Lab2

Objective

1. To learn basic commands of VIM Editor.
2. To learn compiling C file in Linux.

Introduction to VIM Editor

An editor is the program that is used to edit source code. There are many text editors available for Linux, but the two most widely used are Visual Editor Improved (VIM) and Emacs. Here we discuss VIM editor.

1. Creating and opening file

- $ vi <filename> - opens file if it is already exist, otherwise creates new file.
- $ vi - opens VIM editors, without filename.

2. VIM Modes

VIM has many modes, but the two most important are the *Command mode (Norma*l *Mode)*, and the *Insert mode*. In Command mode VIM interprets everything typed as a command to VIM. To insert the text the VIM must be in Insert mode. To handle the file such as saving, quitting from file, VIM must be in Command mode.

Initially VIM will be in command mode.

Adding text to the file

- There are two basic commands to enter to the Insert Mode.
  - a - opens Insert mode and enables to insert text after the cursor.
  - i - opens Inserts mode and enables to inter text before the courser.

- To return to the Normal mode press Esc
- Insert Commands
  - a - Append text after the cursor and enter Insert mode.
  - A - Append text at the end of line and enter insert mode.
  - i - Insert text before the cursor and enter insert mode.
  - I - Insert text at the beginning of the line and enter insert mode.
  - o - Open a line below the cursor and enter the insert mode.
  - O - Open a line above the cursor and enter insert mode.

Moving around in a file (Cursor movement commands).

- h - Move left.
- l - Move right.
- k - Move up.
- j - Move down.

- 7h - Move 7 character to the left.
- 17j - Move 17 lines down.
- 4k - Move 4 lines up.
- 0 - Go to the first character in the line.
- $ - Go to the last character in the line.
- G - Go to the last line in the file.
- nG – Go to line n in the file.
- w - Move to the beginning of the next word.
- 5e – Move to the end of 5th word from the current position.
- Ctrl+F – Scroll forwards (down).
- Ctrl+B – Scroll backwards (up).

Deleting Text

- Must be in Normal mode to delete text using commands.
- Delete Commands
  - x - Deletes one character.
  - dw - Deletes word by word.
  - d$ - Deletes from cursor to the end of the line.
  - dd - Deletes the entire line.
  - nx - Delete n characters, beginning with the character at the cursor
  - n+Delete Key – Same as nx.
  - ndw - Delete n number of words.
  - ndd - Delete n number of lines.

Undo/Redo

- u - undo the last edit.
- U - undoes the all edits in the current line.
- Ctrl+r - redo

Replace text mode

- new text is inserted into the old text into the cursor.
- Replace commands
  - r - replace the character under the cursor.
  - nr - replace the n number of characters.
  - cw - replace the rest of the word.
  - c$ - replace the rest of the line.
  - R - enter Replace mode to make unlimited overwrites (press Esc go back to the Normal mode)

Change, Delete (cut) and yank(copy)

	change	delete	yank
line	cc	dd	yy
Character	ch	dl	yl
Word	cw	dw	yw
Paragraph above	c{	d{	y{
Paragraph below	c}	d}	y}
Sentence back	c(	d(	y(
Sentence forward	c)	d)	y)

Paste

- p - Paste after the cursor.
- P - Paste before the cursor.

Finding Text

Normal mode operation.

- /searchtext - Search the string from the cursor to the end of the file.
- ?searchtext - Search the string from the cursor to the beginning of the file.
- n - Continue search in the same direction.
- N - Continue search in opposite direction.

Save Changes and Exit the Visual Editor normal mode operation.

- :w - write (save) the file.
- :q - quit if saved.
- :wq - save and quit. Or :x or ZZ
- :wq filename - save file with filename.
- :q! - Quit without saving any changes.
- :e! - Abandon the changes and reload the last saved file.

Getting Help

:help - Opens the help window
:help <command> - open help window for the command.
Ctrl+] - jump to the help topic.
:q - quit from help window.

Compiling with GCC

A compiler turns human-readable code into machine readable object code that can actually run. The compilers of choice on Linux systems are all part of the GNU compiler collection, usually known as *gcc.* gcc supports the ANSI standard syntax for C.

Compile, link and run the program.

$ gcc -o <exefile> <sourcefile>
$ ./<exefile> Example:
$ gcc -o hello hello.c
$ ./hello

Problem

Task #1. Practice each and every command mentioned above.

Task #2 . Write the C source file for execute “Hello, World”, and save it as filename 'hello.c' in your directory.