<https://www.cnblogs.com/lidabo/p/6206504.html>

一

在应用程序需要连接外部库的情况下，linux默认对库的连接是使用动态库，在找不到动态库的情况下再选择静态库。使用方式为：

gcc test.cpp -L. -ltestlib

如果当前目录有两个库libtestlib.so libtestlib.a 则肯定是连接libtestlib.so。如果要指定为连接静态库则使用：

gcc test.cpp -L. -static -ltestlib

使用静态库进行连接。

当对动态库与静态库混合连接的时候，使用-static会导致所有的库都使用静态连接的方式。这时需要作用-Wl的方式：

gcc test.cpp -L. -Wl,-Bstatic -ltestlib  -Wl,-Bdynamic -ltestdll

另外还要注意系统的运行库使用动态连接的方式，所以当动态库在静态库前面连接时，必须在命令行最后使用动态连接的命令才能正常连接

，如：

gcc test.cpp -L. -Wl,-Bdynamic -ltestdll -Wl,-Bstatic -ltestlib  -Wl,-Bdynamic

最后的-Wl,-Bdynamic表示将缺省库链接模式恢复成动态链接。

二：查看静态库导出函数

注意：参数信息只能存在于 .h 头文件中  
windows下  
dumpbin   /exports   libxxx.a  
linux 下  
nm   -g   --defined-only   libxxx.a

三

场景是这样的。我在写一个Nginx模块，该模块使用了[MySQL](http://lib.csdn.net/base/14" \o "undefined" \t "_blank)的C客户端接口库libmysqlclient，当然mysqlclient还引用了其他的库，比如libm, libz, libcrypto等等。对于使用mysqlclient的代码来说，需要关心的只是mysqlclient引用到的动态库。大部分情况下，不是每台机器都安装有libmysqlclient，所以我想把这个库静态链接到Nginx模块中，但又不想把mysqlclient引用的其他库也静态的链接进来。  
　　我们知道gcc的-static选项可以使链接器执行静态链接。但简单地使用-static显得有些’暴力’，因为他会把命令行中-static后面的所有-l指明的库都静态链接，更主要的是，有些库可能并没有提供静态库（.a），而只提供了动态库（.so）。这样的话，使用-static就会造成链接错误。  
　　  
之前的链接选项大致是这样的，

|  |  |
| --- | --- |
| 1 | CORE\_LIBS="$CORE\_LIBS -L/usr/lib64/mysql -lmysqlclient -lz -lcrypt -lnsl -lm -L/usr/lib64 -lssl -lcrypto" |

修改过是这样的，

|  |  |
| --- | --- |
| 12 | CORE\_LIBS="$CORE\_LIBS -L/usr/lib64/mysql -Wl,-Bstatic -lmysqlclient**\** -Wl,-Bdynamic -lz -lcrypt -lnsl -lm -L/usr/lib64 -lssl -lcrypto" |

　　其中用到的两个选项：-Wl,-Bstatic和-Wl,-Bdynamic。这两个选项是gcc的特殊选项，它会将选项的参数传递给链接器，作为链接器的选项。比如-Wl,-Bstatic告诉链接器使用-Bstatic选项，该选项是告诉链接器，对接下来的-l选项使用静态链接；-Wl,-Bdynamic就是告诉链接器对接下来的-l选项使用动态链接。下面是man gcc对-Wl,option的描述，

-Wl,option Pass option as an option to the linker. If option contains commas, it is split into multiple options at the commas. You can use this syntax to pass an argument to the option. For example, -Wl,-Map,output.map passes -Map output.map to the linker. When using the GNU linker, you can also get the same effect with -Wl,-Map=output.map.

下面是man ld分别对-Bstatic和-Bdynamic的描述，

-Bdynamic -dy -call\_shared Link against dynamic libraries. You may use this option multiple times on the command line: it affects library searching for -l options which follow it. -Bstatic -dn -non\_shared -static Do not link against shared libraries. You may use this option multiple times on the command line: it affects library searching for -l options which follow it. This option also implies --unresolved-symbols=report-all. This option can be used with -shared. Doing so means that a shared library is being created but that all of the library's external references must be resolved by pulling in entries from static libraries

http://blog.csdn.net/lapal/article/details/5482277

http://blog.chinaunix.net/uid-20737871-id-3083925.html