1 使用apt-get时遇到资源不可用问题  
E: 无法获得锁 /var/lib/dpkg/lock - open (11: 资源暂时不可用) E: 无法锁定管理目录(/var/lib/dpkg/)，是否有其他进程正占用它？

网上搜了一下原因，说是有另外一个程序在运行，导致锁不可用，原因可能是赏析运行更新或安装没有正常完成。这是因为上次更新或者安装没有正常完成。

网上的两种解决方法：

1、终端输入ps -aux命令，列出进程，找到含有apt-get 的进程，直接sudo kill PID解决。不过好像没找着qpt-get的进程。

没办法，此方法不行，只能使用第二种解决方案。

2、强制解锁命令：

sudo rm /var/cache/apt/archives/lock

sudo rm /var/lib/dpkg/lock

## 2 如何源码安装VIM

1. 检查vim是否支持python vim --version

2. 下载vim8源码

git clone https://github.com/vim/vim.git

2. 编译安装vim8

cd vim/src

make clean

./configure --with-features=huge --enable-python3interp --enable-pythoninterp --with-python-config-dir=/usr/lib/python2.7/config-x86\_64-linux-gnu/ --enable-rubyinterp --enable-luainterp --enable-perlinterp --with-python-config-dir=/usr/lib/python2.7/config-x86\_64-linux-gnu/ --enable-multibyte --enable-cscope --prefix=/usr/local/vim/

sudo make install

备注说明:重新安装也可以make distclean

--with-features=huge：支持最大特性  
--enable-rubyinterp：打开对ruby编写的插件的支持  
--enable-pythoninterp：打开对python编写的插件的支持  
--enable-python3interp：打开对python3编写的插件的支持  
--enable-luainterp：打开对lua编写的插件的支持  
--enable-perlinterp：打开对perl编写的插件的支持  
--enable-multibyte：打开多字节支持，可以在Vim中输入中文  
--enable-cscope：打开对cscope的支持  
--with-python-config-dir=/usr/lib/python2.7/config-x86\_64-linux-gnu/ 指定python 路径  
--with-python-config-dir=/usr/lib/python3.5/config-3.5m-x86\_64-linux-gnu/ 指定python3路径  
--prefix=/usr/local/vim：指定将要安装到的路径(自行创建)

如果出现问题请自得安装python-dev 再执行上面命令

sudo apt-get install python-dev  
sudo apt-get install python3-dev（很重要，前提条件）  
sudo apt-get install libncurses5-dev

Python2.7不再支持诸如YCM类插件，以后配置VIM均使用3.5

example

1241 vim --version

1242 make distclean

1243 sudo ./configure --with-features=huge --enable-python3interp --enable-multibyte --enable-cscope ---with-python-config-dir=/usr/lib/python3.5/config-3.5m-x86\_64-linux-gnu/

1244 sudo ./configure --with-features=huge --enable-python3interp --enable-multibyte --enable-cscope --with-python-config-dir=/usr/lib/python3.5/config-3.5m-x86\_64-linux-gnu/

1245 sudo make

1246 sudo make install

###### 3 如何安装vim-plug Unix, Linux

sh -c 'curl -fLo "${XDG\_DATA\_HOME:-$HOME/.local/share}"/nvim/site/autoload/plug.vim --create-dirs \

<https://raw.githubusercontent.com/junegunn/vim-plug/master/plug.vim>'

把plug.vim放到~/.vim/autoload文件夹下

#### Example

" Specify a directory for plugins

" - For Neovim: stdpath('data') . '/plugged'

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

call plug#begin('~/.vim/plugged')

" Make sure you use single quotes

" Shorthand notation; fetches https://github.com/junegunn/vim-easy-align

Plug 'junegunn/vim-easy-align'

" Any valid git URL is allowed

Plug 'https://github.com/junegunn/vim-github-dashboard.git'

" Multiple Plug commands can be written in a single line using | separators

Plug 'SirVer/ultisnips' | Plug 'honza/vim-snippets'

" On-demand loading

Plug 'scrooloose/nerdtree', { 'on': 'NERDTreeToggle' }

Plug 'tpope/vim-fireplace', { 'for': 'clojure' }

" Using a non-master branch

Plug 'rdnetto/YCM-Generator', { 'branch': 'stable' }

" Using a tagged release; wildcard allowed (requires git 1.9.2 or above)

Plug 'fatih/vim-go', { 'tag': '\*' }

" Plugin options

Plug 'nsf/gocode', { 'tag': 'v.20150303', 'rtp': 'vim' }

" Plugin outside ~/.vim/plugged with post-update hook

Plug 'junegunn/fzf', { 'dir': '~/.fzf', 'do': './install --all' }

" Unmanaged plugin (manually installed and updated)

Plug '~/my-prototype-plugin'

" Initialize plugin system

call plug#end()

Reload .vimrc and :PlugInstall to install plugins