Operating System Lab

Experiment 2

Aim: To understand the Vim Editor

Linux has a large number of text editors. However one editor which is available in all distributions as a standard is vi. Now we have the open source variant of vi which is named as Vi Improved to experiment on.

Vim Basics

1. Opening Vim
    - vim ← Open the editor only
    - vim myfile.txt ← Open a file named myfile.txt. If file not existing it is created and opened
    - vim ~/demo/myfile.txt ← Opens or Creates and opens myfile.txt in /home/user/demo/
2. Vim Modes
    - The default mode is command mode
    - to insert text we have to switch to insert mode, this is done by pressing i, this i does not get inserted but all subsequent typing will insert at the cursor location. In the insert mode Vim shows -INSERT- at the bottom of the screen
    - instead we can use a to add text to the character next to the cursor, not at the cursor
    - there is another mode -REPLACE- to be dealt later.
    - to come back to command mode we have to press esc
    - Saving and quitting occurs by command and so we need to come to the command mode
3. Saving a file
    - Saving only :w
    - Save and quit :wq or ZZ
    - Save As :w mynewfile.txt copies the buffer to a new file named mynewfile.txt
4. Quitting
    - Quitting without saving if changes were made :q!
    - Quitting when no change to file has been made :q

Vim Navigation

1. Cursor keys can be used for up down left and right navigation, one character at a time in the document; further,
2. in the command mode
    - k is up
    - j is down
    - h is left (this can be done using backspace too)
    - l is right (this can be done using space bar too)
3. Numeric navigation is done in the command mode by typing like this 4l (four places to left)
4. Home and End
    - 0 to go to the beginning of the line
    - $ to go to the end of the line
    - 4$ to go to the end of the 4th line starting from the current line
    - numeric navigation with 0 won't work similar $
5. Word at a time movement of cursor:-
    - w for forward move with punctuation mark halting
    - W for forward move without any halting

    - b and B for backward move similar to the previous one
    - Numeric navigation is like 5b or 5B
6. Line Navigation
    - 1G to go to the top
    - G to go to the bottom
    - 10G goes to 10th line
    - 10k goes up 10 lines
    - 10j goes down 10 lines
7. Page Navigation
    - ^f for forward navigation
    - ^b for backward navigation

Editing with Vim

1. Change Case by pressing ~ over the characters
2. c for Change text
    - cw deletes till end of the word and gets into insert mode
    - c2w deletes till next two words and gets into insert mode
    - c$ deletes till the end of the line and gets into insert mode C can also be used for that
    - c2$ will delete the current line from cursor to end and following complete line
    - cc deletes entire line and gets into insert mode can also be done with S
    - s for substitute deletes current character and enters insert mode
3. d for Delete works similar to c commands except that it does not change the mode
    - e.g. 2dd deletes 2 lines
    - D deletes from cursor position to end of line
    - x deletes current character, del key can also be used.
    - X works as backspace in command mode
4. r/R for Replace
    - r and any character will replace the current cursor position with the character; there is no change of mode
    - R changes to overtype mode, still enter key is accepted to add a line
5. u for Undo
    - u undoes the previous operation
    - U undoes all edits on a single line atonce
6. y for Copy
    - works similar to c and d but
    - Y works as yy contrary to C and D which starts from current cursor position
7. p for Paste
    - Delete and Change keeps the deleted content in buffer and
    - p pastes it after current cursor
8. . for Repeat
    - previous command like paste and delete can be repeated by pressing dot (.)
9. Some other editing commands
    - A to append text to end of line
    - I to insert text at beginning of line
    - o to create a blank line below cursor
    - O to create a blank line above cursor
    - <number e.g.10>i<text e.g. abc><esc key> will insert abc for 10 times

Operating System Lab

Editor Features

1. To show line number, :set nu

Searching

1. / for searching a pattern
   - /<pattern> e.g. /abc ← will search for a pattern abc from the current cursor position till end of file and repeats from top if the pattern not found. It will halt at the first character of the pattern
   -
2. ? for backward searching e.g. ?abc
3. f for searching in a single line and F for backward search
4. t for searching upto end of the pattern string and T for reverse direction in a single line
5. n for repeating search in forward direction and N to repeat in backward direction
6. ; repeats the search in the line and , repeats it in backward direction
7. y,c and d can be combined e.g. c/abc deletes till abc from cursor and enters insert mode

Find and Replace

1. single line
   - :s/<find pattern>/<replace pattern>
   - :s/<find pattern>/<replace pattern>/g does the replacement in all occurrences in a line
2. multiple line
   - :1,20s/<find pattern>/<replace pattern>g replaces from line 1 to 20
   - :%s/<find pattern>/<replace pattern>g replaces in the entire file

Vim Advanced

1. Opening a file in ReadOnly mode e.g. vi -R demo.txt
2. Opening file at a specific place
   - vi +20 demo.txt (opens demo.txt at line 20)
   - vi +/abc demo.txt (opens at the first occurrence of abc in demo.txt)
   - vi + demo.txt (opens at the last line of demo.txt)
3. Searching and replacing with patterns
   - . matches any single character except newline e.g. a.c matches abc aac axc ...
   - * matches zero or more occurrence of the preceding single character ab* matches a, ab, abb
   - ^ matches beginning of a line e.g. ^ab matches ab placed at beginning of line
   - $ matches end of file e.g. ab$ matches ab placed at end of the line
   - \ is the escape for special characters in search string like . * ^ $
   - [<characters>] match any character placed within paranthesis e.g. [aeiou] match all vowels
   - [<character>-<character>] matches the characters inbetween like [A-Z] or [A-Za-z]
   - word finding \<ab match any word starting with ab and bc\> match any word ending with bc
4. Executing Linux commands within Vim
   - :!ls lists your files
5. Suspending Vim temporarily
   - ctrl+z to suspend vim for the time being and return to the prompt
6. Resuming Vim from suspension
   - at the command prompt type fg ←