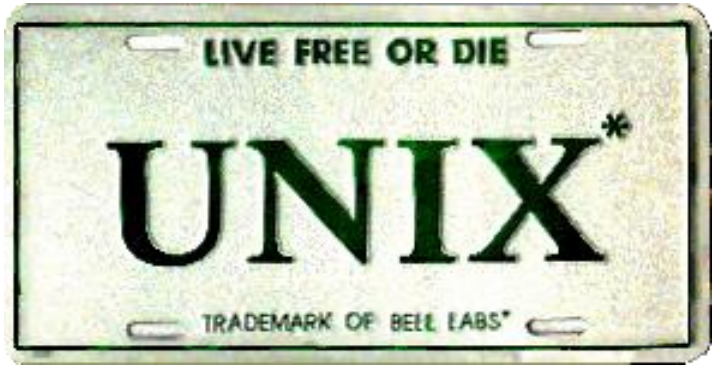

UNIX for Programmers and Users

“UNIX for Programmers and Users”
Third Edition, Prentice-Hall, GRAHAM GLASS, KING ABLES

Editing Files with vi

Editing Files with vi

- The two most popular UNIX text editors are called vi and emacs.
- It’s handy to be reasonably proficient in vi, as it is found on nearly every version of UNIX.
- The vi editor was originally developed for BSD UNIX by Bill Joy of the University of California at Berkeley.
- This is a standard utility for System V and most other versions of UNIX.
- “vi” stands for “visual editor”.
- “vi” wasn't freely available until 2002.
  So, several free **vi**-like editors such as **Vim** (**vi** improved) emerged.
- Functionally, **vim** is almost a proper superset of **vi**. Therefore, everything that is in vi is available in vim.

Editing Files with vi

In Ubuntu, Vim is the only vi-like editor installed by default, and so vi actually starts Vim by default.

Some of the extended vim features:

- Vim has been ported to a much wider range of OS's than vi.
- Vim includes support (syntax highlighting, etc) for several popular programming languages (C/C++, Python, Perl, shell, etc).
- Vim can be used to edit files using network protocols like SSH and HTTP.
- Vim includes multilevel undo/redo.
- Vim can edit files inside a compressed archive (gzip, zip, tar, etc).
- Vim includes a built in diff for comparing files (vimdiff).

Editing a File: vi

To edit an existing file, supply the name of the file as a command-line parameter.

$ vi poem.txt

```
Line 1
Line 2
Line 3
Line 4
Line 5
I always remember standing in the rains,
On a cold and damp September,
Brown Autumn leaves were falling softly to the ground,
Like the dreams of a life as they slide away.
~
~
~
~
"poem.txt" [noeol] 4L, 176C                                  1,1           All
```

Text Entry Mode in vi

- To enter text-entry mode from command mode, press one of the keys in the table below.
- Each key enters you into text-entry mode in a slightly different way:

Key	Action
i	Text is inserted in front of the cursor.
I	Text is inserted at the beginning of the current line.
a	Text is added after the cursor.
A	Text is added to the end of the current line.
o	Text is added after the current line.
O	Text is inserted before the current line.
R	Text is replaced (overwritten).
:	Enter an extended command.

vi: Command Mode

- To edit text, you must enter text-entry mode.
- To transfer from text-entry mode to command mode, press the Esc key.
- If you accidentally press the Esc key when in command mode, nothing bad happens.
- A lot of editing features require parameters and are accessed by pressing the colon (:) key, followed by the command sequence, followed by the Enter key.

vi: Command Mode

- Example, delete lines one through three:

  :1,3d<Enter>

- vi allows you to use the “$” to denote the line number of the last line in the file and the “.” to denote the line number of the line currently containing the cursor.

  Example, delete the current line and the two lines that follow it:

  :.,.+2d<Enter>

vi: Line Ranges

- Here are some other examples of commands for line ranges:

Range	Selects
1,$	all of the lines in the file
1,.	all of the lines from the start of the file to the current line, inclusive
.,$	all of the lines from the current line to the end of the file, inclusive
.-2	the single line that’s two lines before the current line

vi: Comman Editing Features

Common vi editing features:

- cursor movement
- deleting text
- replacing text
- pasting text
- searching text
- search/replacing text
- saving/loading files
- miscellaneous (including how to quit vi)

vi: Cursor Movements

- Here is a table of the common cursor-movement commands:

Movement	Key sequence
Up one line	Arrow Up or the “k” key
Down one line	Arrow Down or the “j” key
Right one character	Arrow Right or the “l” key (will not wrap around)
Left one character	Arrow Left or the “h” key (will not wrap around)
To start of line	^
To end of line	$
Back one word	the “b” key
Forward one word	the “w” key
Down a half screen	Control-d
Forward one screen	Control-f
Up a half screen	Control-u
Back one screen	Control-b
To line nn	:nn<Enter>

vi: Deleting Text

Here is a table of the common text-deletion commands:

Item to delete	Key sequence
Character	Position the cursor over the character and the press “x” key.
Word	Position the cursor at the start of word and then type the two character “dw” .
Line	Position the cursor anywhere on the line and then type the two characters “dd”
Current position to end of current line	Press the “D” key.
Block of lines	:<range>d<Enter>

vi: Replacing Text

Following is a table of the common text-replacement commands:

Item to replace	Key sequence
Character replacement	Position the cursor over the character, press “r” key, and then type the character.
Word	Position the cursor at start of word, type the two characters “cw”, type the replacement text, and press the Esc key.
Line	Position the cursor anywhere on the line, type the two characters “cc”, type the replacement text, and press the Esc key.

vi: Pasting Text

Here is a table of the most common pasting operations:

Action	Key sequence
Copy(yank) lines into paste buffer.	:<range>y<Enter>
Insert (put) paste buffer after current line.	p or :pu<Enter> (contents of paste buffer unchanged)
Insert paste buffer after line nn	:nnpu<Enter> (contents of paste buffer unchanged)

- Example, copy the first two lines into the paste buffer and then paste them after the third line:

  :1,2y
  :3pu

vi: Searching

- Here is a table of the most common search operations:

Action	Key Sequence
Search forward from current position for string sss.	/sss/ <Enter>
Search backward from current position for string sss.	?sss? <Enter>
Repeat last search.	n
Repeat last search in the opposite direction	N

- Example, search for the substring “ark” in line one:

  :1<Enter>
  /ark/<Enter>

vi: Searching/Replacing

- You may perform global search-and-replace operations by using the following commands:

Action	Key Sequence
Replace the first occurrence of sss on each line with ttt.	:<range>s/sss/ttt/<Enter>
Replace every occurrence of sss on each line with ttt (global replace).	:<range>s/sss/ttt/g<Enter>

vi: Searching/Replacing

- Example, replace every occurrence of the substring “re” with “ug1”:

I UG1member walking in the rain,
On a cold and dark September,
Brown Autumn leaves weUG1 falling softly to the ground,
Just like the dUG1ams of a life as they slip away.
~

:1,$s/re/UG1/g

vi: Saving/Loading Files

- Here is a table of the most common save/load file commands:

Action	Key Sequence
Save file as <name>.	:w <name><Enter>
Save file with current name.	:w <Enter>
Forced save into a file that exists.	:w! <name><Enter>
Save only certain lines to another file.	:<range>w <name><Enter>
Edit file <name> instead of current file.	:e <name><Enter>

vi: Saving Files

- Example, save the poem in a file called “rain.doc”:

```
I  remember walking in the rain,
On a cold and dark September,
Brown Autumn leaves were falling softly to the ground,
Just like the dreams of a life as they slip away.
~
```

:w rain.doc

vi: Miscellaneous Commands

- Common miscellaneous commands:

Action	Key Sequence
Redraw screen.	Control-L
Execute command in a subshell and then return to vi.	:!<command><Enter>
Execute command in a subshell and read its output into the edit buffer at the current position.	:r!<command><Enter>
Quit vi if work is saved.	:q<Enter>
Quit vi and discard unsaved work.	:q!<Enter>

Editing Files with emacs

Editing Files with Emacs

- Emacs (Editor MACroS) had its start in Artificial Intelligence community, but evolved into one of the most popular – and powerful – editors in Unix. Richard Stallman and Guy Steele wrote the first version and originated the Free Software Foundation movement.
- Emacs is not as “standard” as vi, but many Unix systems will have it.
- There are also GUI versions of Emacs (e.g., Xemacs).
- To start Emacs type the following:
- $ emacs