**Experiment 2:** Basic functionality and modes of VI editor.

**Date:** 23 February, 2022

**Objective:** To write code in c, python and java using ubuntu terminal

**vi, command, and input modes:**

One of the most important aspects to remember about vi is that most of the commands fall into one of three modes:

1. vi mode: in this mode, most keys on the keyboard are defined to be a specific command. As the key or key sequence is issued, that command is executed. This is the mode vi starts in. At any time, pressing the key returns the user to vi mode.

2. command mode: to reach that mode, one must first be in vi mode, then issue a colon (“:”). That same colon will appear at the bottom left corner of the screen. Then the command may be issued following the colon. One exception to this rule is the search command; a forward slash is issued instead of the colon.

3. input mode: this is where most users expect an editor to start. This “mode” actually refers to commands issued from vi mode but that allows the user to start inputting data into the file.

**Invoking vi:**

Type in the command prompt: vi filename

which will put filename into a buffer, and display the file on the screen. If the file is larger than the screen can display, the screen will act as a window into the file. At the beginning of a session, the screen will display the first part of the file. If filename does not exist, vi will create it. Upon entry to vi, the bottom of the screen will print the name of the file being edited, the number of lines in the file, and the size of the file (in characters).

Write two paragraphs (whatever you want) in the file (for testing the various commands).

**Exiting vi:**

Usually, the new or modified file is saved when you leave vi. However, it is also possible to quit vi without saving the file.

Note: The cursor moves to the bottom of the screen whenever a colon (:) is typed. This type of command is completed by hitting the <Return> (or <Enter>) key.

:x <Enter> quit vi, writing out the modified file to file named in the original invocation

:wq<Enter> quit vi, writing out the modified file to file named in the original invocation

:q! <Enter> quit vi even though latest changes have not been saved for this vi call

**Moving the Cursor:**

Unlike many of the PC and Macintosh editors, the mouse does not move the cursor within the vi editor screen (older versions). You must use the key commands listed below. On some UNIX platforms, the arrow keys may be used as well; however, since vi was designed with the Qwerty keyboard (containing no arrow keys) in mind, the arrow keys sometimes produce strange effects in viand should be avoided.

If you go back and forth between a PC environment and a UNIX environment, you may find that this dissimilarity in methods for cursor movement is the most frustrating difference between the two.

In the table below, the symbol ^ before a letter means that the <Ctrl> key should be held down while the letter key is pressed.

Note: Since following are the commands they will not work in the INSERT mode. Just open the file by writing

vi filename

on the command prompt and execute the commands without pressing ‘I’. Before that, make sure that something is written in the file (refer invoking vi).

j or <Enter>

[or down-arrow]

move cursor down one line

k [or up-arrow] move cursor up one line

h or <Backspace> [or left-arrow]

l or <Space>

[or right-arrow]

move cursor left one character

move cursor right one character

0 (zero) move cursor to start of the current line (the one with the cursor)

$ move cursor to the end of the current line

w move cursor to the beginning of next word

b move the cursor back to the beginning of preceding word

:0<Enter> or 1G move the cursor to the first line in the file

:$<Enter> or G move the cursor to the last line in the file

**Screen Manipulation:**

The following commands allow the vi editor screen (or window) to move up or down several lines and to be refreshed.

Note: Since following are the commands they will not work in the INSERT mode. Just open the file by writing

vi filename

on the command prompt and execute the commands without pressing ‘I’. Before that make sure that something is written in the file (refer invoking vi).

^f move forward one screen

^b move backward one screen

**Adding and Deleting Text**

Unlike PC editors, you cannot replace or delete text by highlighting it with the mouse. Instead, use the commands in the following tables.

Perhaps the most important command is the one that allows you to back up and undo your last action. Unfortunately, this command acts like a toggle, undoing and redoing your most recent action. You cannot go back more than one step.

u undo whatever you just did; a simple toggle

The main purpose of an editor is to create, add, or modify text for a file.

**Inserting or Adding Text:**

The following commands allow you to insert and add text. Each of these commands puts the vi editor into insert mode; thus, the <Esc> key must be pressed to terminate the entry of text and to put the vi editor back into command mode.

Note 1: Since following are the commands they will not work in the INSERT mode. Just open the file by writing

vi filename

on the command prompt and execute the commands without pressing ‘i’. Before that make sure that something is written in the file (refer invoking vi).

Note 2: Each of these commands puts the vi editor into insert mode; thus, the <Esc> key must be pressed to terminate the entry of text and to put the vi editor back into command mode.

i insert text before the cursor, until <Esc> hit

I insert text at beginning of current line, until <Esc> hit

a append text after the cursor, until <Esc> hit

A append text to the end of current line, until <Esc> hit

**Deleting Text:**

The following commands allow you to delete text.

X delete single character under the cursor

Nx delete N characters, starting with a character under the cursor

Dw delete the single word beginning with a character under the cursor

dNw delete N words beginning with a character under cursor; e.g., d5w deletes 5 words

D delete the remainder of the line, starting with the current cursor position

Dd delete entire current line

Ndd or dNd delete N lines, beginning with the current line; e.g., 5dd deletes 5 lines

**Cutting and Pasting Text:**

The following commands allow you to copy and paste text.

Yy copy (yank, cut) the current line into the buffer

Nyy or yNy copy (yank, cut) the next N lines, including the current line, into the buffer

P put (paste) the line(s) in the buffer into the text after the current line

**Searching Text:**

A common occurrence in text editing is to replace one word or phrase by another. To locate instances of particular sets of characters (or strings), use the following commands.

/string search forward for the occurrence of a string in the text

?string search backward for the occurrence of a string in the text

n move to next occurrence of the search string

N move to next occurrence of the search string in opposite direction

The rest of the experiments in the list involve shell programming.

**Follow the following steps in each case to execute the programs:**

1. To write the programs, create new files for each program by writing in the command prompt: vi filename

2. Write the program as plain text (in Insert mode)

3. Save the file and exit

4. Run the file by giving following command in the command prompt: sh filename

Sample output:









