CHAPTER 5

Editors

About this Chapter: This chapter introduces the user to the necessity of editors. Various types of editors, their functions and their usage are explained in detail.

Objectives

- Introduction to vi editor
- The ed and sed editors
- The emacs editor

Introduction

We have so far seen the creation of files and usage of file manipulation commands. This session introduces us to the various types of text editors available in LINUX. The editors help in creating and editing files. The editor in an operating system has the same importance as the editor of a newspaper.

LINUX offers various types of editors like vi, ex, sed, ed, etc. We shall see how to use the vi editor. We shall also understand the usage of the editors sed, emacs and ed.

Among the editors, the vi editor stands as the most powerful and widely used in the LINUX environment. vi and ex editors are basically the same. Their commands are mutually compatible with each other. While ex is a line editor, vi is a full screen editor.

Introduction to vi Editor

vi has become the most powerful editor in the LINUX environment due to many of its features. We shall discuss the basics of the vi editor in this section.

vi stands for "visual". Prior to the introduction of vi editor, programmers in LINUX had to use editors which were line editors. They could see only the line they were editing, which caused problems for them. When vi was introduced, it could give the programmer full screen view. vi was created and written at the University of California at Berkeley by Bill Joy.

Basics of the vi Editor

One important aspect of vi is that it does not provide us with the original copy of the file when invoked. It provides us with a "editing buffer". All changes are done to that copy in the buffer. The original file is replaced when we quit vi .

vi functions in three different modes - the command mode, the insert mode and the ex escape mode.

In the command mode, any key pressed by the user is assumed as commands by the editor. No text is displayed on the screen when any key is pressed.

The insert mode helps us to insert the text we want to. This mode can be invoked by using any of the insert, open or append mode command.

The ex escape mode is introduced to make the vi and the ex editors compatible. When any command is typed in this mode it is shown on the command line. The bottom line of the vi screen is called as the command line. Here messages are displayed and all commands entered in the ex escape mode are displayed.

vi emits a beep sounds if there is any error.

Getting Started with vi

The editor can be invoked by typing vi at the $ prompt. vi, when invoked without a filename takes input from the Standard Input. It is always better to invoke vi along with a file name.

```
$ vi  filename
```

The filename followed by "[New file]" appears at the command line at the bottom of the vi screen. The filename vi can also be invoked to suit our requirements. For example, $ vi +3 rose, opens the file called rose and places the cursor on the third line.

Important keys to remember.

Enter key	Executes commands in the ex escape mode and starts a new line in the insert mode. In the command mode it goes to the next line
Esc key	Returns the vi editor to the command mode
"/"	Helps to search for a particular string within the file. The string along the "/" has to be entered on the status line
"?"	Works the same way as "/" key

Press i [insert mode]

Esc : q ↵ [quit without saving]

Esc : q! ↵ [quit withow saving forcasbly]

Esc : wq ↵ [quit & save]

":"	Used when we want to specify a command in the ex escape mode
:s/X/Y/g	Used when we want to replace every occurrence of the word or the character X with Y in the line where the cursor is positioned
:g/X/s//Y/g	Used when we want to replace every occurrence of the word or the character X with Y in each line of the file

Table 5.1

Movement of the Cursor

All vi commands involve alphabets and the shift and control keys. Since vi is fanatically case sensitive, we must make sure that the Caps lock is off. The cursor movements can be described as follows:

h or backspace	left
l or space bar	right
k or -	up
j or +	down

Table 5.2

However, before we use these commands, we must ensure that we are in the command mode. After typing the text, press esc key to enter into the command mode.

Some More Cursor Movements

Let us try to use some more of the cursor movements commands in order to work faster in vi.

w	Moves forward by a word. Assumes that punctuations are next words.
W	Moves forward by a word and this identifies punctuations as a part of the word.
e	Takes us to the last character of the word.
E	Takes us to the last character of the word and ignores any punctuation.
b	Moves backward by a word.

Esc : wq! ↵ [save & quit forciasally]