在如今这个开源的环境里，想要开发某个功能，我们都会下意识的上网搜索有没有开源库，如果有开源库，那么好，下载下来给它编译好，使用。但是在使用过程中，你是否遇到不知如何将第三方库编译，链接到自己的工程中？怎么改makefile就是改不好？是否看到开源库lib/中pkgconfig文件夹，想都没想这是干什么用的，打开.pc文件也不知所云？那么好，今天我就总结下开源库中pkgconfig文件夹中.pc文件的作用，以及如何用pkg-config工具将开源库集成到自己的工程中去。

1、pkg-config工具的作用

pkg-config简单的说就是向用户提供相应库的路径，版本号，头文件路径等信息的综合调用程序。笔者使用的是Ubuntu系统，我们以OpenEXR库为例看看pkg-config运行的结果，在shell命令行输入：

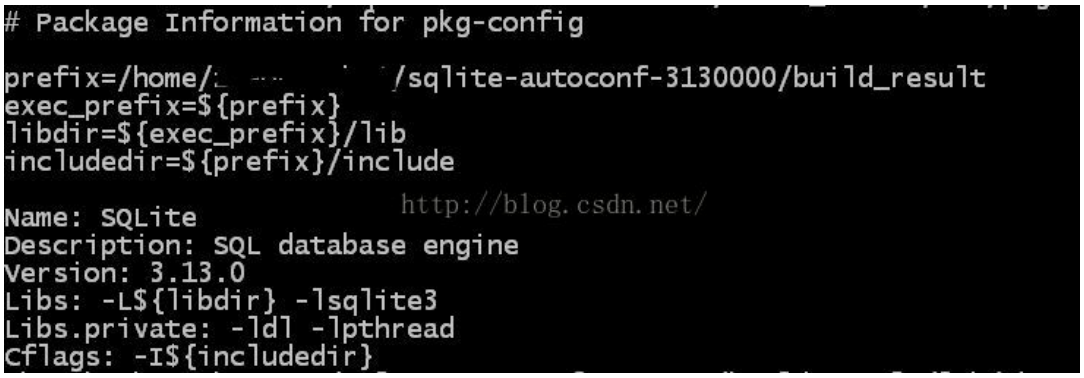
```
@ubuntu:~$ pkg-config  --libs --cflags OpenEXR
-I/usr/include/OpenEXR -lIlmImf -lImath -lHalf -lIex -lIexMath -lIlmThread -lpthread
@ubuntu:~$
```


显示信息为：**-I/usr/include/OpenEXR -lIlmImf -lImath -lHalf -lIex -lIexMath -lIlmThread -lpthread**
这是什么呀?
-I/usr/include/OpenEXR 这不就是我们用gcc编译时的CFLAGS参数吗?
-lIlmImf -lImath -lHalf -lIex -lIexMath -lIlmThread -lpthread这些不就是gcc在链接时使用的LDFLAGS参数吗?

因此当我们需要在自己的工程中编译链接时只需要合理的使用pkg-config工具，把上面那些参数加入到gcc的参数里即可，这个就是pkg-config工具的核心作用，它会检查你的库，产生相应信息，为你集成某个第三方库提供便利。

2、*.pc文件解析

我们知道，第三方库的使用主要涉及头文件的路径设置，库的路径设置以及动态库的环境变量设置。一般来讲，第三方库都会提供一个*.pc文件，pkg-config程序通过读取这个*.pc的文件，获取了库的头文件位置和库的路径等信息，然后告知编译器，实现库的自动使用。一般来说，*.pc文件的大体内容如下格式（以笔者最近使用的SQLite3为例）：



其中，

- prefix一般是指定库的默认安装路径
- exec_prefix一般是指库的另外指定的安装路径
- inludedir指定库的头文件路径
- libdir指定库的lib文件的路径
- Name指定库的名称，比如笔者使用的SQLite数据库
- Description表示库的描述
- Version是版本号
- Cflags是gcc链接头文件的指令,以-I紧接头文件路径设置
- Libs是gcc链接lib文件的指令, 是-L紧接lib文件路径，-l紧接所使用的lib的名字。

3、如何编译链接到你的工程?

这里讲的是工程，我只讲干货，编译单个程序例子网上有很多，就不赘述了。这里注意，使用pkg-config工具提取库的编译和链接参数需要有两个基本前提:
1)库本身按章的时候必须提供一个.pc文件。没有这个文件的说明库不支持pkg-config工具；
2)pkg-config必须要知道去哪找.pc文件；
对于支持pkg-config工具的库来说，库文件的搜索路径实际就是对.pc文件的搜索路径，一般系统的默认搜索路在/usr/lib/pkgconfig 中，库的头文件一般在/usr/include中。而个人使用的第三方库，不能每次编译后都装到/usr目录下吧。所以私有工程在编译链接第三方库时可以通过环境变量PKG_CONFIG_PATH来设置，pkg-config工具将按照设置路径的先后顺序进行搜索，直到找到指定的.pc文件为止。
所以在私有工程的makefile中，先修改环境变量：

export PKG_CONFIG_PATH=/home/水笙/sqlite-autoconf-3130000/build_result/lib/pkgconfig:$PKG_CONFIG_PATH

环境变量设置好后，设置CFLAGS：

CFLAGS += `pkg-config --cflags sqlite3`

这里注意要用``将命令包起来。

然后设置LDFLAGS:
LDFLAGS += `pkg-config --libs sqlite3`

基本通过这三步，工程就可以正确的编译链接第三方库了。

4、运行时指明共享库搜索路径

我们知道，库分为静态库和共享库。静态库.a就是一些.o文件的集合，编译链接后就集成到了你的应用程序中。而共享库，是在程序运行的时才被使用的，其搜索路径是在系统中预先设置的，对于处于搜索路劲之外的库，使用的时候必须设置好环境变量LD_LIBRARY_PATH，否则应用程序找不到，笔者将sqlite3库放到了应用程序文件夹的./lib中，在启动应用前调用下面这句：

export LD_LIBRARY_PATH="./lib"

笔者建议，最好将其写在你的启动脚本里。

完。