

## 一、准备工作

1、下载 Vim74 和适合 7.4 版本的 vim-gdb：

(1)[vim-7.4.tar.bz2](#)

(2)[vimgdb-for-7.4](#)

2、卸载已存在的 vim 以及相关组件

如果已经通过软件中心安装了 vim，则需要将其卸载

```
sudo apt-get remove vim vim-runtime gvim
```

可能还需要移除以下的包：

```
sudo apt-get remove vim-tiny vim-common vim-gui-common
```

3、安装依赖环境

手动编译安装 Vim74，首先先安装编译所需的依赖库和头文件。

```
sudo apt-get build-dep vim
```

然后再编译，如果依赖问题没有解决可能会出现这个提示：

```
no terminal library found
```

```
checking for tgetent()... configure: error: NOT FOUND!
```

```
You need to install a terminal library; for example ncurses.
```

```
Or specify the name of the library with --with-tlib.
```

也可手动安装以下的库等解决依赖，为了保险起见，最好还是执行一下。

```
sudo apt-get install libncurses5-dev libgnome2-dev libgnomeui-dev libgtk2.0-dev libatk1.0-dev  
libbonoboui2-dev libcairo2-dev libx11-dev libxpm-dev libxt-dev python-dev ruby-dev mercurial
```

## 二、不需要 vimgdb 的 Vim 安装

将下载的 vim74 源码包解压出来，进入解压出来的文件目录，编译

```
cd ~/downloads/vim74/
```

```
./configure --with-features=huge --enable-rubyinterp --enable-pythoninterp --with-python-config-  
dir=/usr/lib/python2.7/config-x86_64-linux/ --enable-perlinterp --enable-gui=gtk2 --enable-cscope  
--enable-luainterp --enable-multibyte --enable-xim --prefix=/usr
```

参数说明：

--with-features=huge：支持最大特性

--enable-rubyinterp：启用 Vim 对 ruby 编写的插件的支持

--enable-pythoninterp：启用 Vim 对 python 编写的插件的支持

--enable-luainterp：启用 Vim 对 lua 编写的插件的支持

--enable-perlinterp：启用 Vim 对 perl 编写的插件的支持

--enable-multibyte 和 --enable-xim：需要在 Vim 中输入中文，开启这两个特性

--enable-cscope: Vim 对 cscope 支持

--enable-gui=gtk2: gtk2 支持,也可以使用 gnome, 表示生成 gvim

--with-python-config-dir=/usr/lib/python2.7/config-x86\_64-linux-gnu/ 指定 python 路径, 这是 python2 在 [Ubuntu](#) 64 位机子上的路径, 32 位应该为 /usr/lib/python2.7/config-i386-linux-gnu/

--prefix=/usr: 设定编译安装路径, 注意自己是否有权限访问。

如果 configure 出错, 先用 make distclean 清除之前 configure 产生的文件再 configure。

最后 make 一下然后安装, 因为设定的路径 /usr 普通用户不具有写入权限, 所以需要以 sudo 来安装。

可以不指定 VIMRUNTIMEDIR, 则只要执行 make 就好。

```
make VIMRUNTIMEDIR=/usr/share/vim/vim74
```

```
sudo make install
```

如果想要卸载, sudo make uninstall 以下就好。

检验自己是否安装成功, 可以运行 vim --version 来看看, 可以检查某些特性是否已经开启。有加号的表示特性已开启。

### 三、带有 vimfdb 的 vim 安装

#### 1、安装 vim

将 vim74 的源码包和 vimfdb 的补丁包都解压到 ome 的 Downloads 下

?

1

2

3

4

5

```
: ~/Downloads$ ls
```

```
vim74
```

```
vim-7.4.tar.bz2
```

```
vimbdb-for-vim7.4-master
```

```
vimbdb-for-vim7.4-master.zip
```

然后切换到当前目录下, 打补丁

```
patch -p0 < vimbdb-for-vim7.4-master/vim74.patch
```

接下来切入到 vim74/src, make 安装。

```
cd vim74/src
make
sudo make install
```

默认的当前目录下 Makefile 设置的特性等都能满足一般需求，如果需要添加路径或做些修改等，只要对这个文件进行编辑就好。我将里面的路径改动了，因为默认的/opt/bin 这个路径不在我的\$PATH 里面，而默认却将 vim 的 bin 文件放在这里，导致执行 vim 会找不到命令。如果不作修改，可以将此路径添加到\$PATH 里面就好。

打开之，搜索/opt，按照注释的说明，将 BINDIR, MANDIR, DATADIR 这三行注释掉即可。改完之后的这三行如下：

```
# Uncomment the next line to install Vim in "/usr/bin"
#BINDIR = /opt/bin
# Uncomment the next line to install Vim manuals in "/usr/share/man/man1"
#MANDIR = /opt/share/man
# Uncomment the next line to install Vim help files in "/usr/share/vim"
#DATADIR = /opt/share
```

然后 make 安装。我在我的 ubuntu 上虽然安装成功，但是执行 vim 失败，出现 “Vim:Caught deadly signal ABRT”之类的错误。解决方法是在 make 是添加 CFLAGS="-O2 -D\_FORTIFY\_SOURCE=1"，则为：

```
make CFLAGS="-O2 -D_FORTIFY_SOURCE=1"
```

然后再执行 sudo make install，则最后将 vim 安装到/usr/local/share/vim/vim74 下。

如果想要通过执行 configure 设置一些特性和安装路径，那么记得加--enable-gdb，且是在 vim74/src 目录下执行，不是上面不添加 vimgdb 的直接在 vim74 目录下执行 configure

```
./configure --with-features=huge --enable-rubyinterp --enable-pythoninterp --with-python-config-dir=/usr/lib/python2.7/config-x86_64-linux/ --enable-perlinterp --enable-gui=gtk2 --enable-cscope --enable-luainterpreter --enable-multibyte --enable-xim --enable-gdb --prefix=/usr
```

再检查一下 Makefile 里面的 BINDIR, MANDIR, DATADIR，保证将它们三行注释了。然后 make, 添加 VIMRUNTIMEDIR，以及安装：

```
make VIMRUNTIMEDIR=/usr/share/vim/vim74 CFLAGS="-O2 -D_FORTIFY_SOURCE=1"
sudo make install
```

最后安装完了的 vim 的目录在/usr/share/vim/vim74 下。

运行 vim --version 检查以下 gdb 这一项特性前是否有加号，如果有，则表示 vimgdb 安装成功。

## 2、安装 vimgdb 运行环境

修改一下(cd ~/.vimrc 或 cd /home/用户名/.vimrc)主目录下的.vimrc（没有则创建一个）

加入

```
let g:vimgdb_debug_file = ""
```

```
run macros/gdb_mappings.vim
```

```
map <F2> :run macros/gdb_mappings.vim<CR>
```

这里说明一下，因为 vimgdb 的快捷键 F7 跟 c-support 有冲突，所以我改了一下，这样可以在需要的时候按 F2 载入 vimgdb 的 map，然后 F7 就可以正常工作了。

最后再修改下

```
sudo gedit /usr/share/vim/vim74/macros/gdb_mappings.vim 文件，
```

把最后一行 “call s:Toggle()” 注释掉，前面加一个英文的双引号即为注释。

将 vimgdb-for-vim7.4-master/vimgdb\_runtime 下的文件复制到运行时路径（runtime path）。可以在 vim 里面执行:set runtimepath?来找到 runtimepath。通过执行 echo \$VIMRUNTIME 来查看安装路径。可以看到有好几个路径，~/vim 是当前用户下的，/usr/share/vim/vim74 是所有用户的（这是设置了安装路径之后的，默认的路径应该在/usr/share/vim/vim74）。

然后切换到刚才选择将 vim runtime 路径，打开 vim，执行 “:helptags .”（注意，没有引号，helptags 后面有一个空格和点，点表示当前目录。），这样就能使用 “:help vimgdb”来打开 vimgdb 的帮助文档。

**更多 Vim 相关教程见以下内容：**

把 Vim 打成一个简单实用的 IDE <http://www.linuxidc.com/Linux/2011-06/37032.htm>

Vim 学习指南 <http://www.linuxidc.com/Linux/2013-08/89096.htm>

快速学会 Vi 编辑器 <http://www.linuxidc.com/Linux/2013-08/88586.htm>

强大的 Vim 编辑器 <http://www.linuxidc.com/Linux/2013-07/87544.htm>

在 CentOS 6.2 上搭建 Vim 开发环境 <http://www.linuxidc.com/Linux/2013-07/87363.htm>

CentOS 5.4 安装高亮 Vim 编辑工具 <http://www.linuxidc.com/Linux/2013-06/86508.htm>

Vim 技巧分享:C 语言设置 <http://www.linuxidc.com/Linux/2012-12/77124.htm>

Ubuntu 中设置 Vim 的行号 <http://www.linuxidc.com/Linux/2012-12/75485.htm>

Vim 编辑器使用基础教程 <http://www.linuxidc.com/Linux/2013-05/84031.htm>

更多 Ubuntu 相关信息见 [Ubuntu](http://www.linuxidc.com/topicnews.aspx?tid=2) 专题页面 <http://www.linuxidc.com/topicnews.aspx?tid=2>

本文永久更新链接地址: <http://www.linuxidc.com/Linux/2016-04/129774.htm>