**[编译时如何使用库](http://www.cnblogs.com/york-hust/archive/2012/06/08/2541300.html)**

　　-l参数就是用来指定程序要链接的库，-l参数紧接着就是库名，那么库名跟真正的库文件名有什么关系呢？就拿数学库来说，他的库名是m，他的库文件名是libm.so，很容易看出，把库文件名的头lib和尾.so去掉就是库名了。

　　好了现在我们知道怎么得到库名了，比如我们自已要用到一个第三方提供的库名字叫libtest.so，那么我们只要把libtest.so拷贝到/usr/lib里，编译时加上-ltest参数，我们就能用上libtest.so库了（当然要用libtest.so库里的函数，我们还需要与libtest.so配套的头文件）。放在/lib和/usr/lib和/usr/local/lib里的库直接用-l参数就能链接了，但如果库文件没放在这三个目录里，而是放在其他目录里，这时我们只用-l参数的话，链接还是会出错，出错信息大概是：“/usr/bin/ld: cannot find -lxxx”，也就是链接程序ld在那3个目录里找不到libxxx.so，

　　这时另外一个参数-L就派上用场了，比如常用的X11的库，它放在/usr/X11R6/lib目录下，我们编译时就要用-L/usr/X11R6/lib -lX11参数，-L参数跟着的是库文件所在的目录名。再比如我们把libtest.so放在/aaa/bbb/ccc目录下，那链接参数就是-L/aaa/bbb/ccc -ltest

　　另外，大部分libxxxx.so只是一个链接，以RH9为例，比如libm.so它链接到/lib/libm.so.x，/lib/libm.so.6又链接到/lib/libm-2.3.2.so，如果没有这样的链接，还是会出错，因为ld只会找libxxxx.so，所以如果你要用到xxxx库，而只有libxxxx.so.x或者libxxxx-x.x.x.so，做一个链接就可以了ln -s libxxxx-x.x.x.so libxxxx.so

　　手工来写链接参数总是很麻烦的，还好很多库开发包提供了生成链接参数的程序，名字一般叫xxxx-config，一般放在/usr/bin目录下，比如gtk1.2的链接参数生成程序是gtk-config，执行gtk-config --libs就能得到以下输出"-L/usr/lib -L/usr/X11R6/lib -lgtk -lgdk -rdynamic -lgmodule -lglib -ldl -lXi -lXext -lX11 -lm"，这就是编译一个gtk1.2程序所需的gtk链接参数，xxx-config除了--libs参数外还有一个参数是--cflags用来生成头文件包含目录的，也就是-I参数，在下面我们将会讲到。

　　你可以试试执行gtk-config --libs --cflags，看看输出结果。现在的问题就是怎样用这些输出结果了，最笨的方法就是复制粘贴或者照抄，聪明的办法是在编译命令行里加入这个`xxxx-config --libs --cflags`，比如编译一个gtk程序：gcc gtktest.c `gtk-config --libs --cflags`这样就差不多了。注意`不是单引号，而是1键左边那个键。除了xxx-config以外，现在新的开发包一般都用pkg-config来生成链接参数，使用方法跟xxx-config类似，但xxx-config是针对特定的开发包，但pkg-config包含很多开发包的链接参数的生成，用pkg-config --list-all命令可以列出所支持的所有开发包，pkg-config的用法就是pkg-config pagName --libs --cflags，其中pagName是包名，是pkg-config--list-all里列出名单中的一个，比如gtk1.2的名字就是gtk+，pkg-config gtk+ --libs --cflags的作用跟gtk-config --libs --cflags是一样的。比如：gcc gtktest.c `pkg-config gtk+ --libs --cflags`。