

动态链接库的编译与使用

1.Linux 下的动态链接库

● 动态链接库的基本概念

动态链接库（英语：Dynamic-link library，缩写为 DLL）是微软公司在微软视窗操作系统中实现共享函数库概念的一种实现方式。这些库函数的扩展名是.DLL、.OCX（包含 ActiveX 控制的库）或者.DRV（旧式的系统驱动程序）。

所谓动态链接，就是把一些经常会共用的代码（静态链接的 OBJ 程序库）制作成 DLL 档，当可执行文件调用到 DLL 档内的函数时，Windows 操作系统才会把 DLL 档加载内存内，DLL 档本身的结构就是可执行文件，当程序有需求时函数才进行链接。通过动态链接方式，内存浪费的情形将可大幅降低。静态链接库则是直接链接到可执行文件。

● 动态链接库的好处

- 节省内存和减少交换操作。很多进程可以同时使用一个 DLL，在内存中共享该 DLL 的一个副本。相反，对于每个用静态链接库生成的应用程序，Windows 必须在内存中加载库代码的一个副本。
- 节省磁盘空间。许多应用程序可在磁盘上共享 DLL 的一个副本。相反，每个用静态链接库生成的应用程序均具有作为单独的副本链接到其可执行图像中的库代码。
- 升级到 DLL 更为容易。当 DLL 中的函数发生更改时，只要函数的参数和返回值没有更改，就不需重新编译或重新链接使用它们的应用程序。相反，静态链接的对象代码要求在函数更改时重新链接应用程序。

● 动态链接库的使用实例

实例:用 vs2015 等开发工具建立工程时,可以在应用程序设置中的应用程序类型设置 DLL 表示建立动态链接库,之后可以在 DLL 项目中添加导出函数以便在编程时在附加的动态库中调用。

2. 按要求写出下列几种常用情况的动态链接库的测试样例

很多步骤和之前 makefile 中类似 相同的在此不再赘述
建立子目录和.c 文件在此略过
主要说下 makefile 文件

2.2

makefile

```
root@RHEL-zby:/home/1551265-000104/01
test2:libtest1.so test1.o
    cc -o test2 test2.c -L. -ltest1 -Wl,-rpath .
libtest1.so:test1.c
    cc -c -fpic test1.c
    cc -shared -fpic -o libtest1.so test1.o
clean:
    rm test2 *.o *.so
~
```

```
[root@RHEL-zby 01]# make
cc -c -fpic test1.c
cc -shared -fpic -o libtest1.so test1.o
cc -o test2 test2.c -L. -ltest1 -Wl,-rpath .
[root@RHEL-zby 01]# ./test2
1551265张伯阳 [root@RHEL-zby 01]#
```

之后我改了 test1.c 文件 改了其输出 并使其重新输出.so 文件覆盖之前的文件
运行 test2 可执行文件达到了预期效果

```
[root@localhost 01]# vi test1.c
[root@localhost 01]# gcc -c -fpic test1.c
[root@localhost 01]# gcc -shared -fpic -o test1.so test1.o
[root@localhost 01]# ./test2
155张伯阳 [root@localhost 01]#
```

2.3

将上一题的 makefile 文件中的 gcc 改为 g++ .c 改成.cpp 即可

```
root@RHEL-zby:/home/1551265-000104/02 (ssh)
test2:libtest1.so test1.o
    g++ -g -Wall -o test2 test2.cpp -L. -ltest1 -Wl,-rpath .
libtest1.so:test1.cpp
    g++ -c -fpic test1.cpp
    g++ -shared -fpic -o libtest1.so test1.o
clean:
    rm test2 *.o *.so
~
~
~
```

运行可执行文件 test2 达到预期效果

```
[root@RHEL-zby 02]# make
g++ -c -fpic test1.cpp
g++ -shared -fpic -o libtest1.so test1.o
g++ -g -Wall -o test2 test2.cpp -L. -ltest1 -Wl,-rpath .
[root@RHEL-zby 02]# ./test2
1551265张伯阳 [root@RHEL-zby 02]#
```

修改 test1.cpp 的学号

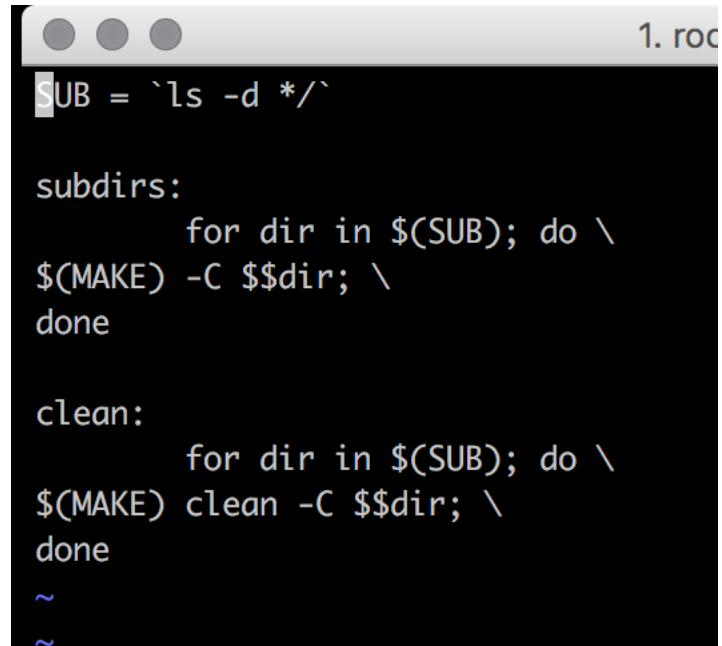
重新编译并修改动态库文件

运行 test2 达到预期效果

```
1551265张伯阳 [root@localhost 02]# vi test1.cpp
[root@localhost 02]# g++ -c -fpic test1.cpp
[root@localhost 02]# g++ -shared -fpic -o test1.so test1.o
[root@localhost 02]# ./test2
155张伯阳 [root@localhost 02]#
```

2.4

还是之前的总 makefile 文件



```
SUB = `ls -d */`

subdirs:
    for dir in $(SUB); do \
$(MAKE) -C $$dir; \
done

clean:
    for dir in $(SUB); do \
$(MAKE) clean -C $$dir; \
done
~
~
```