<https://blog.csdn.net/u014132659/article/details/52252298>

版权声明：本文为博主原创文章，未经博主允许不得转载。 https://blog.csdn.net/u014132659/article/details/52252298

+++原理+++  
linux调用库的方式有三种：  
1、静态链接库  
2、动态链接库  
3、动态加载库  
       动态链接库（共享库）和动态加载库文件没有任何区别，唯一区别是：动态链接库是程序运行一开始就要加载库，动态加载库是程序在运行到函数需要这个函数的实现的时候才加载库。  
       所以动态加载库就需要一组函数来控制什么时候加载库。这些函数是dlopen() dlerror() dlsym() dlclose()  
示例代码：  
int (\*add) (int a, int b);   //声明函数指针，用来接收dlsym返回来的函数，就是接收要用到的函数  
void \*phandle = dlopen("libtest.so", RTLD\_LAZY);  
add = dlsym(phandle, "add");   //参数add是库里定义的add函数，dlsym返回函数地址  
printf("%d\n", add(2, 5));  
编译的时候：  
gcc  main.c  -o  main  -ldl   //要加上参数-ldl。dl库是dlopen等那三个打开库使用的函数。这时候再编译自己的程序的时候就不用像动态链接库那样-L./ -ltest，链接自己的test库了，因为程序中需要这个库的时候就打开这个库使用了。 

上面第2 和第3 就是动态库的两种使用方式：（1）编译的时候链接动态库 （2）运行的时候加载动态库  
我们调用动态链接库有两种方法：一种是编译的时候，指明所依赖的动态链接库，这样loader可以在程序启动的时候，来所有的动态链接映射到内存中；一种是在运行过程中，通过dlopen和dlfree的方式加载动态链接库，动态将动态链接库加载到内存中。  
这两种方式，从编程角度来讲，第一种是最方便的，效率上影响也不大，在内存使用上有些差别。  
第一种方式，一个库的代码，只要运行过一次，便会占用物理内存，之后即使再也不使用，也会占用物理内存，直到进程的终止。  
第二中方式，库代码占用的内存，可以通过dlfree的方式，释放掉，返回给物理内存。  
这个差别主要对于那些寿命很长，但又会偶尔调用各种库的进程有关。如果是这类进程，建议采用第二种方式调用动态链接库

linux gcc 生成链接库  
1、生成动态链接库  
+++原理+++  
参考文章：<http://download.csdn.net/detail/u014132659/9673886>是一个上传的文档  
动态链接库是程序运行是加载的库。  
  
       动态链接库是目标文件的集合， 目标文件在动态链接库中的组织方式是按照特殊方式形成的，库中函数和变量的地址是相对的，不是绝对地址，其真实地址在调用动态库的程序加载时形成的。  
  
       安装库文件就是把库文件放在系统目录下，能让程序找见就可以。系统默认的目录是/lib和/usr/lib或者还有其他的。也可以不放在系统默认路径下，自己在链接库配置文件中配置一个路径。  
  
      链接库的配置文件，即配置程序运行时查找动态库的路径，配置文件是 /etc/ld.so.conf。系统文件 /lib 和 /usr/lib 是默认的查找的目录，不用配置。其余的自己添加的路径可以配置进去。  
  
       动态链接库管理命令ldconfig，运行命令会生成一个在动态库的别名，就是命令里写的 -soname 的名字。重点是这个命令会生成 /etc/ld.so.cache 文件，这个是运行程序时查找动态库的时候需要的。具体这个文件是写了什么东西，还不清楚。  
  
参考文章：<http://www.cnblogs.com/xuxm2007/archive/2010/08/10/1796254.html>  
动态库配置文件 和 LD\_LIBRARY\_PATH 都是解决程序在运行是查找加载动态库问题的。在编译程序的时候，链接过程还是需要加 -L 和 -l参数，指定动态库的具体位置，链接器需要。（一定不要混淆）。一个是编译是查找动态库，具体是链接阶段，一个是运行是加载动态库。  
  
---命令---  具体示例看上面上传的文档  
gcc -Wall -fpic -o libtest.o -c test.c  //-fpic是预处理结果，可以自己man看一下，不用也可以  
gcc -shared -Wl,-soname,libtest.so -o libtest.so.0 libtest.o  //-shared参数，-soname生成别名用的，真实名字是.so.0，有版本号  
gcc -o main main.c -L./ -ltest  
nm lictest.so  //查看库提供的函数，T标记  
2、生成静态链接库  
+++原理+++  
生成静态链接库需要用ar程序  
---命令---  
gcc -c test.c    //生成test.o  
ar -rcs libtest.a test.o   //生成静态库  
nm libtest.a   //查看库提供的函数  
gcc -o main main.c libtest.a   //使用静态库  
或者  
gcc -o main main.c -L./ -static -ltest

+++原理+++  
编译时需要链接动态库，在链接阶段。  
对于静态链接库，编译的时候用-L 和 -l参数， -static参数明确指定库；  
对于动态链接库，编译的时候使用 -L 和 -l参数明确指定库；  
对于动态加载库，编译的时候使用 -ldl 参数。  
运行时需要动态库，因为要使用库中定义的函数。在运行的时候指定动态库的位置，有几种方法：1、配置LD\_LIBRARY\_PATH环境变量 2、把库直接放在系统路径下，路径是/lib 和 /usr/lib或者其他系统路径，不清楚了 3、配置文件 /etc/ld.so.conf 还有其他方式，不清楚了。  
两种情况下使用动态库一定要搞清楚。