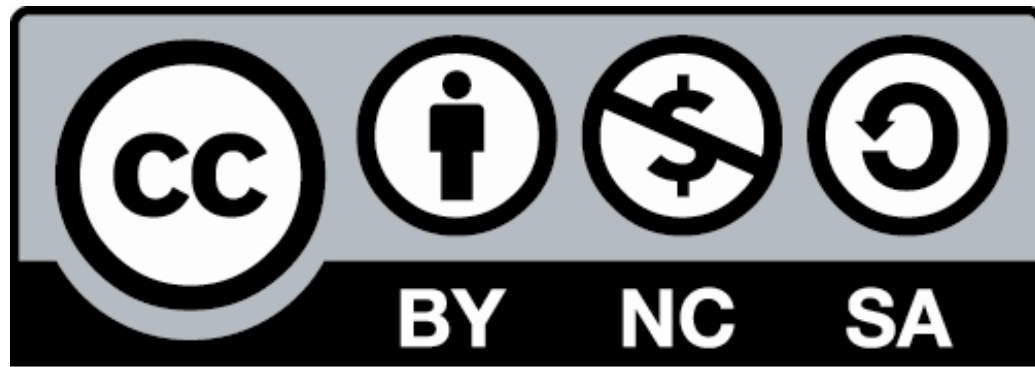


# Make Your VIM & Screen Environment Comfortable

yzlin <[yzlin@cs.nctu.edu.tw](mailto:yzlin@cs.nctu.edu.tw)>

Aug. 7, 2008



本文件採用：  
創用 CC 「姓名標示—非商業性—相同方  
式分享 2.5 台灣版」授權條款



# Outline

- Introduction
- Part 1 – Gnu Screen
- Part 2 – VIM
- Reference

# Introduction

- Do you think your programming environment **COMFORTABLE?**
- “**More comfortable, more powerful**”
- This sharing is about **my Tips & Experience** in setting Screen & VIM making programming environment more comfortable.



# Part 1 – Gnu Screen

# Status Line

- Do you use status line on Screen?

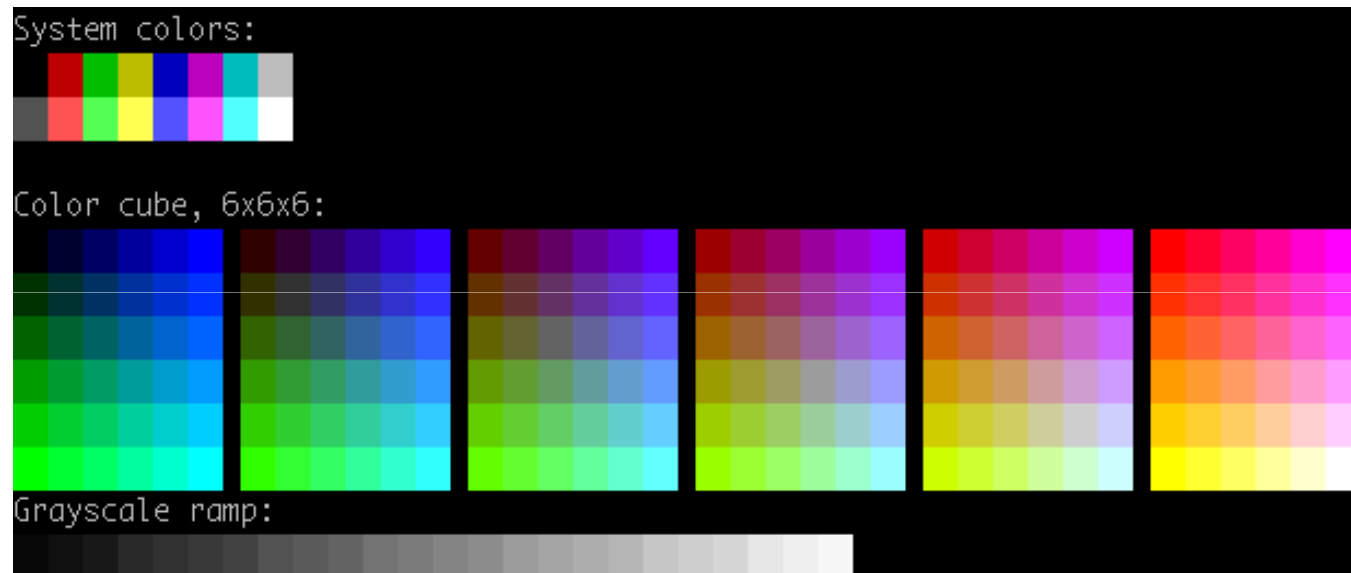
```
0 IRC 1 CSMail 2 ptt 3 ptt2 4 bs2 5 binary 6 MSN 7 BT 8 jm [9] ---  
[ Fri Aug 08 12:00am PHP Training ] UTF-8 stucgi 9:44 07/08/2008 Load: 0.00 0.00 0.00
```

- There are so many info it could show.
- Stop using **Ctrl-a** **"/Ctrl-a w**
- Example: (`$HOME/.screenrc`)

```
caption always "%{.KW} %-w%{.mW}[%n] %t%{.KW}%+w"  
hardstatus alwayslastline "%{R}[ %{w}%1` %{R}]%= %{M}%e  
%{G}%H %{Y} %c %{R} %d/%m/%Y %{K} Load: %1 %{w}"
```

# 256-Color

- 256-color chart in Screen



- Example: (`$HOME/.screenrc`)

```
termcapinfo xterm "Co#256:AB=\E[48;5;%dm:AF=\E[38;5;%dm"  
term $TERM # depend on $TERM  
term xterm # or specified term
```

# Split

```
[10:15am yzlin@vip ~]% ls
bin/ depot/ dev/ doc/ fs/ key/ public_html/ tmp/ www/
[10:15am yzlin@vip ~]%
```

---

```
0 IRC 1 CSMail 2 ptt 3 ptt2 4 bs2 5 binary 6 MSN 7 BT 8 jm [9] --- 10 ---
[10:12am yzlin@stucgi ~]%
```

---

```
0 IRC 1 CSMail 2 ptt 3 ptt2 4 bs2 5 binary 6 MSN 7 BT 8 jm 9 --- [10] ---
[ Fri Aug 08 12:00am PHP Training ] UTF-8 stucgi 10:13 07/08/2008 Load: 0.00 0.00 0.00
```

# Split

- Split window commands:
  - **Ctrl-a S** for splitting
  - **Ctrl-a <Tab>** to switch between windows
  - **Ctrl-a Q** to close splitting windows
- Hotkeys for resizing: (\$HOME/.screenrc)

```
bind = resize =  
bind + resize +1  
bind - resize -1  
bind _ resize max
```

# Quickly Switch

- Hotkeys for quickly switching:
  - **Ctrl-a** *n* to switch to window *n*
  - **Ctrl-a Ctrl-a** to switch between 2 windows
- Want to switch to prev/next window quickly?
  - **Ctrl-a <Left>** for prev window
  - **Ctrl-a <Right>** for next window

```
bindkey \033[D prev  
bindkey \033[C next
```



# More Features?

- So long without updating since ver. 4.0.3, this project relaunched again recently.
- New features:
  - Group windows?
  - User scripts?
  - Vertical Splitting
- We'll see in the future.



# Part 2 – VIM

The most powerful editor in the world



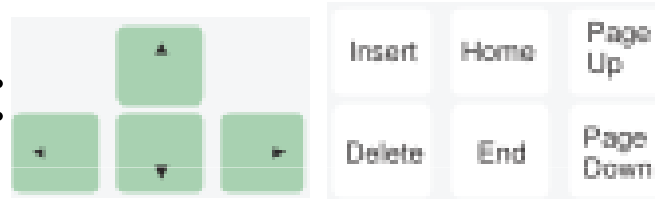
# Intro of VIM

- Why I love VIM?
  - It's incredibly powerful!!
  - I love KEYBOARD much more than MOUSE
  - Free your fingers

# Get Away From This!

- Before starting, let's abandon some BAD habits

- Don't touch these:



- **h/j/k/l** for moving
- **Ctrl-f** for page down, **Ctrl-b** for page up
- **o** for home, **\$** for end

# Move More Quickly

- Start/End of file?
  - `gg` & `G`
- `n` lines from start/end of file?
  - `ngg` & `nG`
- `n%` of file?
  - `n%`

# Quick Quit

- Are you still using these?
  - `:wq<CR>` (exit with save)
  - `:x<CR>` (exit with save)
  - `:q!<CR>` (exit without save)
- Try some more quickly:
  - `ZZ` (exit with save)
  - `ZQ` (exit without save)

# Insert/Replace/Delete

- Basic Commands
  - **i/I**
  - **a/A**
  - **o/O**
  - **s/S**
  - **c/C**
  - **r/R**
  - **d/D**

# Insert/Replace/Delete

- `gi` to go back to where you edit last time
- `cw/ciw` to replace a word
- `dw/diw` to delete a word
- `ci"/ci'/ci[/ci{/ci(/cit`
- `di"/di'/di[/di{/di(/dit`

# Insert/Replace/Delete

- Example:
  - `gUi"` – uppercase text inside double-quoted string
  - `gui[` - lowercase text inside []

# Search

- Make search easily:

- Wrap search

```
set wrapscan
```

- Incremental search

```
set incsearch
```

- Case-insensitive search

```
set ignorecase
```

# Encoding

- 3 major encoding setting in VIM:
  - fileencoding
    - actual encoding for file
  - termencoding
    - which encoding would show on terminal
  - encoding
    - encoding inside VIM
- File encoding priority
  - fileencodings

# Encoding

- fileencoding

```
set fileencoding=utf-8
```

- termencoding

```
set termencoding=utf-8
```

- encoding

```
set encoding=utf-8
```

- fileencodings

```
set fileencodings=utf-8, big5, euc-jp, gbk, euc-kr, utf-  
bom, iso8859-1
```

# Encoding

- Ambiguous Width (only work in UTF-8)

```
set ambiwidth=double
```

- Which will treat the widths of ambiguous chars **DOUBLE**. Default is **SINGLE**.

- Make sure **+iconv**, showing by:

```
vim --version
```

# 256-Color Theme

- VIM supports 256-color scheme

```
set t_Co=256  
syntax on
```

- Make your own color scheme
  - <http://vip.cs.nctu.edu.tw/~yzlin/256Color/256-xterm-24bit-rgb-color-chart.html>
- Or download one:
  - <http://frexx.de/xterm-256-notes/>
  - My color scheme -  
<http://vip.cs.nctu.edu.tw/~yzlin/256Color/yzlin256.vim>

# Visual Mode

- 3 selection in Visual Mode:
  - **v** – character selection
  - **V** – line selection
  - **Ctrl-v** – block selection
- Group insertion:
  - **Ctrl-v** for start/end, then **I** to insert.

# Windows

```
3 md_high.c
77
78 //==== Setup Macroblock encoding
  parameters ====
79 init_enc_mb_params(currMB, &enc_mb, intra,
  bslice);
80
81 // reset chroma intra predictor to default
82 currMB->c_ipred_mode = DC_PRED_8;
83
84 //==== STORE CODING STATE
  E ====
85 //-----
  -----
86 store_coding_state (currMB, cs_cm);
87
88 if (!intra)
89 {
90 //==== set direct motion vectors ====
91 best_mode = 1;
92 if (bslice)
93 {
94 Get_Direct_Motion_Vectors (currMB);
95 }
96
67 if(input->SearchMode == UM_HEX)
68 {
69 UMHEX_decide_intrabk_SAD();
70 }
71 else if (input->SearchMode == UM_HEX_SIMPLE)
72 {
73 smpUMHEX_decide_intrabk_SAD();
74 }
75
76 intra l= RandomIntra (img-
>current_mb_nr); // Forced Pseudo-Random
  Intra
<m=1,C 15%* Aug,07 08 13:53:21
20 #include "transform8x8.h"
21 #include "ratectl.h"
22 #include "mode_decision.h"
23 #include "fmo.h"
24 #include "me_umhex.h"
25 #include "me_umhex_smp.h"
26 #include "macroblock.h"
27
28 /*!
29 *****
  *****
<m=1,C 17%* Aug,07 08 13:52:47 <m=1,C 4%* Aug,07 08 13:53:17
0 IRC 1 CSMail 2 ptt 3 ptt2 4 bs2 5 binary 6 MSN 7 BT [8] jm 9 ---
[ Fri Aug 08 12:00am PHP Training ] UTF-8 stucgi 13:53 07/08/2008 Load: 0.19 0.12 0.07
```

# Windows

- Horizontal/Vertical Splitting
  - `Ctrl-w s/v`
- Switching window
  - `Ctrl-w h/j/k/l`
- Cyclic switching
  - `Ctrl-w Ctrl-w`
- Window resize
  - `Ctrl-w +/-/=/_`

# Tabs

- Tab bar



```
md_high.c azb.c biariencode.c
1 |
2 | /*!
3 | *****
4 | * \file biariencode.c
5 | *
```

- Tab commands
  - `:tabe <file>` to edit `<file>` in a new tab
  - `:tabnew` to create a new file in a new tab
  - `:tabc` to close
  - `:tabp, :tabn`

# Tabs

- Hotkeys:

```
nmap <C-T>c :tabnew<CR>  
nmap <C-T>k :tabclose<CR>  
nmap <C-H> :tabprev<CR>  
nmap <C-L> :tabnext<CR>
```

# Spell Check

- Spell checking on/off:

```
this s the tset paragraf.  
this is the test paragraph.
```

- `:set spell/:set nospell`
- `:set spelllang=en` for change spell language to *English*
- Highlight spell check error

```
hi SpellBad term=underline cterm=underline ctermfg=red
```

- Hotkeys

```
map <F5> :set spell!<CR><BAR>:echo "Spell check: " .  
strpart("OffOn", 3 * &spell, 3)<CR>
```

# miscellaneous

- See partial commands as you type

```
set showcmd
```

- Backup

```
set backup  
Set backupdir=$HOME/.vim/backup
```

- <Tab>
  - `:set tabstop=4`
  - `:set shiftwidth=4`
  - `:set softtabsop=8`

# miscellaneous

- <Tab> hotkey – visual <Tab>

```
noremap ,st :set list listchars=tab:>-<CR>  
noremap ,ct :set nolist<CR>
```

- Folding

```
set foldenable
```

- Select first then : '<, '>fo
- Folder close/open – zc/zo
- Hotkey

```
nnoremap <SPACE> @=((foldclosed(line('.')) < 0) ? 'zc' :  
'zo')<CR>
```

# miscellaneous

- Last editing position

```
autocmd BufReadPost *  
  \ if line("'\"") > 0 && line("'\"") <= line("$") |  
  \ exe "normal g`\"" |  
  \ endif
```

- Net edit

```
vim scp://HOST/PATH/TO/FILE
```

# Programming Assistant

- Show match braces

```
set showmatch
```

- Highlight match pairs

```
hi MatchParen ctermbg=Yellow
```

- % to move between matching items `()`, `[]`, `{}`,  
etc.

# Programming Assistant of PHP

- Highlight SQL syntax in string

```
echo "DELETE FROM `test` WHERE `name` = 'test'";
```

```
let php_sql_query=1
```

- Highlight HTML in string

```
echo "<h2>title</h2><table><tr><td>cell1</td></tr></table>";
```

```
let php_htmlInStrings=1
```

- Although...sometime it make things worse...

```
echo 'the order is not right';
```

# Programming Assistant of PHP

- Folding for classes & functions

```
let php_folding=1
```

- Disable short tags

```
let php_noShortTags=1
```

# Plugin – NERDTree

```
md_high.c azb.c 2 1218092358_NERD_tree_
1 | Press ? for help | 1
2 | | | 2 /*|
3 | .. (up a dir) | 3 *****
4 | /amd/account/gcs/96/9655630 | *****
5 | l-annexb.c | 4 * \file biariencode.c
6 | l-azb.c | 5 *
7 | l-biariencode.c | 6 * \brief
8 | l-block.c | 7 * Routines for binary arithmetic encoding.
9 | l-cabac.c | 8 *
10 | l-configfile.c | 9 * This modified implementation of the M Coder is based on JVT-
11 | l-context_ini.c | U084
12 | l-decoder.c | 10 * with the choice of M_BITS = 16.
13 | l-explicitGolomb.c | 11 *
14 | l-filehandle.c | 12 * \author
15 | l-fmo.c | 13 * Main contributors (see contributors.h for copyright,
16 | l-header.c | address and affiliation details)
17 | l-image.c | 14 * - Detlev Marpe <marpe@hhi.de>
18 | l-img_chroma.c | 15 * - Gabi Blaettermann
19 | l-img_luma.c | 16 * - Gunnar Marten
20 | l-intrarefresh.c | 17 *****
21 | l-leaky_bucket.c | *****
22 | l-lencod.c | 18 */
23 | l-lencod.c.orig | 19
24 | l-loopFilter.c | 20 #include "global.h"
< Top* Aug,07 08 14:59:18 <00-1,totlin=481,BfNm=3,C Top* Aug,07 08 14:59:18
:NERDTreeToggle
0 IRC 1 CSMail 2 ptt 3 ptt2 4 bs2 5 binary 6 MSN 7 BT [8] jm 9 --- 10 ---
[ Fri Aug 08 12:00am PHP Training ] UTF-8 stucgi 14:59 07/08/2008 Load: 0.26 0.10 0.10
```

# Plugin – autocomplpop

```
md_high.c + azb.c biariencode.c
12
13 static int group_map[4][4] = {
14     {1, 2, 2, 1},
15     {4, 3, 3, 4},
16     {4, 3, 3, 4},
17     {1, 2, 2, 1}
18 };
19
20 static imgpel *src_line, *ref_line, *ref_line1, *ref_line2;
21
22 pr[]
23 propagate_carry biariencode.c
24 processed biariencode.c
25 probability biariencode.c *****
26 prob biariencode.c
27 pre biariencode.c
28 prev_mb_nr md_high.c *****
29 prevMB md_high.c
30 predictor md_high.c teMvList***** cand_azb)
31 prediction md_high.c
32 precompute md_high.c
33
34 if ((*cand_azb = (AzbCandidateMvList*****))calloc(img->height_blk,
    sizeof(AzbCandidateMvList*****)) == NULL)
<02 08 00:19:47)[+]line=022,col=04,totlin=275[+],BfNm=2,C 4%* Aug,07 08 15:01:10
match in file biariencode.c
-- Keyword completion (^N^P) match 1 of 10
0 IRC 1 CSMail 2 ptt 3 ptt2 4 bs2 5 binary 6 MSN 7 BT [8] jm 9 --- 10 ---
[Fri Aug 08 12:00am PHP Training ] UTF-8 stucgi 15:01 07/08/2008 Load: 0.08 0.08 0.08
```

# Plugin – snippetsEmu

```
1 <?php
2 class task
3 {
4     public function test()
5     {
6         for ( $i=«»; $i < «»; $i++ )
7         {
8             «»
9         }
10        «»
11    }
12 }
13 ?>
```



# Reference

# Reference

- Vim tips
  - <http://www.vim.org/tips/index.php>
  - [http://vim.wikia.com/wiki/Main\\_Page](http://vim.wikia.com/wiki/Main_Page)
- Vim scripts: <http://www.vim.org/scripts/index.php>
- Plugins
  - NERDTree - [http://www.vim.org/scripts/script.php?script\\_id=1658](http://www.vim.org/scripts/script.php?script_id=1658)
  - autocmplpop - [http://www.vim.org/scripts/script.php?script\\_id=1879](http://www.vim.org/scripts/script.php?script_id=1879)
  - snippetsEmu - [http://www.vim.org/scripts/script.php?script\\_id=1318](http://www.vim.org/scripts/script.php?script_id=1318)
  - taglist - [http://www.vim.org/scripts/script.php?script\\_id=273](http://www.vim.org/scripts/script.php?script_id=273)
  - Surround - [http://www.vim.org/scripts/script.php?script\\_id=1697](http://www.vim.org/scripts/script.php?script_id=1697)
- “VIM for (PHP) Programmers” by Andrei Zmievski