**P.A. AZIZ COLLEGE OF ENGINEERING AND TECHNOLOGY (PAACET)**

Green Hills, Karakulam P.O., Thiruvananthapuram, Kerala 695564



**LABORATORY MANUAL**

**NETWORKING AND SYSYTEM ADMINISTRATION LAB**

**SUBJECT CODE: 20MCA136**

**MASTER OF COMPUTER APPLICATION DEPARTMENT**

**MCA 2nd SEMESTER**

**NAME: ACHSAH MARIYAM GEORGE**

**REGISTER NO: PAA21MCA-2001**

**YEAR: ACADEMIC YEAR 2021-’23**

**P.A. AZIZ COLLEGE OF ENGINEERING AND TECHNOLOGY**

GREEN HILLS, KARAKULAM P.O., THIRUVANANTHAPURAM, KERALA 695564



**CERTIFICATE**

***This is to certified that Mrs. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Register No. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) has satisfactorily completed the course in \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_as by the APJ Abdul Kalam University for \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ year, of semester \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ of Master of Computer Applications in the Academic year \_\_\_\_\_\_\_\_\_\_\_\_\_\_.***

**Head of the Department (MCA) Faculty-in-charge**

**Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Internal Examiner External Examiner**

**CONTENT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl. No.** | **Date** | **Name of Experiment** | **Page No.** | **Remarks** |
| 1 | 23.03.2022 | Introduction To Computer Hardware | 4 |  |
| 2 | 23.03.2022 | Computer Hardware Specifications | 7 |  |
| 3 | 30.03.2022 | Linux Installation Using Virtual Machine On Windows 11 | 8 |  |
| 4 | 30.03.2022 | Installation Of Windows Operating System On A Bare Machine | 10 |  |
| 5 | 06.04.2022 | Basic Linux Commands | 12 |  |
| 6 | 06.04.2022 | Familiarisation With Vi Editor | 17 |  |
| 7 | 13.04.2022 | Linux File System | 19 |  |
| 8 | 13.04.2022 | Familiarisation To Linux Shell And Shell Scripting | 22 |  |
| 9 | 20.04.2022 | Linux Shell Scripting Problems | 25 |  |
| 10 | 27.04.2022 | Linux Command | 38 |  |
| 11 | 04.05.2022 | Network Command | 39 |  |
| 12 | 11.05.2022 | Installation Of Lamp On Ubuntu | 41 |  |
| 13 | 11.05.2022 | Installation Of Laravel | 43 |  |
| 14 | 18.05.2022 | Familiarisation To Command Line Tool For Networking | 44 |  |
| 15 | 18.05.2022 | Familiarisation Of Static And Dynamic Ip | 47 |  |
| 16 | 25.05.2022 | Concept Of Subnets And CIDR Address Scheme | 48 |  |
| 17 | 25.05.2022 | Concept Of Subnet Mask | 50 |  |
| 18 | 01.05.2022 | Setting Up Of Firewall On Lan | 51 |  |
| 19 | 01.06.2022 | Wireshark And TCPDUMP | 52 |  |
| 20 | 08.06.2022 | Analyse Packets Using Wireshark | 54 |  |
| 21 | 08.06.2022 | Familiarisation To Hypervisors And Virtual Machines | 55 |  |
| 22 | 15.06.2022 | Installing Software From Source Code | 56 |  |

**EXPERIMENT - 1**

**INTRODUCTION TO COMPUTER HARDWARES**

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

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

**IMPORTANT HARDWARE COMPONENTS**

**MOTHERBOARD**

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

A motherboard usually contains significant sub-systems, such as the central processor, the chipset's input/output and memory controllers, interface connectors, and other components integrated for general use. Motherboard means specifically a PCB with expansion capabilities. As the name suggests, this board is often referred to as the "mother" of all components attached to it.

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

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

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

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

**RANDOM ACCESS MEMORY (RAM)**

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

**DAUGHTER CARD**

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

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

**BUS SLOTS**

Bus slot or expansion port, an expansion slot is a connection or port inside a computer or motherboard or riser cards . It provides an installation point for a hardware expansion card to be connected. For example, if you wanted to install a new video card in the computer, you'd purchase a video expansion card and install that card into the compatible expansion slot.

Commonly founded are

• AMR - Modem, sound card.

• CNR - Modem, network card, sound card.

• EISA - SCSI, network card, video card.

• ISA - Network card, sound card, video card.

• PCI - Network card, SCSI, sound card, video card.

• PCI Express - Video card, modem, sound card, network card.

• VESA - Video card.

**SMPS**

The working of SMPS is simply understood by knowing that the transistor used in LPS is used to control the voltage drop while the transistor in SMPS is used as a controlled switch. The disadvantages of LPS such as lower efficiency, the need for large value of capacitors to reduce ripples and heavy and costly transformers etc. are overcome by the implementation of Switched Mode Power Supplies.

**READ-ONLY MEMORY (ROM)**

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

**HARD DRIVE**

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

**GRAPHICS PROCESSING UNIT (GPU)**

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

**INTERFACING PORTS**

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

• Internal port − It connects the motherboard to internal devices like hard disk drive, CD drive, internal modem, etc.

• External port − It connects the motherboard to external devices like modem, mouse, printer, flash drives, etc.

Some of them are given below

**SERIAL PORT**

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

**PARALLEL PORT**

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

**USB PORT**

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

**PS/2 PORTS**

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

**VGA PORT**

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

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

**HDMI PORT**

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

**EXPERIMENT - 2**

**COMPUTER HARDWARE SPECIFICATIONS**

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

• Processor speed, model and manufacturer. Processor speed is typically indicated in gigahertz (GHz). The higher the number, the faster the computer.

• Random Access Memory (RAM), This is typically indicated in gigabytes (GB). The more RAM in a computer the more it can do simultaneously.

• Hard disk (sometimes called ROM) space. This is typically indicated in gigabytes (GB) and refers generally to the amount of information (like documents, music and other data) your computer can hold.

• Other specifications might include network (ethernet or wi-fi) adapters or audio and video capabilities.

**EXPERIMENT-3**

**LINUX INSTALLATION USING VIRTUAL MACHINE ON WINDOWS 11**

**REQUIREMENTS**

• Good internet connection to download software and Linux ISO.

• Windows system with at least 12 GB of free space.

• Windows system with 4GB of RAM. (It can work with less RAM as well, but your system will start to lag while using Linux in the virtual machine.)

• Make sure to enable virtualization in the BIOS

**STEPS**

Step 1: Download and install VirtualBox

• Go to the website of Oracle VirtualBox and get the latest stable version from here: <https://www.virtualbox.org/wiki/Downloads>

• Just double-click on the downloaded .exe file and follow the instructions on the screen.

• It is like installing any regular software on Windows.

Step 2: Download the Linux ISO

• Next, you need to download the ISO file of the Linux distribution.

• You can get this image from the official website of the Linux distribution you are trying to use.

• I am using Ubuntu in this example, and you can download ISO images for Ubuntu from the link below:

Creating a VirtualBox VM:

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

• Open VirtualBox.

• Then, click on Machine > New…

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

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

• For RedHat, it should be at least 2048 MB (2 GB).

• For RedHat Server 20.04 LTS, it should be at least 512 MB.

• Once you’re done, click on Next >.

• Now, select Create a virtual hard disk now and click on Create . This serves as the hard disk of the virtual Linux system.

• It is where the virtual system will store its files.

• select Dynamically allocated and click on Next >.

• You can choose either the “Dynamically allocated” or the “Fixed size” option for creating the virtual hard disk.

• set the virtual hard disk size for the VM.

• It should be at least 20.0 GB.

• Once you’re done, click on Create

Starting the VM:

Now, select the VM and click on Start.

The VM should start and boot from the RedHat ISO image.

• From here, you can do a Normal installation or Minimal installation.

• Normal installation comes with all the apps as usual.

• Minimal installation comes with a limited number of apps. It saves a lot of disk spaces.

• If you have internet connection on your computer, you can check Download updates while installing Ubuntu to download all the necessary updates while installing Ubuntu on your computer.

Once you’re done, click on Continue

• As this is a VM, I won’t go through the trouble of manually partitioning the hard drive.

• Just select Erase disk and install Ubuntu and click on Install Now.

• The Ubuntu installer will automatically create all the necessary partitions in your virtual hard drive.

Try to choose a password that you can remember.

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

After creating an account, you can start using Ubuntu.

**EXPERIMENT - 4**

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

**INTRODUCTION**

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

**Requirements :**

1. PC/Machine with at least 4GB RAM

2. An 8GB DVD or USB Flash drive.

3. Rufus software (<https://rufus.ie/en_US/>).

4. Required Windows 10 OS/OEM.

5. Strong stable internet connection.

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

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

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

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

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

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

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

**STEP 2: PREPARING BOOTBALE MEDIUM**

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

**Preparing Bootable DVD:**

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

**Preparing Bootable Pen Drive:**

1. Download the Rufus software from the official website.

(https://rufus.ie/en\_US/)&(https://github.com/pbatard/rufus/r eleases/download/v3.14/rufus-3.14.exe)

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

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

**STEP 3 : SYSTEM BOOTING & INSTALLATION**

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

1. Insert the bootable DVD or flash drive into the computer and restart it.

2. Usually the BIOS will automatically boot using the bootable medium, otherwise manually bot it up on the boot menu.

3. Choose the required language and layout settings in the initial installation window and press next and install button.

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

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

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

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

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

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

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

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

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

**EXPERIMENT – 5**

**BASIC LINUX COMMANDS**

1. **pwd**

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

1. **cd**

To navigate through the Linux files and directories, use the cd . It requires either the full path or the name of the directory, depending on the current working directory that you’re in. Let’s say you’re in /home/username/Documents and you want to go to Photos, a subdirectory of Documents. To do so, simply type the following command: cd Photos. Another scenario is if you want to switch to a completely new directory, for example,/home/username/Movies. In this case, you have to type cd followed by the directory’s absolute path: cd /home/username/Movies.

There are some shortcuts to help you navigate quickly:

● cd .. (with two dots) to move one directory up

● cd to go straight to the home folder

● cd- (with a hyphen) to move to your previous directory.

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

1. **ls**

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

If you want to see the content of other directories, type ls and then the directory’s path. For example, enter ls /home/username/Documents to view the content of Documents. There are variations you can use with the ls command:

● ls -R will list all the files in the sub-directories as well

● ls -a will show the hidden files

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

● ls -t lists files sorted in the order of “last modified”

● -r option will reverse the natural sorting order. Usually used in combination with other switches such as ls -tr. This will reverse the time-wise listing.

1. **cat**

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

● cat > filename creates a new file

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

1. **cp**

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

● cp -i will ask for user’s consent in case of a potential file overwrite.

● cp -p will preserve source files’ mode, ownership and timestamp.

● cp -r will copy directories recursively.

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

1. **mv**

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

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

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

1. **mkdir**

Use mkdir command to make a new directory — if you type mkdir Music it will create a directory called Music.

There are extra mkdir commands as well:

● To generate a new directory inside another directory, use this Linux basic command mkdir Music/Newfile

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

1. **rmdir**

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

1. **rm**

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

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

1. **touch**

The touch command allows you to create a blank new file through the Linux command line. As an example, enter touch /home/username/Documents/Web.html to create an HTML file entitled Web under the Documents directory.

1. **locate**

You can use this command to locate a file, just like the search command in Windows. What’s more, using the -i argument along with this command will make it case-insensitive, so you can search for a file even if you don’t remember its exact name. To search for a file that contains two or more words, use an asterisk (\*). For example, locate -i school\*note command will search for any file that contains the word “school” and “note”, whether it is uppercase or lowercase.

1. **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

1. **grep**

Another basic Linux command that is undoubtedly helpful for everyday use is grep. It lets you search through all the text in a given file. To illustrate, grep blue notepad.txt will search for the word blue in the notepad file. Lines that contain the searched word will be displayed fully. You should refer to some grep tutorial Useful for command line use as well. Usually output of a previous command is piped into the grep command. For example ls -l | grep “kernel”

1. **sudo**

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

1. **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.

1. **du**

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

1. **head**

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

1. **tail**

This one has a similar function to the head command, but instead of showing the first lines, NETWORKING & SYSTEM ADMINISTRATION LAB 66 the tail command will display the last ten lines of a text file. For example, tail -n filename.ext.

1. **diff**

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

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

1. **tar**

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

1. **chmod**

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

1. **chown**

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

1. **ps**

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

1. **Kill**

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

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

● SIGTERM (15) — requests a program to stop running and gives it some time to save all of its progress. If you don’t specify the signal when entering the kill command, this signal will be used.

● SIGKILL (9) — forces programs to stop immediately. Unsaved progress will be lost. Besides knowing the signals, you also need to know the process identification number (PID) of the program you want to kill. If you don’t know the PID, simply run the command ps ux.

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

kill [signal option] PID.

You can issue kill -9 PID

1. **ping**

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

1. **wget**

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

1. **uname**

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

1. **top**

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

1. **history**

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

1. **man**

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

1. **echo**

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

1. **zip, unzip**

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

1. **hostname**

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

1. **useradd, userdel**

( This is available only to system admins) Since Linux is a multi-user system, this means more than one person can interact with the same system at the same time. useradd is used to create a new user, while passwd is adding a password to that user’s account. To add a new person named John type, useradd John and then to add his password type, passwd 123456789.

**EXPERIMENT – 6**

**FAMILIARISATION WITH VI EDITOR**

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

1) It is available in almost all Linux Distributions

2) It works the same across different platforms and Distributions

3) It is user-friendly. Hence, millions of Linux use it for their editing needs

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

1. **VI OPERATIONAL MODES**

● Vi Command Mode:

❏ The vi editor opens in this mode, and it only understands commands.

❏ In this mode, you can perform administrative tasks such as move the cursor and cut, copy, paste the text.

❏ This mode also saves the changes you have made to the file.

❏ Commands are case sensitive. The right letter case should be used.

● Vi Insert Mode:

❏ This mode is for inserting text in the file.

❏ You can switch to the Insert mode from the command mode by pressing 'i' on the keyboard

❏ Once you are in Insert mode, any key would be taken as an input for the file on which you are currently working.

❏ To return to the command mode and save the changes you have made you need to press the Esc key

1. **STARTING VI**

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

a. If the file does not exist, a screen will appear with just a cursor at the top followed by tildes (~) in the first column.

b. If the file does exist, the first few lines of the file will appear.

c. The status line at the bottom of the screen shows error messages and provides information and feedback, including the name of the file.

1. **Vi EDITING COMMANDS**

• i - Insert at cursor (goes into insert mode)

• a - Write after cursor (goes into insert mode)

• A - Write at the end of line (goes into insert mode)

• ESC - Terminate insert mode

• u - Undo last change

• U - Undo all changes to the entire line

• o - Open a new line (goes into insert mode)

• dd - Delete line

• 3dd - Delete 3 lines.

• D - Delete contents of line after the cursor

• C - Delete contents of a line after the cursor and insert new text. Press ESC key to end insertion.

• dw - Delete word

• 4dw - Delete 4 words

• cw - Change word

• x - Delete character at the cursor

• r - Replace character

• R - Overwrite characters from cursor onward

• s - Substitute one character under cursor continue to insert

• S - Substitute entire line and begin to insert at the beginning of the line

• ~ - Change case of individual character

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

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

**Moving within a file :**

★ k - Move cursor up

★ j - Move cursor down

★ h - Move cursor left

★ l - Move cursor right

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

**Saving and Closing the file**

★ Shift+zz - Save the file and quit

★ :w - Save the file but keep it open

★ :q - Quit without saving

★ :wq - Save the file and quit

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

**EXPERIMENT – 7**

**LINUX FILE SYSTEM**

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

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

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

The Linux file system contains the following sections:

1. The root directory (/)

2. A specific data storage format (EXT3, EXT4, BTRFS, XFS)

3. A partition or logical volume having a particular file system.

Some key features of linux file system are as following:

• Partition, Directories, and Drives

• Case Sensitivity

• File Extensions

• Hidden files

1) Ext, Ext2, Ext3 and Ext4 file system

• The file system Ext stands for Extended File System

• Ext file system is an older version, and is no longer used due to some limitations.

• Ext2 is the first Linux file system that allows managing two terabytes of data

• Ext3 is developed through Ext2; it is an upgraded version of Ext2 and contains backward compatibility.

• Ext4 file system is the faster file system among all the Ext file systems.

2) JFS File System

• JFS stands for Journaled File System

• It is an alternative to the Ext file system.

• It can also be used in place of Ext4, where stability is needed with few resources

3) ReiserFS File System

• ReiserFS is an alternative to the Ext3 file system

• It has improved performance and advanced features.

4) XFS File System

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

5) Btrfs File System

• Btrfs stands for the B tree file system

• It is used for fault tolerance, repair system, fun administration, extensive storage configuration, and more

6) Swap File System

• The swap file system is used for memory paging in Linux operating system during the system hibernation

• A system that never goes in hibernate state is required to have swap space equal to its RAM size.

**LINUX FILE SYSTEM HIERARCHY**

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

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

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

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

ROOT DIRECTORY

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

● Every single file and directory starts from the root directory

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

● /root is the root user’s home directory, which is not the same as

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

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

● common commands that are used by all the users, like cat, cp, cd, ls, etc.

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

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

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

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

● /lib/modules: The '/lib/modules' stores kernel modules and has a directory for each installed kernel.Modules are meant to use extra hardware support without making a new kernel.

● /lib32 and /lib64: During compilation time of libraries you'll encounter through the directories named '/lib32' and '/lib64' which will clarify register size to be used. A 64-bit system may have compatibility for 32-bit binary.

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

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

Its main purpose is to store optional application software packages.

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

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

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

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

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

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

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

/home-The '/home' directory stores users personnel files. After the '/home' there is a directory which is generally named at the user's name like we have '/home/username’. Inside this directory we have our sub-directories like Desktop, Downloads, Documents, pictures, etc

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

• '/root' directory is different from (/) root. /srv-The term 'srv' is short for service.

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

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

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

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

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

FILE PERMISSIONS:

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

1.User

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

2.Group

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

3.Other

• Any other user who has access to a file. This person has neither created the file, nor he belongs to a usergroup who could own the file.

• Every file and directory in your UNIX/Linux system has following 3 permissions defined for all the 3 owners

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

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

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

• chmod Linux command, used to change the read, write, and execute permissions of files and directories

• chmod +rwx filename to add permissions.

• chmod -rwx directory name to remove permissions.

• chmod +x filename to allow executable permissions.

• chmod -wx filename to take out write and executable permissions.

**EXPERIMENT – 8**

**FAMILIARISATION TO LINUX SHELL AND SHELL SCRIPTING**

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

**Kernel**

• The kernel is a computer program that is the core of a computer’s operating system, with complete control over everything in the system.

• It manages following resources of the Linux system –

\* File management

\* Process management

\* I/O management

\* Memory management

\* Device management etc.

**Shell**

• A shell is a special user program which provides an interface to users to use operating system services.

• Shell accept human readable commands from the user and convert them into something which the kernel can understand.

• The shell gets started when the user logs in or starts the terminal.

• Shell is broadly classified into two categories:

\* Command Line Shell

\* Graphical shell

• The shell in the linux operation takes input in the form of commands, processes them and gives an output . • When you are in the terminal the shell issues a command prompt.

• It is usually $ where you can type the input which is then executed by hitting Enter key.

Command Line Shell

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

Graphical Shells

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

**BASH (Bourne Again Shell)** :

• It is the most widely used shell in Linux systems.

• It is used as the default login shell in Linux systems

• It can also be installed on Windows OS.

**Shell Scripting**

• Shell script is a series of command(s) stored in a plain text file.

• These files are called Shell Scripts or Shell Programs.

• Each shell script is saved with .sh file extension.

• eg: myscript.sh

**Creating a shell script**

1. Creating a file using any text editor

2. Start the script with #/bin/sh

3. Write some code

4. Save the script file as filename.sh

5. For executing the script type bash filename.sh

**Variables in Shell**

• Variables store data in the form of character and number.

• In Linux (Shell), there are two types of variable:

• System variables

- Created and maintained by Linux itself.

- This type of variable defined in CAPITAL LETTERS.

• User defined variables (UDV)

- Created and maintained by the user.

- This type of variable is defined in lower letters.

**The read Statement**

• Used to get input (data from user) from keyboard and store (data) to variables.

• Syntax:

◦ read variable1, variable2,...variableN

**Conditional Statements**

The if...else statements

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

Unix Shell supports following forms of if…else statement –

• if...fi statement

• if...else...fi statement

• if...elif...else...fi statement

• if condition

-used for making decisions in shell script.

-If the condition is true

-Then command1 is executed.

Syntax:

if condition

then

command1 if condition is true or if exit status of

condition is 0

fi

• if...else...fi

-If given condition is true

-Then command1 is executed

-Otherwise command2 is executed.

Syntax:

if condition

then

condition is zero (true - 0)

execute all commands up to else statement

else

if condition is not true then

execute all commands up to fi

fi

**Loops in Shell Scripts**

• Bash supports:

1. for loop
2. while loop

Note that in each and every loop,

• First, the variable used in loop condition must be initialized, then execution of the loop begins.

• A test (condition) is made at the beginning of each iteration.

• The body of the loop ends with a statement that modifies the value of the test (condition) variable.

1. **for Loop**

Syntax:

for (( expr1; expr2; expr3 ))

do

repeat all statements between do and done until

expr2 is TRUE

done

1. **while Loop**

Syntax:

while [ condition ]

do

command1

command2

..

....

done

**Functions**

• Function is a series of instructions/commands.

• Function performs particular activity in shell.

• Code in a function is only executed when a function is ‘called’

Syntax:

function-name ( )

{

Function body

}

**EXPERIMENT – 9**

**LINUX SHELL SCRIPTING PROBLEMS**

**Problem 1**

**Aim:** Write a shell script to get current date, time, username and current working

**Algorithm:**

1. Read the logname
2. Read the current date
3. Read the user
4. Read the current directory
5. Save and quit the shell

**Program Code**

echo "HELLO,$LOGNAME"

echo "CURRENT DATE IS $(date)"

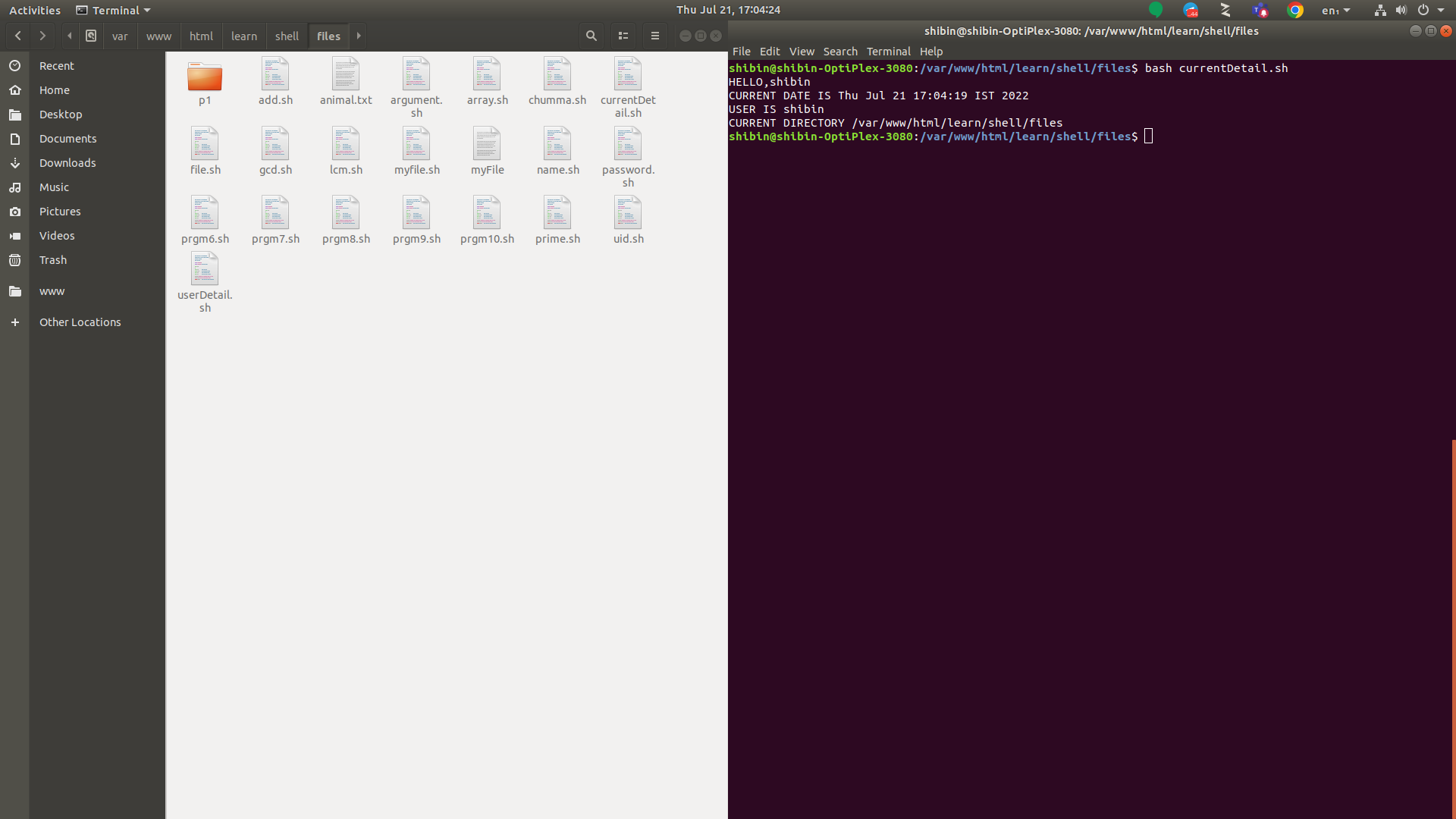
echo "USER IS $(whoami)"

echo "CURRENT DIRECTORY $(pwd)"

**Result:**

Thus the program has been executed successfully

**Output:**

****

**Problem 2**

**Aim:** Pass arguments to a script in Linux

**Algorithm:**

1. Create vi file named argument.sh
2. Read the first number, second number, third number, fourth number and fifth number
3. Save and quit the shell
4. Enter the number to be read in space between

**Program Code**

echo "First number : $1"

echo "Second number : $2"

echo "Third number : $3"

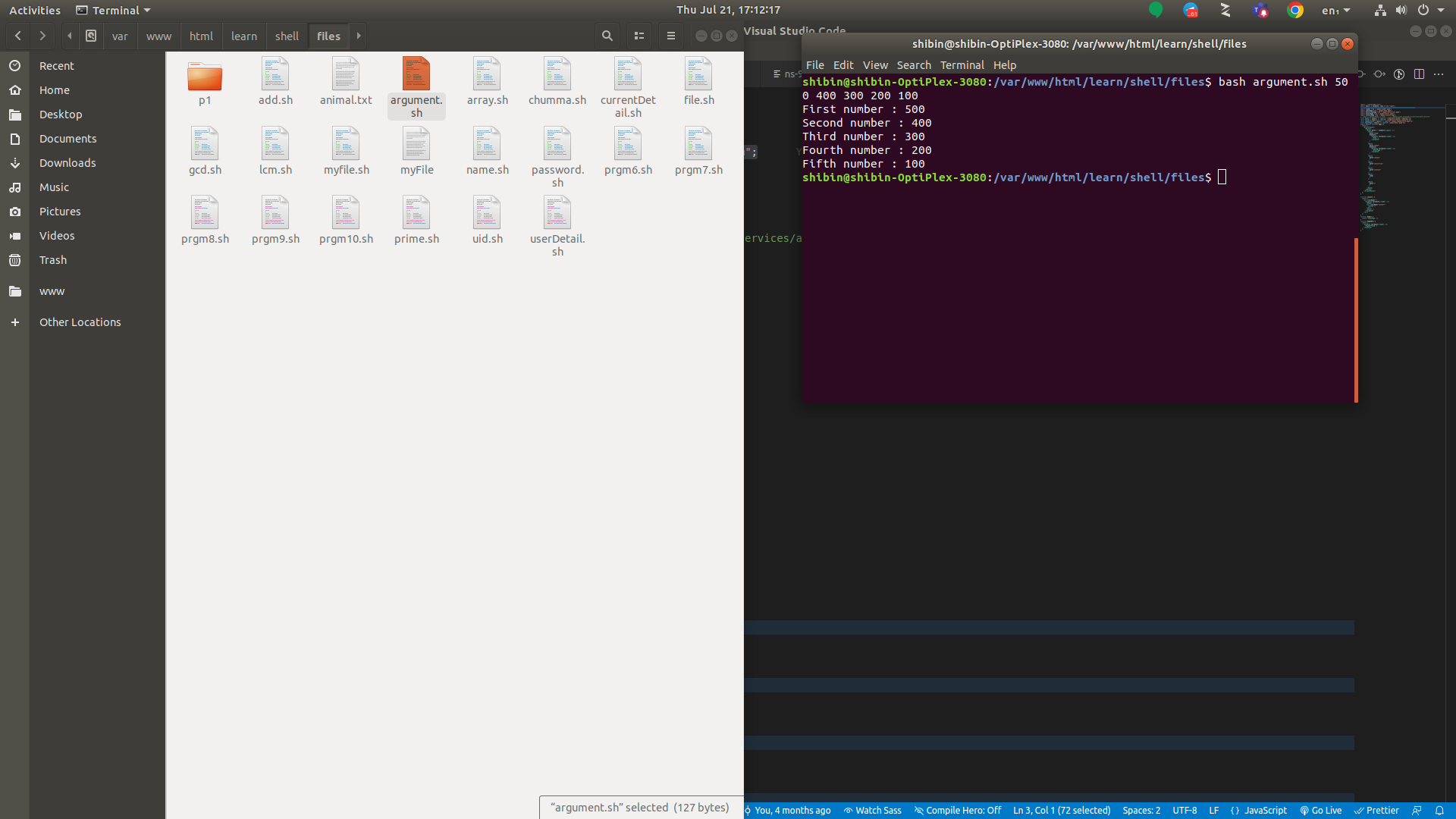
echo "Fourth number : $4"

echo "Fifth number : $5"

**Result:**

Thus the program has been executed successfully

**Output:**

****

**Problem 3**

**Aim:** Set an array in Linux

**Algorithm:**

1. Create vi file named array.sh
2. Create an array
3. Write the different method to call an array
4. Save and quit the shell

**Program Code**

array=(hello, I am Achsah)

echo ${array[@]}

echo ${array[\*]}

echo ${array[@]:0}

echo ${aaray[\*]:0}

echo ${array[0]}

echo ${array[1]}

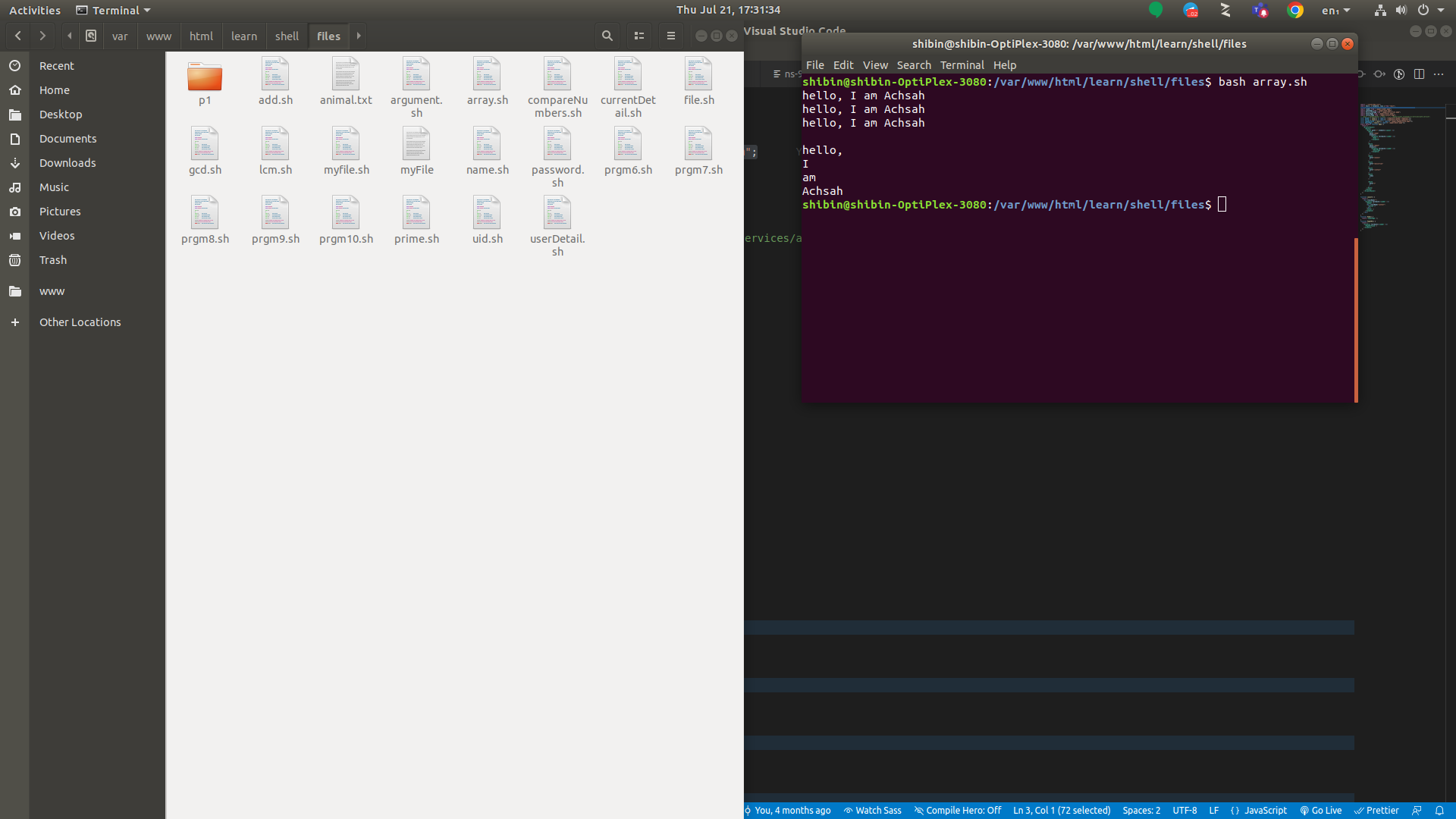
echo ${array[2]}

echo ${array[3]}

**Result:**

Thus the program has been executed successfully

**Output:**

****

**Problem 4**

**Aim:** Script to compare Numbers

**Algorithm:**

1. Create vi file named compareNumbers.sh
2. Declare two variables
3. If the value of var2 is greater than var1, print var2 is greater than var1
4. In second case, of the value of var1 is greater than 1000, print var1 is greater than 1000
5. Else print var1 is less than 1000
6. Save and quit the shell

**Program Code**

var1=100

var2=200

if [ $var2 -gt $var1 ]

then

    echo "$var2 is greater than $var1"

fi

# second comparison

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

if [ $var1 -gt 1000 ]

then

    echo "$var1 is greater than 1000"

else

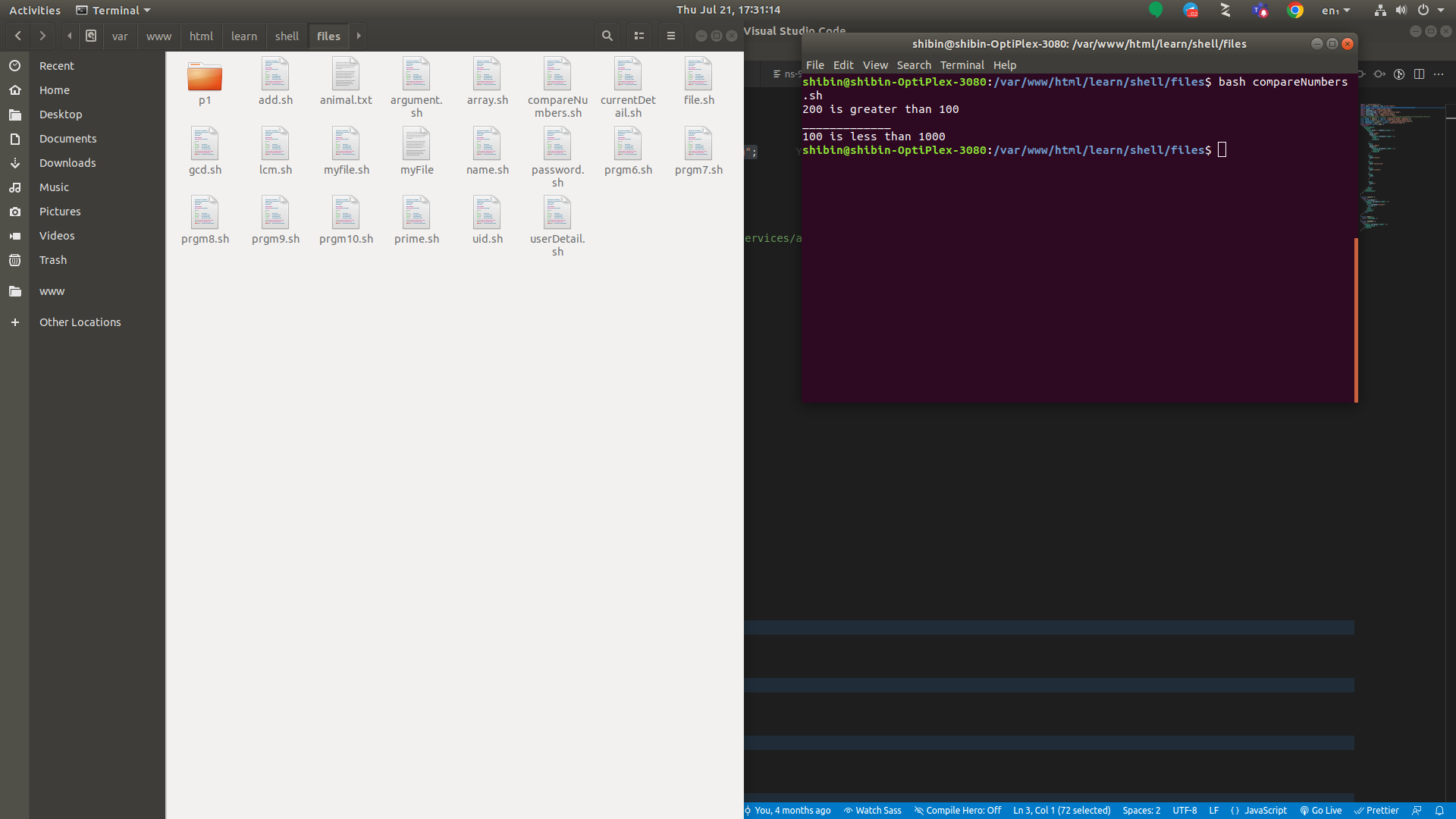
    echo "$var1 is less than 1000"

fi

**Result:**

Thus the program has been executed successfully

**Output:**



**Problem 5**

**Aim:** Print all array Indexes

**Algorithm:**

1. Create vi file named array.sh
2. Create an array
3. Read the array index and value
4. With for statement declare the value of array along with the array index
5. Save and quit the shell

**Program Code**

arr=(10 20 30 40 50)

echo "index : value"

for ((i=0; i<${#arr[@]}; i++))

do

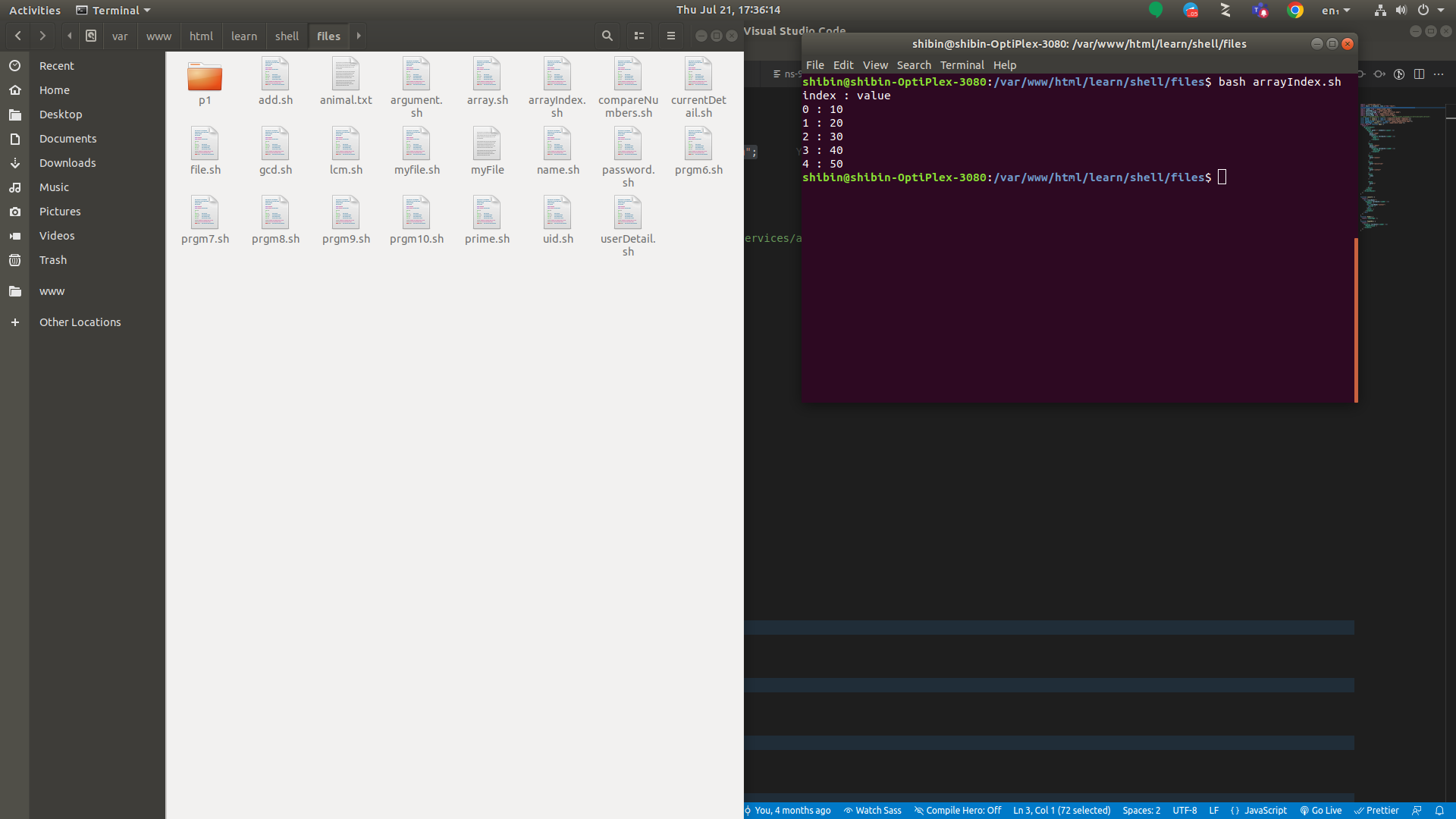
    echo "$i : ${arr[$i]}"

done

**Result:**

Thus the program has been executed successfully

**Output:**

****

**Problem 6**

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

**Algorithm:**

1. Create vi file named prgm6.sh
2. Read the file path
3. If the password enabled for the file, print $File passwords are enabled
4. If there permission to execute $File, print “You have permission to execute $File.
5. else, print “You have no permission to execute $File.
6. Save and quit the shell

**Program Code**

FILE="/var/www/html/learn/shell/files/file.sh"

if [ -e "$FILE" ]

then

echo "$FILEpasswords are enabled"

fi

if [ -x "$FILE" ]

then

echo "You have permission to execute $FILE"

else

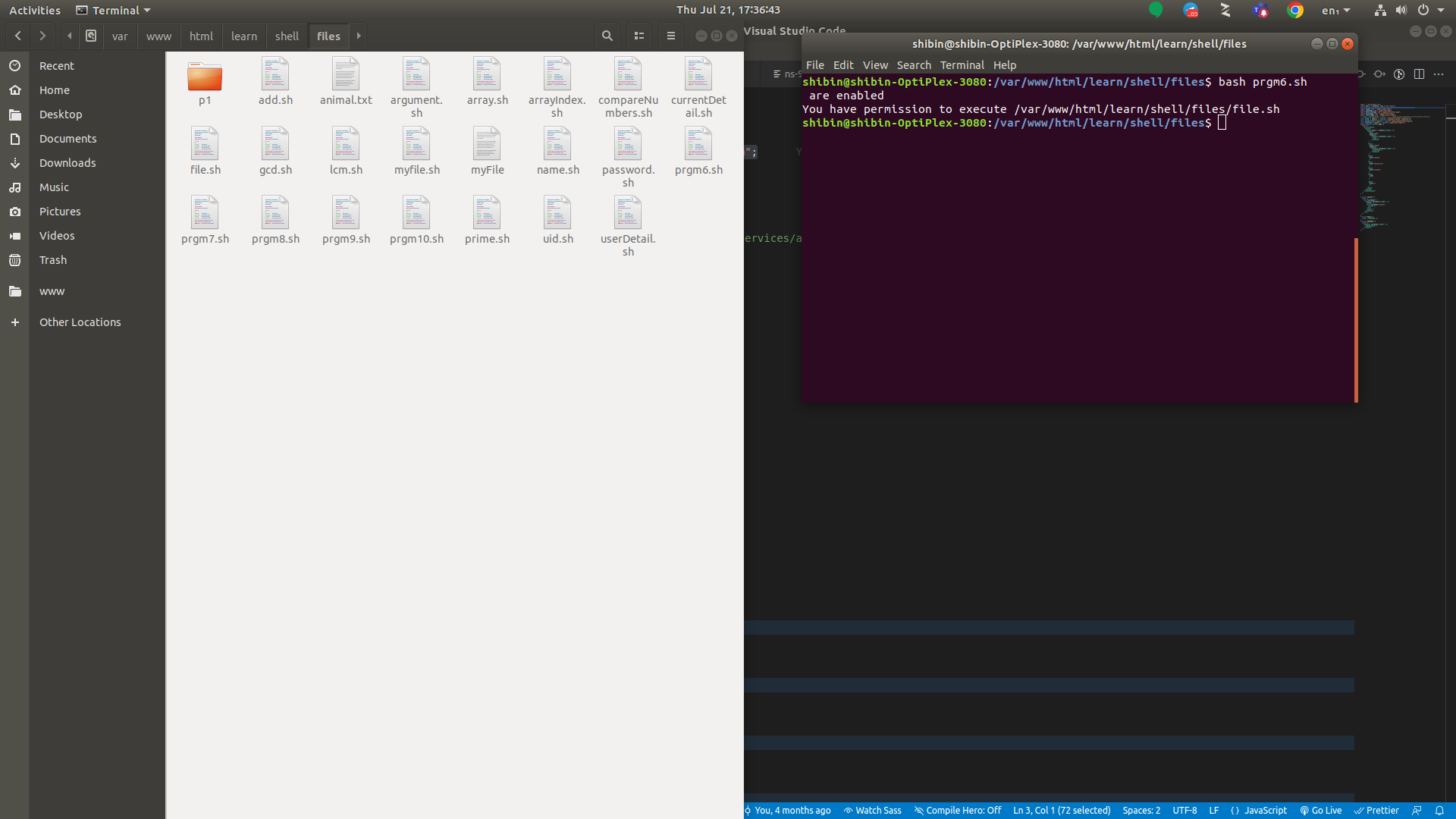
echo "You have no permission to execute $FILE"

fi

**Result:**

Thus the program has been executed successfully

**Output:**

****

**Problem 7**

**Aim:** To check if a directory exists

**Algorithm:**

1. Create vi file named prgm7.sh
2. If the given directory exists, then print $DirectoryName exists
3. Else print $DirectoryName doen not exists
4. Save and quit the shell

**Program Code**

if [ -d "/var/www/html/learn/shell/files/p1" ]

then

echo "/var/www/html/learn/shell/files/p1 exists."

else

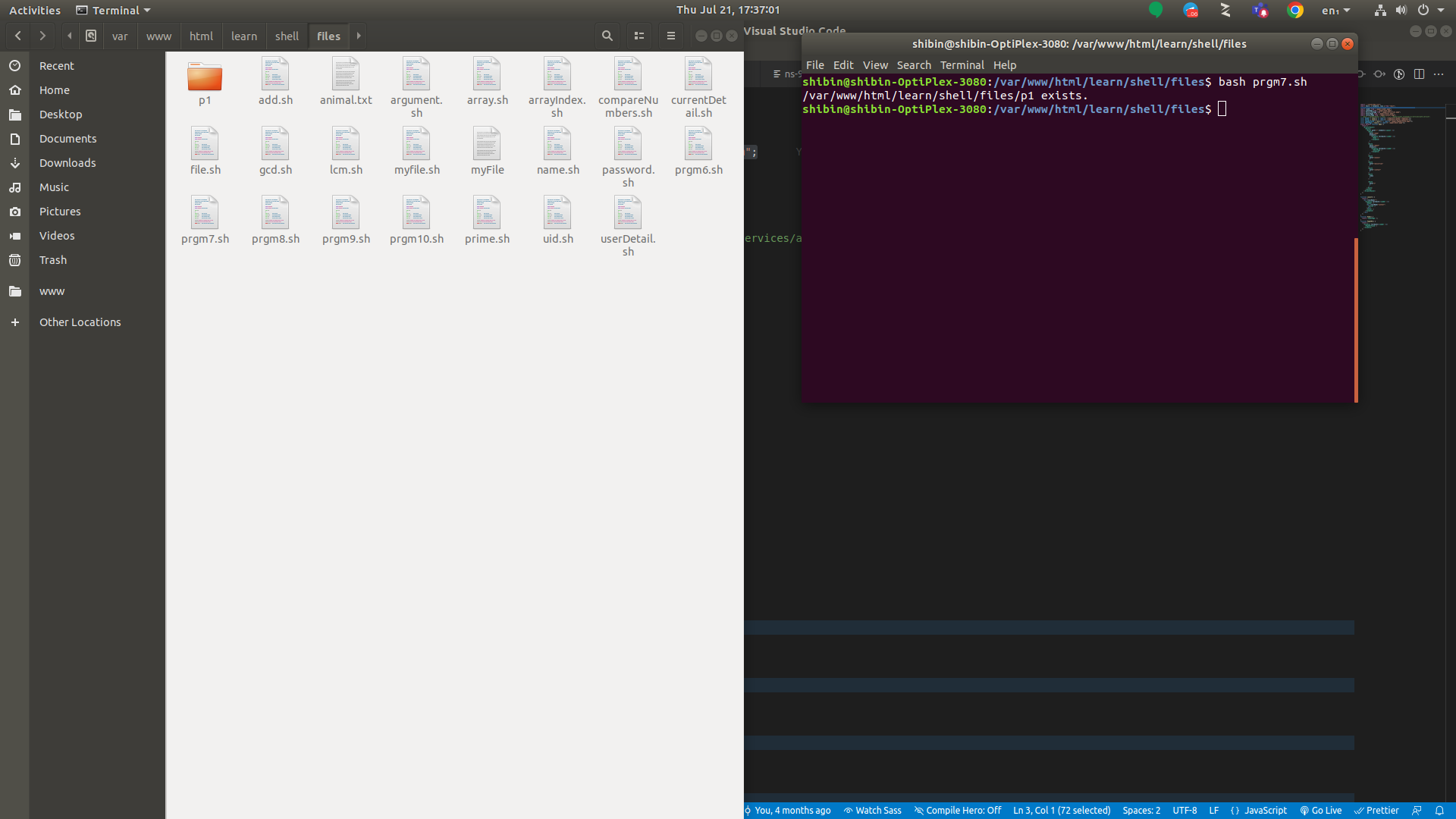
echo "Error: Directory /var/www/html/learn/shell/files/p1 does not exists."

Fi

**Result:**

Thus the program has been executed successfully

**Output:**

****

**Problem 8**

**Aim:** Write a shell script to print a number in reverse order. It should support

* 1. the following requirements.
  2. The script should accept the input from the command line.
  3. If you don’t input any data, then display an error message to execute the script correctly

**Algorithm:**

1. Create vi file named prgm8.sh
2. Read the number to be reversed.
3. Check the input
4. If input is 0, throw an error
5. Else processes the code using while loop
6. With for statement declare the value of array along with the array index
7. Save and quit the shell

**Program Code**

echo "Enter a number to be reversed :"

read n

if [ $n == 0 ]

then

echo "Please provide the correct input in the below format."

echo "Usage: $0 number"

echo " This script will reverse the given number."

echo " For eg. $0 1234, will print 4321"

exit 1

else

rev=0

sd=0

while [ $n -gt 0 ]

do

sd=`expr $n % 10`

rev=`expr $rev \\* 10 + $sd`

n=`expr $n / 10`

done

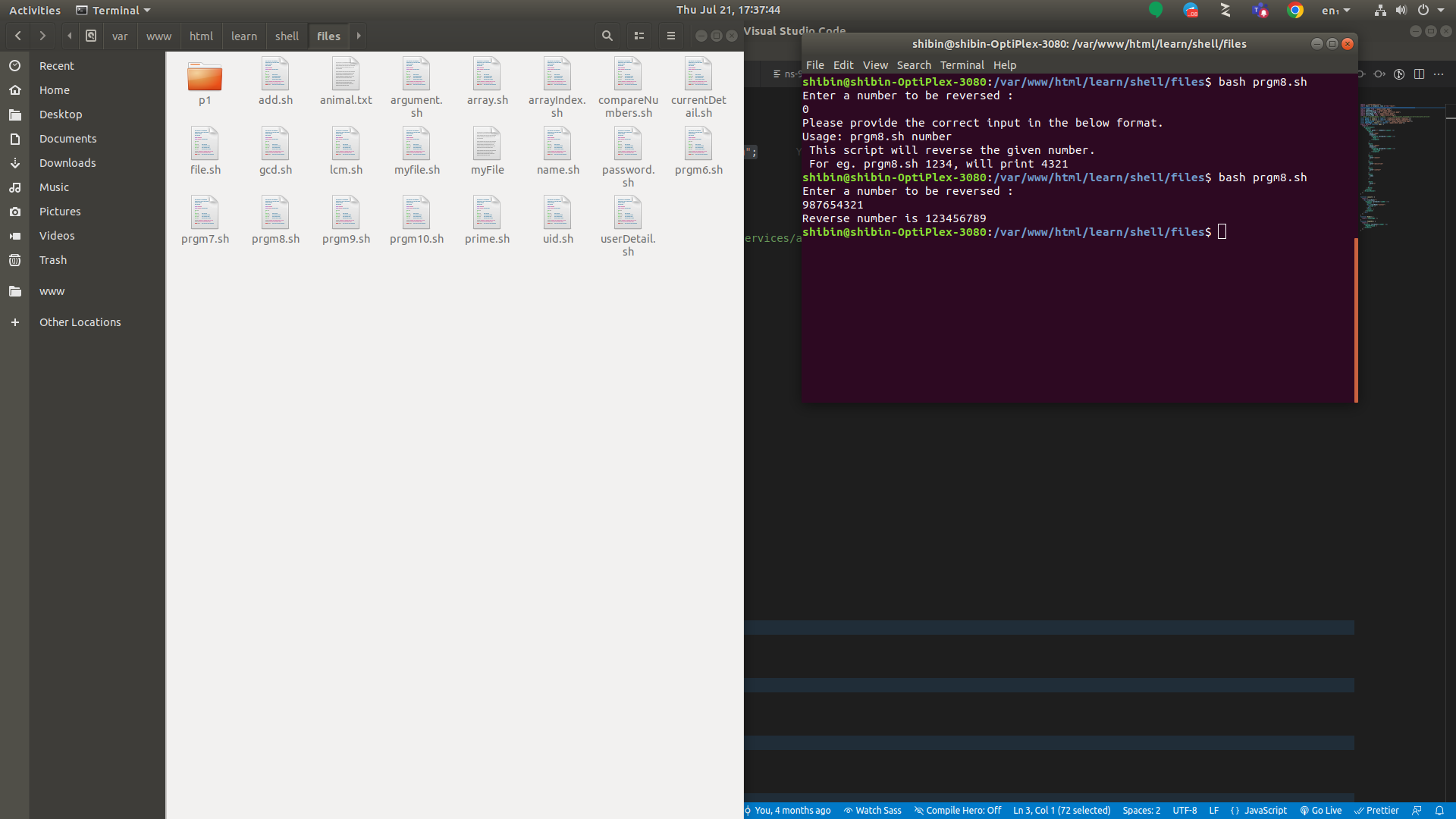
echo "Reverse number is $rev"

fi

**Result:**

Thus the program has been executed successfully

**Output:**

****

**Problem 9**

**Aim:** Write a shell script to print first ten elements of Fibonacci series.

**Algorithm:**

1. Create vi file named prgm9.sh
2. Read the number of elements.
3. Display the Fibonacci series using for loop.
4. Save and quit the shell.

**Program Code**

N=10

a=0

b=1

echo "The Fibonacci series is : "

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

do

echo -n " $a "

fn=$((a + b))

a=$b

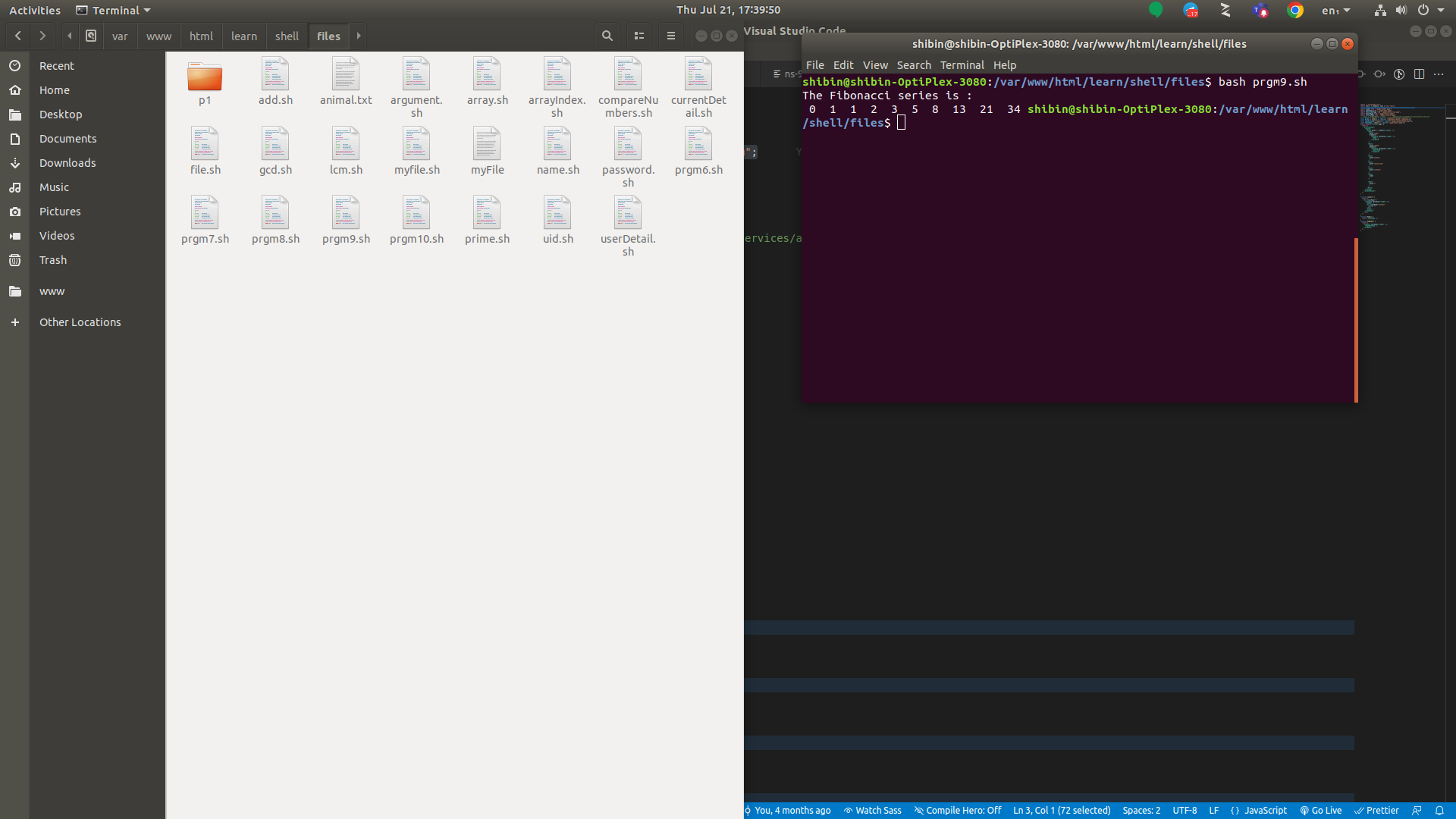
b=$fn

done

**Result:**

Thus the program has been executed successfully

**Output:**

****

**Problem 10**

**Aim:** Write a shell script to validate password strength there are a few assumptions for the password string.

1. Length – minimum of 8 characters. Contain both alphabet and number.
2. Include both the small and capital case letters.
3. If the password doesn’t comply with any of the above conditions, then the script should report it as a <Weak Password>.

**Algorithm:**

1. Create vi file named prgm10.sh
2. Read the password elements.
3. Check the password length.
4. If the password length does not contains eligibility then throw an error.
5. Save and quit the shell.

**Program Code**

echo "Enter your password"

read password

len="${#password}"

if test $len -ge 8 ; then

echo "$password" | grep -q [0-9]

if test $? -eq 0 ; then

echo "$password" | grep -q [A-Z]

if test $? -eq 0 ; then

echo "$password" | grep -q [a-z]

if test $? -eq 0 ; then

echo "Strong Password"

else

echo "Weak Password -> Should include a lower case letter."

fi

else

echo "Weak Password -> Should include a capital case letter."

fi

else

echo "Weak Password -> Should use numbers in your password."

fi

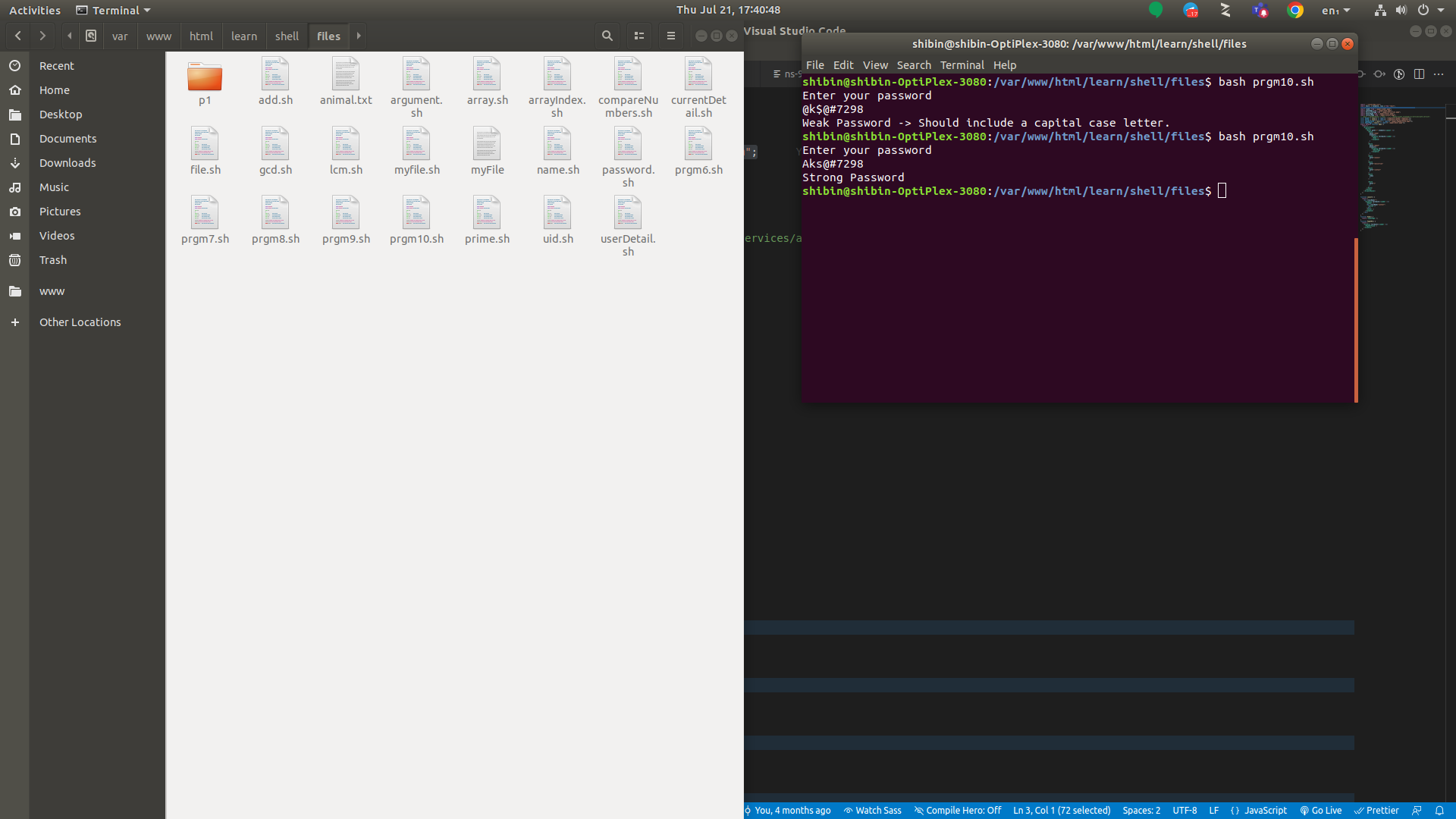
else

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

fi

**Result:** Thus the program has been executed successfully

**Output:**

****

**Problem 11**

**Aim:** Write a shell script to compute the GCD of two numbers.

**Algorithm:**

1. Create vi file named gcd.sh
2. Read the two number elements.
3. Calculate the GCD of the given two numbers
4. Display the output
5. Save and quit the shell.

**Program Code**

echo Enter two numbers with space in between

read a b

m=$a

if [ $b -lt $m ]

then

m=$b

fi

while [ $m -ne 0 ]

do

x=`expr $a % $m`

y=`expr $b % $m`

if [ $x -eq 0 -a $y -eq 0 ]

then

echo gcd of $a and $b is $m

break

fi

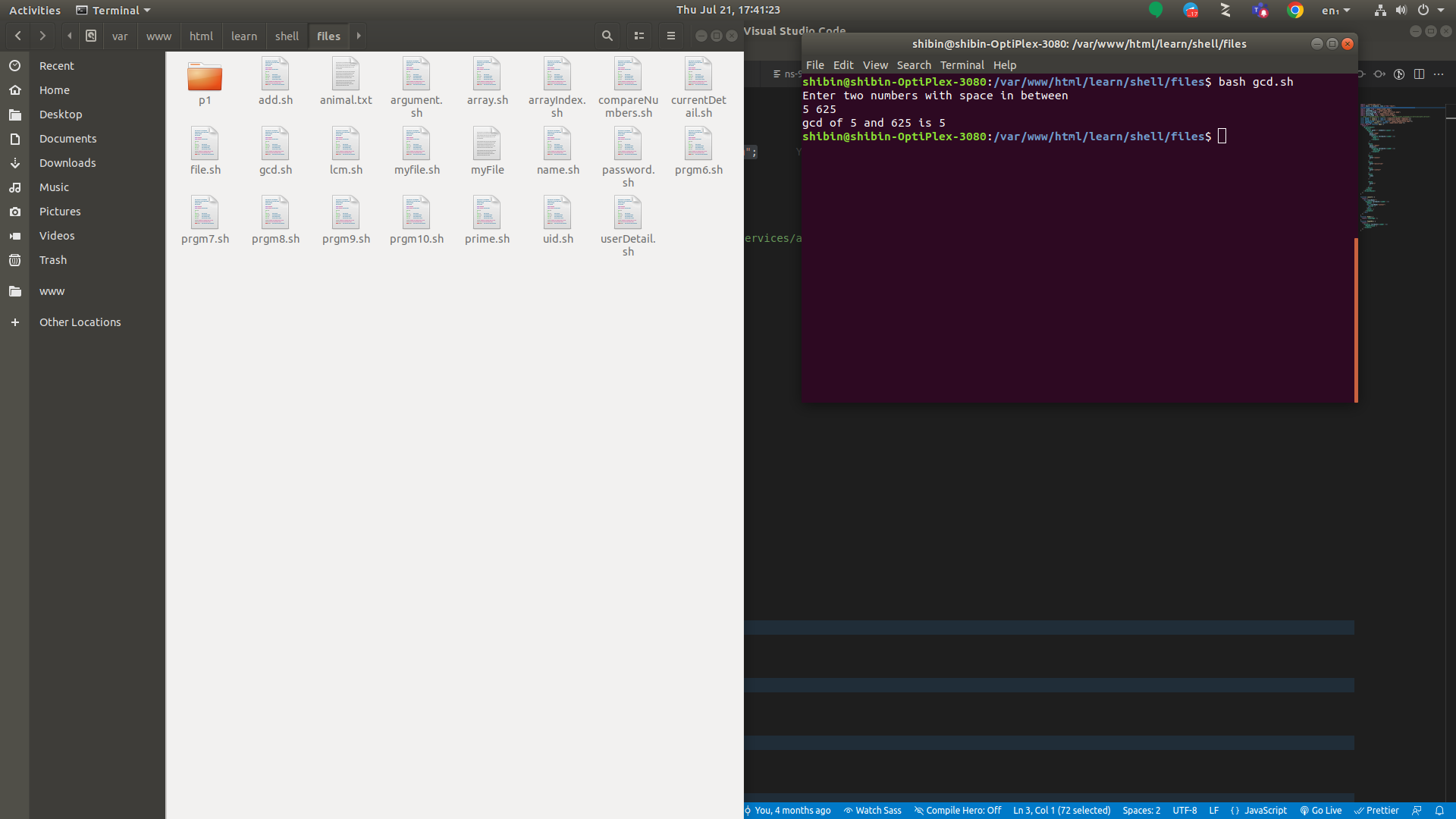
m=`expr $m - 1`

done

**Result:**

Thus the program has been executed successfully

**Output:**

****

**Problem 12**

**Aim:** Write a shell script to compute the LCM of two numbers.

**Algorithm:**

1. Create vi file named lcm.sh
2. Read the two number elements.
3. Calculate the LCM of the given two numbers
4. Display the output
5. Save and quit the shell.

**Program Code**

echo "Enter first no"

read a

echo "Enter 2nd no"

read b

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

while [ $b -ne 0 ]

do

    r=`expr $a % $b`

    a=$b

    b=$r

done

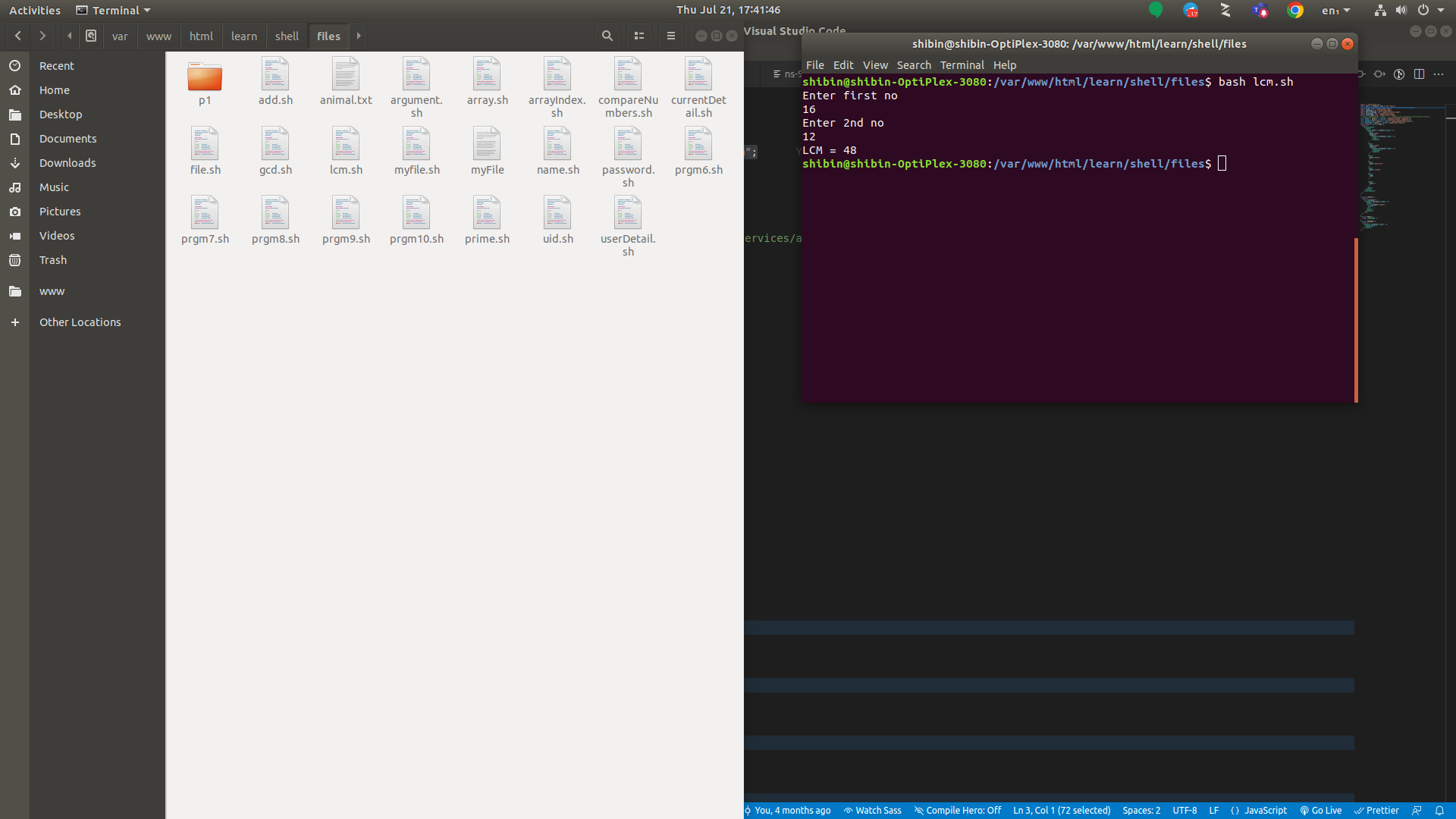
LCM=`expr $p / $a`

echo "LCM = $LCM"

**Result:**

Thus the program has been executed successfully

**Output:**

****

**Problem 13**

**Aim:** Write a shell script to find whether a given number is prime.

**Algorithm:**

1. Create vi file named lcm.sh
2. Read the two number elements.
3. Check the number whether the number prime.
4. Display the output.
5. Save and quit the shell.

**Program Code**

i=2

rem=1

echo "Enter a number"

read num

if [ $num -lt 2 ]

then

    echo -e "$num is not prime \n"

    exit 0

fi

while [ $i -le `expr $num / 2` -a $rem -ne 0 ]

do

    rem=`expr $num % $i`

    i=`expr $i + 1`

done

if [ $rem -ne 0 ]

then

    echo -e "$num is prime\n"

else

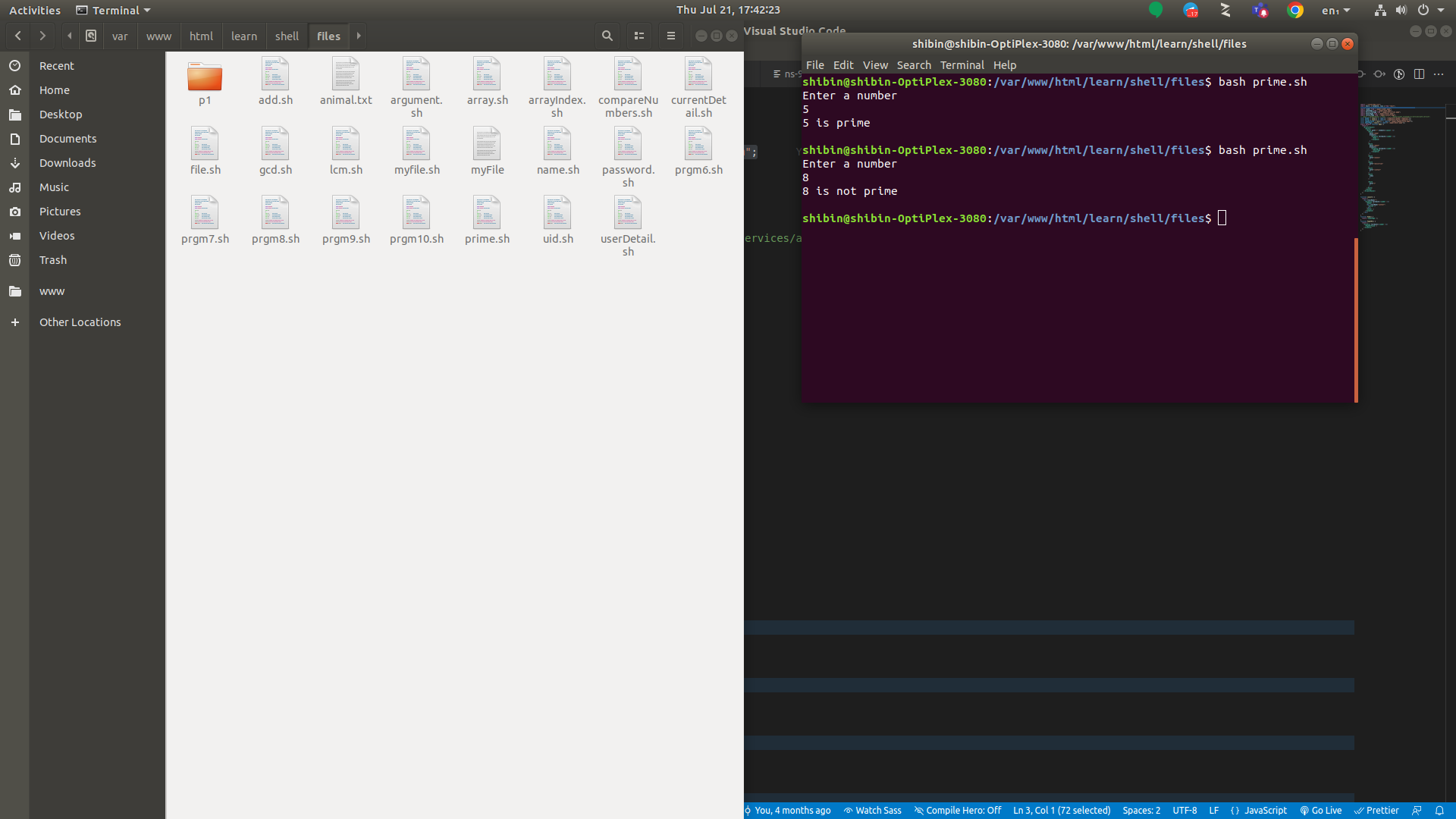
    echo -e "$num is not prime\n"

fi

**Result:**

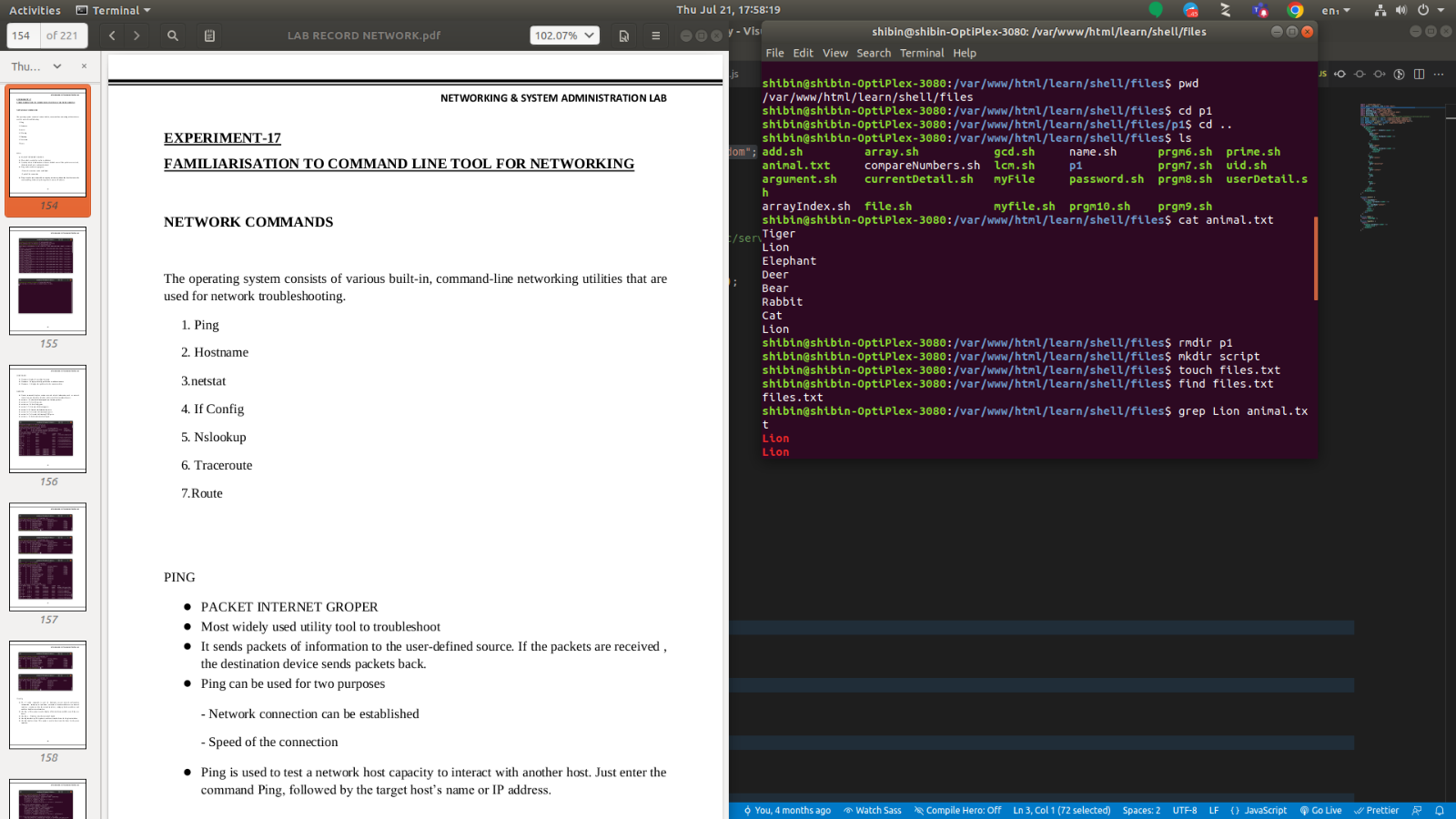
Thus the program has been executed successfully

**Output:**

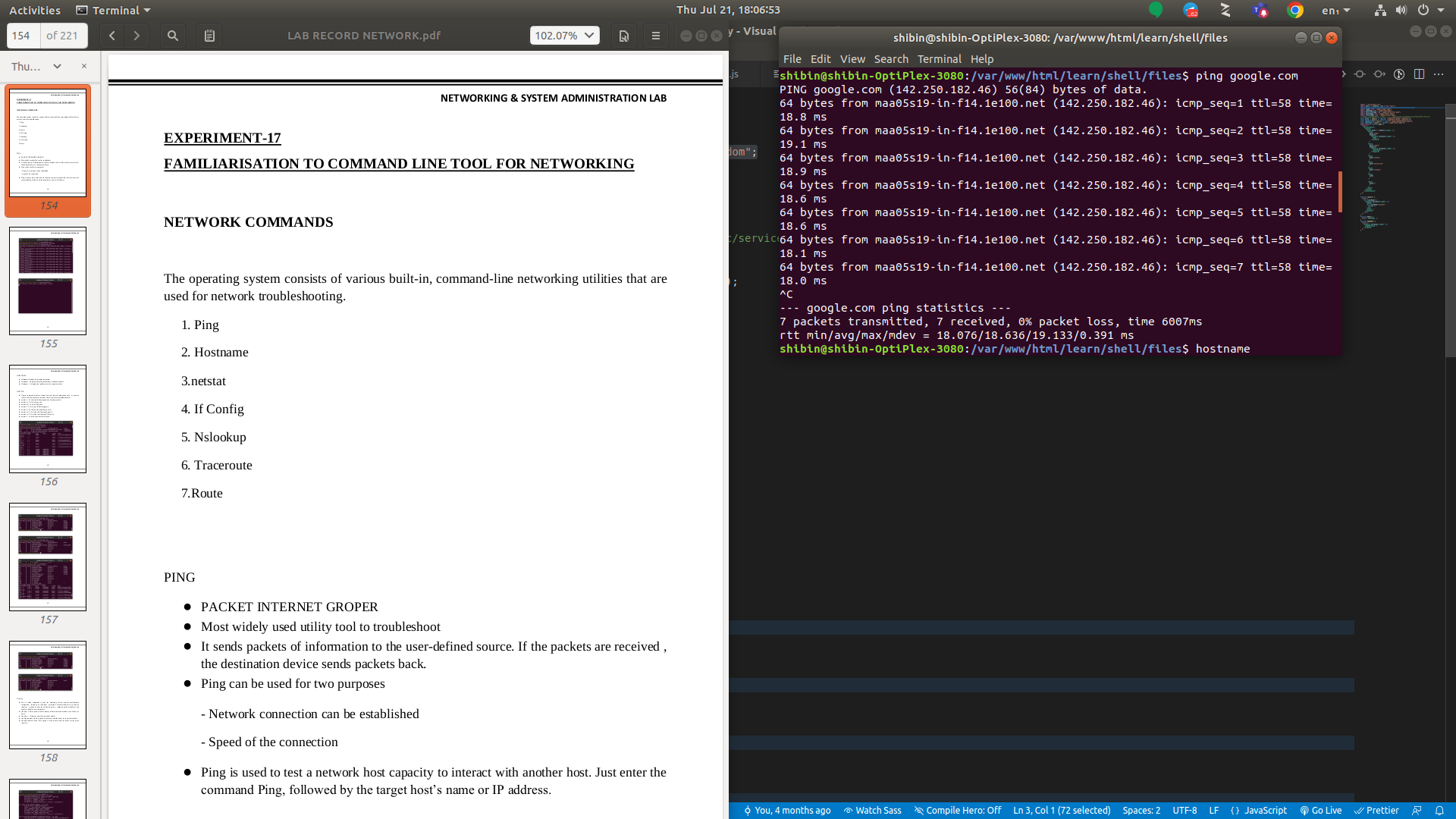
****

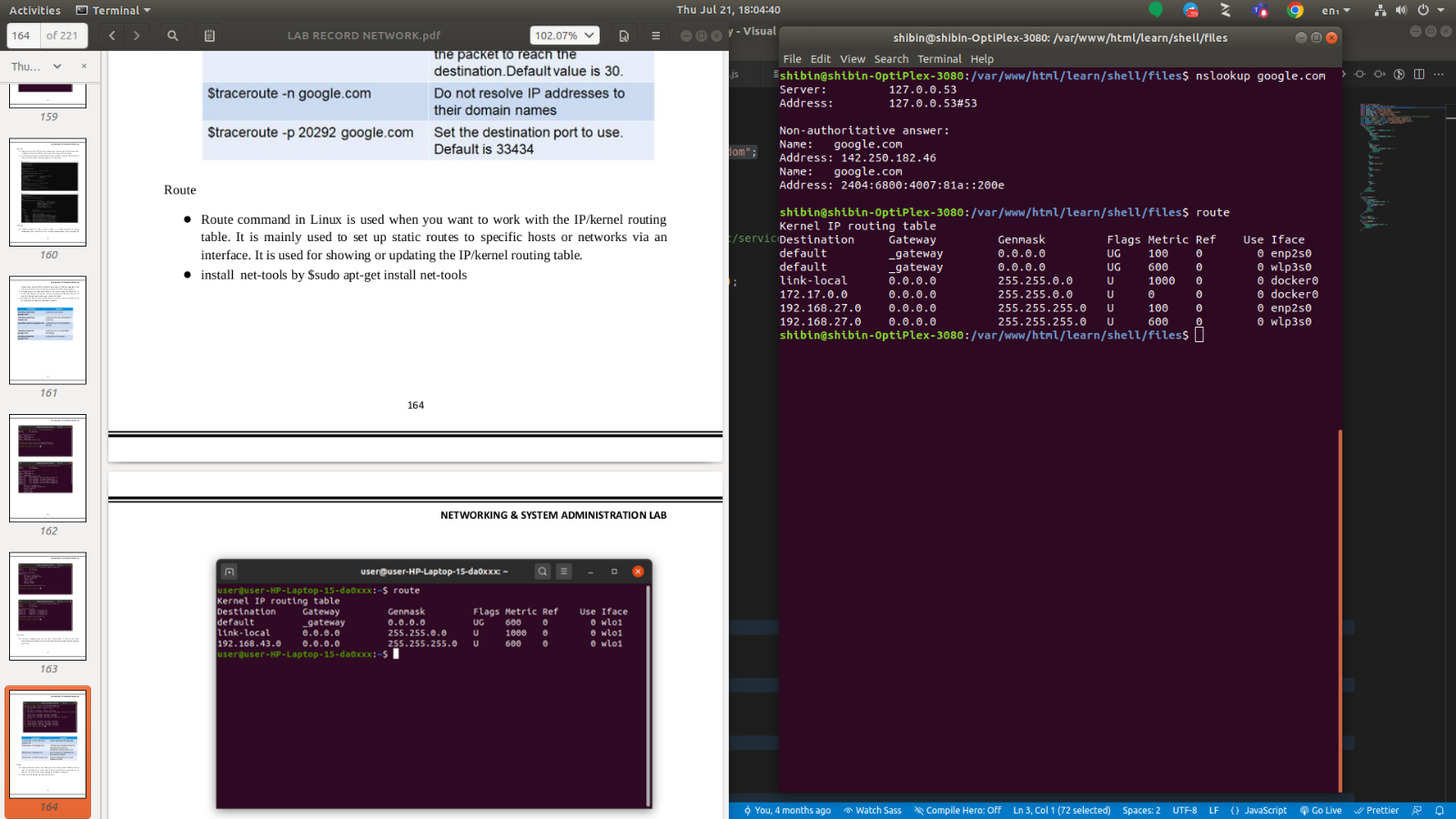
**LINUX COMMAND**

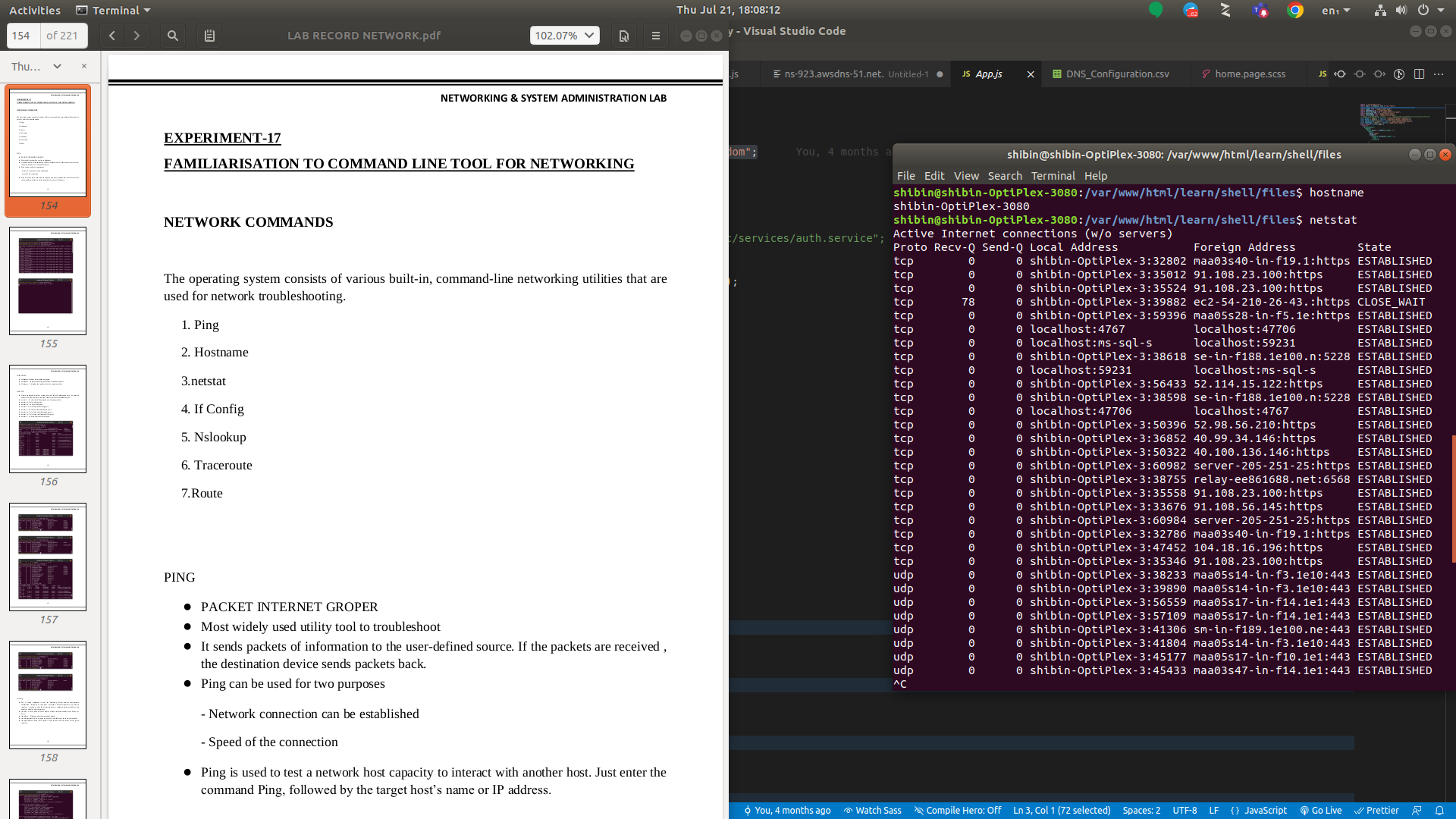
****

****

**NETWORK COMMAND**

****

****

****

****

**EXPERIMENT – 10**

**INSTALLATION OF LAMP ON UBUNTU**

**Introduction**

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

Step 1 — Installing Apache and Updating the Firewall

First, make sure your apt cache is updated with:

**$ sudo apt update**

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

**$ sudo apt install apache2**

Adjust the Firewall to Allow Web Traffic

**$ sudo ufw app list**

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

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

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

**sudo ufw allow "Apache Full”**

Step 2 — Installing MySQL

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

MySQL is a database management system. Basically,

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

Again, use apt to acquire and install this software:

**$ sudo apt install mysql-server**

When the installation is complete, run a simple security script that comes pre-installed with MySQL which will remove some dangerous defaults and lock down access to your database system.

Start the interactive script by running:

**$ sudo mysql\_secure\_installation**

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

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

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

**$ sudo mysql**

To exit the MySQL console, type:

**$ exit**

Step 3 — Installing PHP

Once again, leverage the apt system to install PHP.

In addition to the php package, you’ll also need libapache2-mod-php to integrate PHP into Apache, and the php-mysql package to allow PHP to connect to MySQL databases.

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

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

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

Step 4 — Testing PHP Processing on your Web Server

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

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

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

**sudo nano /var/www/your\_domain/info.php**

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

The address you will want to visit is:

[**http://your\_domain/info.php**](http://your_domain/info.php)

**EXPERIMENT – 11**

**INSTALLATION OF LARAVEL**

**LARAVEL**

•Laravel is an open-source PHP framework, which is robust and easy to understand. It follows a model-view-controller design pattern. Laravel reuses the existing components of different frameworks which helps in creating a web application. The web application thus designed is more structured and pragmatic.

•Laravel offers a rich set of functionalities which incorporates the basic features of PHP frameworks like CodeIgniter, Yii and other programming languages like Ruby on Rails. Laravel has a very rich set of features which will boost the speed of web development.

•If you are familiar with Core PHP and Advanced PHP, Laravel will make your task easier. It saves a lot of time if you are planning to develop a website from scratch. Moreover, a website built in Laravel is secure and prevents several web attacks.

**ADVANTAGES OF LARAVEL**

•Laravel offers you the following advantages, when you are designing a web application based on it –

•The web application becomes more scalable, owing to the Laravel framework.

•Considerable time is saved in designing the web application, since Laravel reuses the components from other frameworks in developing web applications.

•It includes namespaces and interfaces, thus help to organize and manage resources

**COMPOSER**

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

**INSTALLATION OF LARAVEL**

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

https://getcomposer.org/download/

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

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

•Step 3 − Create a new directory anywhere in your system for your new Laravel project. After that, move to the path where you have created the new directory and type the following command there to install Laravel. composer create-project –prefer-dist laravel/laravel folder\_name But the latest version(currently 8.x) just requires you to type

composer create-project laravel/laravel folder\_name

•Step 4 − The above command will install Laravel in the current directory. Start the Laravel service by executing the following command. php artisan serve

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

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

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

**EXPERIMENT – 12**

**FAMILIARISATION TO COMMAND LINE TOOL FOR NETWORKING**

**NETWORK COMMANDS**

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

1. Ping

2. Hostname

3. netstat

4. If Config

5. Nslookup

6. Traceroute

7.Route

PING

● PACKET INTERNET GROPER

● Most widely used utility tool to troubleshoot

● It sends packets of information to the user-defined source. If the packets are received , the destination device sends packets back.

● Ping can be used for two purposes

- Network connection can be established

- Speed of the connection

● Ping is used to test a network host capacity to interact with another host. Just enter the command Ping, followed by the target host’s name or IP address.

HOSTNAME

● Hostname :displays the machine hostname

● Hostname –f :displays the fully qualified host and domain name

● Hostname –I :displays the ip address for the current machine

NETSTAT

● Netstat command displays various network related information such as network connections, routing tables, interface statistics, multicast memberships etc.

● netstat -a : To show both listening and non-listening sockets

● netstat -at : To list all tcp ports.

● netstat -au : To list all udp ports.

● netstat -l : To list only the listening ports.

● netstat -lt : To list only the listening tcp ports

● netstat -lu : To list only the listening udp ports.

● netstat -lx : To list only the listening UNIX ports.

● netstat -s : To list the statistics for all ports

If config

● The if config commands is used for displaying current network configuration information , setting up an ip address , netmask or broadcast address to an network interface , creating an alias for network interface , setting up hardware address and enable or disable network interface

● ifconfig –a :This option is used to display all the interfaces available, even if they are down.

● ifconfig -s : Display a short list, instead of details

● ifconfig interface up :This option is used to activate the driver for the given interface

● ifconfig interface down :This option is used to deactivate the driver for the given interface.

Ip config

● Displays all current TCP/IP network configuration values and refreshes Dynamic Host Configuration Protocol (DHCP) and Domain Name System (DNS) settings.

● Used without parameters, ipconfig displays Internet Protocol version 4 (IPv4) and IPv6 addresses, subnet mask, and default gateway for all adapters.

Nslookup

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

● nslookup google.com :nslookup followed by the domain name will display the “A Record” (IP Address) of the domain. Use this command to find the address record for a domain. It queries domain name servers and gets the details.

● nslookup 192.168.0.10 : Reverse DNS lookup we can also do the reverse DNS look-up by providing the IP Address as argument to nslookup.

Traceroute

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

Route

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

● install net-tools by $sudo apt-get install net-tools

**STATIC AND DYNAMIC IP**

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

● It is required to set up an Ubuntu static IP address in order to access a device remotely and without losing a connection over the network.

● It is used to connect to an IP camera, home file server, game server, and many other devices.

● A static IP address is necessary only for servers and not for personal PCs

STEPS FOR SETTING STATIC IP

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

● Select manual and enter your desired IP address, netmask, gateway and DNS settings.

● Once ready click the Apply button.

● Turn OFF and ON switch to apply your new network static IP configuration settings

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

DYNAMIC IP

● A dynamic IP address as its name suggests is a temporary IP address assigned by a DHCP server for every new network.

● A dynamic IP address is used due to the shortage of IP addresses on IPV4. ● A single dynamic IP address can be used between many devices

Configuring a dynamic ip address

Step1: type the command in the terminal sudo nano /etc/netplan/01-network-manager-all.yaml

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

● dhcp4: yes

● dhcp6: yes

**SETTING UP A FIREWALL FOR LAN**

● The default firewall configuration tool for Ubuntu is ufw.

● Developed to ease iptables firewall configuration.

● ufw provides a user-friendly way to create an IPv4 or IPv6 host-based firewall.

● ufw by default is initially disabled.

● ufw is not intended to provide complete firewall functionality via its command interface, but instead provides an easy way to add or remove simple rules.

● It is currently mainly used for host-based firewalls.”

● To ENABLE ufw

Step1:check current firewall status sudo ufw status

Step2:to enable firewall sudo ufw enable

Step3:to check status sudo ufw status

● To Disable ufw

Step1:To disable firewall

sudo ufw disable

Step2:to check status

sudo ufw status

**EXPERIMENT – 13**

**FAMILIARISATION OF STATIC AND DYNAMIC IP**

**STATIC IP**:

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

2. It is required to set up an Ubuntu static IP address in order to access a device remotely and without losing a connection over the network.

3. It is used to connect to an IP camera, home file server, game server, and many other devices.

4. A static IP address is necessary only for servers and not for personal PCs.

**STEPS FOR SETTING STATIC IP:**

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

2. Click on the settings icon to start IP address configuration Select IPv4 tab.

3. Select manual and enter your desired IP address, netmask, gateway and DNS settings.

4. Once ready click Apply button.

5. Turn OFF and ON switch to apply your new network static IP configuration settings.

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

**DYNAMIC IP**

1. A dynamic IP address as its name suggests is a temporary IP address assigned by a DHCP server for every new network.

2. A dynamic IP address is used due to the shortage of IP addresses on IPV4.

3. A single dynamic IP address can be used between many devices.

Configuring a dynamic ip address

Step1: type the command in the terminal

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

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

dhcp4: yes

dhcp6: yes

Step3: Apply the changes with sudo netplan apply command.

**EXPERIMENT – 14**

**CONCEPT OF SUBNETS AND CIDR ADDRESS SCHEME**

**SUBNETS:**

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

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

1100 0000 - 1010 1000 - 0000 0000 - 0000 1111

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

1111 1111 - 1111 1111 - 1111 1111 - 0000 0000

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

1100 0000 - 1010 1000 - 0000 0000 - 0000 0000

This can be expressed as 192.168.0.0. The host specification is then the difference between these original value and the host portion. In our case, the host is 0000 1111 or 15. The idea of subnetting is to take a portion of the host space of an address, and use it as an additional networking specification to divide the address space again. For instance, a netmask of 255.255.255.0 as we saw above leaves us with 254 hosts in the network (you cannot end in 0 or 255 because these are reserved).

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

1100 0000 - 1010 1000 - 0000 0000

The host portion is: 0000 1111 We can use the first bit of our host to designate a subnetwork.

We can do this by adjusting the subnet mask from this: 1111 1111 - 1111 1111 - 1111 1111 - 0000 0000 To this: 1111 1111 - 1111 1111 - 1111 1111 - 1000 0000

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

**CIDR NOTATION:**

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

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

1100 0000 - 1010 1000 - 0000 0000 - 0000 0000

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

1100 0000 - 1010 1000 - 0000 0001 - 0000 0000

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

1111 1111 - 1111 1111 - 1111 1110 - 0000 0000

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

### EXPERIMENT – 15

### CONCEPT OF SUBNET MASK

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

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

11000000.10101000.01111011.10000100 - IP address (192.168.123.132)

11111111.11111111.11111111.00000000 - Subnet mask (255.255.255.0)

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

11000000.10101000.01111011.00000000 - Network address (192.168.123.0)

00000000.00000000.00000000.10000100 - Host address (000.000.000.132)

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

Some other common subnet masks are:

Decimal Binary 255.255.255.192 1111111.11111111.1111111.11000000

255.255.255.224 1111111.11111111.1111111.11100000

Internet RFC 1878 (available from InterNIC-Public Information Regarding Internet Domain Name

Registration Services) describes the valid subnets and subnet masks that can be used on TCP/IP networks

### EXPERIMENT – 16

### SETTING UP OF FIREWALL ON LAN

**ufw-uncomplicated firewall:**

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

**To Enable uf:**

Step1:check current firewall status

**sudo ufw status**

Step2:to enable firewall

**sudo ufw enable**

Step3: to check status

**sudo ufw status**

**To Disable ufw:**

Step1: To disable firewall

**sudo ufw disable**

Step2: to check status

**sudo ufw status**

### EXPERIMENT – 17

### WIRESHARK AND TCPDUMP

**What is WIRESHARK ?**

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

**Where it use ?**

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

### Steps to install WIRESHARK

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

Step 2: run application and click on noted

Step 3: select components and click next

Step 4: choose default destination location

Step 5: installation start running

Step 6: click on I agree

Step 7: complete installation part and click finish

**How it capture packets?**

* Wireshark captures packets and lets you examine their contents
* Select any interface to capture its packet
* No. shows the number of captured packet or index number.
* Time shows the time of capture
* Source shows the source ip of the packet or the packet is originally generated from which source ip.
* Destination shows the destination ip where the packet is going.
* Protocol shows which kind of protocol communication is held between the source and destination.
* Info shows the data payload in the packet

### Features of WIRESHARK

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

### TCPDUMP

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

### Steps to install tcpdump

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

**sudo apt update**

**sudo apt install tcpdump**

### TCPDUMP command examples

1. Display Available Interfaces

**# tcpdump –D**

1. Capture Packets from Specific Interface

**# tcpdump -i any**

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

1. Print Captured Packets in ASCII

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

**# tcpdump -A -i any**

1. Capture Only N Number of Packets

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

**# tcpdump -c 5 -i any**

1. Display Captured Packets in HEX and ASCII

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

**# tcpdump -XX -i any**

1. Capture and Save Packets in a File

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

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

### Testing network services with Netcat [nc]

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

Step 1: $ sudo apt-get install netcat

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

Step 3: Set up the server using netcat in listening mode.

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

Step 4: Creating the server with netcat

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

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

### EXPERIMENT – 18

### ANALYSE PACKETS USING WIRESHARK

1. List 3 different protocols that appear in the protocol column in the unfiltered packet-listing window.

Support your answer with an appropriate screenshot from your computer.

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

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

The time interval between a particular HTTP GET message and HTTP OK message is 20.762037849s - 20.666049819s = 0.09598803s

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

gaia.cs.umass.edu: 128.119.245.12

Local computer: 192.168.160.204

1. Print the two HTTP messages (GET and OK) referred to in question 2 above. To do so, select Print from the Wireshark File command menu, and select the “Selected Packet Only” and “Print as displayed” radial buttons, and then click OK

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

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

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

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

### EXPERIMENT – 19

### FAMILIARISATION TO HYPERVISORS AND VIRTUAL MACHINES

**Virtual Machine:**

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

**Hypervisor:**

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

**Hypervisor types**

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

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

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

#### Type 2: hosted hypervisors

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

Eg. VMWare Workstation, VirtualBox

**Benefits of Hypervisor**

1. Speed
2. Efficiency
3. Flexibility
4. Portability

### EXPERIMENT – 20

### INSTALLING SOFTWARE FROM SOURCE CODE

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

**Package Managers**

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

**>> Configure Script:**

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

**>> Make:**

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

**>> Make Command:**

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

$ configure

$ make

$ make install

**Installation Steps:**

Step 1: Open the Linux terminal and enter

***sudo apt update***

This command is used to install the latest versions of the packages currently installed on the user's system from the sources enumerated in /etc/apt/sources. The installed packages which have new packages available are retrieved and installed.

Step 2: Enter

***sudo apt install build-essential***

build-essential is called a meta-package. It in itself does not install anything. Instead, it is a link to several other packages that will be installed as dependencies. In the case of the build-essential metapackage, it will install everything required for compiling basic software written in C and C++..

Step 3: Enter

***cd /usr/local/src/***

The cd command, also known as chdir (change directory), is a command-line shell command used to change the current working directory in various operating systems.

Step 4: Enter

***sudo wget http://www.noip.com/client/linux/noip-duc-linux.tar.gz***

The wget command is a command line utility for downloading files from the Internet. It supports downloading multiple files, downloading in the background, resuming downloads, limiting the bandwidth used for downloads and viewing headers.

Step 5: Enter

***sudo tar xf noip-duc-linux.tar.gz and cd noip-2.1.9-1/***

The Linux ‘tar’ stands for tape archive, is used to create Archive and extract the Archive files. tar command in Linux is one of the important commands which provides archiving functionality in Linux. We can use Linux tar command to create compressed or uncompressed Archive files and also maintain and modify them.

Step 6: Enter

***sudo make install***

The make install command will copy the built program, and its libraries and documentation, to the correct locations.