

查阅相关资料，动手写测试程序，基于 CentOS 7，给出下列知识点的回答：

1 Linux 下的动态链接库（下述问题只是引导你发现差异，不代表你组织文章的顺序）

- 动态链接库的基本概念
- 动态链接库的好处
- 动态链接库的编译方法
- 动态链接库的使用实例

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

2.1 每个人的目录结构要求如下（假设学号为 1651234，各人按实修改）：首先建立“学号-000105”子目录（可位于任意子目录下），下面再建立若干空的子目录，示例如下：

```
1651234-000105
|-- 01
|-- 02
```

2.2 在子目录 01 下建立两个源程序文件，分别是 test1.c/test2.c（分别打印自己的学号及姓名），并写出满足要求的 makefile 文件

<pre>//test1.c #include <stdio.h> int fun() { printf();//打印你的学号 return 0; }</pre>	<pre>//test2.c #include <stdio.h> int fun(); int main() { fun(); printf();//打印你的姓名 return 0; }</pre>
---	--

- 执行 make 后即可将 test1.c 编译为动态链接库文件 libtest1.so，放在本目录下，而 test2.c 编译形成的可执行文件 test2 中会调用 libtest1.so 中的 fun() 函数
- 编译器**必须**选用 gcc
- 在可执行文件 test2 生成后，如果换用其它同学的同名 libtest1.so（移动到你的目录下），则 test2 执行时会打印其它同学的序号和你的姓名（不准再次生成 test2）
- 执行 make clean 后即可将.o 及可执行文件都清除掉，仅保留.c 源程序

2.3 在子目录 02 下建立两个源程序文件，分别是 test1.cpp/test2.cpp（分别打印自己的学号及姓名），并写出满足要求的 makefile 文件

<pre>//test1.cpp #include <iostream> using namespace std; int fun() { cout //打印你的学号 return 0; }</pre>	<pre>//test2.cpp #include <iostream> using namespace std; int fun(); int main() { fun(); cout //打印你的姓名 return 0; }</pre>
---	--

- 执行 make 后即可将 test1.cpp 编译为动态链接库文件 libtest1.so，放在本目录下，而 test2.cpp 编译形成的可执行文件 test2 中会调用 libtest1.so 中的 fun() 函数

- 编译器**必须**选用 c++/g++
- 在可执行文件 test2 生成后，如果换用其它同学的同名 libtest1.so（移动到你的目录下），则 test2 执行时会打印其它同学的序号和你的姓名（不准再次生成 test2）
- 执行 make clean 后即可将.o 及可执行文件都清除掉，仅保留.c 源程序

2.4 在 1651234-000105 目录下写一个满足下列要求的 makefile 文件

- 执行 make 后，依次调用 01/02 目录下 makefile，分别编译出可执行文件/动态链接库文件等，放在各子目录下
- 执行 make clean 后即可将.o 及可执行文件都清除掉，仅保留.c 源程序
- 本小题的 makefile，要考虑到子目录数量不定，每个子目录的名称不定，但每次 make 后仍能编译所有含 makefile 文件的子目录（测试时将 01/02 子目录换名，提交作业时仍换回 01/02 的名称）

【本次作业的统一批改方法说明：】

1、 首先建立 16-000105 目录（可位于任意目录下）

2、 本次作业，每位同学上交一个 linux-so.tar.bz2 文件，截止时间到后，会从每人的交作业目录中复制出来，全部放在 16-000105 目录中

示例如下：

```
16-000105
|-- 1651234-linux-so.tar.bz2      (第 1 位同学的作业压缩包)
...
`-- 1654321-linux-so.tar.bz2     (最后 1 位同学的作业压缩包)
```

3、 进入到 16-000105 目录下，用 tar -xvjf 1651234-linux-so.tar.bz2 解压每个文件，**要求得到一个“学号-000105”子目录**，下面再包含了各个小题的子目录

示例如下（为了简化，**未显示**所有的 tar.bz2 文件）：

```
16-000105
|-- 1651234-000105 (第 1 位同学解压得到的子目录)
|   |-- 01          (第 2.2 小题对应的子目录)
|   |   |-- *.c      (第 2.2 小题的若干源程序文件)
|   |   `-- makefile (第 2.2 小题对应的 makefile 文件)
|   |...
|   |-- 02          (第 2.3 小题对应的子目录)
|   |   |-- *.cpp    (第 2.3 小题的若干源程序文件)
|   |   `-- makefile (第 2.3 小题对应的 makefile 文件)
|   `-- makefile     (第 2.4 小题对应的 makefile 文件)
.....
|-- 1654321-000105 (最后 1 位同学解压得到的子目录)
|   |...
```

```
`-- check.sh      (老师事先建好的 shell 文件，预备编译所有同学的本次作业，具体的实现方式是进入到每个学号对应的目录后调用该目录下的总 makefile)
```

4、 进入 16-000105 目录，进行一次 ./check.sh，就能检查完所有作业

5、 **无法顺利编译则不能得分，对应学号及子目录名错则不能得分**

6、 **作业提交时清除所有的中间文件及生成的可执行文件、源程序备份文件等**

【作业要求:】

- 1、**10月10日前**网上提交
- 2、每题所占平时成绩的具体分值见网页
- 3、超过截止时间提交作业则不得分