

CHAPTER 7

vi

Objectives:

- Understand why you use vi instead of other text editors.
- Have a working knowledge of the 6 basic vi commands: i, a, o, r, x, dd

About vi

vi is pronounced v. i. (not 6 or veye). vi is the standard text editor available on most unix systems. There are other text editors that are much easier to learn to use, but they have features that make them undesirable to use in some cases. A big problem is line wrapping, where the editor will automatically put a hard return in a line for you when it reaches the end. In some cases, where a command is longer than one line, if a hard return were put in the end of the code line would be ignored.

vi uses simple character commands to complete various tasks. We will cover the basic commands needed to get around in vi.

Starting a vi session

In an xterm, you can start or open a new file by typing `vi filename`. This opens the vi editor and creates that new file for you. If a file already exists, the command is the same, vi and the name of the file.

```
machine~% vi testing
```

vi modes

There are two modes in vi, command and edit. You can be in command mode and move around the document freely using the various ways to move, but if you were in edit mode, the characters you type to move around normally may end up as text in your document.

Command mode:

As mentioned, vi uses simple character commands to accomplish things. You can ensure that you are in command mode at any time by hitting escape (Esc). This takes you out of any edit mode and into command mode and will not hurt anything if you are already in command mode and hit escape to be sure.

Edit mode:

You enter edit mode by giving one of the editing commands, such as a, i, o, etc.

- a
 - appends or adds immediately after the cursor where as A adds at the end of the current line.
- i
 - inserts immediately after the cursor where as I inserts at the beginning of the line.
- o
 - opens a line immediately below the current line and O opens a line above the current line.

Removing Text:

Here are some basic commands for removing text. :

- x
 - deletes a single character
- dd
 - deletes an entire line

Replacing or correcting text:

You may need to go back and change a single character or replace a few words. That can also be done simply by using vi's simple character commands.

- r
 - replaces a single character and R replaces until you tell it to stop.
- cw
 - change word and will mark the end of the word with a \$ and allow you to type until you have finished your corrections and hit escape.
- ct
 - allows you to change the text to a specific spot, then stops where as C changes to the end of the line.

Moving around:

To move around in vi you can get out of edit mode and use the arrows, but that would require you to take your hands off home row. The creators of vi also included commands on moving around.

- 0
 - (zero) means the beginning of the line
- \$
 - means the end.
- h j k l
 - move you left, down, up and right, respectively
- w or W
 - moves one word at a time forward and
- b or B
 - moves you one word at a time backwards.
- : (Colon)
 - then a number moves you to that line in the document
- G
 - takes you to the bottom.

Combining commands:

Let's now look at ways to combine these commands and do cool things that will save you time. For an example, use the sentence "The quick brown fox jumped over the lazy dogs". The cursor placement is shown with the `_`.

If you want to change fox to wolf you could use `ct` ("change to" then press space), or `R` to replace all characters until you tell it to stop, but in this case you are going from 3 characters to 4 so this would not be the best.

The quick brown fox jumped over the lazy dog.

If you wanted to change to the end of the sentence you could `R` or `C` to change to the end of the line or `d$` which would delete to the end of the line. Then you would have to type new text there. Most commands can be combined, so play with it and have fun.

The quick brown fox jumped over the lazy dogs

Search and Replace:

You will need to search and replace in vi. You do this in command mode. Here is how you do it:

- `:s/old/new/`
 - replaces the first occurrence of *old* with *new* on the current line.
- `:s/old/new/g`

- add g (global) to replace all occurrences of *old* with *new* on the current line.
- `:%s/old/new/g`
 - add % to replace all occurrences of *old* with *new* in the entire file.
- `:50,100s/old/new/g`
 - replace all occurrences of *old* with *new* from line 50 to 100

vi Command References

More Information about vi can me found here:

<http://www.thomer.com/thomer/vi/vi.html>

<http://unix.t-a-y-l-o-r.com/Vi.html>

vi Reference Table

^ = Control

Basic Commands	Mnemonic	Description
vi filename	opens	Opens file specified, if no file by that name, then creates a blank file.
vi +10 filename	opens	Opens with cursor at line 10
View	read only	Opens a read only version of the file
:w filename	writes	Saves to specified filename
:wq	write quit	Save and exit the program
Esc	escape	Reset to command mode
:e filename	edit	Load another file for editing
:r filename	read	Brings in another file below cursor
:w!	hard save	Looks at Permissions again incase they have changed since the file was opened.
:q!	Hard quit	Quits with out saving changes or asking about saving changes.
Cursor Movement		
0	Zero	Beginning of current line
\$	end	End of current line
h, j, k, l	left, jump, klmb, right	Moves the cursor
(arrow keys)		Cursor moves accordingly in command mode
(space)		One character move accordingly
w	word	One word to the right
W	word	One word right
b	back	One word to the left or backwards
B	back	One word to the left or backwards
:0	zero	To the top of the file
G	goto	to the bottom of the file
^f	forward	One page forward
^b	Backwards	One page backward
^u	up	Half a page up
^d	down	Half a page down
/	search	Search forward (n for next)
?	search	Search backward (n for next)
Adding Text		
a	add	Adds text immediately after cursor
A	add	Add text at end of line
i	insert	Insert text immediately after cursor

I	insert	Insert text at beginning of line
o	open	Open line below cursor
O	open	Open a new line above cursor
Removing / Retrieving text		
x	cross-cut	Delete single character under cursor
dd	delete	Delete current line
d0	delete zero	Delete to the beginning of the line
d\$	delete dollar	Delete to the end of the line
dw	delete w	delete word
yy	yank	Copy current line to memory
yw	yank word	Copy word
p	put	Put buffer after cursor
P	put	Put buffer before cursor
.	(dot)	Repeat last command
Modify text		
r	replace	Replace single character under cursor
R	replace	Replace multiple characters
s	substitute	Substitute multiple characters
S	substitute	Substitute entire line
cw	change word	Change word, starting with current character
ct	clear to	Clear from current character to given character
C	change	Change to end of line
>>	indent	Move line one tabstop in
<<	outdent	Move line one tab stop out
u	undo	Toggles undo and redo
^j	join	Join next line to here
Numeric Prefixes		
10dd		Deletes 10 lines starting at cursor
10x		Deletes 10 characters starting under cursor
5rQ		Replaces the next 5 characters with 5 letter Qs
5sQ		Replaces the next 5 characters with a single Q
5Rabc		Replaces the next 15 characters with repeating abc
30Blahla		Add 3 new lines containing "Blahla" below cursor

10/02/03