

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 下的动态编译（下述问题只是引导你发现差异，不代表你组织文章的顺序）
 - 什么叫动态编译？
 - 给出 `printf("hello, world");` 程序的 gcc 动态编译命令，可执行文件字节数是多少？
 - 给出 `cout << "hello, world";` 程序的 c++/g++ 动态编译命令，可执行文件字节数是多少？
 - 给出第一周作业中 `mysql_demo.cpp` 的动态编译命令，可执行文件字节数是多少？
 - 如何查找某个可执行文件所依赖的动态链接库？
- 2 Linux 下的 gcc 静态编译（下述问题只是引导你发现差异，不代表你组织文章的顺序）
 - 什么叫静态编译？
 - 给出 `printf("hello, world");` 程序的 gcc 静态编译命令，可执行文件字节数是多少？
- 3 Linux 下的 c++/g++ 静态编译（下述问题只是引导你发现差异，不代表你组织文章的顺序）
 - 按第一周要求安装完成的 CentOS 7 系统是不能静态编译 cpp 程序的，需要再安装相关包，给出相应的安装步骤
 - 给出 `cout << "hello, world";` 程序的 c++/g++ 静态编译命令，可执行文件字节数是多少？
- 4 按要求写出下列几种常用情况的静态编译测试样例
 - 4.1 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-000104”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：

1651234-000104

|-- 01

|-- 02

- 4.2 在子目录 01 下建立 `test.c`（打印自己的学号及姓名），并写出满足要求的 `makefile` 文件

```
//test.c
#include <stdio.h>
int main()
{
    printf();//打印你的学号+姓名
    return 0;
}
```

- 执行 `make` 后即可将 `test.c` 静态编译后形成一个可执行文件 `test`
- 编译器**必须**选用 `gcc`
- 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
- 如果不是从 CentOS 7 的原始 iso 中取得的安装包，则文档中写清楚所用静态库的版本号及安装方法

- 4.3 在子目录 02 下建立 `test.cpp`（打印自己的学号及姓名），并写出满足要求的 `makefile` 文件

```
//test.cpp
#include <iostream>
using namespace std;

int main()
{
    cout << ;//打印你的学号+姓名
}
```

```
return 0;
}
```

- 执行 make 后即可将 test.cpp 静态编译后形成一个可执行文件 test
- 编译器**必须**选用 c++/g++
- 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
- **如果不是从 CentOS 7 的原始 iso 中取得的安装包，则文档中写清楚所用静态库的版本号及安装方法**

4.4 在 1651234-000104 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 16-000104 目录（可位于任意目录下）

2、 本次作业，每位同学上交一个 linux-static_compile.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 16-000104 目录中

示例如下：

```
16-000104
|-- 1651234-linux-static_compile.tar.bz2      （第 1 位同学的作业压缩包）
...
`-- 1654321-linux-static_compile.tar.bz2      （最后 1 位同学的作业压缩包）
```

3、 进入到 16-000104 目录下，用 tar -xvjf 1651234-linux-static_compile.tar.bz2 解压每个文件，**得到**一个“学号-000104”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）：

```
16-000104
|-- 1651234-000104 （第 1 位同学解压得到的子目录）
|   |-- 01          （第 4.2 小题对应的子目录）
|   |   |-- *.c      （第 4.2 小题的若干源程序文件）
|   |   `-- makefile （第 4.2 小题对应的 makefile 文件）
|   |...
|   |-- 02          （第 4.3 小题对应的子目录）
|   |   |-- *.cpp    （第 4.3 小题的若干源程序文件）
|   |   `-- makefile （第 4.3 小题对应的 makefile 文件）
|   `-- makefile     （第 4.4 小题对应的 makefile 文件）
.....
|-- 1654321-000104 （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 16-000104 目录，进行一次 ./check.sh，就能检查完所有作业（只识别特定学号）

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

【额外加分作业:】

实现第一次作业中测试程序 `mysql_demo.cpp` 的静态编译的, 能获得额外加分 1 分

提交要求: 1、首先建立 1651234-990101 目录 (可位于任意目录下)

- 2、此目录下包含 `mysql_demo.cpp` 及相应 `makefile` 文件, 编译后可执行文件名为 `mysql_demo`
- 3、如果有为了完成本题而需要额外补充的组件, 给出安装方法 (转换为 PDF) 文件, 放在相同目录下即可
- 4、检查方法为: 用 `make` 进行编译, 然后将生成的可执行文件复制到**未安装 MySQL 开发组件**的机器上, 能运行即为通过
- 5、如果有为了完成本题而需要额外补充的组件, 给出安装方法 (转换为 PDF) 文件, 放在相同目录下即可
- 6、如果不是从 CentOS 7 的原始 iso 中取得的安装包, 则文档中写清楚所用静态库的版本号及安装方法
- 7、提交时间同正常作业