

## Gvim开发环境配置笔记--Windows篇

环境: Windows7 旗舰版

1、安装Gvim7.3 : 下载地址<http://www.vim.org/download.php#pc>。

2、安装中文帮助: vimcdoc-1.8.0-setup.exe 地

址<http://vimcdoc.sourceforge.net/>

会自动识别gvim的安装路径,

安装完后, gvim菜单中文出现乱码, 在\_vimrcset文件中增加:

” 配置多语言环境, 解决中文乱码问题

```
if has("multi_byte")
    " UTF-8 编码
    set encoding=utf-8
    set termencoding=utf-8
    set formatoptions+=mM
    set fencs=utf-8,gbk
    if v:lang =~? '^(zh/)/|/(ja/)/|/(ko/)'
        set ambiwidth=double
    endif
    if has("win32")
        source $VIMRUNTIME/delmenu.vim
        source $VIMRUNTIME/menu.vim
        language messages zh_CN.utf-8
    endif
else
    echoerr "Sorry, this version of (g)vim was not compiled with
+multi_byte"
endif
```

输入: help, 显示中文帮助, 说明安装成功:

## 导航

[博客园](#)

[首页](#)

[新随笔](#)

[联系](#)

[订阅](#) [XML](#)

[管理](#)

## 统计

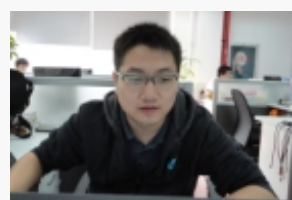
随笔 - 31

文章 - 57

评论 - 87

引用 - 0

## 公告

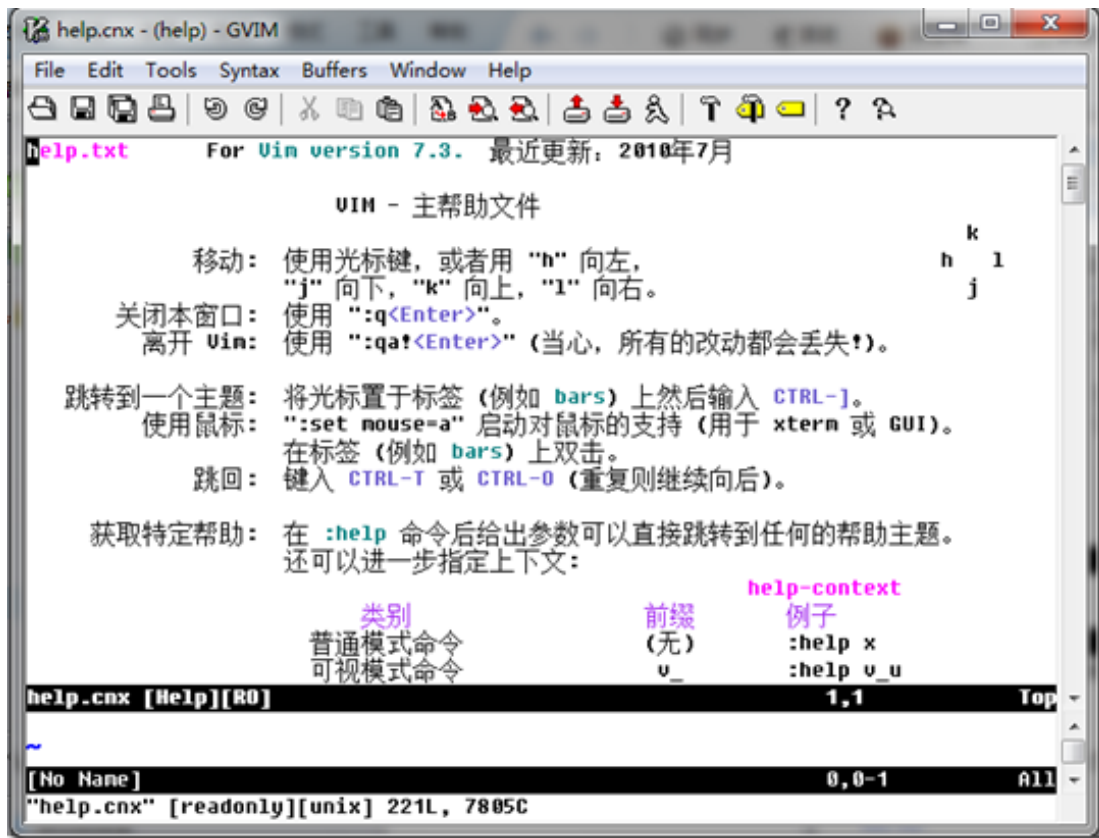


这个世界并不会在意你的自尊,  
而是要求你在自我感觉良好之前先有所成就

新浪微博: @红心李

不堪回首: @Junotes





### 3、设置语法高亮

编辑安装目录下的\_vimrc文件（例如：我的在D:\Program Files\Vim）

加入以下内容：

```
set nu!

colorscheme desert

syntax enable

syntax on
```

再打开gvim，打开一个源代码文件：



[我的豆瓣主页](#)

[豆瓣douban.com](#)



昵称: [红心李](#)  
 园龄: 7年1个月  
 粉丝: 80  
 关注: 4  
[+加关注](#)

搜索

找找看

谷歌搜索

常用链接

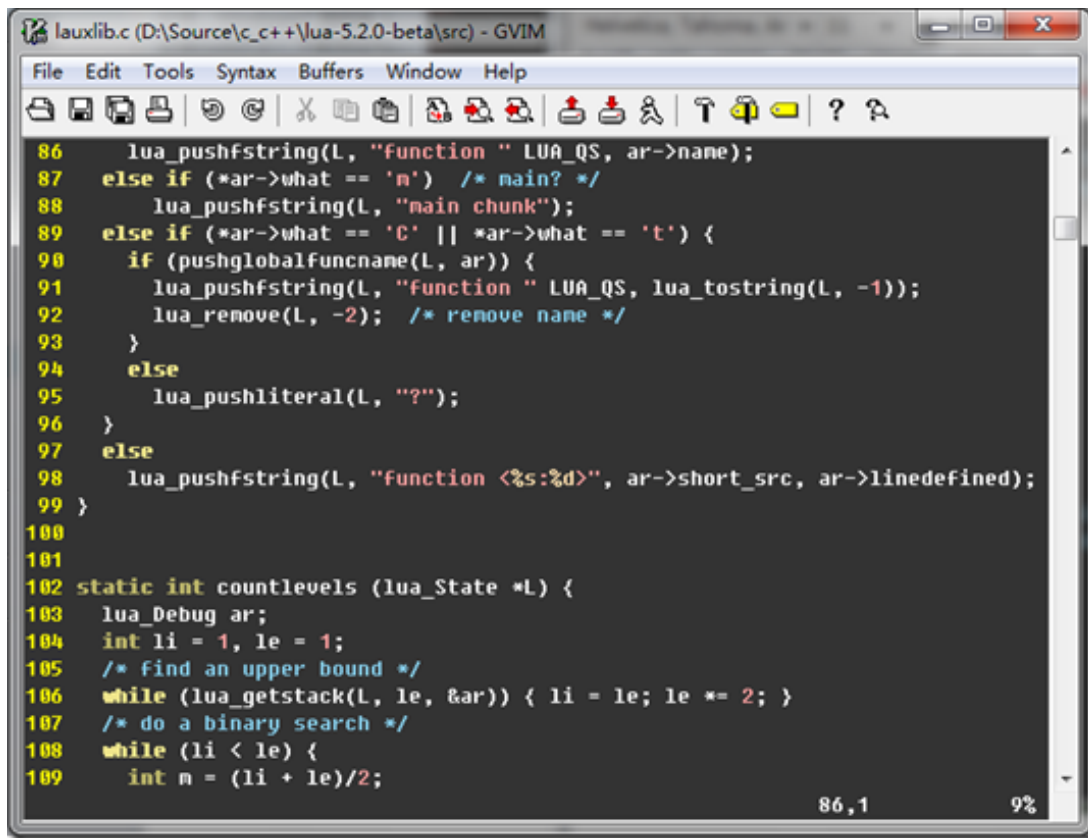
[我的随笔](#)  
[我的评论](#)  
[我的参与](#)  
[最新评论](#)  
[我的标签](#)

我的标签

[Instagram架构](#) (3)  
[MINA](#) (3)  
[哈希表](#) (2)  
[内存池](#) (1)  
[图像处理](#) (1)  
[进程通信 内存映射](#) (1)  
[linux sar](#) (1)  
[Nginx](#) (1)  
[Programming in Objective C](#) (1)  
[Quora 架构](#) (1)  
[更多](#)

随笔分类 (37)

[.NET](#) (1)  
[algorithm](#) (3)  
[Architecture](#) (4)  
[C\\_C++](#) (7)  
[image processing](#) (1)  
[iOS Develop](#) (1)  
[Java](#) (3)



这些设置使得gvim可以显示行号，并使用了desert配色方案，而且打开了语法高亮功能（用不同颜色显示注释、关键字、字符串等）。

我们还可以让函数名也高亮起来，

这里对C、C++的代码进行配置：在D:\Program Files\Vim\vim73\syntax下找到 c.vim 和 cpp.vim，分别添加以下内容：

```
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
```

重新打开gvim，效果如下：

[linux \(4\)](#)  
[Python \(2\)](#)  
[Thinking \(1\)](#)  
[Web \(7\)](#)  
[互联网观察 \(3\)](#)

随笔档案 (31)

2015年6月 (1)  
2014年10月 (1)  
2014年9月 (1)  
2014年8月 (1)  
2012年11月 (1)  
2012年10月 (2)  
2012年9月 (1)  
2012年8月 (3)  
2012年7月 (2)  
2012年5月 (4)  
2012年4月 (7)  
2012年2月 (2)  
2012年1月 (4)  
2011年12月 (1)

积分与排名

积分 - 58381  
排名 - 3279

最新评论

[1. Re:Gvim开发环境配置笔记-Windows篇](#)  
perfect!

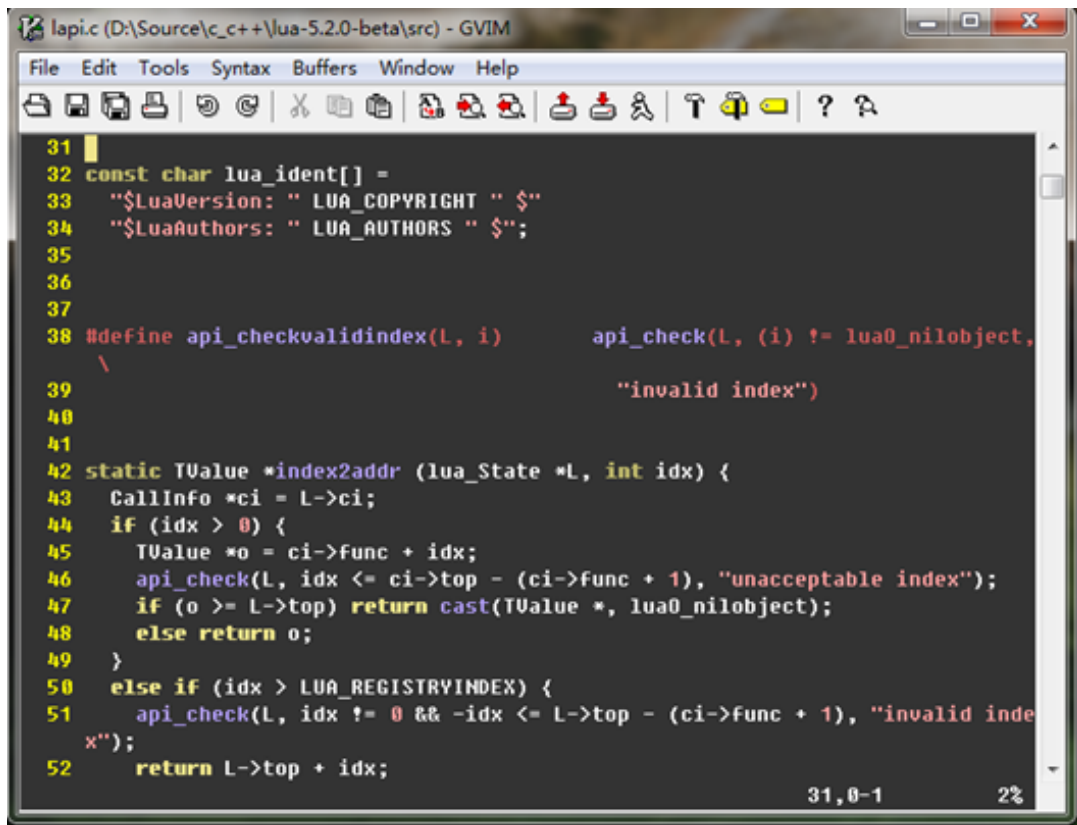
[2. Re:CentOS5.5编译安装gvim.7.3 失败记录](#)  
谢谢喽！！！！  
--hou52

[3. Re:Gvim开发环境配置笔记-Windows篇](#)  
用心了  
--ji

[4. Re:Gvim开发环境配置笔记-Windows篇](#)  
文章很有价值，对大家帮助很大。知识在于分享，不吝啬分享知识的人都是伟大的！！在 ” python编译插 VimPdb “一节，通过修改 \_vimrc来解决调试经常卡的问题。我遇到一个问题，很景使用。问.....  
--11

[5. Re:Gvim开发环境配置笔记-Windows篇](#)  
收藏啦，感谢博主  
--失落水

阅读排行榜



```
31
32 const char lua_ident[] =
33 "$LuaVersion: " LUA_COPYRIGHT " $"
34 "$LuaAuthors: " LUA_AUTHORS " $";
35
36
37
38 #define api_checkvalidindex(L, i)      api_check(L, (i) != lua0_nilobject,
39 \
40                                     "invalid index")
41
42 static TValue *index2addr (lua_State *L, int idx) {
43     CallInfo *ci = L->ci;
44     if (idx > 0) {
45         TValue *o = ci->func + idx;
46         api_check(L, idx <= ci->top - (ci->func + 1), "unacceptable index");
47         if (o >= L->top) return cast(TValue *, lua0_nilobject);
48         else return o;
49     }
50     else if (idx > LUA_REGISTRYINDEX) {
51         api_check(L, idx != 0 && -idx <= L->top - (ci->func + 1), "invalid index");
52         return L->top + idx;
```

#### 4、程序中跳转

将ec57w32.zip解压，在解压后将文件夹中的ctags.exe复制到

D:\ProgramFiles\Vim\vim73下，并编辑\_vimrc文件，添加以下内容：

```
set tags=tags;
set autochdir
```

然后将D:\ProgramFiles\Vim\vim73加到环境变量的path中。在需要查看的源代码目录下执行命令ctags -R；然后再用gvim打开源代码文件：

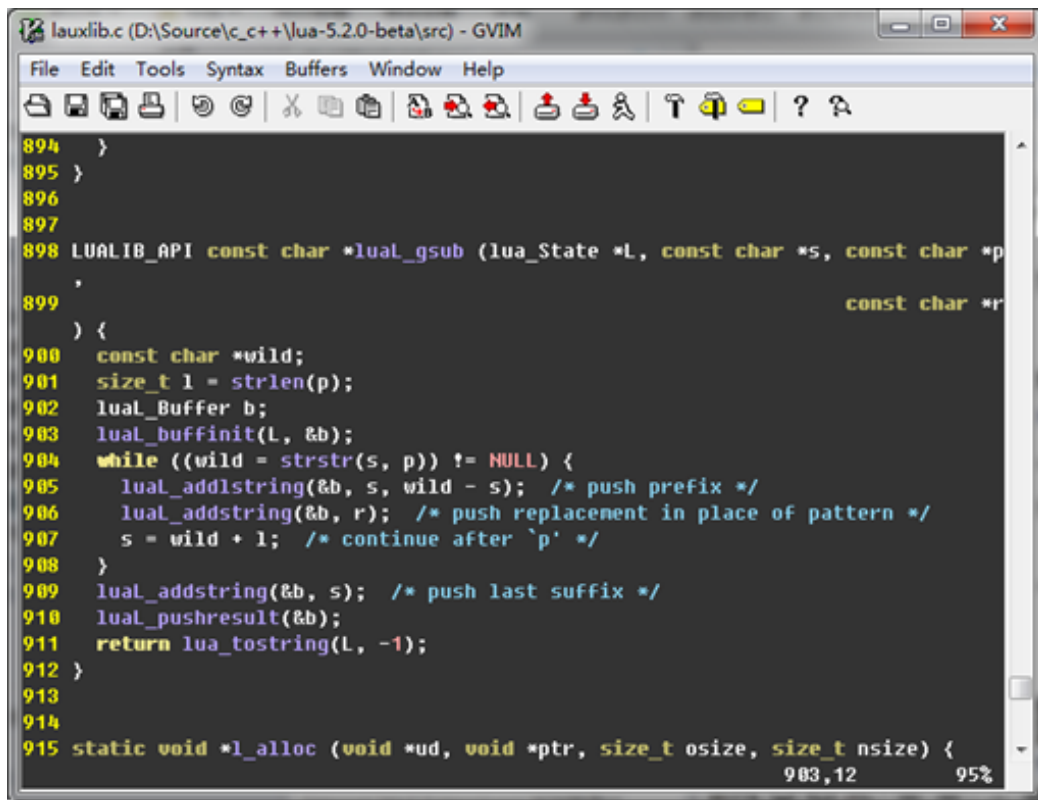
1. Gvim开发环境配置笔记 Windows篇 (63757)
2. 哈希表的C实现（一） (9958)
3. CentOS搭建python开发环境 (9282)
4. nginx源码学习——内存管理 (8706)
5. Mina框架研究（3） (748)
6. 哈希表的C实现（三）——传说中的暴雪版 (5748)
7. Quora的技术探索 (5506)
8. Mina框架研究（1） (498)
9. Linux命令扫盲 之 sed (4732)
10. Mina框架研究（2） (4732)

#### 评论排行榜

1. Gvim开发环境配置笔记 Windows篇 (17)
2. 图像处理学习之路 —— 从 BMP (10)
3. Quora的技术探索 (9)
4. 哈希表的C实现（三）——传说中的暴雪版 (7)
5. Windows进程通信 —— 共享内存（1） (6)
6. nginx源码学习——内存管理 (5)
7. Mina框架研究（2） (4)
8. Mina框架研究（3） (4)
9. 认识http协议(一) (4)
10. 你真的会使用assert吗？ (4)

#### 推荐排行榜

1. Gvim开发环境配置笔记 Windows篇 (20)
2. Quora的技术探索 (16)
3. 图像处理学习之路 —— 从 BMP (13)
4. nginx源码学习——内存管理 (6)
5. Log4Net指南 (5)
6. CentOS搭建python开发环境 (4)
7. 认识http协议(一) (4)
8. Instagram的技术探索 (4)
9. Instagram的技术探索 (4)
10. 哈希表的C实现（三）——传说中的暴雪版 (3)



```
luaolib.c (D:\Source\c_c++\lua-5.2.0-beta\src) - GVIM
File Edit Tools Syntax Buffers Window Help
894 }
895 }
896
897
898 LUA_LIB_API const char *luaL_gsub (lua_State *L, const char *s, const char *p
,
899                                     const char *r
) {
900     const char *wild;
901     size_t l = strlen(p);
902     luaL_Buffer b;
903     luaL_buffinit(L, &b);
904     while ((wild = strstr(s, p)) != NULL) {
905         luaL_addstring(&b, s, wild - s); /* push prefix */
906         luaL_addstring(&b, r); /* push replacement in place of pattern */
907         s = wild + l; /* continue after 'p' */
908     }
909     luaL_addstring(&b, s); /* push last suffix */
910     luaL_pushresult(&b);
911     return lua_tostring(L, -1);
912 }
913
914
915 static void *l_alloc (void *ud, void *ptr, size_t osize, size_t nsize) {
983,12 95%
```

按住“CTRL”键，点击对应的函数名或“CTRL+】”，会自动跳转到函数的定义部分，“CTRL+T”则返回；

## 5、源代码分析工具 taglist

将taglist\_45.zip解压，解压后包含一个doc文件夹和一个plugin文件夹，将其中内容分别复制到d:\Program Files\Vim\vim73下的doc及plugin中。

在\_vimrc文件中加入以下内容：

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





src\_id=754

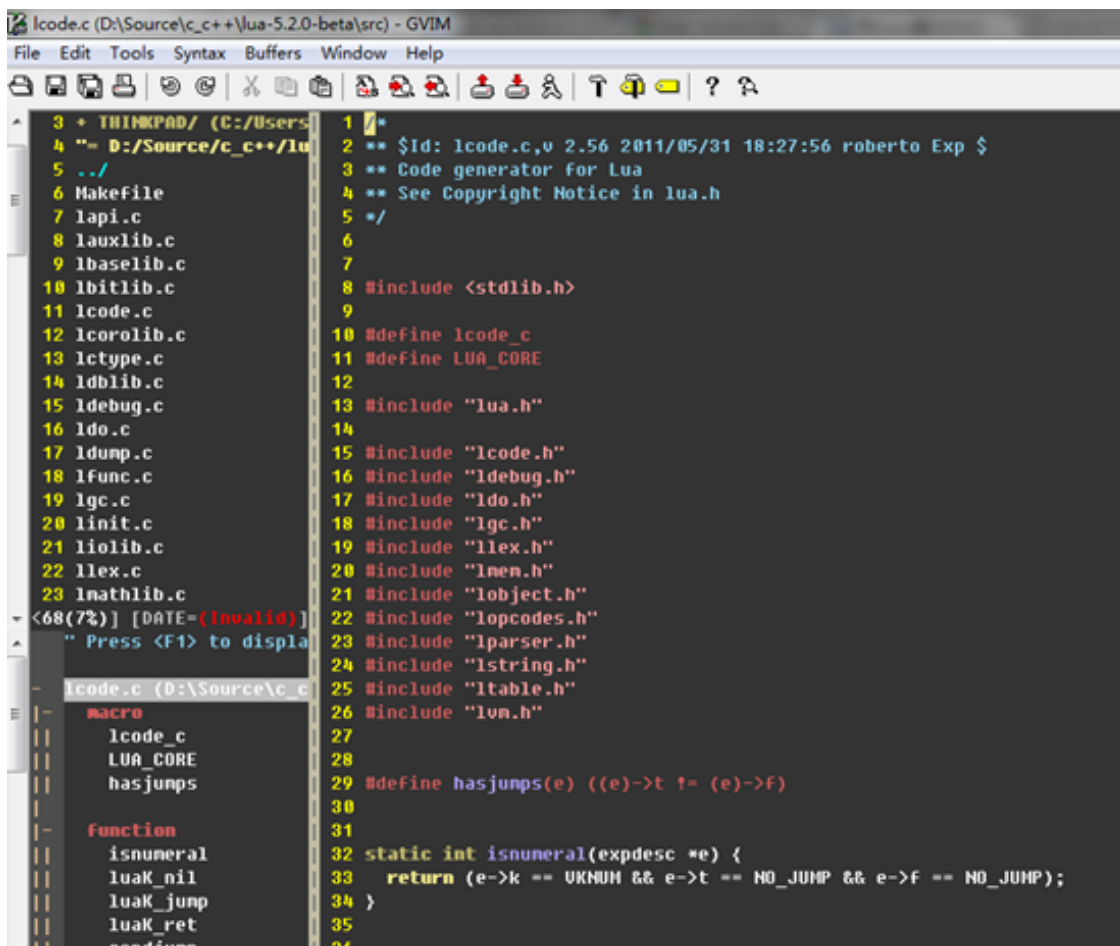
将winmanager.zip解压和拷贝，解压后包含一个doc文件夹和一个plugin文件夹，将其中内容分别复制到d:\Program Files\Vim\vim73下的doc及plugin中

在\_vimrc文件中加入以下内容：

```
let g:winManagerWindowLayout='FileExplorer|TagList'
```

```
nmap wm :WMToggle<cr>
```

用gvim打开代码文件，normal状态下输入命令“wm”，窗口如下：



左上角是浏览文件的窗口，左下角的是TagList窗口；

## 7、多文档编辑

minibufexpl.vim 地址[http://www.vim.org/scripts/download\\_script.php](http://www.vim.org/scripts/download_script.php)

src\_id=3640

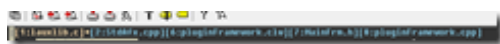
解压后将minibufexpl.vim复制到d:\Program Files\Vim\vim73\plugin，在\_vimrc中添加：

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

```
let g:miniBufExplMapWindowsNavVim=1
```

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

用GVIM打开多个源代码文件后，如图：



ctrl+Tab, 切换到前一个buffer, 并在当前窗口打开文件;

ctrl+shift+Tab, 切换到后一个buffer, 并在当前窗口打开文件;

ctrl+箭头键, 可以切换到上下左右窗口中;

ctrl+h, j, k, l, 切换到上下左右的窗口中。

## 8、在工程中快速查找

grep.vim 地址[http://www.vim.org/scripts/download\\_script.php?src\\_id=7645](http://www.vim.org/scripts/download_script.php?src_id=7645)

解压后把grep.vim文件拷贝到d:\Program Files\Vim\vim73\plugin, 在\_vimrc中添加:

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

因为windows下是不带grep的, 需要下载Grep for Windows

<http://gnuwin32.sourceforge.net/packages/grep.htm>

下载后在环境变量中增加grep的路径;

用gvim打开一个源代码文件, 光标选择需要查找的内容, 按F3, 确定要查找的内容和搜索范围, gvim会在弹出的QuickFix窗口中列出所有符合条件的搜索结果

```
398 LUALIB_API const char *lua_gsub (lua_State *L, const char *s, const char *p,
399                                     const char *r) {
400     const char *wild;
401     size_t l = strlen(p);
402     luaL_Buffer b;
403     luaL_buffinit(L, &b);
404     while ((wild = strstr(s, p)) != NULL) {
405         luaL_addlstring(&b, s, wild - s); /* push prefix */
406         luaL_addstring(&b, r); /* push replacement in place of pattern */
407         s = wild + l; /* continue after 'p' */
408     }
409     luaL_addstring(&b, s); /* push last suffix */
410     luaL_pushresult(&b);
411     return lua_tostring(L, -1);
412 }
413
414 auxlib.c
415 1 || [Search results for pattern: lua_call]
416 2 |api.c|832| LUA_API void lua_callk (lua_State *L, int nargs, int nresults, int ctx,
417 3 |auxlib.c|715| lua_call(L, 1, 1);
418 4 |auxlib.c|884| lua_call(L, 1, 1); /* open module */
419 5 |baselib.c|32| lua_call(L, 1, 1);
420 6 |baselib.c|205| lua_call(L, 1, 3); /* get 3 values from metamethod */
421 7 |baselib.c|306| lua_call(L, 0, 1); /* call it */
422 8 |baselib.c|365| lua_callk(L, 0, LUA_MULTRET, 0, dofilecont);
423 9 |dblib.c|271| lua_call(L, 2, 0);
424 10 |do.c|408| /* finish 'lua_callk' */
```

## 9、h\c切换 (针对C、C++)

下载插件: a.vim 地址[http://www.vim.org/scripts/download\\_script.php?](http://www.vim.org/scripts/download_script.php?src_id=7218)

[src\\_id=7218](http://www.vim.org/scripts/download_script.php?src_id=7218)

将a.vim复制到d:\Program Files\Vim\vim73\plugin, 在\_vimrc中添加:

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



用gvim打开源码文件后，按F12即可以在c/h文件中切换，也可以通过输入:A实现。

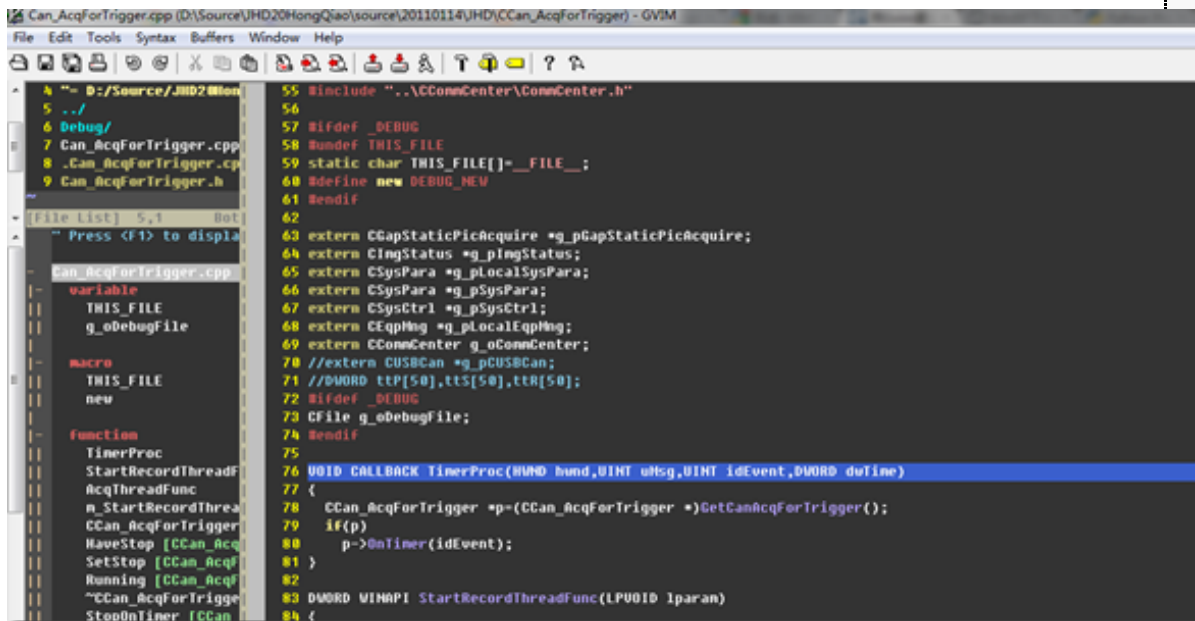
## 10、高亮书签

下载插件: visualmark.vim 地

址[http://www.vim.org/scripts/download\\_script.php?src\\_id=4700](http://www.vim.org/scripts/download_script.php?src_id=4700)

将visualmark.vim复制到d:\Program Files\vim\vim73\plugin。

用gvim打开源码文件，将光标定位在需要添加书签的地方，按下ctrl+F2，即添加了书签。



使用F2在书签之间正向切换，shift+F2反向切换。

## 11、Python代码补全

下载插件[pydictio-1.2.zip](#)

解压后，将里面的python\_pydictio.vim文件复制到D:\Program Files\vim\vimfiles\ftplugin

将complete-dict 和 pydictio.py复制到D:\Program Files\vim\vimfiles\ftplugin\pydictio

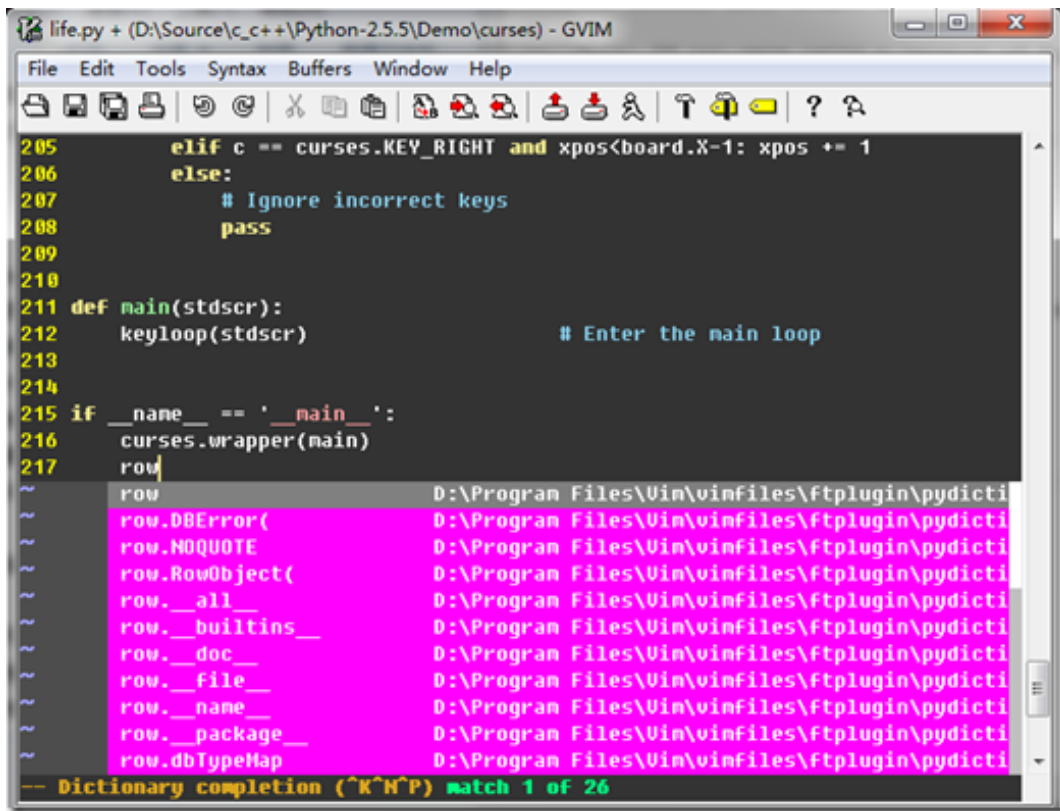
在\_vimrc文件中增加配置如下：

filetype plugin on “启用filetype插件

```
let g:pydictio_location = 'D:\Program  
Files\vim\vimfiles\ftplugin\pydictio\complete-dict'
```

let g:pydictio\_menu\_height = 20 ” 设置弹出菜单的高度，默认是15

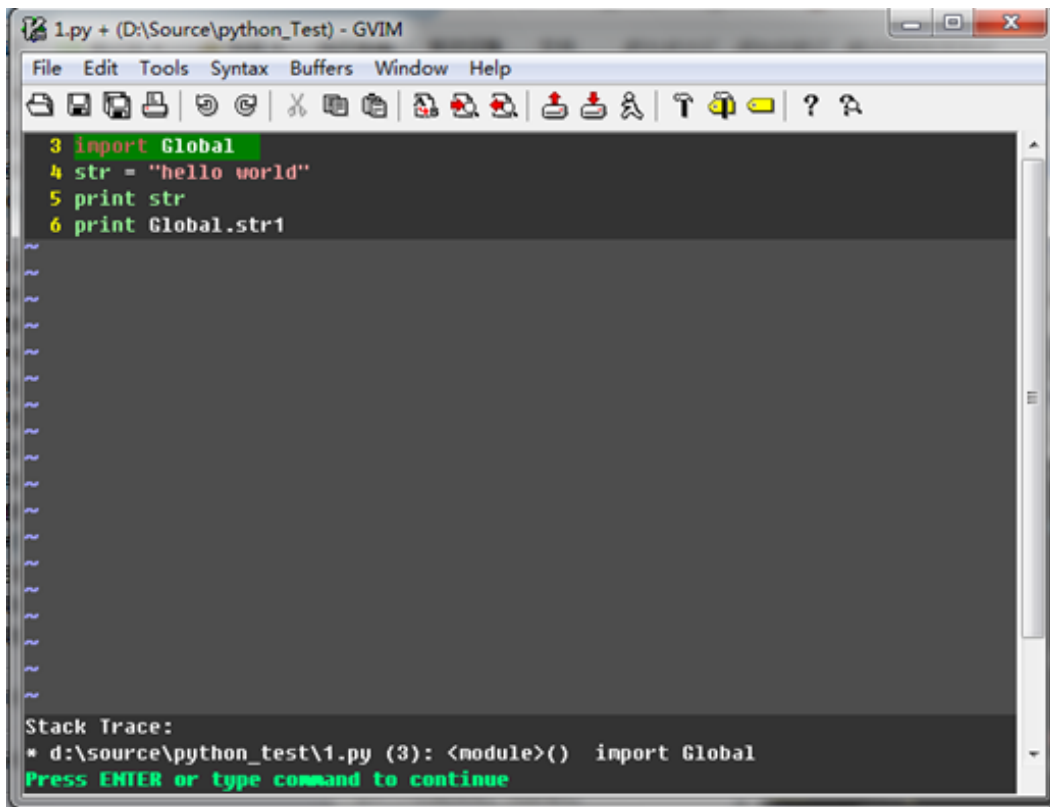
用gvim打开一个py文件，输入row+[tab]，就可以看到自动补全的代码选项了



## 12、python编译

下载VimPdb，解压后拷贝VimPdb.py，VimPdb.vim到plugin即可。

用vim代开python代码文件，按F5运行，然后按F2设置断点，在运行到断点后可以用按F12可以查看Stack Trace，F3查看变量和参数值。



不过这种方法使用了一下，经常会死掉，不知道为什么，所以在网上找到了另外一种方法：

在\_vimrc增加



```
python << EOF

import time
import vim
def SetBreakpoint():
    nLine = int( vim.eval( 'line(".")' ))
    strLine = vim.current.line
    i = 0
    strWhite = ""
    while strLine[i] == ' ' or strLine[i] == "\t":
        i += 1
        strWhite += strLine[i]
    vim.current.buffer.append(
        "%(space)spdb.set_trace() %(mark)s Breakpoint %(mark)s" %
        {'space':strWhite, 'mark': '#' * 30}, nLine - 1)
    for strLine in vim.current.buffer:
        if strLine == "import pdb":
            break
        else:
            vim.current.buffer.append( 'import pdb', 0)
            vim.command( 'normal j1')
            break
vim.command( 'map <C-M> :py SetBreakpoint()<cr>')

def RemoveBreakpoints():
    nCurrentLine = int( vim.eval( 'line(".")' ))
    nLines = []
    nLine = 1
    for strLine in vim.current.buffer:
        if strLine == 'import pdb' or strLine.lstrip()[:15] ==
'pdb.set_trace()':
            nLines.append( nLine)
            nLine += 1
    nLines.reverse()
    for nLine in nLines:
        vim.command( 'normal %dG' % nLine)
        vim.command( 'normal dd')
        if nLine < nCurrentLine:
            nCurrentLine -= 1
    vim.command( 'normal %dG' % nCurrentLine)
vim.command( 'map <C-U> :py RemoveBreakpoints()<cr>')

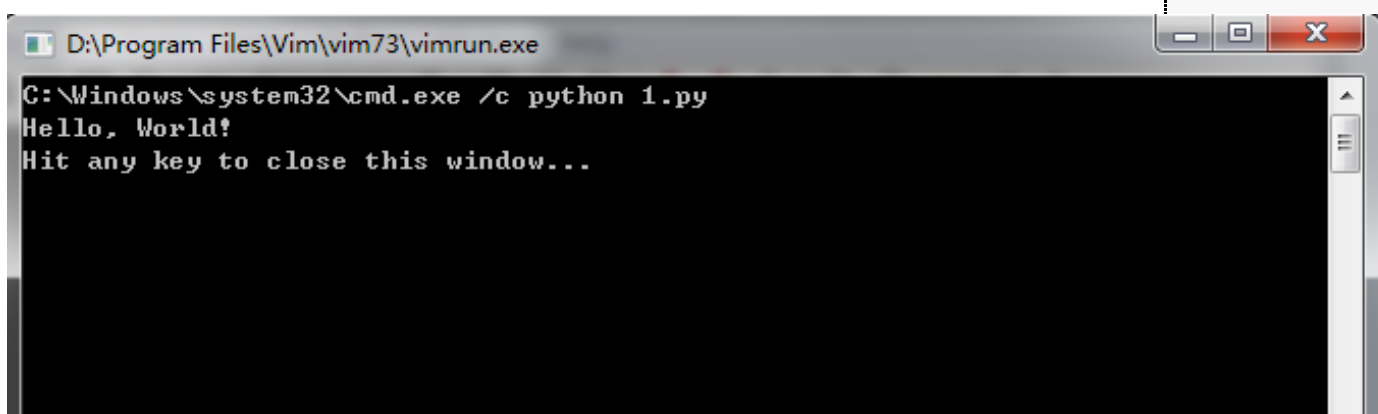
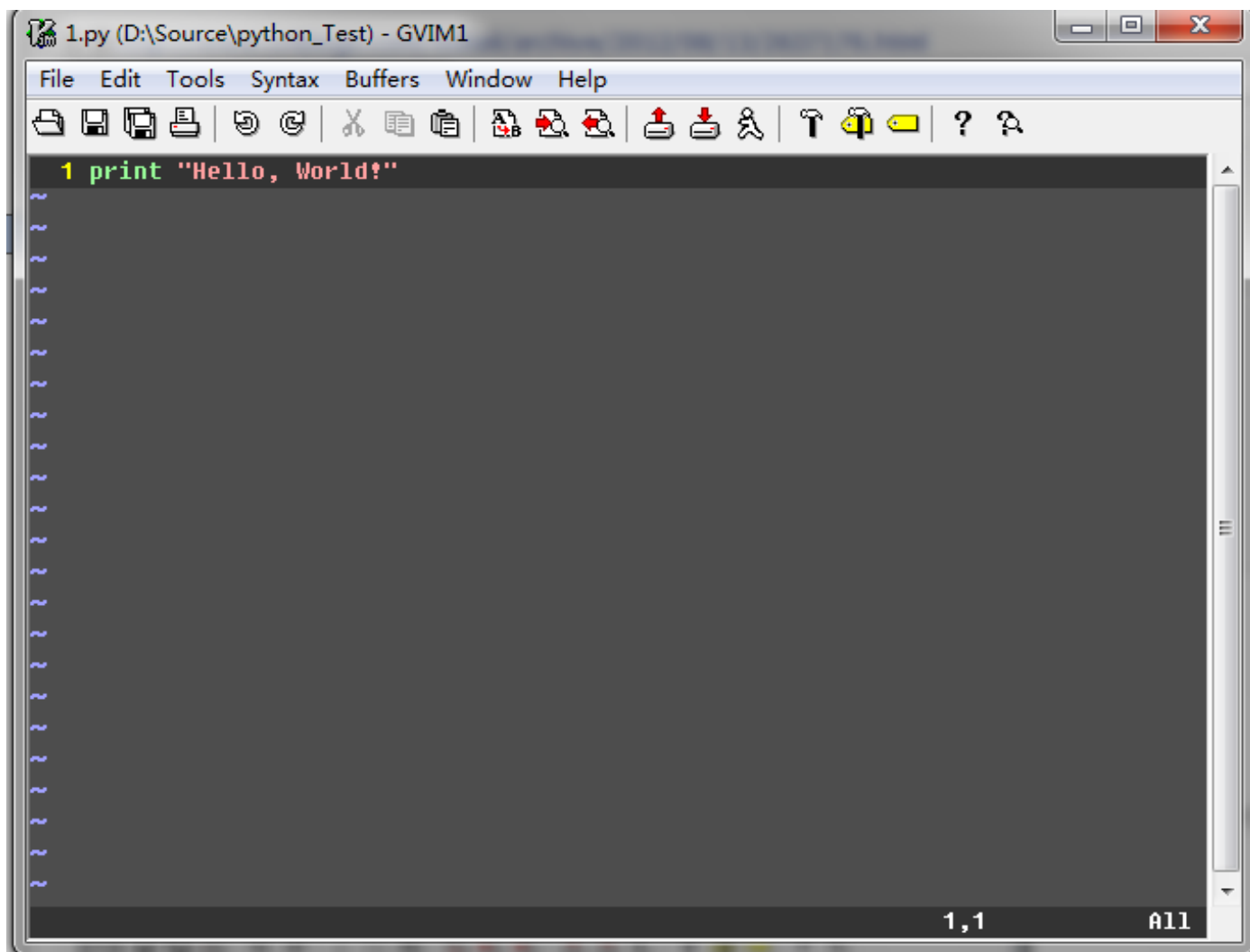
vim.command( 'map <C-D> :!python %<cr>')

EOF
```



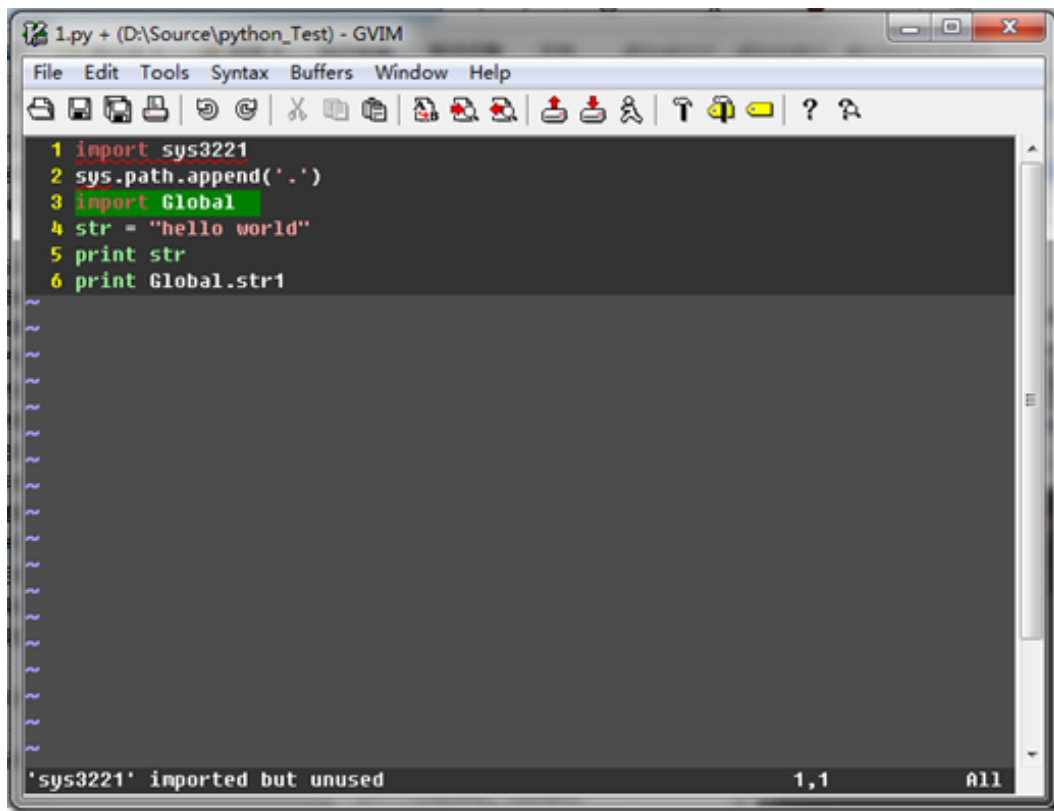
然后在要调试的代码里面用ctrl-M设断点

ctrl-D运行



### 13、Python代码检查

下载[pyflakes](#)，解压缩后把pyflakes.vim文件和pyflakes目录拷贝到ftplugin\python目录中，打开一个有问题的python源代码文件，执行命令:cc，即可进行代码检查：



分类: [Python](#), [linux](#)

标签: [gvim](#) [python](#) [windows](#)

绿色通道:

好文要顶

关注我

收藏该文

与我联系



[红心李](#)  
关注 - 4  
粉丝 - 80  
[+加关注](#)

20

0

推荐

反对

(请您对文章做出评价)

« 上一篇: [《Programming in Objective-C 3rd Edition》阅读笔记 \(1\)](#)  
» 下一篇: [Mina框架研究 \(1\)](#)

posted on 2012-08-13 22:58 [红心李](#) 阅读(63757) 评论(17) [编辑](#) [收藏](#)

评论

- #1楼 2012-08-14 08:03 [jay.windows](#)
- 多谢楼主~~~ 试试看 , 尤其是WM和在工程中快速查找很有用
- 支持(1) 反对(0)
- #2楼 2012-08-14 09:43 [qiang.xu](#)
- 不错, 顶一个
- 支持(0) 反对(0)
- #3楼 2012-08-14 10:08 [sumuu](#)
- 不错支持下..
- 支持(0) 反对(0)

#4楼 2012-08-14 16:52 四月不留 

在linux下 vim的ctag插件可以利用Ctrl+]跳转到指定的函数，但是在windows下只能按住Ctrl+鼠标左键跳转，不知为何。

支持(0) 反对(0)

#5楼 2012-08-14 16:53 四月不留 

楼主vim 工具里的第一个“跳转到tag” 快捷键上写着是g`J么

支持(0) 反对(0)

#6楼 2013-07-31 14:51 bing2011 

nice 非常感谢楼主 受教了

支持(0) 反对(0)

#7楼 2013-09-01 16:26 雪域迷城 

你好，LZ，我看了你的文章之后去配置函数的高亮，但是我加了之后还是没反应，而且我按你的第一步和第二步走没发现乱码，就没加上你深蓝的一段代码进去，是否跟这个有关系。

支持(0) 反对(0)

#8楼 2013-11-08 22:52 jiangnan314 

多谢

支持(0) 反对(0)

#9楼 2013-12-11 11:05 zhanwc 

ec57w32.zip这个文件在哪里下载呀

支持(0) 反对(0)

#10楼 2014-01-26 15:31 Spix Mojo 

写得挺好的。帮助很大。

支持(0) 反对(0)

#11楼 2014-03-31 16:39 Jack Ge 

不错

支持(0) 反对(0)

#12楼 2014-07-14 20:14 孤独剑客zzy 

mark

支持(0) 反对(0)

#13楼 2014-11-06 15:37 凜凍 

非常感谢楼主的教程

支持(0) 反对(0)

#14楼 2014-11-22 17:21 失落水的鱼 



#15楼 2015-01-22 20:00 [liguoo](#) 

文章很有价值，对大家帮助会很大。知识在于分享，不辞辛苦分享知识的人都是伟大的！！

在 ”python编译插件VimPdb“一节，通过修改\_vimrc来解决调试经常卡的问题。我遇到一个问题，很影响使用。

问题描述：设置的断点行没有正常缩进，pdb.set前还多出一个字符。如下

```
create_engine('root', 'jlg234bob', 'test')
updb.set_trace() ##### Breakpoint #####
update('drop table if exists users')
```

问题分析：下面代码出自函数SetBreakpoint()，行号32。

i最初为0，先加1变为1，然后通过strLine[i]来取空字符。这样会从第二个位置开始取，位置向后移动了一位。前面少去一个空字符，后面多取一个非空字符。因此导致上面的结果。

```
i = 0
strWhite = ""
while strLine[i] == ' ' or strLine[i] == "\t":
    i += 1
strWhite += strLine[i]
```

解决：行“i+=1”和下一行“strWhite += strLine[i]”上下位置调换。修改后如下：

```
i = 0
strWhite = ""
while strLine[i] == ' ' or strLine[i] == "\t":
    strWhite += strLine[i]
    i += 1
```

支持(0) 反对(0)

#16楼 2015-04-17 14:48 [jihite](#) 

用心了


支持(0) 反对(0)

#17楼 2015-07-11 11:18 [jiu~](#) 

perfect!

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

 注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#)网站首页。

[博客园首页](#) [博问](#) [新闻](#) [闪存](#) [程序员招聘](#) [知识库](#)

最新IT新闻：

- [特斯拉明年在中国销售SUV新车Model X](#)
- [阿里云数据库首推数据压缩功能 5倍压缩成本降80%](#)
- [美投资者“挽留”中概股 私有化热潮令美投资者不满](#)
- [罗玉凤你学不会](#)
- [软件也给力：中国已成App Store最大下载区](#)

» [更多新闻...](#)

最新知识库文章：

- [优秀程序员眼中的整洁代码](#)
  - [怎样看待比自己强的人](#)
  - [编程王道，唯“慢”不破](#)
  - [程序员要有持续产出](#)
  - [代码审查过程](#)
- » [更多知识库文章...](#)