for short options too.

       -a, --all

              do not ignore entries starting with .

       -A, --almost-all

              do not list implied . and ..

       --author

              with -l, print the author of each file

       -b, --escape

              print C-style escapes for nongraphic characters

       --block-size=SIZE

              with -l, scale sizes by SIZE when printing them; e.g., '--block-size=M'; see SIZE format below

       -B, --ignore-backups

              do not list implied entries ending with ~

       -c     with -lt: sort by, and show, ctime (time of last modification of file status information); with -l: show ctime and  sort  by

              name; otherwise: sort by ctime, newest first

       -C     list entries by columns

**2. ls, echo, read**

**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.

mits@mits-H610M-H-V2-DDR4:~$ ls

document.docx  document.pdf  india  snap

There are variations you can use with the ls command:

* ls -R will list all the files in the sub-directories aswell

mits@mits-H610M-H-V2-DDR4:~$ ls -R.:document.docx document.pdf india snap

* ls –l – long listing

mits@mits-H610M-H-V2-DDR4:~$ ls -ltotal 252-rw-rw-r-- 1 mits mits 26431 Feb 5 11:36 document.docx-rw-rw-r-- 1 mits mits 218271 Feb 5 10:16 document.pdf-rw-rw-r-- 1 mits mits 454 Feb 5 10:48 india drwx------ 4 mits mits 4096 Feb 5 10:08 snap

* ls -a will show the hidden files

mits@mits-H610M-H-V2-DDR4:~$ ls -a. .bash\_logout .config .fontconfig .lesshst .mca.swp .profile .sudo\_as\_admin\_successful.. .bashrc document.docx .gnupg .local .mca.txt.swp snap .thunderbird.bash\_history .cache document.pdf india .~lock.document.docx# .mozilla .ssh

* ls -al will list the files and directories with detailed information like the perimission, size, owner, etc.

mits@mits-H610M-H-V2-DDR4:~$ ls -altotal 364 -rw-rw-r-- 1 mits mits 84 Feb 5 11:36 .~lock.document.docx#-rw------- 1 mits mits 12288 Apr 15 2024 .mca.swp-rw------- 1 mits mits 12288 Apr 15 2024 .mca.txt.swpdrwx------ 3 mits mits 4096 Oct 10 15:30 .mozilla-rw-r--r-- 1 mits mits 807 Jan 24 2024 .profiledrwx------ 4 mits mits 4096 Feb 5 10:08 snapdrwx------ 2 mits mits 4096 Mar 4 2024 .ssh-rw-r--r-- 1 mits mits 0 Jan 24 2024 .sudo\_as\_admin\_successfuldrwx------ 6 mits mits 4096 Oct 10 15:30 .thunderbird

* ls -t lists files sorted in the order of “lastmodified”

mits@mits-H610M-H-V2-DDR4:~$ ls -tdocument.docx india document.pdf snap

* ls -r option will reverse the natural sorting order. Usually used incombination with other switches such as ls -tr. This will reverse thetime-wise listing.

mits@mits-H610M-H-V2-DDR4:~$ ls -rsnap india document.pdf document.docx

**echo:** echo command is used to move some data into a file. 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 2. head.

mits@mits-H610M-H-V2-DDR4:~$ echo "god is love"

god is love

mits@mits-H610M-H-V2-DDR4:~$ echo -e "god\nis\nlove"

god

is

love

**read:** read the contents of a line into a variable. The read command can be used with and without arguments. read command is used to read [options] [name...] . $read $read var1 var2 var3. $echo &quot;[$var1] [$var2] [$var3].

mits@mits-H610M-H-V2-DDR4:~$ echo "Enter your name:"; read;

Enter your name:

Gokul

mits@mits-H610M-H-V2-DDR4:~$ echo "Enter your name:"; read name;

echo "hello" $name;

Enter your name:

gokul

hello Gokul

**3. more, less, cat**

**more:** Like cat command, more command displays the content of a file. Only difference is that, in case of larger files, &#39; cat&#39; command output will scroll off your screen while &#39; more&#39; command displays output one screenful at a time. Enter key

mits@mits-H610M-H-V2-DDR4:~$ more -p india

india is my country

i love my country

all indians are my brothers and sisters

india, officially the Republic of India,[j][20] is a country in South Asia.

 It is the seventh-largest country by area

since its independence in 1947, the world's most populous democracy

Bounded by the Indian Ocean on the south

he Arabian Sea on the southwest

Bay of Bengal on the southeast

shares land borders with Pakistan to the west

China, Nepal, and Bhutan to the north

**less:** The 'less' command is same as 'more' command but include some more features. It automatically adjusts with the width and height of the terminal window, while 'more' command cuts the content as the width of the terminal window get shorter

mits@mits-H610M-H-V2-DDR4:~$ less india

india is my country

i love my country

all indians are my brothers and sisters

india, officially the Republic of India,[j][20] is a country in South Asia.

 It is the seventh-largest country by area

since its independence in 1947, the world's most populous democracy

Bounded by the Indian Ocean on the south

he Arabian Sea on the southwest

Bay of Bengal on the southeast

shares land borders with Pakistan to the west

China, Nepal, and Bhutan to the north

india (END)

**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.

mits@mits-H610M-H-V2-DDR4:~$ cat > india

india is my country

i love my country

all indians are my brothers and sisters

india, officially the Republic of India,[j][20] is a country in South Asia.

^C

mits@mits-H610M-H-V2-DDR4:~$ cat india

india is my country

i love my country

all indians are my brothers and sisters

india, officially the Republic of India,[j][20] is a country in South Asia

mits@mits-H610M-H-V2-DDR4:~$ cat -n  india

     1 india is my country

     2 i love my country

     3 all indians are my brothers and sisters

india, officially the Republic of India,[j][20] is a country in South Asia.

**4. cd, mkdir, pwd, find**

**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.

mits@ mits-H610M-H-V2-DDR4:~$ cd s1mca

mits@mits-H610M-H-V2-DDR4:~/s1mca$ cd s2mca

mits@mits-H610M-H-V2-DDR4:~/s1mca/s2mca$ cd ..

mits@mits-H610M-H-V2-DDR4:~/s1mca$ cd ..

mits@ mits-H610M-H-V2-DDR4:~$

**mkdir :** Use mkdir command to make a new directory — if you type mkdir Music it will create a directory called Music.To generate a new directory inside another directory, use this Linux basic command

mits@mits-H610M-H-V2-DDR4:~$ lsdocument.docx f1 india kerala linuxtext snapmits@mits-H610M-H-V2-DDR4:~/$ mkdir s1mcamits@mits-H610M-H-V2-DDR4:~/$ mkdir s2mca

mits@mits-H610M-H-V2-DDR4:~$ lsdocument.docx f1 india kerala linuxtext snap s1mca s2mca

**pwd (Print Working Directory):** 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 starts with a forward slash (/). An example of an absolute path is /home/username.

mits@mits-H610M-H-V2-DDR4:~$ pwd

/home/mits

**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.

mits@mits-H610M-H-V2-DDR4:~$:~$find ~ -name "india"

/home/mits/india

mits@mits-H610M-H-V2-DDR4:~$:~$find . -name "\*.txt"

./sample.txt

./file1.txt

./file2.txt

**5. mv, cp, rm ,tar**

**mv :** The primary use of the mv command is to move files, 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. mv file.txt

/home/username/Documents .To rename files, the Linux is mv oldname.ext newname.ext.

mits@mits-H610M-H-V2-DDR4:~/s2mca$ ls

apple  f1  linux  new  orange

mits@mits-H610M-H-V2-DDR4:~/s2mca$ mv f1 f2

mits@mits-H610M-H-V2-DDR4:~/s2mca$ ls

apple  f2  linux  new  orange

**cp :** cp command issued to copy files from the currentdirectorytoa 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 fileisnotexistingor thesource file is newer than the destination file.

mits@mits-H610M-H-V2-DDR4:~/gokul$ ls

file1

mits@mits-H610M-H-V2-DDR4:~/gokul$ cp file1 file2

mits@mits-H610M-H-V2-DDR4:~/gokul$ ls

file1  file2

**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. Be very careful with this command and double-check which directory you are in. This will delete everything and there is no undo. To remove a file use rm filename.

mits@mits-H610M-H-V2-DDR4:~$ lsdocument.docx grapes india kerala linuxtext mca mint s2mca snap yellowmits@mits-H610M-H-V2-DDR4:~$ rm mcarm: cannot remove 'mca': Is a directorymits@mits-H610M-H-V2-DDR4:~$ rm -r mcamits@mits-H610M-H-V2-DDR4:~$ lsdocument.docx grapes india kerala linuxtext mint s2mca snap yellow

**tar:** The Linux ‘tar’stands for tape archive, is used to createArchive and extract theArchive files Linux tar command to create compressed or uncompressed Archive files.

mits@mits-H610M-H-V2-DDR4:~$ tar -cvf archive.tar myfolder

myfolder/

myfolder/sample.txt

**6. wc, cut, paste**

**wc :** wc stands for word count. Used for counting purpose. It is used to find out number of lines, word count, byte and characters count in the files specified in the file arguments. #wc state.txt 6 8 54 state.tx . #wc state.txt capital.txt wc -l state.txt wc

-w state.txt capital.txt wc -c state.txt .wc -m state.txt

mits@mits-H610M-H-V2-DDR4:~$ cat state

Kerala

Tamil nadu

Goa

mits@mits-H610M-H-V2-DDR4:~$ wc state

3 4 22 state

mits@mits-H610M-H-V2-DDR4:~$ wc -l state

3 state

mits@mits-H610M-H-V2-DDR4:~$ wc -c state

22 state

mits@mits-H610M-H-V2-DDR4:~$ wc -w state

4

**cut :** The cut command is used for cutting out the sections from each line offiles and writing the result to standard output. It can be used to cut parts of aline by byte position, character and file.

mits@mits-H610M-H-V2-DDR4:~$ cat > state

andhra pradesh

arunachal pradesh

assam

bihar

Chhattisgarh

india

^C

mits@mits-H610M-H-V2-DDR4:~$ cut -b 1,2,3,4 state

andh

arun

assa

biha

Chha

indi

mits@mits-H610M-H-V2-DDR4:~$ cut -b 1-3,6-7 state

anda

aruch

ass

bih

Chhti

ind

mits@mits-H610M-H-V2-DDR4:~$ cut -b 3- state

dhra pradesh

unachal pradesh

sam

har

hattisgarh

dia

**paste :** It is used to join files horizontally (parallel merging) by outputting lines consisting of lines from each file specified, separated by tab as delimiter, to the standard output. paste [OPTION]... [FILES]...$ paste state.txt capital.txt.

mits@mits-H610M-H-V2-DDR4:~$ cat number

1

2

3

4

5

mits@mits-H610M-H-V2-DDR4:~$ cat state

arunachal pradesh

assam

andhra pradesh

bihar

chattisgrah

mits@mits-H610M-H-V2-DDR4:~$ cat capital

itanagar

dispur

hyderabad

patna

raipur

mits@mits-H610M-H-V2-DDR4:~$ paste number state capital

1   arunachal pradesh   itanagar

2   assam   dispur

3   andhra pradesh  hyderabad

4   bihar   patna

5   chattisgrah raipur

mits@mits-H610M-H-V2-DDR4:~$ paste -d "|" number state capital

1|arunachal pradesh|itanagar

2|assam|dispur

3|andhra pradesh|hyderabad

4|bihar|patna

5|chattisgrah|Raipur

mits@mits-H610M-H-V2-DDR4:~$ paste -d "," number state capital

1,arunachal pradesh,itanagar

2,assam,dispur

3,andhra pradesh,hyderabad

4,bihar,patna

5,chattisgrah,raipur

mits@mits-H610M-H-V2-DDR4:~$ paste -d "|," number state capital

1|arunachal pradesh,itanagar

2|assam,dispur

3|andhra pradesh,hyderabad

4|bihar,patna

5|chattisgrah,raipur

mits@mits-H610M-H-V2-DDR4:~$ paste -s number state capital

1   2   3   4   5

arunachal pradesh   assam   andhra pradesh  bihar   chattisgrah

itanagar    dispur  hyderabad   patna   raipur

mits@mits-H610M-H-V2-DDR4:~$ paste -s -d ":" number state capital

1:2:3:4:5

arunachal pradesh:assam:andhra pradesh:bihar:chattisgrah

itanagar:dispur:hyderabad:patna:raipu

**7. head, tail, grep, expr**

**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. If you only want to show the first five lines, type head -n 5 filename.txt.

mits@mits-H610M-H-V2-DDR4:~$ head india

india is my country

i love my country

all indians are my brothers and sisters

india, officially the Republic of India,[j][20] is a country in South Asia.

 It is the seventh-largest country by area

since its independence in 1947, the world's most populous democracy

Bounded by the Indian Ocean on the south

he Arabian Sea on the southwest

Bay of Bengal on the southeast

shares land borders with Pakistan to the west

mits@mits-H610M-H-V2-DDR4:~$ head -5 india

india is my country

i love my country

all indians are my brothers and sisters

india, officially the Republic of India,[j][20] is a country in South Asia.

 It is the seventh-largest country by area

**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. tail -n filename.txt.

mits@mits-H610M-H-V2-DDR4:~$ tail india

i love my country

all indians are my brothers and sisters

india, officially the Republic of India,[j][20] is a country in South Asia.

 It is the seventh-largest country by area

since its independence in 1947, the world's most populous democracy

Bounded by the Indian Ocean on the south

he Arabian Sea on the southwest

Bay of Bengal on the southeast

shares land borders with Pakistan to the west

China, Nepal, and Bhutan to the north

mits@mits-H610M-H-V2-DDR4:~$ tail -5 india

Bounded by the Indian Ocean on the south

he Arabian Sea on the southwest

Bay of Bengal on the southeast

shares land borders with Pakistan to the west

China, Nepal, and Bhutan to the north

**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. Usually output of a previous command is piped into the grep command. For example, ls -l |grep “kernel”.

mits@mits-H610M-H-V2-DDR4:~$ cat exmple

unix is great os. unix was developed in Bell labs.

learn operating system.

Unix linux which one you choose.

uNix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.

mits@mits-H610M-H-V2-DDR4:~$ grep -c "unix" exmple

2

mits@mits-H610M-H-V2-DDR4:~$ grep -h "unix" exmple

unix is great os. unix was developed in Bell labs.

uNix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.

mits@mits-H610M-H-V2-DDR4:~$ grep -i "unix" exmple

unix is great os. unix was developed in Bell labs.

Unix linux which one you choose.

uNix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.

mits@mits-H610M-H-V2-DDR4:~$ grep -l "unix" exmple

exmple

mits@mits-H610M-H-V2-DDR4:~$ grep -n "unix" exmple

1:unix is great os. unix was developed in Bell labs.

4:uNix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.

**expr :** The expr command evaluates a given expression and displays its corresponding output. It is used for: . Basic operations like addition, subtraction, multiplication, division, and modulus on integers. Evaluating regular expressions,string operations like substring, length of strings etc. Performing operations on variables inside a shell script.

mits@mits-H610M-H-V2-DDR4:~$ a="20"

mits@mits-H610M-H-V2-DDR4:~$ b="4"

mits@mits-H610M-H-V2-DDR4:~$ expr $a + $b

24

mits@mits-H610M-H-V2-DDR4:~$ expr $a - $b

16

mits@mits-H610M-H-V2-DDR4:~$ expr $a \\* $b

80

mits@mits-H610M-H-V2-DDR4:~$ expr $a / $b

5

**8. chmod, chown**

**chmod :** To change directory permissions of file/ Directory in Linux. #chmod who what which file/directory 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. #chmod u+x test #chmod g- rwx test #chmod o-r test 4

mits@mits-H610M-H-V2-DDR4:~$ ls -l

drwxrwxr-x 2 mits mits    4096 Feb 20 11:47  mca

-rw-rw-r-- 1 mits mits      18 Feb  6 14:16  mint

-rw-rw-r-- 1 mits mits     148 Mar  5 11:22  myfile1.txt

-rw-rw-r-- 1 mits mits     148 Mar  5 11:22  myfile2.txt

mits@mits-H610M-H-V2-DDR4:~$ chmod 777 myfile1.txt

mits@mits-H610M-H-V2-DDR4:~$ ls -l

drwxrwxr-x 2 mits mits    4096 Feb 20 11:47  mca

-rw-rw-r-- 1 mits mits      18 Feb  6 14:16  mint

-rwxrwxrwx 1 mits mits     148 Mar  5 11:22  myfile1.txt

-rw-rw-r-- 1 mits mits     148 Mar  5 11:22  myfile2.txt

**chown :** The chown command allows you to change the user and/or group ownership of a given file, directory. #chownTom Test

mits@mits-H610M-H-V2-DDR4:~$ ls -l

-rw-rw-r-- 1 mits mits     148 Mar  5 11:22  myfile1.txt

-rw-rw-r-- 1 mits mits     148 Mar  5 11:22  myfile2.txt

mits@mits-H610M-H-V2-DDR4:~$ sudo chown gokul myfile1.txt

mits@mits-H610M-H-V2-DDR4:~$ ls -l

-rwxrwxrwx 1 gokul mits     148 Mar  5 11:22  myfile1.txt

-rw-rw-r-- 1 mits  mits     148 Mar  5 11:22  myfile2.txt

**9. Redirections & Piping:** A pipe is a form of redirection to send the output of one command/program/process to another command/program/process for further processing. Pipe is used to combine two or more commands, the output of one command acts as input to another command, and this command’s output may act as input to the next command and so on.

mits@mits-H610M-H-V2-DDR4:~$echo "Hello" > output.txt

mits@mits-H610M-H-V2-DDR4:~$echo "Linux" >> output.txt

mits@mits-H610M-H-V2-DDR4:~$ls -l | grep ".txt"

**10. useradd, usermod, userdel, passwd**

**useradd :** 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

mits@mits-H610M-H-V2-DDR4:~$ sudo useradd gokul

[sudo] password for mits:

mits@mits-H610M-H-V2-DDR4:~$ cat /etc/passwd | grep gokul

gokul:x:1003:1003::/home/gokul:/bin/sh

mits@mits-H610M-H-V2-DDR4:~$ cat /etc/passwd | grep "gokul"

gokul:x:1003:1003::/home/gokul:/bin/sh

.

**usermod :** usermod command is used to change the properties of a user in Linux through the command line command-line utility that allows you to modify a user’s login information.

mits@mits-H610M-H-V2-DDR4:~$ sudo usermod -l gookul gokul

mits@mits-H610M-H-V2-DDR4:~$ cat /etc/passwd | grep "gookul"

gookul:x:1003:1004::/home/gokul:/bin/sh

**userdel :**Remove a user is very similar to adding a new user. To delete theusers account type, userdel UserName

mits@mits-H610M-H-V2-DDR4:~$ sudo userdel gokul

mits@mits-H610M-H-V2-DDR4:~$ cat /etc/passwd | tail -3

mca:x:1001:1001::/home/mca:/bin/sh

mysql:x:128:136:MySQL Server,,,:/nonexistent:/bin/false

exam:x:1002:1002:Exam,,,:/home/exam:/bin/bash

**passwd :** Changes passwords for user accounts. A normal user may only change the password for their own account, while the superuser may change the password for any account

**11. df,top, ps**

**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.

mits@mits-H610M-H-V2-DDR4:~$ df

Filesystem     1K-blocks     Used Available Use% Mounted on

tmpfs             789976     2284    787692   1% /run

/dev/sda5      216888480 16665692 189132664   9% /

tmpfs            3949860        0   3949860   0% /dev/shm

tmpfs               5120        4      5116   1% /run/lock

efivarfs             256      125       127  50% /sys/firmware/efi/efivars

/dev/sda1          98304    31816     66488  33% /boot/efi

**top:** top command is used to show the Linux processes. It provides a dynamic real- time view of the running system

Syntax:

top [options]

mits@mits-H610M-H-V2-DDR4:~$ top

top - 14:35:28 up  1:28,  1 user,  load average: 0.52, 0.44, 0.39

Tasks: 337 total,   1 running, 336 sleeping,   0 stopped,   0 zombie

%Cpu(s):  1.0 us,  0.7 sy,  0.0 ni, 98.2 id,  0.0 wa,  0.0 hi,  0.1 si,  0.0 st

MiB Mem :   7714.6 total,    447.0 free,   3179.0 used,   4088.6 buff/cache

MiB Swap:   2048.0 total,   2048.0 free,      0.0 used.   3508.2 avail Mem

**ps :** The ps command, short for Process Status, is a command line utility that is used to display or view information related to the processes running in a Linux system. PID – This is the unique process ID TTY– This is the type of terminal that the user is logged in to . TIME – This is the time in minutes and seconds that the process has been running .CMD – The command that launched the process

Syntax:

ps [options]

mits@mits-H610M-H-V2-DDR4:~/gokul$ ps

    PID TTY          TIME CMD

   4182 pts/0    00:00:00 bash

   9697 pts/0    00:00:00 ps

**3.Execute the following scenario using basic Linux commands**

a) Login to your home directory

b) List contents of your current working directory

c) List all contents of your current working directory, including hidden files

d) Make a directory called April2024 inside your current working directory. Change to the directory April2024

e) Create an empty file name file1

f) Make a copy of file1 to file2

g) Copy file1 from the current working directory and save it as the name file2 in one Directory up from the current directory

h) Clear the terminal window

mits@mits-H610M-H-V2-DDR4:~$ pwd

/home/mits

mits@mits-H610M-H-V2-DDR4:~$ ls

document.docx  document.pdf  india  snap

mits@mits-H610M-H-V2-DDR4:~$ ls -a

.    .bash\_logout  .config  .fontconfig  .lesshst   .mca.swp      .profile  .sudo\_as\_admin\_successful

..  .bashrc       document.docx  .gnupg       .local       .mca.txt.swp  snap      .thunderbird

.bash\_history  .cache        document.pdf   india  .~lock.document.docx#  .mozilla   .ssh

mits@mits-H610M-H-V2-DDR4:~$ mkdir April2024

mits@mits-H610M-H-V2-DDR4:~$ cd April2024

mits@mits-H610M-H-V2-DDR4:~/April2024$ touch file1

mits@mits-H610M-H-V2-DDR4:~/April2024$ ls

file1

mits@mits-H610M-H-V2-DDR4:~/April2024$ cp file1 file2

mits@mits-H610M-H-V2-DDR4:~/April2024$ ls

file1 file2

mits@mits-H610M-H-V2-DDR4:~/April2024$ cp file1 ../file2

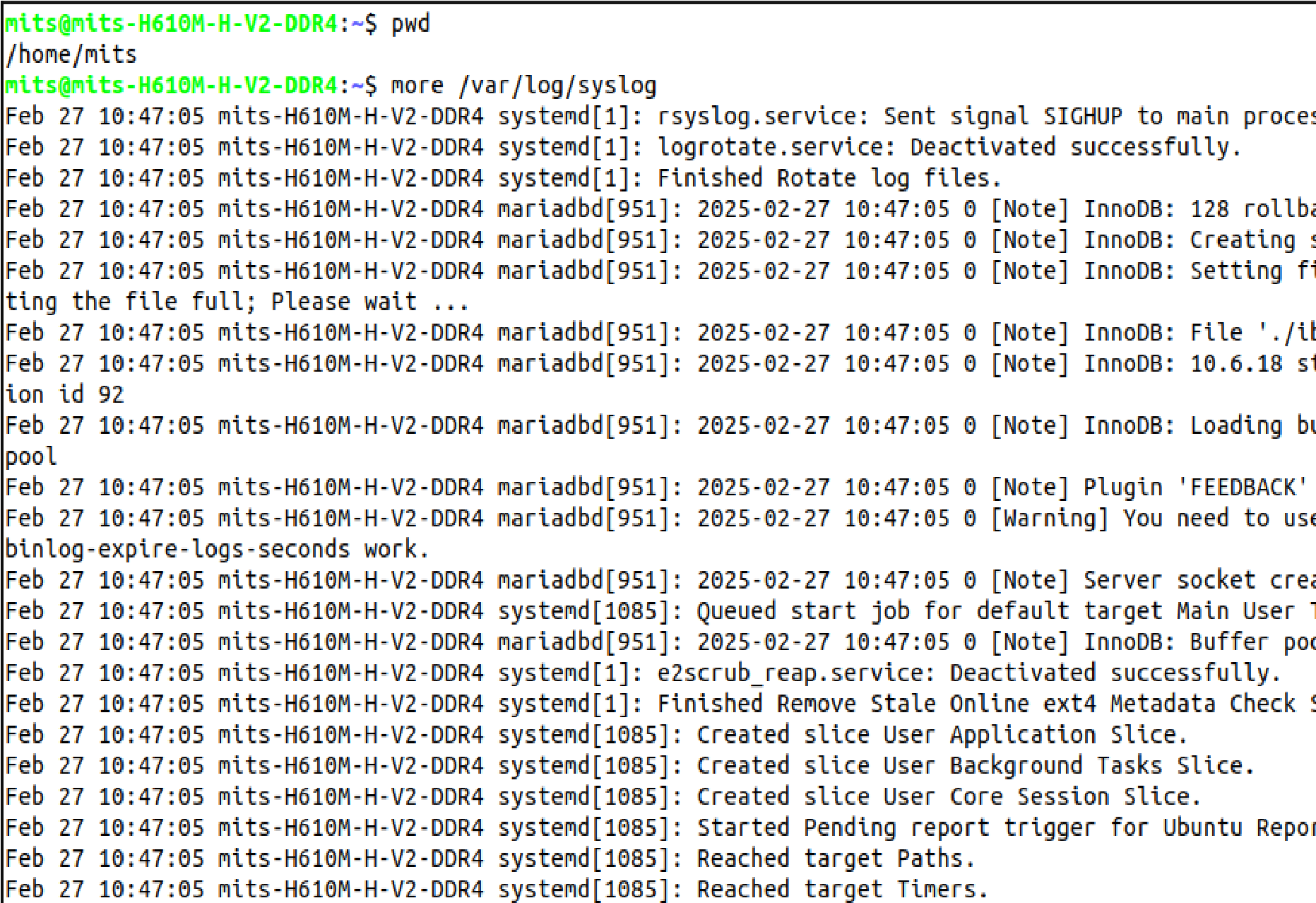
mits@mits-H610M-H-V2-DDR4:~/April2024$ clear

**4.Execute the following scenario using basic Linux commands**

a) Login to your home directory

b) Write the contents of syslog (located in the /var/log/ directory) onto thescreen a page at a time.

c) Read documentation on a command: less



**5.Execute the following scenario using basic Linux Commands**

a) Create an untitled document myfile.txt using anyone editor

b) Place the following text in myfile.txt and save it

Neo: What are you trying to tell me?

That I can dodge bullets?

Morpheus: No, Neo. I'm trying to tell

you that when you're ready, you won't have to.

c) Count the number of characters, words, and lines in the file

d) Find the occurrence of the word “tell” in the file e.

e) Make two copies of myfile.txt with names myfile1.txt and myfile2.txt

f) List all the filenames with the word file in the present working directory

mits@nits-H610M-H-V2-DDR4:~$ touch myfile.txt

mits@mits-H610M-H-V2-DDR4:~$ gedit myfile.txt

mits@mits-H610M-H-V2-DDR4:~$ cat myfile.txt

Neo: What are you trying to tell me?

That I can dodge bullets?

Morpheus: No, Neo. I'm trying to tell

you that when you're ready, you won't have to.

mits@mits-H610M-H-V2-DDR4:~$ wc myfile.txt

4 29 148 myfile.txt

mits@mits-H610M-H-V2-DDR4:~$ wc -c myfile.txt

148 myfile.txt

mits@mits-H610M-H-V2-DDR4:~$ wc -w myfile.txt

29 nyfile.txt

mits@mits-H610M-H-V2-DDR4:~$ wc -l myfile.txt

4 myfile.txt

mits@mits-H610M-H-V2-DDR4:~$ grep -c "tell" myfile.txt

2

mits@mits -H610M-H-V2-DDR4:~$ grep "tell" myfile.txt

Neo: What are you trying to tell me?

Morpheus: No, Neo. I'm trying to tell

mits@mits-H610M-H-V2-DDR4:~$ cp myfile.txt myfile1.txt

mits@nits-H610M-H-V2-DDR4:~$ cp myfile.txt myfile2.txt

mits@mits-H610M-H-V2-DDR4:-$ ls \*file\*

file2 myfile1.txt myfile2.txt myfile.txt

**6.File System Hierarchy**

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

**Procedure**

1. /root – Root User Directory

This is the home directory of the root user. The root user’s home directory is located at /root. Which is noteworthy because it is, unlike the rest of the users’ home directories, not located in /home.

2. /bin – essential utilities

The directory contains the core system programs and important utilities. For example, commonly used and well known commands such as “cat” are located in “/bin”. The reason for this is that if these utilities are not stored in this directory, there is no certainty that the system will have access to them if there isn’t a file system mounted.

3. /etc – Configuration files

The configuration files of BIOS and other similar files can be found in /etc. You can edit these configuration files in a text editor as you see fit. Basically, every single kind of configuration file is located in /etc, including but not limited to system configuration files.

4. /sbin – System Administration Programs

The /sbin directory is similar to the /bin directory in that it contains essential programs. But it differs with the addition that it is intended to be used by the root user.

5. /usr – User Shared Read-Only Data

The /usr directory is used to contain applications and files that are used and shared by and between users.

6. /var – Variable Data

The /var directory is used like the /usr directory, only instead of being read-only, it is writable. This directory contains system logs and other various variable data.

7. /dev – Devices or Files

Linux displays connected devices as files and the /dev directory contains these files. Though, the thing is, as you can see by the title, these are not “actual” files, they just appear as files. /dev is also where physical drives can be mounted.

8. /home – Home Folder Containment

There’s a home folder for every user on your system and each one is contained together in the /home directory. These folders are created using the name of your user name.

For example, your user name is jaise, so your home folder would be located in /home/jaise.

These home folders contain your user data files and configuration files that are specific to the user, which is also the one of the only types of configuration files that are stored elsewhere besides /etc as we explained above. If one wants to modify other files on a system, they must become the root user, as each user only has write permissions for their own home folder.

9. /lib – Libraries for Programs

Each program or binary uses specific libraries to function and the /lib directory is where these libraries can be located.

10. /mnt – Temporary Mounts

This directory is used for mounting temporary file systems. If you are using a file system for a very specific purpose and for a relatively brief period of time, you would probably mount it in /mnt. Though you can mount it anywhere on the system if you so choose.

11. /opt – Optional Packages

The /opt directory contains a set of subdirectories where optional software packages are located and managed by the package manager.

12. /proc – Kernel and Process Pseudo Files

The /proc directory is another interesting case of a directory that contains these “fake” files, very similarly to the /dev directory that we discussed earlier in this list. These files are special files that are actually, and interestingly, system and process information.

13. /root – Root User Directory

Every user has his own home directory. This is the home directory of the root user. The root user’s home directory is located at /root.

Which is noteworthy because it is, unlike the rest of the users’ home directories, not located in /home. Like we’ve said above, in an earlier section of this article, /root is different from the root directory “/”, and this fact should be committed to memory if possible.

**Output Screenshot**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**7.Shell Script**

1. Write a shell script to find sum,product,quotient,difference & remainder of 2 numbers

**PROGRAM**

echo "enter a and b"

read a b

sum=`expr $a + $b`

diff=`expr $a - $b`

pro=`expr $a \\* $b`

quo=`expr $a / $b`

mod=`expr $a % $b`

echo "sum="$sum

echo "diff="$diff

echo "pro="$pro

echo "quo="$quo

echo "mod="$mod

**OUTPUT**

mits@mits-H610M-H-V2-DDR4:~/gokul linux$ bash arithmetic.sh

enter a and b

15 6

sum=21

diff=9

pro=90

quo=2

mod=3

2. Write a shell script to swap 2 numbers with & without using a temporary variable.

**PROGRAM**

echo "enter a and b"

read a b

echo "before swap"

echo $a $b

a=$((a+b))

b=$((a-b))

a=$((a-b))

echo "after swap"

echo $a $b

**OUTPUT**

mits@mits-H610M-H-V2-DDR4:~/gokul linux$ bash swap.sh

enter a and b

8 6

before swap

8 6

after swap

6 8

3.Write a shell script that accepts two integers as its arguments and computes the value of the first number raised to the power of the second number.

**PROGRAM**

echo "enter a and b"

read a b

c=$((a\*\*b))

echo $c

**OUTPUT**

mits@mits-H610M-H-V2-DDR4:~/gokul linux$ bash power.sh

enter a and b

6 3

216

4.Write a shell script to add two floating point numbers.

**PROGRAM**

echo "enter a and b"

read a b

echo "scale=3;$a+$b" | bc

**OUTPUT**

mits@mits-H610M-H-V2-DDR4:~/gokul$ bash floatadd.sh

enter a and b

4.5 6.7

11.2

5.Assign the values 5,6,10,2 to the variables a,b,c &d. Write a shell script to evaluate the expression (a\*b\*c)/d

**PROGRAM**

a=5

b=6

c=10

d=2

echo $(((a\*b\*c)/d))

**OUTPUT**

mits@mits-H610M-H-V2-DDR4:~/gokul$ bash expr.sh

150

6.Write a shell script to

a)Find the length of a word/sentence

**PROGRAM**

echo "enter a sentence"

read a

echo "length=${#a}"

**OUTPUT**

mits@mits-H610M-H-V2-DDR4:~/gokul$ bash strings1.sh

enter a sentence

welcome

length=7

b)Concatenate 2 strings

**PROGRAM**

echo "enter string1"

read b

echo "enter string2"

read c

d=${b}${c}

echo "after concatenate=$d"

**OUTPUT**

mits@mits-H610M-H-V2-DDR4:~/gokul$ bash strings2.sh

enter string1

gokul

enter string2

raj

after concatenate=gokulraj

c)Find & replace string

**PROGRAM**

read -p "Enter the original string: " original

read -p "Enter the string to find: " to\_find

read -p "Enter the string to replace with: " replace\_with

mod=${original//$to\_find/$replace\_with}

echo "The modified string is: $mod"

**OUTPUT**

mits@mits-H610M-H-V2-DDR4:~/gokul$ bash strings3.sh

Enter the original string: hello welcome all

Enter the string to find: all

Enter the string to replace with: everyone

The modified string is: hello welcome everyone

7.Write a shell script to find simple interest.

**PROGRAM**

echo "enter amount"

read p

echo "enter rate of interest"

read r

echo "enter no of years"

read n

value=$(echo "($p\*$n\*$r)/100" | bc)

echo $value

**OUTPUT**

mits@mits-H610M-H-V2-DDR4:~/gokul$ bash interest.sh

enter amount

5600

enter rate of interest

4

enter no of years

5

1120

8. Write a shell script to find

a.Area & circumference of a circle

**PROGRAM**

echo "enter radius"

read r

area=$(echo "3.14\*$r\*$r" | bc)

perim=$(echo "2\*3.14\*$r" | bc)

echo "area of traingle:"$area

echo "perimeter of traingle:"$perim

**OUTPUT**

mits@mits-H610M-H-V2-DDR4:~/gokul$ bash circleq.sh

enter radius

5

area of traingle:78.50

perimeter of traingle:31.40

b.Area & perimeter of a rectangle

**PROGRAM**

echo "enter length of rectangle"

read l

echo "enter breadth of rectangle"

read b

area=$(echo "$l\*$b" | bc)

perim=$(echo "2\*($l+$b)" | bc)

echo "area of rectangle:"$area

echo "perimeter of rectangle:"$perim

**OUTPUT**

mits@mits-H610M-H-V2-DDR4:~/gokul linux$ bash rectangleq.sh

enter length of rectangle

6

enter breadth of rectangle

5

area of rectangle:30

perimeter of rectangle:22

c.Area & perimeter of a square

**PROGRAM**

echo "enter side of square"

read s

area=$(echo "$s\*$s" | bc)

perim=$(echo "4\*$s" | bc)

echo "area of traingle:"$area

echo "perimeter of traingle:"$perim

**OUTPUT**

mits@mits-H610M-H-V2-DDR4:~/gokul linux$ bash sqaureq.sh

enter side of square

5

area of traingle:25

perimeter of traingle:20

9. Write a shell script to find the largest of 2 numbers.

**PROGRAM**

echo "enter value 1"

read x

echo "enter value 2"

read y

if [ $x -gt $y ]

then

echo $x "is greater"

elif [ $y -gt $x ]

then

echo $y "is greater"

else

echo "both equal"

fi

**OUTPUT**

mits@mits-H610M-H-V2-DDR4:~/gokul linux$ bash large2.sh

enter value 1

12

enter value 2

5

12 is greater

10. Write a shell script that computes the gross salary of a employee according to the following rules :

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

ii) If basic salary is >=1500 then HRA =Rs500 and DA=98% of the basic.

**PROGRAM**

echo "enter employee name"

read n

echo "enter basic salary"

read bs

if [ $bs -lt 1500 ]

then

hra=$(echo "$bs/10" | bc)

da=$(echo "($bs\*9)/10" | bc)

gs=$(echo "$bs+$hra+$da" | bc)

echo "name:"$n

echo "gross salary:"$gs

else

da=$(echo "($bs\*98)/100" | bc)

gs=$(echo "$bs+500+$da" | bc)

echo "name:"$n

echo "gross salary:"$gs

fi

**OUTPUT**

mits@mits-H610M-H-V2-DDR4:~/gokul linux$ bash salary.sh

enter employee name

gokul

enter basic salary

1800

name:gokul

gross salary:4064

mits@mits-H610M-H-V2-DDR4:~/gokul linux$ bash salary.sh

enter employee name

gokul

enter basic salary

1200

name:gokul

gross salary:2400

11. Write a shell script to find the largest of 3 numbers.

**PROGRAM**

echo "enter value 1"

read x

echo "enter value 2"

read y

echo "enter value 3"

read z

if [ $x -gt $y ]

then

if [ $x -gt $z ]

then

echo $x "is greater"

else

echo $z "is greater"

fi

else

if [ $y -gt $z ]

then

echo $y "is greater"

else

echo $z "is greater"

fi

fi

**OUTPUT**

mits@mits-H610M-H-V2-DDR4:~/gokul linux$ bash large3.sh

enter value 1

4

enter value 2

10

enter value 3

6

10 is greater

12.Write a shell script that receives any number of file names as arguments check if every arguments applied is a file or a directory and reports accordingly,whenever the argument is a file or directory.

**PROGRAM**

for f in "$@"

do

if [ -f $f ]

then

echo "$f is a file"

elif [ -d $f ]

then

echo "$f is a directory"

else

echo "enter valid file"

fi

done

**OUTPUT**

mits@mits-H610M-H-V2-DDR4:~/gokul linux$ bash checkfile.sh s2mca

s2mca is a directory

mits@mits-H610M-H-V2-DDR4:~/gokul linux$ bash checkfile.sh abcd

enter valid file

mits@mits-H610M-H-V2-DDR4:~/gokul linux$ bash checkfile.sh circleq.sh

circleq.sh is a file

13. Write a shell script to calculate the sum of digits of a number

**PROGRAM**

echo "enter number"

read n

s=0

while [ $n -gt 0 ]

do

k=$(( $n % 10 ))

n=$(( $n / 10 ))

s=$(( $s + $k ))

done

echo "sum of digit=$s"

**OUTPUT**

mits@mits-H610M-H-V2-DDR4:~/gokul linux$ bash dgsum.sh

enter number

562

sum of digit=13

14. Write a shell script to display the first 10 even numbers.

**PROGRAM**

a=2

while [ $a -le 20 ]

do

echo $a

a=`expr $a + 2`

done

**OUTPUT**

mits@mits-H610M-H-V2-DDR4:~/gokul linux$ bash evenno.sh

2

4

6

8

10

12

14

16

18

20

15. Write a shell script to display even numbers less than 10

**PROGRAM**

a=2

while [ $a -lt 10 ]

do

echo $a

a=`expr $a + 2`

done

**OUTPUT**

mits@mits-H610M-H-V2-DDR4:~/gokul linux$ bash evenless.sh

2

4

6

8

16. Write a shell script to convert the contents of a file into uppercase.

**PROGRAM**

echo "enter file name"

read b

if [ ! -f $b ]

then

echo "file doesnt exsist"

else

tr 'a-z' 'A-Z'<$b

fi

**OUTPUT**

mits@mits-H610M-H-V2-DDR4:~/gokul linux$ cat file3

hello welcome

mits@mits-H610M-H-V2-DDR4:~/gokul linux$ bash toupper.sh

enter file name

file3

HELLO WELCOME

17. Write a shell script that delete all lines containing a specified word

**PROGRAM**

read -p "Enter file name:" fname

if [ -f $fname ]

then

read -p "Enter word to delete:" word

echo "File before removing $word:"

cat $fname

grep -v -i $word $fname > test

mv test $fname

echo "File after removing $word:"

cat $fname

else

echo "The file $fname is not existing"

fi

**OUTPUT**

mits@mits-H610M-H-V2-DDR4:~/gokul linux$ bash dltline.sh

Enter file name:file1

Enter word to delete:hi

File before removing hi:

hello

hi

welcome

abc

File after removing hi:

hello

welcome

abc

18. Write a shell script to find the factorial of given integer.

**PROGRAM**

echo "enter number"

read a

fact=1

for ((i=1;i<=a;i++))

do

fact=$((fact\*i))

done

echo "factorial of $a is $fact"

**OUTPUT**

mits@mits-H610M-H-V2-DDR4:~/gokul linux$ bash fact.sh

enter number

5

factorial of 5 is 120

19. Write a shell script to find whether a given number is prime

**PROGRAM**

echo "Enter the number"

read num

for ((i=2;i<num;i++))

do

if [ $((num % i)) -eq 0 ]

then

echo "$num is not a prime number."

exit

fi

done

echo "$num is a prime."

**OUTPUT**

mits@mits-H610M-H-V2-DDR4:~/gokul linux$ bash primenum.sh

Enter the number

23

23 is a prime.

20. Write a shell script to print the pattern

1

2 2

3 3 3

4 4 4 4

**PROGRAM**

for ((i=1;i<=4;i++))

do

for ((j=1;j<=i;j++))

do

echo -n $i ""

done

echo ""

done

**OUTPUT**

mits@mits-H610M-H-V2-DDR4:~/gokul linux$ bash pattern.sh

1

2 2

3 3 3

4 4 4 4

21.Write a shell program to perform simple calculator operations like addition, subtraction,multiplication or division depending upon the user input

**PROGRAM**

echo "enter number 1"

read num1

echo "enter number 2"

read num2

s=0

while [ $s -eq 0 ]

do

echo "1.addition

2.substraction

3.multiplication

4.division

5.modulus

6.exit"

echo "choose operation"

read n

case $n in

"1")

sum=$((num1+num2))

echo "sum=$sum"

;;

"2")

diff=$((num1-num2))

echo "difference=$diff"

;;

"3")

pro=$((num1\*num2))

echo "product=$pro"

;;

"4")

quo=$((num1/num2))

echo "quotient=$quo"

;;

"5")

mod=$((num1%num2))

echo "modulus=$mod"

;;

"6")

echo "exit"

s=1

break;

;;

\*)

echo -n "Invalid"

;;

esac

done

**OUTPUT**

mits@mits-H610M-H-V2-DDR4:~/gokul linux$ bash calculator.sh

enter number 1

10

enter number 2

4

1.addition

2.substraction

3.multiplication

4.division

5.modulus

6.exit

choose operation

1

sum=14

1.addition

2.substraction

3.multiplication

4.division

5.modulus

6.exit

choose operation

2

difference=6

1.addition

2.substraction

3.multiplication

4.division

5.modulus

6.exit

choose operation

3

product=40

1.addition

2.substraction

3.multiplication

4.division

5.modulus

6.exit

choose operation

4

quotient=2

1.addition

2.substraction

3.multiplication

4.division

5.modulus

6.exit

choose operation

5

modulus=2

1.addition

2.substraction

3.multiplication

4.division

5.modulus

6.exit

choose operation

6

exit

22. Write a shell program to find the factorial of a given number using until loop.

**PROGRAM**

echo "enter number"

read n

i=1

fact=1

until [[ $i -gt $n ]]

do

fact=$((fact\*i))

((i++))

done

echo "factorial : $fact"

**OUTPUT**

mits@mits-H610M-H-V2-DDR4:~/gokul linux$ bash factusinguntil.sh

enter number

6

factorial : 720

mits@mits-H610M-H-V2-DDR4:~/gokul linux$ bash factusinguntil.sh

enter number

5

factorial : 120

23. Write a menu driven shell program

a.to find sum of digits of a number

b.find the largest of two numbers

c.to display sum of squares of the digits of a number

d.exit

**PROGRAM**

s=0

while [ $s -eq 0 ]

do

echo "1.sum of digit of number

2.largest of two number

3.sum of squares of digit of number

4.exit"

echo "choose operation"

read n

case $n in

"1")

echo "enter a number"

read n1

sum=0

while [ $n1 -gt 0 ]

do

k=$(( $n1 % 10 ))

n1=$(( $n1 / 10 ))

sum=$(( $sum + $k ))

done

echo "sum of digit=$sum"

;;

"2")

echo "enter number 1"

read n2

echo "enter number 2"

read n3

if [ $n2 -gt $n3 ]

then

echo "$n2 is greater"

elif [ $n3 -gt $n2 ]

then

echo "$n3 is greater"

else

echo "both equal"

fi

;;

"3")

echo "enter a number"

read n4

sum1=0

while [ $n4 -gt 0 ]

do

k=$(( $n4 % 10 ))

n4=$(( $n4 / 10 ))

sum1=$(( $sum1 + $k\*\*2 ))

done

echo "sum of squares of digit=$sum1"

;;

"4")

echo "exit"

s=1

;;

\*)

echo -n "Invalid"

;;

esac

done

**OUTPUT**

mits@mits-H610M-H-V2-DDR4:~/gokul linux$ bash menudrivenpgm.sh

1.sum of digit of number

2.largest of two number

3.sum of squares of digit of number

4.exit

choose operation

1

enter a number

234

sum of digit=9

1.sum of digit of number

2.largest of two number

3.sum of squares of digit of number

4.exit

choose operation

2

enter number 1

15

enter number 2

13

15 is greater

1.sum of digit of number

2.largest of two number

3.sum of squares of digit of number

4.exit

choose operation

3

enter a number

123

sum of squares of digit=14

1.sum of digit of number

2.largest of two number

3.sum of squares of digit of number

4.exit

choose operation

4

exit

24. Write a menu driven shell program

1.puffs

2.bun

3.juice

4.exit

**PROGRAM**

s=0

sum1=0

sum2=0

sum3=0

sum4=0

while [ $s -eq 0 ]

do

echo "              FOOD MENU                 "

echo "----------------------------------------"

echo "ITEMS "

echo "1. Puffs  -20 Rs"

echo "2. BUN  -7 Rs"

echo "3. JUICE  -10 Rs"

echo "choose item"

read n

case $n in

"1")

echo "Enter Quantity Of Puffs"

read q1

sum1=$(($sum1+(q1\*20)))

sum4=$((sum4+sum1))

;;

"2")

echo "Enter Quantity Of Bun"

read q2

sum2=$(($sum2+(q2\*7)))

sum4=$((sum4+sum2))

;;

"3")

echo "Enter Quantity Of of Juice"

read q3

sum3=$(($sum3+(q3\*10)))

sum4=$((sum4+sum3))

;;

\*)

echo "invalid"

;;

esac

echo "do you have more items to order Yes-0 No-1"

read s

done

echo "------------------------------------------------------"

echo "                     CAFE 24"

echo "------------------------------------------------------"

echo " BILL "

echo "------------------------------------------------------"

echo " Item       Unit Price       Quantity       Price"

echo "------------------------------------------------------"

if [ $q1 -gt 0 ]

then

echo " PUFFS      20 Rs            $q1              $sum1 Rs"

fi

if [ $q2 -gt 0 ]

then

echo " BUN        7 Rs             $q2              $sum2 Rs"

fi

if [ $q3 -gt 0 ]

then

echo " JUICE      20 Rs            $q3              $sum3 Rs"

fi

echo "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"

echo " "

echo "TOTAL BILL AMOUNT: $sum4 Rs"

echo "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"

echo "THANK YOU FOR SHOPPING"

echo "------------------------------------------------------"

**OUTPUT**

mits@mits-H610M-H-V2-DDR4:~/gokul linux$ bash menudrivenpgm2.sh

FOOD MENU

----------------------------------------

ITEMS

1. Puffs -20 Rs

2. BUN -7 Rs

3. JUICE -10 Rs

choose item

1

Enter Quantity Of Puffs

4

do you have more items to order Yes-0 No-1

0

FOOD MENU

----------------------------------------

ITEMS

1. Puffs -20 Rs

2. BUN -7 Rs

3. JUICE -10 Rs

choose item

2

Enter Quantity Of Bun

2

do you have more items to order Yes-0 No-1

0

FOOD MENU

----------------------------------------

ITEMS

1. Puffs -20 Rs

2. BUN -7 Rs

3. JUICE -10 Rs

choose item

3

Enter Quantity Of of Juice

4

do you have more items to order Yes-0 No-1

1

------------------------------------------------------

CAFE 24

------------------------------------------------------

BILL

------------------------------------------------------

Item Unit Price Quantity Price

------------------------------------------------------

PUFFS 20 Rs 4 80 Rs

BUN 7 Rs 2 14 Rs

JUICE 20 Rs 4 40 Rs

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

TOTAL BILL AMOUNT: 134

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

THANK YOU FOR SHOPPING

------------------------------------------------------

25. Write a shell script to prepare mark list of s2mca students.Input internal and extrenal marks obtained by student for 4 theory subjects adbms, acn, ipr, ob. Register no of students must be mcao1,mca02...etc..

**PROGRAM**

echo "enter no of students"

read n

for ((i=1;i<=n;i++))

do

echo "enter student name"

read name[$i]

echo "enter internal mark of IPR"

read ipri[$i]

echo "enter external mark of IPR"

read ipre[$i]

echo "enter internal mark of ADBMS"

read adbmsi[$i]

echo "enter external mark of ADBMS"

read adbmse[$i]

echo "enter internal mark of OB"

read obi[$i]

echo "enter external mark of OB"

read obe[$i]

echo "enter internal mark of ACN"

read acni[$i]

echo "enter external mark of ACN"

read acne[$i]

iprt[$i]=$((ipri[$i]+ipre[$i]))

adbmst[$i]=$((adbmsi[$i]+adbmse[$i]))

obt[$i]=$((obi[$i]+obe[$i]))

acnt[$i]=$((acni[$i]+acne[$i]))

done

echo ""

for ((i=1;i<=n;i++))

do

echo ""

echo "                            Mark List          "

echo "                           \*\*\*\*\*\*\*\*\*\*\*         "

echo ""

echo "Name:${name[$i]}"

echo "RegNo:MUTMCA$i"

echo ""

echo "-------------------------------------------------------------------------------"

echo " Subject Name        Internal Mark        External Mark        Total Mark      "

echo " IPR                 ${ipri[$i]}                   ${ipre[$i]}                   ${iprt[$i]}         "

echo " ADBMS               ${adbmsi[$i]}                   ${adbmse[$i]}                   ${adbmst[$i]}       "

echo " OB                  ${obi[$i]}                   ${obe[$i]}                   ${obt[$i]}          "

echo " ACN                 ${acni[$i]}                   ${acne[$i]}                   ${acnt[$i]}         "

done

**OUTPUT**

mits@mits-H610M-H-V2-DDR4:~/gokul linux$ bash marklist.sh

enter no of students

2

enter student name

gokul

enter internal mark of IPR

12

enter external mark of IPR

12

enter internal mark of ADBMS

13

enter external mark of ADBMS

13

enter internal mark of OB

14

enter external mark of OB

14

enter internal mark of ACN

15

enter external mark of ACN

15

enter student name

abhijith

enter internal mark of IPR

21

enter external mark of IPR

21

enter internal mark of ADBMS

22

enter external mark of ADBMS

22

enter internal mark of OB

23

enter external mark of OB

23

enter internal mark of ACN

24

enter external mark of ACN

24

                            Mark List

                           \*\*\*\*\*\*\*\*\*\*\*

Name:gokul

RegNo:MUTMCA1

-------------------------------------------------------------------------------

 Subject Name        Internal Mark        External Mark        Total Mark

 IPR                 12                   12                   24

 ADBMS               13                   13                   26

 OB                   14                   14                   28

 ACN                 15                   15                   30

                            Mark List

                           \*\*\*\*\*\*\*\*\*\*\*

Name:abhijith

RegNo:MUTMCA2

-------------------------------------------------------------------------------

 Subject Name        Internal Mark        External Mark        Total Mark

 IPR                 21                   21                   42

 ADBMS               22                   22                   44

 OB                   23                   23                   46

 ACN                 24                   24                   48

**8.LAMP INSTALLATION PROCEDURE**

**Install apache**

updating the local package index to reflect the latest upstream changes:

**sudo apt update && sudo apt install ufw**

Then, install the apache2 package:

**sudo apt install apache2**

Install Apache using apt:

It is recommended that you enable the most restrictive profile that will still allow the traffic You have configured. Since we haven’t configured SSL for our server yet in this guide, we will only need to allow traffic on port 80:

**sudo ufw allow 'Apache'**

You can verify the change by typing:

sudo ufw status

The output will provide a list

**Output**

Status: active

To Action From

|  |  |  |
| --- | --- | --- |
| OpenSSH | ALLOW | Anywhere |
| Apache | ALLOW | Anywhere |
| OpenSSH (v6) | ALLOW | Anywhere (v6) |
| Apache (v6) | ALLOW | Anywhere (v6) |

**Checking your Web Server**

At the end of the installation process, Ubuntu 20.04 starts Apache. The web server should already be up and running.

Check with the systemd init system to make sure the service is running by typing:

**sudo systemctl status apache2**

**Output**

● apache2.service - The Apache HTTP Server

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

active (running) since Thu 2020-04-23 22:36:30 UTC; 20h ago

Docs: https://httpd.apache.org/docs/2.4/ Main PID:

29435 (apache2)

Tasks: 55 (limit: 1137)

Memory: 8.0M

CGroup: /system.slice/apache2.service

├─29435 /usr/sbin/apache2 -k start

├─29437 /usr/sbin/apache2 -k start

└─29438 /usr/sbin/apache2 -k start

As confirmed by this output, the service has started successfully. However, the best way to test this is to request a page from Apache.

We can access the default Apache landing page to confirm that the software is running properly through your IP address. If you do not know your server’s IP address, you can get it a few different ways from the command line.

Try typing this at your server’s command prompt:

**hostname -I**

We will get back a few addresses separated by spaces. You can try each in your web browser to determine if they work.

Another option is to use the Icanhazip tool, which should give you your public IP address as read from another location on the internet:

**curl -4 icanhazip.com**

When you have your server’s IP address, enter it into your browser’s address bar:

**http://your\_server\_ip**

You should see the default Ubuntu 20.04 Apache web page:



**Install mariadb**

* Install mariaDB

**sudo apt install mariadb-server mariadb-client**

* Check mariadb Installation

**sudo systemctl status mysql**

(if it is not working sudo systemctl start mysql )

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Install PHP**

**Install phpmyadmin**

Install phpmyadmin

**sudo apt install phpmyadminphp-mbstringphp-zipphp-gdphp-jsonphp-curl**

( It ask for webserver select apache2, select dbconfiguration and set password )

Restart apache2

sudo systemctl restart apache2

**Ansible installation**

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.

**Update OS to latest version. Command:** sudo apt-get update

root@localhost:~# sudo apt-get update

Hit:1 <http://mirrors.linode.com/ubuntu> cosmic InRelease

Get:2 <http://mirrors.linode.com/ubuntu> cosmic-updates InRelease [88.7 kB] Get:3 <http://mirrors.linode.com/ubuntu> cosmic-backports InRelease [74.6 kB]

Get:4 <http://mirrors.linode.com/ubuntu> cosmic-updates/main i386 Packages [200 kB] Get:5 <http://mirrors.linode.com/ubuntu> cosmic-updates/main amd64 Packages [223 kB] Get:6 <http://security.ubuntu.com/ubuntu> cosmic-security InRelease [88.7 kB]

Get:7 <http://security.ubuntu.com/ubuntu> cosmic-security/main amd64 Packages [131 kB] Get:8 <http://security.ubuntu.com/ubuntu> cosmic-security/main i386 Packages [109 kB] Fetched 915 kB in 1s (700 kB/s)

Reading package lists... Done

**Add Ansible package to your Ubuntu system.**

Command: sudo apt-add-repository --yes --update ppa:ansible/ansible root@localhost:~# sudo apt-add-repository --yes --update ppa:ansible/ansible

Hit:1 <http://mirrors.linode.com/ubuntu> cosmic InRelease

Hit:2 <http://mirrors.linode.com/ubuntu> cosmic-updates InRelease

Get:3 <http://mirrors.linode.com/ubuntu> cosmic-backports InRelease [74.6 kB] Hit:4 <http://security.ubuntu.com/ubuntu> cosmic-security InRelease

Get:5 <http://ppa.launchpad.net/ansible/ansible/ubuntu> cosmic InRelease [15.9 kB]

Get:6 <http://ppa.launchpad.net/ansible/ansible/ubuntu> cosmic/main i386 Packages [536 B]

Get:7 <http://ppa.launchpad.net/ansible/ansible/ubuntu> cosmic/main amd64 Packages[536B]

Get:8 http://ppa.launchpad.net/ansible/ansible/ubuntu/ cosmic/main Translation-en [344 B] Fetched 91.9 kB in 1s (64.7 kB/s)

Reading package lists... Done

**Install Ansible completely.**

**COMMAND   sudo apt install ansible.**

A screenshot of a computer

AI-generated content may be incorrect.

**COMMAND : ansible –version**

A screen shot of a computer

AI-generated content may be incorrect.

**Install Ansible and pyVmomi**

Python should already be installed on Mac but you might need to install pip by running this command.

Once pip is installed, we can use it to install Ansible with the following command.

We also need to install pyVmomi which is the Python SDK for the VMware vSphere API that allows you to manage ESX, ESXi, and vCenter.

That’s all the dependencies installed, we’re now ready to create our Ansible playbook.

**Create Ansible Playbook**

Ansible playbooks are YAML configuratiom files that describe what actions to run on a remote host. For this example, we’ll create a simple playbook called deploy-vms.yml that will use the vmware\_guest module to deply a VM from template.

Create the file.

**vim deploy-vms.yml**

Add the following contents to the file. You’ll want to change the variables in the vars section to match the details of your vCenter.

---

- hosts: localhost

gather\_facts: no

vars:

vcenter\_server: "10.1.1.100"

vcenter\_user: ["administrator@vsphe](mailto:administrator@vsphere.local)r[e.local"](mailto:administrator@vsphere.local)

vcenter\_pass: "Pa$$w0rd"

datacenter\_name: "Datacenter"

cluster\_name: "Cluster"

tasks:

- name: Clone the template

vmware\_guest:

hostname: "{{ vcenter\_server }}"

username: "{{ vcenter\_user }}"

password: "{{ vcenter\_pass }}"

validate\_certs: False

name: web02

template: template-ubuntu-18.04

datacenter: "{{ datacenter\_name }}"

folder: /{{ datacenter\_name }}/vm

cluster: "{{ cluster\_name }}"

datastore: "iscsi-datastore01"

networks:

- name: VM Network

ip: 10.1.1.22

netmask: 255.255.255.0

gateway: 10.1.1.2

type: static

dns\_servers: 10.1.1.2

customization:

hostname: "web02"

dns\_servers:

- 10.1.1.2

- 1.1.1.1

state: poweredon

wait\_for\_ip\_address: yes

delegate\_to: localhost

The playbook above will create a new VM called web02 and place it on a datastore called iscsi-datastore01. It will be cloned from a template called template-ubuntu-18.04.

Run the playbook

After making the required changes to the deploy-vms.yml file, save it, then run the following command to deploy the VM.

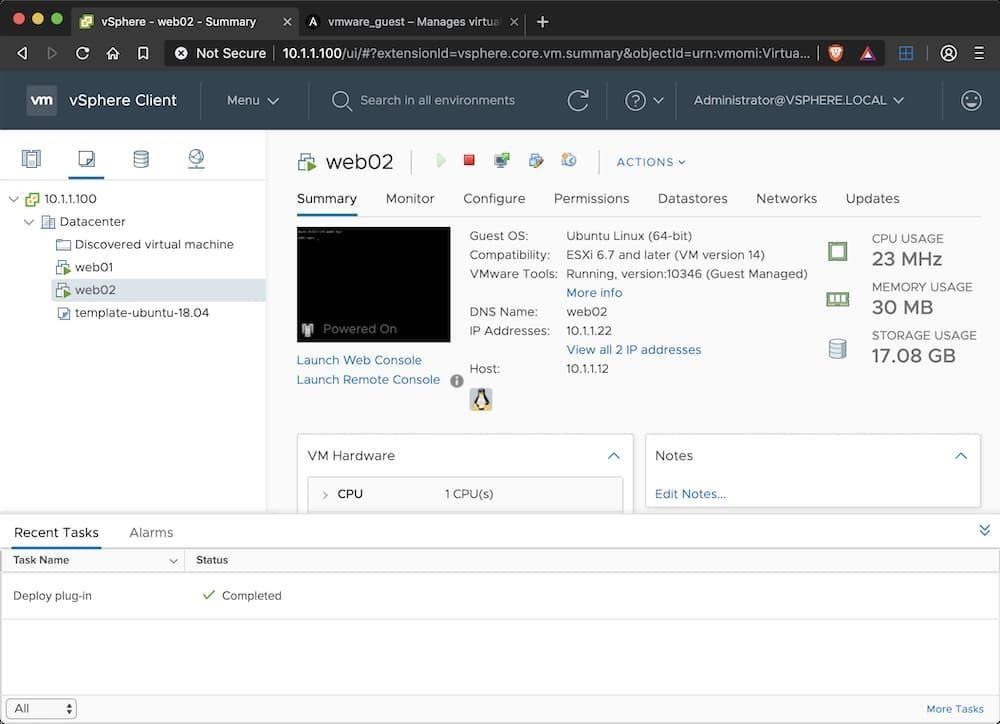
**ansible-playbook deploy-vms.yml**

The screenshot below shows the playbook in action.



The playbook might take a while to complete because we used the wait\_for\_ip\_address: yes option, which means the Ansible command finishes once the VM has been cloned, and the network is configured with the static IP address specified.

The screenshot below shows the VM has been created in vCenter.



**TCPDUMP**

Execute tcpdump and its options on your own system,and submit the output screenshot as a document.

A computer screen with white text

AI-generated content may be incorrect.

Sudo tcpdump

A computer screen with white text

AI-generated content may be incorrect.

Sudo apt update

A computer screen with white text

AI-generated content may be incorrect.

Sudo tcpdump

A screenshot of a computer program

AI-generated content may be incorrect.

Sudo tcpdump -D

A computer screen with white text

AI-generated content may be incorrect.

Sudo tcpdump –c 5

A computer screen shot of a program

AI-generated content may be incorrect.

**Sudo tcpdump –i enp2s0**

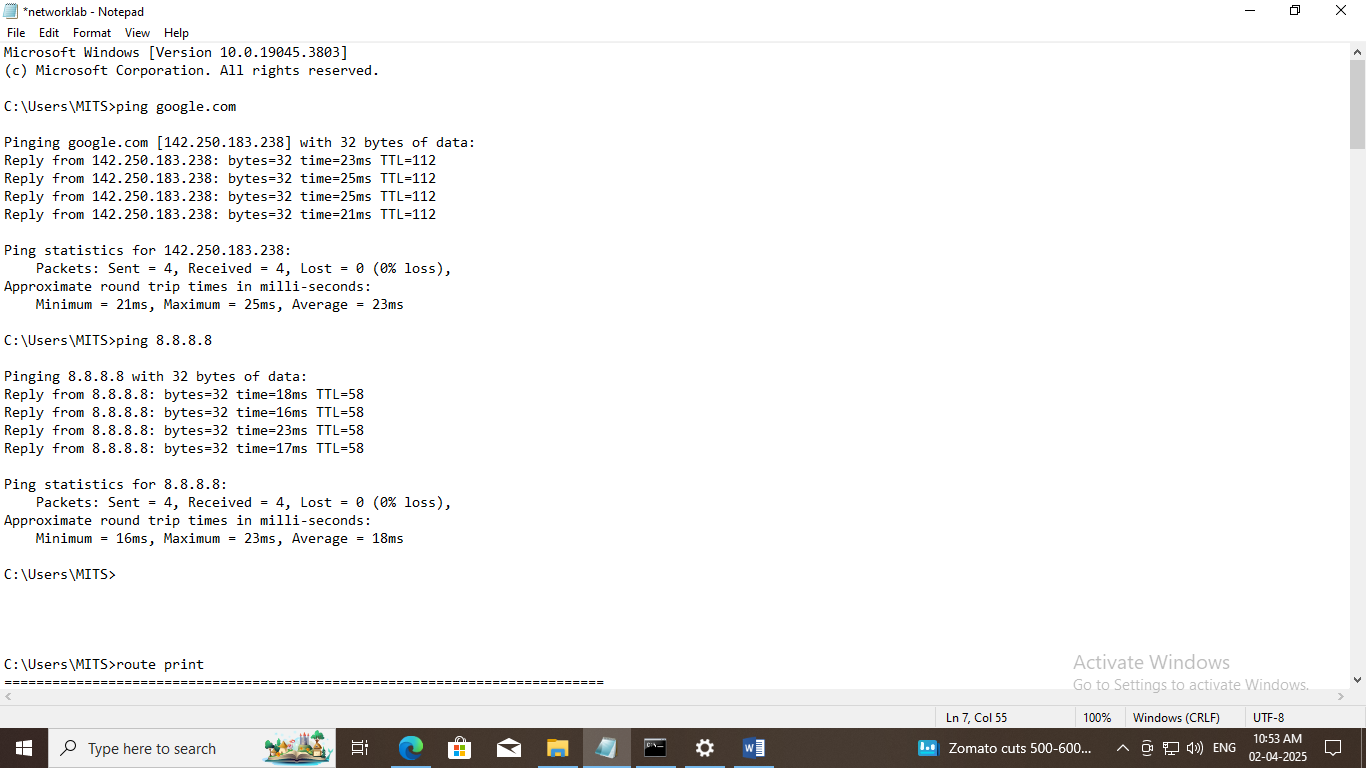
A screenshot of a computer program

AI-generated content may be incorrect.

**9.Network Commands**

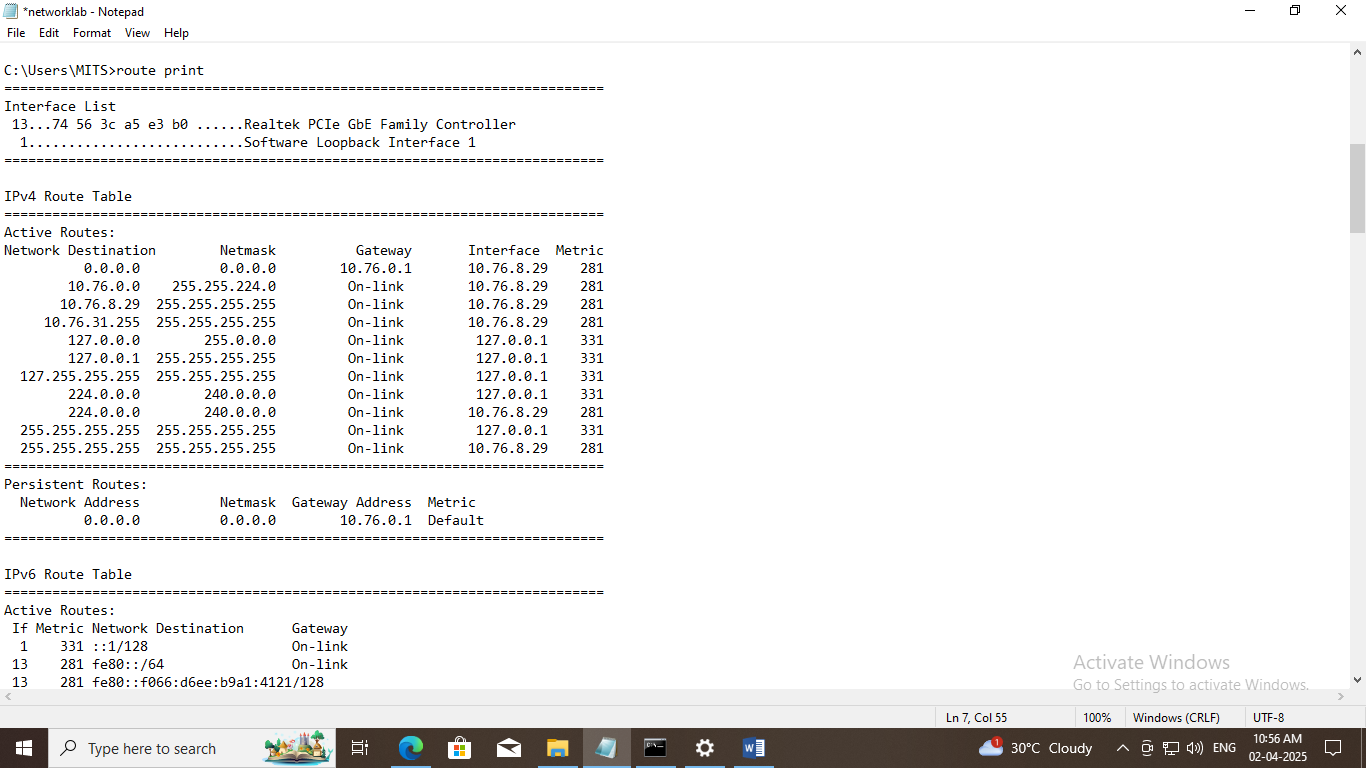
**1.Ping**

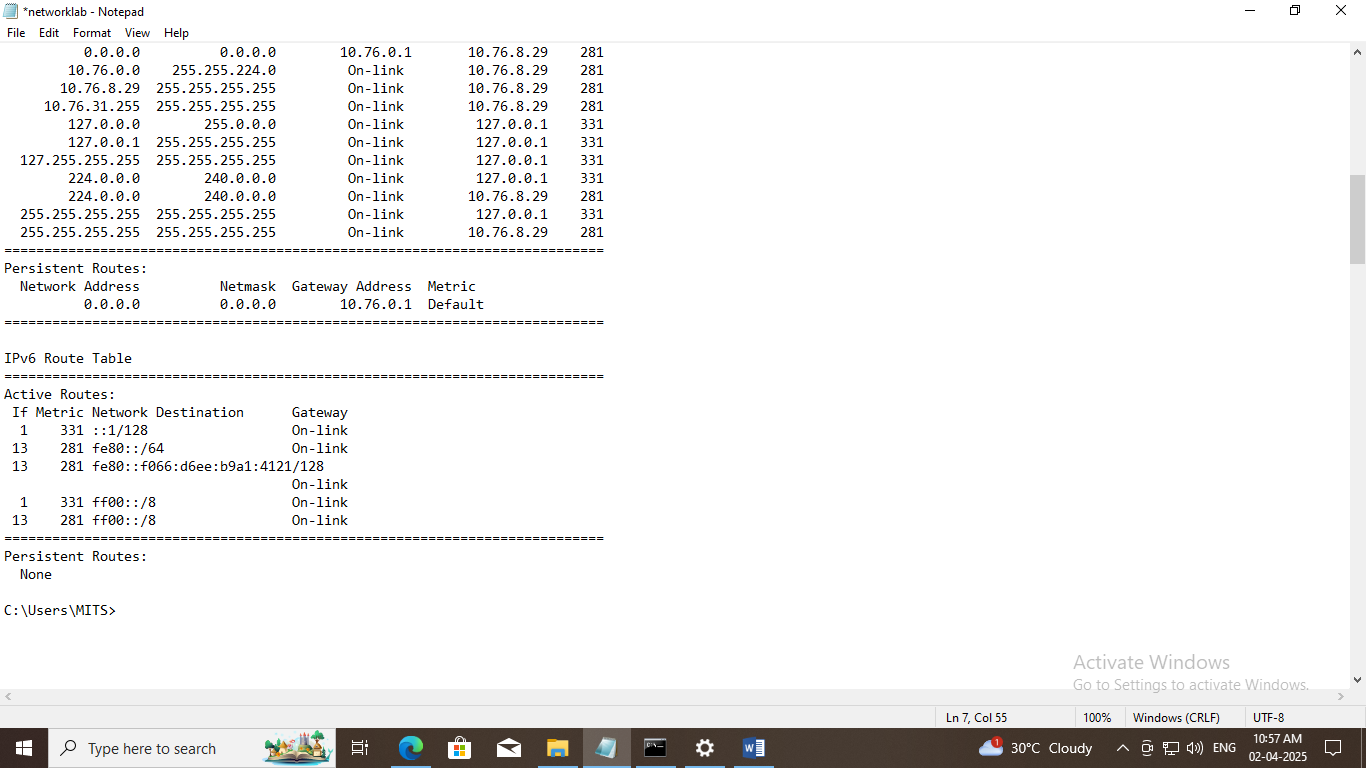
A ping is a basic Internet command that allows a user to test and verify whether a given destination IP address exists and can accept requests in computer network administration. Ping is also used for diagnosis to confirm that the computer the user tries to reach is operational. Ping can be used with any operating system (OS) that supports networking, including the majority of embedded network administration software.



**2.Route**

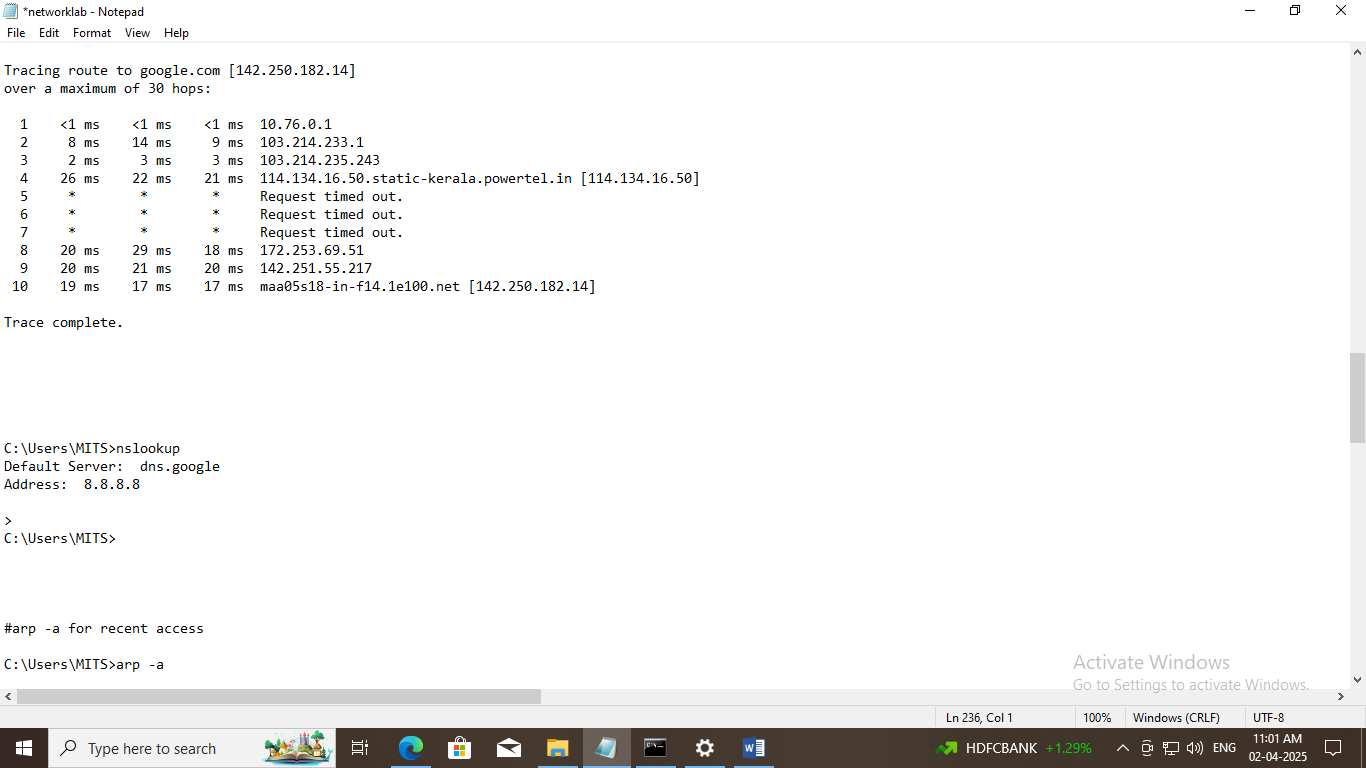
The Route command is a powerful tool used to view and manipulate the IP routing table in various operating systems, including Unix-like systems and Microsoft Windows. This command is essential for network administrators to manage network routes and ensure efficient data transmission.





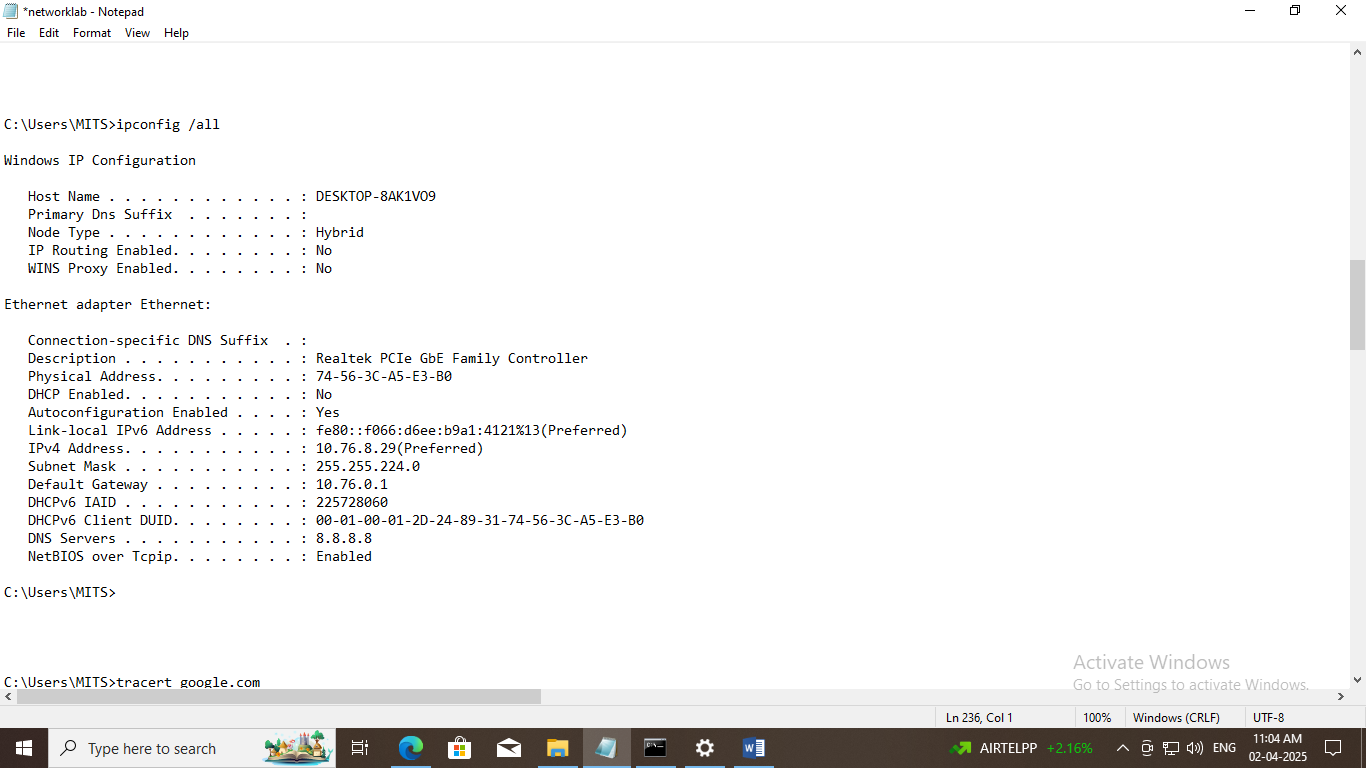
**3.Nslookup**

This command helps diagnose the Domain Name System (DNS) infrastructure and comes with a number of sub-commands. These are mainly for systems administrators. The primary interest for average PC users is its use to find the computer name corresponding to a numeric IP. For example, if you want to know who is &quot;216.109.112.135&quot; , enter &quot;nslookup 216.109.112.135&quot; and you will find that it is (or was anyway) a Yahoo computer. My firewall keeps a log of the IPs involved in the attempts to probe my computer and I sometimes look a few up to see who they are. (There are also Who is search sites available on the Web as mentioned in the Ipconfig section.)



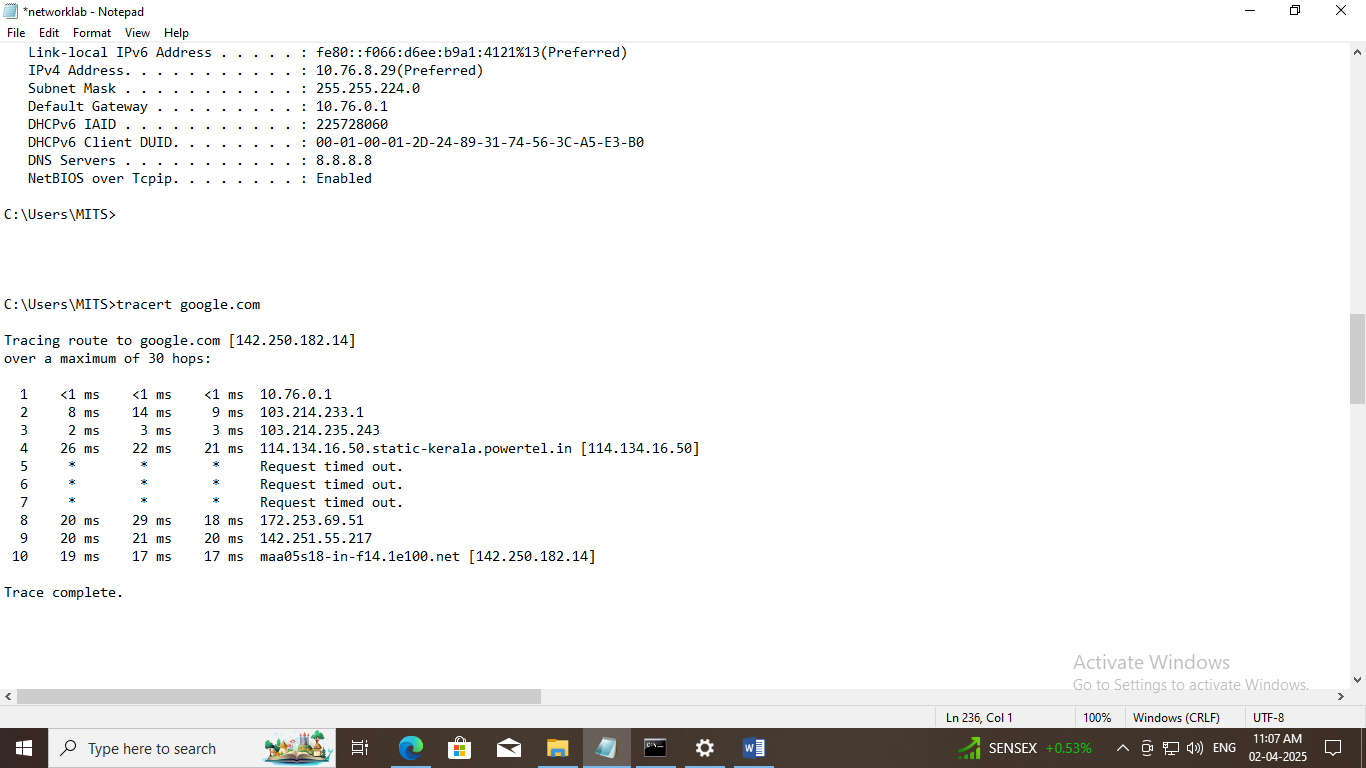
**4. Ipconfig**

The Windows IP Configuration tool (ipconfig) is the command-line equivalent of the accessory &quot;Winipcfg&quot; that was present in Windows 9X/Me. It is used to display the TCP/IP network configuration values. To open it, enter &quot;ipconfig&quot; in the command prompt. If you are connected directly to the Internet, you will obtain your IP address.



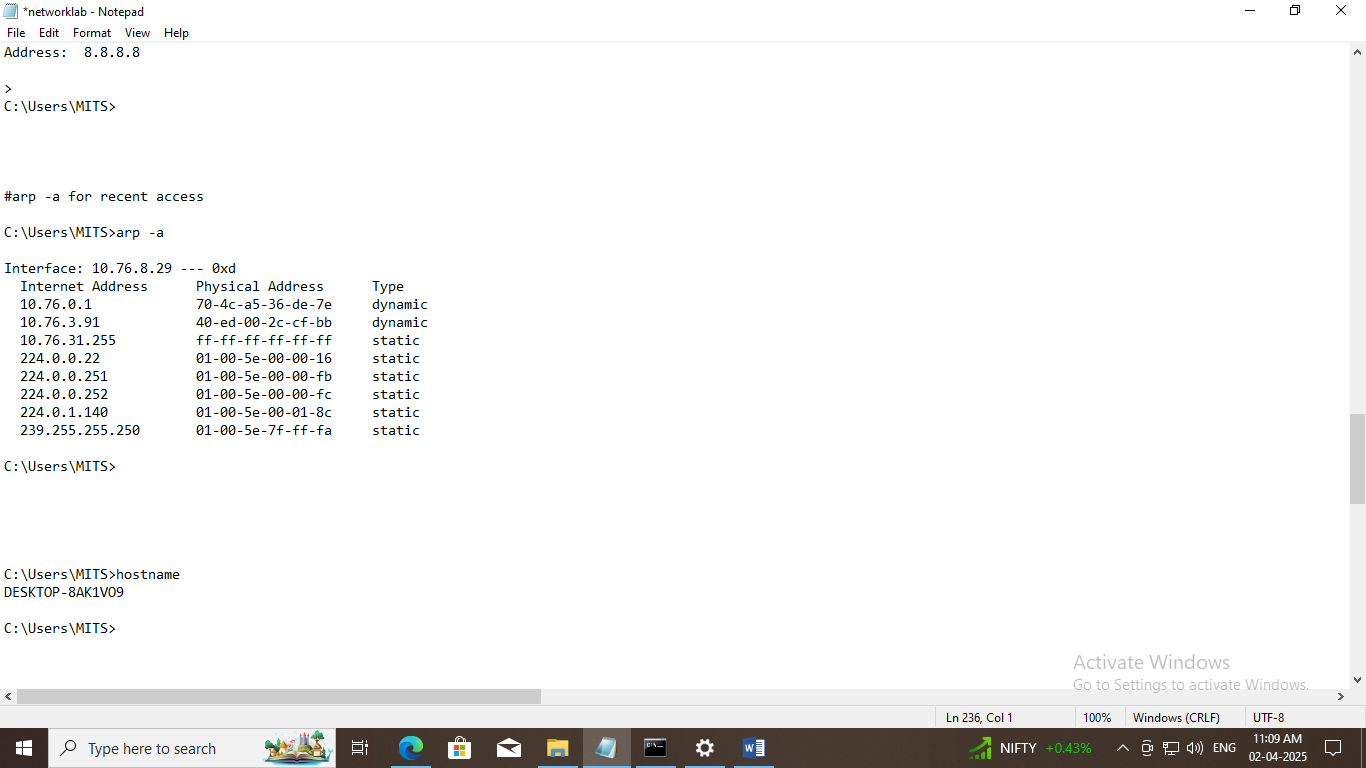
**5.Tracert**

Tracert (traceroute) is another old tool borrowed from Unix. The actual path between two computers on the Internet is not a straight line but consists of numerous segments or &quot;hops&quot; from one intermediate computer to another. Tracert shows each step of the path taken. It can be interesting to see just how convolute edit is. The times for each hop and the IP addresses for each intermediate computer are displayed. Tracert shows up to 30 hops. It is convenient for finding if there is one particular segment that is causing a slow or bad connection. A typical command might be &quot;tracert dell.com&quot;.



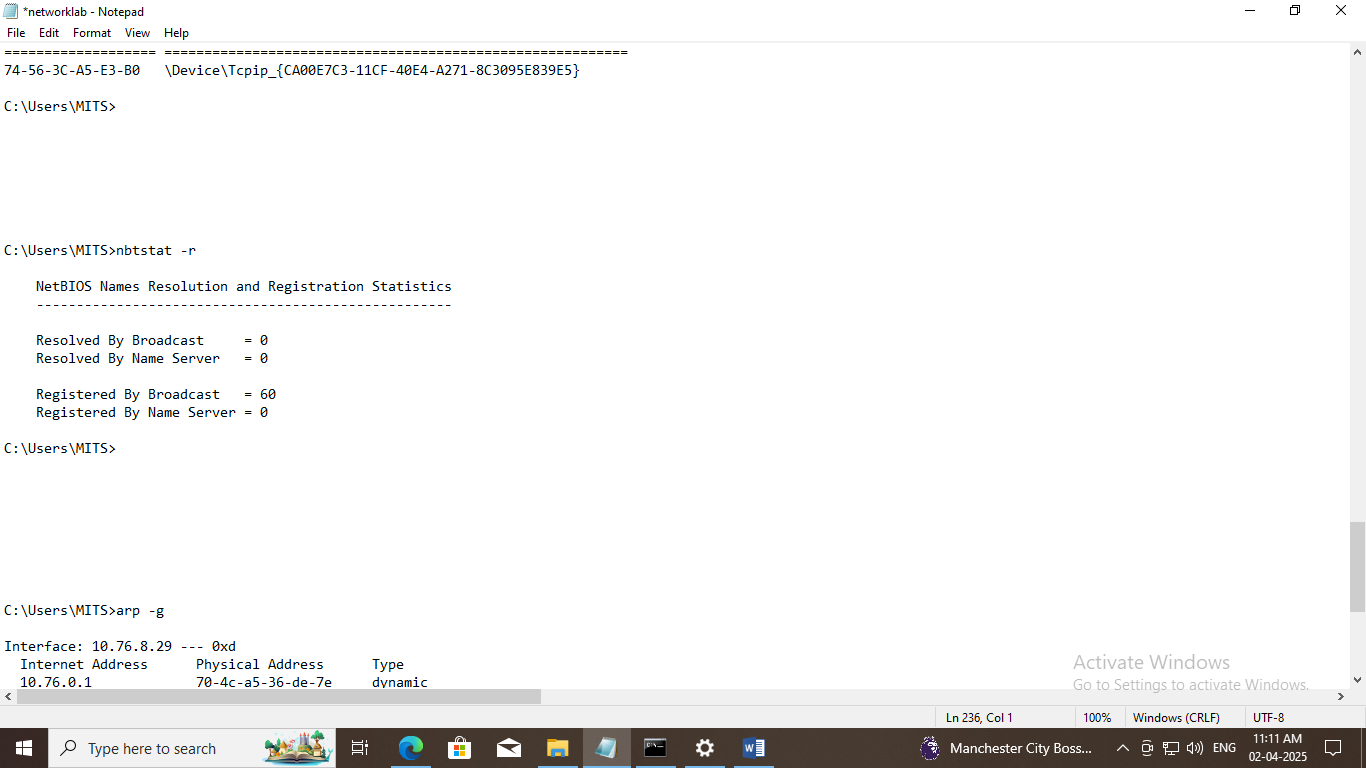
**6.arp**

The ARP command corresponds to the Address Resolution Protocol. Although it is easy to think of network communications in terms of IP addressing, packet delivery is ultimately dependent on the Media Access Control (MAC) address of the device’s network adapter. This is where the Address Resolution Protocol comes into play. Its job is to map IP addresses to MAC addresses. Windows devices maintain an ARP cache, which contains the results of recent ARP queries. You can see the contents of this cache by using the ARP -A command. If you are having problems communicating with one specific host, you can append the remote host’s IP address to the ARP -A command.



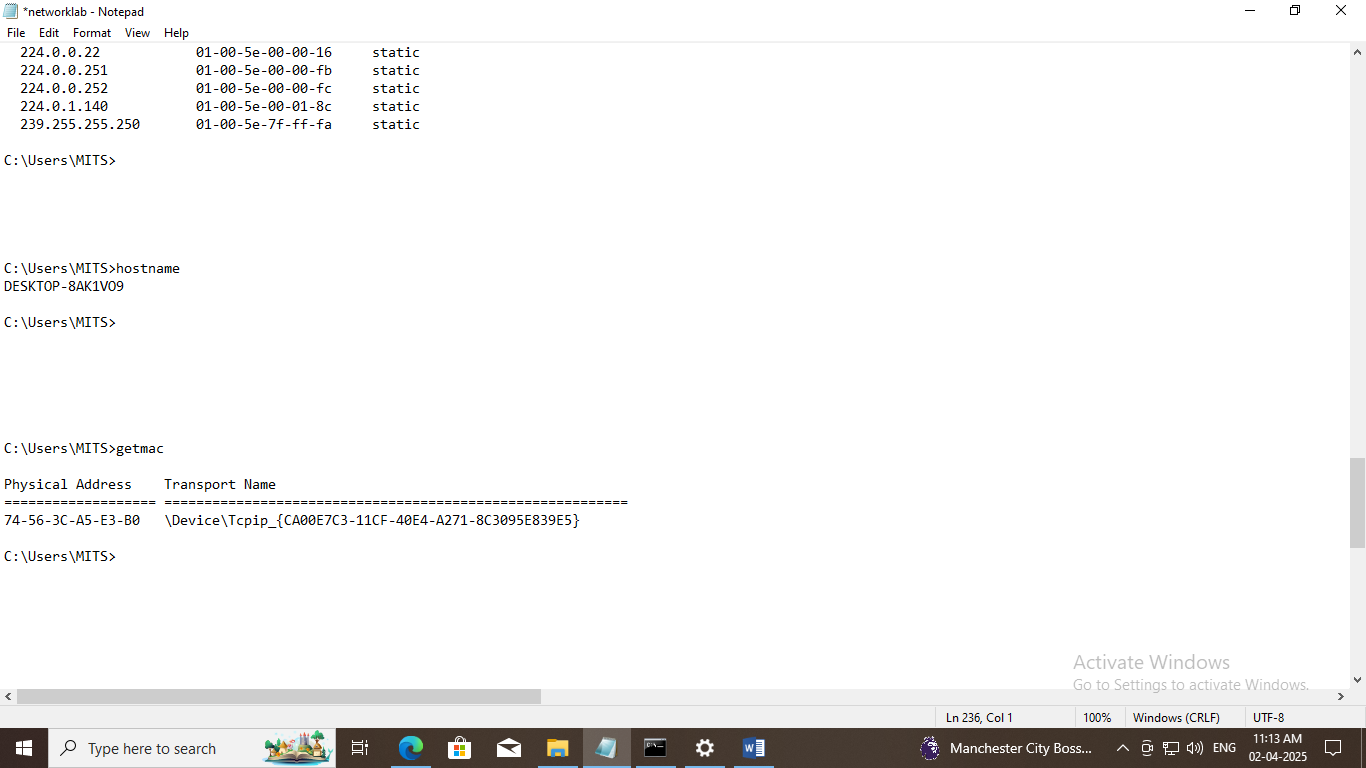
**7.nbtstat**

As I am sure you probably know, computers that are running a Windows operating system are assigned a computer name. Oftentimes, there is a domain name or a workgroup name that is also assigned to the computer. The computer name is sometimes referred to as the NetBIOS name. Windows uses several different methods to map NetBIOS names to IP addresses, such as broadcast, LMHost lookup, or even using the nearly extinct method of querying a WINS server. Of course, NetBIOS over TCP/IP can occasionally break down. The NbtStat command can help you to diagnose and correct such problems. The NbtStat -n command for example, shows the NetBIOS names that are in use by a device. The NbtStat -r command shows how many NetBIOS names the device has been able to resolve recently.



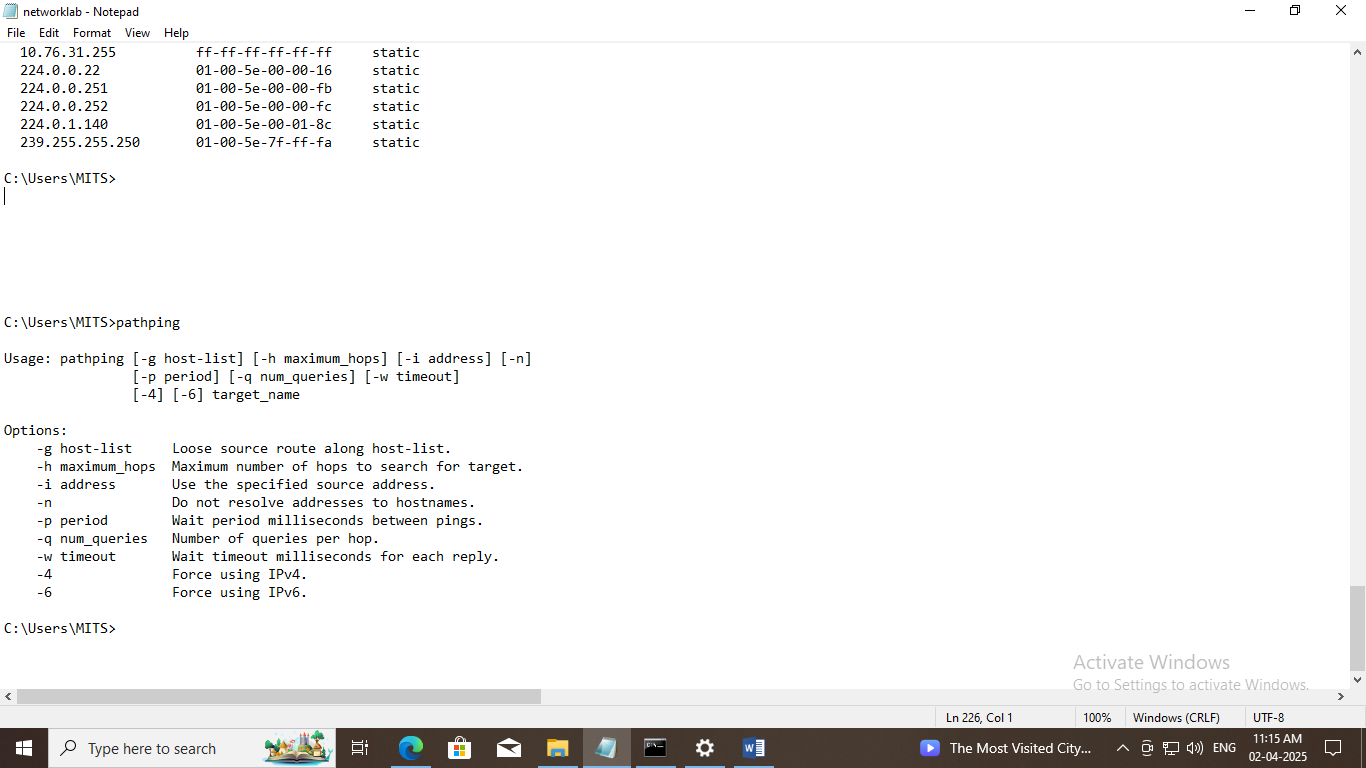
**8.hostname**

The previously discussed NbtStat command can provide you with the host name that has been assigned to a Windows device, if you know which switch to use with the command. However, if you’re just looking for a fast and easy way of verifying a computer’s name, then try using the Hostname command. Typing Hostname at the command prompt returns the local computer name.



**9. pathping**

Earlier, I talked about the Ping utility and the Tracert utility, and the similarities between them. As you might have guessed, the PathPing tool is a utility that combines the best aspects of Tracert and Ping. Entering the PathPing command.



**10.getmac**

Another very simple command that shows the MAC address of your network interfaces.

