INTRO TO VI

vi is one of the fastest and most powerful text editors you can use. No it doesn't implicitly have things like bolding or centering text. It isn't a word processor. It DOES provide everything you need for easy and fast program editing though.

NOTE: These are NOT all the commands vi has, but rather a subset that should get you started using the vi editor successfully and quickly. FOr additional commands, get a good UNIX reference with a chapter on using vi, or the vi editor reference published by O'Reilly And Associates.

STARTING VI

To start vi from a UNIX prompt, simply type:

```
vi filename.cc
```

This will bring up a blank screen with the tilde character in the first column of each row. This means you are now ready to start editing your file.

VI MODES

One of the first things you have to get used to with vi is the fact that it has two specific modes. There is a command mode and an insert mode. The reason this is important is because you can only do certain actions in certain modes. For example, you can not move around your file with the arrow keys unless you are in command mode. Attempting to do so in insert mode will produce odd (undesirable) effects.

You always start out in command mode when first loading a file. Command mode will let you perform searches, perform cut, copy, paste, delete, and several other operations. Insert mode will let you type new data to your file from the keyboard.

SWITCHING BETWEEN MODES

Switching to command mode from insert mode is easy. Just press the key. Switching to insert mode from command mode is a little different and you have several options. To start inserting code at the exact point of the cursor, simply press the 'i' key (for insert). To start adding text to the position immediately after the current cursor position, press 'a' (for append). (The reason this will be useful will become evident.) To begin adding text at the end of the line the cursor is currently on, press 'A', and to begin adding text to the beginning of the line the cursor is currently on, press 'I'. Finally, two lesser used commands are 'o' and 'O'. The 'O' command will put you in insert mode by creating a new (blank) line directly above the current line the cursor is on. Similarly, 'o' creates a new line directly after the current line, and starts inserting at that point.

BASIC WAYS TO ENTER INSERT MODE

i	Enter insert mode at current position
a	Enter insert mode at the next position from cursor
I	Enter insert mode at the beginning of the current line
A	Enter insert mode at the end of the current line
O	Enter insert mode with a new line above the current line
o	Enter insert mode with a new line below the current line
<ESC>	Leave insert mode, return to command mode

SAVING YOUR FILE (AND QUITTING)

In a perfect world, you would type everything in the first time, it would work wonderfully, and you would never edit the file again. In this case, you have learned all the commands you need to know, so this seems like a good place to explain how to save your work, and quit out of the vi editor.

While in command mode, type the ':' key. You should see a colon appear on the very bottom like of your screen. If it appears at the location of the cursor, you are still in insert mode and will not be able to enter commands. Following the colon, you can type 'w' followed by the enter key to save the file with the current file name. To save the file with a different name, just use the command ':w newfilename.cpp' Now the question is: how do you exit so you can return to a prompt and compile your program? Simple. Use the command ':q' followed by enter. If the file hasn't been saved in its current format, you are warned, and vi does NOT exit. To override this (if you don't want to save the changes) just use the command ':q!' instead. Finally, if you just made a change, and you want to save your work and exit to the prompt, there is no need to run these two separate commands. Instead, combine them to perform the following command: ':wq' followed by enter. It will write your file and then exit to a UNIX prompt.

:w	Write (save) file with current filename
:q	Quit the vi editor
:q!	Quit the vi editor WITHOUT saving changes
:wq	Write (save) the file and quit to UNIX prompt

DELETING AND CHANGING

This isn't a perfect world though, and you will need to be able to delete characters from your file, delete lines, and change things as well. To delete a character, move your cursor to the character you want gone, and while in command mode (press escape to be sure) press 'x'. The character will disappear. If you want to delete 5 characters in a row, issue the command '5x'. In fact, you can put any number before the x to remove that number of characters. Deleting a whole line this way would be tedious, so you can use the 'dd' command to delete an entire line at a time. Similarly to the x command, you can issue '7dd' to delete 7 lines of text in a row. Deleting lines character by character would be tedious, and deleting a whole word at a time can be tedious. vi knows this and provides a command to make your life easier. Position your cursor at the beginning of a word. Instead of counting the number of letters (say 7) and issuing the command '7x', you can simple issue the command 'dw' to delete a word at a time.

Finally, lets say you want to change the word 'hello' to 'goodbye'. There is a command to do this without having to delete the word hello, enter insert mode and type out goodbye. Instead, since hello is a 5 letter word, just issue the command '5s'. This will start changing the 5 characters at the location of your cursor. You are automatically placed in insert mode, so you can type the word goodbye and the word hello will be overwritten and changed to goodbye as desired.

x	Delete the character at cursor's current position
nx	Delete n characters at cursor's position
dd	Delete the entire line at cursor's position
ndd	Delete n lines at cursor's position
dw	Delete the entire word at the cursor's location
s	Delete the character at cursor position, and enter insert mode
ns	Change n characters from cursor's position

MOVING AROUND THE DOCUMENT

You can use the arrow keys to get anywhere in your document, but often, this isn't very fast. For instance, to scroll down a full page, you can simply issue the command 'ctrl-f', meaning forward. Similarly, you can go backwards a full page by issuing 'ctrl-b'. What if you only want to go up or down a half page though? 'ctrl-d' with go down a half page, and 'ctrl-u' will go up a half page. To move your cursor forward one full word, issue the command 'w'. To go directly to line number 52 in your file, enter the command ':52'. To move the cursor to the very beginning of the current line, issue the command '0' (thats a zero), and to move the cursor to the end of the current line, issue the command '$'. Note also that if your arrow keys do not work, the standard vi commands for moving the cursor are: 'h' for left, 'l' for right, 'j' for down, and 'k' for up.

Ctrl-f	Move forward one full screen
Ctrl-b	Move backward one full screen
Ctrl-d	Move down one half screen
Ctrl-u	Move up one half screen
w	Move one full word to the right
:n	Go directly to line number n
0	Move to the beginning of the current line
$	Move to the end of the current line
h	Move the cursor left one character
l	Move the cursor right one character
j	Move the cursor down one line
k	Move the cursor up one line

SEARCHING FOR TEXT

Lets say your cursor is in the middle of your document, and you want to go forward to the next instance of the word 'goodbye'. All you have to do is issue the search command as follows: '/goodbye' and press enter. If you want to go backwards to the most recent prior instance of the word goodbye, change the command to '?goodbye' and press enter. But what if you find that it wasn't the next instance of goodbye you were looking for, but the one AFTER that? Do you have to type that whole command out again? Nope. Just issue the command to repeat the same exact search 'n'.

/text	Will move to the next instance of the text following the /
?text	Will move to the prior instance of the text following the ?
n	Perform the exact same search that was just performed