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

ref

1. vim document(help): http://vimdoc.sourceforge.net/htmldoc/
2. editmoin: http://magicsword.wordpress.com/tag/vim/

basic config

1. vimdiff: http://www.ibm.com/developerworks/cn/linux/l-vimdiff/
2. expand: http://vimdoc.sourceforge.net/htmldoc/eval.html#expand()
3. http://michael.peopleofhonoronly.com/vim/
4. for ruler, color space, ColorColumn:

    1.
```
:set ruler
:set t_Co=256
:highlight ColorColumn ctermbg=39
:set colorcolumn=80

:highlight Pmenu ctermbg=5
```

useful tips

find a file

1. ":set path?" display path var used in, such as

```
1) :set path+=C:\
2) :set path?
path=.,C:\
```

No.	cmd	description	example
1.	find a file		

1.1	:edit	browse current dir

```
"
==============================================================================
======
" Netrw Directory Listing
(netrw v109)
"   Sorted by      name
"   Sort sequence:
[\/]$,\.h$,\.c$,\.cpp$,*,\.info$,\.swp$,\.o$\.obj$,\.bak$
"   Quick Help: <F1>:help  -:go up dir  D:delete  R:rename  s:sort-by
x:exec
"
==============================================================================
======
../
./
check/
Makefile
autocmd.txt
change.txt
eval.txt~
filetype.txt~
help.txt.info
```

No.	cmd	description	example
1.2	:echo $PATH	dump env var	d:\Program Files\Source Insight 3;d:\Program Files\ARM\ADSv1_2\bin;C:\WINDOWS\system32;
1.3	:set path?	display path var used in vim, used by gf command	1) :set path+=C:\ 2) :set path? path=.,C:\
1.4	**gf**	Edit the file whose name is under or after the cursor.Mnemonic: "goto file" Uses the 'path' option as a list of directory names to look for the file	
1.5	:find <file>	find a file in the path	

buffer operation

No	operaiont	description	example
1	:buffers	show all buffers (the same as :ls)	

```
:buffers
  2
"\mrvl\GaloisSoftware\GaloisSoftware_bak\Drivers\Galois_SoC\VPP\source
\vpp_api.c" 第 511 行
  3
"\mrvl\GaloisSoftware\GaloisSoftware_bak\Drivers\Galois_SoC\Common\inc
```

```
lude\Firmware_Berlin_C2\vpp.h" 第 1 行
  5 %a
"\mrvl\GaloisSoftware\GaloisSoftware_bak\PE\Core\source\core_vchn.c" 第
2673 行
```

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

mark operation

No	mark operation	description	example
1	local file mark		
1.1	ma	define a file local mark "a" use little character	
1.2	'a or `a	got o local file mark "a"	
2	<:>global file mark		
2.1	mA	define a global mark “A” use upper character	
2.2	'A or `A	go to global file mark “A”	
2.3	:marks	display all marks defined	
2.4		trace back to the cursor position before doing a jump	
2.5	CTRL-o	Don't forget that you can use CTRL-O and CTRL-I to jump to older and newer positions without placing marks there.	
2.6	CTRL-i	Don't forget that you can use CTRL-O and CTRL-I to jump to older and newer positions without placing marks there	

Windows relative command

Summary: 1.A buffer is the in-memory text of a file. 2.A window is a viewport on a buffer. 3.A tab page is a collection of windows.

modify window boundary

1. ctrl+w +、ctrl+w -：先按下 ctrl+w 再按下加號或減號，是增加或減少列數，也就是調整上下分割時用的。
2. ctrl+w >、ctrl+w <：先按下 ctrl+w 再按下大於或小於符號，是左右分隔線向右或向左的意思，主要是調整左右分割實用的。
3. 調整視窗垂直大小: <Ctrl-W> [+|-]
4. 可以用20<Ctrl-W>[+|-] 来一次调整20个单位
5. 調整視窗水平大小: <Ctrl-W> [<|>]
6. 將目前視窗垂直打開到最到: <Ctrl-W> _
7. 將目前視窗水平打開到最大: <Ctrl-W> |
8. "x" in Tlist window will toggle Tlist window full screen display

Navigate between windows

1. 切換到下方視窗: <Ctrl-W> j
2. 切換到上方視窗: <Ctrl-W> k
3. 切換到左方視窗: <Ctrl-W> h
4. 切換到右方視窗: <Ctrl-W> l
5. 切換到下一個視窗: <Ctrl-W><Ctrl-W>

locate currently edited file in NERDTree window

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

jump in a file

1. 大写A：光标跳到行尾，并进入insert模式
2. CRTL+^ : line head(first characther of line)
3. CRTL+$: line end
4. 0(zero): column zero of line
5. CRTL+o : go back to previous edit point
6. CRTL+i : go forward to next edit point
7. G: file end
8. gg: file head
9. CTRL+]: jump to define <within ctags and cscope>
10. CTRL+t: back to pre-jump tag<within ctags and cscope>
11. \\w: show beginning of word forward and then select to jump <within easymotion plugin>
12. \\b: show beginning of work backard and then select to jump <within easymotion plugin>
13. o(not zero): enter a new line below current line and enter insert mode
14. O(not zero, big cap): enter a new line above current line and enter insert mode
15. %: map 1) #if-#else-#endif 2){}
16. [#: go to end of #if-#else-#endif

17.]#: go to begin of #if-#else-#endif
18. [[: jump to front of function body or front of previous function body
19. []: jump to end of function body
20.]]: jump to the front of next function body

screen scrolling

1. CTRL-U: 命令会使文本向下滚动半屏
2. CTRL-D: 命令将窗口向下移动半屏
3. CTRL-F: full screen move forward
4. CTRL-B: full screen move backworad
5. zz: make the current line scroll to the center of screen
6. zt: make the current line top of the screen
7. zb: make the current line bottom of the screen

need not confused with folding command

file format stuff

1. 回想计算机的史前史, 那时的打字机使用两个字符来开始一个新行. 首先是一个字符命令使打印头移回开始位置(回车, <CR>), 然后另一个字符命令控制向前进纸一行(进纸, <LF>).
2. 在计算机诞生之初, 存储设备十分昂贵. 于是有人就提出没有必要用两个字符来表示一行的结束. UNIX一族决定只用进纸一个字符<LineFeed>来表示行尾. 来自苹果阵营的人则把回车<CR>作为换行的标准.MS-DOS(和微软的Windows)仍然决定沿用古老的回车换行<CR><LF>传统.
3. Vim以下面的名字代表三种不同的格式:

unix	<LF>
dos	<CR><LF>
mac	<CR>

fileformats

1. :set fileformats=unix,dos ->这个命令就可以让Vim能自动识别UNIX格式和MS-DOS格式
2. :set fileformat? =>show current file format
3. :edit ++ff=unix file.txt ->Vim也允许你强制指定文件格式: "++"字符串告诉Vim后面紧接着的是一个选项名, 对该选项的设置将覆盖它的默认值. "++ff"代表的选项是°fileformat°. 你也可以指定为"++ff=mac"或"++ff=dos".不过并不是每个选项都有这种用法, 目前来说只有"++ff"和"++enc"可以这样用.当然也可以用这两个选项的全称"++fileformat"和"++encoding".

change file format

1. 你也可以利用°fileformat°选项来转换文件的格式. 假如你有一个MS-DOS格式的文件README.TXT. 现在你想把它转换为UNIX格式:
2. vim README.TXT
3. Vim会识别出这是一个dos格式的文件. 现在把它变为UNIX格式的
4. :set fileformat=unix
5. :write

binary file edit

general binary file operation

1. vim -b bin/java => 查看二进制文件
2. :set display=uhex => 转换成16进制显示
3. :set display= =>转换回default显示方式

use xxd to view binary file

1. vim -b bin/java => 查看二进制文件
2. :%!xxd => Show file in the following form

```
0000000: 1f8b 0808 39d7 173b 0203 7474 002b 4e49 ....9..;..tt.+NI
0000010: 4b2c 8660 eb9c ecac c462 eb94 345e 2e30 K,.....b..4^.0
0000020: 373b 2731 0b22 0ca6 c1a2 d669 1035 39d9 7;°1."".....i.59.
```

1. :%!xxd -r =>revert back to original display mode
2. :%!xxd --help

```
Usage:
       xxd [options] [infile [outfile]]
    or
       xxd -r [-s [-]offset] [-c cols] [-ps] [infile [outfile]]
Options:
    -a          toggle autoskip: A single '*' replaces nul-lines.
Default off.
    -b          binary digit dump (incompatible with -ps,-i,-r).
Default hex.
    -c cols     format <cols> octets per line. Default 16 (-i: 12, -
ps: 30).
    -E          show characters in EBCDIC. Default ASCII.
    -g          number of octets per group in normal output. Default
2.
    -h          print this summary.
    -i          output in C include file style.
    -l len      stop after <len> octets.
    -ps         output in postscript plain hexdump style.
```

```
    -r          reverse operation: convert (or patch) hexdump into
binary.
    -r -s off   revert with <off> added to file positions found in
hexdump.
    -s [+][-]seek  start at <seek> bytes abs. (or +: rel.) infile
offset.
    -u          use upper case hex letters.
    -v          show version: "xxd V1.10 27oct98 by Juergen Weigert".
```

```
#echo "edf14d57ab" | xxd -r -p > h.bin
tinge@tinge-server:~/test$ xxd -g1 h.bin
0000000: ed f1 4d 57 ab                                    ..MW.
```

trace src callstack with mark

1. go through the function callstack, and add a global mark on every function in call sequence
2. mA, mB
3. `A, B`

replace/Substitute

1. :%s/{pattern}/{string}/g
2. 删除每行的前5个字符 :%s/^.\{5}//gic
3. substitute

syntax highlight for arm asm

1. set vim xterm-256color display=>:set t_Co=256

high light for arm asm

listchars

1. enable listchars
    1. set listchars=tab:>-,trail:-
2. disable listchars
    1. set nolist

plentiful plugin

1. for plugin config and usage detail, refer to the word doc in the attachment, here only list

some tips for different plugin
2. the following sub title is some highlight features and function key for some plugins

easymotion plugin

1. \\w: Beginning of word forward
2. \\b: Beginning of word backward.
3. \\e: End of word forward.
4. \\ge: End of word backward
5. \\j: Line downward
6. \\k: Line upward.

session manager plugin

1. :SessionList-> list all session saved
2. :SessionSave-> command saves the current editing session. If v:this_session is empty it asks for a session name
3. :SessionClose-> command wipes out all buffers, kills cscope and clears variables with session name

loopup file plugin

1. let g:LookupFile_AllowNewFiles = 0 "not allowed to create new file which doesn't exists

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>

DirDiff plugin

1. vim . -c ":DirDiff DIRECTORY_A DIRECTORY_B"

Commands can be used inside the diff window

1. enter or o: - Diff open: open the diff file(s) where your cursor is at
2. s: Synchronize the current diff. You can also select a range (through visual) and press 's' to synchronize differences across a range.
3. u: - Diff update: update the diff window

4. x: - Sets the exclude pattern, separated by ','
5. i: - Sets the ignore pattern, separated by ','
6. a: - Sets additional arguments for diff, eg. -w to ignore white space,
7. q: - Quit DirDiff

The following comamnds can be used in the Vim diff mode

1. \dg - Diff get: maps to :diffget<CR>
2. \dp - Diff put: maps to :diffput<CR>
3. \dj - Diff next: (think j for down)
4. \dk - Diff previous: (think k for up)

config

1. Sets default exclude pattern:
    1. let g:DirDiffExcludes = "CVS,*.class,*.exe,.*.swp"
2. Sets default ignore pattern:
    1. let g:DirDiffIgnore = "Id:,Revision:,Date:"

problem met

ctrl-s and ctrl-q

1. windows下的编辑器使用惯了, 今天使用vim, 无意中又按了一下CTRL+S, 结果vim像停掉了一样, 按什么键都不起作用了. 以前也碰到这种情况, 解决的办法是直接关了ssh客户端软件, 然后重新连接, 重新打开那个文件.
2. 今天我直接在google上查了一下, 发现直接按CTRL+q解决问题.
3. 问题的原因: CTRL+s表示停止向终端输出; CTRL+q恢复向终端输出

record and q

1. when enter record status, press "q" will exit

vimdiff stuff

1.]c: 跳转到下一个diff点:]c 1.[c: 跳转到前一个diff点: [c 1.2]c: 如果在命令前加上数字的话，可以跳过一个或数个差异点，从而实现跳的更远。比如如果在位于第一个差异点的行输入"2]c"，将越过下一个差异点，跳转到第 三个差异

点。 1.dp: 当前文件的内容复制到另一个文件里: dp(diff put) 1.do: 如果希望把另一个文件的内容复制到当前行 中，可以使用命令：do (diff "get"，之所以不用dg，是因为dg已经被另一个命令占用了，所以用了diff "obtain") 1.directly edit: 如果希望手工修改某一行，可以使用通常的vim操作。 1.diffupdate: 在修改一个或两个文件之后，vimdiff会试图自动来重新比较文件，以便实时地反映比较结果。但是有时候会处理失败，这个时候就需要手工来刷新。 :diffupdate

- zo:（folding open）打开折叠代码。之所以用z这个字母，是因为它看上去比较像折叠着的纸:)
- zc:（folding close）重新折叠起来
- 补充一条： 如果想交换上/下、左/右两个分隔窗口的位置，可以使用 ctrl-w,r 命令

useful function shortcut

1. 字母调换: xp
    1. example: #defien=>#define
2. 大小写转换: ~
    1. example: #**d**efine=>#**D**efine

.vimrc

1. I see <leader> in many .vimrc files, and I am wondering what the meaning of it is? What is it used for? Just a general overview of the purpose and usage.
    1. The <Leader> key is mapped to \ by default. So if you have a map of <Leader>t, you can execute it by default with \t. For more detail or re-assigning it using the mapleader variable, see :help leader

search in vim

brief

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

vim (last edited 2014-05-21 08:21:30 by tingezhang)