

# Ubuntu常用软件的安装和备份

前面一节我们介绍了Ubuntu虚拟机的安装，今天就介绍一些Ubuntu常用软件的软装。大部分极其简单，敲个命令，输入下密码就能解决。少部分非常简单，设置下环境变量就可以解决。有些安装比较蛋疼的可以借助第三方平台比如Anaconda等来安装，也可以归纳到非常简单的行列中。所以，So easy!!!

## 1 第一件事：

打开终端（**CONTROL** + **ALT** + **T**），创建 `~/bin` 文件夹。

```
qli@bigbrosci:~$ mkdir ~/bin
```

这个文件夹用于存储一些日常的脚本文件，可执行的小程序。平时勤备份也不会占用太大空间。

## 2 命令安装程序

复制下面的 `sudo apt-get` 命令，输入到 `Terminal` 中，回车，输入密码即可。

```
sudo apt-get update
sudo apt-get install python3-pip
sudo apt-get install curl
curl https://bootstrap.pypa.io/get-pip.py --output get-pip.py
sudo python get-pip.py

sudo apt-get install vim
sudo apt-get install vim-gtk
sudo apt-get install sshfs
sudo apt-get install rename
sudo apt-get install xclip
sudo apt-get install git
sudo apt-get install gnuplot
sudo apt-get install xcrysden
sudo apt-get tree
sudo apt-get install inkscape

sudo apt-get install fish
sudo apt-get install tmux
sudo apt-get install texlive-full
sudo apt-get install kile
```

pip 用来安装python程序；

vim就不介绍了；

sshfs 用来把服务器挂载到本地电脑，方便传输数据；

rename 可以很方便批量命名；

xclip 可以把命令的输出复制到剪切板里面，方便鼠标粘贴；

git 用来从git-hub下载，上传，管理一些脚本程序等，

gnuplot 画图用的

xcrysden 可视化软件

tree 方便显示文件目录

inkscape 一款非常好用的做图软件

fish, tmux 装逼效果明显，没对象的重点关注下，好好学习一番，当然功能没的说，很强大。

Latex 写文章的可以使用最后两个命令安装。

其他的暂时也想不起来，如果你有推荐的，可以发邮件给我，后面再添上去。（[lqcata@gmail.com](mailto:lqcata@gmail.com)）

### 3 安装Anaconda

它可以帮助你节省很多时间和精力去安装软件。下载后，按照下面命令，然后一路狂摁回车或者输入 yes 即可，官网也有安装教程，这里就不累赘了。通过Anaconda安装其他软件，可以参考前面的教程：<https://www.bigbrosci.com/2020/09/19/A20/>

```
qli@bigbrosci:~/Downloads$ ls
Anaconda3-2020.07-Linux-x86_64.sh  p4vasp-0.3.30.tgz
qli@bigbrosci:~/Downloads$ bash Anaconda3-2020.07-Linux-x86_64.sh
Welcome to Anaconda3 2020.07

In order to continue the installation process, please review the license
agreement.
Please, press ENTER to continue

>>>
```

Anaconda安装的软件可以通过<https://anaconda.org> 来查询，都会列出来安装的命令，比如RDkit：

```
conda install -c rdkit rdkit
```

另一个需要注意的是，Anaconda安装后，最喜爱的 p4vasp 不能正常使用了。解决办法如下：

- 1) 找到p4v的目录，一般在 `/usr/bin/`
- 2) 将 `p4v` 中的 `python` 全部改为 `python2`
- 3) 将 `p4v.py` 第一行中的 `python` 改为 `python2`。

```
(base) qli@bigbrosci:~$ p4v
File "/usr/bin/p4v.py", line 9
    print """You need to get version 2.0 (or later) of PyGTK for this to work.
You
        can get source code from http://www.pygtk.org """
        ^
SyntaxError: invalid syntax
(base) qli@bigbrosci:~$ which p4v
/usr/bin/p4v
(base) qli@bigbrosci:~$ sudo vi /usr/bin/p4v
[sudo] password for qli:
(base) qli@bigbrosci:~$ sudo vi /usr/bin/p4v.py
(base) qli@bigbrosci:~$ grep python /usr/bin/p4v
export PYTHONPATH=$PYTHONPATH:/usr/lib/python2.7/dist-packages
exec python2 /usr/bin/p4v.py "$@"
(base) qli@bigbrosci:~$ grep python /usr/bin/p4v.py
#!/usr/bin/env python2
```

#### 4 其他可执行的程序

这里我们以VESTA为例，这种方法非常简单实用。

4.1) 下载解压VESTA程序包。（<https://jp-minerals.org/vesta/en/>） 直接通过官网下载即可，没必要去群里面求助，也没必要往群里面上传。

4.2) 本人习惯性地将其解压到 `/home/qli/Documents/VESTA-gtk3`

4.3) 设置环境变量，将下面的 `export` 两行复制到 `~/.bashrc` 文件中， `source` 下 `.bashrc` 文件，然后直接敲命令 `VESTA` 即可打开可视化界面。

```
qli@bigbrosci:~/Documents/VESTA-gtk3$ ls
asfcd      Library_License  RIETAN.sh      style      VESTA-core
bvparm2016.cif  libVESTA.so      spgra.dat      style.ini   VESTA_def.ini
elements    LICENSE          spgro.dat      template.ins VESTA-gui
elements.ini  mspgr.dat        STRUCTURE_TIDY tmp          VESTA.ini
img          PowderPlot       STRUCTURE_TIDY.sh VESTA      wyckoff.dat
qli@bigbrosci:~/Documents/VESTA-gtk3$ pwd
/home/qli/Documents/VESTA-gtk3
qli@bigbrosci:~/Documents/VESTA-gtk3$ cd
qli@bigbrosci:~$ vi .bashrc
export VESTA='/home/qli/Documents/VESTA-gtk3'
export PATH=$PATH:$VESTA
qli@bigbrosci:~$ source .bashrc
```

```
qli@bigbrosci:~$ VESTA
```

## 5 Ubuntu的备份

本人喜欢瞎折腾，也经常会把电脑搞崩溃，备份就显得尤其重要。对于Ubuntu来说，前面我们在虚拟机的安装中提到存储Ubuntu系统的 `vdi` 文件，如果你有足够大的硬盘，可以定期将一个稳定版本的 `vdi` 保存起来，万一系统崩溃，删掉坏的，导入最新的备份即可。

如果没有用虚拟机，直接装的Ubuntu系统，或者双系统，可以尝试定期保存 `~/bin` 和 `~/.bashrc` 文件，本人之前就是这样做的（现在懒得折腾，系统不坏了），它们不怎么占空间，打包压缩下，邮件基本就可以上传。如果系统坏掉了，非得安装一个新的才能解决的时候，只需要重复本教程，下载一些常用的软件，以及用备份的 `~/bin` 和 `~/.bashrc` 文件还原一下设置即可。数据一般来说都直接存储在服务器中，相关备份不在本教程讨论范围之内。

当然还有其他的办法备份和恢复虚拟机，比如创建当前Ubuntu版本的iso文件，定期备份当前整个系统以便引导修复等等，精力有限，不再累赘。有兴趣的可以尝试。

## 6 打赏

如果感觉本文对你的相关研究有帮助，欢迎打赏，支持作者的热心付出。如果你也有自己擅长的操作等，热烈欢迎无私分享，可以通过QQ（122103465）或者邮件（[lqcata@gmail.com](mailto:lqcata@gmail.com)）联系。



“感觉有帮助，请大师兄喝杯咖啡吧！”

BigBro-Q 的赞赏码