注意：本文档讲述的不是独立使用 Linux C++ 编译C++程序的方法，而是将已经使用VS2019编写并调试完成的C++程序再用Linux C++ 编译一遍

★ 用多个编译器完成同一个作业时，希望共用一个源程序文件，目的是避免同一个作业维护多个源程序文件所带来的冲突及错误

★ 单独使用Linux C++编写C++程序的方法请自行摸索

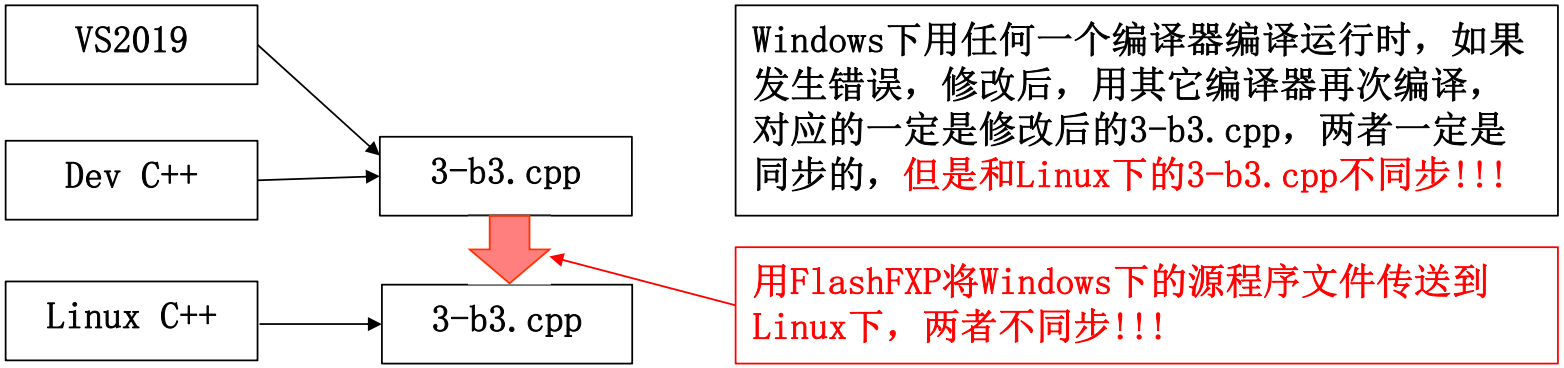
★ 为了统一，强制要求每个作业首先用VS2019完成，在调试通过的基础上用Dev C++及Linux C++再次编译，从而体验同一程序在不同编译器中可能出现的差异

★ 用多个编译器完成同一个作业时，希望共用一个源程序文件，目的是避免同一个作业维护多个源程序文件所带来的冲突及错误

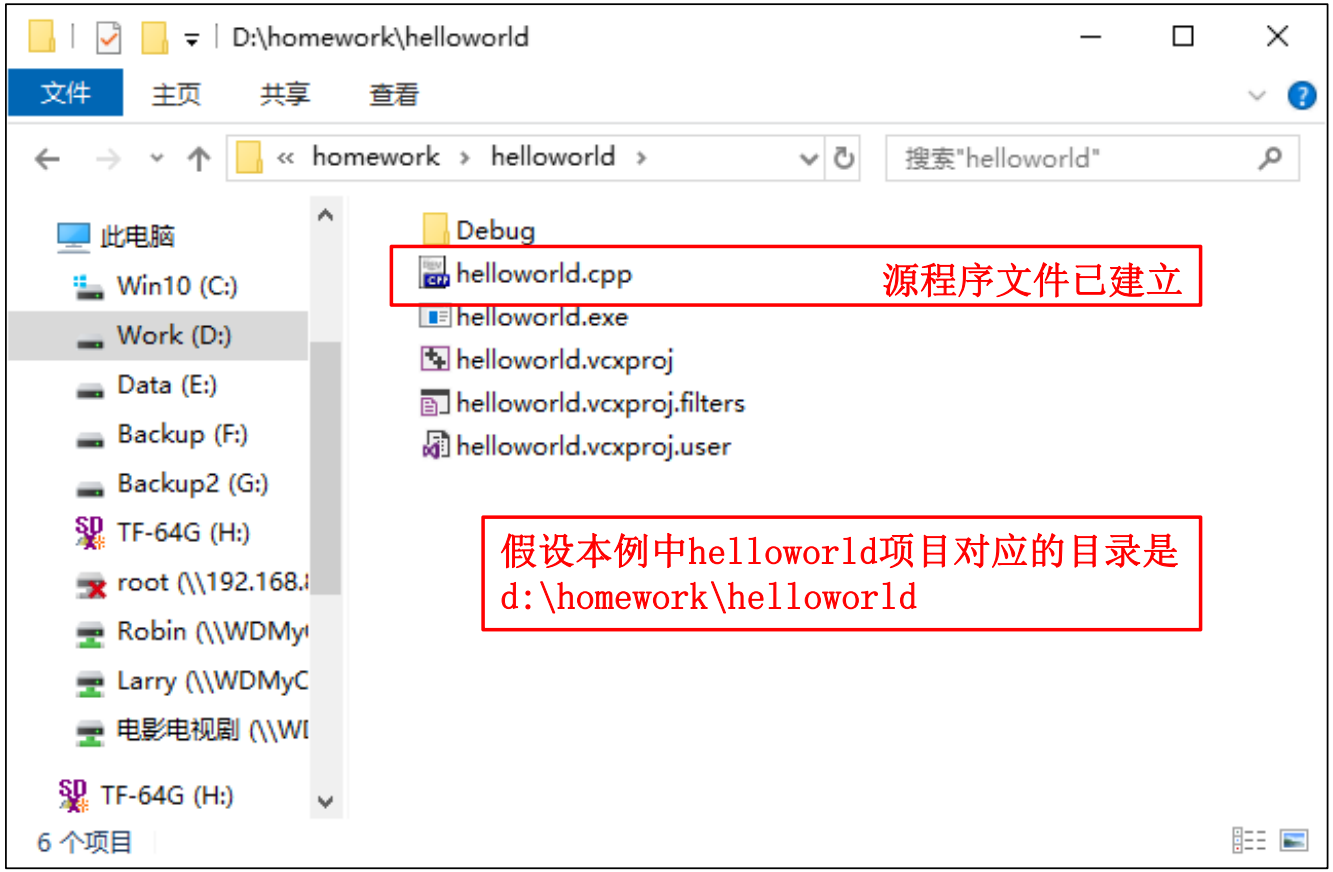
例：完成作业3-b3（对应源代码为 3-b3.cpp，要求能同时适应三个编译器）

前提：Windows下的两个编译器对应同一个 3-b3.cpp，已验证完毕，现需要进行Linux下的验证

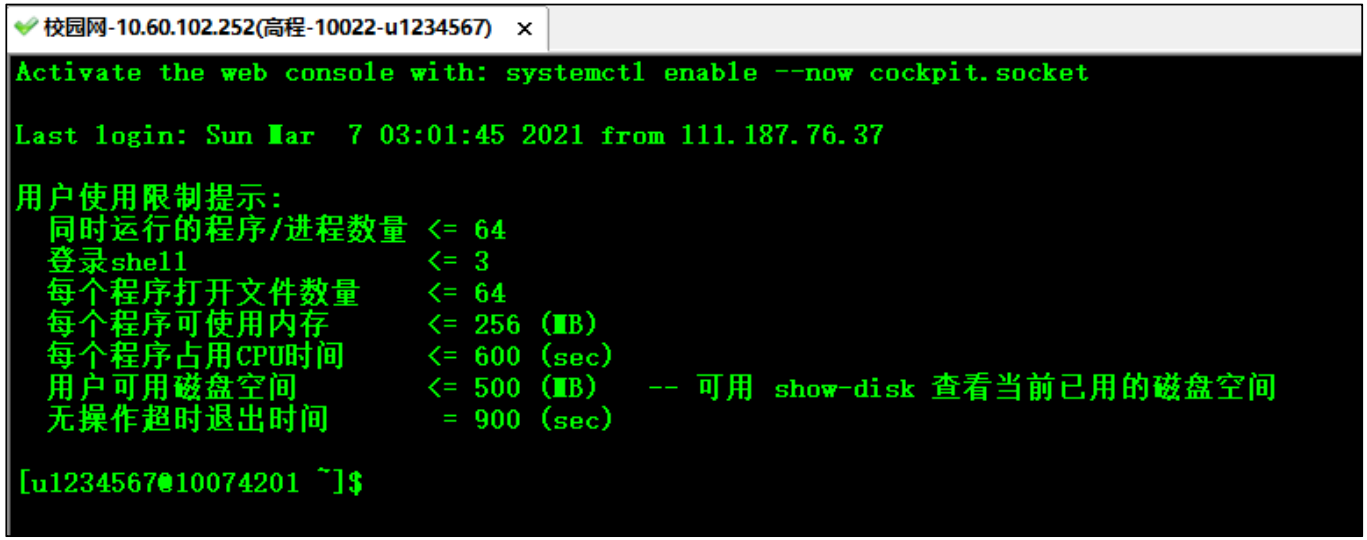
方法：用FlashFXP将Windows已验证好的 3-b3.cpp 上传到Linux服务器上，如果出现错误，则在Windows下修改，用 VS2019 + DevC++ 编译器验证正确后，再次上传至Linux服务器验证



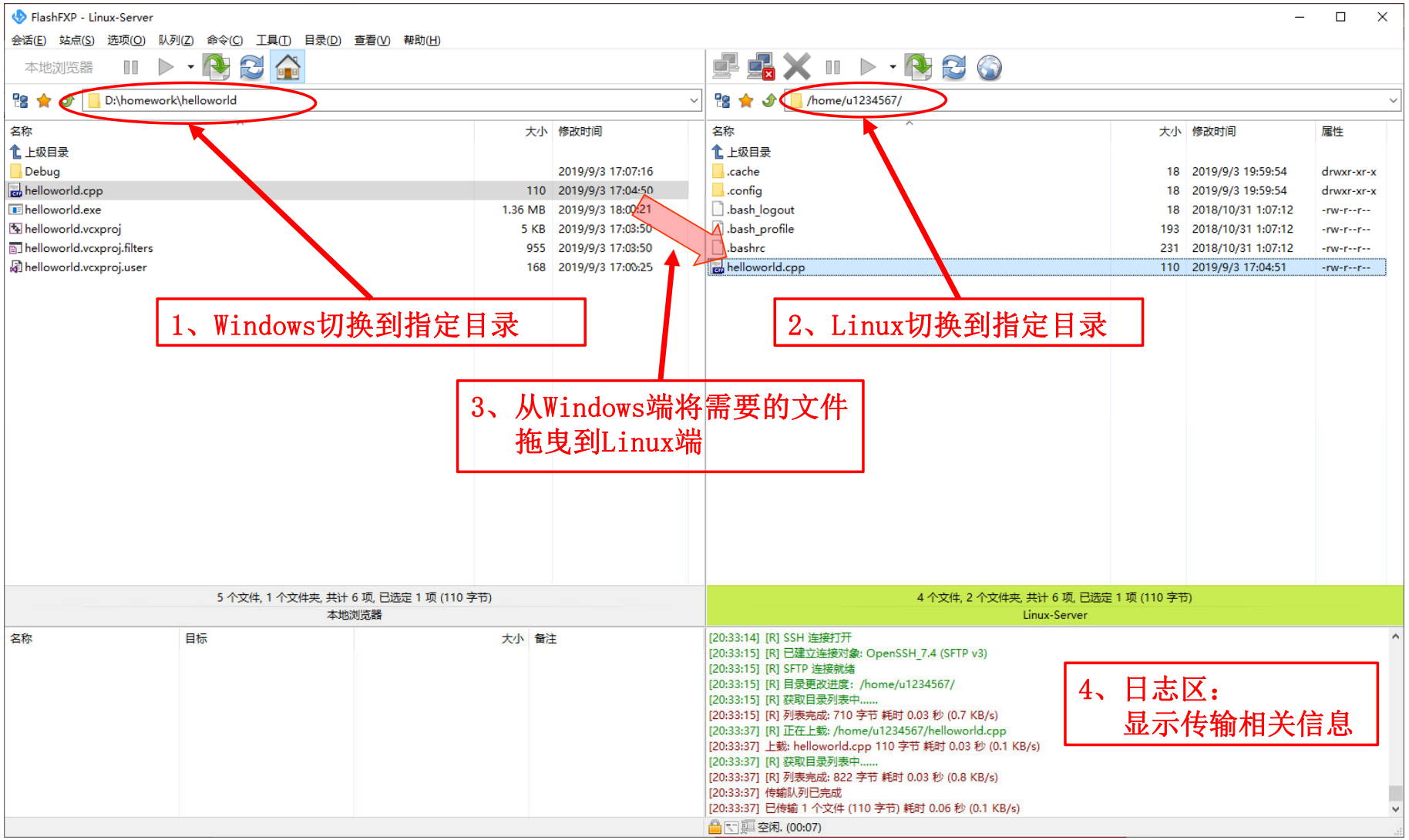
第1步：假设在VS2019中已经建立了一个"helloworld"项目，对应的 helloworld.cpp 已经在VS2019下建立，并在 VS2019 和 Dev C++ 中已验证通过



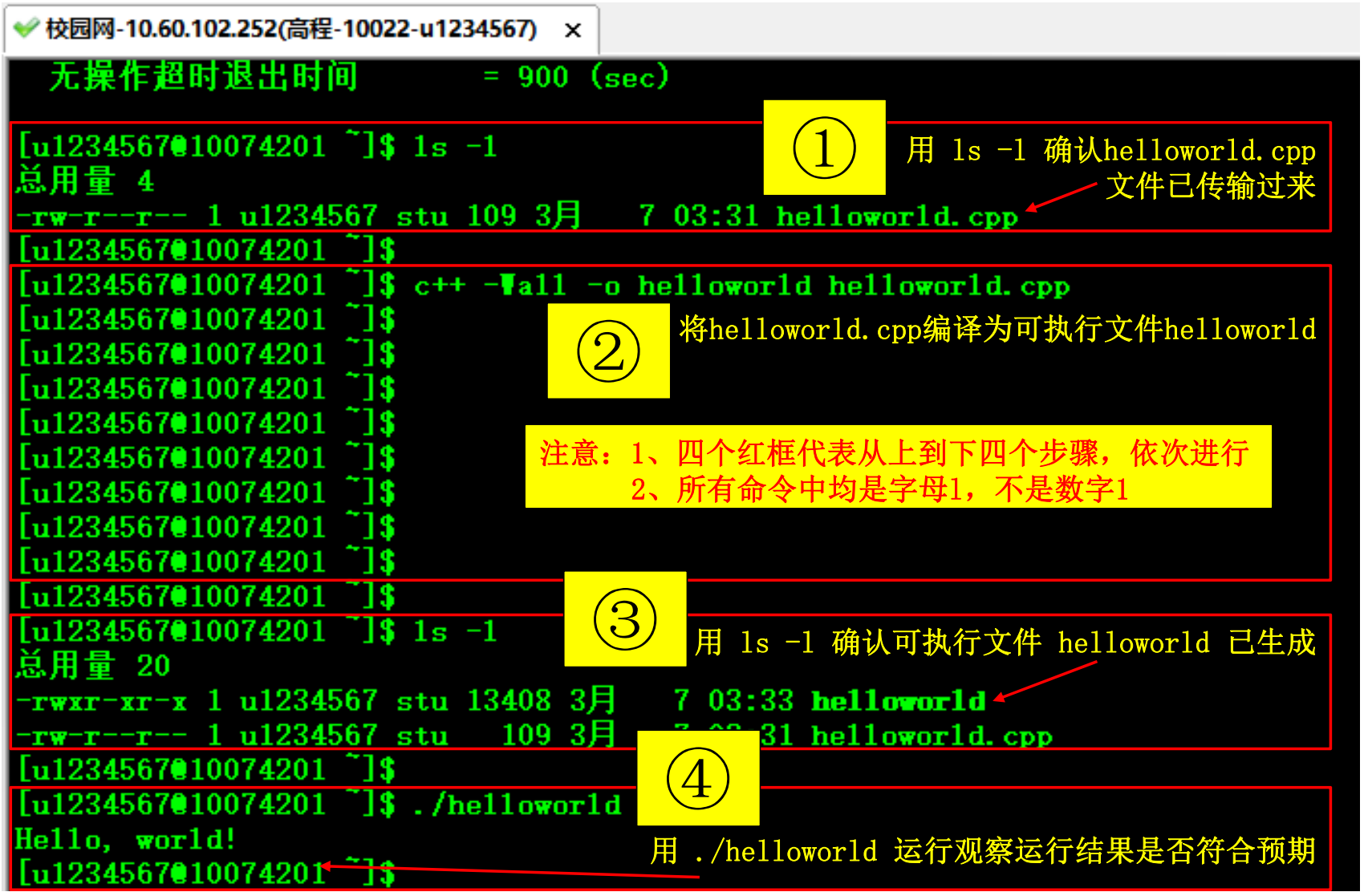
第2步：用SecureCRT远程登录Linux服务器



第3步：启动FlashFXP，将 d:\homework\helloworld 文件夹下的 helloworld.cpp 复制到 Linux 的 用户文件夹下(本例中是/home/u1234567)



第4步：在Linux下编译刚才传输过来的helloworld.cpp文件并运行



基本使用方法：

★ 编译单源程序文件（*.c）的基本方法：

gcc -Wall -o 可执行文件名 源程序文件名

- 以helloworld.c为例，编译命令为： gcc -Wall -o helloworld helloworld.c
- 不能写成 gcc -Wall -o helloworld.c helloworld（即不可以先源程序名再可执行文件名，否则后果自行体会）

★ 编译单源程序文件（*.cpp）的基本方法：

c++ -Wall -o 可执行文件名 源程序文件名

- 以helloworld.cpp为例，编译命令为： c++ -Wall -o helloworld helloworld.cpp
- 不能写成 c++ -Wall -o helloworld.cpp helloworld（即不可以先源程序名再可执行文件名，否则后果自行体会）

★ 编译多源程序文件的基本方法：

c++ -Wall -o 可执行文件名 源程序文件名1 … 源程序文件名n

- 上学期的4-b16为例（求一元二次方程的根）

> 16、 题目及要求同 4-b14，要求 main 函数在 **4-b16-main.cpp** 中，其余 4 个函数分别放在 **4-b16-sub1.cpp**、**4-b16-sub2.cpp**、**4-b16-sub3.cpp**、**4-b16-sub4.cpp** 中，四个函数的声明放在 **4-b16.h** 中，以上六个文件共同生成可执行文件(**需要在 4-b16-main.cpp 中加入#include "4-b16.h"**)

编译命令为：c++ -Wall -o 4-b16 4-b16-main.cpp 4-b16-sub1.cpp 4-b16-sub2.cpp 4-b16-sub3.cpp 4-b16-sub4.cpp

- ◆ 五个cpp文件名在命令行中出现的顺序无限制
- ◆ 头文件(4-b16.h)不能出现在编译命令中

★ 编译若有错误，则下面会出现错误提示，如果正确，则无提示

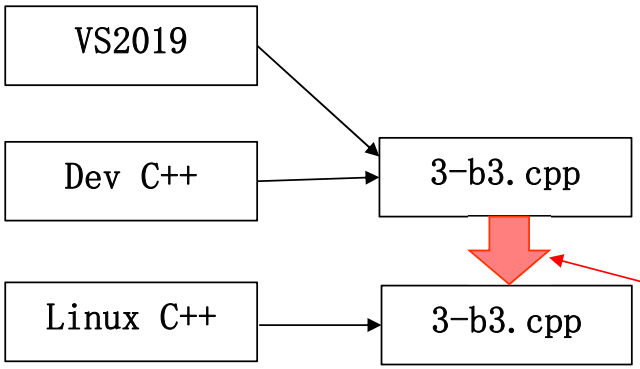
★ 如果需要修改源程序文件，可以在VS2019/DevC++下修改并保存后，再次用FlashFXP传输过来，并再次编译

- Linux下编辑源程序文件(*.cpp/*.c/*.h)的方法本课程不做要求
- 学有余力的同学可自学vi/vim的简单操作（掌握查找、替换、插入、删除、修改、存盘/不存盘退出的基本命令即可）

说明：

★ 通过步骤1-4，一个程序在Linux下编译并验证完成

★ 再次提醒，如果发生源程序修改，一定要用FlashFXP再次上传并编译、运行



Windows下用任何一个编译器编译运行时，如果发生错误，修改后，用其它编译器再次编译，对应的一定是修改后的3-b3.cpp，两者一定是同步的，但是和Linux下的3-b3.cpp不同步!!!

用FlashFXP将Windows下的源程序文件传送到Linux下，两者不同步!!!