

环境: VirtualBox 6.0 + Ubuntu 16.04

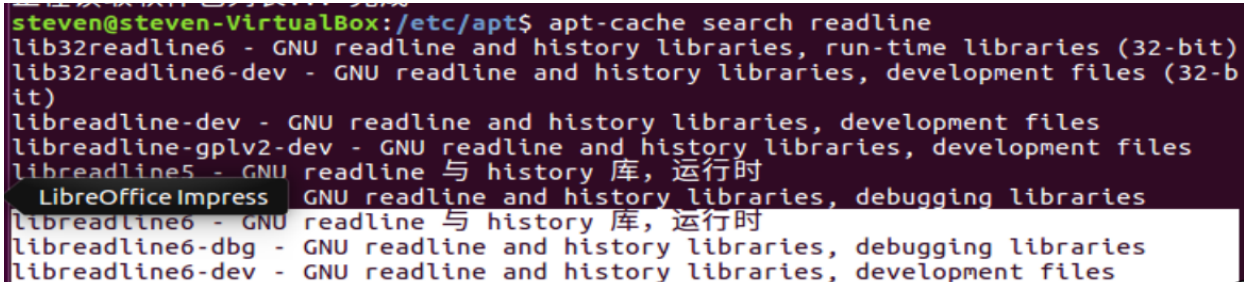
流程:

1.安装readline dtrace zlib

1.1 apt-get 安装

apt-cache search readline

查找是否有 libreadline6-dev、libreadline6 库

A terminal window showing the command 'apt-cache search readline' and its output. The output lists several packages related to readline, including lib32readline6, lib32readline6-dev, libreadline-dev, libreadline-gplv2-dev, libreadline5, LibreOfficeImpress, libreadline6, libreadline6-dbg, and libreadline6-dev. The package 'LibreOfficeImpress' is highlighted with a green box.

```
steven@steven-VirtualBox:/etc/apt$ apt-cache search readline
lib32readline6 - GNU readline and history libraries, run-time libraries (32-bit)
lib32readline6-dev - GNU readline and history libraries, development files (32-bit)
libreadline-dev - GNU readline and history libraries, development files
libreadline-gplv2-dev - GNU readline and history libraries, development files
libreadline5 - GNU readline 与 history 库, 运行时
LibreOfficeImpress - GNU readline and history libraries, debugging libraries
libreadline6 - GNU readline 与 history 库, 运行时
libreadline6-dbg - GNU readline and history libraries, debugging libraries
libreadline6-dev - GNU readline and history libraries, development files
```

1.2 apt-get 更新

如果1.1查找不到, 需要更新apt源

1.2.1备份默认的apt源

```
$ cd /etc/apt
```

```
$ sudo cp sources.list sources.list.backup
```

1.2.2 编辑 sources.list

```
deb http://archive.ubuntu.com/ubuntu/ raring main restricted universe multiverse
deb http://archive.ubuntu.com/ubuntu/ raring-security main restricted universe
multiverse
deb http://archive.ubuntu.com/ubuntu/ raring-updates main restricted universe
multiverse
deb http://archive.ubuntu.com/ubuntu/ raring-proposed main restricted universe
multiverse
deb http://archive.ubuntu.com/ubuntu/ raring-backports main restricted universe
multiverse
deb-src http://archive.ubuntu.com/ubuntu/ raring main restricted universe
multiverse
deb-src http://archive.ubuntu.com/ubuntu/ raring-security main restricted
universe multiverse
deb-src http://archive.ubuntu.com/ubuntu/ raring-updates main restricted
universe multiverse
deb-src http://archive.ubuntu.com/ubuntu/ raring-proposed main restricted
universe multiverse
deb-src http://archive.ubuntu.com/ubuntu/ raring-backports main restricted
universe multiverse
```

1.2.3 更新

```
sudo apt-get update
```

1.3 安装

```
sudo apt-get install libreadline6 libreadline6-dev systemtap-sdt-dev zlib1g-dev
```

2. 问题总结

2.1 端口冲突

之前安装过PG时会存在与默认5432端口冲突的情况，解决方式是换个端口号，建议15432
修改安装目录下 (/home/username/postgresql/data) 的配置文件，找到 #port = 5432
改成 port = 15432，注意去掉注释#

2.2 could not connect to server

```
server starting
steven@steven-VirtualBox:~/postgresql$ psql -d postgres
psql: could not connect to server: 没有那个文件或目录
        Is the server running locally and accepting
        connections on Unix domain socket "/tmp/.s.PGSQL.5432"?
steven@steven-VirtualBox:~/postgresql$ psql -d postgres -p 15432
psql (9.6.12)
Type "help" for help.

postgres=#
```

指定修改后的端口 psql -p 15432 连接

或者可能没有启动pg server 确认是否已经启动PG

```
server starting
steven@steven-VirtualBox:~/postgresql$ ps ux|grep postgres
steven    2031  0.0  0.7 177560 16208 pts/4    S   14:54   0:00 /home/steven/postgresql/bin/postgres -D /home/steven/postgresql/data
steven    2033  0.0  0.1 177560  2688 ?        Ss   14:54   0:00 postgres: check pointer process
steven    2034  0.0  0.1 177560  2688 ?        Ss   14:54   0:00 postgres: writer process
steven    2035  0.0  0.1 177560  2688 ?        Ss   14:54   0:00 postgres: wal writer process
steven    2036  0.0  0.2 177964  5372 ?        Ss   14:54   0:00 postgres: autovacuum launcher process
steven    2037  0.0  0.1  32576  2076 ?        Ss   14:54   0:00 postgres: stats collector process
steven    2039  0.0  0.0  21312   976 pts/4    S+   14:54   0:00 grep --color=auto postgres
```

2.3 useradd 与 adduser

建议直接使用adduser

adduser: 比较方便，直接使用命令：sudo adduser “用户名” 然后设置密码，一路回车就OK了。并且系统默认为新建的用户在/home目录下建立以用户名为名字的家目录，并且默认加入sudo用户组，自动设置shell。

useradd：使用不太方便，需要自己设置家目录，需要自己把用户加入sudo用户组，还需要自己设置shell等等，当希望把家目录放到别的硬盘下，这时候的useradd就显得很有用了

2.4 role does not exist

```
postgres=# \q
dhc@steven-VirtualBox:~$ psql -p 15432 -d postgres
psql: FATAL:  role "dhc" does not exist
dhc@steven-VirtualBox:~$ psql -p 15432 -d postgres -U steven
psql (9.6.12)
Type "help" for help.

postgres=#
```

原因，不是当前用户所创建的DB。 以-U属性 声明以DB的Owner角色登录¹¹

2.5 <value optimized out>

```
(gdb) info local
tuple = <value optimized out>
scandesc = <value optimized out>
estate = <value optimized out>
direction = <value optimized out>
slot = <value optimized out>
( " )
```

未关闭编译优化，关于编译优化的说明：

-O0：这个等级（字母“O”后面跟个零）关闭所有优化选项，也是CFLAGS或CXXFLAGS中没有设置-O等级时的默认等级。这样就不会优化代码，这通常不是我们想要的。

-O1：这是最基本的优化等级。编译器会在不花费太多编译时间的同时试图生成更快更小的代码。这些优化是非常基础的，但一般这些任务肯定能顺利完成。

-O2：-O1的进阶。这是推荐的优化等级，除非你有特殊的需求。-O2会比-O1启用多一些标记。设置了-O2后，编译器会试图提高代码性能而不会增大体积和大量占用的编译时间。

-O3：这是最高最危险的优化等级。用这个选项会延长编译代码的时间，并且在使用gcc4.x的系统里不应全局启用。自从3.x版本以来gcc的行为已经有了极大地改变。在3.x，-O3生成的代码也只是比-O2快一点点而已，而gcc4.x中还未必更快。用-O3来编译所有的软件包将产生更大体积更耗内存的二进制文件，大大增加编译失败的机会或不可预知的程序行为（包括错误）。这样做将得不偿失，记住过犹不及。在gcc 4.x中使用-O3是不推荐的。

-Os：这个等级用来优化代码尺寸。其中启用了-O2中不会增加磁盘空间占用的代码生成选项。这对于磁盘空间极其紧张或者CPU缓存较小的机器非常有用。但也可能产生些许问题，因此软件树中的大部分ebuild都过滤掉这个等级的优化。使用-Os是不推荐的。

正如前面所提到的，-O2是推荐的优化等级。如果编译软件出现错误，请先检查是否启用了-O3。再试试把CFLAGS和CXXFLAGS倒回到较低的等级，如-O1甚或-O0 -g2 -

gdb (用来报告错误和检查可能存在的问题) , 再重新编译。

-O0 不进行优化处理。

-O 或 -O1 优化生成代码。

-O2 进一步优化。

-O3 比 -O2 更进一步优化, 包括 inline 函数