**静态库和动态库**

**一、如何生成静态库和动态库**

源码经过编译生成对象文件，对象文件进一步处理可以生成可执行程序或者库文件。

对于可执行程序和动态库，此“进一步处理”指的是“链接”，使用gcc命令。

对于静态库，此“进一步处理”指的是“打包”，使用ar命令。

**1. 使用gcc生成可执行程序举例：**

gcc -c \*.c //将\*.c文件编译成同名的.o文件。-c参数表示仅编译

gcc \*.o -o exe //将\*.o链接成可执行程序exe。通过-o指定生成对象的名称

**2 使用ar生成静态库举例：**

gcc -c \*.c //将\*.c文件编译成同名的.o文件。

ar rcs lib\*.a \*.o //将\*.o文件打包成静态库。ar是打包工具，需要参数rcs，参数具体含义可查man手册。linux中静态库使用lib前缀，.a后缀

**3. 使用gcc –shared生成动态库举例：**

gcc -c \*.o -fPIC //将\*.c文件编译成同名的\*.o文件，注意此处需要有-fPIC参数。表示生成位置无关代码，更利于动态库的共享。

gcc -shared \*.o -o lib\*.so //将\*.o打包成动态库。-shared表示生成动态库。linux中动态库使用lib前缀，.so后缀

**二、库的用法：**

**用法1：通过库的绝对路径或相对路径引用。**

gcc test.c ./lib\*.a //链接当前目录下的静态库lib\*.a

或

gcc test.c ./lib\*.so //链接当前目录下的动态库lib\*.so

gcc test.c /mylib/lib\*.a //链接/mylib目录下的静态库lib\*.a

或

gcc test.c /mylib/lib\*.so //链接/mylib目录下的动态库lib\*.so

**用法2：将库文件拷贝到系统库目录/usr/lib中，通过"-l+库的名字"引用**

cp lib\*.so lib\*.a /usr/lib

gcc test.c -lwyj //去掉前缀lib和后缀.a或.so之后，就是库的名字，-l参数之后只需要库名字。

注意：

若系统库中同时存在同名的静态库和动态库，默认链接动态库，

当只有静态库时，才链接静态库。

在gcc链接时，使用-static参数，可指定链接系统库中库文件时，只链接静态库。若没有静态库只有动态库，则也返回链接失败。

**三、代码示例：**

有如下3个文件：fun1.c fun2.c test.c

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*fun1.c\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

void fun()

{

printf("====This is static library.\n");

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*fun2.c\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

void fun()

{

printf("====This is shared library.\n");

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*test.c\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void fun(void);

int main()

{

fun();

return 0;

}

gcc -c fun1.c //将fun1.c编译成对象文件fun1.o

gcc -c fun2.c -fPIC //将fun2.c编译成位置无关的对象文件fun2.o

ar rcs libtemp.a fun1.o //将对象文件fun1.o打包成命名为libtemp.a静态库

gcc -shared fun2.o -o libtemp.so //将对象文件fun2.o链接成命名为libtemp.so动态库

/\*\*\*\*这时，我们的静态库libtemp.a和动态库libtemp.so就生成了\*\*/

gcc test.c ./libtemp.a -o mystatic //链接静态库生成可执行程序mystatic

gcc test.c ./libtemp.so -o myshared //链接动态库生成可执行程序myshared

/\*\*\*因为libtemp.a是静态库，编译时就已经链接到了mystatic，所以删除掉libtemp.a之后，mystatic依然可以执行。

\*\*\*而libtemp.so是动态库，是在运行时需要定位的，所以删除掉libtemp.so，甚至当前目录换了之后，myshared就无法执行。

\*\*\*\*\*/

使用file查看mystaic，会发现mystatic依然是“动态可执行文件”，因为在mystatic的时候，还用到了系统库中的libc库，系统库目录中同时存在libc.so和libc.a，所以默认链接动态库。

因为连接到了动态库，所以生成的依然是动态可执行文件。

使用前面讲到的知识。连接时增加-static参数，指定全部使用静态库。再次生成的mystatic2就不是动态可执行文件了。

gcc test.c ./libtemp.a -o mystatic2 -static

还可以将libtemp.a和libtem.so拷贝到系统库目录/usr/lib中。

这样就可以使用下面的命令生成mystatic和myshared了

gcc test.c -ltemp -o myshared //若/usr/lib中存在libtemp.so

gcc test.c -ltemp -o mystatic //若库中只存在libtemp.a

gcc -static test.c -ltemp -o mystatic2 //要求库中存在所有需要链接的库的静态库版本。