

Running vi

To run vi and create a new file, simply run the command 'vi' from any shell prompt:

```
$ vi
```

Alternatively, to load an existing text file into vi, run the command 'vi [filename]', from the shell prompt:

```
$ vi myfile.txt
```

Your file will be loaded into vi, and the cursor will be placed at the beginning of the first line. Note that an empty line is shown as a tilde (~).

Command Mode and Insert Mode

vi is different from many editors, in that it has two main modes of operation: command mode, and insert mode. This is the cause of much of the confusion when a new user is learning vi, but it is actually very simple to understand.

When you first load the editor, you will be placed into command mode. To switch into insert mode, simply press the 'i' key. Although nothing will change on the screen to indicate the new mode, any thing that you type from now on will appear in the screen - this is what you are used to if you have ever used any other editor, or word processor. Try typing a few lines of text. When you press 'return' or 'enter' a new line will be created, and you may continue typing.

When you have finished typing, you may return to command mode. This is done by pressing your 'Esc' key. In command mode, key presses do not appear on the screen, but instead are used to indicate various commands to vi.

At first, you may often mistake command mode and insert mode. For example, you may think you are in insert mode, and start typing your text, when in fact you are in command mode, and each keypress you make will issue a command to vi. Be careful - you may accidentally modify or delete parts of your file.

If you are unsure which mode you are in, press 'Esc'. If you were in insert mode, you will be returned to command mode. If you were already in command mode, you will be left in command mode (possibly with a 'beep', to indicate that you were already in command mode).

An Exercise

Try this simple exercise to get used to the two modes:

1. Load vi (if you haven't already), by typing:

```
$ vi
```

(don't forget - the \$ is the system's prompt, and may be different on your system. You only type the 'vi' part, shown in bold.)

The screen should show a blank file, with each blank line represented by a tilde (~), for example:

```
~  
~
```

By default you are in command mode.

2. Switch to insert mode, by pressing the 'i' key. Then type some text, using ENTER or RETURN to start new lines. For example:

```
Hello.  This is my first session in the vi editor.
This is the second line of text.
~
~
~
```

3. When you have finished entering your sample text, press 'Esc' to return to command mode.

4. Now we will learn a useful command: to save the file. The command for this is ':w' (note the colon before the 'w'). After the 'w', put a space, and the name you want to store the file as. For example:

```
:w firstfile
```

Type it now. Notice how the text ':w' appears at the bottom on the screen. When you have finished the command, press ENTER or RETURN. You should see a confirmation that the file has been saved, which may include the number of lines in the file, and possibly the file size.

5. To finish this simple introduction to the editor, we will learn one final command: How to close the editor. The command for this is ':q' (again, with a colon before the 'q'). Don't forget you need to be in command mode. If you're not already in command mode, or you're not sure which mode you're in, press 'Esc' now. Then issue the command:

```
:q
```

You should be returned to the UNIX prompt.

Sometimes, you may have made changes to a file that you do not want to save. This may be because you have decided the changes are incorrect, or you have become confused using vi (not usual at first!), and incorrectly made some changes, maybe by typing into command mode instead of insert mode. To exit vi without saving, and ignoring any warnings about unsaved data, use a variation of the ':q' command, with an exclamation mark after it:

```
:q!
```

This will return you to the prompt, without saving any changes to the file, and with no warnings about unsaved data. Use this command carefully.

A Simple (but useful) Electronic Diary

As an exercise, create an electronic diary, that you can use to store your schedule, one week at a time. This will use many of the skills you have learned so far, and will let you see how you already know enough to use UNIX to help with every day tasks. Follow these steps:

1. Use vi to create a file named 'weekly-diary-template'. The file should look something like this:

```
My Schedule for Week Commencing:
Day          AM                      PM                      Evening
-----
Mon
Tue
Wed
Thur
Fri
Sat
Sun
```

2. Use the 'cp' command to make a new copy of the file, which should be named after the date of the Monday in the current week, for example '12-October-2009'. The command may be something like:

```
$ cp weekly-diary-template diary-16-Nov-2015
```

You should now have two files: the template, and the new copy of the template, which is named after the current week.

3. Use vi to fill in your schedule. Only make changes to the version of the file including the date. Leave the template as it is, ready to create next week's diary when you need it.

4. Whenever you need to start a new week's diary, use a similar 'cp' command. Again, copy the template file to a new file containing the date of the Monday in that week. For example, the following week, you may use:

```
$ cp weekly-diary-template diary-16-Nov-2015
```

Start using the electronic diary on a regular basis. This will give you good practice in vi, and basic file management commands.

Vi Tips

	Topic	Hints
1	The vi operating modes.	Command mode. The vi editor opens in command mode awaiting a command. Press the ESC key to enter the command mode at any other time. Text input mode. Enter text input mode by pressing i , a or o commands (see below) External mode. This mode is used to access Unix system commands whilst still in the editor.
2	How to get into and out of the editor.	vi filename - Invoke the text editor to edit a file view filename - View (read-only) a text file. :q! Abort the edit session without saving. :w Write the file to disk and stay in the editor. :wq Write the file to disk and quit the editor.
3	Cursor and display control	Ctrl/e - Scroll forward 1 line (Ctrl/Y for backwards) Ctrl/d - Scroll forward 1/2 screen (Ctrl/U for backwards) Ctrl/f - Scroll forward 1 screen (Ctrl/B for backwards) Use j , k , h , l keys to move cursor up, down, left, right Use H , M , L to move cursor to top (home), middle or lower position of current screen.
4	How to insert and delete text.	i - Insert new text until ESC is pressed (to return back to command mode). a - Append new text until ESC is pressed. o - Open a new line after the current line and add new text until ESC is pressed. x,X - delete characters dd - delete current line
5	How to undo your mistakes.	u - Undo the most recent text change. U - Undo all of the changes made to the current line. Repaint the screen by pressing Ctrl/L then Ctrl/R . This is sometimes useful if you have got the screen in an unusual state.

Examples of use of metacharacters. When used in the vi editor press the ESC key then the colon key before typing in the metacharacters. Note how the search strings are placed between forward slashes. If you look at the examples above you can imagine how complicated a search can get. The examples below will give you a flavour of what can be achieved, if you want to do anything more complicated, it is quite acceptable to proceed with a book on your desk or assisted by the man pages or a website.

	Example	Explanation
1	/Monday/	This positions the cursor on the first occurrence of the word Monday .
2	/^Monday/	This positions the cursor on the first line that starts with the word Monday.
3	/Monday\$/	This positions the cursor on the first line that ends with the word Monday.
4	/Monday/p	Perform a search for the word Monday and print out any line containing that word.
5	10,20	Perform a search restricted between lines 10 and 20 for the

Getting in and out of the vi text editor

Command	Description
:q	Abort an editing session. The current editing command must be finished before the abort will work.
:q!	Force an immediate abort - no matter what start the current session is in.
:w	Write the file to disk.
:wq	Write the file to disk then exit the editor.

Cursor movement

Command	Description. Move...
h	one space left
j	one line down
k	one line up
l	one space right
^	to beginning of current line
\$	to end of current line
Enter	to beginning of first word on next line
1G	to the first visible character in the first line of the file
G	to the first visible character in the last line of the file
nG	to the first visible character in the nth line of the file
CTRL/G	Display info on the current position of the cursor.
:n	line n
w	beginning of next word
e	end of next word
CTRL/E	Forward 1 line
CTRL/Y	Backward 1 line
CTRL/D	Forward 1/2 screen
CTRL/U	Backward 1/2 screen
CTRL/F	Forward 1 screen
CTRL/B	Backward 1 screen
	Type n before the command for multiple steps (n is an integer)

Inserting Text

Command	Description
a	Append to the right of the cursor.
A	Append at the end of the current line.
i	Insert starting at the left of the cursor position.
I	Insert at the beginning of the current line.
o	Open a line below the cursor, then enter insert mode
O	Open a line above the cursor, then enter insert mode
:r <i>newfile</i>	Add the contents of the file <i>newfile</i> starting below the current line.

Deleting Text

Command	Description
x	Delete a single character at the current cursor position (ccp). (4x deletes 4 characters.)
X	Delete single character to the left of ccp. (4X deletes 4 characters to the left of ccp).
dw	Delete word. (4dw deletes 4 words)
dd	Delete line. (4dd deletes 4 lines)
cw	Delete word (leaving you in insert mode, i.e. change word)
cc	Delete line (leaving you in insert mode, i.e. change line)
s	Change character - delete character and start insert mode
D	Delete from cursor to end of line
dG	Delete all lines, including current line to end of file.
d1G	Delete all lines, including current line to beginning of file.
dnG	Delete all lines, including current line to line n in the file.
J	Join the current line with the line that follows.