Linux System Administration

August 4, 2014

Contents

1 Editing

- Commands preceded with "$" imply that you should execute the command as a general user - not as root.
- Commands preceded with "#" imply that you should be working as root.
- Commands with more specific command lines (e.g. "rtrX>" or "mysql>") imply that you are executing commands on remote equipment, or within another program.

2 vi Cheat Sheet

2.1 Open and edit a file

```
vi filename             (fn=filename)
vi -r filename          Recover a file from a crashed session
vi + filename           Place the cursor on last line of file.
vi +n filename          Place the cursor on line "n" of file.
vi +/pat filename       Place cursor on line with first occurrence of "pat"tern
```

2.2 Close

```
:w                      Write the file to disk. Don't exit.
:w!                     Write the file to disk even if read/only.
:wq                     Write the file to disk and exit.
:wq!                    Write the file to disk even if read/only and quit.
:q                      Quit the file (only if no changes).
:q!                     Quite the file even if changes.
```

2.3 Movement

```
A                       Move to end of line, change to insert mode.
h                       Move 1 space backwards (back/left arrow).
j                       Move down 1 line (down arrow).
k                       Move up 1 line (up arrow).
l                       Move 1 space forwards (forward/right arrow)
w                       Move cursor to start of next word.
W                       Same as "w".
b                       Move cursor to start of previous word.
B                       Same as "b".
:n                      Go to line number "n" in the file.
```

2.4 Editing text

```
i                       Enter in to input mode.
o                       Add a line below cursor and enter in to input mode.
x                       Delete character (del key in some cases).
D                       Delete line from right of cursor to end of line.
dd                      Delete entire line.
u                       Undo last edit or restore current line.
p                       Put yanked text before the cursor.
yy                      Yank current line.
```

2.5 Searching

```
/pattern              Search for "pattern" in the file going forwards.
?pattern              Search for "pattern" in the file going backwards.
n                     Find the next occurrence of pattern found forwards.
N                     Find next occurrence of pattern found backwards.
```

2.6 Copy/Cut and Paste

```
nyyp                  Copy n lines to buffer, paste below cursor
nyyP                  Copy n lines to buffer, paste above cursor
nddp                  Cut n lines and copy to buffer, paste below cursor
nddP                  Cut n lines and copy to buffer, paste above cursor
```

3 Exercises

3.1 Practice using vi

Remember The vi editor uses “modes”

The easiest thing to do if you get confused in vi is to press the ESCape key a couple of times and start over with what you were doing.

To begin log in or use a terminal with your sysadm user already logged in and do:

```
$ cd
$ vi temp.txt
```

vi will create the file “temp.txt” for you.

3.1.1 Inserting text

Press the "i" key to switch to input mode.

Type something like:

```
VI is great! I think I'll be using vi from now on instead of Word

I like VI!
```

3.1.2 Save the file

To save the file, do:

- Press the ESCape key for command mode
- Type ":wq" to save and quit the file (notice the ":" before the "wq"!).
- Remember to press (or) after entering a command

3.1.3 Edit an existing file

Copy a large file to your home directory so that you can play around with some more vi commands. We'll copy over the file `/usr/share/aptitude/COPYING` for this exercise. To do this do:

```
$ cd
$ cp /usr/share/aptitude/COPYING .
```

This makes a copy of the file `/usr/share/aptitude/COPYING` into your home directory, under the same name.

Edit the file, but let's start at the bottom of the file:

```
$ vi + sysctl.conf.bak
```

Go to the first line of the file. Notice the colon (":") before the "1".

```
:1
```

press

Go to line 10, add a new line, and add in some text:

```
:10
```

press

Press the “o” key

Now add the following text:

```
##
## A sample comment
##
```

3.1.4 Deleting text

To delete the three lines you just created:

- Move to the first line of new text
- Press the ESCape key
- Press “dd” to delete a line, repeat until the text is gone

Save the file, but don’t exit.

```
:w
```

press

3.1.5 Practice copying and pasting text.

Go to line 12, copy 9 lines of text, go to the bottom of the file, place the text there:

```
<ESC>       (go to command mode)
:12         (go to line 12 of the file)
9yy         ("yank" 9 lines of text and place in copy buffer)
G           (go to the end of the file)
p           (place the contents of the copy buffer here)
```

If want to undo this you would type (in command mode):

```
u
```

3.1.6 Search and replace

Go to the top of the file, replace all occurrences of "GNU" with "COW", but prompt for each change:

```
<ESC>
:1
:%s/GNU/COW/gc
```

Answer "y" (yes) or "n" (no) to a few prompts then escape from this mode by pressing CTRL-C.

Go to line 1, search for "kernel", move to the end of the line, add some text:

```
<ESC>
:1
/kernel
SHIFT-A
"some text"
<ESC>
```

Now let's exit from the file and not save the few changes we've made.

```
:q!
```

press

3.2 Practice, Practice, Practice!

As you should be able to see vi is extremely powerful as an editor, but not necessarily intuitive. The best way to get good at using vi is to practice.

Make sure you are logged in as sysadm, then do the following:

```
$ cd
$ vi COPYING
```

In this file practice some of the following elements:

- Moving around
  - By word
  - End of line
  - Start of line

  - Top of file
  - Bottom of file

  - To an absolute line number

- Copying and pasting multiple lines (use vi commands)
- Copying and pasting single lines (use vi commands)
- Copying and pasting multiple lines (use your mouse buffer)
- Copying and pasting single lines (use your mouse buffer)
- Search for items backwards and forwards
- Replacing text

And, anything else you wish to practice.

3.3 Repeat the exercises above using *another* editor

Use joe, nano, jed, etc.