**Text Editors**

**VI Editors:-**

There are three editors are available in Linux: Ed, Ex and Vi. The Ed editor is

basically a line editor that means Ed assigns line numbers to the lines in file. An

improved version of Ed is called Ex editor. It understands all the commands of Ed.

The VI text editor. It is screen editor rather than line editor. It shows you as

much of the file as it can fit on the screen. Vi the first full screen editor. It allowed the user to view & edit the entire document at the same time. It creating & editing files. Vi written by Bill Jay when he was a student at university of California.

Vi can handle files that contain text no fancy formatting, no fonts, no embedded

graphics just plain simple text. You can create files, edit file and print them. It cannot do bold face, running headers or footers, italic or all that other fancy. Vi editor improved the next version is “Vim”

Vi is visual editor used to enter and edit text files containing data or documents

or programs. It displays the contents of files on the screen and allows the user to add, insert, delete or change parts of the text.

Example: vi filename

vi bcs

Modes of operation: The vi program has three modes.

**I I a A**

**<ESC> :x,:wq Enter**

**Command Mode:**

In this mode all key pressed by for the user are interpreted to be editor commands. For example, if you hit a ' h ' the cursor is moved one position to the left. In command the keys that are hit not displayed on the screen. You can see change on screen.

**Command mode following:**

1. Deletion

2. Navigation

3. Pattern Search

4. Joining Lines

1**) Deletion*:*** If you want to delete a character, move the cursor to that character

and press ' x ' the character will disappear and the line will readjust to the

change. To erase three characters in a row, press ' x ' three times or press ' 3x '.

x : Delete one character.

nx : Delete n characters. I.e n- Number of characters.

Vi adjust the text so that no gap appears in place of deleted string.

2**) Navigation*:*** In command mode move the cursor by one position to another

position. The horizontal and vertical movements of the cursor.

h or Backspace : Moves cursor left & ctrl h.

l or Space bar : Moves cursor right.

k : Moves cursor up.

j : Moves cursor down.

b : Moves backward to beginning of word.

e : Moves forward to end of word.

w : Moves forward to begin of word.

# : Moves the cursor to the end of the current line.

o or | (pipe ) : Moves to beginning of line

**a) Paging and Scrolling**: You can use the control keys to change the view of

displays text on your window.

[ ctrl + f ] : Scroll the screen forward a full window.

[ ctrl + b ] : Scroll the screen backward a full window.

[ ctrl + d ] : Scroll half page forward.

[ ctrl + u ] : Scroll half page backward.

[ ctrl + l ] : Redraws the screen.

**b) Moving Cursor between lines**: If you want to know line number of your

current cursor position. You can use the command to move the cursor to a

specific line numbers.

G : Moves to end of line.

1G : Moves to beginning of line.

40G : Moves the cursor to line 40 numbers.

**3) Pattern Search**: Line containing a string can also be located by patterning to

string with the ' / 'forward slash. To locate the first occurrence of the string

“linux” simple enter /linux <enter>. When you press the ' / ' is repeat previous

search forward and ' ? ' searches backward.

n : Repeat the last search command.

N : Repeat the search command in the opposite direction.

/ : Search forward.

? : Search backward.

**4) Joining Lines**: The lines joined with ' J ' command.

J : Joins the line immediately below the current line with current line.

**5) Commands to Copy Lines:**

Yy : Yank or anchor on current line (which is to be copied)

Nyy : Yank n number of lines.

P : Paste yanked line after or before cursor position.

**Ex Mode ( Exit )**

You can edit a file using Vi editor. You show save your work by writing the buffer content to disk. You may also need to quit Vi editor. You can save the changes.

These features are adequately handled by Ex mode the essential save and exit

command of this mode.

**a) Saving your work and quit the work.**

:w : Saves file & remains in editing mode.

:x : Saves file & quit editing mode.

:wq : Save file & quit editing mode.

:w : Like “Save As” in Ms-Window.

:q : Quit editing mode when no change are mode to file.

:q! : Quit editing mode but after abandoning changes.

**Input Mode**

This mode permits insertion of new text, editing of existing text or replacement of existing text. Each of these operations can be performed only after changing over from the command mode to insertion mode. The insertion mode also known as Input- Text mode. ' i ' to indicate that you wish to insert the text before the current cursor.

I : Insert text to left of cursor.

A : Append text to right of cursor.

O : Open line above.

I : Insert text at beginning of line.

r ch : Replace single character under cursor with ch.

R : Replace text from cursor to right

S : Replace single character under cursor with any number of character.

S : Replace entire line.

Cw : Change Word.

**Options to Vi**

Vi has a couple of options a useful option [ -r ] helps solves as much of a file as possible after a system crash.

***Ex : vi -r emp***

This does not guarantee complete retrieval but in must cases has between

found to recover quite a bit the file Linux , you can start Vi the normal way and then enter the Ex mode command recover.

Vi can also be invoked with the [ + ] option followed by the line number or a

pattern to position the specified line.

Ex : vi +40 chapter1 // Cursor on 40 th line.

vi + chapter // Cursor on last line.

**Advantages of vi editor**

 Vi is universally available on Linux & Unix systems. It has been around so long

in a stable form that it is essentially bug free.

 Vi has many powerful commands that utilize just the alphanumeric keys -- it

does not require special function keys.

 Vi is a small program that does not require a lot of system memory or CPU

time.

It works very fast, even on large files.

 While vi is not programmable, it has a simple way to let other Linux programs,

such as the sort utility, work on selected portions of your file. This adds the

functionality of all those programs to the editor.

 Vi is completely terminal device independent. It will work with any kind of

terminal. A system file describes the capabilities and control sequences of each

kind of terminal for vi. The entire program needs to know is what type of

terminal you have.

 Almost editors on the GUI libraries and X-Windows. But vi is a shell based

editor.

 Running vi won’t need any external libraries like QT or GTK

 vi will give you the best performance

 You can use vi to edit your files on a remote machine through SSH.

**Vim Editor**

vim is almost a proper superset of vi. Therefore, everything that is in vi is available in vim. Vim adds onto those extended features:

 Vim has been ported to a much wider range of OS's than vi.

 Vim includes support (syntax highlighting, code folding, etc) for several b

Popular programming languages (C/C++, Python, Perl, shell, etc).

 Vim integrates with c scope.

 Vim can be used to edit files using network protocols like SSH and HTTP.

 Vim includes multilevel undo/redo.

 Vim allows the screen to be split for editing multiple files.

 Vim can edit files inside a compressed archive (gzip, zip, tar, etc).

 Vim includes a built in diff for comparing files (vimdiff).

 Vim includes support for plugins, and finer control over config and startup files.

 Vim can be scripted with vimscript, or with an external scripting language (e.g.

python, perl, shell).

 The newer version of vi text editor is called vim (Vi IMproved).

 vim is the standard Linux and UNIX text editor.

 The default text editor is vi, unless explicitly changed by the system

administrator.

**Advantages**

 Speed: Do more with fewer keystrokes

 Simplicity: Not Dependence on Mouse/GUI

 Availability: Included with most UNIX like OSes

**Disadvantages**

 Difficulty: Difficult for beginners.

 Patience and persistence will lead to many great benefits on your way to

becoming a Linux user.

**Archive and File compression commands**

It is useful to store a group of files in one file for easy backup, for transfer to another directory, or for transfer to another computer. It is also useful to compress large files; compressed files take up less disk space and download faster via the Internet.

It is important to understand the distinction between an archive file and a compressed file. An archive file is a collection of files and directories stored in one file. The archive file is not compressed — it uses the same amount of disk space as all the individual files and directories combined. A compressed file is a collection of files and directories that are stored in one file and stored in a way that uses less disk space than all the individual files and directories combined. If disk space is a concern, compress rarely-used files, or place all such files in a single archive file and compress it.

There are two purposes for using file compression:

you can reduce the storage space;

 When transferring files over the network, you can reduce network traffic.

There are many compression tools under Linux:

 compress / uncompress

 zip / unzip

 bzip2/bunzip2

**Gzip & gunzip Command [ Compressing & Decompressing ]**

This command used for a compressing and decompressing the data in a

file. It provides the extension .gz to the compressed filename and removes the original file. The file size is changed after compression.

**Syntax : $ gzip [ filename ]**

***Options* :**

|  |  |
| --- | --- |
| -r | Compresses all file in a subdirectories |
| -d | Uncompressing a gzipped file or gunzip command. |
| -d | Display The Compressed File Content |

***Example* :** $ wc -c info.html

3875302 info.html

file size before use of gzip command large.

$ gzip info.html

file size after use of gzip command is small.

$ wc -c info.html.gz

788096 info.html.gz

$ gzip -l info.html.gz

Compressed uncompressed ratio uncompressed\_name

788096 3875302 79.6% info.html

$ gunzip info.html.gz // uncompress file.

OR

$ gzip -d info.html

$ gzip -r program // compress all files in program

$ gunzip -r program or gzip –dr program // Decompress all files in program

**Zip and Unzip Command [ Compressing & Archiving together ]**

This command also used for compress archive a data in a file. zip require the first argument to be compressed filename; the remaining argument are interpreted

as files and directories to be compressed.

Syntax : $ zip [ compressed filename with .zip extension ] [ filenames ]

Examples :$ zip archive.zip info.html book.txt

adding : info.html ( deflated 80% )

adding : book.txt ( deflated 66% )

$ unzip archive.zip

Archive : archive. zip

inflating : info.html

inflating : book.txt

$ unzip -v archive.zip // [ -v ] option for viewing the archive.

Archive : archive.zip

**Difference Between zip and gzip**

1. GZIP can achieve better compression compared to ZIP.

2. ZIP is capable of archiving and compressing multiple files, while GZIP is only

capable of compression.

3. You can easily extract individual files from a large ZIP file, but not from a GZIP

tarball.

4. ZIP is fairly popular on Windows, while GZIP is more popular on Linux or

UNIXlike operating systems.