**TABLE OF CONTENTS**

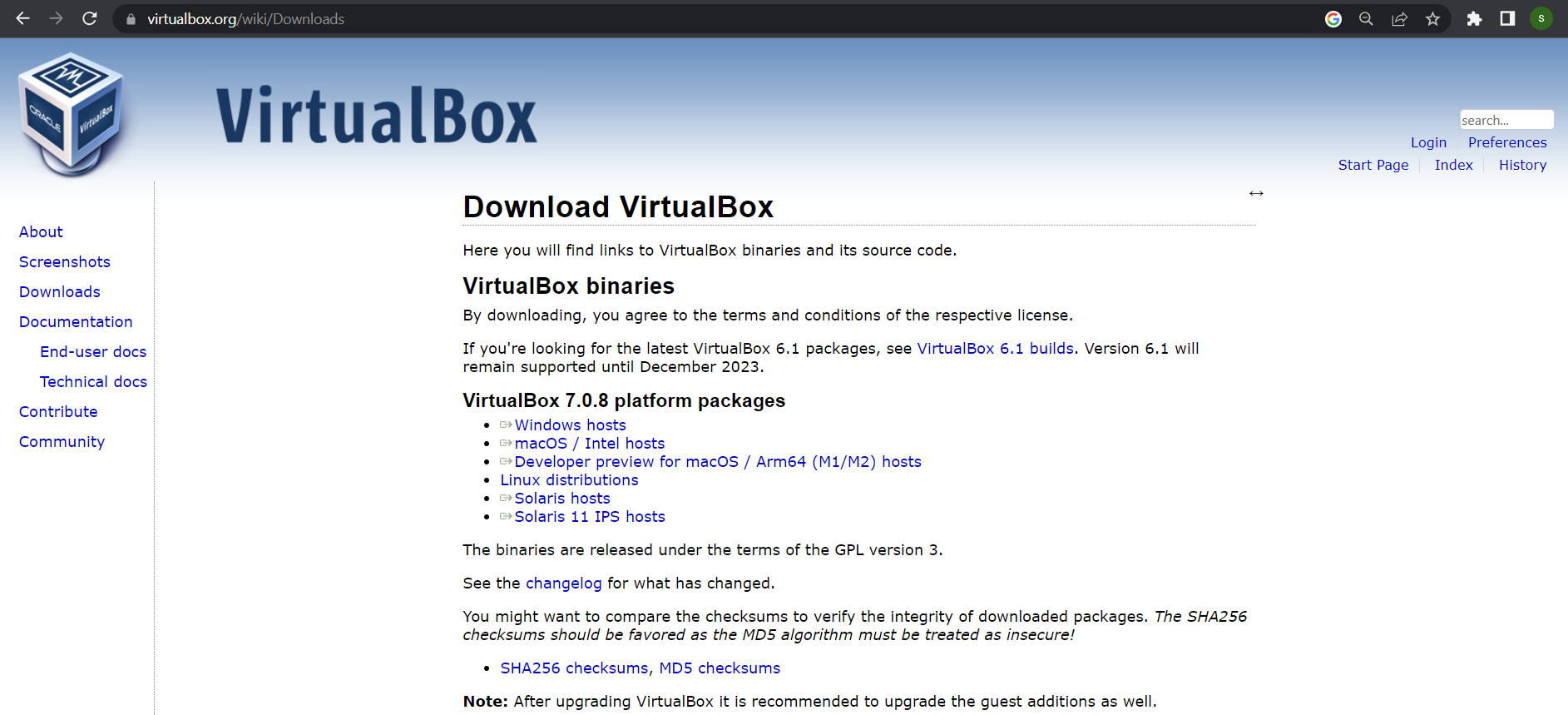
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No** | **PROGRAM** | **Date Assigned** | **Date Submitted** | **Signature** |
| 1 | Lab 1  To install a Linux distribution operating system using a virtual machine and also using dual boot. | 12-07-23 | 19-07-23 |  |
| 2 | Lab 2  To study Linux general purpose utility command list ( mv, rm, less, head, tail, grep, wc, ps, sleep, jobs, kill, bg, fg, man, locate, find, ls, who, whoami, cat, touch, mkdir, cp, pwd, uname, chmod, cd, >, >>,\*, ? ). | 19-07-23 |  |  |
| 3 | Lab 3  To study and implementation of the File Transfer Protocol (FTP) in Linux. | 02-08-23 |  |  |
| 4 | Lab 4  To install Docker in Linux and run hello world on it. | 09-08-23 |  |  |
| 5 | Lab 5  To implement Network File System (NFS) in Linux. | 16-08-23 |  |  |
| 6 | Lab 6   * Write a shell script to display list of users currently logged in the system. * Write a shell script to display “HELLO WORLD” * Write a shell script to check whether the given number is even or odd. | 23-08-23 |  |  |
| 7 | Lab 7   * Write a shell script to search whether element is present in the list or not. * Write a shell script to check whether given file is a directory or not. * Write a shell script to count the number of files in a directory. | 30-08-23 |  |  |
| 8 |  | 13-09-23 |  |  |
| 9 |  |  |  |  |
| 10 |  |  |  |  |

**Lab 1**

**Aim-** To install a Linux distribution operating system using a virtual machine and also using dual boot.

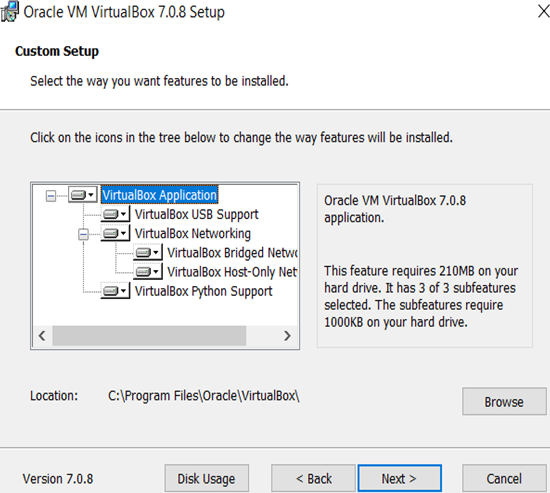
**Installation of Linux using Virtual machine**

Step 1: Install Oracle Virtual Box.



A screen shot of a computer

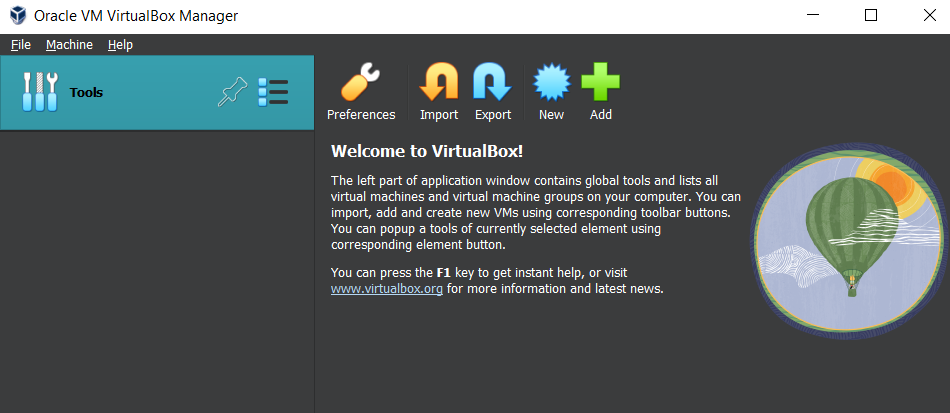
Description automatically generated

A screenshot of a computer

Description automatically generated 

Step 2: Click finish and then open virtual box.

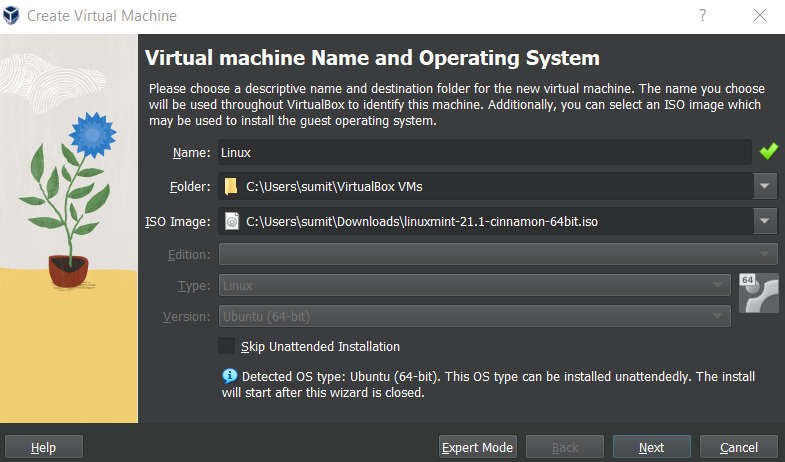


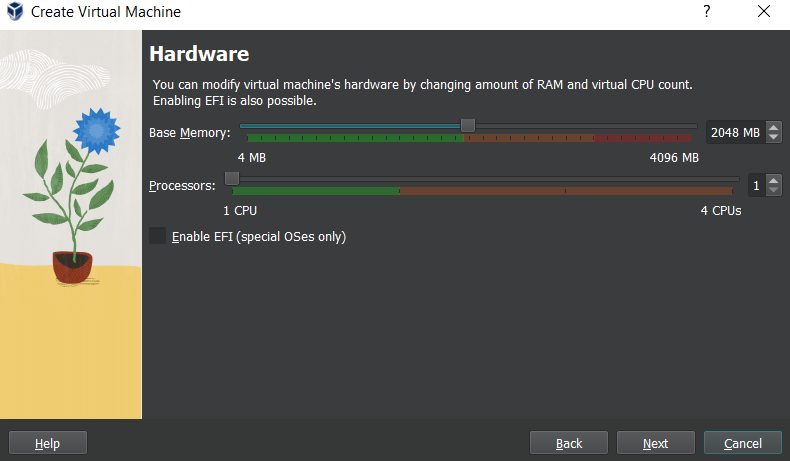
Step 3: Download the ISO file of the Linux distribution. Eg Mint Linux 64-bit. Use Advancedhosters.com

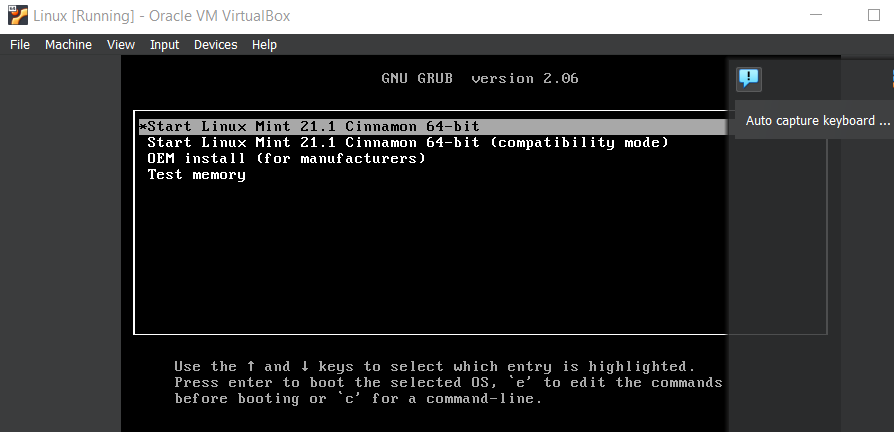
A screenshot of a computer

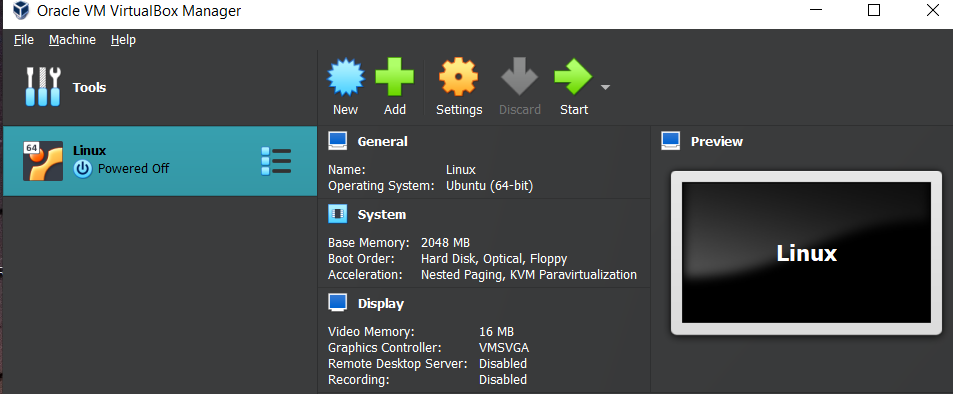
Description automatically generated

Step 4: Click on New in the virtual box and select the Linux ISO image and click next.











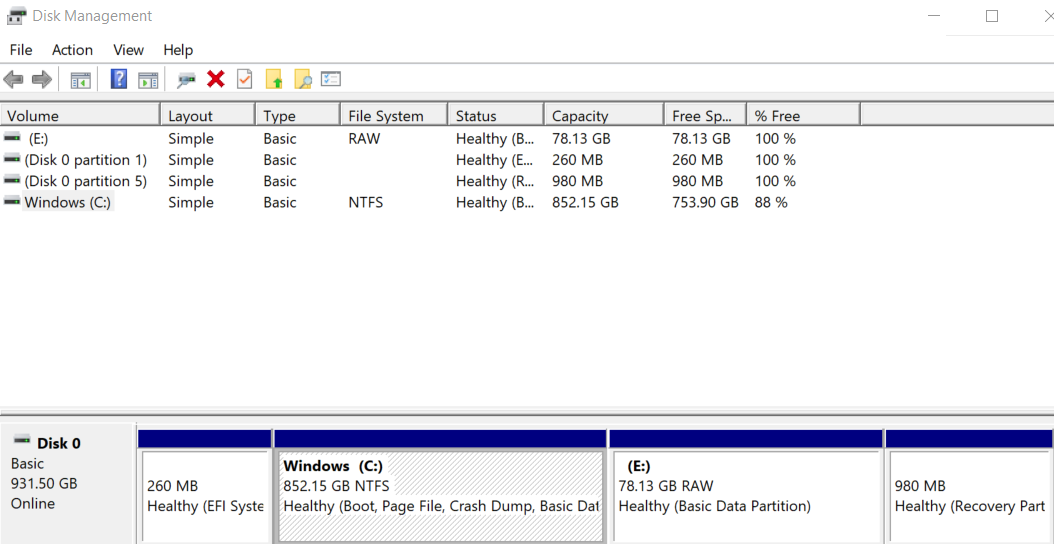
Step 5: Click on install Linux Mint.

A screenshot of a computer

Description automatically generated

**Installation of Linux using Dual boot**

Step 1: Create disk partition in the system.



Step 2: Download the Linux distribution for e.g. Mint Linux 64 bit ISO image. Use Advancedhosters.com

A screenshot of a computer

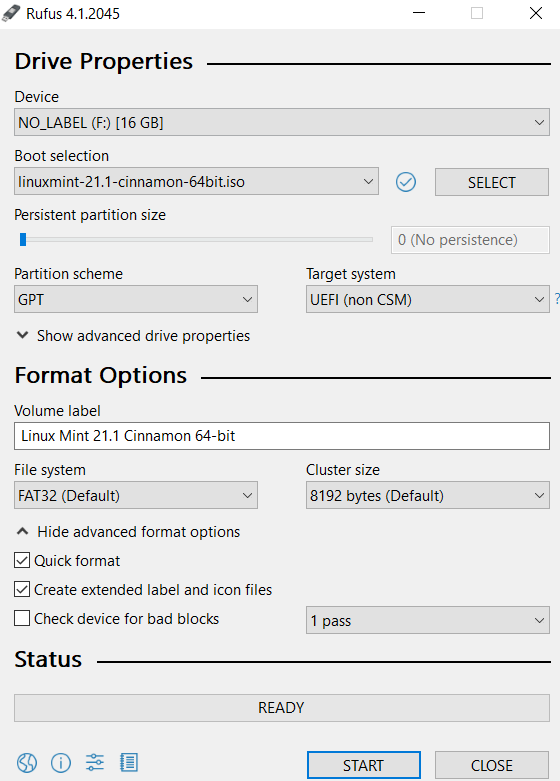
Description automatically generated

Step 3: Create ISO bootable USB flash drive. For this install Rufus in the system.

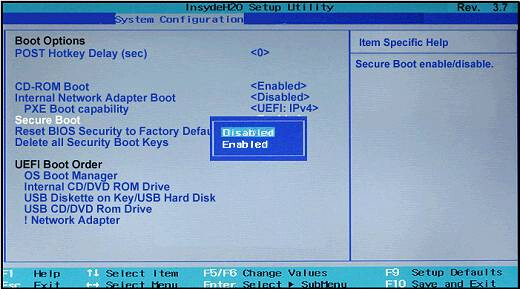
A screenshot of a computer

Description automatically generated

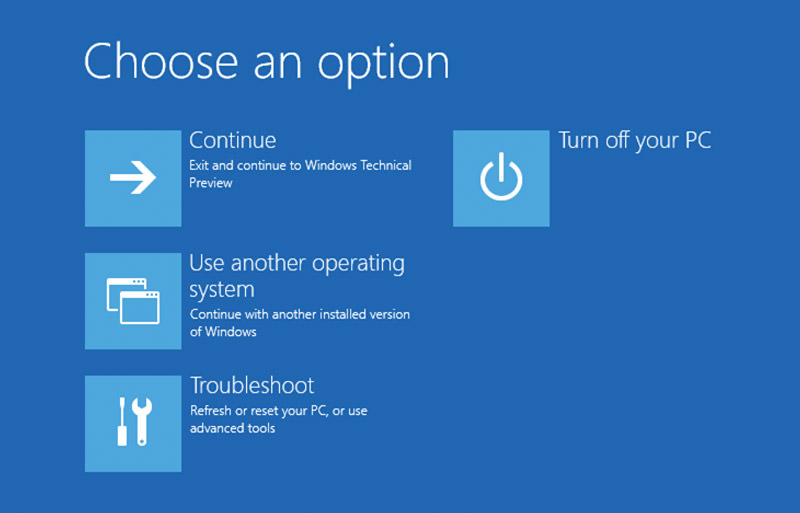
Step 4: Start Rufus and select the downloaded ISO file and click start it will create a bootable USB drive.



Step 5: Disable the secure boot option in the BIOS so that the OS can be installed.



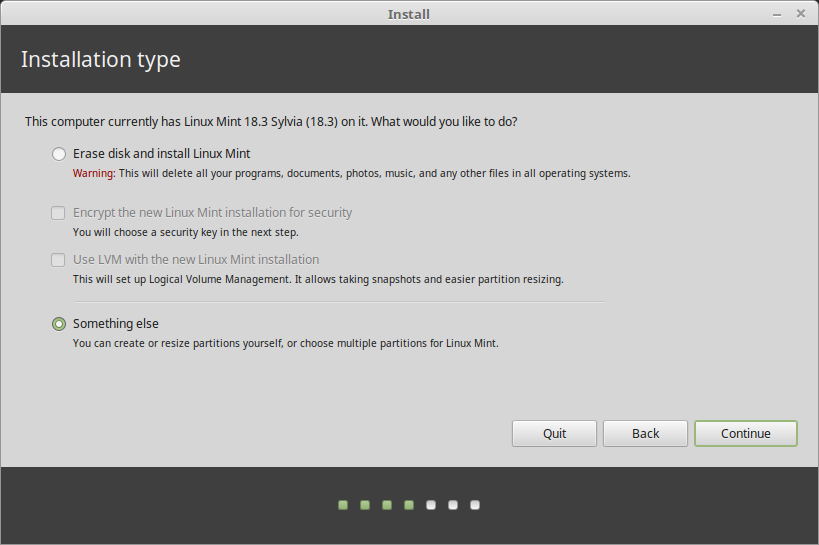
Step 6: Press and hold shift and click restart in the start->power button.

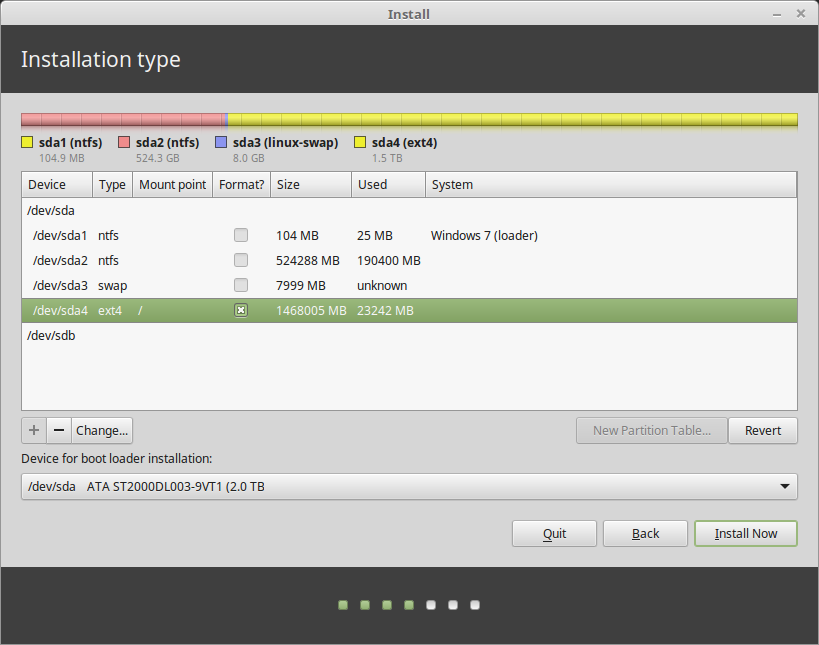


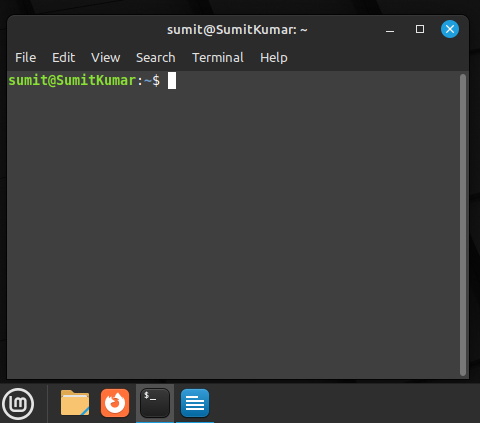


Step 7: Click on install Linux Mint.

Step 8: Select the partition and click install.







|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Total Marks** | **Marks Obtained** | **Comments** |
| Concept (A) | 2 |  |  |
| Implementation (B) | 2 |  |  |
| Performance (C) | 2 |  |  |
| Total | 6 (to be scaled down to 1) | | |

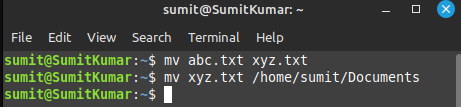
**Lab 2**

**Aim-** To study Linux general purpose utility command list ( mv, rm, less, head, tail, grep, wc, ps, sleep, jobs, kill, bg, fg, man, locate, find, ls, who, whoami, cat, touch, mkdir, cp, pwd, uname, chmod, cd, >, >>,\*, ? ).

**Command and Output**

1. **mv**- this command stands for move. It has two functions renaming a file or directory and moving a file or directory to another location.

Syntax for renaming a file:- mv [source\_file\_name(s)] [new\_file\_name]

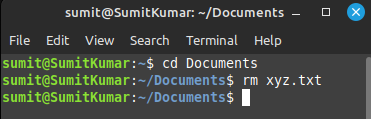
Syntax for moving a file:- mv [source\_file\_name(s)] [Destination\_path]

2. **rm-** This command is used to remove objects such as files, directories, symbolic links and so on from the file system.

Syntax:- rm [options] filename

-i: (Interactive Deletion) option makes the command ask user for confirmation before removing

-f: (Force Deletion): rm prompts for confirmation removal if a file is write protected.

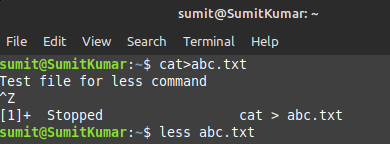
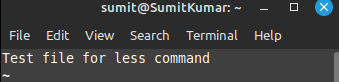
-r: (Recursive Deletion): command performs a tree-walk and will delete all the files and sub-directories recursively of the parent directory.

3. **less-** It is used to read the contents of a text file one page (one screen) at a time. It has faster access because if a file is large, it doesn’t access the complete file, but accesses it page by page.

Syntax:- less [options] filename

-E: Automatically exit when reaching the end of the file.

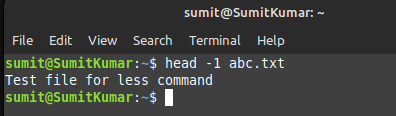
-f: Force non-regular files to be opened.

-I: Ignore cases when searching.

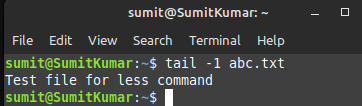
4. **head-** This command prints the first lines of one or more files (or piped data) to standard output. By default, it shows the first 10 lines.

Syntax:- head [option] filename

-n: shows the specified number of lines -c: show the specified number of bytes

-v: show the file name tag

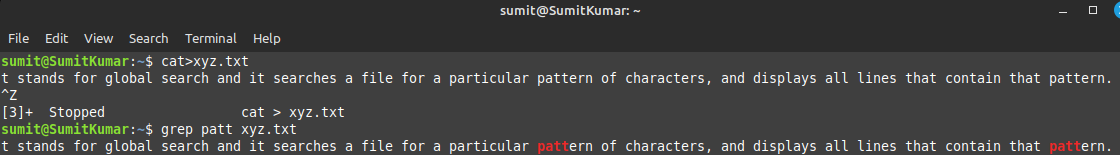
5. **tail-** It prints the last N number of data of the given input. By default it prints the last 10 lines of the specified files. The options for this command are same as that of head command.

Syntax:- tail [options] filename

6. **grep-** It stands for global search and it searches a file for a particular pattern of characters, and displays all lines that contain that pattern.

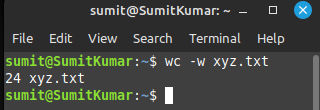
Syntax:- grep [options] pattern [files]

-i: Ignores, case for matching -l: Displays list of a filenames only.

-n: Display the matched lines and their line numbers.

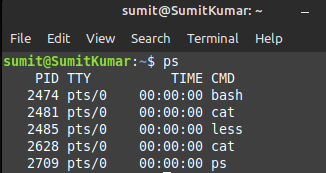
7. **wc-** It stands for word count. It is mainly used for counting purpose.

Syntax:- wc [options] filename

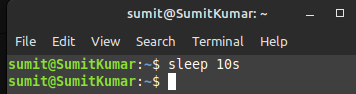
-l: prints the number of lines present in a file -w: prints the number of words present in a file

8. **ps-** It stands for process status. It lists current running processes and their PIDs.

Syntax:- ps [options]



9. **sleep-** This command is used to create a dummy job. This job creates delays in execution.

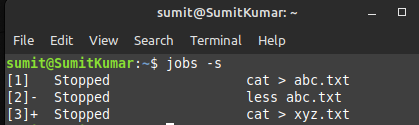
Syntax:- sleep time[second/minutes]

10. **jobs-** It is used to list the jobs that are running in the background and in the foreground.

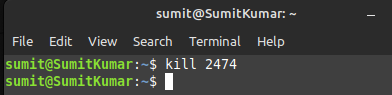
Syntax:- jobs [ JOB ]

-p Lists process IDs only.

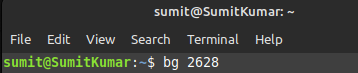
-r Restrict output to running jobs.

-s Restrict output to stopped jobs.

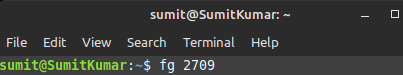
11. **kill-** This command is used to terminate processes manually. If we don’t specify the signal, the default signal `TERM` is sent to terminate the process

Syntax:- kill [signal] PID

12. **bg-** It resumes suspended process while keeping them running in the background.

Syntax:- bg [ job ]

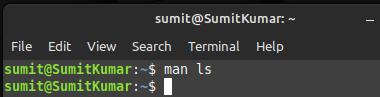
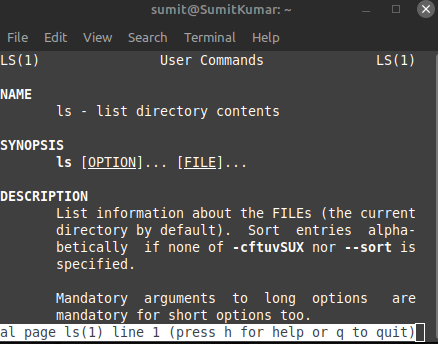
13. **fg-** It moves a background job in the current shell environment into the foreground.

Syntax:- fg [ %job ]

14. **man-** It is used to display the user manual of any command that can run on the terminal.

Syntax:- man [options] [command name]

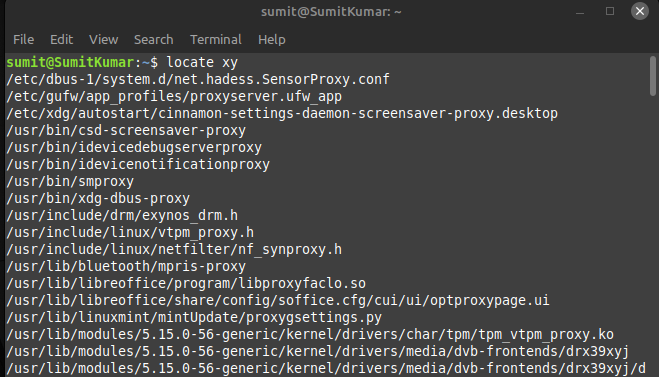
No Option: It displays the whole manual of the command.

Section-num: It is used to display only a specific section of a manual.

15. **locate-** It is used to find the files by name.

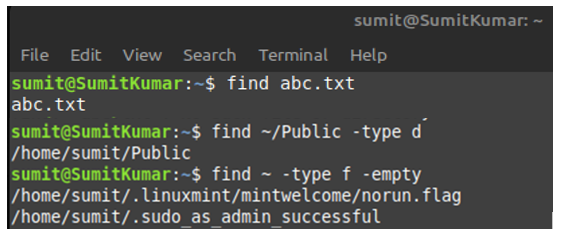
Syntax: locate [option] PATTERN

-b: matches only base name -c: writes number of matching entries only.



16. **find-** It is used to find files and directories and perform subsequent operations on them.

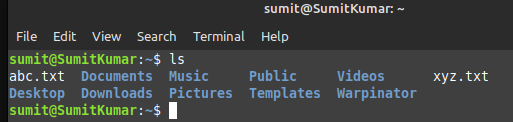
Syntax: find [start searching from] [expression what to find] [-options] [what to find]



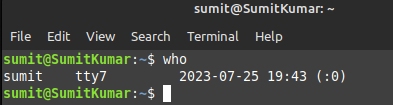
17. **ls-** It lists directory contents of files and directories

Syntax: ls [option] [file/directory]

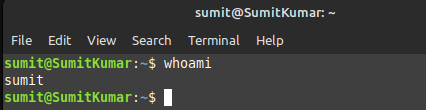
-l: known as a long format that displays detailed information about files and directories.

-a: Represent all files Include hidden files and directories in the listing.

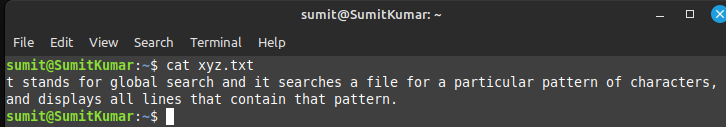
18. **who-** This command is used to find time of last system boot, current run level of the system and list of logged in users.

Syntax: who [options] [filename]

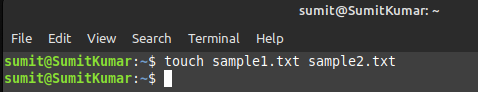
19. **whoami-** It displays the username of the current user when this command is invoked.

Syntax: whoami

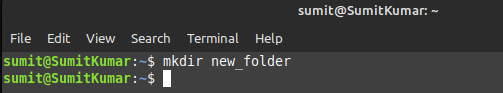
20. **cat-** It reads data from the file and gives its content as output. It helps us to create, view, and concatenate files.

Syntax: cat filename1

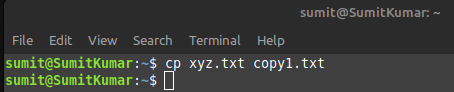
21. **touch-** It is used to create a file without any content. The file created using touch command is empty. It can create multiple files.

Syntax: touch filename1 filename2…

22. **mkdir-** This command can create multiple directories at once as well as set the permissions for the directories.

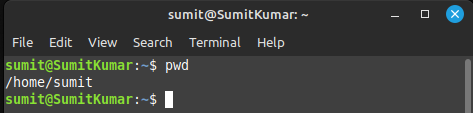
Syntax: mkdir [options] [directories]

23. **cp-** This command is used to copy files or groups of files or directories. It creates an exact image of a file on a disk with a different file name.

Syntax: cp Src\_file Dest\_file

24. **pwd-** It stands for Print Working Directory. It prints the path of the working directory, starting from the root.

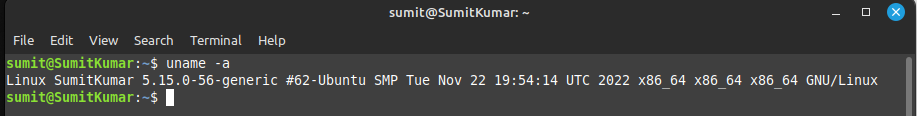
Syntax: pwd [options]

-L: Prints the symbolic path. -P: Prints the actual path.

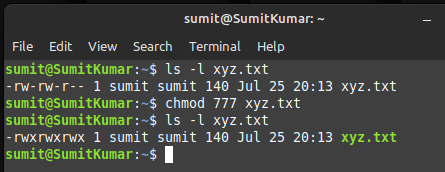
25. **uname**- The command displays the information about the system.

Syntax: uname [option]

-a: prints information in the order of kernel name, network hostname, kernel release date, etc

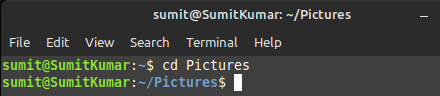
-s: It prints the kernel name. -v: It prints the version of the current kernel.

26. chmod- This command is used to change the access mode of a file. In this the permissions have three categories: read, write, and execute.

Syntax: chmod [options] [mode] [File\_name]

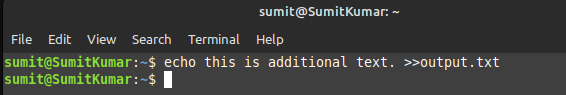
27. cd- It is used to move efficiently from the current working directory to different directories in our System.

Syntax: cd [directory\_name]

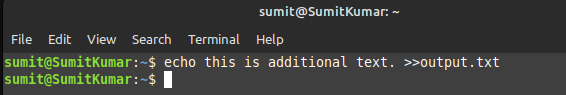
Syntax: cd / For going to root directory

28. >- This redirection operator is used to redirect the output of a command to a file instead of displaying it on the terminal. It overwrites the file.

E.g. $ echo This is a text file. > myfile.txt



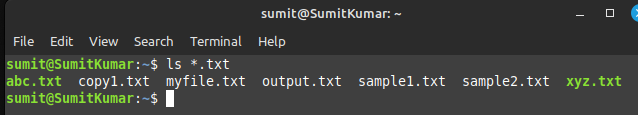
29. >>- This redirection operator is used to redirect the output of a command to a file. It appends the output to the file and does not overwrite it.

E.g. $ echo this is additional text >> output.txt

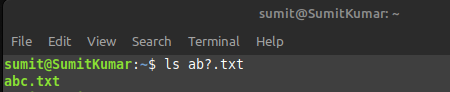
30. \*- This wildcard character is used to represent and match any combination of characters in file or directory names.

Matching Multiple Characters: $ ls \*.txt

Matching Multiple Files: $ rm \*.bak



31. ?- This wildcard character is used to represent and match a single character in file or directory name.

Matching a Single Character: $ ls file?.txt

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Total Marks** | **Marks Obtained** | **Comments** |
| Concept (A) | 2 |  |  |
| Implementation (B) | 2 |  |  |
| Performance (C) | 2 |  |  |
| Total | 6 (to be scaled down to 1) | | |

**Lab 3**

**Aim-** To study and implementation of the File Transfer Protocol (FTP) in Linux

The File Transfer Protocol (FTP) is a longstanding network protocol designed for transferring files between computers over a TCP-based network, typically the Internet. Developed in the early 1970s, FTP has evolved over time to enhance security, efficiency, and functionality. Its primary purpose is to facilitate the uploading, downloading, and management of files on remote servers.

The architecture of FTP follows a client-server model. In this setup, one computer functions as the client, initiating requests for files or directories from another computer acting as the server. The client establishes the connection, and the server responds to the client's commands and requests.

FTP offers two modes of operation: active and passive. Active mode involves the client opening a random port, with the server connecting to it for data transfer. However, this mode can pose challenges due to firewall and NAT configuration. Passive mode, on the other hand, sees the server opening a random port, which the client then connects to. This mode is generally more compatible with modern network setups.

FTP employs two communication channels: the control channel and the data channel. The control channel is used for sending commands and receiving responses, while the data channel is responsible for transferring actual file data. In active mode, the server initiates the data channel, while in passive mode, the client takes the initiative.

FTP on Linux:

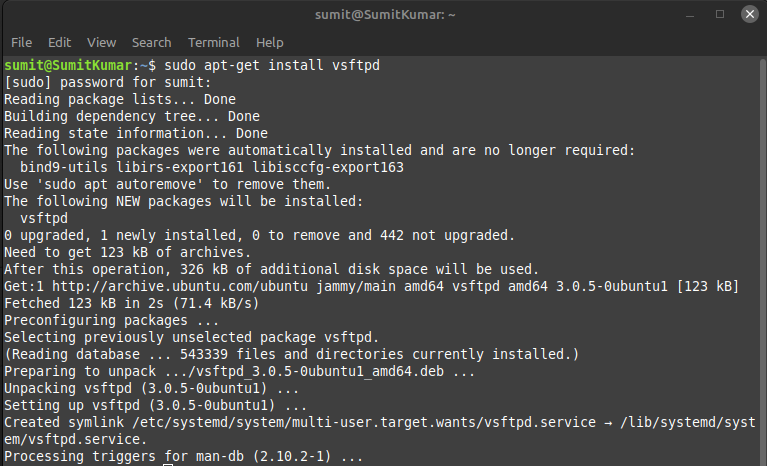
1. Command Line Interface: Linux and Unix systems provide a command-line interface that allows users to interact with FTP servers directly from the terminal. To initiate an FTP session, users typically open a terminal window and enter the ftp command followed by the target server's address.
2. Authentication: Upon connecting to the FTP server, users are prompted to provide their login credentials, including a username and password. Depending on server configuration, anonymous FTP access may be available.
3. Basic Commands: Users can issue various FTP commands in the terminal to navigate directories, list files, upload and download. Some common commands include:

* ls : List contents of the current directory on the remote server.
* cd : Change the remote directory.
* get : Download a file from the remote server.
* put : Upload a file to the remote server.
* pwd : Print the current remote directory.
* quit : End the FTP session and exit.

1. Binary and ASCII Modes: FTP allows users to transfer files in either binary or ASCII mode. Binary mode is used for transferring binary files like images or executables, while ASCII mode is used for text files to ensure proper line endings and character encoding.
2. Passive Mode: In many cases, Passive mode (PASV) is preferred when using FTP from a Linux/Unix terminal. Passive mode helps navigate firewall and NAT issues by allowing the client to establish a data connection to the server, which is particularly useful in scenarios where the server is behind a firewall.
3. Scripting: Linux/Unix systems enable users to create FTP scripts using shell scripting languages like Bash. This allows for automating repetitive FTP tasks, such as regular backups or scheduled file transfers.
4. Security Considerations: Using FTP in its original form can pose security risks, especially when transmitting login credentials in plain text. As a result, it's recommended to use more secure alternatives like Secure File Transfer Protocol (SFTP) or FTP Secure (FTPS), which provide encrypted communication.
5. FTP Clients: While the default ftp command-line tool is available on most Linux/Unix systems, there are also thirdparty FTP client applications that offer enhanced features, user interfaces, and support for secure protocols. Examples include FileZilla and lftp.

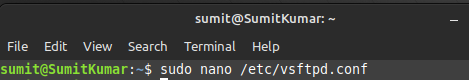
Step 1) Install vsftpd

Open a terminal window and run the following command to install vsftpd:



Step 2) Configure vsftpd:

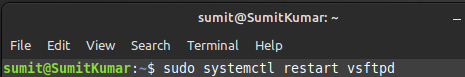
Once vsftpd is installed, you'll need to configure it. The configuration file for vsftpd is located at /etc/vsftpd.conf. You can edit it with a text editor like nano or vim:



* anonymous\_enable: Set this to NO if you want to allow only authenticated users to access your FTP server.
* local\_enable: Set this to YES to allow local Linux users to access the FTP server.
* write\_enable: Set this to YES if you want users to be able to upload files.

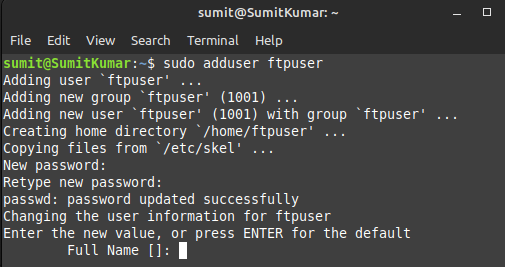
Step 3) Restart vsftpd:

After making changes to the configuration file, you should restart the vsftpd service for the changes to take effect:



Step 4) Create FTP User Accounts:

We can use the existing Linux Mint user accounts for FTP access, create new FTP-specific user accounts. To create a new user, use the adduser command:

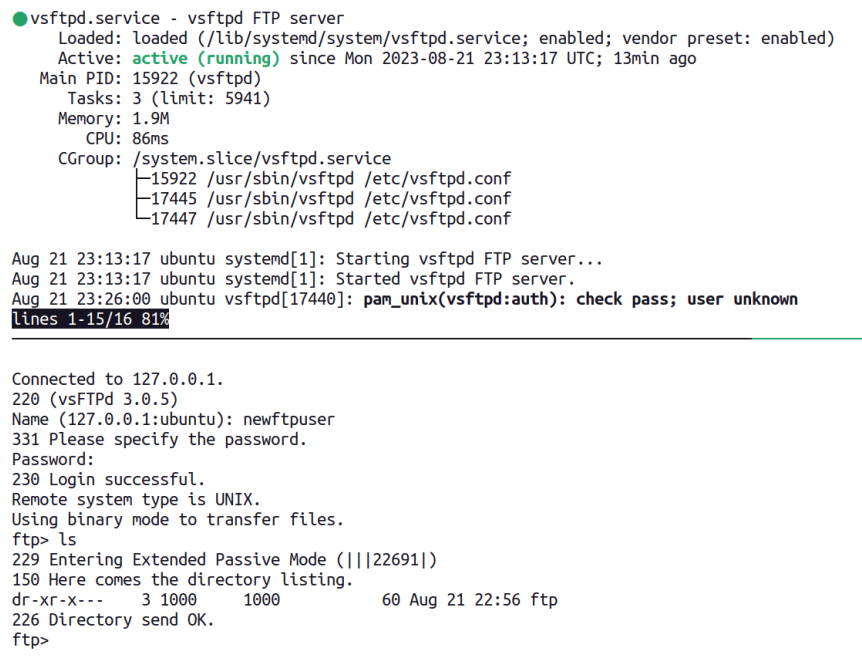


Step 5) Set Permissions:

Ensure that the user accounts have appropriate permissions to access their home directories and upload files. By default, when you create a user with adduser, a home directory is created for them in /home/username. Make sure the permissions are set correctly, especially if you want users to be able to write files.

Step 6) Firewall Configuration

If you are running a firewall on your Linux Mint system, make sure to open the necessary ports for FTP. FTP uses port 21 for control connections and additional ports for data transfer. You may need to configure your firewall to allow these ports.



|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Total Marks** | **Marks Obtained** | **Comments** |
| Concept (A) | 2 |  |  |
| Implementation (B) | 2 |  |  |
| Performance (C) | 2 |  |  |
| Total | 6 (to be scaled down to 1) | | |

**Lab 4**

**Aim-** To install Docker in Linux and run hello world on it.

**Docker** is a platform and toolset for developing, deploying, and running applications inside containers. Containers are lightweight, standalone, and executable packages that include everything needed to run a piece of software, including the code, runtime, libraries, and system tools. Docker provides a way to package and distribute applications in a consistent and isolated environment, making it easier to deploy and manage software across different environments and systems.

**Docker Container:** A container is a runnable instance of a Docker image. It encapsulates an application and its dependencies in an isolated environment. Containers are highly portable and can run consistently on any system that supports Docker, regardless of differences in the underlying infrastructure.

**Docker Image:** An image is a lightweight, read-only blueprint that contains all the instructions and dependencies needed to create a container. Images are used as the basis for creating containers. They can be stored and versioned in Docker registries, making it easy to share and distribute applications.

**Docker Engine:** The Docker Engine is the core component of Docker that runs and manages containers. It consists of a server and a command-line interface (CLI). The server is responsible for building, running, and managing containers, while the CLI provides a user-friendly interface for interacting with Docker.

**Docker file:** A Docker file is a plain text configuration file that specifies the steps and instructions for building a Docker image. It includes information such as the base image, environment variables, dependencies, and application code.

**Docker Compose:** Docker Compose is a tool for defining and running multi-container Docker applications. It allows you to define complex applications consisting of multiple containers and their interdependencies in a single YAML file.

**Docker Registry**: A Docker registry is a centralized repository for storing and sharing Docker images. Docker Hub is a popular public registry, but organizations can also set up private registries for their own images.

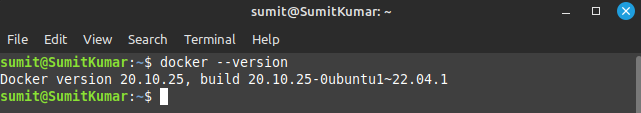
**Docker Swarm and Kubernetes:** Docker Swarm and Kubernetes are container orchestration tools that allow you to manage and scale containers in a clustered environment. They provide features for load balancing, service discovery, and automated deployment of containers.

Step 1: Open terminal in Linux and install docker files.

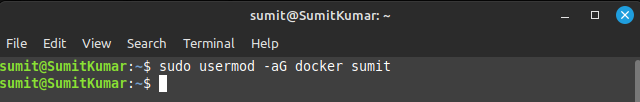
A screenshot of a computer program

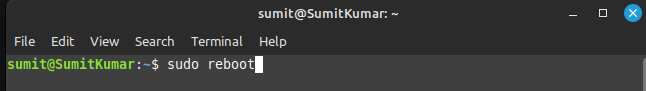
Description automatically generated

Step 2: Check the docker version installed.

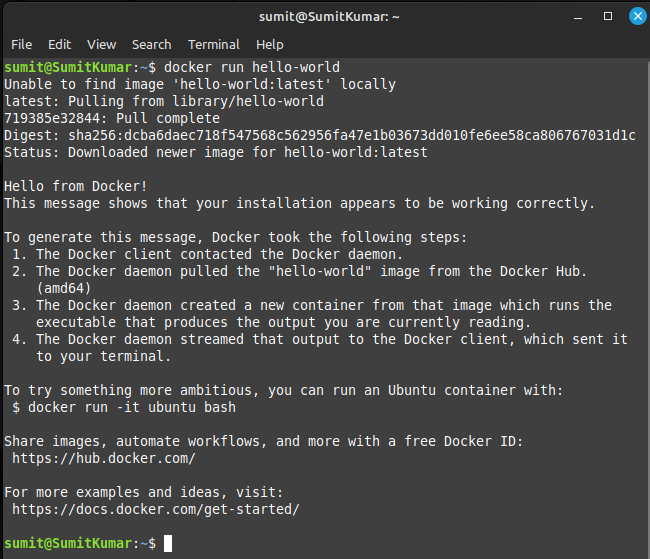


Step 3: Add current user to the docker group so that the system user can use docker.



Step 4: Reboot the system.

Step 5: Run the hello world image.



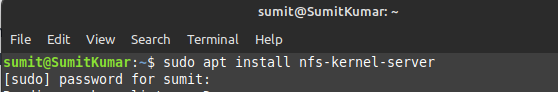
|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Total Marks** | **Marks Obtained** | **Comments** |
| Concept (A) | 2 |  |  |
| Implementation (B) | 2 |  |  |
| Performance (C) | 2 |  |  |
| Total | 6 (to be scaled down to 1) | | |

**Lab 5**

**Aim-** To implement Network file system in Linux,

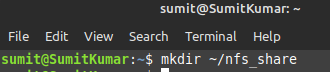
NFS (Network File System) in Linux Mint involves setting up an NFS server to share directories with other computers on the network and configuring client machines to access those shared directories.

Step 1: Install NFS Server on the Server Machine

1. Open a terminal on your Linux Mint server.
2. Install the NFS server package:

Step 2: Create a Directory to Share

1. Choose a directory on server that you want to share with NFS. For example, you can create a directory named "nfs\_share" in your home directory:



1. Set appropriate permissions for the shared directory:

A black background with white text

Description automatically generated

1. Configure the directory to be shared by adding it to the NFS exports file:



Add the following line to the file to export your directory with read-write access:

A black and white text

Description automatically generated

Save and exit the text editor.

1. Reload the NFS server configuration to apply the changes:



Step 3: Start and Enable the NFS Server

1. Start the NFS server:

A black background with white text

Description automatically generated

1. Enable the NFS server to start automatically at boot:



Step 4: Configure the Firewall

A screenshot of a computer program

Description automatically generatedIf you have a firewall running on your server, you need to allow NFS traffic through it. By default, NFS uses ports 2049 (TCP and UDP). You can open these ports using ufw (Uncomplicated Firewall):

Step 5: Mount NFS Shares on Client Machines

On the client machines (other computers on your network), you can mount the NFS share. Replace server\_ip with the IP address of your NFS server and nfs\_share with the name of the shared directory.



To make the NFS share mount automatically at boot, add an entry to the /etc/fstab file:



Add the following line:

Save and exit the text editor.

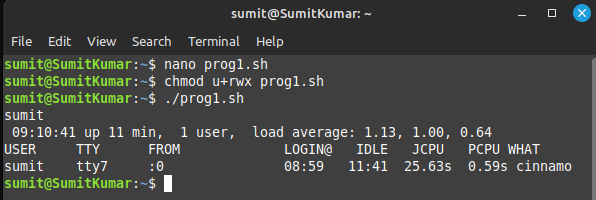
Now, whenever the client machine boots, it will automatically mount the NFS share.

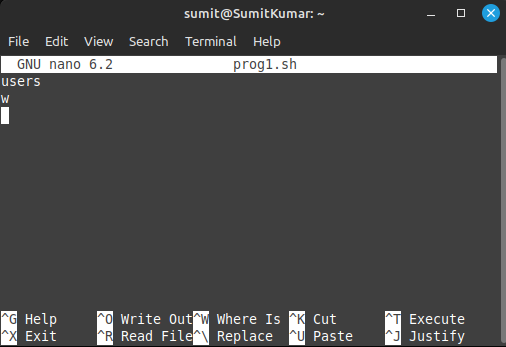
**Lab 6**

**Aim-** Write a shell script to display list of users currently logged in the system.

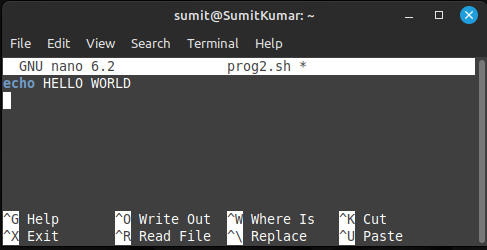
* Write a shell script to display “HELLO WORLD”
* Write a shell script to check whether the given number is even or odd.

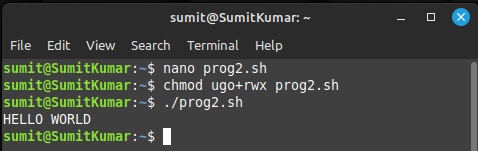
Q1) Write a shell script to display list of users currently logged in the system.



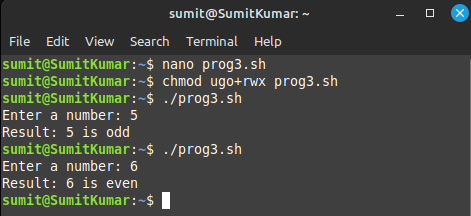


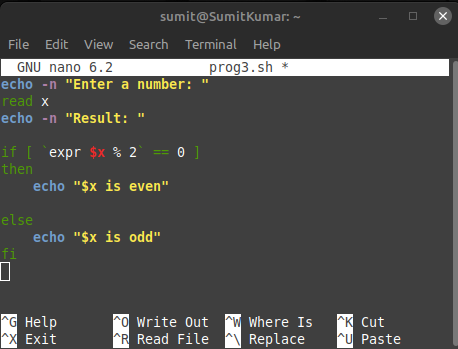
Q2) Write a shell script to display “HELLO WORLD

”



Q3) Write a shell script to check whether the given number is even or odd.





|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Total Marks** | **Marks Obtained** | **Comments** |
| Concept (A) | 2 |  |  |
| Implementation (B) | 2 |  |  |
| Performance (C) | 2 |  |  |
| Total | 6 (to be scaled down to 1) | | |

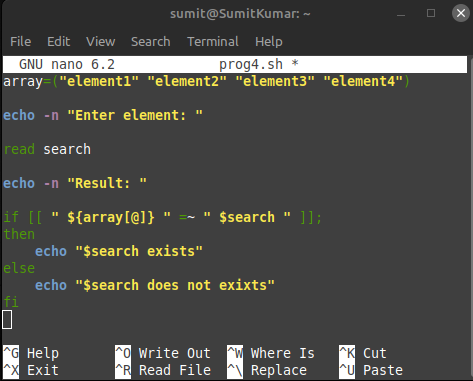
**Lab 7**

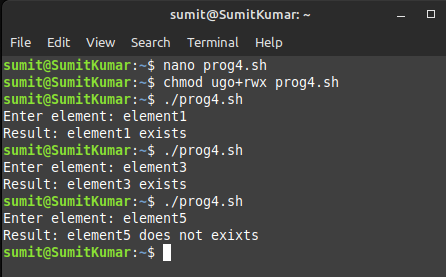
**Aim-** Write a shell script to search whether element is present in the list or not.

Write a shell script to check whether the given file is a directory or not.

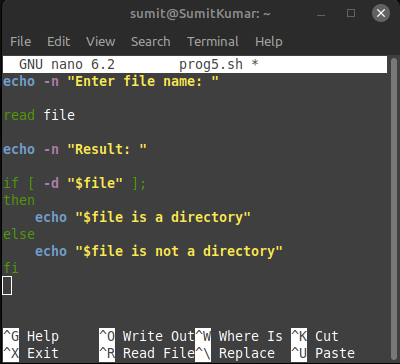
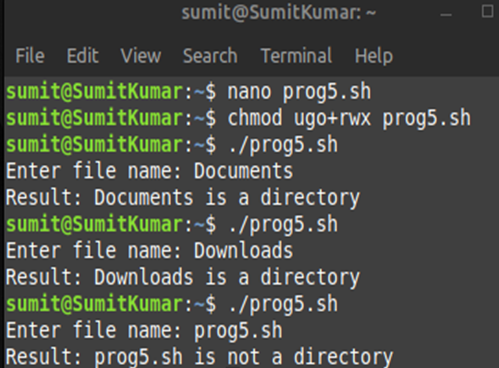
Write a shell script to count the number of files in a directory.

Q1) Write a shell script to search whether element is present in the list or not.

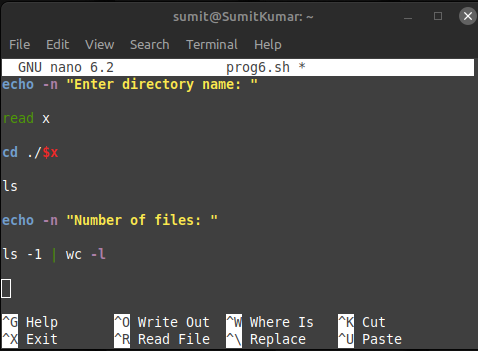


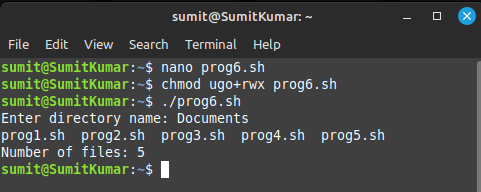


Q2) Write a shell script to check whether the given file is a directory or not.

Q3) Write a shell script to count the number of files in a directory.





|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Total Marks** | **Marks Obtained** | **Comments** |
| Concept (A) | 2 |  |  |
| Implementation (B) | 2 |  |  |
| Performance (C) | 2 |  |  |
| Total | 6 (to be scaled down to 1) | | |

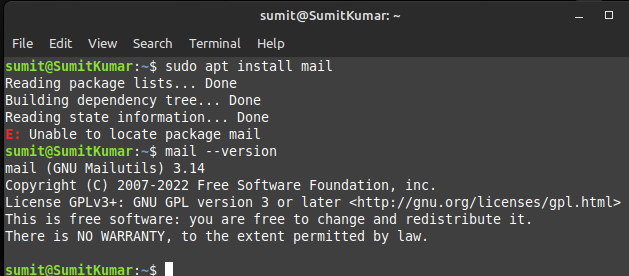
**Lab 8**

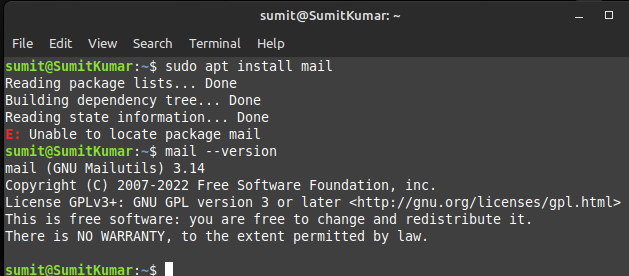
**Aim-** To implement SMTP and send an email using a shell script in Linux.

SMTP stands for Simple Mail Transfer Protocol. It is a standardized communication protocol used for the transmission of electronic mail (email) over the internet. SMTP is primarily responsible for sending, relaying, and routing outgoing email messages from a sender's email client or email server to the recipient's email server.

Message Transfer: SMTP is used to transfer email messages between mail servers. When you send an email, your email client (e.g., Outlook, Thunderbird, or a web-based email service) uses SMTP to communicate with your email provider's SMTP server. The provider's server then routes the message to the recipient's email server.

Step 1: Install mail package.





Step 2: Go to the gmail and in security tab enable the two step verification.

A screenshot of a computer

Description automatically generated

A screen shot of a computer

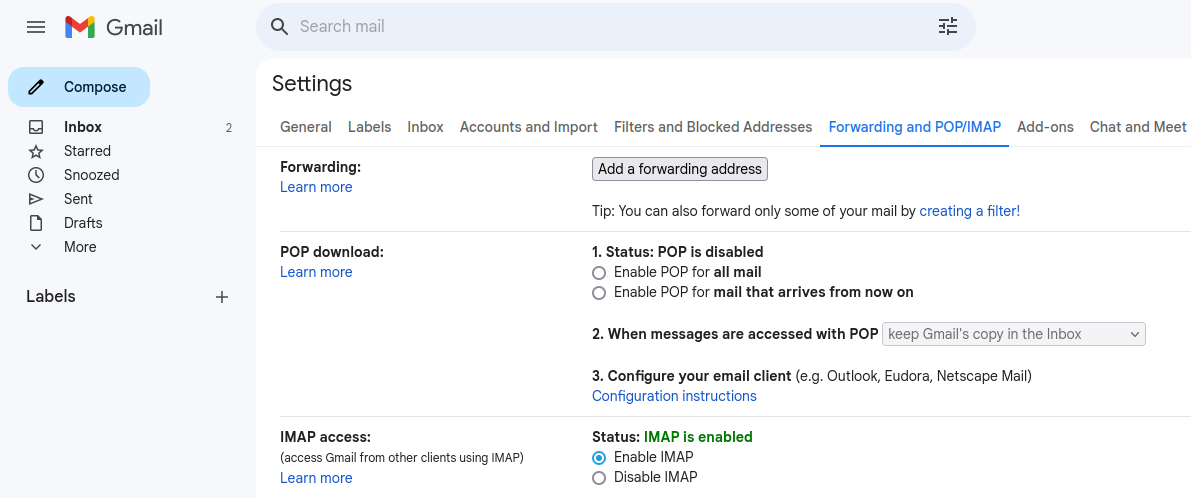
Description automatically generated

Step 3: Setup the app password and copy the 16 letter app password.

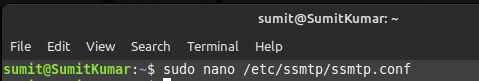
A screenshot of a computer

Description automatically generated

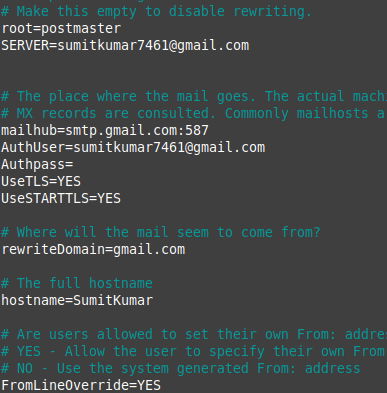
Step 4: Enable IMAP from gmail settings.



Step 5: Modify the ssmtp.conf file.



Step 6: Add the server, mailhub, Authuser, authuser and other details.



Step 7: Write the email with mail command and a body after it press CTRL + D to send the mail.

A screen shot of a computer

Description automatically generated

Step 8: The mail is received.

