
tinge@tinge-work-server: ~/android
4: [No Name]
buffers
VIM
:Matrix
tinge@tinge-work-server|0< 1:android 2:bash 3*vim 4:pycolor 5:bash 6:[tmux]
>2014-05-28|09:19

Contents

install

```
1. git clone https://github.com/tingezhang/vtconfig.git
2. cd vtconfig
3. cp .vimrc .tmux.conf .tmux.status.conf ~/
4. mkdir ~/.vim/bundle
5. git clone https://github.com/gmarik/Vundle.vim.git ~/.vim/bundle/Vundle.vim
6. vim +PluginInstall +qall
7. modify ~/.bashrc, add "alias tmux='tmux -2'" at the end of file
8. install font for status line:
   1. git clone https://github.com/runsisi/consolas-font-for-powerline
   1. put all font into system font directory
   1. config putty or terminal to use powerline-consola font to
```

install font on window

1. copy all powerline-consola font file to "Control Panel\All Control Panel Items\Fonts"
2. restart putty and set font to "powerline consola"

command for plugins

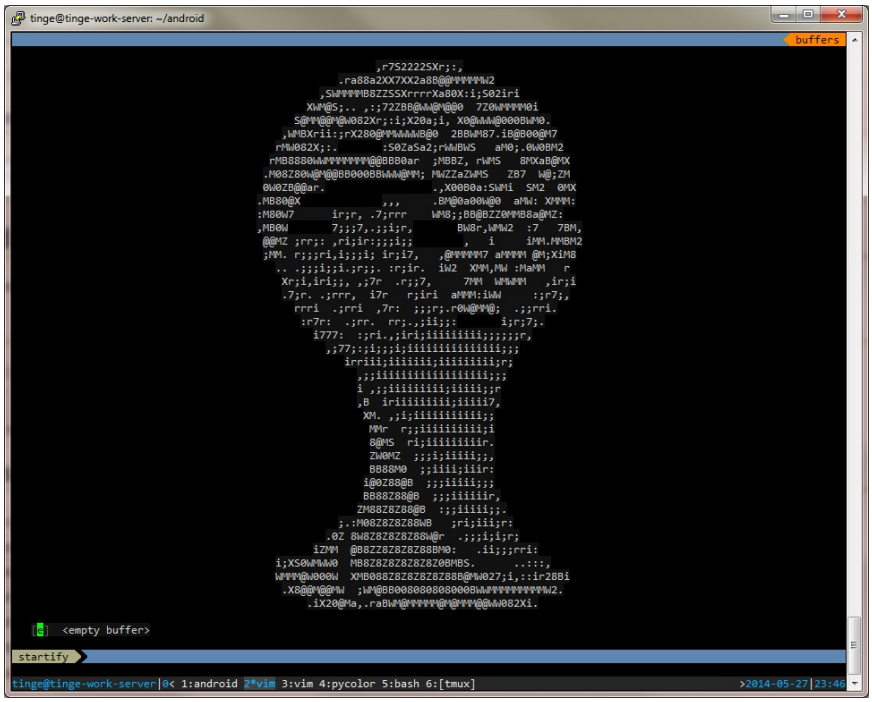
1. some tips for most often used commands/configs for plugins

vim-startify

1. https://github.com/mhinz/vim-startify

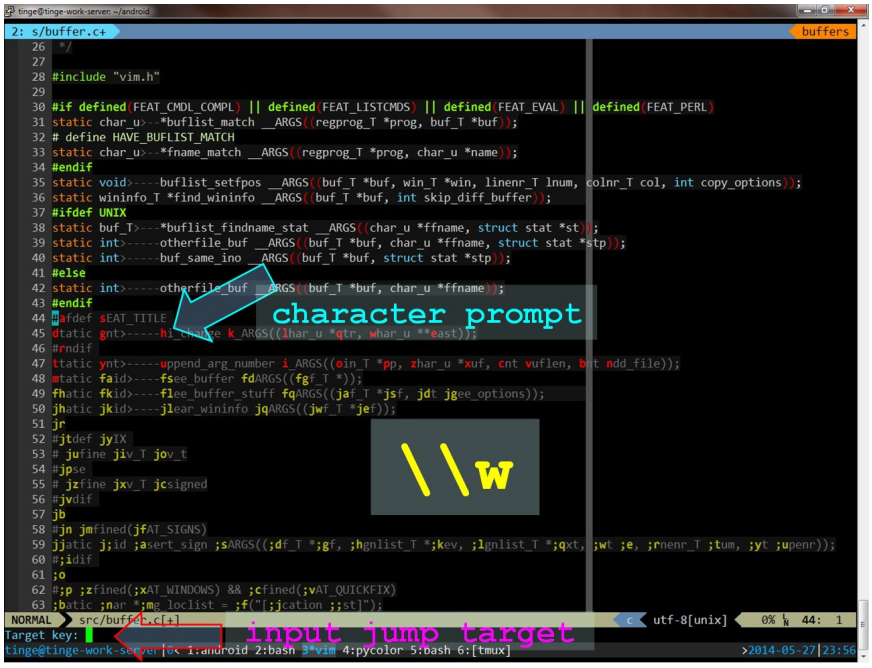
config	description

g:startify_custom_header	config the custom header



vim-easymotion

Command	Description
\\w	Beginning of word forward
\\b	Beginning of word backward
\\e	End of word forward
\\ge	End of word backward
\\j	Line downward
\\k	Line upward

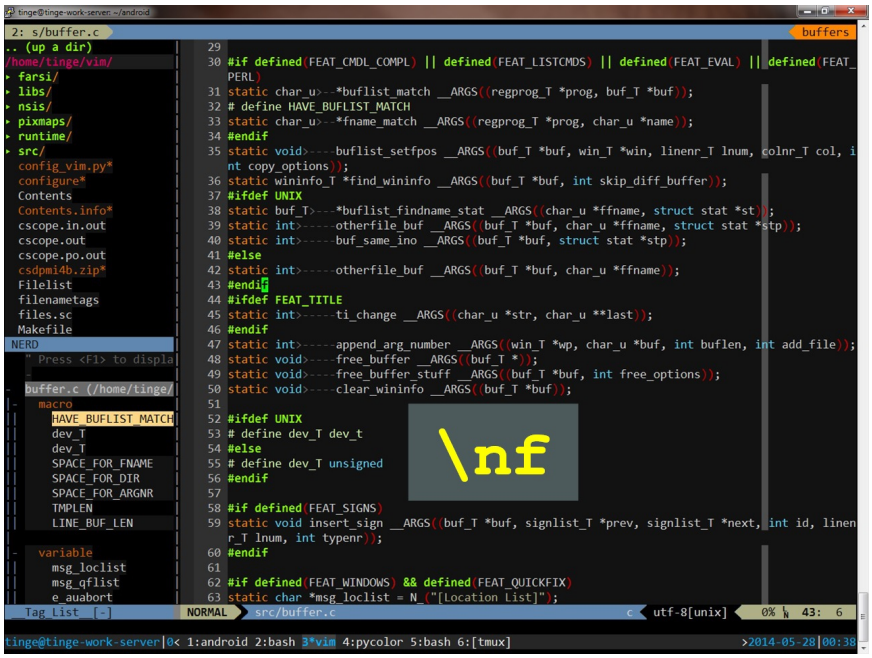


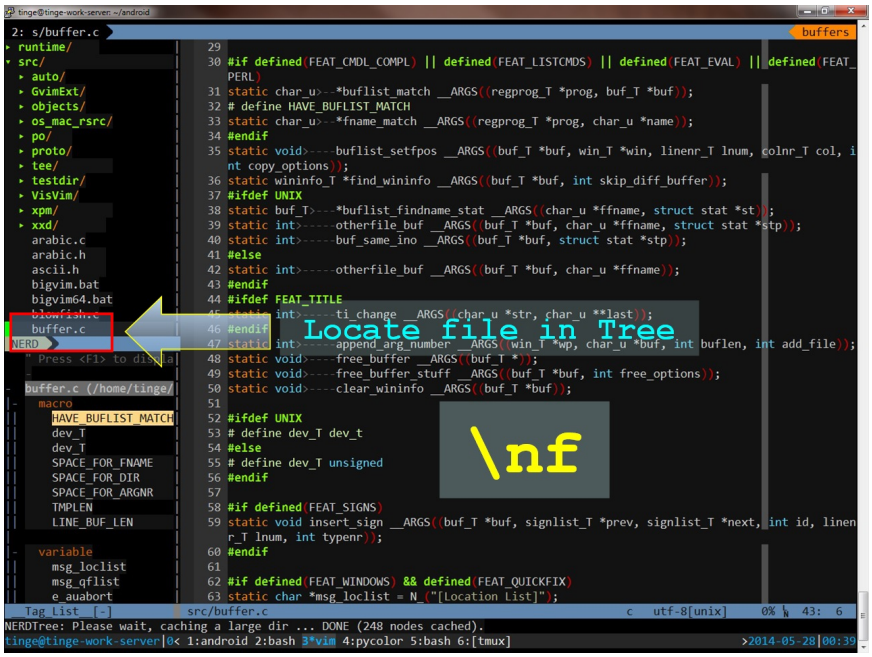
nerdtree

1. Add the following mapping in ~/.vimrc

```
:noremap <leader>nf :NERDTreeFind<CR>
```

Command	Description
\nf	find currently edited buffer in the NERDTree Window



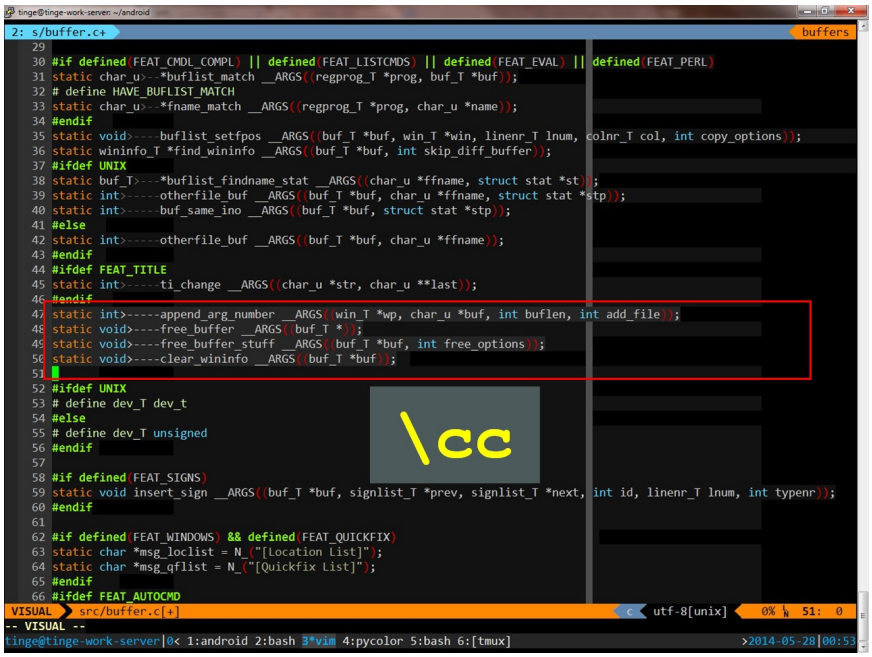


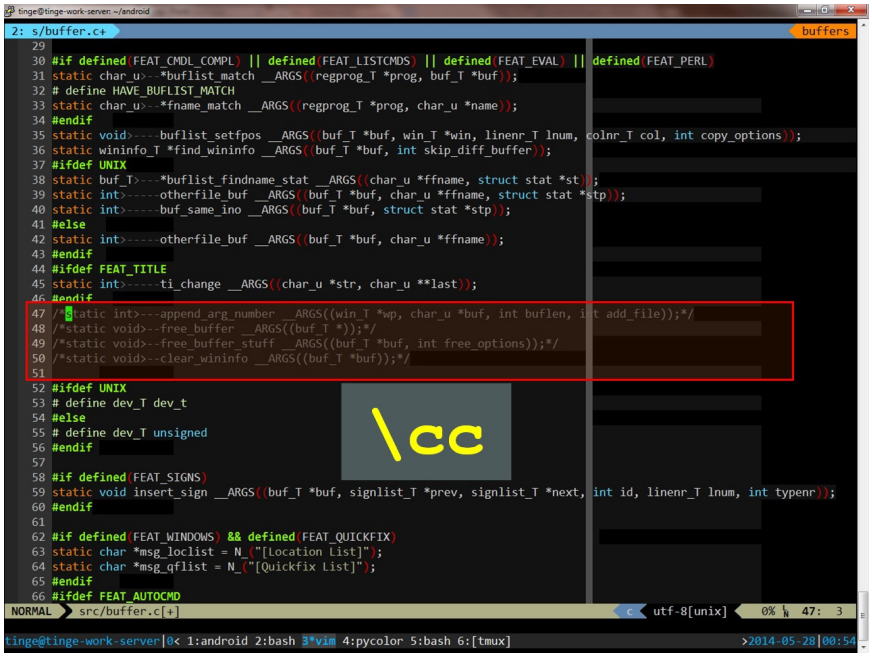
nerdcommenter

1. https://github.com/scrooloose/nerdcommenter
2. plugin to help comment/uncomment select code block

Command	Description

\cc	Comment out the current line or text selected in visual mode
\cu	Uncomments the selected line(s).
\cA	Adds comment delimiters to the end of line and goes into insert mode between them.
\c$	Comments the current line from the cursor to the end of line.



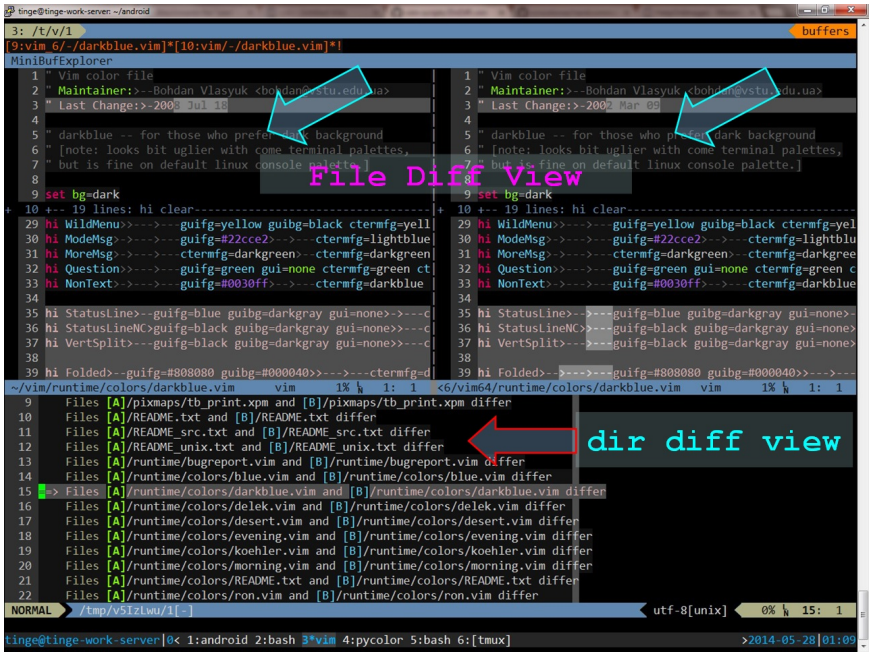


DirDiff

1. https://github.com/vim-scripts/DirDiff.vim
2. plugins to help compare diff between two directory
3. vim . -c ":DirDiff DIRECTORY_A DIRECTORY_B"

Command	Description
s	Synchronize the current diff. You can also select a range (through visual) and press 's' to synchronize differences across a range.

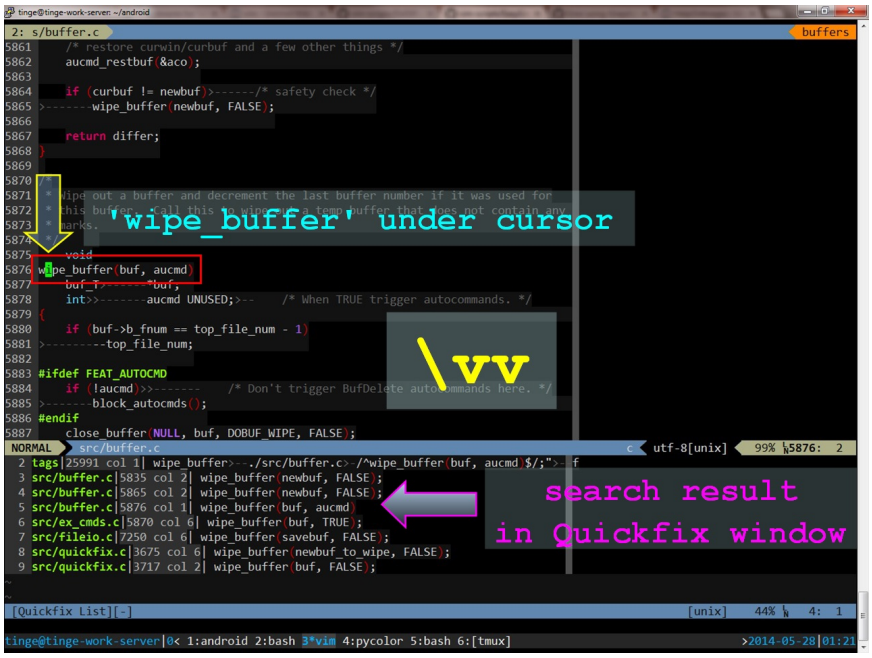
\dj	Diff next: (think j for down)
\dk	Diff previous: (think k for up)



EasyGrep

1. https://github.com/vim-scripts/EasyGrep
2. based on ":vimgrep" command to find tag under cursor easily

Command	Description
<Leader>vv (double v)	Grep for the word under the cursor, match all occurences



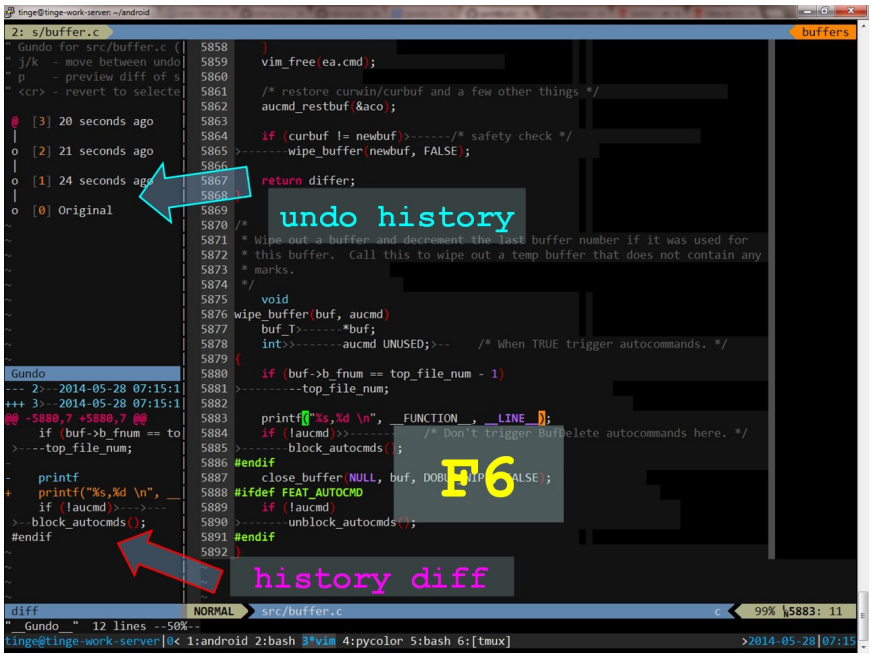
gundo

1. https://github.com/sjl/gundo.vim

2. Gundo.vim is Vim plugin to visualize your Vim undo tree.
3. Add the following mapping in ~/.vimrc

```
nnoremap <F6> :GundoToggle<CR>
```

Command	Description
F6	Toggle undo window



ctrlp

1. https://github.com/kien/ctrlp.vim
2. Full path fuzzy file, buffer, mru, tag, ... finder for Vim.

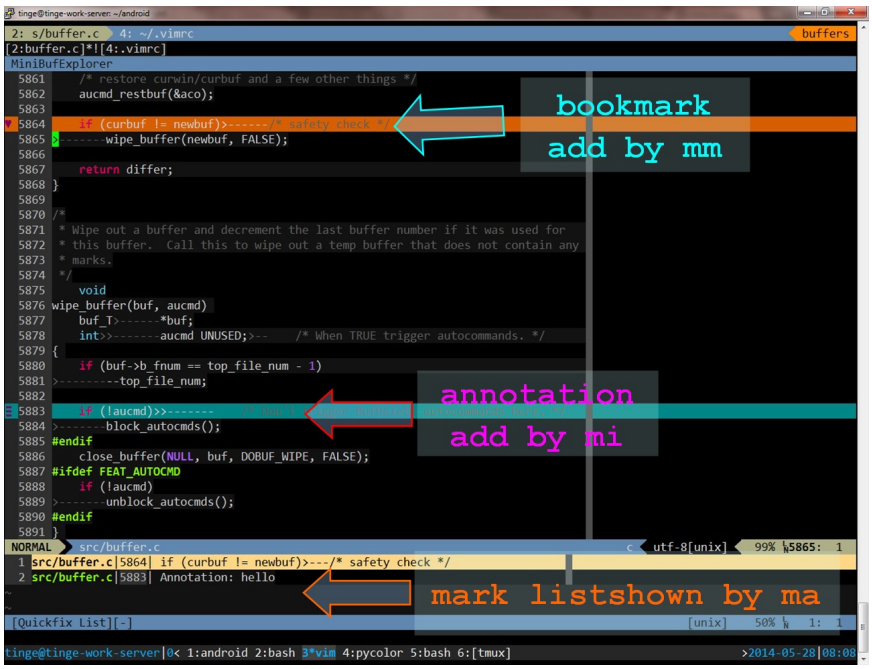
Command	Description
<Ctrl-p>	invoke CtrlP in find file mode
<Ctrl-f> | <Ctrl-b>	cycle between three modes. (Find files, Find Buffer, Find MRU)
<Ctrl-d>	switch to filename only search instead of full path.
<Ctrl-j> | <Ctrl-k> | narrow keys	to navigate the result list

vim-bookmarks

1. https://github.com/MattesGroeger/vim-bookmarks
2. bookmaks which works independently from vim marks
3. add the following config into .vimrc

```
let g:bookmark_sign = '♥'
let g:bookmark_highlight_lines = 1
highlight BookmarkSign ctermbg=166 ctermfg=89
highlight BookmarkAnnotationSign ctermbg=30  ctermfg=89
highlight BookmarkLine ctermbg=166 ctermfg=NONE
highlight BookmarkAnnotationLine ctermbg=30  ctermfg=NONE
```

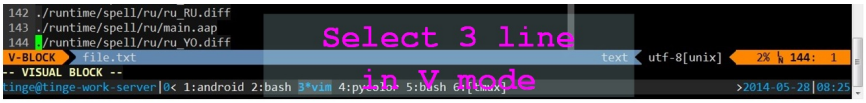
Command	Description
mm	Add/remove bookmark at current line
mi	Add/edit/remove annotation at current line
ma	Show all bookmarks
mn	Jump to next bookmark in buffer
mp	Jump to previous bookmark in buffer
mc	Clear bookmarks in current buffer only
mx	Clear bookmarks in all buffers

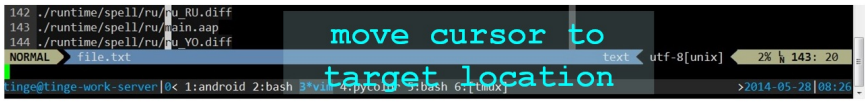


vim-multiple-cursors

1. https://github.com/terryma/vim-multiple-cursors
2. Porting of multiple selection feature from Sublime

Command	Description
<Ctrl-n>	select next key
<Ctrl-p>	in Visual mode will remove the current virtual cursor and go back to the previous virtual cursor location.
<Ctrl-x>	in Visual mode will remove the current virtual cursor and skip to the next virtual cursor location.

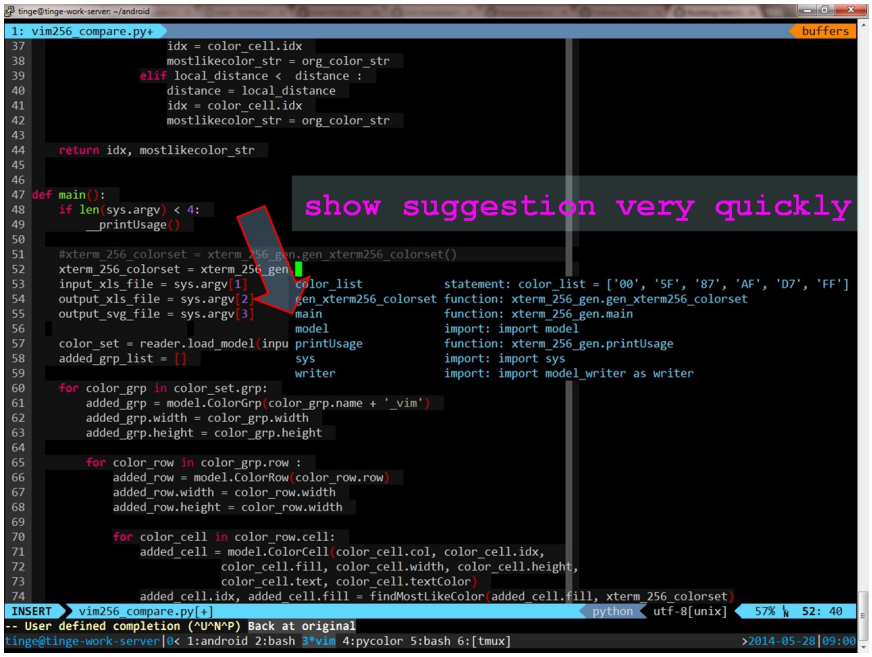




YouCompleteMe

1. https://github.com/Valloric/YouCompleteMe
2. Vim Killer plugin for code completion engine
3. C/S architecture, make the completion suggestion very fast
4. Configure is a little complicated
5. Make sure Vim 7.3.584 with python2 support. some time need rebuild vim to the latest version(7.4)

1. https://github.com/Valloric/YouCompleteMe/wiki/Building-Vim-from-source



syntastic

1. https://github.com/scrooloose/syntastic
2. Syntastic is a syntax checking plugin for Vim that runs files through external syntax checkers and displays any resulting errors to the user
3. following picture is from his github

matrix

1. https://github.com/vim-scripts/matrix.vim--Yang
2. screen save for funny

useful command

buffer operation

No	Command	description	example
1	:buffers		show all buffers (the same as :ls)
2	:buffer 2(b2)		show buffer 2
3	:bf		bfirst: show first buffer
4	:bn		bnext: show next buffer
5	:bp		bprevious: show previous buffer
6	:bd 3		bdelete: delete buffer
7	:bufdo[!]{cmd}		Execute {cmd} in each buffer in the buffer list
8	:sb[n]		split window to show buffer [N] in horizontal mode
9	:e#		edit the last edit buffer

Windows command

1. Add the following definition in ~/.vimrc

```
noremap <C-J>     <C-W>j
noremap <C-K>     <C-W>k
noremap <C-H>     <C-W>h
noremap <C-L>     <C-W>l

noremap <C-Down>  <C-W>j
noremap <C-Up>    <C-W>k
noremap <C-Left>  <C-W>h
noremap <C-Right> <C-W>l

:set mouse=a
:set ttymouse=xterm2
```

Modify window boundary

Command	Description
<C-W> +, <C-W> -	adjust window row number
<C-W> >, <C-W> <	adjust window column number
20<C-W>[+|=]	adjust 20 unit one time
<C-W> |	adjust window to maximum of column
<C-W> _	adjust window to maximum of row
'x' in Tlist window	toggle TList window full screen display

add map in .vimrc as the following

Navigate between windows

Command	Description
<C-J> | <C-Down> | <C-W> j	switch to the window below
<C-K> | <C-UP> | <C-W> k	switch to the window above
<C-H> | <C-Left> | <C-W> h	switch to the window left
<C-L> | <C-Right> | <C-W> l	switch to the window right

<C-W> <C-W>	switch between windows

locate currently edited file in NERDTree window

1. :NERDTreeFind ->Find the current file in the tree.

jump in a file

Command	Description
A	Cursor jump to end of line and enter 'insert' mode
CRTL+^	line head(first characther of line)
CRTL+$	line end
0(zero)	column zero of line
CRTL+o	go back to previous edit point
CRTL+i	go forward to next edit point
G	file end
gg	file head
CTRL+]	jump to define <within ctags and cscope>
CTRL+t	back to pre-jump tag<within ctags and cscope>
\\w	show beginning of word forward and then select to jump <within easymotion plugin>
\\b	show beginning of work backard and then select to jump <within easymotion plugin>
o(not zero)	enter a new line below current line and enter insert mode
O(not zero, big cap)	enter a new line above current line and enter insert mode
%	map 1) #if-#else-#endif 2){}
[#	go to end of #if-#else-#endif
]#	go to begin of #if-#else-#endif
[[	jump to front of function body or front of previous function body
[]	jump to end of function body
]]	jump to the front of next function body

search in vim

command	usage	description
vimgrep		
	:vimgrep /{pattern}/[g][j] {file} ...	
	:vimgrep /MarketId/ **/*.c **/*.h	and then :cw open quickfix window to watch the search list
grep		
	::grep -n --exclude=*.h -R MarketId . -n	and then :cw open quickfix window to watch grep result

quickfix		
	:cc	show the detail error info
	:cp	jump to previous item
	:cn	jump to next item
	:cl	list all item
	:cw	if any items, open quickfix window
	:copen	open quickfix window, accept param for the height, for example: :copen 10
	:cclose	close quickfix window
	:col	previous old items list
	:cnew	next new items list
EasyGrep		
	<Leader>vv	Grep for the word under the cursor, match all occurences, like 'gstar'
	<leader>vV	Grep for the word under the cursor, match whole word, like 'star'
	<Leader>va	like vv, but add to existing list
	<Leader>vA	like vV, but add to existing list
	<Leader>vr	Perform a galobal search search on the word under the cursor and prompt for a pattern with which to replace it
	<Leader>vo	Select the files to search in and set grep options

map for cscope

1. :nmap <C-f>s :cs find s <C-R><C-W><CR>
2. :nmap <C-f>g :cs find g <C-R><C-W><CR>
3. :nmap <C-f>c :cs find c <C-R><C-W><CR>
4. :nmap <C-f>t :cs find t <C-R><C-W><CR>

vim (last edited 2014-05-28 08:25:47 by tingezhang)