Advanced Programming in the UNIX Environment

Week 05, Segment 2: Unix Development Tools: The Editor

Department of Computer Science
Stevens Institute of Technology

Jan Schaumann
jschauma@stevens.edu
https://stevens.netmeister.org/631/
Software Development Tools

The UNIX Userland is an IDE – essential tools that follow the paradigm of “Do one thing, and do it right” can be combined.

The most important tools are:

• $EDITOR
• the compiler toolchain
• gdb(1) – debugging your code
• make(1) – project build management, maintain program dependencies
• diff(1) and patch(1) – report and apply differences between files
• cvs(1), svn(1), git(1) etc. – revision control, distributed project management
$EDITOR

Know your $EDITOR. Core functionality:

- syntax highlighting
- efficient keyboard maneuvering
- setting markers, using buffers
- copy, yank, fold e.g. blocks
- search and replace
- window splitting
- autocompletion
- jump to definition / manual page
- applying external commands and filters

Partial https://xkcd.com/378/
Examples: `vim`

Efficient keyboard maneuvering:
- up, down, left, right (`h, j, k, l`)

https://dave.cheney.net/2017/08/21/the-here-is-key

Jan Schaumann

2021-09-03
$EDITOR Examples: vim

Efficient keyboard maneuvering:

- up, down, left, right (h, j, k, l)
- move by word, go to end (w, b, e)
- search forward, backward, move to beginning or end of line (/, ?, ^, $)
- page up or down (^D, ^B)
- center page, top or bottom (zz, zt, zb)
- move to matching brace, move to beginning/end of code block (%], [{}]
- move through multiple files (:n, :prev, :rew)
$EDITOR Examples: vim

Efficient keyboard maneuvering:

• up, down, left, right (h, j, k, l)
• move by word, go to end (w, b, e)
• search forward, backward, move to beginning or end of line (/, ?, ^, $)
• page up or down (^D, ^B)
• center page, top or bottom (zz, zt, zb)
• move to matching brace, move to beginning/end of code block (%], [)
• move through multiple files (:n, :prev, :rew)

Jan Schaumann
Examples: vim

Copy, yank, fold, markers, buffers etc.:

- set and display markers (m[a-zA-Z], :marks)
- select visual blocks (v, V)
- format / indent selected block (=)
- delete, yank, use of buffers (d, y, ”xy, ”xp)
- fold sections (zf, zA)
- autocomplete (^N)

Jan Schaumann
$EDITOR Examples: vim

Integration with compiler, debugger, make(1) etc.:

• build your project (:make)
• show / go to error (:cc [N])
• jump to next / previous error (:cn, :cp)
• list errors (:cl, :copen)
vi / vim graphical cheat sheet

Esc

-normal
-mode

~ toggle case

! external filter

@ play macro

# prev ident

$ eol

% goto match

^ "soft" bol

& repeat

* next ident

begin sentence

end sentence

"hard" bol

end line

0 "hard" bol

+ next line

= auto line

W next WORD

E end WORD

R replace mode

T 'till

Y yank line

U undo

I insert at bol

O open above

P paste before

begin parag.

} end parag.

A append at eol

S submit line

delete line

D delete char

F "back" find char

G go to line

j join lines

k help

l scroll top

m set mark

n prev (find)

p next (find)

q quit

Z extra ends

X back space

C change to eol

V visual lines

B prev word

N prev(find)

M screens mid/1

< un-indent

> indent

? find (rev.)

repeat cmd

/ find

Main command line commands ('ex'): 

:w (save), :q (quit), :q! (quit w/o saving)

:e f (open file f),

t!$/x/y/g (replace 'x' by 'y' filewide),

'h (help in vim), :s (new file in vim).

Other important commands:
CTRL-R: redo (vim),
CTRL-E-/B: page up/down,
CTRL-E/-Y: scroll line up/down,
CTRL-V: block-visual mode (vim only)

Visual mode:
Move around and type operator to act on selected region (vim only)

Notes:
(1) use 'x before a yank/paste/del command to use that register ('clipboard') (x=a..z, *)
(e.g. "a$y to copy rest of line to reg 'a")

(2) type in a number before any action to repeat it that number of times
(e.g. 2p, 3w, 5i, 4d)

(3) duplicate operator to act on current line (dd = delete line, >> = indent line)

(4) ZZ to save & quit, ZZ to quit w/o saving

(5) zi: scroll cursor to top, zb: bottom, zz: center

(6) gg: top of file (vim only), gi: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

https://duckduckgo.com/?q=vim+tutorial
Other powerful Unix IDE integrations:

- code index and definitions via e.g., ctags(1)
- a terminal multiplexer (e.g. screen(1) or tmux(1))
- copious use of Ctrl+Z (i.e., the shell’s job control mechanisms)

See our tool tips for more details:

- [https://youtu.be/TWog5NklSws](https://youtu.be/TWog5NklSws)
- [https://youtu.be/vxTXXaCr4s8](https://youtu.be/vxTXXaCr4s8)