Linux Systems and Open Source Software

VIM and tmux

Chia-Heng Tu
Dept. of Computer Science and Information Engineering
National Cheng Kung University
Fall 2022

Outline

- Vim
  - History
  - Usage
  - Customization
  - Plugins
- tmux
  - Usage
  - Configuration

VIM

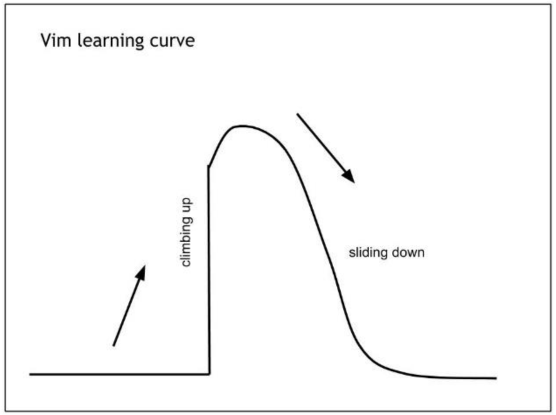
What is Text Editor?

- A text editor is a type of computer program that edits *plain text*
  - E.g., VIM、Sublime Text、Notepad++、Gedit、Emacs、Nano、 Atom
- **Plain text** vs. **Rich text**
- Plain text, created and edited by text editors, exclusively consists of character representation
  - Each character is represented by a fixed-length sequence of one, two, or four bytes, or as a variable-length sequence of one to four bytes, in accordance to specific character encoding conventions, such as ASCII, ISO/IEC 2022, UTF-8, or Unicode
- Rich text, such as that created by word processors or desktop publishing software, may contain metadata in addition to plain texts
  - such as, character formatting data (e.g. typeface, size, weight and style), paragraph formatting data (e.g. indentation, alignment, letter and word distribution, and space between lines or other paragraphs), and page specification data (e.g. size, margin and reading direction)
  - Rich text can be saved in binary format (e.g. DOC), text files adhering to a markup language (e.g. RTF or HTML), or in a hybrid form of both (e.g. Office Open XML)
  - Source code editor within the Integrated Development Environment (IDE) E.g., Code::Blocks、Dev-C++、Eclipse、NetBeans、Xcode、Microsoft Visual Studio Code

Courtesy of https://en.wikipedia.org/wiki/Text_editor

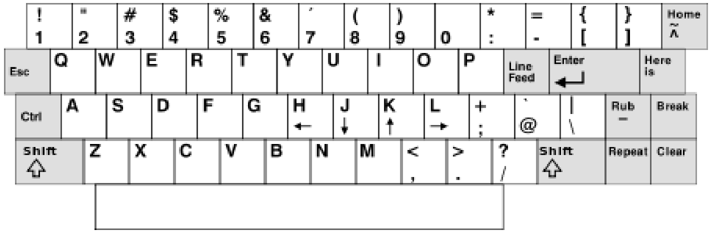
VIM (Vi IMproved)

- *Vim the Six Billion Dollar editor
  - Better, Stronger, Faster
- Customization
- Various plugins
- Text-based user interfaces (TUI)
  - Develop without GUI (e.g., remote to the server)



Quotes from *https://yannesposito.com/Scratch/en/blog/Learn-Vim-Progressively/
User:D0ktorz - reworked in SVG, GPL, https://commons.wikimedia.org/w/index.php?curid=1228427
https://pascalprecht.github.io/posts/why-i-use-vim

History of VIM



! 1 " 2 # 3 $ 4 % 5 & 6 ' 7 (8) 9 0 * : = - { [}] Home ^
Esc Q W E R T Y U I O P Line Feed Enter Here is
Ctrl A S D F G H ← J ↓ K ↑ L → + ; ` @ | \ Rub Break
Shift Z X C V B N M < , > . ? / Shift Repeat Clear

ADM3A鍵盤排法

- vi by Bill Joy in 1976
- **Vi IM**itation 1.0, by Bram Moolenaar in 1988
- **Vi IM**proved 2.0, 1993
  - 1994 3.0 Multiple windows (視窗分割)
  - 1996 4.0 GUI (gVim)
  - 1998 5.0 Syntax highlighting
  - 2001 6.0 Folding (程式碼折疊), plugins (外掛程式)
  - 2005 6.4 Perl, Python, and Ruby support
  - 2006 7.0 Spell checking (拼字檢查)、code completion (程式碼補全)
  - 2016 8.0 Asynchronous (非同步) I/O, jobs, lambdas

Variants from VIM

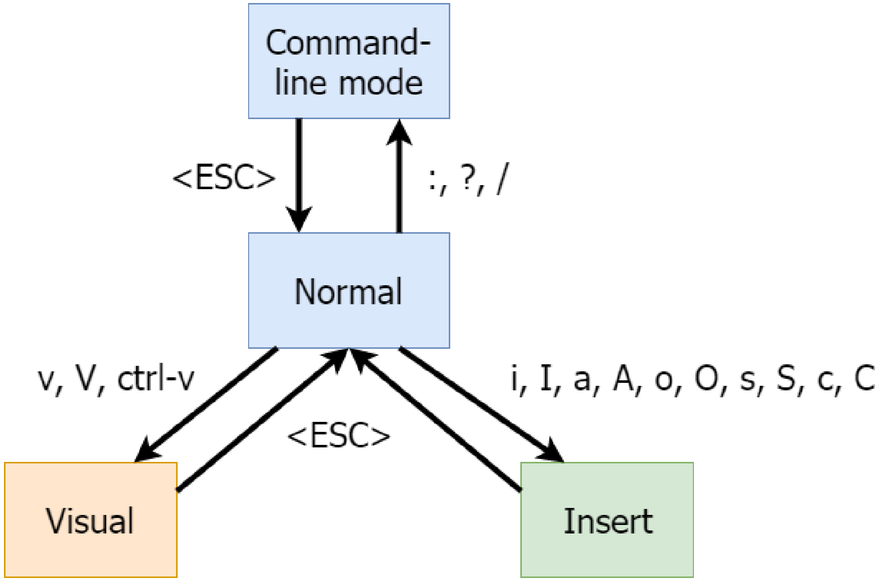
- Vrapper: Vim-like editing in Eclipse
- VSCodeVim: Vim for Visual Studio Code
- Emacs evil mode
- neovim

http://vrapper.sourceforge.net/home/
https://github.com/VSCodeVim/Vim
https://www.emacswiki.org/emacs/Evil
https://neovim.io/

Operating Modes of Vim

- While there are a total of 14 modes, the following modes are used more often
- Normal
  - move cursor, copy, paste, delete, find
  - Enter all the normal editor commands
- Insert
  - type text
  - The text you type is inserted into the buffer
- Visual
  - select line, block
  - Like *Normal mode*, but the movement commands extend a highlighted area
  - When a non-movement command is used, it is executed for the highlighted area
- Command-line mode
  - commands, search
  - One line of text at the bottom of the window
  - This is for the Ex commands, ":", the pattern search commands, "?" and "/", and the filter command, "!"

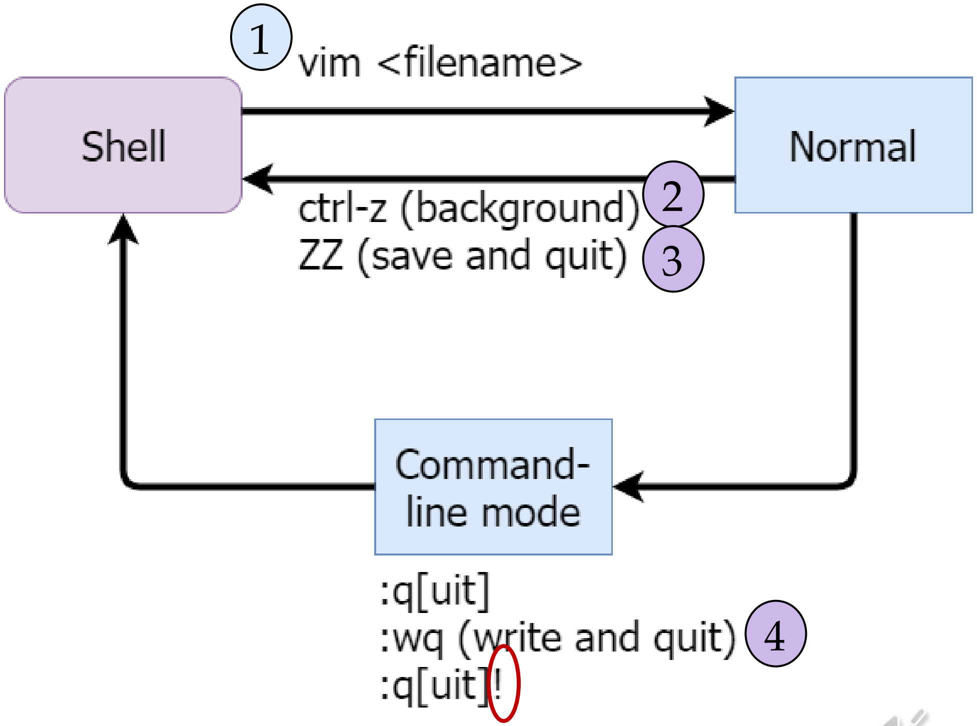
Switching between Vim modes



Usage – Enter and Exit

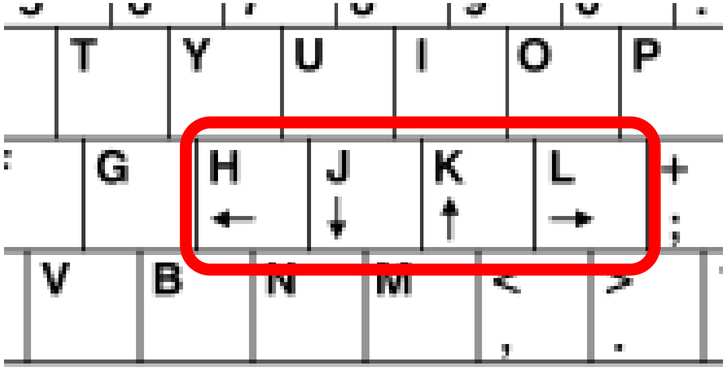
- Open a file
  1. Use Vim to open the file
- Exit vim
  2. Quit Vim and Vim runs in background
  3. Save the current file status and quit Vim
  4. Save data and quit Vim

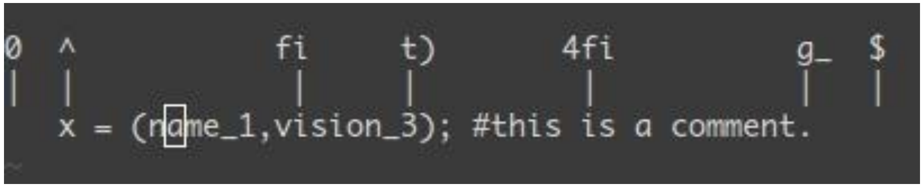
  – "!" means to force Vim to execute the given commands



Usage – Move in Normal Mode

- Up and down
  - **k, j**
  - **gg, G** (first, last line of the file)
- Left and right
  - **h, l**
- Find character in this line
  - **f, F, t, T**
  - **0, ^, $**





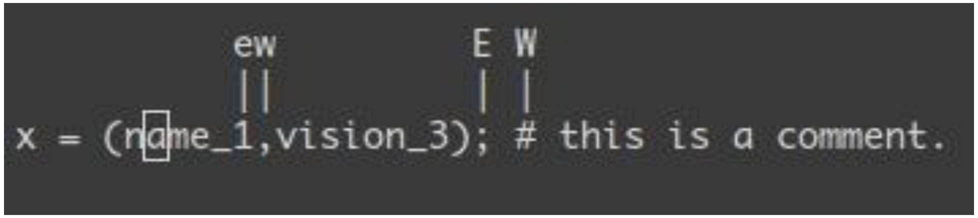
```
0  ^          fi     t)        4fi          g_  $
|  |          |      |         |            |   |
   x = (name_1,vision_3); #this is a comment.
```

0 → go to column 0
^ → go to first character on the line
$ → go to the last column
g_ → go to the last character on the line
fa → go to next occurrence of the letter **a** on the line; in this context,
; will further find the next (resp. previous) occurrence.
t, → go to just before the character **,**.
3fa → find the 3rd occurrence of a on this line.
F and **T** → like f and t but backward.

http://yannesposito.com/Scratch/en/blog/Learn-Vim-Progressively/

Usage – Move in Normal mode (cont.)

- Move among words
  - w, W, e, E, b, B



```
          ew         E W
          ||         | |
x = (name_1,vision_3); # this is a comment.
```

w → go to the start of the following word,
e → go to the end of this word.

By default, words are composed of letters and the underscore character. Let’s call a WORD a group of letter separated by blank characters. If you want to consider WORDS, then just use uppercase characters:
W → go to the start of the following WORD separated by blanks
E → go to the end of this WORD.
b, B → similar with w, W, but in reverse direction

- Special movements
  - %
    - move to matched parenthes
    - Go to the corresponding (, {, [
  - H, M, L
    - move to the top, middle, bottom of the window, recpectively

http://yannesposito.com/Scratch/en/blog/Learn-Vim-Progressively/

Usage – Edit

- Switch to the Insert mode
  - **i, I**: insert text before the cursor/the first character
  - **a, A**: append text after the cursor/the end of line
  - **o, O**: begin a new line below/above the cursor
  - **s, S**: replace
  - **c, C**: replace

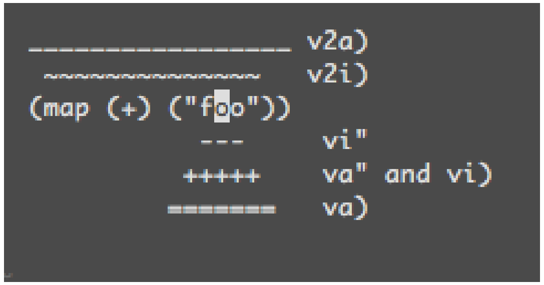
Usage – Edit in Normal Mode (Cont.)

- Edit in Normal mode

Operations	Key
Copy (yank)	y
Copy line	Y, yy
Delete (cut)	d, x
Delete (cut) line	dd
Paste	p, P
Undo	u
Redo	Ctrl r

Usage – Misc.

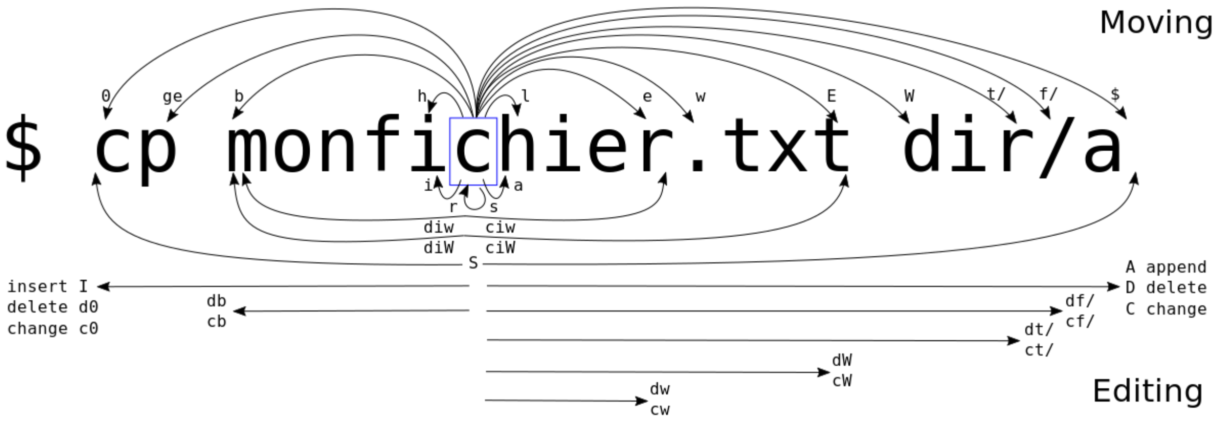
- Text object selection while in Normal mode
  - d, v, c + i + w, s, ", >,),], }
  - e.g., `viw`
- Cheat sheet of the previous commands



```
 _____________________  v2a)
   ~~~~~~~~~~~~~~~~~~   v2i)
(map (+) ("foo"))
             ---         vi"
            +++++        va" and vi)
           =======       va)
```

Suppose the cursor is on the first o of (map (+) ("foo")).

vi" → will select foo.
va" → will select "foo".
vi) → will select "foo".
va) → will select ("foo").
v2i) → will select map (+) ("foo")
v2a) → will select (map (+) ("foo"))

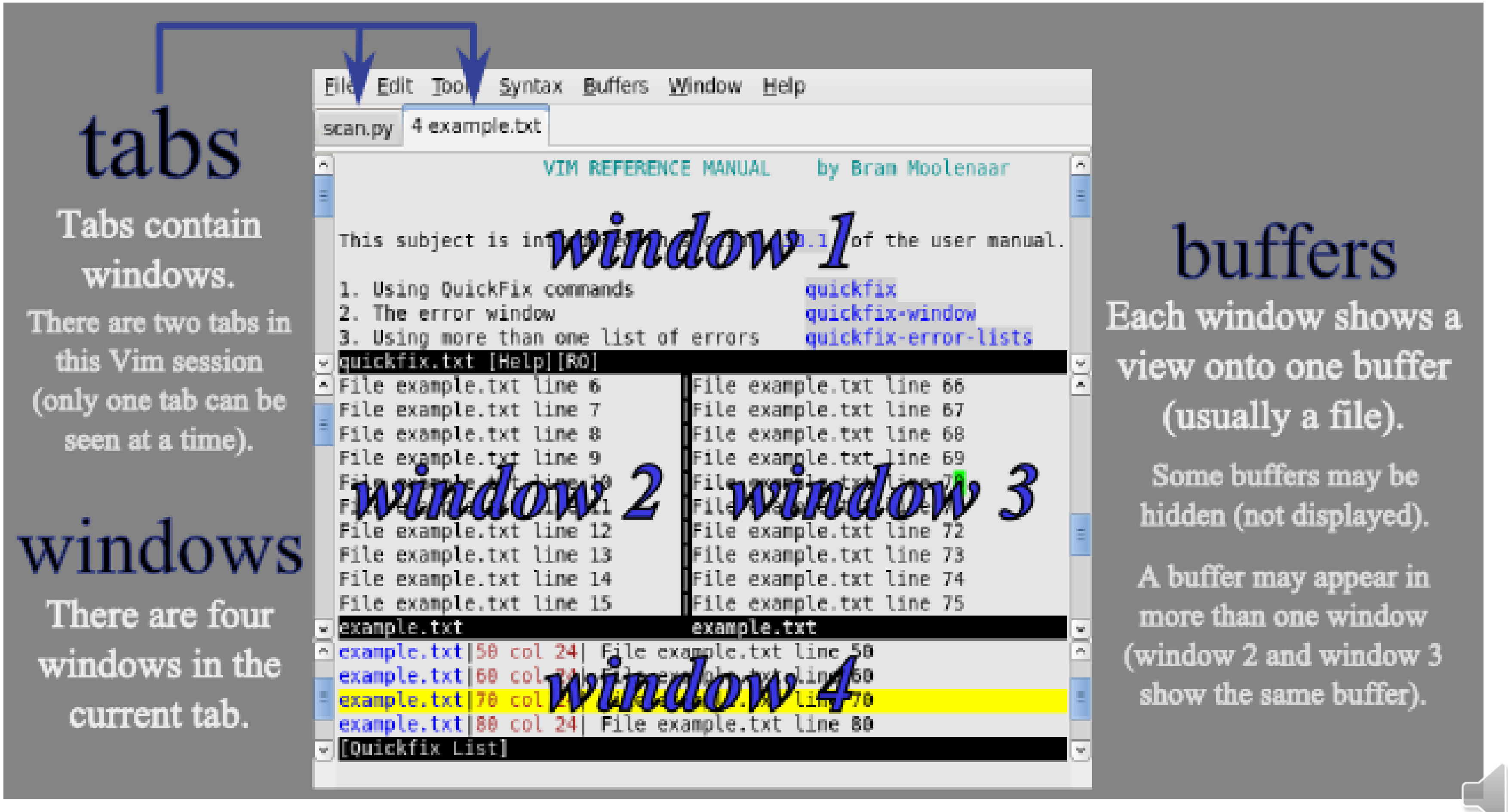


http://yannesposito.com/Scratch/en/blog/Learn-Vim-Progressively/
https://trmm.net/Moving_efficiently
https://vimhelp.org/motion.txt.html#object-select

Buffer, Tab and Window

- A **buffer** refers to Vim's memory holding data of a file
- A **window** displaying the contents of a new (empty) buffer
- A **tab** contains one or more windows

gvim scan.py	start GUI Vim editing file scan.py in first tab
:tabe example.txt	edit file example.txt in a new tab
:cwin	open quickfix window (window 4)
Ctrl-W w	move cursor to other window
:vsp	split window vertically (windows 2 and 3)
:help quickfix	open help (window 1)



```
                 VIM REFERENCE MANUAL    by Bram Moolenaar


This subject is in[illegible]0.1 of the user manual.

1. Using QuickFix commands                quickfix
2. The error window                       quickfix-window
3. Using more than one list of errors     quickfix-error-lists
quickfix.txt [Help][RO]
File example.txt line 6             |File example.txt line 66
File example.txt line 7             |File example.txt line 67
File example.txt line 8             |File example.txt line 68
File example.txt line 9             |File example.txt line 69
Fi[illegible] 10                    |Fil[illegible]
F[illegible]1                       |Fil[illegible]
File example.txt line 12            |File example.txt line 72
File example.txt line 13            |File example.txt line 73
File example.txt line 14            |File example.txt line 74
File example.txt line 15            |File example.txt line 75
example.txt                          example.txt
example.txt|50 col 24| File example.txt line 50
example.txt|60 col [illegible]60
example.txt|70 col [illegible] line 70
example.txt|80 col 24| File example.txt line 80
[Quickfix List]
```

tabs

Tabs contain windows.

There are two tabs in this Vim session (only one tab can be seen at a time).

windows

There are four windows in the current tab.

buffers

Each window shows a view onto one buffer (usually a file).

Some buffers may be hidden (not displayed).

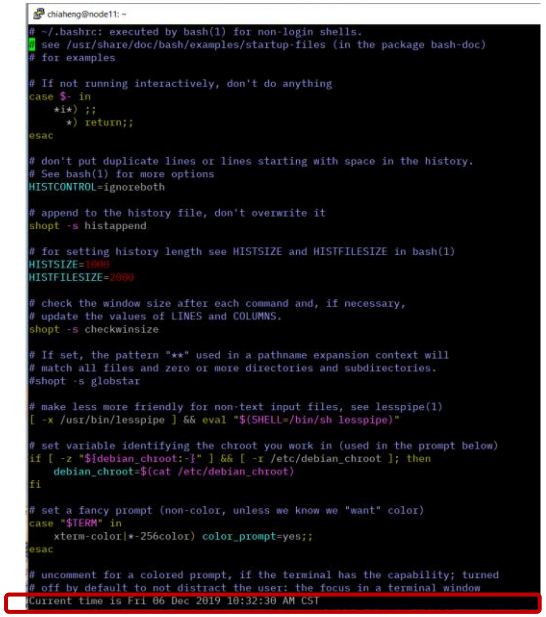
A buffer may appear in more than one window (window 2 and window 3 show the same buffer).

https://vim.fandom.com/wiki/Buffers

Customization – Key Mapping

- Binding a key to do complex task in command-line mode
  - Example of displaying the current date and time by pressing **<F2>** in Normal mode
  - **:map <F2> :echo 'Current time is ' . strftime('%c')<CR>**
  - Key mapping commands in different modes

:nmap - Display normal mode maps
:imap - Display insert mode maps
:vmap - Display visual and select mode maps
:smap - Display select mode maps
:xmap - Display visual mode maps
:cmap - Display command-line mode maps
:omap - Display operator pending mode maps



chiaheng@node11: ~

```
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
    *i*) ;;
      *) return;;
esac

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000

# check the window size after each command and, if necessary,
# update the values of LINES and COLUMNS.
shopt -s checkwinsize

# If set, the pattern "**" used in a pathname expansion context will
# match all files and zero or more directories and subdirectories.
#shopt -s globstar

# make less more friendly for non-text input files, see lesspipe(1)
[ -x /usr/bin/lesspipe ] && eval "$(SHELL=/bin/sh lesspipe)"

# set variable identifying the chroot you work in (used in the prompt below)
if [ -z "${debian_chroot:-}" ] && [ -r /etc/debian_chroot ]; then
    debian_chroot=$(cat /etc/debian_chroot)
fi

# set a fancy prompt (non-color, unless we know we "want" color)
case "$TERM" in
    xterm-color|*-256color) color_prompt=yes;;
esac

# uncomment for a colored prompt, if the terminal has the capability; turned
# off by default to not distract the user: the focus in a terminal window
Current time is Fri 06 Dec 2019 10:32:30 AM CST
```

COMMANDS			MODES
:map	:noremap	:unmap	Normal, Visual, Select, Operator-pending
:nmap	:nnoremap	:nunmap	Normal
:vmap	:vnoremap	:vunmap	Visual and Select
:smap	:snoremap	:sunmap	Select
:xmap	:xnoremap	:xunmap	Visual
:omap	:onoremap	:ounmap	Operator-pending
:map!	:noremap!	:unmap!	Insert and Command-line
:imap	:inoremap	:iunmap	Insert
:lmap	:lnoremap	:lunmap	Insert, Command-line, Lang-Arg
:cmap	:cnoremap	:cunmap	Command-line
:tmap	:tnoremap	:tunmap	Terminal-Job

https://vimhelp.org/map.txt.html#%3Amap-commands
https://vim.fandom.com/wiki/Mapping_keys_in_Vim_-_Tutorial_(Part_1)
https://vim.fandom.com/wiki/Mapping_keys_in_Vim_-_Tutorial_(Part_2)

Customization – Your Preferences

- The configuration file records your Vim preferences
  - $HOME/.vimrc or $HOME/.vim/vimrc
- Loaded the settings at Vim startup
  - Contain initialization commands
  - Each line in the file is executed as an Ex command line
    - Your key mapping commands may be put in this file
  - Use comment to explain your setting
  - Example:

```
:map <F2> :echo 'Current time is ' . strftime('%c')<CR>
" Display line numbers on the left
set number
```

- You can find many useful **.vimrc** examples online to decorate and set up your Vim environment

https://vimhelp.org/starting.txt.html#vimrc

Basic Settings in .vimrc

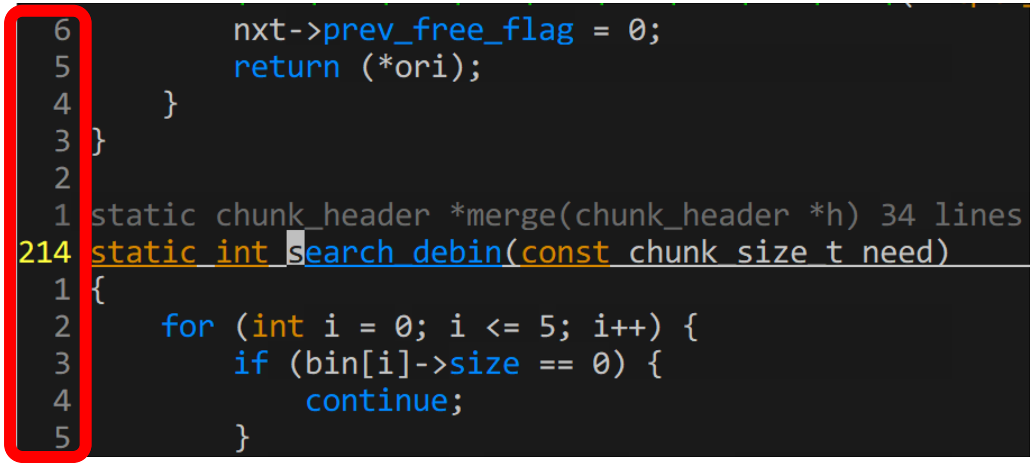
- Key mapping

  nnoremap <F2> :set norelativenumber!<CR>:set nonumber!<CR>

- Line number

  set number

  set relativenumber



```
  6         nxt->prev_free_flag = 0;
  5         return (*ori);
  4     }
  3 }
  2
  1 static chunk_header *merge(chunk_header *h) 34 lines
214 static int search_debin(const chunk_size_t need)
  1 {
  2     for (int i = 0; i <= 5; i++) {
  3         if (bin[i]->size == 0) {
  4             continue;
  5         }
```

- Searching

  " highlight search result

  set hlsearch

  " show result before typing finished

  set incsearch

https://hackmd.io/@aben20807/ryg-CBkmm?type=slide#/3/5

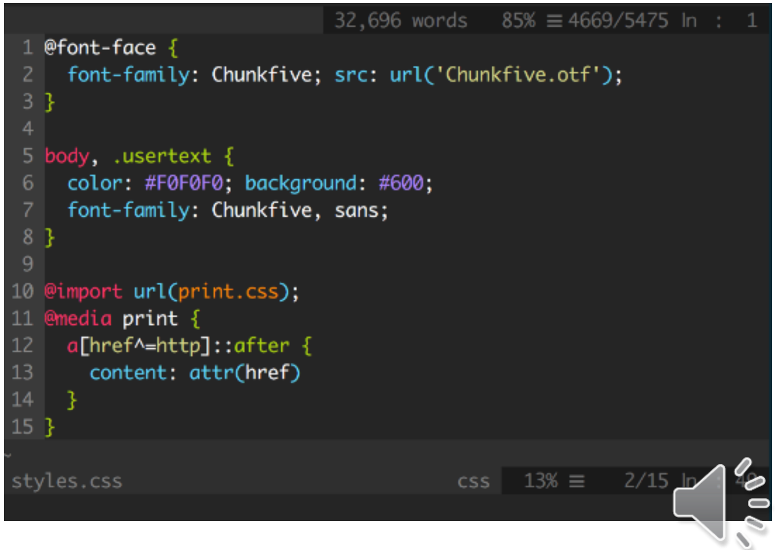
Basic Settings in .vimrc (Cont.)

- Indent size

```
set expandtab     " replace tab to spaces
set tabstop=2     " use 2 spaces to replace a tab
set shiftwidth=2  " indent width
```

- Color theme
  - A pre-configured theme for you to edit texts, e.g., monokai

```
syntax on
set t_Co=256
colorscheme monokai
```



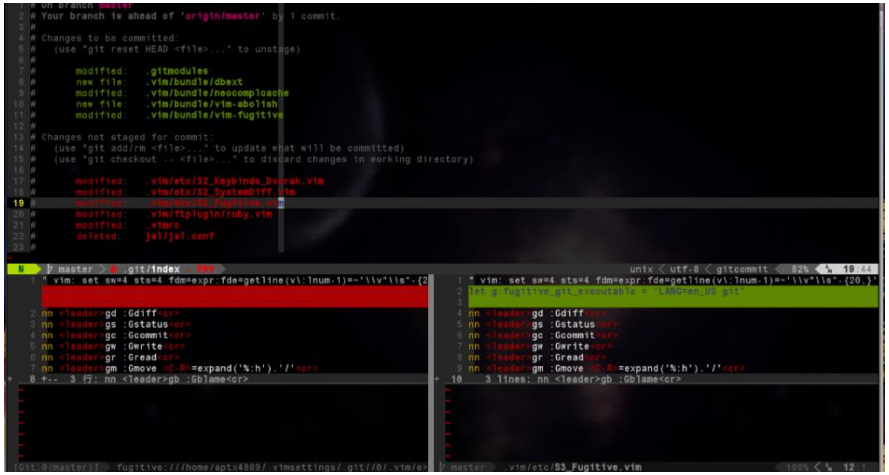
```
32,696 words    85% ≡ 4669/5475 ln  :  1
@font-face {
  font-family: Chunkfive; src: url('Chunkfive.otf');
}

body, .usertext {
  color: #F0F0F0; background: #600;
  font-family: Chunkfive, sans;
}

@import url(print.css);
@media print {
  a[href^=http]::after {
    content: attr(href)
  }
}
styles.css                css  13% ≡  2/15 ln
```

Plugins

- Similar to browsers, Vim has plugins to help you do your work faster
  - **functionlist.vim**: To navigate between functions in a long file
  - **fugitive.vim**: Add Git support in Vim, as shown in the right
- The vim-plug is a plugin manager for Vim
  - On-demand loading
  - Can review and rollback updates
  - Branch/tag/commit support



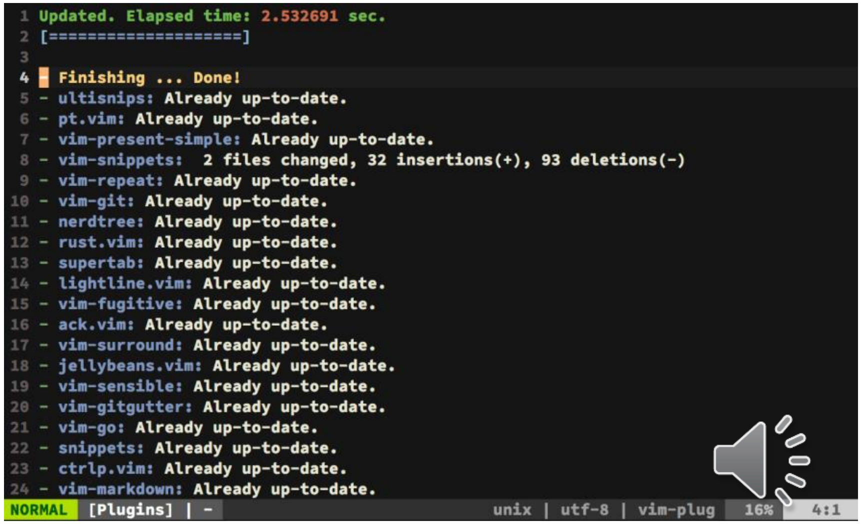
```
# On branch master
# Your branch is ahead of 'origin/master' by 1 commit.
#
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       modified:   .gitmodules
#       new file:   .vim/bundle/dbext
#       modified:   .vim/bundle/neocomplcache
#       new file:   .vim/bundle/vim-abolish
#       modified:   .vim/bundle/vim-fugitive
#
# Changes not staged for commit:
#   (use "git add/rm <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   .vim/etc/S2_Keybinds_Dvorak.vim
#       modified:   .vim/etc/S2_SystemDiff.vim
#       modified:   .vim/etc/S3_Fugitive.vim
#       modified:   .vim/ftplugin/ruby.vim
#       modified:   .vimrc
#       deleted:    jsl/jsl.conf
#
 N  master  .git/index  PRV                      unix < utf-8 < gitcommit  82%  19:44
```

```
" vim: set sw=4 sts=4 fdm=expr:fde=getline(v\:lnum-1)=~'\\v"\\s*-{2

nn <leader>gd :Gdiff<cr>
nn <leader>gs :Gstatus<cr>
nn <leader>gc :Gcommit<cr>
nn <leader>gw :Gwrite<cr>
nn <leader>gr :Gread<cr>
nn <leader>gm :Gmove <C-R>=expand('%:h').'/'<cr>
+--  3 行: nn <leader>gb :Gblame<cr>
[Git:0(master)]  fugitive:///home/aptx4869/.vimsettings/.git//0/.vim/e>
```

```
" vim: set sw=4 sts=4 fdm=expr:fde=getline(v\:lnum-1)=~'\\v"\\s*-{20,}'
let g:fugitive_git_executable = 'LANG=en_US git'

nn <leader>gd :Gdiff<cr>
nn <leader>gs :Gstatus<cr>
nn <leader>gc :Gcommit<cr>
nn <leader>gw :Gwrite<cr>
nn <leader>gr :Gread<cr>
nn <leader>gm :Gmove <C-R>=expand('%:h').'/'<cr>
   3 lines: nn <leader>gb :Gblame<cr>
 master  .vim/etc/S3_Fugitive.vim                 100%  12:1
```



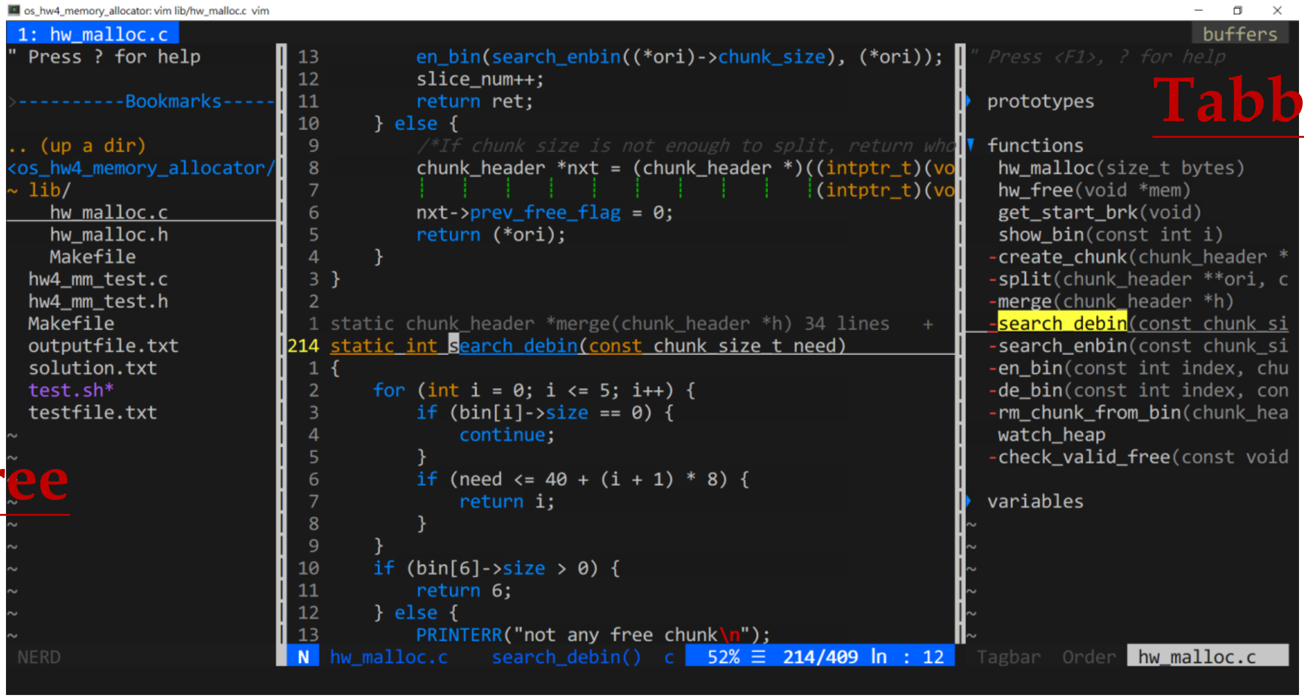
```
Updated. Elapsed time: 2.532691 sec.
[====================]

- Finishing ... Done!
- ultisnips: Already up-to-date.
- pt.vim: Already up-to-date.
- vim-present-simple: Already up-to-date.
- vim-snippets:  2 files changed, 32 insertions(+), 93 deletions(-)
- vim-repeat: Already up-to-date.
- vim-git: Already up-to-date.
- nerdtree: Already up-to-date.
- rust.vim: Already up-to-date.
- supertab: Already up-to-date.
- lightline.vim: Already up-to-date.
- vim-fugitive: Already up-to-date.
- ack.vim: Already up-to-date.
- vim-surround: Already up-to-date.
- jellybeans.vim: Already up-to-date.
- vim-sensible: Already up-to-date.
- vim-gitgutter: Already up-to-date.
- vim-go: Already up-to-date.
- snippets: Already up-to-date.
- ctrlp.vim: Already up-to-date.
- vim-markdown: Already up-to-date.
NORMAL  [Plugins] | -                  unix | utf-8 | vim-plug  16%  4:1
```

https://c7.se/switching-to-vim-plug/

More Plugins

- NERDTree: file system explorer
- Tabbar: class outline viewer



os_hw4_memory_allocator: vim lib/hw_malloc.c vim

```
 1: hw_malloc.c                                                                              buffers
" Press ? for help
>----------Bookmarks-----
.. (up a dir)
<os_hw4_memory_allocator/
~ lib/
    hw_malloc.c
    hw_malloc.h
    Makefile
  hw4_mm_test.c
  hw4_mm_test.h
  Makefile
  outputfile.txt
  solution.txt
  test.sh*
  testfile.txt
~
~
~
~
~
~
~
~
~
~
NERD
```

```
            en_bin(search_enbin((*ori)->chunk_size), (*ori));
            slice_num++;
            return ret;
        } else {
            /*If chunk size is not enough to split, return who
            chunk_header *nxt = (chunk_header *)((intptr_t)(vo
            ¦   ¦   ¦   ¦   ¦   ¦   ¦   ¦   ¦   ¦(intptr_t)(vo
            nxt->prev_free_flag = 0;
            return (*ori);
        }
}

static chunk_header *merge(chunk_header *h) 34 lines    +
static int search_debin(const chunk_size_t need)
{
    for (int i = 0; i <= 5; i++) {
        if (bin[i]->size == 0) {
            continue;
        }
        if (need <= 40 + (i + 1) * 8) {
            return i;
        }
    }
    if (bin[6]->size > 0) {
        return 6;
    } else {
        PRINTERR("not any free chunk\n");
 N  hw_malloc.c    search_debin()  c   52% ☰  214/409 ln  : 12
```

```
" Press <F1>, ? for help

▸ prototypes

▾ functions
   hw_malloc(size_t bytes)
   hw_free(void *mem)
   get_start_brk(void)
   show_bin(const int i)
  -create_chunk(chunk_header *
  -split(chunk_header **ori, c
  -merge(chunk_header *h)
  -search_debin(const chunk_si
  -search_enbin(const chunk_si
  -en_bin(const int index, chu
  -de_bin(const int index, con
  -rm_chunk_from_bin(chunk_hea
   watch_heap
  -check_valid_free(const void

▸ variables
~
~
~
~
~
~
Tagbar  Order  hw_malloc.c
```

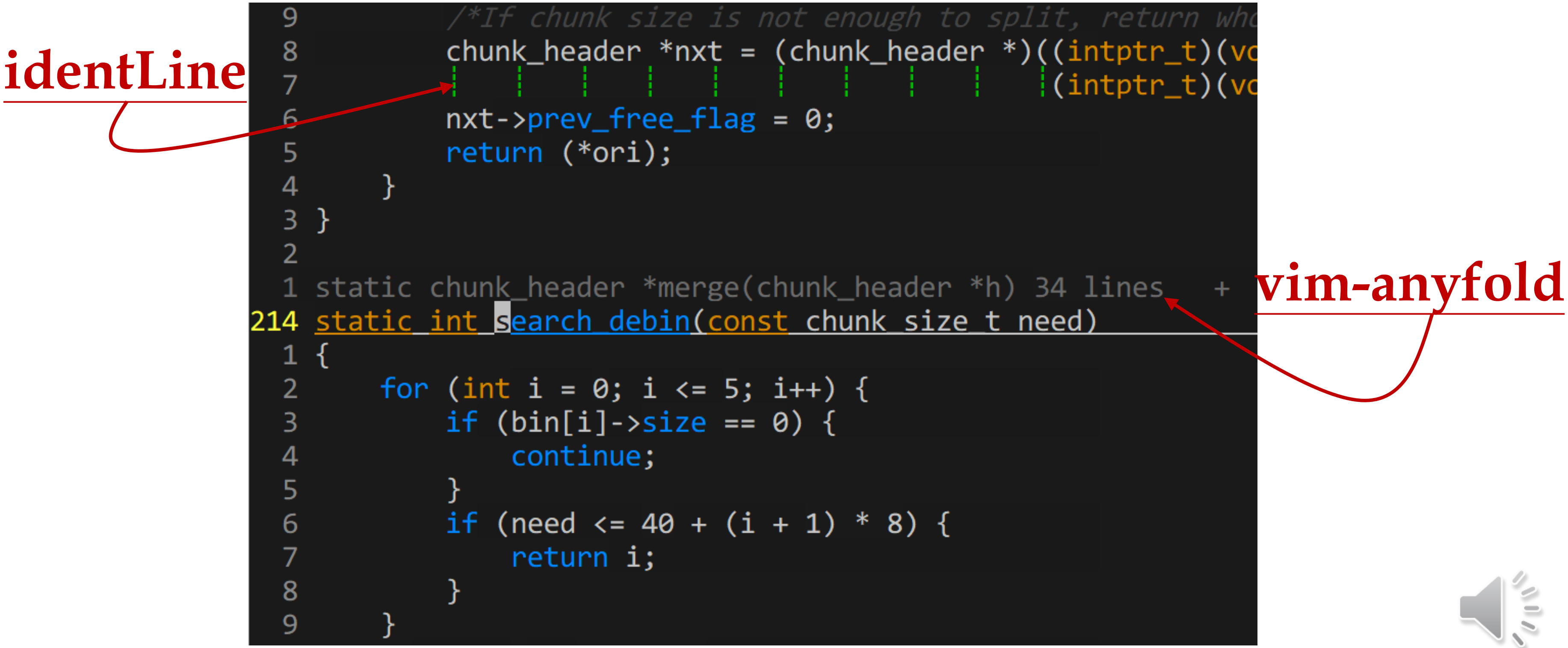
Tabbar

NERDTree

https://hackmd.io/@aben20807/ryg-CBkmm?type=slide#/3/5

More Plugins (Cont.)

- indentLine: display the indention levels
- vim-anyfold: folding texts (e.g., function body, comments)



identLine

vim-anyfold

```
    /*If chunk size is not enough to split, return whe
    chunk_header *nxt = (chunk_header *)((intptr_t)(v
    ¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦(intptr_t)(v
    nxt->prev_free_flag = 0;
    return (*ori);
  }
}

static chunk_header *merge(chunk_header *h) 34 lines   +
static int search_debin(const chunk_size_t need)
{
  for (int i = 0; i <= 5; i++) {
    if (bin[i]->size == 0) {
      continue;
    }
    if (need <= 40 + (i + 1) * 8) {
      return i;
    }
  }
```

https://hackmd.io/@aben20807/ryg-CBkmm?type=slide#/3/5

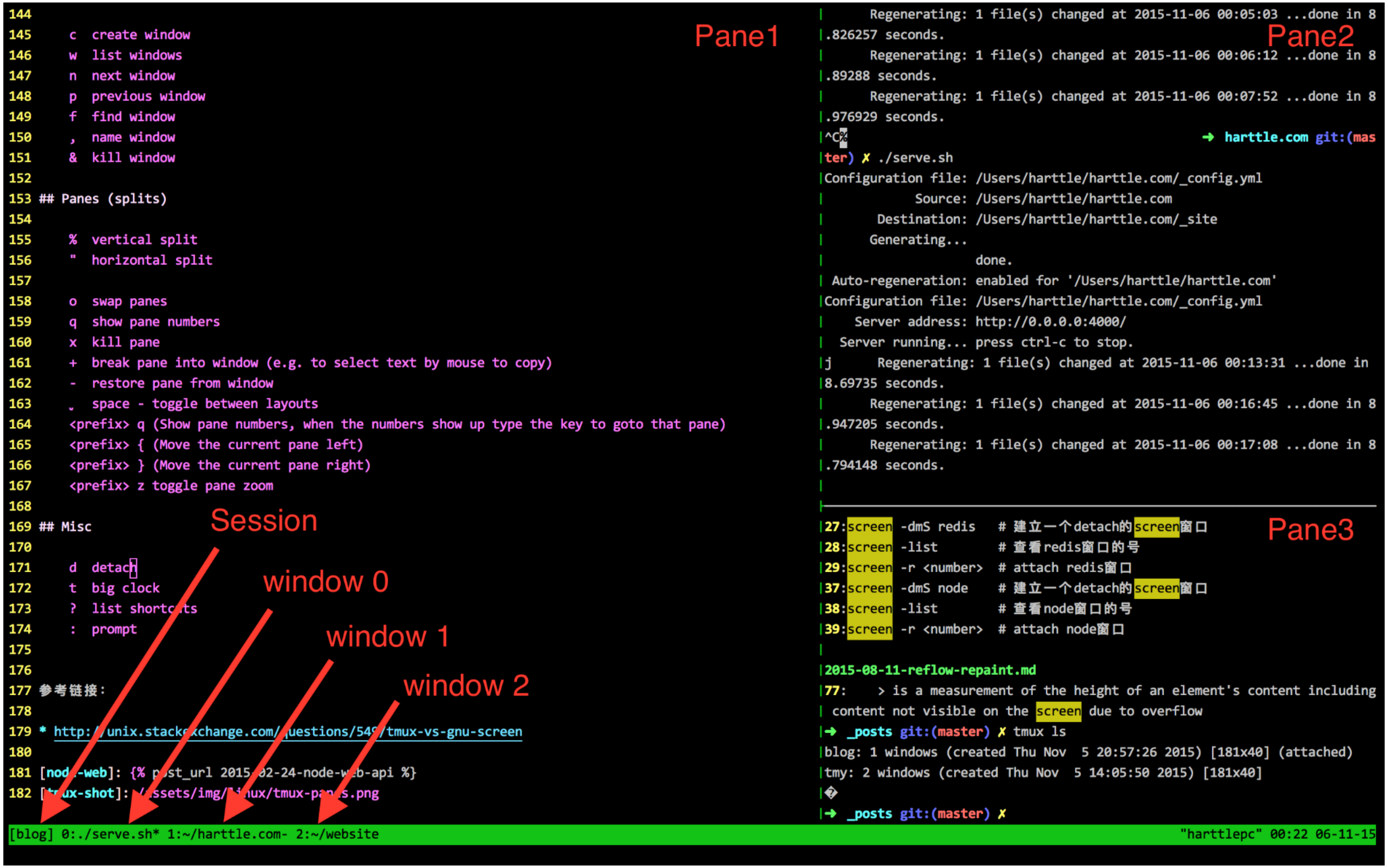
TMUX

tmux

- Terminal multiplexer
  - Within one terminal window you can open multiple windows and split-views
  - Enables a number of terminals to be created, accessed, and controlled from a **single screen**
  - **Can be detached from a screen and continue running in the background**
- An easier-to-use and powerful alternative to GNU Screen

https://github.com/tmux/tmux/wiki

Session, Window, and Pane



```

    c  create window
    w  list windows
    n  next window
    p  previous window
    f  find window
    ,  name window
    &  kill window

## Panes (splits)

    %  vertical split
    "  horizontal split

    o  swap panes
    q  show pane numbers
    x  kill pane
    +  break pane into window (e.g. to select text by mouse to copy)
    -  restore pane from window
    ⍽  space - toggle between layouts
    <prefix> q (Show pane numbers, when the numbers show up type the key to goto that pane)
    <prefix> { (Move the current pane left)
    <prefix> } (Move the current pane right)
    <prefix> z toggle pane zoom

## Misc

    d  detach
    t  big clock
    ?  list shortcuts
    :  prompt


参考链接：

* http://unix.stackexchange.com/questions/549/tmux-vs-gnu-screen

[node-web]: {% post_url 2015-02-24-node-web-api %}
[tmux-shot]: /assets/img/linux/tmux-panes.png
```

```
      Regenerating: 1 file(s) changed at 2015-11-06 00:05:03 ...done in 8
.826257 seconds.
      Regenerating: 1 file(s) changed at 2015-11-06 00:06:12 ...done in 8
.89288 seconds.
      Regenerating: 1 file(s) changed at 2015-11-06 00:07:52 ...done in 8
.976929 seconds.
^C%                                                   ➜ harttle.com git:(mas
ter) ✗ ./serve.sh
Configuration file: /Users/harttle/harttle.com/_config.yml
            Source: /Users/harttle/harttle.com
       Destination: /Users/harttle/harttle.com/_site
      Generating...
                    done.
 Auto-regeneration: enabled for '/Users/harttle/harttle.com'
Configuration file: /Users/harttle/harttle.com/_config.yml
    Server address: http://0.0.0.0:4000/
  Server running... press ctrl-c to stop.
j      Regenerating: 1 file(s) changed at 2015-11-06 00:13:31 ...done in
8.69735 seconds.
      Regenerating: 1 file(s) changed at 2015-11-06 00:16:45 ...done in 8
.947205 seconds.
      Regenerating: 1 file(s) changed at 2015-11-06 00:17:08 ...done in 8
.794148 seconds.

```

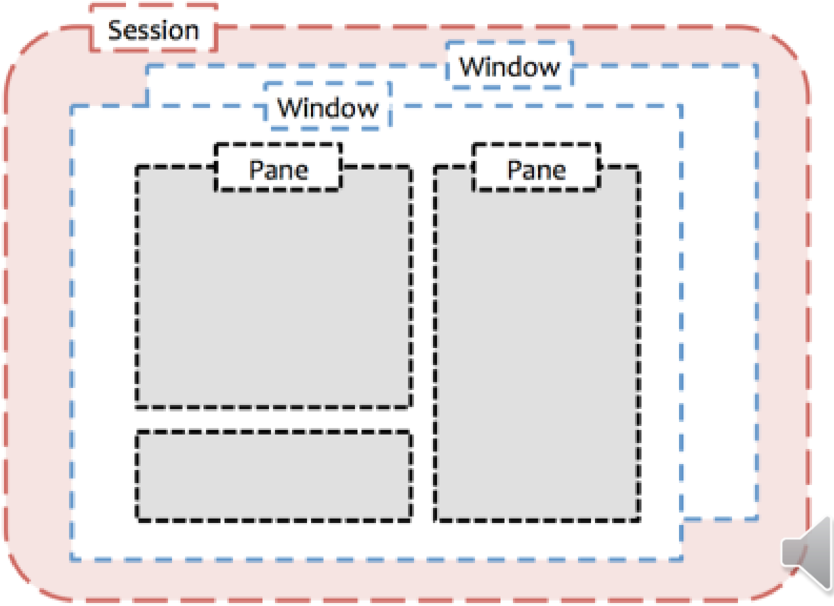
```
27:screen -dmS redis   # 建立一个detach的screen窗口
28:screen -list        # 查看redis窗口的号
29:screen -r <number>  # attach redis窗口
37:screen -dmS node    # 建立一个detach的screen窗口
38:screen -list        # 查看node窗口的号
39:screen -r <number>  # attach node窗口

2015-08-11-reflow-repaint.md
77:    > is a measurement of the height of an element's content including
 content not visible on the screen due to overflow
➜  _posts git:(master) ✗ tmux ls
blog: 1 windows (created Thu Nov  5 20:57:26 2015) [181x40] (attached)
tmy: 2 windows (created Thu Nov  5 14:05:50 2015) [181x40]
�
➜  _posts git:(master) ✗
```

```
[blog] 0:./serve.sh* 1:~/harttle.com- 2:~/website                    "harttlepc" 00:22 06-11-15
```

Basic Functions

- **$ tmux** *# Start the session*
- **$ tmux a** *# Attach to first available session*
- prefix key: **Ctrl-b**
  - **Ctrl-b d** Detach client from current session
  - **Ctrl-b c** Create window
  - **Ctrl-b ,** Rename window
  - **Ctrl-b n** Next window
  - **Ctrl-b p** Previous window
  - **Ctrl-b w** List all windows
  - **Ctrl-b %** Split pane horizontally
  - **Ctrl-b "** Split pane vertically



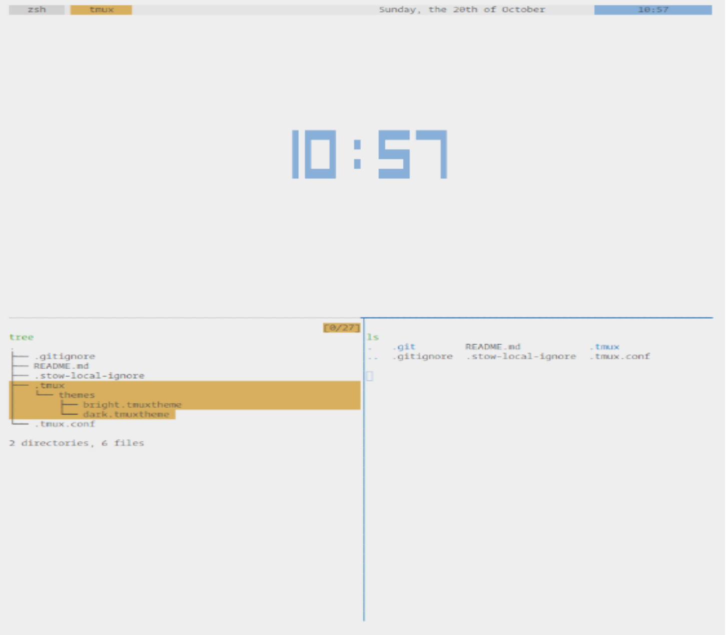
https://umarkalim.wordpress.com/2014/05/13/tmux-workspace/

.tmux.conf

- To set up tmux with **~/.tmux.conf**

```
# split panes using | and –
bind | split-window -h -c '#{pane_current_path}'
bind - split-window -v -c '#{pane_current_path}’

# switch panes using Alt-arrow without prefix
bind -n M-Left select-pane -L
bind -n M-Right select-pane -R
bind -n M-Up select-pane -U
bind -n M-Down select-pane -D
```

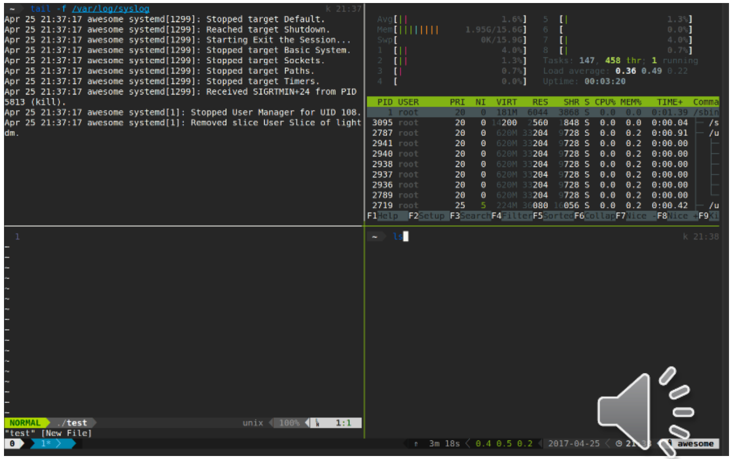


```
 zsh   tmux                       Sunday, the 20th of October        10:57

                                  10:57

                                       [0/27]
tree                                           ls
.                                              .   .git        README.md           .tmux
├── .gitignore                                 ..  .gitignore  .stow-local-ignore  .tmux.conf
├── README.md
├── .stow-local-ignore
├── .tmux
│   └── themes
│       ├── bright.tmuxtheme
│       └── dark.tmuxtheme
└── .tmux.conf

2 directories, 6 files
```



```
~  tail -f /var/log/syslog                                   k 21:37
Apr 25 21:37:17 awesome systemd[1299]: Stopped target Default.
Apr 25 21:37:17 awesome systemd[1299]: Reached target Shutdown.
Apr 25 21:37:17 awesome systemd[1299]: Starting Exit the Session...
Apr 25 21:37:17 awesome systemd[1299]: Stopped target Basic System.
Apr 25 21:37:17 awesome systemd[1299]: Stopped target Sockets.
Apr 25 21:37:17 awesome systemd[1299]: Stopped target Paths.
Apr 25 21:37:17 awesome systemd[1299]: Stopped target Timers.
Apr 25 21:37:17 awesome systemd[1299]: Received SIGRTMIN+24 from PID
5813 (kill).
Apr 25 21:37:17 awesome systemd[1]: Stopped User Manager for UID 108.
Apr 25 21:37:17 awesome systemd[1]: Removed slice User Slice of light
dm.
```

```
  Avg[||                1.6%]   5  [|                 1.3%]
  Mem[|||||||||   1.95G/15.6G]   6  [                  0.0%]
  Swp[              0K/15.9G]   7  [|                 4.0%]
  1  [||                4.0%]   8  [|                 0.7%]
  2  [||                1.3%]   Tasks: 147, 458 thr; 1 running
  3  [|                 0.7%]   Load average: 0.36 0.49 0.22
  4  [                  0.0%]   Uptime: 00:03:20

  PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Comma
    1 root       20   0  181M  6044  3868 S  0.0  0.0  0:01.39 /sbin
 3095 root       20   0 14200  2560  1848 S  0.0  0.0  0:00.04 ├─ /s
 2787 root       20   0  620M 33204  9728 S  0.0  0.2  0:00.91 ├─ /u
 2941 root       20   0  620M 33204  9728 S  0.0  0.2  0:00.00 │
 2940 root       20   0  620M 33204  9728 S  0.0  0.2  0:00.00 │
 2938 root       20   0  620M 33204  9728 S  0.0  0.2  0:00.00 │
 2937 root       20   0  620M 33204  9728 S  0.0  0.2  0:00.00 │
 2936 root       20   0  620M 33204  9728 S  0.0  0.2  0:00.00 │
 2789 root       20   0  620M 33204  9728 S  0.0  0.2  0:00.00 │
 2719 root       25   5  224M 36080 16056 S  0.0  0.2  0:00.42 ├─ /u
F1Help  F2Setup F3SearchF4FilterF5SortedF6CollapF7Nice -F8Nice +F9Ki
```

```
~  ls                                                        k 21:38
```

```
NORMAL  ./test                          unix  100%   1:1
"test" [New File]
 0  1*                       3m 18s  0.4 0.5 0.2  2017-04-25  21:38  awesome
```

https://github.com/kimat/tmux
http://peterforgacs.github.io/2017/04/25/Tmux/

References

- Learn Vim Progressively
- Vim.js
  - Run Vim program online within your browser
- Vim help files
- Vim Tips Wiki
- mhinz/vim-galore

- TMUX Guide documentation
- tmux/tmux
- tmux(1) manual page