Vi Editor:

→ It is the most popular and classic text editor in the Linux

Modes:

(1) vi command mode:

→ The editor opens in command mode and u can move the cursor and cut, copy,paste the text

→ This mode saves the changes you have made to the file

(2) vi insert mode:

→ This mode is for inserting text in the file

→ You can switch to the insert mode from the command mode by passing 'i' on the keyboard

Editing commands:

→ I – insert at cursor

→ a – write after cursor

→ 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

→ 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

→ dw- delete word, 4dw- delete 4 words

→ 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

Copy and Paste commands:

yy	Copies the current line
yw	Copies the current word from the character the lowercase w cursor is on,until the end of the word
p	Puts the copied text after the cursor
P	Puts the yanked text before the cursor

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

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 command mode to exit the editor and save changes to the file

Nano Editor:

→ Nano is a user-friendly and simple text editor
→ Unlike any other command-line editor , it does not have any mode

Editing files:

→ To **create and open** a new file use "$nano new_filename"
→ To **save** a file press "Ctrl+o"

Searching and Replacing:

→ Ctrl+w : Used to search for a text, after pressing type the search term and hit enter.

The cursor will move to the first match

→ Alt+w : To move to the next match
→ Ctrl+\ : If you want to search and replace. Enter the search term and the text to be replaced

Copying, cutting and pasting :

→ Alt+a : To select text, move the cursor to the beginning for the text and hit alt+a. This will set a selection mark
→ Ctrl+ 6 : To cancel the selection
→ Ctrl+k : Will cut the selected text
→ Ctrl+u : To paste the text move the cursor to where you want to put the text and then hit ctrl+u

Saving and Exiting:

→ Ctrl+o : Used to save the changes made to the file. If the file doesn't already exist , it will be created once you save it
→ Ctrl+x : To exit nano

★ To save the file, you must have at write permissions to the file
★ If you are creating a new file, you need to have write permission to the directory where the file is created

GREP COMMANDS:

→ Grep filter searches a file for a particular pattern of characters and displays all lines that contain that pattern

Example: grep [options] pattern [files]

Options Description:

-c	This prints only a count of the lines that match a pattern
-h	Display the matched lines, but do not display the filenames
-i	Ignores , case for matching
-l	Displays list of filenames only
-n	Displays the matched lines and their line numbers
-w	Match whole word
-v	This prints out all the lines that do not match the pattern
-A n	Prints searched line and n lines after the result
-B n	Prints searched line and n lines before the result
-C n	Prints searched line and n lines after before the result

Some commands:

1. $grep -i “unix” file.txt : Case insensitive search. The -i option enables to search for a string case insensitively in the given file. It matches the word like “UNIX”, “Unix”, “unix”

2. $grep –c “unix” file.txt : Displaying the count of number of matches, basically finds the number of lines that matches the given string/pattern

3. $grep -l “unix” f1.txt f2.txt f3.txt : Display the file names that matches the pattern

4. $grep –w “unix” file.txt : Checking for the whole word in a file

5. $grep –o “unix” file.txt: Displaying only the matched pattern. It displays the entire line which has matched the string

6. $grep –v “unix” file.txt: Inverting the pattern match. Display’s the the lines that are not matched with the specified search string

7. $grep “^unix” file.txt: Matching the lines that start with a string. The ^ regular expression pattern specifies the start of a line