

starts with a / (or slash). A path can lead to directories physically located on other computers. The network system makes this transparent to the user. Any path which begins at your home directory starts with ~ (or tilde and slash). Any path which begins with ~username (or tilde username) begins in that particular user's home directory.

Examples

/banana/consumers/monkeys/

would refer to the **monkeys** subdirectory of the **consumers** subdirectory of the **banana** subdirectory of the root directory.

~/cs102/lab7/painter.p

would refer to the **painter.p** file of the **lab7** subdirectory of the **cs102** subdirectory of your home directory.

~/brubble/advice/friends/FRED/bowling

would refer to the **bowling** file or subdirectory of the **FRED** subdirectory of the **friends** subdirectory of the **advice** subdirectory of **brubble**'s home directory.

1.6 Final Note on Unix

This operating system will become more comfortable to you the more you use it. Do not get discouraged. If you have questions, ask a friend or a computer lab assistant. Once you have learned UNIX, you will be glad you did. UNIX is a powerful operating system, and since it is a portable operating system, you may find UNIX everywhere.

2 vi

vi is a visual editor available on the UNIX system. Please remember when using commands that vi is case-sensitive, so whether a letter is upper- or lower-case DOES matter.

2.1 Two Modes of the vi Editor

Before you begin working with the vi editor, you should know that there are two modes in vi - the **command** mode and the **insert** mode.

Command mode is used to move through text, search for words, save a file, etc. This mode covers everything except inserting code. **Insert** mode is only for inserting text into a file.

Using **i**, **I**, **a**, **A**, **cw**, **o**, **O**, or **R** will place you into **insert** mode from **command** mode. These commands will be described further in Section 2.2.2 of this guide.

IMPORTANT!! Use the **escape** key to return to **command** mode from **insert** mode. If you are not sure which mode you are in, hit the **escape** key to be sure to be in **command** mode again.

If the keys you are pressing are not achieving the results they should, you may be in the wrong mode, or you may have accidentally pressed the **Caps Lock** key. The little green light on the **Caps Lock** key should not be lit.

Sometimes you may hit **escape** to switch to **command** mode, and find that the words **INSERT MODE** still appear at the bottom left corner of the window. Hit **escape** again to be sure you are actually in **command** mode. If the words still appear in the lower left corner of the window, you may want to “refresh” the screen by pressing the **control** and **l** keys at the same time. (That is the lower-case letter “l” in case you were not certain.) Now, if you are not in **insert** mode, the words stating otherwise will disappear from your screen.

IMPORTANT!! Do NOT use the arrow keys to move around your text while in **insert** mode!! Strange things will happen!! And, if you use an arrow to “back up” along one line of text while in **insert** mode, the text is still showing on the screen even though moving backwards through the text with the arrow key has technically deleted everything up to where you “backed” the cursor. When you get out of **insert** mode you lose text you thought was there. This can be frustrating! In time, you will learn to **escape** from **insert** mode FIRST, before moving around in your file to make any necessary corrections.

2.2 Useful vi Commands

2.2.1 Moving Around Within Text in Command Mode

←, ↓, ↑, and →	Move through text: left, down, up, and right
<Control> f	Move forward one screen
<Control> b	Move backward one screen
<Control> d	Move half a screen down
<Control> u	Move half a screen up
G	Move to end of file
1G	Move to beginning of file
65G	Move to line 65 in the file
\$	Move to end of line
0	Move to beginning of line (To be clear, the command is the ‘zero’ character.)
/myword	Moves forward to an occurrence of myword, or states Pattern not found
?myword	Moves backwards to myword, or states Pattern not found
n	Moves to next occurrence of myword in the same direction as the search
N	Moves to next occurrence of myword in opposite direction as the search

2.2.2 Inserting Text

All of these commands will put you in **insert** or **append** mode. Enter the text and then press **escape** to get back into **command** mode.

i	insert text before cursor
a	insert text after cursor (append it)
I	insert text at beginning of current line
A	insert text after end of current line
o	open a line below current line and insert text
O	open a line above current line and insert text
<ESCAPE>	to switch back to command mode from insert mode

2.2.3 Deleting Text

x	delete one character
dw	delete one word
10dw	delete 10 words
dd	delete one line
25dd	delete 25 lines
D	delete from cursor to the end of that line

Note: all the commands above store the deleted text in a temporary buffer. The contents of the buffer can be retrieved back into the file using the **P** and **p** commands described in the “moving text” section below.

2.2.4 Moving Text

yy	yanks one line into a buffer
	without deleting that line from the file
3yy	yanks three lines into a buffer
	without deleting those lines from the file
p	puts line(s) from buffer into file after the current line
P	puts line(s) from buffer into file before the current line
J	joins the line below the current line with the current line

You would use the **yy** and **p** commands when wanting to **copy** text from one part of the file to another. You would use the **dd** and **p** commands when you want to actually **move** text from one part of the file to another.

2.2.5 Changing Text

r	replace a character by moving to the character you wish to replace. Then, type r and the character you want to replace the old character with. (This new character can even be a space!)
R	replaces everything until you press <ESCAPE> again
cw	change a word by moving to the word, or portion of word, you wish to change. Type cw and begin typing in the new word, or portion of word. You will see a \$ floating at the end of the word you are changing to remind you that you are in change mode. You must press the escape key when you are finished typing the word(s). Note: you CAN replace a single word with several words.

2.2.6 Saving and Quitting

:w	writes the file (saves it) while remaining in the file
:w new_file_name	saves the current file under a new name
:wq	writes the file and quits the vi session, closing the file
:q	quit when you have NOT changed a file
:q!	really quit - without saving ANY changes made to the file (Be careful with this command. You might use it if you accidentally opened the wrong file and made changes, which you then realized you did not wish to keep.)

You should use the **:w** command often to save your file when you are working on it. This will help protect you in the event of a system failure. This method will also allow you to save code in one window before compiling the file in another window. If there is an error, you can make changes and save with the **:w** command in the code window. Then you can compile again in the other window.

Also, you will generally use **:wq** when you wish to end your **vi** session, saving all changes you made to the file.

2.3 Error Recovery

u	is the undo command which will undo the last step you did
U	restores current line to state in which you most recently started editing it (When in doubt, press escape to be certain you are in command mode. Make sure the Caps Lock is not on if strange things are happening.?)
<CONTROL> q	escape the "no-key-does-anything stuck-state."
<CONTROL> l	remove the insert or append words in the lower corner of the screen if you are now in command mode instead.
:q	to quit a file you opened and do NOT wish to change, or to close a file you did not mean to create (and into which you did not insert text).
:q!	to quit a file you unintentionally created, and into which you inserted text. (Again, be careful with this command!)

Note: certain keystrokes can cause the colon to appear in the lower left corner of the window. If this happens, and pressing **return** does not put you back into **command** mode, entering **vi** should work.

If you wish to view a file you do NOT wish to change, you may use the **view** command instead of **vi**.

2.4 Final Note on vi

The best way to learn anything, including vi, is PRACTICE, PRACTICE, PRACTICE!! So you may try creating some text files to practice using these new commands and become familiar with the two modes and how to switch from one to another. vi does get easier after a while, once you become familiar with it - believe it or not.

3 X-Windows

Please note: This discussion applies only to the environment created for J.I.C.S. users on the University of Tennessee Computer Science Department. This will dictate the manner in which the windows and other items are situated on your screen. This setup will also determine how the mouse buttons will react in different instances. We present this here as an example of how an X windowing environment would be used. For details about the configuration of the X windows at other sites, consult the local computer systems administrator.

X is an environment which allows great flexibility through the use of **windows**. For example, one could be remotely logged on one or more machines in different windows while