



个人资料



wooin

- 访问: 905696次
- 积分: 4929分
- 排名: 第1571名
- 原创: 27篇 转载: 83篇
- 译文: 0篇 评论: 807条

文章搜索

文章分类

- C & C++(2)
- C & C++(9)
- Firefox Dev Maillist(0)
- Linux(51)
- Linux(23)
- Microwindows/Nano-X(2)
- Movie(1)
- Other(7)
- Other(7)
- VB(1)
- VB(0)
- 宗教历史(6)

文章存档

- 2008年09月(1)
- 2008年08月(1)
- 2008年06月(1)
- 2008年05月(2)
- 2008年04月(1)

展开

阅读排行

- 手把手教你把Vim改装成一个IDE编程环境(图文) (444827)

博客专家福利 【限时活动】建专辑得大奖 专访荣浩: 流程的永恒之道 当青春遇上互联网, 能否点燃你的创业梦 推荐有礼--找出您心中的技术大牛

手把手教你把Vim改装成一个IDE编程环境(图文)

<http://blog.csdn.net/wooin/article/details/1858917>

分类: Linux 2007-10-31 10:15 444952人阅读 评论(539) 收藏 举报
vim编程idebuffersearchtags

手把手教你把Vim改装成一个IDE编程环境(图文)

By: 吴垠
Date: 2007-09-07
Version: 0.5
Email: lazy.fox.wu#gmail.com
Homepage: <http://blog.csdn.net/wooin>
Copyright: 该文章版权由吴垠和他可爱的老婆小包子所有。可在非商业目的下任意传播和复制。对于商业目的下对本文的任何行为需经作者同意。
联系方式: lazy.fox.wu#gmail.com

1 写在前面

Linux下编程一直被诟病的一点是: 没有一个好用的IDE, 但是听说Linux牛人, 黑客之类的也都不用IDE. 但是对我等从Windows平台转过来的Coder来说, 一个好用的IDE是何等的重要啊, 估计很多人就是卡在这个门槛上了, "工欲善其事, 必先利其器"嘛, 我想如果有一个很好用的IDE, 那些Linux牛人也会欢迎的. 这都是劳动人民的美好愿望罢了, 我今天教大家把gvim改装成一个简易IDE, 说它"简易"是界面上看起来"简易", 但功能绝对不比一个好的IDE差, 该有的功能都有, 不该有的功能也有, 你就自己没事偷着乐吧, 下面我开始介绍今天的工作了.

本文会教你:

- 中文帮助手册的安装
- vim编程常用命令
- 语法高亮
- 在程序中跳来跳去: Ctags 的使用
- 教你高效地浏览源码 -- 插件: TagList
- 文件浏览器和窗口管理器 -- 插件: WinManager
- Cscope 的使用
- QuickFix 窗口
- 快速浏览和操作Buffer -- 插件: MiniBufExplorer
- c/h文件间相互切换 -- 插件: A
- 在工程中查找 -- 插件: Grep
- 高亮的书签 -- 插件: VisualMark
- 自动补全
- 加速你的补全 -- 插件: SuperTab

本文不会教你:

- 如何使用vim. 本文不会从零开始教你如何使用vim, 如果你是第一次接触vim, 建议你先看看其他的vim入门的教程, 或者在shell下输入命令: vimtutor,



- 我的vimrc(30570)
- 在Redhat Linux 9中编译和配置gvim 7.0(29720)
- 手把手教你把Vim改装成一个IDE编程环境(图文) -- 续1(25375)
- 《手把手教你把Vim改装成一个IDE编程环境 Ver: 0.7》下载(21567)
- 在Linux 2.6内核下编译可以加载的内核模块(18894)
- 佛和道的区别(17060)
- 伊斯兰教、基督教历史与达芬奇密码(15480)
- 在Redhat9 Linux下安装,配置Subversion 1.3.1(13563)
- 大唐惊变(12993)

评论排行

- 手把手教你把Vim改装成一个IDE编程环境(图文)(538)
- 手把手教你把Vim改装成一个IDE编程环境(图文) -- 续1(54)
- 《手把手教你把Vim改装成一个IDE编程环境 Ver: 0.7》下载(50)
- 我的vimrc(28)
- 在Redhat9 Linux下安装,配置Subversion 1.3.1(20)
- 在Redhat Linux 9中编译和配置gvim 7.0(12)
- 自己写的几个简单常用的脚本(10)
- 用Microwindows(Nano-X)编写"hello world"(10)
- 《Linux设备驱动程序(第三版)》学习笔记之一: scull设备的使用(10)
- 我们也应该支持这些地区独立(9)

推荐文章

- * 互联网创业梦已醒
- * ANDROID内存优化
- * Ken Burns特效的幻灯片
- * Android ViewGroup事件分发机制
- * 布局优化利器和ViewStub
- * Android仿QQ5.0侧滑菜单ResideMenu源码分析

最新评论

- 手把手教你把Vim改装成一个IDE编程环境(图文) udc577: 经实测, :help taglist.txt应该是:help taglist
- 手把手教你把Vim改装成一个IDE编程环境(图文) Chinarc: @Chinarc:然后直接按wm就可以显示
- 手把手教你把Vim改装成一个IDE编程环境(图文) Chinarc: @Devil_2009:"-- WinManager setting --let g:winMana...
- 手把手教你把Vim改装成一个IDE编程环境(图文) Chinarc: 2、WinManager没起作用! 打开文件时出现: vi sem3.c找不到映射请按 ENTER 或其...

这是一个简单的入门教程.

- 编程技巧.
- vim脚本的编写.

我的工作环境是: Fedora Core 5

gvim是自己编译的7.0, 如果你还没有安装gvim, 请看我的这篇文章<在Redhat Linux 9中编译和配置gvim 7.0>

由于本人一直从事C语言工作, 所以下面这些例子都是在C语言程序中演示的, 其他语言的没有试过, 如果有朋友在别的语言下有问题, 可以跟我讨论一些, 我会尽量帮助你们的.

本文用的示范源码是vim7.1的源码, 可以在www.vim.org下载到源码包:vim-7.1.tar.bz2, 你也可以不用下载, 就用你自己程序的源码, 关系不大的. 我把源码解压在我的home目录下: ~-vim71

下面对文中的一些名字定义一下:

- 文中用到的一些用<>括起来的符号比如<C-T>, <C-S-A>, 之类的, 你可以用下面的命令看看解释:

```
:help keycodes
```

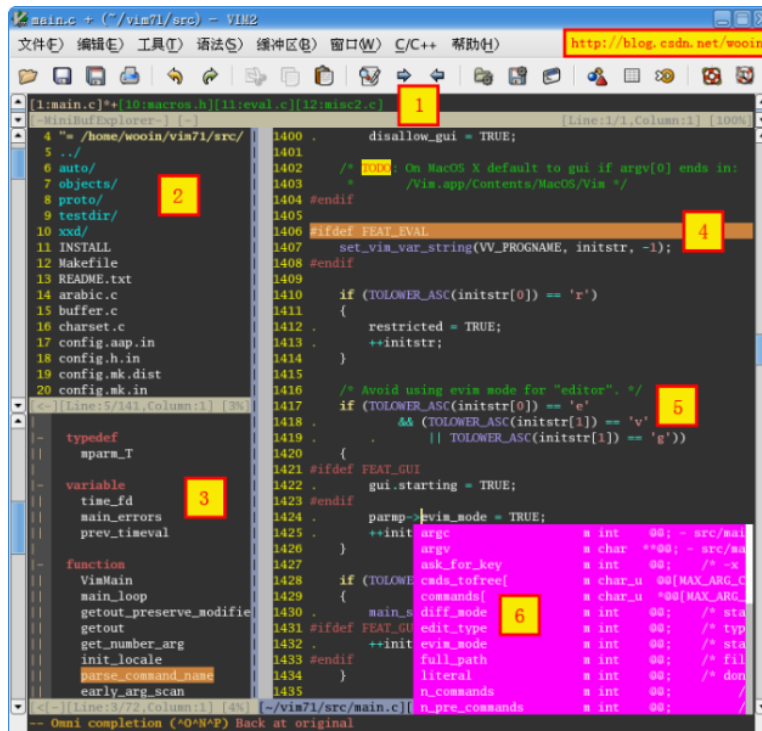
- 文中说的一些路径, 比如:

```
~/vim/plugin
~/vim/doc
~/vim/syntax
```

如果你没有, 就自己创建.

- 文中说到的.vimrc文件都是指 ~/vimrc

先给大家看张图片, 我是vim的界面, 解解馋先^_^



(--- 图1 ---)

对照上图的图标, 我在本文中将会教你以下这些功能:

1	简洁明了的Buffer浏览和操作
2	文件浏览器
3	tag浏览器
4	高亮的书签
5	更丰富的语法高亮
6	成员变量的下拉, 自动补全



- 手把手教你把Vim改装成一个IDE编程环境(图文)
hlcshi: 高亮运算符一句不就不就搞定: syn match cOperator display ...
- 手把手教你把Vim改装成一个IDE编程环境(图文)
hlcshi: @Devil_2009:"/"改成"\"。
- 手把手教你把Vim改装成一个IDE编程环境(图文)
hlcshi: 2007年, 我还在干啥呀! 快速补全, 一般tab缩进只要在行首用到。写个函数定义tab键就行了, 也不用...
- 手把手教你把Vim改装成一个IDE编程环境(图文)
飞飞哥儿: 太牛逼额
- 手把手教你把Vim改装成一个IDE编程环境(图文)
魂--: 很好, 看了网上其他人的一直没搞定。L Z可以发一份配置文件给我吗?
214717161@qq.com ...
- 手把手教你把Vim改装成一个IDE编程环境(图文)
gbchqc: @Devil_2009:2. winmanager没起作用, 是因为nmap wm :WMToggle...

Linux

- Firefox中文网



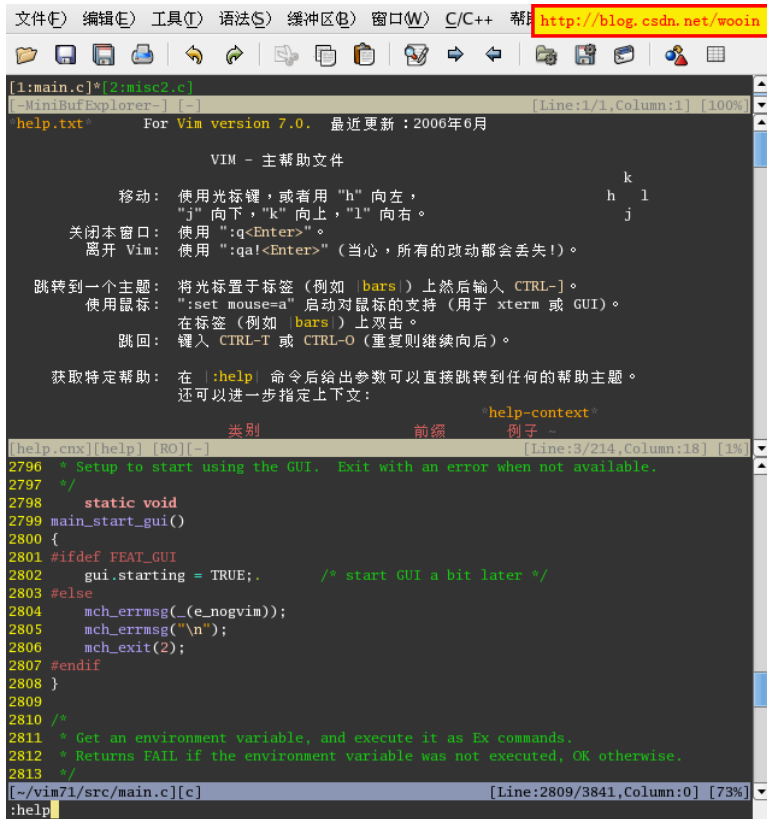
- Firefox 中文官方网站
- ZCLinux -- 资讯与技术社区



- Mozilla Firefox中文社区

vim自带的帮助手册是英文的, 对平时编程的人来说没有多大阅读困难, 何况还有“星级译王”呢, 可偏偏有一帮人将其翻译成了中文, 可偏偏我又挡不住诱惑将它安装了, 唉.....又痛失一个学习英文的好机会, 下不为例。

大家看看我的中文帮助界面吧:



(--- 图2 ---)

安装方法:

在下面的网站下载中文帮助的文件包:

<http://vimcdoc.sf.net> (English)

<http://vcd.gro.clinux.org> (中文)

下载的文件包应该是类似这样的: vimcdoc-1.5.0.tar.gz

解压后其中有个doc文件夹, 将其中的内容全部复制到~/vim/doc, 或者vim安装目录下的doc目录中, 此时vim中的help信息已经是中文的了。

注意:

- 如果无法显示中文, 在~/vimrc中增加下面这句试试:

```
set helplang=cn
```

- 帮助文件的文本是utf-8编码的, 如果想用vim直接查看, 需要在~/vimrc中设置:

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

3 vim编程常用命令

建议先看看帮助手册中的下面章节, 其中有关tags文件的部分你可以先跳过, 在后面的章节中会讲到, 到时候你在回来看看, 就觉得很简单了:

```
:help usr_29
:help usr_30
```

下面是我常用的一些命令, 放在这里供我备忘:

%	跳转到配对的括号去
[[跳转到代码块的开头去(但要求代码块中{必须独占一行)
gD	跳转到局部变量的定义处
"	跳转到光标上次停靠的地方, 是两个', 而不是一个"
mx	设置书签,x只能是a-z的26个字母
`x	跳转到书签处("`"是1左边的键)
>	增加缩进,"x">"表示增加以下x行的缩进
<	减少缩进,"x"<"表示减少以下x行的缩进

写程序没有语法高亮将是一件多么痛苦的事情啊, 幸亏vim的作者是个程序员(如果不是, 那可NB大了), 提供了语法高亮功能, 在上面的图片中大家也可以看到那些注释, 关键字, 字符串等, 都用不同颜色显示出来了, 要做到这样, 首先要在你的 ~/.vimrc 文件中增加下面几句话:

```
syntax enable
syntax on
```

再重新启动vim, 并打开一个c程序文件, 是不是觉得眼前突然色彩缤纷了起来...

如果你不喜欢这个配色方案你可以在"编辑->配色方案"(gvim)中选择一个你满意的配色方案, 然后在~/.vimrc文件中增加下面这句:

```
colorscheme desert
```

desert是我喜欢的配色方案, 你可以改成你的. 如果菜单中的配色方案你还不满意(你也太花了吧), 没关系, 在 vim.org 上跟你一样的人很多, 他们做了各种各样的颜色主题, 你可以下载下来一个一个的试, 多地可以看到你眼花. 如果这样你还不满意(你还真是XXXX), 没关系, vim的作者早想到会有你这种人了, 你可以创建你自己的颜色主题, 把下面的这篇文档好好学习一下吧:

```
:help syntax.txt
```

更炫的语法高亮:

你可能会发现很多东西没有高亮起来, 比如运算符号, 各种括号, 函数名, 自定义类型等, 但是看上面的图片, 我的运算符号和函数名都加亮了^_^, 想知道为什么吗? 哇哈哈哈哈.... 让我来教你吧 ...

主要的思路是新建一个语法文件, 在文件中定义你要高亮的东东, 想高亮什么就高亮什么, 用vim就是这么自信. 所谓的语法文件就是vim用来高亮各种源文件的一个脚本, vim靠这个脚本的描述来使文件中的不同文本显示不同的颜色, 比如C语言的语法文件放在类似于这样的一个路径中:

```
/usr/share/vim/vim64/syntax/c.vim
```

其他语言的语法文件也可以在这个路径中找到, 你的也许不在这个路径中, 不管它, 在你自己的HOME下新建一个语法文件, 新建一个空文件:

```
~/.vim/syntax/c.vim
```

在其中加入

```

=====
" Highlight All Function
=====
syn match cFunction "<[a-zA-Z_][a-zA-Z_0-9]*/>[^()]*" (me=e-2
syn match cFunction "<[a-zA-Z_][a-zA-Z_0-9]*/>/s*" (me=e-1
hi cFunction gui=NONE guifg=#B5A1FF

=====
" Highlight All Math Operator
=====
" C math operators
syn match cMathOperator display "[+/*%="
" C pointer operators
syn match cPointerOperator display "->/|/."
" C logical operators - boolean results
syn match cLogicalOperator display "[!<>]=/"
syn match cLogicalOperator display "=="
" C bit operators
syn match cBinaryOperator display "/(&/|/|/^\|<<|>>)/=/"
syn match cBinaryOperator display "/~"
syn match cBinaryOperatorError display "/~="
" More C logical operators - highlight in preference to binary
syn match cLogicalOperator display "&&/|||"
syn match cLogicalOperatorError display "/(&&/|||/)"

" Math Operator
hi cMathOperator guifg=#3EFFE2

```

```
hi cPointerOperator      guifg=#3EFFE2
hi cLogicalOperator     guifg=#3EFFE2
hi cBinaryOperator      guifg=#3EFFE2
hi cBinaryOperatorError guifg=#3EFFE2
hi cLogicalOperator     guifg=#3EFFE2
hi cLogicalOperatorError guifg=#3EFFE2
```

再打开你的C文件看看, 是不是又明亮了许多. 还有一个压箱底的要告诉你, 如果你自己增加了一个类型或者结构之类的, 怎么让它也象"int", "void"这样高亮起来呢? 再在上面的文件 ~/.vim/syntax/c.vim中添加下面的东东:

```
" =====
" My Own DataType
" =====
syn keyword cType      My_Type_1 My_Type_2 My_Type_3
```

这样你自己的类型My_Type_1, My_Type_2, My_Type_3就可以向"int"一样高亮起来了, 这样的缺点是每增加一个类型, 就要手动在这里添加一下, 如果有人知道更简单的方法请一定要告诉我, 用下面的地址:

Email	: lazy.fox.wu#gmail.com
Homepage	: http://blog.csdn.net/wooin

5 在程序中跳来跳去: Ctags 的使用

哇, 这下可厉害了, Tag文件(标签文件)可是程序员们的看家宝呀, 你可以不用它, 但你不能不知道它, 因为Linux内核源码都提供了"make tags"这个选项. 下面我们就来介绍Tag文件.

tags 文件是由ctags程序产生的一个索引文件, ctags程序其是叫"Exuberant Ctags", 是Unix上面ctags程序的替代品, 并且比它功能强大, 是大多数Linux发行版上默认的ctags程序. 那么tags文件是做什么用的呢? 如果你在读程序时看了一个函数调用, 或者一个变量, 或者一个宏等等, 你想知道它们的定义在哪儿, 怎么办呢? 用grep? 那会搜出很多不相干的地方. 现在流行用的是<C-], 谁用谁知道呀, 当光标在某个函数或变量上时, 按下"Ctrl+]", 光标会自动跳转到其定义处, 够厉害吧, 你不用再羡慕Visual Studio的程序员了, 开始羡慕我吧~_~.

你现在先别急着去按<C-], 你按没用的, 要不要我干什么呀, 你现在要做的是查查你电脑里有没有ctags这个程序, 如果有, 是什么版本的, 如果是Ctags 5.5.4, 就象我一样, 你最好去装一个Ctags 5.6, 这个在后面的自动补全章节中会用到. 在这个网站: <http://ctags.sourceforge.net>, 下载一个类似 ctags-5.6.tar.gz 的文件下来(现在好像5.7版的也出来了, 不过我还没用过):

用下面的命令解压安装:

```
$ tar -xzf ctags-5.6.tar.gz
$ cd ctags-5.6
$ make
# make install // 需要root权限
```

然后去你的源码目录, 如果你的源码是多层的目录, 就去最上层的目录, 在该目录下运行命令:

```
ctags -R
```

我现在以 vim71 的源码目录做演示

```
$ cd /home/wooin/vim71
$ ctags -R
```

此时在/home/wooin/vim71目录下会生成一个 tags 文件, 现在用vim打开 /home/wooin/vim71/src/main.c

```
$ vim /home/wooin/vim71/src/main.c
```

再在vim中运行命令:

```
:set tags=/home/wooin/vim71/tags
```

该命令将tags文件加入到vim中来, 你也可以将这句话放到~/.vimrc中去, 如果你经常在这个工程编程的话.

下面要开始真刀实枪的开干了, 如下图, 将光标放在setmouse()函数上



```

1:main.c] [7:term.c]
MiniBuf Explorer [-]
4 "= /home/wooin/vim71/src/ 969 # endif
5 ../ 970 . got_int = TRUE;
6 auto/ 971 . need_wait_return = FALSE;
7 objects/ 972 . global_busy = FALSE;
8 proto/ 973 . exmode_active = 0;
9 testdir/ 974 . skip_redraw = FALSE;
10 xxd/ 975 . RedrawingDisabled = 0;
11 INSTALL 976 . no_wait_return = 0;
12 Makefile 977 # ifdef FEAT_EVAL
13 README.txt 978 . msg_skip = 0;
14 arabic.c 979 # endif
980 . emsg_off = 0;
981 # ifdef FEAT_MOUSE
982 . setmouse();
983 # endif
984 . settmode(TMODE_RAW);
985 . starttermcap();
986 . scroll_start();
987 . redraw_later_clear();
988 }
989 #endif
990
991 clear_oparg(&oa);
992 while (!cmdwin
993 #ifdef FEAT_CMDWIN
994 . || cmdwin_result == 0
995 #endif

```

(--- 图3 ---)

此时按下<C-J>, 光标会自动跳到setmouse()函数的定义处, 见下图:

```

1:main.c] [7:term.c]*
MiniBuf Explorer [-]
4 "= /home/wooin/vim71/src/ 3311 /*
5 ../ 3312 int
6 auto/ 3313 swapping_screen()
7 objects/ 3314 {
8 proto/ 3315 . return (full_screen && *T_TI != NUL);
9 testdir/ 3316 }
10 xxd/ 3317
11 INSTALL 3318 #ifdef FEAT_MOUSE
12 Makefile 3319 /*
13 README.txt 3320 * setmouse() - switch mouse on/off depending on current
14 arabic.c 3321 * mode and 'mouse'
3322 */
3323 void
3324 setmouse()
3325 {
3326 #ifdef FEAT_MOUSE_TTY
3327 . int checkfor;
3328 #endif
3329 #ifdef FEAT_MOUSE_SHAPE
3330 . update_mousetheshape(-1);
3331 #endif
3332
3333 #ifdef FEAT_MOUSE_TTY /* Should be outside proc, but may
3334 . break MOUSESHAPE */
3335 #ifdef FEAT_GUI
3336 /* In the GUI the mouse is always enabled. */

```

(--- 图4 ---)

如果此时你还想再跳回刚才的位置, 你还可以按<C-T>, 这样又跳回到setmouse()函数被调用的地方了, 变量, 结构, 宏, 等等, 都可以的, 赶快试试吧.....

此时在回头学习一下第3节中说的vim手册吧

```
:help usr_29
```

不过还有一个小瑕疵, 你修改程序后, 比如增加了函数定义, 删除了变量定义, tags文件不能自动rebuild, 你必须手动再运行一下命令:

```
$ ctags -R
```

使tags文件更新一下, 不过让人感到欣慰的是vim不用重新启动, 正在编写的程序也不用退出, 马上就可以又正确使用<C-J>和<C-T>了. 如果有人知道更简单的方法请一定一定要告诉我, 用下面的地址:

Email	: lazy.fox.wu#gmail.com
Homepage	: http://blog.csdn.net/wooin

6. 教你高效地浏览源码 -- 插件: TagList

下载地址	http://www.vim.org/scripts/script.php?script_id=273
版本	4.4
安装	在 ~/.vim 目录下解压taglist_xx.zip



```
手册 :help taglist.txt
```

在Windows平台上用过 Source Insight看程序的人肯定很熟悉代码窗口左边那个Symbol窗口, 那里面列出了当前文件中的所有宏, 全局变量, 函数名等, 在查看代码时用这个窗口总揽全局, 切换位置相当方便, 今天告诉你一个vim的插件: Taglist, 可以同样实现这个功能.

上一节已经告诉你ctags的用法了, ctags的基本原理是将程序程序中的一些关键字(比如:函数名, 变量名等)的名字, 位置等信息通过一个窗口告诉你, 如果你已经安装好taglist, 则可以用下面的命令看看taglist自带的帮助文件:

```
:help taglist.txt
```

下面是我翻译的其中的第一段"Overview", 供大家现了解一下taglist, 翻译的不好, 请指教: "Tab List"是一个用来浏览源代码的Vim插件, 这个插件可以让你高效地浏览各种不同语言编写的源代码, "Tag List"有以下一些特点:

- * 在Vim的一个垂直或水平的分割窗口中显示一个文件中定义的tags(函数, 类, 结构, 变量, 等)
- * 在GUI Vim中, 可以选择把tags显示在下拉菜单和弹出菜单中
- * 当你在多个源文件/缓冲区间切换时, taglist窗口会自动进行相应地更新.
当你打开新文件时, 新文件中定义的tags会被添加到已经存在的文件列表中, 并且所有文件中定义的tags会以文件名来分组显示
- * 当你在taglist窗口中选中一个tag名时, 源文件中的光标会自动跳转到该tag的定义处
- * 自动高亮当前的tag名
- * 按类型分组各tag, 并且将各组显示在一个可折叠的树形结构中
- * 可以显示tag的原始类型和作用域
- * 在taglist窗口可选择显示tag的原始类型替代tag名
- * tag列表可以按照tag名, 或者时间进行排序
- * 支持以下语言的源文件: Assembly, ASP, Awk, Beta, C, C++, C#, Cobol, Eiffel, Erlang, Fortran, HTML, Java, Javascript, Lisp, Lua, Make, Pascal, Perl, PHP, Python, Rexx, Ruby, Scheme, Shell, Slang, SML, Sql, TCL, Verilog, Vim and Yacc.
- * 可以很简单的扩展支持新的语言. 对新语言支持的修改也很简单.
- * 提供了一些函数, 可以用来在Vim的状态栏或者在窗口的标题栏显示当前的tag名
- * taglist中的文件和tags的列表可以在被保存和在vim会话间加载
- * 提供了一些用来取得tag名和原始类型的命令
- * 在控制台vim和GUI vim中都可以使用
- * 可以和winmanager插件一起使用. winmanager插件可以让你同时使用文件浏览器, 缓冲区浏览器和taglist插件, 就像一个IDE一样.
- * 可以在Unix和MS-Windows系统中使用

首先请先在你的~/.vimrc文件中添加下面两句:

```
let Tlist_Show_One_File=1  
let Tlist_Exit_OnlyWindow=1
```

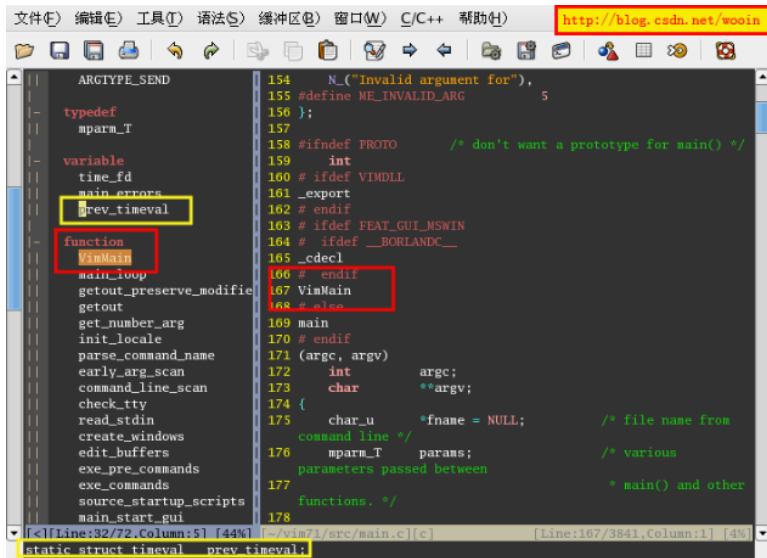
此时用vim打开一个c源文件试试:

```
$ vim ~/.vim/src/main.c
```

进入vim后用下面的命令打开taglist窗口, 如图5:

```
:Tlist
```





(--- 图5 ---)

左边的窗口就是前面介绍的TagList窗口，其中列出了main.c文件中的tag，并且按照“typedef”，“variable”，“function”等进行了分类。将光标移到VimMain上，如图中左边红色的方框，按下回车后，源程序会自动跳转到VimMain的定义处，如图中右边的红色方框。这就是TagList最基本也是最常用的操作。再教你一个常用的操作，你在浏览TagList窗口时，如果还不想让源码跳转，但是想看看tag在源码中完整的表达，可以将光标移到你想要看的tag上，如图中上边黄色的方框，然后按下空格键，在下面的命令栏中，如图下边黄色的方框，会显示该tag在源码中完整的写法，而不会跳转到源码处。

TagList插件我就介绍到这里，其实它还有很多用法和设置，我没法一一地教你了，好在TagList有一份详细的帮助手册，用下面的命令打开手册，好好学习一下吧：

```
:help taglist.txt
```

7. 文件浏览器和窗口管理器 -- 插件: WinManager

下载地址	http://www.vim.org/scripts/script.php?script_id=95
版本	2.x
安装	在 ~/.vim 目录下解压winmanager.zip
手册	:help winmanager

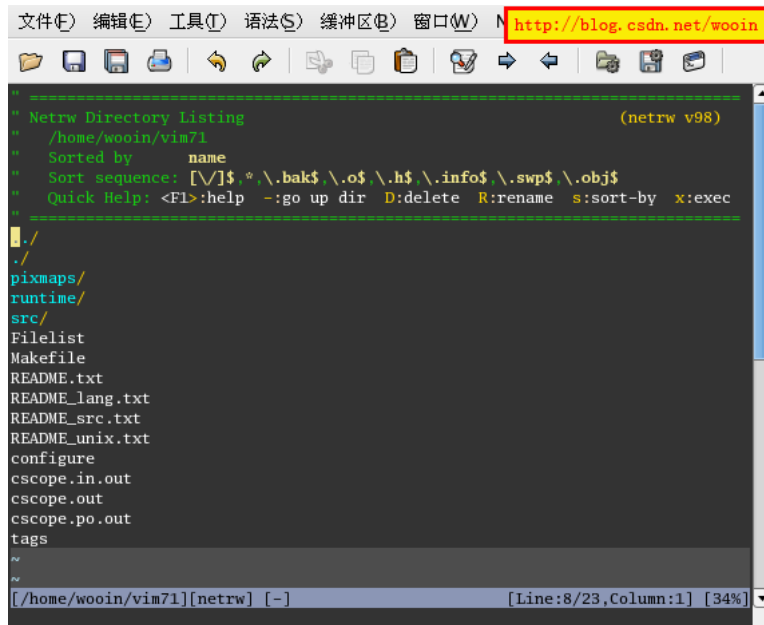
在图1中大家可以看到在图标2标识的地方是一个文件浏览器，里面列出了当前目录中的文件，可以通过这个浏览器来浏览工程中的源文件，是不是越来越像常见的IDE了，当光标停在某个文件或文件夹的时候，按下回车，可以打开该文件或文件夹。

这个东东是怎么调出来的呢？其实这个也是由插件实现的，这个插件是netrw.vim，只不过你不用下载和安装，这个插件已经是标准的vim插件，已经随vim一起安装进你的系统里了，现在先简单演示一下，进入“~/vim71”文件夹后运行vim，然后在vim中运行命令：

```
:e ~/vim71
```



你将在vim看到如下图所示的界面:



(--- 图6 ---)

在该界面上你可以用下面的一些命令来进行常用的目录和文件操作:

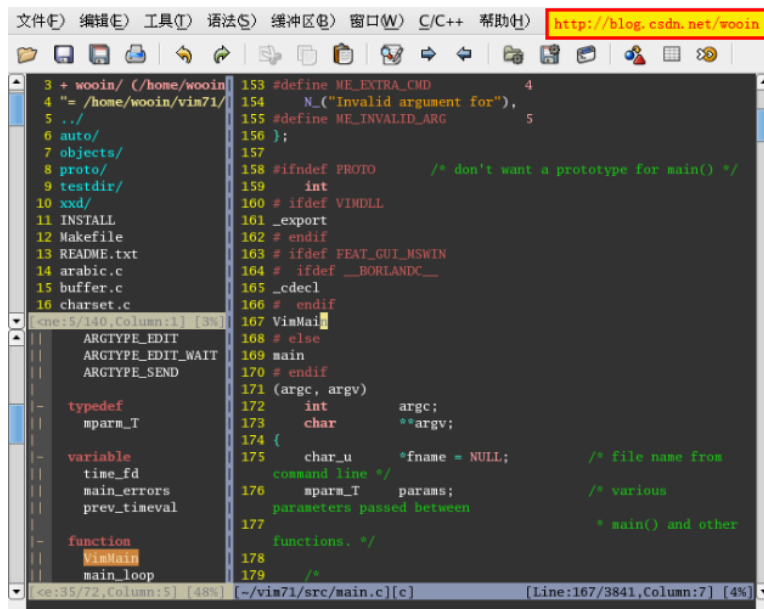
<F1>	显示帮助
<cr>	如果光标下是目录,则进入该目录;如果光标下文件,则打开该文件
-	返回上级目录
c	切换vim 当前工作目录正在浏览的目录
d	创建目录
D	删除目录或文件
i	切换显示方式
R	文件或目录重命名
s	选择排序方式
x	定制浏览方式,使用你指定的程序打开该文件

我这里不是教你怎么用netrw.vim插件,而是要教你通过WinManager插件来将TagList窗口和netrw窗口整合起来,就像图1中的图标2和3组成的那个效果

现在在你的~/.vimrc中增加下面两句

```
let g:winManagerWindowLayout='FileExplorer|TagList'
nmap wm :WMToggle<cr>
```

然后重启vim,打开~/vim71/src/main.c,在normal状态下输入"wm",你将看到图7的样子:



(--- 图7 ---)

其中左上边是netrw窗口, 左下边是TagList窗口, 当再次输入"wm"命令时这两个窗口又关闭了.

WinManager的功能主要就是介绍的这些, 但是它还有其他一些高级功能, 还可以支持其他几个插件, 如果你觉得介绍的还不够你用, 建议你把它帮助手册好好研究一下, 用下面的命令可以调出帮助手册:

```
:help winmanager
```

8. Cscope 的使用

这下更厉害了, 用Cscope自己的话说 - "你可以把它当做是超过频的ctags", 其功能和强大程度可见一斑吧, 关于它的介绍我就不详细说了, 如果你安装好了前文介绍的中文帮助手册, 用下面的命令看看介绍吧:

```
:help if_cscop.txt
```

我在这里简单摘抄一点, 供还在犹豫的朋友看看:

Cscope 是一个交互式的屏幕下使用的工具, 用来帮助你:

- * 无须在厚厚的程序清单中翻来翻去就可以认识一个 C 程序的工作原理。
- * 无须熟悉整个程序就可以知道清楚程序 bug 所要修改的代码位置。
- * 检查提议的改动 (如添加一个枚举值) 可能会产生的效果。
- * 验证所有的源文件都已经作了需要的修改; 例如给某一个现存的函数添加一个参数。
- * 在所有相关的源文件中对一个全局变量改名。
- * 在所有相关的位置将一个常数改为一个预处理符号。

它被设计用来回答以下的问题:

- * 什么地方用到了这个符号?
- * 这是在什么地方定义的?
- * 这个变量在哪里被赋值?
- * 这个全局符号的定义在哪里?
- * 这个函数在源文件中的哪个地方?
- * 哪些函数调用了这个函数?
- * 这个函数调用了哪些函数?
- * 信息 "out of space" 从哪来?
- * 这个源文件在整个目录结构中处于什么位置?
- * 哪些文件包含这个头文件?

安装Cscope:

如果你的系统中有cscope命令, 则可以跳过这一小段, 如果没有, 就先跟着我一起安装一个吧.

在Cscope的主页: <http://cscope.sourceforge.net> 下载一个源码包, 解压后编译安装:

```
#!/configure
# make
# make install // 需要root权限
```

先在~/.vimrc中增加一句:

```
:set cscopequickfix=s-,c-,d-,i-,t-,e-
```

这个是设定是否使用 quickfix 窗口来显示 cscope 结果, 用法在后面会说到.

跟Ctags一样, 要使用其功能必须先为你的代码生成一个cscope的数据库, 在项目的根目录运行下面的命令:

```
$ cd /home/wooin/vim71/
$ cscope -Rbq
# 此后会生成三个文件
$ ll cscope.*
-rw-rw-r-- 1 wooin wooin 1.1M 2007-09-30 10:56 cscope.in.out
-rw-rw-r-- 1 wooin wooin 6.7M 2007-09-30 10:56 cscope.out
-rw-rw-r-- 1 wooin wooin 5.1M 2007-09-30 10:56 cscope.po.out
# 打开文件, 开始Coding
$ cd src
$ vi main.c
```

进入vim后第一件事是要把刚才生成的cscope文件导入到vim中来, 用下面的命令:

```
:cs add /home/wooin/vim71/cscope.out /home/wooin/vim71
```



上面这条命令很重要, 必须写全, 不能只写前半句:

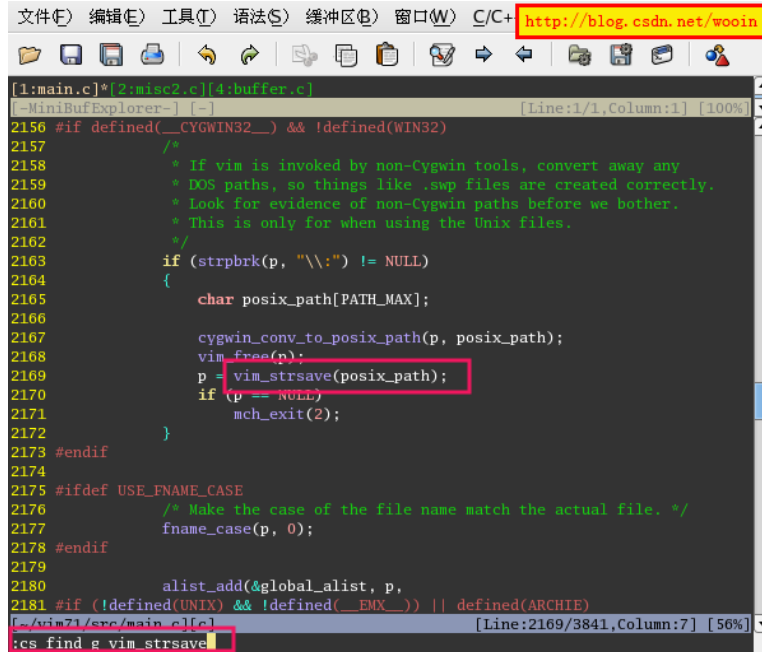
```
:cs add /home/wooin/vim71/cscope.out
```

因为源码是多级目录的, 如果这样写, cscope是无法在子目录中的源码中工作的, 当然, 如果你的源码都在同一级目录中就无所谓了. 如果你要经常用cscope的话, 可以把上面那句加到~/.vimrc中去.

下面我们来操练一下, 查找函数vim_strsave()的定义, 用命令:

```
:cs find g vim_strsave
```

如下图:

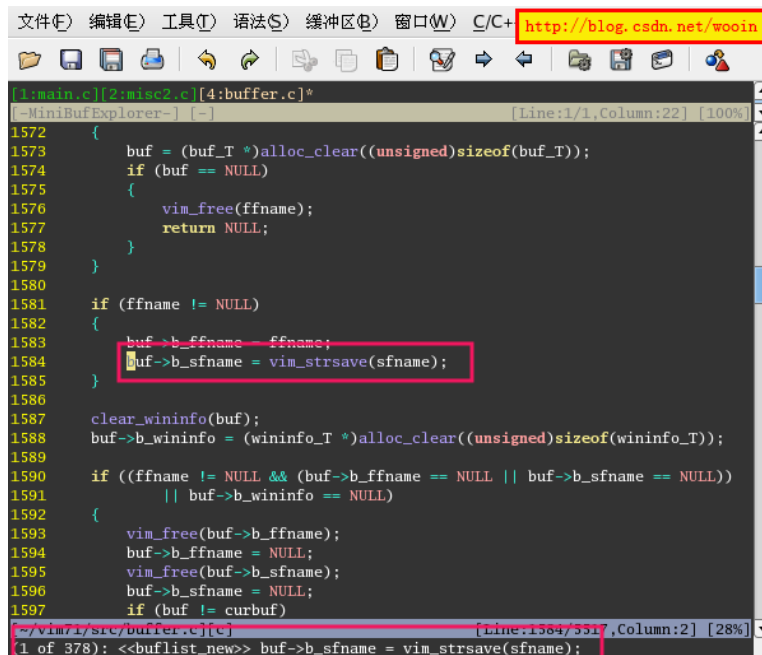


(--- 图8 ---)

按下回车后会自动跳转到vim_strsave()的定义处. 此时你肯定会说Ctags也可以做到这个呀, 那么下面说的这个Ctags就无法做到了, 我想查找vim_strsave()到底在那些地方被调用过了, 用命令:

```
:cs find c vim_strsave
```

按下回车后vim会自动跳转到第一个符合要求的地方, 并且在命令栏显示有多少符合要求的结果, 如图:

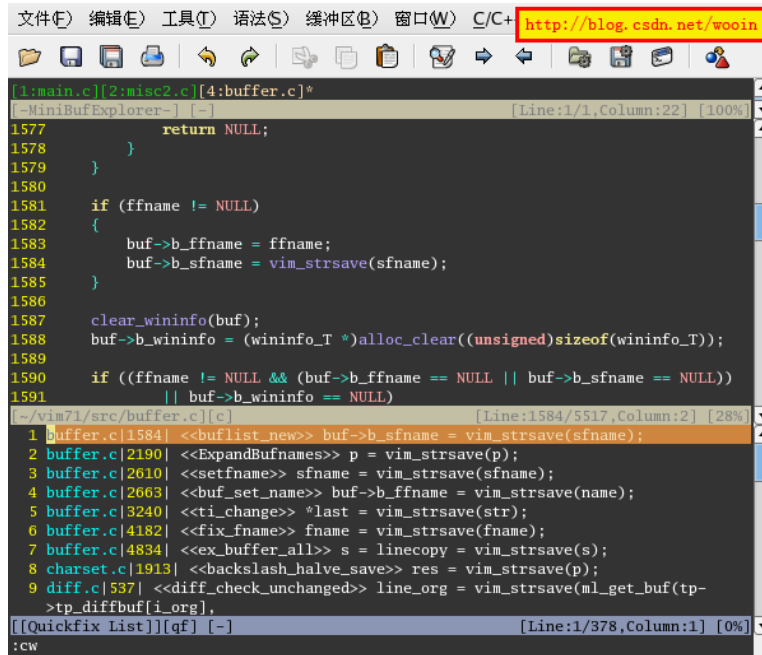


(--- 图9 ---)

如果自动跳转的位置你不满意, 想看其他的结果, 可以用下面的命令打开QuickFix窗口:

```
:cw
```

如图:



(--- 图10 ---)

这时你就可以慢慢挑选了^_^

cscope的主要功能是通过同的子命令“find”来实现的

“cscope find”的用法:

cs find c|d|e|f|g|i|s|t name

0 或 s	查找本 C 符号(可以跳过注释)
1 或 g	查找本定义
2 或 d	查找本函数调用的函数
3 或 c	查找调用本函数的函数
4 或 t	查找本字符串
6 或 e	查找本 egrep 模式
7 或 f	查找本文件
8 或 i	查找包含本文件的文件

如果每次查找都要输入一长串命令的话还真是件讨厌的事情, Cscope的帮助手册中推荐了一些快捷键的用法, 下面是其中一组, 也是我用的, 将下面的内容添加到 ~/.vimrc 中, 并重启 vim:

```
nmap <C->s :cs find s <C-R>=expand("<word>")<CR><CR>
nmap <C->g :cs find g <C-R>=expand("<word>")<CR><CR>
nmap <C->c :cs find c <C-R>=expand("<word>")<CR><CR>
nmap <C->t :cs find t <C-R>=expand("<word>")<CR><CR>
nmap <C->e :cs find e <C-R>=expand("<word>")<CR><CR>
nmap <C->f :cs find f <C-R>=expand("<file>")<CR><CR>
nmap <C->i :cs find i ^<C-R>=expand("<file>")<CR>${<CR>
nmap <C->d :cs find d <C-R>=expand("<word>")<CR><CR>
```

当光标停在某个你要查找的词上时, 按下 <C->g, 就是查找该对象的定义, 其他的同理。

按这种组合键有一点技巧, 按了 <C->后要马上按下一个键, 否则屏幕一闪就回到 normal 状态了 <C->g 的按法是先按 "Ctrl+Shift+.", 然后很快再按 "g"

很奇怪, 其中的这句:

```
nmap <C->i :cs find i ^<C-R>=expand("<file>")<CR>${<CR>
```

在我的 vim 中无法工作, 但是我改成:

```
nmap <C->i :cs find i <C-R>=expand("<file>")<CR><CR>
```

就可以正常工作了, 不知道是什么原因? 有哪位朋友知道请告诉我。

cscope 的其他功能你可以通过帮助手册自己慢慢学习

reset : 重新初始化所有连接。

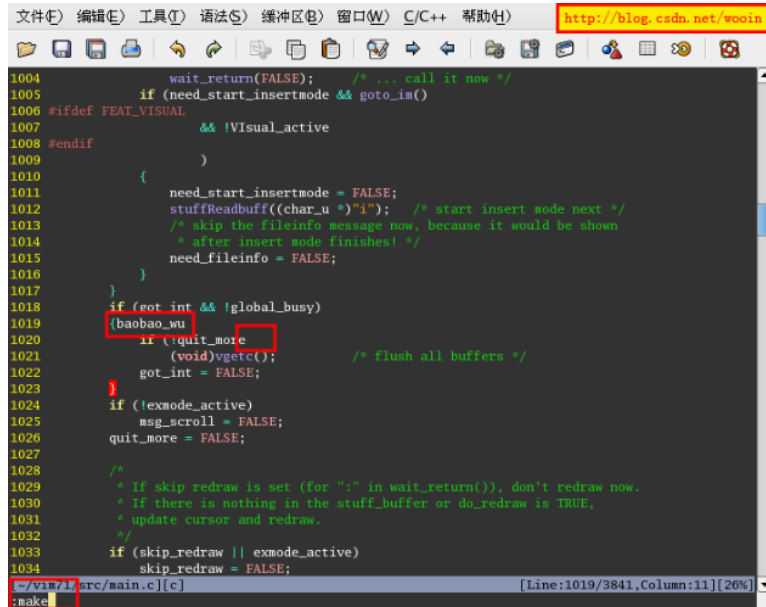
用法 : cs reset

9. QuickFix 窗口

在上一节的图10中大家可以看到在窗口下面有一个显示查询结果的窗口, 这个窗口中列出了查询命令的查询结果, 用户可以从这个窗口中选择每个结果进行查看, 这个窗口叫"QuickFix"窗口, 以前也是一个vim的插件来的, 只不过现在成了vim的标准插件, 不用你在去安装了, QuickFix窗口的主要作用就是上面看到的那个功能: 输出一些供选择的结果, 可以被很多命令调用, 更详细的介绍和使用方法请用下面的命令打开QuickFix的手册来学习吧:

```
:help quickfix
```

这里我一个常用的例子来再介绍一种QuickFix窗口的使用方法. 这个例子是要模仿平时我们编程时, 当编译出错时, QuickFix会把出错的信息列出来, 供我们一条条地查看和修改. 首先还是用vim打开~/vim71/src/main.c, 事先最好先编译过vim71, 否则一会儿编译的时候有点慢, 或者你也可以自己写一个小的有错误的程序来跟着我做下面的步骤, 见下图:



```
1004         wait_return(FALSE); /* ... call it now */
1005         if (need_start_insertmode && goto_in())
1006             #ifdef FEAT_VISUAL
1007                 && !Visual_active
1008             #endif
1009         )
1010         {
1011             need_start_insertmode = FALSE;
1012             stuffReadbuff((char_u *)"i"); /* start insert mode next */
1013             /* skip the fileinfo message now, because it would be shown
1014              * after insert mode finishes! */
1015             need_fileinfo = FALSE;
1016         }
1017     }
1018     if (got_int && !global_busy)
1019         baobao_wu
1020         if (!quit_more
1021             (void)vgetc(); /* flush all buffers */
1022             got_int = FALSE;
1023         }
1024     if (!exmode_active)
1025         msg_scroll = FALSE;
1026     quit_more = FALSE;
1027
1028     /*
1029     * If skip redraw is set (for ":" in wait_return()), don't redraw now.
1030     * If there is nothing in the stuff_buffer or do_redraw is TRUE,
1031     * update cursor and redraw.
1032     */
1033     if (skip_redraw || exmode_active)
1034         skip_redraw = FALSE;
1035 }
1036
1037 #endif
1038
1039 #endif
1040
1041 #endif
1042
1043 #endif
1044
1045 #endif
1046
1047 #endif
1048
1049 #endif
1050
1051 #endif
1052
1053 #endif
1054
1055 #endif
1056
1057 #endif
1058
1059 #endif
1060
1061 #endif
1062
1063 #endif
1064
1065 #endif
1066
1067 #endif
1068
1069 #endif
1070
1071 #endif
1072
1073 #endif
1074
1075 #endif
1076
1077 #endif
1078
1079 #endif
1080
1081 #endif
1082
1083 #endif
1084
1085 #endif
1086
1087 #endif
1088
1089 #endif
1090
1091 #endif
1092
1093 #endif
1094
1095 #endif
1096
1097 #endif
1098
1099 #endif
1100
1101 #endif
1102
1103 #endif
1104
1105 #endif
1106
1107 #endif
1108
1109 #endif
1110
1111 #endif
1112
1113 #endif
1114
1115 #endif
1116
1117 #endif
1118
1119 #endif
1120
1121 #endif
1122
1123 #endif
1124
1125 #endif
1126
1127 #endif
1128
1129 #endif
1130
1131 #endif
1132
1133 #endif
1134
1135 #endif
1136
1137 #endif
1138
1139 #endif
1140
1141 #endif
1142
1143 #endif
1144
1145 #endif
1146
1147 #endif
1148
1149 #endif
1150
1151 #endif
1152
1153 #endif
1154
1155 #endif
1156
1157 #endif
1158
1159 #endif
1160
1161 #endif
1162
1163 #endif
1164
1165 #endif
1166
1167 #endif
1168
1169 #endif
1170
1171 #endif
1172
1173 #endif
1174
1175 #endif
1176
1177 #endif
1178
1179 #endif
1180
1181 #endif
1182
1183 #endif
1184
1185 #endif
1186
1187 #endif
1188
1189 #endif
1190
1191 #endif
1192
1193 #endif
1194
1195 #endif
1196
1197 #endif
1198
1199 #endif
1200
1201 #endif
1202
1203 #endif
1204
1205 #endif
1206
1207 #endif
1208
1209 #endif
1210
1211 #endif
1212
1213 #endif
1214
1215 #endif
1216
1217 #endif
1218
1219 #endif
1220
1221 #endif
1222
1223 #endif
1224
1225 #endif
1226
1227 #endif
1228
1229 #endif
1230
1231 #endif
1232
1233 #endif
1234
1235 #endif
1236
1237 #endif
1238
1239 #endif
1240
1241 #endif
1242
1243 #endif
1244
1245 #endif
1246
1247 #endif
1248
1249 #endif
1250
1251 #endif
1252
1253 #endif
1254
1255 #endif
1256
1257 #endif
1258
1259 #endif
1260
1261 #endif
1262
1263 #endif
1264
1265 #endif
1266
1267 #endif
1268
1269 #endif
1270
1271 #endif
1272
1273 #endif
1274
1275 #endif
1276
1277 #endif
1278
1279 #endif
1280
1281 #endif
1282
1283 #endif
1284
1285 #endif
1286
1287 #endif
1288
1289 #endif
1290
1291 #endif
1292
1293 #endif
1294
1295 #endif
1296
1297 #endif
1298
1299 #endif
1300
1301 #endif
1302
1303 #endif
1304
1305 #endif
1306
1307 #endif
1308
1309 #endif
1310
1311 #endif
1312
1313 #endif
1314
1315 #endif
1316
1317 #endif
1318
1319 #endif
1320
1321 #endif
1322
1323 #endif
1324
1325 #endif
1326
1327 #endif
1328
1329 #endif
1330
1331 #endif
1332
1333 #endif
1334
1335 #endif
1336
1337 #endif
1338
1339 #endif
1340
1341 #endif
1342
1343 #endif
1344
1345 #endif
1346
1347 #endif
1348
1349 #endif
1350
1351 #endif
1352
1353 #endif
1354
1355 #endif
1356
1357 #endif
1358
1359 #endif
1360
1361 #endif
1362
1363 #endif
1364
1365 #endif
1366
1367 #endif
1368
1369 #endif
1370
1371 #endif
1372
1373 #endif
1374
1375 #endif
1376
1377 #endif
1378
1379 #endif
1380
1381 #endif
1382
1383 #endif
1384
1385 #endif
1386
1387 #endif
1388
1389 #endif
1390
1391 #endif
1392
1393 #endif
1394
1395 #endif
1396
1397 #endif
1398
1399 #endif
1400
1401 #endif
1402
1403 #endif
1404
1405 #endif
1406
1407 #endif
1408
1409 #endif
1410
1411 #endif
1412
1413 #endif
1414
1415 #endif
1416
1417 #endif
1418
1419 #endif
1420
1421 #endif
1422
1423 #endif
1424
1425 #endif
1426
1427 #endif
1428
1429 #endif
1430
1431 #endif
1432
1433 #endif
1434
1435 #endif
1436
1437 #endif
1438
1439 #endif
1440
1441 #endif
1442
1443 #endif
1444
1445 #endif
1446
1447 #endif
1448
1449 #endif
1450
1451 #endif
1452
1453 #endif
1454
1455 #endif
1456
1457 #endif
1458
1459 #endif
1460
1461 #endif
1462
1463 #endif
1464
1465 #endif
1466
1467 #endif
1468
1469 #endif
1470
1471 #endif
1472
1473 #endif
1474
1475 #endif
1476
1477 #endif
1478
1479 #endif
1480
1481 #endif
1482
1483 #endif
1484
1485 #endif
1486
1487 #endif
1488
1489 #endif
1490
1491 #endif
1492
1493 #endif
1494
1495 #endif
1496
1497 #endif
1498
1499 #endif
1500
1501 #endif
1502
1503 #endif
1504
1505 #endif
1506
1507 #endif
1508
1509 #endif
1510
1511 #endif
1512
1513 #endif
1514
1515 #endif
1516
1517 #endif
1518
1519 #endif
1520
1521 #endif
1522
1523 #endif
1524
1525 #endif
1526
1527 #endif
1528
1529 #endif
1530
1531 #endif
1532
1533 #endif
1534
1535 #endif
1536
1537 #endif
1538
1539 #endif
1540
1541 #endif
1542
1543 #endif
1544
1545 #endif
1546
1547 #endif
1548
1549 #endif
1550
1551 #endif
1552
1553 #endif
1554
1555 #endif
1556
1557 #endif
1558
1559 #endif
1560
1561 #endif
1562
1563 #endif
1564
1565 #endif
1566
1567 #endif
1568
1569 #endif
1570
1571 #endif
1572
1573 #endif
1574
1575 #endif
1576
1577 #endif
1578
1579 #endif
1580
1581 #endif
1582
1583 #endif
1584
1585 #endif
1586
1587 #endif
1588
1589 #endif
1590
1591 #endif
1592
1593 #endif
1594
1595 #endif
1596
1597 #endif
1598
1599 #endif
1600
1601 #endif
1602
1603 #endif
1604
1605 #endif
1606
1607 #endif
1608
1609 #endif
1610
1611 #endif
1612
1613 #endif
1614
1615 #endif
1616
1617 #endif
1618
1619 #endif
1620
1621 #endif
1622
1623 #endif
1624
1625 #endif
1626
1627 #endif
1628
1629 #endif
1630
1631 #endif
1632
1633 #endif
1634
1635 #endif
1636
1637 #endif
1638
1639 #endif
1640
1641 #endif
1642
1643 #endif
1644
1645 #endif
1646
1647 #endif
1648
1649 #endif
1650
1651 #endif
1652
1653 #endif
1654
1655 #endif
1656
1657 #endif
1658
1659 #endif
1660
1661 #endif
1662
1663 #endif
1664
1665 #endif
1666
1667 #endif
1668
1669 #endif
1670
1671 #endif
1672
1673 #endif
1674
1675 #endif
1676
1677 #endif
1678
1679 #endif
1680
1681 #endif
1682
1683 #endif
1684
1685 #endif
1686
1687 #endif
1688
1689 #endif
1690
1691 #endif
1692
1693 #endif
1694
1695 #endif
1696
1697 #endif
1698
1699 #endif
1700
1701 #endif
1702
1703 #endif
1704
1705 #endif
1706
1707 #endif
1708
1709 #endif
1710
1711 #endif
1712
1713 #endif
1714
1715 #endif
1716
1717 #endif
1718
1719 #endif
1720
1721 #endif
1722
1723 #endif
1724
1725 #endif
1726
1727 #endif
1728
1729 #endif
1730
1731 #endif
1732
1733 #endif
1734
1735 #endif
1736
1737 #endif
1738
1739 #endif
1740
1741 #endif
1742
1743 #endif
1744
1745 #endif
1746
1747 #endif
1748
1749 #endif
1750
1751 #endif
1752
1753 #endif
1754
1755 #endif
1756
1757 #endif
1758
1759 #endif
1760
1761 #endif
1762
1763 #endif
1764
1765 #endif
1766
1767 #endif
1768
1769 #endif
1770
1771 #endif
1772
1773 #endif
1774
1775 #endif
1776
1777 #endif
1778
1779 #endif
1780
1781 #endif
1782
1783 #endif
1784
1785 #endif
1786
1787 #endif
1788
1789 #endif
1790
1791 #endif
1792
1793 #endif
1794
1795 #endif
1796
1797 #endif
1798
1799 #endif
1800
1801 #endif
1802
1803 #endif
1804
1805 #endif
1806
1807 #endif
1808
1809 #endif
1810
1811 #endif
1812
1813 #endif
1814
1815 #endif
1816
1817 #endif
1818
1819 #endif
1820
1821 #endif
1822
1823 #endif
1824
1825 #endif
1826
1827 #endif
1828
1829 #endif
1830
1831 #endif
1832
1833 #endif
1834
1835 #endif
1836
1837 #endif
1838
1839 #endif
1840
1841 #endif
1842
1843 #endif
1844
1845 #endif
1846
1847 #endif
1848
1849 #endif
1850
1851 #endif
1852
1853 #endif
1854
1855 #endif
1856
1857 #endif
1858
1859 #endif
1860
1861 #endif
1862
1863 #endif
1864
1865 #endif
1866
1867 #endif
1868
1869 #endif
1870
1871 #endif
1872
1873 #endif
1874
1875 #endif
1876
1877 #endif
1878
1879 #endif
1880
1881 #endif
1882
1883 #endif
1884
1885 #endif
1886
1887 #endif
1888
1889 #endif
1890
1891 #endif
1892
1893 #endif
1894
1895 #endif
1896
1897 #endif
1898
1899 #endif
1900
1901 #endif
1902
1903 #endif
1904
1905 #endif
1906
1907 #endif
1908
1909 #endif
1910
1911 #endif
1912
1913 #endif
1914
1915 #endif
1916
1917 #endif
1918
1919 #endif
1920
1921 #endif
1922
1923 #endif
1924
1925 #endif
1926
1927 #endif
1928
1929 #endif
1930
1931 #endif
1932
1933 #endif
1934
1935 #endif
1936
1937 #endif
1938
1939 #endif
1940
1941 #endif
1942
1943 #endif
1944
1945 #endif
1946
1947 #endif
1948
1949 #endif
1950
1951 #endif
1952
1953 #endif
1954
1955 #endif
1956
1957 #endif
1958
1959 #endif
1960
1961 #endif
1962
1963 #endif
1964
1965 #endif
1966
1967 #endif
1968
1969 #endif
1970
1971 #endif
1972
1973 #endif
1974
1975 #endif
1976
1977 #endif
1978
1979 #endif
1980
1981 #endif
1982
1983 #endif
1984
1985 #endif
1986
1987 #endif
1988
1989 #endif
1990
1991 #endif
1992
1993 #endif
1994
1995 #endif
1996
1997 #endif
1998
1999 #endif
2000
2001 #endif
2002
2003 #endif
2004
2005 #endif
2006
2007 #endif
2008
2009 #endif
2010
2011 #endif
2012
2013 #endif
2014
2015 #endif
2016
2017 #endif
2018
2019 #endif
2020
2021 #endif
2022
2023 #endif
2024
2025 #endif
2026
2027 #endif
2028
2029 #endif
2030
2031 #endif
2032
2033 #endif
2034
2035 #endif
2036
2037 #endif
2038
2039 #endif
2040
2041 #endif
2042
2043 #endif
2044
2045 #endif
2046
2047 #endif
2048
2049 #endif
2050
2051 #endif
2052
2053 #endif
2054
2055 #endif
2056
2057 #endif
2058
2059 #endif
2060
2061 #endif
2062
2063 #endif
2064
2065 #endif
2066
2067 #endif
2068
2069 #endif
2070
2071 #endif
2072
2073 #endif
2074
2075 #endif
2076
2077 #endif
2078
2079 #endif
2080
2081 #endif
2082
2083 #endif
2084
2085 #endif
2086
2087 #endif
2088
2089 #endif
2090
2091 #endif
2092
2093 #endif
2094
2095 #endif
2096
2097 #endif
2098
2099 #endif
2100
2101 #endif
2102
2103 #endif
2104
2105 #endif
2106
2107 #endif
2108
2109 #endif
2110
2111 #endif
2112
2113 #endif
2114
2115 #endif
2116
2117 #endif
2118
2119 #endif
2120
2121 #endif
2122
2123 #endif
2124
2125 #endif
2126
2127 #endif
2128
2129 #endif
2130
2131 #endif
2132
2133 #endif
2134
2135 #endif
2136
2137 #endif
2138
2139 #endif
2140
2141 #endif
2142
2143 #endif
2144
2145 #endif
2146
2147 #endif
2148
2149 #endif
2150
2151 #endif
2152
2153 #endif
2154
2155 #endif
2156
2157 #endif
2158
2159 #endif
2160
2161 #endif
2162
2163 #endif
2164
2165 #endif
2166
2167 #endif
2168
2169 #endif
2170
2171 #endif
2172
2173 #endif
2174
2175 #endif
2176
2177 #endif
2178
2179 #endif
2180
2181 #endif
2182
2183 #endif
2184
2185 #endif
2186
2187 #endif
2188
2189 #endif
2190
2191 #endif
2192
2193 #endif
2194
2195 #endif
2196
2197 #endif
2198
2199 #endif
2200
2201 #endif
2202
2203 #endif
2204
2205 #endif
2206
2207 #endif
2208
2209 #endif
2210
2211 #endif
2212
2213 #endif
2214
2215 #endif
2216
2217 #endif
2218
2219 #endif
2220
2221 #endif
2222
2223 #endif
2224
2225 #endif
2226
2227 #endif
2228
2229 #endif
2230
2231 #endif
2232
2233 #endif
2234
2235 #endif
2236
2237 #endif
2238
2239 #endif
2240
2241 #endif
2242
2243 #endif
2244
2245 #endif
2246
2247 #endif
2248
2249 #endif
2250
2251 #endif
2252
2253 #endif
2254
2255 #endif
2256
2257 #endif
2258
2259 #endif
2260
2261 #endif
2262
2263 #endif
2264
2265 #endif
2266
2267 #endif
2268
2269 #endif
2270
2271 #endif
2272
2273 #endif
2274
2275 #endif
2276
2277 #endif
2278
2279 #endif
2280
2281 #endif
2282
2283 #endif
2284
2285 #endif
2286
2287 #endif
2288
2289 #endif
2290
2291 #endif
2292
2293 #endif
2294
2295 #endif
2296
2297 #endif
2298
2299 #endif
2300
2301 #endif
2302
2303 #endif
2304
2305 #endif
2306
2307 #endif
2308
2309 #endif
2310
2311 #endif
2312
2313 #endif
2314
2315 #endif
2316
2317 #endif
2318
2319 #endif
2320
2321 #endif
2322
2323 #endif
2324
2325 #endif
2326
2327 #endif
2328
2329 #endif
2330
2331 #endif
2332
2333 #endif
2334
2335 #endif
2336
2337 #endif
2338
2339 #endif
2340
2341 #endif
2342
2343 #endif
2344
2345 #endif
2346
2347 #endif
2348
2349 #endif
2350
2351 #endif
2352
2353 #endif
2354
2355 #endif
2356
2357 #endif
2358
2359 #endif
2360
2361 #endif
2362
2363 #endif
2364
2365 #endif
2366
2367 #endif
2368
2369 #endif
2370
2371 #endif
2372
2373 #endif
2374
2375 #endif
2376
2377 #endif
2378
2379 #endif
2380
2381 #endif
2382
2383 #endif
2384
2385 #endif
2386
2387 #endif
2388
2389 #endif
2390
2391 #endif
2392
2393 #endif
2394
2395 #endif
2396
2397 #endif
2398
2399 #endif
2400
2401 #endif
2402
2403 #endif
2404
2405 #endif
2406
2407 #endif
2408
2409 #endif
2410
2411 #endif
2412
2413 #endif
2414
2415 #endif
2416
2417 #endif
2418
2419 #endif
2420
2421 #endif
2422
2423 #endif
2424
2425 #endif
2426
2427 #endif
2428
2429 #endif
2430
2431 #endif
2432
2433 #endif
2434
2435 #endif
2436
2437 #endif
2438
2439 #endif
2440
2441 #endif
2442
2443 #endif
2444
2445 #endif
2446
2447 #endif
2448
2449 #endif
2450
2451 #endif
2452
2453 #endif
2454
2455 #endif
2456
2457 #endif
2458
2459 #endif
2460
2461 #endif
2462
2463 #endif
2464
2465 #endif
2466
2467 #endif
2468
2469 #endif
2470
2471 #endif
2472
2473 #endif
2474
2475 #endif
2476
2477 #endif
2478
2479 #endif
2480
2481 #endif
2482
2483 #endif
2484
2485 #endif
2486
2487 #endif
2488
2489 #endif
2490
2491 #endif
2492
2493 #endif
2494
2495 #endif
2496
2497 #endif
2498
2499 #endif
2500
2501 #endif
2502
2503 #endif
2504
2505 #endif
2506
2507 #endif
2508
2509 #endif
2510
2511 #endif
2512
2513 #endif
2514
2515 #endif
2516
2517 #endif
2518
2519 #endif
2520
2521 #endif
2522
2523 #endif
2524
2525 #endif
2526
2527 #endif
2528
2529 #endif
2530
2531 #endif
2532
2533 #endif
2534
2535 #endif
2536
2537 #endif
2538
2539 #endif
2540
2541 #endif
2542
2543 #endif
2544
2545 #endif
2546
2547 #endif
2548
2549 #endif
2550
2551 #endif
2552
2553 #endif
2554
2555 #endif
2556
2557 #endif
2558
2559 #endif
2560
2561 #endif
2562
2563 #endif
2564
2565 #endif
2566
2567 #endif
2568
2569 #endif
2570
2571 #endif
2572
2573 #endif
2574
2575 #endif
2576
2577 #endif
2578
2579 #endif
2580
2581 #endif
2582
2583 #endif
2584
2585 #endif
2586
2587 #endif
2588
2589 #endif
2590
2591 #endif
2592
2593 #endif
2594
2595 #endif
2596
2597 #endif
2598
2599 #endif
2600
2601 #endif
2602
2603 #endif
2604
2605 #endif
2606
2607 #endif
2608
2609 #endif
2610
2611 #endif
2612
2613 #endif
2614
2615 #endif
2616
2617 #endif
2618
2619 #endif
2620
2621 #endif
2622
2623 #endif
2624
2625 #endif
2626
2627 #endif
2628
2629 #endif
2630
2631 #endif
2632
2633 #endif
2634
2635 #endif
2636
2637 #endif
2638
2639 #endif
2640
2641 #endif
2642
2643 #endif
2644
2645 #endif
2646
2647 #endif
2648
2649 #endif
2650
2651 #endif
2652
2653 #endif
2654
2655 #endif
2656
2657 #endif
2658
2659 #endif
2660
2661 #endif
2662
2663 #endif
2664
2665 #endif
2666
2667 #endif
2668
2669 #endif
2670
2671 #endif
2672
2673 #endif
2674
2675 #endif
2676
2677 #endif
2678
2679 #endif
2680
2681 #endif
2682
2683 #endif
2684
2685 #endif
2686
2687 #endif
2688
2689 #endif
2690
2691 #endif
2692
2693 #endif
2694
2695 #endif
2696
2697 #endif
2698
2699 #endif
2700
2701 #endif
2702
2703 #endif
2704
2705 #endif
2706
2707 #endif
2708
2709 #endif
2710
2711 #endif
2712
2713 #endif
2714
2715 #endif
2716
2717 #endif
2718
2719 #endif
2720
2721 #endif
2722
2723 #endif
2724
2725 #endif
2726
2727 #endif
2728
2729 #endif
2730
2731 #endif
2732
2733 #endif
2734
2735 #endif
2736
2737 #endif
2738
2739 #endif
2740
2741 #endif
2742
2743 #endif
2744
2745 #endif
2746
2747 #endif
2748
2749 #endif
2750
2751 #endif
2752
2753 #endif
2754
2755 #endif
2756
2757 #endif
2758
2759 #endif
2760
2761 #endif
2762
2763 #endif
2764
2765 #endif
2766
2767 #endif
2768
2769 #endif
2770
2771 #endif
2772
2773 #endif
2774
2775 #endif
2776
2777 #endif
2778
2779 #endif
2780
2781 #endif
2782
2783 #endif
2784
2785 #endif
2786
2787 #endif
2788
2789 #endif
2790
2791 #endif
2792
2793 #endif
2794
2795 #endif
2796
2797 #endif
2798
2799 #endif
2800
2801 #endif
2802
2803 #endif
2804
2805 #endif
2806
2807 #endif
2808
2809 #endif
2810
2811 #endif
2812
2813 #endif
2814
2815 #endif
2816
2817 #endif
2818
2819 #endif
2820
2821 #endif
2822
2823 #endif
2824
2825 #endif
2826
2827 #endif
2828
2829 #endif
2830
2831 #endif
2832
2833 #endif
2834
2835 #endif
2836
2837 #endif
2838
2839 #endif
2840
2841 #endif
2842
2843 #endif
2844
2845 #endif
2846
2847 #endif
2848
2849 #endif
2850
2851 #endif
2852
2853 #endif
2854
2855 #endif
2856
2857 #endif
2858
2859 #endif
2860
2861 #endif
2862
2863 #endif
2864
2865 #endif
2866
2867 #endif
2868
2869 #endif
2870
2871 #endif
2872
2873 #endif
2874
2875 #endif
2876
2877 #endif
2878
2879 #endif
2880
2881 #endif
2882
2883 #endif
2884
2885 #endif
2886
2887 #endif
2888
2889 #endif
2890
2891 #endif
2892
2893 #endif
2894
2895 #endif
2896
2897 #endif
2898
2899 #endif
2900
2901 #endif
2902
2903 #endif
2904
2905 #endif
2906
2907 #endif
2908
2909 #endif
2910
2911 #endif
2912
2913 #endif
2914
2915 #endif
2916
2917 #endif
2918
2919 #endif
2920
2921 #endif
2922
2923 #endif
2924
2925 #endif
2926
2927 #endif
2928
2929 #endif
2930
2931 #endif
2932
2933 #endif
2934
2935 #endif
2936
2937 #endif
2938
2939 #endif
```

在下面的QuickFix窗口中我们可以找到每一个编译错误, 同样你可以用鼠标点击每一条记录, 代码会马上自动跳转到错误处, 你还可以用下面的命令来跳转:

```
:cn // 切换到下一个结果
:cp // 切换到上一个结果
```

如果你经常使用这两个命令, 你还可以给他们设定快捷键, 比如在~/.vimrc中增加:

```
nmap <F6> :cn<cr>
nmap <F7> :cp<cr>
```

其还有其他的命令/插件也会用到QuickFix窗口, 但是用法基本上的都是类似的, 本文后面还会用到QuickFix窗口, 接着往下看吧。

10. 快速浏览和操作Buffer -- 插件: MiniBufExplorer

下载地址	http://www.vim.org/scripts/script.php?script_id=159
版本	6.3.2
安装	将下载的 minibufexpl.vim文件丢到 ~/.vim/plugin 文件夹中即可
手册	在minibufexpl.vim 文件的头部

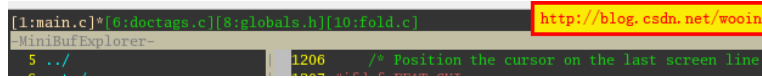
在编程的时候不可能永远只编辑一个文件, 你肯定会打开很多源文件进行编辑, 如果每个文件都打开一个vim进行编辑的话那操作起来将是多麻烦啊, 所以vim有buffer(缓冲区)的概念, 可以看vim的帮助:

```
:help buffer
```

vim自带的buffer管理工具只有:ls, :bnext, :bdelete 等的命令, 既不好用, 又不直观. 现在隆重向你推荐一款vim插件(plugin): MiniBufExplorer

使用方法:

重新启动vim, 当你只编辑一个buffer的时候MiniBufExplorer派不上用场, 当你打开第二个buffer的时候, MiniBufExplorer窗口就自动弹出来了, 见下图:



(--- 图13 ---)

上面那个狭长的窗口就是MiniBufExplorer窗口, 其中列出了当前所有已经打开的buffer, 当你把光标置于这个窗口时, 有下面几个快捷键可以用:

<Tab>	向前循环切换到每个buffer名上
<S-Tab>	向后循环切换到每个buffer名上
<Enter>	在打开光标所在的buffer
d	删除光标所在的buffer

以下的两个功能需要在~/.vimrc中增加:

```
let g:miniBufExplMapCTabSwitchBufs = 1
```

<C-Tab>	向前循环切换到每个buffer上,并在但前窗口打开
<C-S-Tab>	向后循环切换到每个buffer上,并在但前窗口打开

如果在~/.vimrc中设置了下面这句:

```
let g:miniBufExplMapWindowNavVim = 1
```

则可以用<C-h,j,k,l>切换到上下左右的窗口中去,就像:

C-w,h j k l 向"左,下,上,右"切换窗口.

在~/.vimrc中设置:

```
let g:miniBufExplMapWindowNavArrows = 1
```

是用<C-箭头键>切换到上下左右窗口中去

11. c/h文件间相互切换 -- 插件: A

下载地址	http://www.vim.org/scripts/script.php?script_id=31
版本	
安装	将a.vim 放到 ~/.vim/plugin 文件夹中



手册	无
----	---

下面介绍它的用法:

作为一个C程序员,日常Coding时在源文件与头文件间进行切换是再平常不过的事了,直接用vim打开其源/头文件其实也不是什么麻烦事,但是只用一个按键就切换过来了,这是多么贴心的功能啊...

安装好a.vim后有下面的几个命令可以用了:

:A	在新Buffer中切换到c/h文件
:AS	横向分割窗口并打开c/h文件
:AV	纵向分割窗口并打开c/h文件
:AT	新建一个标签页并打开c/h文件

其他还有一些命令,你可以在它的网页上看看,我都没用过,其实也都是大同小异,找到自己最顺手的就行了.

我在~/.vimrc中增加了一句:

```
nnoremap <silent> <F12> :A<CR>
```

意思是按F12时在一个新的buffer中打开c/h文件,这样在写程序的时候就可以不假思索地在c/h文件间进行切换,减少了按键的次数,思路也就更流畅了,阿弥陀佛....

12. 在工程中查找 -- 插件: Grep

下载地址	http://www.vim.org/scripts/script.php?script_id=311
版本	1.8
安装	把grep.vim 文件丢到 ~/.vim/plugin 文件夹就好了
手册	在grep.vim 文件头部

下面介绍它的用法:

vim 有自己的查找功能,但是跟shell中的grep比起来还是有些差距的,有时Coding正火急火燎的时候,真想按下F3,对光标所在的词来个全工程范围的grep,不用敲那些繁琐的命令,现在福音来了,跟我同样懒的人不在少数,在grep.vim脚本的前部可以找到一些说明文档:

:Grep	按照指定的规则在指定的文件中查找
:Rgrep	同上,但是是递归的grep
:GrepBuffer	在所有打开的缓冲区中查找
:Bgrep	同上
:GrepArgs	在vim的argument filenames (:args)中查找
:Fgrep	运行fgrep
:Rfgrep	运行递归的fgrep
:Egrep	运行egrep
:Regrep	运行递归的egrep
:Agrep	运行agrep
:Ragrep	运行递归的agrep

上面的命令是类似这样调用的:

:Grep	[<grep_options>] [<search_pattern> [<file_name(s)>]]
:Rgrep	[<grep_options>] [<search_pattern> [<file_name(s)>]]
:Fgrep	[<grep_options>] [<search_pattern> [<file_name(s)>]]
:Rfgrep	[<grep_options>] [<search_pattern> [<file_name(s)>]]
:Egrep	[<grep_options>] [<search_pattern> [<file_name(s)>]]
:Regrep	[<grep_options>] [<search_pattern> [<file_name(s)>]]
:Agrep	[<grep_options>] [<search_pattern> [<file_name(s)>]]
:Ragrep	[<grep_options>] [<search_pattern> [<file_name(s)>]]
:GrepBuffer	[<grep_options>] [<search_pattern>]

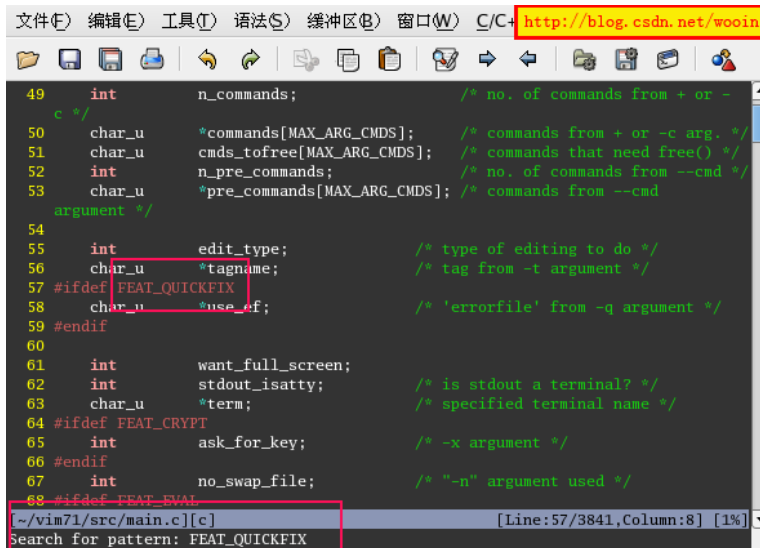


```
!Bgrep [<grep_options>] [<search_pattern>]
!GrepArgs [<grep_options>] [<search_pattern>]
```

但是我从来都不用敲上面那些命令的^_^, 因为我在~/.vimrc中增加了下面这句:

```
nnoremap <silent> <F3> :Grep<CR>
```

比如你想在/home/wooin/vim71/src/main.c中查找"FEAT_QUICKFIX", 则将光标移到"FEAT_QUICKFIX"上, 然后按下F3键, 如下图:



(--- 图14 ---)

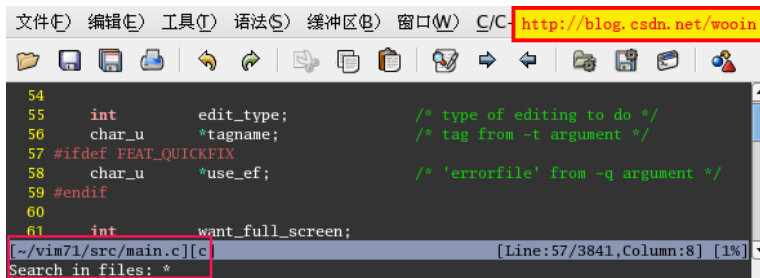
在最下面的命令行会显示:

```
Search for pattern: FEAT_QUICKFIX
```

此时你还可以编辑该行, grep支持正则表达式, 你想全词匹配的话可以改成:

```
Search for pattern: /<FEAT_QUICKFIX/>
```

然后按下回车:



(--- 图15 ---)

在最下面的命令行会显示:

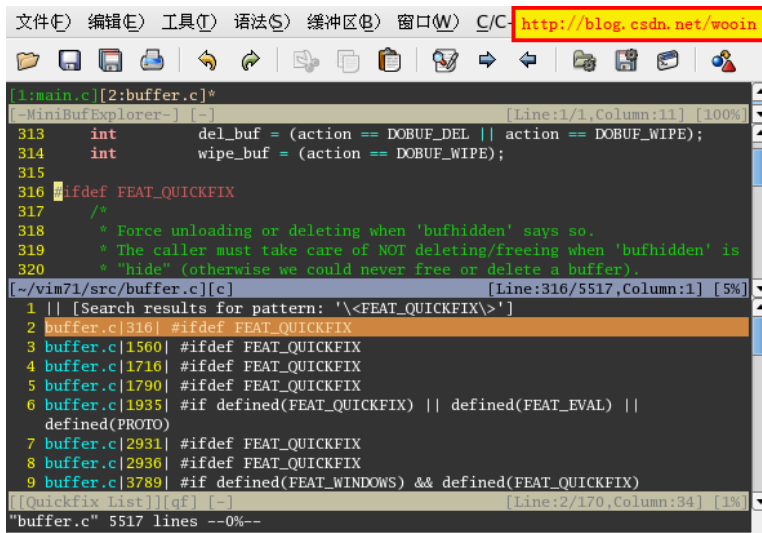
```
Search in files: *
```

是问你搜索范围, 默认是该目录下的所有文件, 此时你还可以编辑该行, 比如你只想搜索源码文件:

```
Search in files: *.c *.h
```

然后在按下回车, 会在弹出的QuickFix窗口中列出所有符合条件的搜索结果, 你可以在其中查找你想要的结果, 如下图:





(--- 图16 ---)

其实还有一些其他功能和设置,但是我都没有用过,这些功能再加上正则表达式,已经够我用了,其他的你可以在网页上看看它的文档,如果有什么惊人发现记得跟我互通有无,共同进步哦...

13. 高亮的书签 -- 插件: VisualMark

下载地址	http://www.vim.org/scripts/script.php?script_id=1026
版本	
安装	把visualmark.vim 文件丢到 ~/.vim/plugin 文件夹就好了
手册	无

下面介绍它的用法:

vim也和其他编辑器一样有"书签"概念,在vim中叫"Mark",可以用下面的命令查看相关说明:

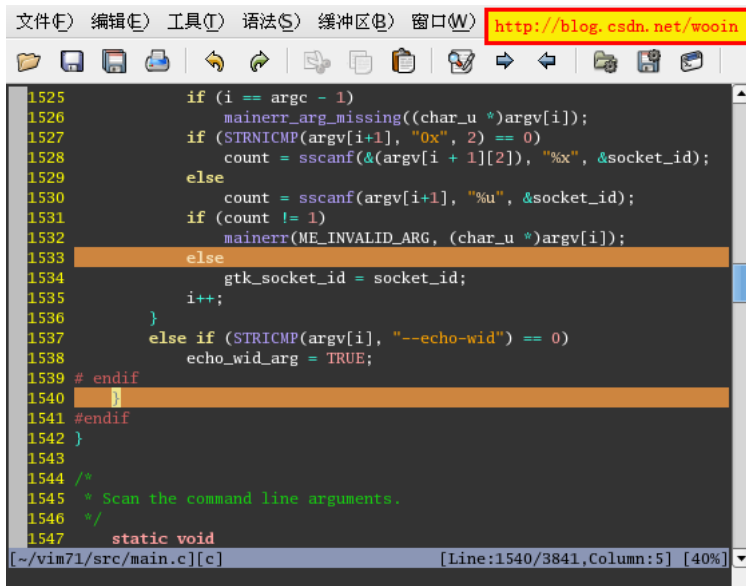
```
:help Mark
```

该"书签"有个很大的缺点:不可见.

我下面要介绍的Visual Mark插件跟vim中的"Mark"没有什么关系,并不是使其可见,而是自己本身就是"可见的书签",接着往下看就明白了,用作者的话说就是"类似UltraEdit中的书签".

另外,网上也有使vim中的Mark可见的插件,但是我试了一下,好像没Visual Mark好用,我就不介绍了.

按照上面的方法安装好Visual Mark后,你什么也不用设置,如果是gvim,直接在代码上按下Ctrl+F2,如果是vim,用"mm",怎么样,发现光标所在的行变高亮了吧,见下图:



(--- 图17 ---)

如果你设置了多个书签,你可以用F2键正向在期间切换,用Shift+F2反向在期间切换.

好了,我Visual Mark介绍完了,够简单吧^_^.

如果你嫌书签的颜色不好看, 你还可以自己定义, 不过是修改这个插件脚本的源码, 在目录 `~/vim/plugin/` 中找到并打开 `visualmark.vim`, 找到下面这段代码:

```
if &bg == "dark" // 根据你的背景色风格来设置不同的书签颜色
  highlight SignColor ctermfg=white ctermbg=blue guifg=wheat guibg=peru
else // 主要就是修改guibg的值来设置书签的颜色
  highlight SignColor ctermbg=white ctermfg=blue guibg=grey guifg=RoyalBlue3
endif
```

我还有几个不满意的地方:

- 1 这个书签不能自动保存, 关闭vim就没了.
- 2 切换书签时不能在不同文件间切换, 只能在同一个文件中切换

如果哪位朋友能解决这两个问题, 请一定要告诉寡人啊...还是用下面的地址:

Email	: lazy.fox.wu#gmail.com
Homepage	: http://blog.csdn.net/wooin

14. 自动补全

用过Microsoft Visual Studio的朋友一定知道代码补全功能, 输入一个对象名后再输入"."或者">", 则其成员名都可以列出来, 使Coding流畅了许多, 实现很多懒人的梦想, 现在我要告诉你, 这不再是Microsoft Visual Studio的专利了, vim也可以做到! 下面由我来教你, 该功能要tags文件的支持, 并且是ctags 5.6版本, 可以看看前文介绍tags文件的章节.

我这里要介绍的功能叫"new-omni-completion(全能补全)", 你可以用下面的命令看看介绍:

```
:help new-omni-completion
```

你还需要在 `~/vimrc` 文件中增加下面两句:

```
filetype plugin indent on
```

打开文件类型检测, 加了这句才可以用智能补全

```
set completeopt=longest,menu
```

关掉智能补全时的预览窗口

请确定你的Ctags 5.6已经安装好, 并且生成的tags文件已经可以用了, 那么我们就要抄家伙开搞了.

用vim打开源文件

```
$ vi /home/wooin/vim71/src/main.c
```

设置tags文件

```
:set tags=/home/wooin/vim71/tags
```

随便找一个有成员变量的对象, 比如"pamp", 进入Insert模式, 将光标放在">"后面, 然后按下"Ctrl+X Ctrl+O", 此时会弹出一个下列菜单, 显示所有匹配的标签, 如下图:



在你的~/.vimrc文件中加上这两句:

```
let g:SuperTabRetainCompletionType=2
let g:SuperTabDefaultCompletionType="<C-X><C-O>"
```

以后当你准备按"Ctrl+X Ctrl+O"的时候直接按<Tab>就好了, 够爽吧

我稍微再介绍一下上面那两句配置信息:

```
let g:SuperTabDefaultCompletionType="<C-X><C-O>"
" 设置按下<Tab>后默认的补全方式, 默认是<C-P>,
" 现在改为<C-X><C-O>. 关于<C-P>的补全方式,
" 还有其他的补全方式, 你可以看看下面的一些帮助:
" :help ins-completion
" :help compl-omni
```

```
let g:SuperTabRetainCompletionType=2
" 0 - 不记录上次的补全方式
" 1 - 记住上次的补全方式,直到用其他的补全命令改变它
" 2 - 记住上次的补全方式,直到按ESC退出插入模式为止
```

问题:

但是现在我的<Tab>键不好用了, 我以前爱用<Tab>进行缩进, 如果前面有字符按下<Tab>键后就会进行补全, 而不是我想要的缩进功能, 不知道有没有快捷键可以暂时关闭和激活SuperTab键的功能. 如果哪位朋友知道, 请一定记得告诉我啊....还是用下面的地址:

Email	: lazy.fox.wu#gmail.com
Homepage	: http://blog.csdn.net/wooin

@ Help poor children in Uganda!

"帮助可怜的乌干达儿童"

vim的作者开发了这么个强大, 好用的编辑器, 并且是完全开源, 完全免费的, 不知道比尔盖茨会不会觉得这个人真是不可理喻. 作者对用户用户的唯一期望就是帮助乌干达的儿童, 可以用命令:hlep kcc查看详细的内容,

- 1、捐助一个读小学的孩子: 每月 17 欧元 (或更多)。
- 2、捐助一个读中学的孩子: 每月 25 欧元 (或更多)。
- 3、捐助诊所: 每月或每季度, 数额不限。
- 4、一次性捐赠。

有条件的就资助一下, 没条件的就像我一样帮助宣传一下吧, 这也算是开源精神.....

=====

多谢各位兄弟捧场, 网页上图片显示的问题已经搞定了, 可能过阵子有空了还会对本文进行升级, 主要是增加一些新插件的使用介绍, 如果大家有好用的插件也可以推荐给我.

不知道哪儿有空间可以将该文上传供大家下载, 免得邮件麻烦.

=====

回复大家:

=====

deathmual@gmail.com 发表于2007-10-31 11:55:36 IP: 192.168.3.*

收一份先, 谢谢了, 在这里看不到图片, 可能是公司的网络问题.

不好意思, 我文章的问题, 我已经改好了, 可以看到图片了

duyanning 发表于2007-10-31 12:02:19 IP: 202.117.10.*

IDE?

能在vim中调试程序吗?



本文介绍的暂时还不行, 但是有一些插件可以支持gdb, 我试了一下不怎么好用, 我会继续关注的.

dlwxn 发表于2007-11-01 10:33:48 IP: 211.143.246.*

还有你前面的超链接错误.

本文会教你:

1. 中文帮助手册的安装
2. vim编程常用命令

file:///C:/Documents%20and%20Settings/Administrator/桌面
/programming_with_gvim_05.tar/programming_with_gvim_05/programming_with_gvim.html#2

是的, 我晚点抽空改一下, 没想到问题这么多...大家先将就看吧, 抱歉^_^

lbaby 发表于2007-11-01 10:34:47 IP: 222.128.6.*

弱弱的问一句: 为啥版权是归你和你老婆 所有?

我可以肯定的是, 这块好铁是你打出来的.

我老婆太可爱了嘛^_^



[上一篇](#) 《Linux设备驱动程序（第三版）》学习笔记之三: sleepy设备的使用

[下一篇](#) 我的vimrc

主题推荐

[编程](#)

[ide](#)

[linux内核](#)

[正则表达式](#)

[全局变量](#)

猜你在找

[Ubuntu下vim+ctags的配置](#)

[“抄袭事件”判决书](#)

[moto & google笔试题目-STL/C++面试题](#)

[KMP算法原理与实现 \(精简\)](#)

[自己选择的路、跪着也要走完](#)

[见过最好的git入门教程](#)

[“割绳子”的作者, 你如此歧视、无视、鄙视中国](#)

[linux驱动面试题目汇总](#)

[C++学习之深入理解虚函数--虚函数表解析](#)

[GDB 查看死锁](#)

[查看评论](#)

暂无评论

[查看更多评论](#)

发表评论

用户名:

评论内容: [插入代码](#)

* 以上用户言论只代表其个人观点, 不代表CSDN网站的观点或立场

核心技术类目

[全部主题](#) [Hadoop](#) [AWS](#) [移动游戏](#) [Java](#) [Android](#) [iOS](#) [Swift](#) [智能硬件](#) [Docker](#)
[OpenStack](#) [VPN](#) [Spark](#) [ERP](#) [IE10](#) [Eclipse](#) [CRM](#) [JavaScript](#) [数据库](#) [Ubuntu](#)
[NFC](#) [WAP](#) [jQuery](#) [BI](#) [HTML5](#) [Spring](#) [Apache](#) [.NET](#) [API](#) [HTML](#) [SDK](#) [IIS](#)

