Text editor in Linux:

Nano:- Nano is a simple and user-friendly text editor.

- Create a new file or edit an existing one:
- # nano filename.txt
- Save the changes and exit:
- Press Ctrl + O (Write Out) and then Ctrl + X (Exit).
- View the content of a file:
- # nano filename.txt

Vim:- Vim is a powerful and highly configurable text editor.

- Create a new file or edit an existing one:
- # vim filename.txt
- Save the changes and exit:
- In command mode, type :wq and press Enter.
- View the content of a file:
- # vim filename.txt

Emacs:- Emacs is another powerful text editor with extensive features.

- Create a new file or edit an existing one:
- # emacs filename.txt
- Save the changes and exit:
- Press Ctrl + X followed by Ctrl + S to save and Ctrl + X followed by Ctrl + C to exit.
- View the content of a file:
- # emacs filename.txt

Gedit:- Gedit is a lightweight graphical text editor on GNOME desktop.

- Create a new file or edit an existing one:
- # gedit filename.txt
- Save the changes and exit:
- Close the Gedit window, and it will prompt you to save any unsaved changes.
- View the content of a file:
- # gedit filename.txt

Vi:- Vi is a classic text editor present on most Unix-based systems.

- Create a new file or edit an existing one:
- # vi filename.txt
- Save the changes and exit:
- In command mode, type :wq and press Enter.
- View the content of a file:
- # vi filename.txt

Cat:- Cat is not a text editor but a command to concatenate and display file content.

- Create a new file or overwrite an existing one:
- # cat > filename.txt
- After typing the content, press Ctrl + D to save and exit.
- View the content of a file:
- # cat filename.txt

Echo:- Echo can be used to append content to a file.

- Append text to a file:
- # echo "New content" >> filename.txt

pwd (Print/Present Working Directory):-

- This command displays the current working directory, i.e., the directory you are currently in.
- # pwd

ls (List): - The ls command is used to list files and directories in the current directory.

- # ls
- Additional options:
  - ls -l: Long listing format, showing detailed information.
  - ls -a: Include hidden files and directories (those starting with a dot).

cd (Change Directory): The cd command is used to change your current working directory.

- # cd /path/to/directory # Absolute path
- # cd directory_name # Relative path
- # cd .. # Move one directory up (parent directory)
- # cd # Change to the home directory

mkdir (Make Directory):- The mkdir command allows you to create a new directory.

- # mkdir new_directory
- To create multiple directories at once:
  - # mkdir dir1 dir2 dir3

rmdir (Remove Directory):- The rmdir command is used to remove an empty directory. If the directory contains any files or subdirectories, it will not work.

- # rmdir empty_directory
- **rm (Remove):- The rm command is used to remove files and directories.**
- To remove a file:
  - # rm file.txt
- To remove a directory and its contents recursively (use with caution as it permanently deletes files and subdirectories):
  - # rm -r directory_name

mv (Move): - The mv command is used to move or rename directories.

- To rename a directory:
  - # mv old_directory_name new_directory_name
- To move a directory to another location:
  - # mv directory_name /path/to/destination

cp (Copy):- The cp command is used to copy files and directories.

- To copy a file:
- # cp file.txt /path/to/destination
- To copy a directory and its contents recursively:
- # cp -r directory_name /path/to/destination

useradd:

- This command is used to create a new user account.
- # useradd username
- By default, this command creates a user with minimal settings. You may need to set a password and other user information separately.

passwd:

- The passwd command is used to set or change a user's password.
- # passwd username
- You will be prompted to enter and confirm the new password for the specified user.

usermod:

- The usermod command is used to modify existing user account settings.
- Add a user to a supplementary group:
- # usermod -aG groupname username
- Change a user's home directory:
- # usermod -d /path/to/new/home username
- Change a user's login shell:
- # usermod -s /path/to/new/shell username

userdel:

- The userdel command is used to delete a user account.
- # userdel username

groupadd:

- This command is used to create a new group.
- # groupadd groupname

groupmod:

- The groupmod command is used to modify existing group settings.
- # groupmod -n new_groupname old_groupname
- This changes the group's name from old_groupname to new_groupname.

groupdel:

- The groupdel command is used to delete a group.
- # groupdel groupname

id:

- The id command is used to display user and group information for a specified user.
- # id username

who and w:

- Both who and w commands show information about currently logged-in users.
    - # who
    - # w

Group management in linux:

groupadd:

- This command is used to create a new group.
- # groupadd groupname

groupmod:

- The groupmod command is used to modify existing group settings.
- # groupmod -n new_groupname old_groupname
- This changes the group's name from old_groupname to new_groupname.

groupdel:

- The groupdel command is used to delete a group.
- # groupdel groupname

usermod:

- Although the usermod command is primarily used for user management, it also has options to manage a user's group membership.
- Add a user to a supplementary group:
- # usermod -aG groupname username

Remove a user from a group:

- # usermod -G "" username

id:

- The id command is used to display user and group information for a specified user.
- # id username

getent:

- The getent command is used to query the system databases, including group information.
- # getent group groupname

- **APT** (Advanced Package Tool) - Debian, Ubuntu, Linux Mint, and their derivatives.
- **DNF** (Dandified Yum) - Fedora, RHEL 8 and newer, CentOS 8 and newer.
- **YUM** (Yellowdog Updater, Modified)- CentOS 7 and older, RHEL 7 and older (though these distributions are transitioning to DNF).
- **Pacman** - Arch Linux and its derivatives (e.g., Manjaro).
- **Zypper** - openSUSE and SUSE Linux Enterprise.
- **APK** (Alpine Package Keeper) - Alpine Linux.
- **Portage** - Gentoo Linux.
- **Snap** - Ubuntu, Debian, Fedora, Arch Linux, CentOS, and many others.
- **Flatpak** - Ubuntu, Fedora, CentOS, and more. Like Snap, Flatpak is a universal package manager.

Synaptics package in linux:

- Synaptics is not a package manager; it is a touchpad driver commonly used in laptops and other devices to enable touchpad functionality in various operating systems, including Linux.
- In Linux, the Synaptics touchpad driver is often used in conjunction with X.Org or Wayland display servers to provide touchpad support and features.
- The package manager used in some Linux distributions for software management is called "Synaptic" (without the 's' at the end). Synaptic is a graphical front-end for the Advanced Package Tool (APT) and is used primarily in Debian-based distributions, such as Ubuntu and Linux Mint.
- Synaptic allows users to search for, install, update, and remove software packages from their system in a user-friendly way. It provides a graphical interface that simplifies package management compared to using command-line package managers like apt-get or aptitude.

To install Synaptic package manager on a Debian-based system, you can use the following command:

- # apt install synaptic
- # yum install synaptic

Rsync

- Rsync (Remote Sync) is a powerful and widely-used command-line utility for efficiently synchronizing files and directories between local and remote systems. It uses the "rsync algorithm" that only transfers the parts of files that have changed, reducing the amount of data transferred and speeding up synchronization.
- Rsync can work over SSH, and it's commonly used for backup, mirroring, and data migration purposes. It is available on most Linux distributions and can be used on various platforms.
- Example:
    - # rsync [options] source destination
    - # rsync -avz /path/to/local/directory user@remote:/path/to/remote/directory

LFTP:

- lftp is a sophisticated command-line FTP (File Transfer Protocol) client that supports a variety of network protocols like FTP, FTPS, HTTP, HTTPS, SFTP, and more. It provides a rich set of features, including mirroring, parallel file transfers, and background transfers.
- lftp has a user-friendly and scriptable interface, making it a popular choice for automated file transfers and backup tasks.
- Basic Usage:
    - lftp [options] [ftp(s)://]host[:port]
- Example (connecting to an FTP server):
    - lftp ftp.example.com
- Example (connecting to an SFTP server):
    - lftp sftp://user@server

GFTP:

- GFTP is a graphical FTP client that provides a user-friendly and easy-to-use interface for transferring files between a local machine and remote servers using FTP, FTPS, and SFTP protocols. It is primarily used for desktop environments and provides a similar experience to traditional FTP clients with graphical interfaces.
- GFTP offers features like bookmarking, drag-and-drop transfers, and simultaneous multiple connections to facilitate efficient file transfers

FTP (File Transfer Protocol)

FTP (File Transfer Protocol) is a network protocol used to transfer files between a client and a server on a computer network. It allows users to upload and download files from a remote server over the internet or a local network. In Linux, you can interact with FTP servers using various command-line and graphical FTP clients.

FTP:

- # ftp ftp.example.com
- Example commands:
- To download a file from the server:
    - # get filename
- To upload a file to the server:
    - # put filename
- To list files and directories on the server:
    - # ls
- To change the current directory on the server:
    - # cd directory_name

LFTP :- Command-Line FTP

- To start lftp, simply type lftp followed by the FTP server's hostname or IP address:
  - # lftp ftp.example.com
- If the FTP server requires authentication, lftp will prompt you to enter the username and password:
  - # Username: your_username
  - # Password: your_password
    - Enter your FTP account's username and password when prompted.
- Once connected, you can use various lftp commands to interact with the FTP server. Here are some examples:
- To list files and directories on the FTP server's current directory:
  - # ls
- To change the working directory on the FTP server:
  - # cd remote_directory
- To navigate to your local machine's directory:
  - # lcd local_directory
- To upload a file to the FTP server:
  - # put local_file
- To download a file from the FTP server:
  - # get remote_file
- To download an entire directory and its contents:
  - # mirror remote_directory

NGINX is a popular open-source web server and reverse proxy server software that is widely used in Linux and other operating systems. It was developed to address the C10k problem, which refers to efficiently handling a large number of concurrent connections (10,000 or more) while serving web content quickly and efficiently.

Originally released in 2004, NGINX has gained significant popularity due to its high performance, low resource usage, and ability to handle a large number of simultaneous connections efficiently. It is known for its scalability and is often used as a front-end web server or reverse proxy to offload tasks from backend web servers.

Key features of NGINX in Linux include:

1. **High Performance**: NGINX is designed to handle a large number of concurrent connections and deliver content quickly and efficiently, making it suitable for high-traffic websites and applications.
2. **Reverse Proxy**: NGINX can act as a reverse proxy, forwarding client requests to backend servers and caching content, which can improve the performance and load distribution in a web application infrastructure.
3. **Load Balancing**: NGINX can be used as a load balancer to distribute incoming traffic across multiple backend servers, ensuring better resource utilization and redundancy.
4. **SSL/TLS Termination**: NGINX can handle SSL/TLS encryption and decryption, offloading this computationally intensive task from backend servers and improving their performance.
5. **URL Rewriting**: NGINX supports powerful URL rewriting and redirection rules, which are useful for managing complex website structures and SEO purposes.
6. **WebSockets** Support: NGINX has built-in support for WebSockets, allowing real-time communication between clients and servers.
7. **Access Control and Authentication**: NGINX can enforce access control rules and handle authentication to restrict access to specific resources.

NGINX is commonly used alongside application servers like Apache HTTP Server or backend platforms like Node.js, Ruby on Rails, and PHP-FPM. It acts as a reverse proxy to handle incoming client requests and efficiently distribute them to the appropriate backend server based on predefined rules.

NGINX is available as open-source software under the 2-clause BSD license, and its lightweight, efficient architecture has made it a popular choice for serving static and dynamic content on the web. It is widely used by many high-traffic websites, web applications, and web services to ensure reliable and high-performance delivery of content.