Vim环境配置和插件安装

# 环境配置

vim环境配置需要在.vim/vimrc中配置自己的vim配置选项。应用于全局的配置选项需要在/etc/vim/vimrc中配置，也可以新建/etc/vim/vimrc.local，并在这个文件中配置自己的vim配置选项。

# 插件安装

## vim-plug

这里我选择先安装vim插件管理器vim-plug进行插件管理，vim-plug本身就是一个插件，github下载地址为：

<https://github.com/junegunn/vim-plug.git>

安装和使用均参考vim-plug.README文档：简单来说，安装就是git clone vim-plug仓，然后拷贝plug.vim到/vim/share/vim/vim82/autoload目录下。

vim-plug安装完毕后需要在vimrc文件中添加以下代码段：

call plug#begin("/home/guo/.vim/plugged")

" The default plugin directory will be as follows:

" - Vim (Linux/macOS): '~/.vim/plugged'

" - Vim (Windows): '~/vimfiles/plugged'

" - Neovim (Linux/macOS/Windows): stdpath('data') . '/plugged'

" You can specify a custom plugin directory by passing it as the argument

" - e.g. `call plug#begin('~/.vim/plugged')`

" - Avoid using standard Vim directory names like 'plugin'

" Initialize plugin system

Plug 'https://github.com/preservim/nerdtree.git', { 'on': 'NERDTreeToggle' }

Plug 'https://github.com/vim-scripts/gtags.vim.git'

Plug 'https://github.com/ycm-core/YouCompleteMe.git', { 'do': './install.py' }

Plug 'https://github.com/vim-scripts/taglist.vim.git'

call plug#end()

Tips：

其中plug#begin(<通过plug-vim安装的插件地址>)，这里由于草蛋的YCM安装脚本install.py不允许以root用户运行，因此我这里只能将插件地址安装在本地文件目录下，并把plugged目录下所有文件owner都改成了本地用户，而不是超级用户。

安装插件的方式有很多，可以输入git仓名（不行），或者输入git链接地址（可以）进行下载，再或者先下载到本地然后再安装。同时，安装插件的时候可以添加额外选项，如：{‘on’: ‘NERDTreeToggle’}当NERDTreeToggle命令使用的时候再加载插件；{‘do’: ‘./install.py -all’}安装完成后执行./install.py –all命令等。以上安装方式均可以在vim-plug例程中找到。

配置完成后按:wq保存退出。然后重新打开vim，使用:PlugInstall命令自动安装和加载插件。（国内受网络限制影响可能会出现安装失败的问题，如有出现TLS error的问题大概率是由于网络原因，重新下载几次就可以）。

## Nerdtree

Nerdtree是vim目录查看工具，通过vim-plug进行下载和安装，github链接为：

https://github.com/preservim/nerdtree.git

Nerdtree安装完毕:wq退出vim再打开就可以了，要注意的是先在vimrc中配置几个选项：

快捷键：

nnoremap <leader>n :NERDTreeFocus<CR>

nnoremap <C-n> :NERDTree<CR>

nnoremap <C-t> :NERDTreeToggle<CR>

nnoremap <C-f> :NERDTreeFind<CR>

打开vim自动弹出nerdtree:

" Start NERDTree and leave the cursor in it.

autocmd VimEnter \* NERDTree

关闭vim自动关闭nerdtree:

" Exit Vim if NERDTree is the only window remaining in the only tab.

autocmd BufEnter \* if tabpagenr('$') == 1 && winnr('$') == 1 && exists('b:NERDTree') && b:NERDTree.isTabTree() | quit | endif

## gtags

gtags是基于global的函数查找工具，github链接为：

https://github.com/vim-scripts/gtags.vim.git

需要配合global工具进行安装，使用说明在global官网：

<http://www.gnu.org/software/global/>.

可以通过nnoremap添加快捷键，需要事先在代码目录下使用gtags命令生成三个索引文件方可使用。

### Youcompleteme

又称YCM，传说中最难安装的vim插件，github链接：

https://github.com/ycm-core/YouCompleteMe.git

最稳妥的方法是按照README一步一步来进行安装，难点在于国内的垃圾网络，由于ycm仓库下依赖一大堆的submodule，有许多因为网络原因使得安装无法进行，因此需要手动一个一个的git clone到对应的目录下面；安装完成后还需要到安装目录下执行python3 install.py –all，这个命令会帮助你完成安装clangd和libclang，仍然是由于网络原因，clangd或libclang需要手动通过apt安装，再通过配置vim全局变量[g:ycm\_clangd\_binary\_path](https://github.com/ycm-core/YouCompleteMe/blob/master)使ycm可以识别clangd自动补全工具。

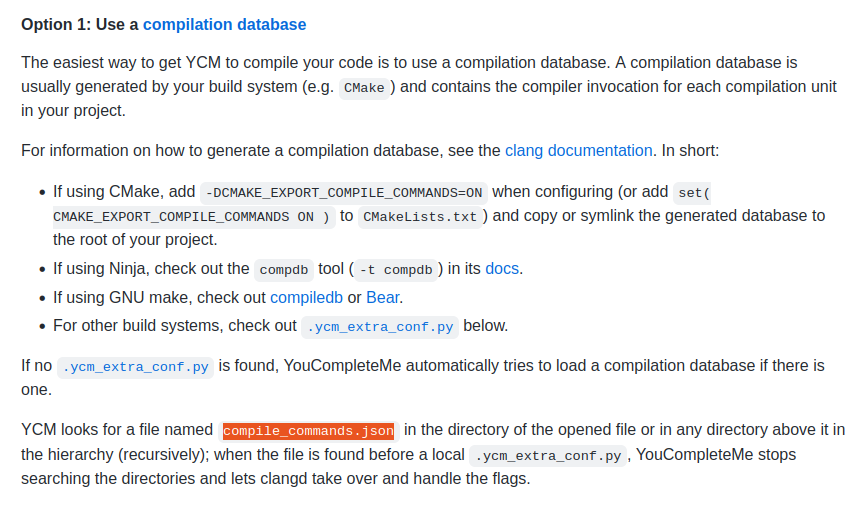
然后通过配置.ycm\_extra\_conf.py文件，或者使用compiledb工具执行compiledb make bzImage modules生成补全数据库文件compile\_commands.json后,配合ycm已经clangd才能正确的进行代码补全。

ycm安装：

clangd安装：



编译数据库安装:



## taglist

taglist是基于ctags的符号查找插件，可以查找源代码中的符号，github源代码目录为：

https://github.com/vim-scripts/taglist.vim

需要配合ctags生成tag文件进行查找，帮助文档为：

[http://vim-taglist.sourceforge.net/manual.html#taglist-using](http://vim-taglist.sourceforge.net/manual.html" \l "taglist-using)

也可以在taglist目录下通过快捷键F1帮助查找。