Wei Fei (2538810)
EECS678 Lab1 Report
Prasanth Vivekanandan
02/02/14

1. Open a file with some C source code. Copy the first 12 lines of text from this file. Create three new files named a.c, b.c, and c.c and paste this text at the top of each new file. Save each new file.

Answer:

Step1: Open terminal and go to the file include simple.c;

Step2: Type "**vim simple.c**" to enter vim editor;

Step3: Type "**12yy**" to directly copy the first 12 lines of text from the file, and type "**:q**" to quit;

Step4: Type "**vim a.c**" to create a new source file named "a.c", and then press "**P**" to paste the 12 lines that we copied on step 3. After that, type "**:wq**" to save and quit;

Step5: As similar as step 4, type "**vim b.c**" and "**vim c.c**" to create the other two source files and press "**P**" to paste those 12 lines of codes.

2. Open two different source files for editing, ensuring both are visible on screen simultaneously, and switch between editing each of these two files and issuing commands to a terminal you have open.

Answer:

Step1: Type "**vim simple.c**" to enter vim editor and view codes of the file;

Step2: Type "**:split a.c**" to open another file, and then we can see two screens on the terminal. If we want to split screens vertically, just type "**:vsplit a.c**:".

Step3: Use "**Ctrl + w**" to switch between two screens. If I want to issue commands on the top screen, just switch to top screen and may type "**i**" to insert something or type "**q**" to quit. And for the bottom screen, we can do similar operations as well.

3. As you are reading the code for a large C program (with multiple source files spanned across multiple directories), you come across a call to an unknown function. Find the definition of this function. Go back to the calling context where you started.

Answer:

Step1: Go to the same file as "simple.c", and type "**ctags –R**" to generate a new file named "tags;

Step2: Type "vim simple.c" to enter vim editor and view codes of the file;

Step3: Move mouse to a function that we want to find the definition, and press "**Ctrl +]**" to go to the definition of the target function.

4. Given a file with a million lines of text, remove the whitespace (spaces, tabs, and newlines) from the beginning of every line. That is, when you have finished, each line should start with a non-whitespace character.

Answer:

Step1: Type "**vim simple.c**" to enter vim editor and view codes of the file;

Step2: Type "**:%s/^\s\+**" to remove the white space from the beginning of every line.

5. Find and replace every occurrence of the string 'Bill Self' in your source file with the string 'basketball genius Bill Self' (assume that case matters). After you're done, reformat your file so that each line adheres to an 80 character text width. If you are using vim, you may assume that your vimrc has the appropriate textwidth and formatoptions settings so that lines are formatted correctly when the formatting command is used.

Answer:
Step1: Enter the vim editor to view the file and type "**:%s/Bill Self/basketball genius Bill Self**" to replace all string "Bill Self" to the string "basketball genius Bill Self";
Step2: To reformat the file, type "**:set textwidth=80**" to set an 80 character text width, and then type "**gggqG**" to go through all of the codes in order to reformat the entire file. Or we can just type "**gq**" and press the direction button to select which area we want to change.