

2019年8月						
≤	一	二	三	四	五	六
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

昵称: [coding my life](#)
园龄: 5年4个月
粉丝: 22
关注: 0
[加关注](#)

搜索

找搜索

谷歌搜索

常用链接

[我的随笔](#)
[我的评论](#)
[我的参与](#)
[最新评论](#)
[我的标签](#)

我的标签

[lua\(6\)](#)
[linux\(4\)](#)
[qt creator\(3\)](#)
[信号\(3\)](#)
[debian\(3\)](#)
[ubuntu\(3\)](#)
[boost\(2\)](#)
[c++\(2\)](#)
[shell\(2\)](#)
[自定义\(2\)](#)
[更多](#)

随笔分类

[coding\(42\)](#)
[备忘](#)
[技术杂谈\(20\)](#)

随笔档案

[2019年7月\(1\)](#)
[2019年6月\(2\)](#)
[2019年4月\(1\)](#)
[2019年3月\(1\)](#)
[2019年1月\(1\)](#)
[2017年8月\(1\)](#)
[2017年7月\(1\)](#)
[2017年5月\(1\)](#)
[2016年7月\(1\)](#)
[2016年4月\(2\)](#)
[2016年3月\(1\)](#)
[2016年2月\(1\)](#)
[2016年1月\(1\)](#)
[2015年9月\(2\)](#)
[2015年7月\(3\)](#)
[2015年6月\(3\)](#)
[2015年4月\(3\)](#)
[2015年3月\(2\)](#)
[2015年2月\(1\)](#)
[2015年1月\(9\)](#)
[2014年12月\(4\)](#)
[2014年11月\(6\)](#)
[2014年10月\(4\)](#)
[2014年9月\(1\)](#)
[2014年6月\(1\)](#)

解决Qt5.7.0 cannot find -lGL

很久没用Qt了，这次要做一个协议编辑器，在ubuntu 14.04上安装了最新版本的Qt 5.7.0。界面改用扁平化风格，第一感觉还不错。按默认步骤创建了一个gui程序，编译运行，报了一个错：cannot find -lGL。

作为一个用惯了makefile的程序员，知道是少了库文件，而且名字应该为libGL.a或者libGL.so。到网上搜索一下，发现GL是OpenGL的缩写，显然是缺少OpenGL的开发库。安装命令为：sudo apt-get install libgl1-mesa-dev。在网上搜索关键字"cannot find -lGL"，得到的答案就是安装开发库。

OpenGL是一个巨大的库，而我只是做个编辑器而已，又不是什么2D、3D的程序，明显用不到OpenGL的。于是我到stackoverflow搜索了一下，找到了答案：<http://stackoverflow.com/questions/18406369/qt-cant-find-lgl-error>



you don't need to install anything. libGL is already installed with Ubuntu, you just need to soft link it. (works for ubuntu 14.x and 15.x)

1.First locate the GL library
2.Then link it under /usr/lib
3.If the library is missing, it can be installed via libgl1-mesa-dev package
Here is how you could do this:

```
$ locate libGL
/usr/lib/i386-linux-gnu/mesa/libGL.so.1
/usr/lib/i386-linux-gnu/mesa/libGL.so.1.2.0
/usr/lib/x86_64-linux-gnu/libGLEW.so.1.10
/usr/lib/x86_64-linux-gnu/libGLEW.so.1.10.0
/usr/lib/x86_64-linux-gnu/libGLEWmx.so.1.10
/usr/lib/x86_64-linux-gnu/libGLEWmx.so.1.10.0
/usr/lib/x86_64-linux-gnu/libGLU.so.1
/usr/lib/x86_64-linux-gnu/libGLU.so.1.3.1
/usr/lib/x86_64-linux-gnu/mesa/libGL.so.1
/usr/lib/x86_64-linux-gnu/mesa/libGL.so.1.2.0
/usr/lib/x86_64-linux-gnu/mesa-egl/libGLESv2.so.2
/usr/lib/x86_64-linux-gnu/mesa-egl/libGLESv2.so.2.0.0
$ sudo ln -s /usr/lib/x86_64-linux-gnu/mesa/libGL.so.1 /usr/lib/libGL.so
```



我的系统确实存在libGL.so.1.2.0这个文件。按他的方法解决了问题。如果你的系统确实没这个文件，那真是要安装了。

程序能跑起来了，不过并没有解决我的疑问：Qt的gui真的需要依赖OpenGL么？搜索官方的文档，找到了答案：<http://doc.qt.io/qt-5/linux-requirements.html>

在Qt for X11 Requirements的列表里，并没有OpenGL。在OpenGL Dependencies里，也指明在Qt使用OpenGL需要#include <QtOpenGL>并且需要在.pro项目配置文件里加上Qt += opengl。这说明我的程序根本不依赖OpenGL，决定手动测试一下。

清理项目，重新编译一次，然后切换到编译输出：

```
/opt/Qt5.7.0/5.7/gcc_64/bin/uc_...SPEditor/mainwindow.ui -o ui_mainwindow.h
g++ -c -pipe -g -std=gnu++11 -Wall -W -D_REENTRANT -fPIC -DQT_QML_DEBUG -DQT_WIDGETS_LIB -DQT_GUI_LIB -DQT_CORE_LIB -I./SPEditor -I./opt/Qt5.7.0/5.7/gcc_64/include -I/opt/Qt5.7.0/5.7/gcc_64/include/QtWidgets -I/opt/Qt5.7.0/5.7/gcc_64/include/QtGui -I/opt/Qt5.7.0/5.7/gcc_64/include/QtCore -I. -I/opt/Qt5.7.0/5.7/gcc_64/mkspecs/linux-g++ -o main.o ./SPEditor/main.cpp
g++ -c -pipe -g -std=gnu++11 -Wall -W -D_REENTRANT -fPIC -DQT_QML_DEBUG -DQT_WIDGETS_LIB -DQT_GUI_LIB -DQT_CORE_LIB -I./SPEditor -I./opt/Qt5.7.0/5.7/gcc_64/include -I/opt/Qt5.7.0/5.7/gcc_64/include/QtWidgets -I/opt/Qt5.7.0/5.7/gcc_64/include/QtGui -I/opt/Qt5.7.0/5.7/gcc_64/include/QtCore -I. -I/opt/Qt5.7.0/5.7/gcc_64/mkspecs/linux-g++ -o mainwindow.o ./SPEditor/mainwindow.cpp
/opt/Qt5.7.0/5.7/gcc_64/bin/moc -DQT_QML_DEBUG -DQT_WIDGETS_LIB -DQT_GUI_LIB -DQT_CORE_LIB -I/opt/Qt5.7.0/5.7/gcc_64/mkspecs/linux-g++ -I/home/xzc/Documents/code/SPEditor -I/opt/Qt5.7.0/5.7/gcc_64/include -I/opt/Qt5.7.0/5.7/gcc_64/include/QtWidgets -I/opt/Qt5.7.0/5.7/gcc_64/include/QtGui -I/opt/Qt5.7.0/5.7/gcc_64/include/QtCore -I. -I/usr/include/c++/4.8 -I/usr/include/x86_64-linux-gnu/c++/4.8 -I/usr/include/c++/4.8/backward -I/usr/lib/gcc/x86_64-linux-gnu/4.8/include -I/usr/local/include -I/usr/lib/gcc/x86_64-linux-gnu/4.8/include-fixed -I/usr/include/x86_64-linux-gnu -I/usr/include -I./SPEditor/mainwindow.h -o moc_mainwindow.cpp
g++ -c -pipe -g -std=gnu++11 -Wall -W -D_REENTRANT -fPIC -DQT_QML_DEBUG -DQT_WIDGETS_LIB -DQT_GUI_LIB -DQT_CORE_LIB -I./SPEditor -I./opt/Qt5.7.0/5.7/gcc_64/include -I/opt/Qt5.7.0/5.7/gcc_64/include/QtWidgets -I/opt/Qt5.7.0/5.7/gcc_64/include/QtGui -I/opt/Qt5.7.0/5.7/gcc_64/include/QtCore -I. -I/opt/Qt5.7.0/5.7/gcc_64/mkspecs/linux-g++ -o moc_mainwindow.o moc_mainwindow.cpp
g++ -Wl,-rpath,/opt/Qt5.7.0/5.7/gcc_64/lib -o SPEditor main.o mainwindow.o moc_mainwindow.o -L/opt/Qt5.7.0/5.7/gcc_64/lib -lQt5Widgets -L/usr/lib64 -lQt5Gui -lQt5Core -lGL -lpthread
20:44:50: 进程 "/usr/bin/make" 正常退出。
20:44:50: Elapsed time: 00:08.
```

1 问题 2 Search Results 3 应用程序输出 4 编译输出 5 Debugger Console

可以看到编译的参数，我手动进入到对应的目录，然后改一下编译参数：

```
xzc@xzc-HP-ProBook-4446s:~/Documents/code/build-SPEditor-Desktop_Qt_5_7_0_GCC_64bit-Debug$ rm SPEditor
xzc@xzc-HP-ProBook-4446s:~/Documents/code/build-SPEditor-Desktop_Qt_5_7_0_GCC_64bit-Debug$ g++ -Wl,-rpath,/opt/Qt5.7.0/5.7/gcc_64/lib -o SPEditor main.o mainwindow.o moc_mainwindow.o -L/opt/Qt5.7.0/5.7/gcc_64/lib -lQt5Widgets -L/usr/lib64 -lQt5Gui -lQt5Core -lpthread
xzc@xzc-HP-ProBook-4446s:~/Documents/code/build-SPEditor-Desktop_Qt_5_7_0_GCC_64bit-Debug$ ./SPEditor
xzc@xzc-HP-ProBook-4446s:~/Documents/code/build-SPEditor-Desktop_Qt_5_7_0_GCC_64bit-Debug$
```

不使用-lGL参数编译链接的程序，也可以正常跑，验证了我的想法。

既然不依赖OpenGL，为什么要链接OpenGL呢？首先在链接参数里加额外-lGL对生成的程序并没有影响，你的程序里没有调用OpenGL的方法，不会有任何额外的代码链接进去。而Qt是一个跨平台的库，里面包含一些3D的库。要Qt Creator非常智能地根据程序生成依赖，这个有点难，也没必要。所以我记得作者干脆把这些常用的库都加到链接参数里，一了百了。

这些预先定好的参数，按平台放在安装目录的mkspecs目录的，比如我的就在opt/Qt5.7.0/5.7/gcc_64/mkspecs。存细看链接参数，你会发现

-l/opt/Qt5.7.0/5.7/gcc_64/mkspecs/linux-g++-要这样的路径，这是Qt的跨平台配置。我的机子上，在

2014年5月(6)
2014年4月(5)

最新评论

1. [Re:解决Qt5.7.0 cannot find -lGL](#)

@ 白夜猩我在5.11.2环境下，修改qt_lib_gui_private.pri文件，成功了...

--chwit

2. [Re:解决Qt5.7.0 cannot find -lGL](#)

为什么我的 Qt5.8.0 ubuntu 下改了还是没有用，

--白夜猩

3. [Re:Protocol Buffers与FlatBuffers效率对比](#)

你确定json 要比二进制快。感觉json比二进制快 完全不符合直觉和常理啊。因为像FlatBuffer二进制完全无须 转换啊

--SoarNo1

4. [Re:重写boost内存池](#)

虽然时间已经很久远了，但我的测试结果是 boost.pool 的时间是0，new 版本的时间是6~7.

--开学五年级了

5. [Re:Protocol Buffers与FlatBuffers效率对比](#)

还可以。

--我不清楚是谁

阅读排行榜

1. [更改debian的软件源sources.list\(21973\)](#)

2. [解决Qt5.7.0 cannot find -lGL\(19146\)](#)

3. [sem_timedwait的用法\(12398\)](#)

4. [MongoDB的数据类型\(10491\)](#)

5. [打造自己个性的notepad ++\(9765\)](#)

评论排行榜

1. [ubuntu14.04折腾迅雷xware\(6\)](#)

2. [解决Qt5.7.0 cannot find -lGL\(4\)](#)

3. [使用ssh远程执行命令批量导出数据到本地\(2\)](#)

4. [Protocol Buffers与FlatBuffers效率对比\(2\)](#)

5. [重写boost内存池\(2\)](#)

推荐排行榜

1. [ubuntu14.04折腾迅雷xware\(6\)](#)

2. [解决Qt5.7.0 cannot find -lGL\(3\)](#)

3. [xmind教程\(2\)](#)

4. [利用pyinstaller将python脚本打包发布\(2\)](#)

5. [打造自己个性的notepad ++\(1\)](#)

/opt/Qt5.7.0/5.7/gcc_64/mkspecs/common/linux.conf可以找到这样的配置

```
#
# qmake configuration for common linux
#

QMAKE_PLATFORM      += linux

include(unix.conf)

QMAKE_CFLAGS_THREAD += -D_REENTRANT
QMAKE_CXXFLAGS_THREAD += $$QMAKE_CFLAGS_THREAD
QMAKE_LFLAGS_GCSECTIONS = -Wl,--gc-sections

QMAKE_LFLAGS_REL_RPATH = -Wl,-z,origin
QMAKE_REL_RPATH_BASE   = $ORIGIN

QMAKE_INCDIR          =
QMAKE_LIBDIR           =
QMAKE_INCDIR_X11       =
QMAKE_LIBDIR_X11       =
QMAKE_INCDIR_OPENGL   =
QMAKE_LIBDIR_OPENGL   =
QMAKE_INCDIR_OPENGL_ES2 = $$QMAKE_INCDIR_OPENGL
QMAKE_LIBDIR_OPENGL_ES2 = $$QMAKE_LIBDIR_OPENGL
QMAKE_INCDIR_EGL       =
QMAKE_LIBDIR_EGL       =
QMAKE_INCDIR_OPENVG   =
QMAKE_LIBDIR_OPENVG   =

QMAKE_LIBS             =
QMAKE_LIBS_DYNLOAD     = -ldl
QMAKE_LIBS_X11         = -lXext -lX11 -lm
QMAKE_LIBS_NIS         = -lnsl
QMAKE_LIBS_EGL         = -lEGL
QMAKE_LIBS_OPENGL      = -lGL
QMAKE_LIBS_OPENGL_ES2  = -lGLESv2
QMAKE_LIBS_OPENVG      = -lOpenVG
QMAKE_LIBS_THREAD      = -lpthread
QMAKE_LIBS_LIBDEV      = -ludev

QMAKE_CFLAGS_WAYLAND   =
QMAKE_INCDIR_WAYLAND   =
QMAKE_LIBS_WAYLAND_CLIENT = -lwayland-client
QMAKE_LIBS_WAYLAND_SERVER = -lwayland-server
QMAKE_LIBDIR_WAYLAND   =
QMAKE_DEFINES_WAYLAND  =
QMAKE_WAYLAND_SCANNER  = wayland-scanner

QMAKE_CFLAGS_XCB       =
QMAKE_LIBS_XCB         =
QMAKE_DEFINES_XCB      =

QMAKE_AR               = ar cqs
QMAKE_OBJCOPY          = objcopy
QMAKE_NM               = nm -P
QMAKE_RANLIB           =

QMAKE_STRIP            = strip
QMAKE_STRIPFLAGS_LIB  += --strip-unneeded
```

如果你的程序确实用不着OpenGL，并且以后也不打算写这样的程序，把QMAKE_LIBS_OPENGL = -lGL这个去掉就可以了。

分类: [coding](#)

标签: [Qt](#), [OpenGL](#), [cannot find -lGL](#)



 [coding my life](#)
[关注 - 0](#)
[粉丝 - 22](#)

[+加关注](#)

» 上一篇: [对内存对齐的进一步理解](#)

» 下一篇: [ubuntu apt-get 使用shadowsocks代理](#)

3 0
[推荐](#) [反对](#)

posted on 2016-07-16 21:20 [coding my life](#) 阅读(19146) 评论(4) [编辑](#) [收藏](#)

FeedBack:

#1楼 2016-09-02 18:18 [谭迅](#)

非常感谢...解决了我的问题!---谢谢--

支持(0) 反对(0)

#2楼 2017-05-21 11:32 [\[0\]](#)

经过再次验证，Qt5.8.0 ubuntu 下依然管用。博主也说的很清楚。赞一个。

支持(0) 反对(0)

#3楼 2018-10-19 16:10 [slide](#)

为什么我的 Qt5.8.0 ubuntu 下改了还是没有用，

支持(0) 反对(0)


#4楼 2019-01-16 20:54 [chwit](#)

@ 白夜猩

我在5.11.2环境下，修改qt_lib_gui_private.pri文件，成功了

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

 注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#) 网站首页。

[【推荐】超50万C++/C#源码: 大型实时仿真组态图形源码](#)

[【推荐】零基础轻松玩转云上产品，获赠礼加返百元大礼](#)

[【推荐】华为IoT平台开发者套餐9.9元起，购买即送免费课程](#)



相关博文:

- [解决Qt下ssl出错的办法](#)
- [qt.network.ssl: QSslSocket: cannot call unresolved function SSLv23_client_method](#)
- [基于Linux（中标麒麟）上QT的环境搭建](#)
- [解决Qt5.7.0cannotfind-IGL](#)
- [Linux下QT、cannotfind-IGL、](#)

华为云微认证 限时 **5折 26元起**

最新 IT 新闻:

- [宇宙的命运取决于什么？暗能量的性质！](#)
- [ofo这一年的艰难自救](#)
- [科学家发现脑损伤会导致大量的神经元产生，不过.....](#)
- [从被唱衰到真香，共享充电宝是怎么逆风翻盘的？](#)
- [AMD CEO苏姿丰：显卡交火已经不是关注的重点](#)
- » [更多新闻...](#)

Copyright © 2019 coding my life

Powered by .NET Core 3.0 Preview 8 on Linux Powered by: [博客园](#) 模板提供: [沪江博客](#)