Learning "vi" Editing Commands

How to Learn the "vi" Editor

Simple "vi" ( oxymoron )

"vi" as a religion

© by Stan Reichardt

St. Louis Unix Users Group
INTRODUCTION:

- The choice of editor is critical to the new user.
- The learning of an editor is a major point of failure for users new to UNIX, UNIX-Like, and GNU/Linux OSs.

- The "vi" editor is standard - it's everywhere (almost).
- You will (eventually) need to know it.

GOALS:

- Quick Overview of "vi" & "vi" like editors
- Learn Simple "vi"
  - least number of commands necessary
  - don't fail
  - Eventually learn more "vi"
Definitions

- **font** - assortment of printing type of one size and style
- **line editor** - text editor that deals with one line at a time
- **text file** - has lines of text - no binary controls or data
- **text processing** - editing with embedded commands to format documents
- **vedit** - on some systems, a simplified setup of "vi" for beginning users
- **vi** - the visual mode of the "ex" editor
- **view** - "vi" run as a read-only file viewer
- **visual editor** - uses whole screen to display lines
- **word processing** - program to edit text, pictures & formatting codes in one file
- **wysiwyg (whizy-wig)** - What You See Is What You Get
Background

- 1969 ARPAnet born & UNIX by Ken Thompson at Bell Labs
- Jan 1974 Berkely gets AT&T version 4 of UNIX for nearly free
- 1975 Bill Joy and Chuck Haley arrive as graduate students
- they build popular *Pascal* compiler & "ex" line-at-a-time editor
- early 1977 BSD UNIX distributed with *Pascal* & "ex" editor {30}
- ADM-3a terminals with screen-addressable cursors arrive
- Bill Joy enhances "ex" to be "vi" the VInual screen editor
- Bill Joy also writes basis for "termcap" monitor interpreter
- expanded user community feedback enhances system & tools
- 1978 2BSD UNIX distributed with *Pascal*, "vi" & "termcap" {75}
CONs - DISADVANTAGES

- The <CAPS LOCK> key is your enemy!!!!!
- Hard to learn everything
- Mode confusing to new users
- Less than intuitive
- Mouse assistance NOT universal
- NOT for brain dead & too lazy to learn crowd
PROs - ADVANTAGES

● Fingers never leave keyboard area
  ● NOT mouse dependent
  ● NOT PFkey dependent
● Fast
● Powerful
● Content Oriented
● Learned keystokes are common in other UNIX applications
● TAGS (advanced feature)
Available:

- On Virtually Every UNIX/UNIX-like platform
- As Clones for other O/Ss
  - MKS toolkit "vi" - commercial collection of tools
  - ELviS - "vi" clone
  - Sterling Software "vi" - commercial M$DOS clone
  - Stevie - predecessor to VIM
  - VIM for GNU/Linux distributions - M$DOS/Windows
Need to Know

**GETTING IN:**
- invoked as "vi" for editing
- invoked as "view" for read only viewing

**GETTING OUT:**
- `ZZ` - Write (save) any changes and quit file
- `:q` - Quit file (discarding any changes)
- `:w` - to write out your changes to a file
- `:w filename` - write out to a new file name
- `:w!` - Write (save) file (over-riding protection)
More to Know

● COMMAND STRUCTURE:
  ● Editing commands can take most movement commands as objects.

    Examples: dw deletes word and y$ yanks to end of line into a buffer.

  ● Most vi commands can be preceded by a count.

    Examples: 5x erases five characters, 5i? inserts five "?" characters and 2dw deletes two words.

● CONFIGURATION CONTROL:
  ● Invoked by using the ":" {shift colon} key
  ● :set all - - to show all ( or almost all ) options
  ● :help set - - to get more information { VIM }
Least Commands Necessary

http://www.lpi.org - Objective 1.7.1:
Perform basic file editing ops using vi

- basic modes
- navigation
- inserting
- editing & deleting text
- finding text
- copying text
BASIC MODES

- Mode confusing to new users
  - command mode - keystrokes perform editing functions
  - insert mode - keystrokes are inserted into file
- NOT all implementations of vi show current mode on screen
- <ESC> - - key to go into command mode

TIPS TO HELP

- The <CAPS LOCK> key is your enemy!!!!!
- <ESC> - key to go into command mode, i.e. exit insert mode and return control to you
- u,U - undo last edit, restore current line
- <Ctrl>-L - redraw the screen
THE BASIC "vi" COMMANDS

- \textit{h,j,k,l} - left, down, up, right \{avoid arrow keys\}
- \textit{i,a} - Insert text before, after cursor
- \textit{r} - replace with the next typed character
- \textit{x} - delete character
- \textit{dd} - delete current line
- \textit{/pattern} - search forward for \textbf{pattern}
- \textit{yy} - yank (copy) current line into a buffer
- \textit{p,P} - put deleted (or yanked) text before, after cursor
More on navigation

- \(h,j,k,l\) - left, down, up, right \{ avoid arrow keys \}
- \(0,\$\) - first, last position of current line
- \(w,W,b,B\) - Forward, backward by word
- \(e,E\) - end of word
- \(H,M,L\) - jump to Home, Middle, Low screen line
- \(<\text{Ctrl}>F\) - move forward a screen
- \(<\text{Ctrl}>B\) - move backward a screen
- \(<\text{Ctrl}>E\) - shift display one line up
- \(<\text{Ctrl}>Y\) - shift display one line down

More on inserting

- \(i,a\) - Insert text before, after cursor
- \(l,A\) - Insert text at beginning, end of line
- \(o,O\) - Open new line below, above cursor
More on editing & deleting text

- \textit{r} - replace with the next typed character
- \textit{R} - replace until \textit{<ESC>} (overtype mode)
- \textit{cw} - change word until \textit{<ESC>}
- \textit{dw} - delete a word
- \textit{4dw} - delete 4 words
- \textit{dd} - delete current line
- \textit{3dd} - delete 3 lines
- \textit{D} - delete to end of line
- \textit{x} - delete character
- \textit{6x} - delete 6 characters
- \textit{xp} - delete a single character, then put back (to swap two letters)
- \textit{~} - swap case of character
More on finding text

- `/pattern` - search forward for `pattern`
- `?pattern` - search backward for `pattern`
- `n,N` - Repeat last search in same, opposite direction
- `fx` - search forward for character `x` in current line
- `Fx` - search backward for character `x` in current line

More on copying text

- `yy` - yank (copy) current line into a buffer
- `yw` - yank (copy) a word into a buffer
- `p,P` - put deleted (or yanked) text before, after cursor
ALTERNATIVE EDITORS

- **VIM** - *Vi IMproved* - "vi" clone on many GNU/Linux distributions, Unix, M$-DOS, Win3.x/95/98/NT/OS2/Mac/Atari/Amiga/VMS

- **elvis** - "vi" clone
- **vile** - "vi" clone
- **emacs** (the other religion)
- **pico**
- **aee** { *Another Easy Editor* }
- and about 30 others…
Resources

BOOKS:
- Almost every introductory UNIX book
- "Learning the VI Editor" by Linda Lamb - O'Reilly Book +
- "Rescued by UNIX" by Augie Hansen - 7 lessons - Jamsa Press

OTHER:
- man vi -- Read The Friendly Manual (RTFM)
- VIM has :help available internally
  - VIM ":help tutor" to find location of tutor.vi file
  - tutor.vi - text file used to teach "vi" in 25-30 minutes
- www.vim.org - VIM HOME PAGE
- www.thomer.com/thomer/vi/vi.html - VI LOVERS HOME PAGE
- www.networkcomputing.com/unixworld/tutorial/009/009.html - advanced "vi" tutorial (for power users)
- www.freshmeat.net - points to most UNIX+ editor packages
- www.linuxapps.com - points to GNU/Linux editor packages
SUMMARY:

● Recap the key points
● Suggestions and observations
● Questions, comments and other feedback on these materials to: stan@mail.sluug.org