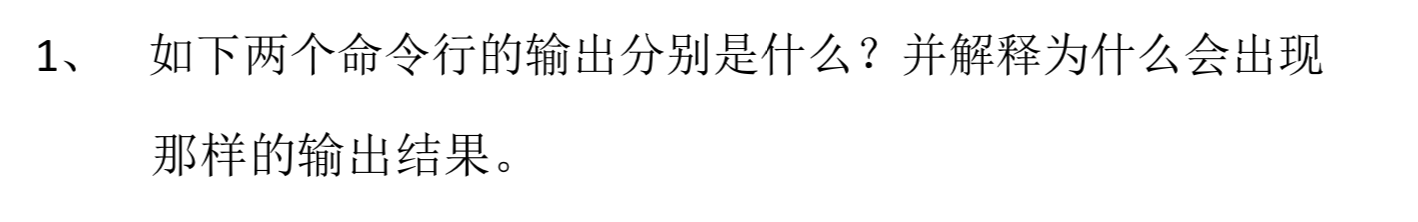
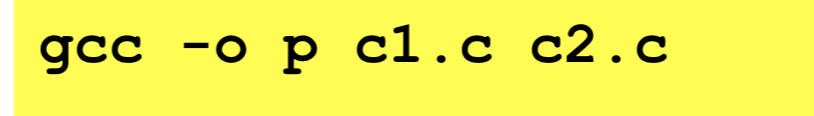
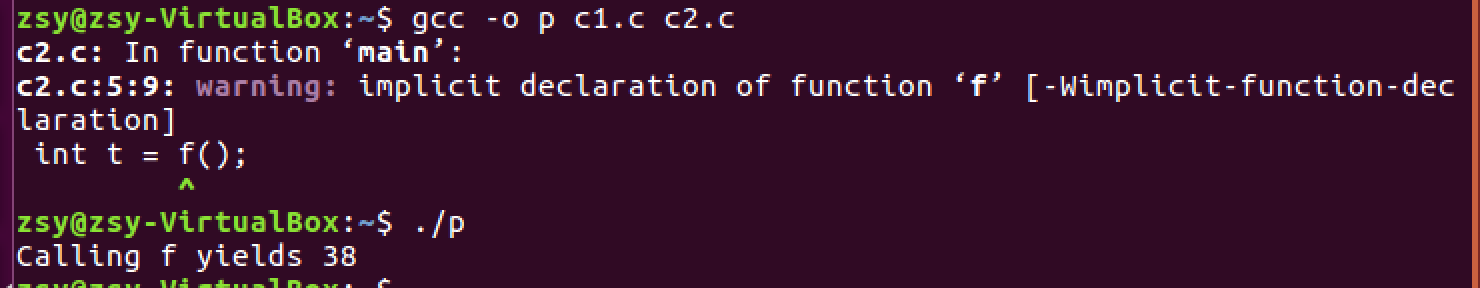
作业



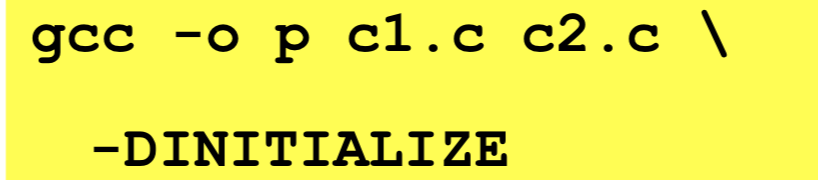


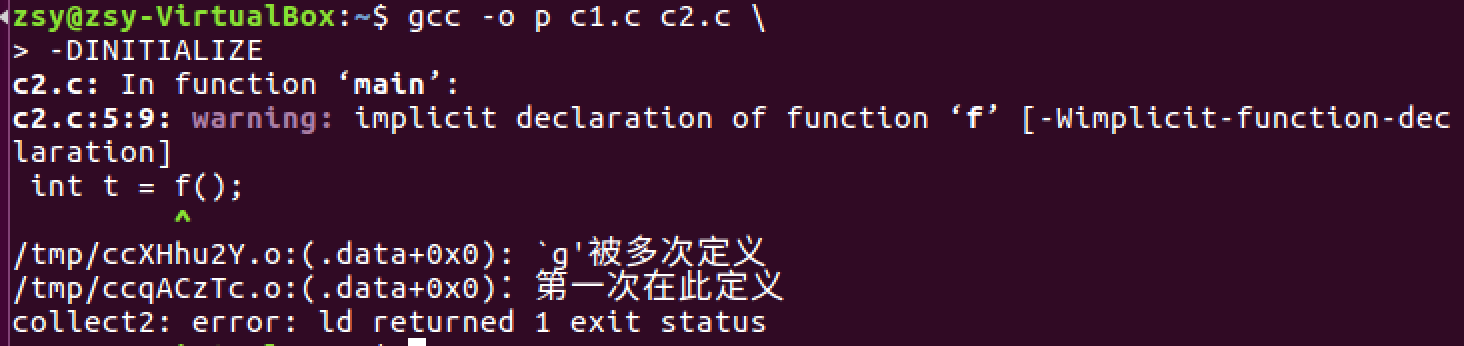


命令行输出警告：f函数在函数所在的c2文件中定义了，但是没有在与之相关联的.h文件中声明

结果：上面没有init，所以init=0，定义了g但是没有赋值

所以会执行if语句的内容，g变成37，再执行f函数，g又加一，变成38



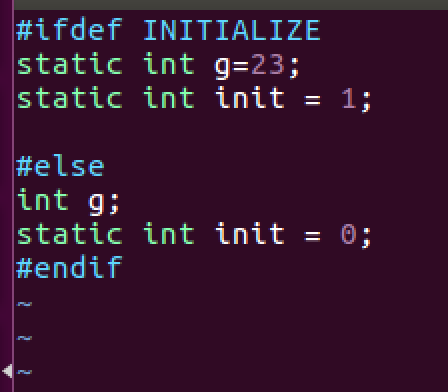


理应结果：下面执行init，所以init=1，g=23，所以不会执行if语句的内容，g还是23，再执行f函数，g又加一，变成24

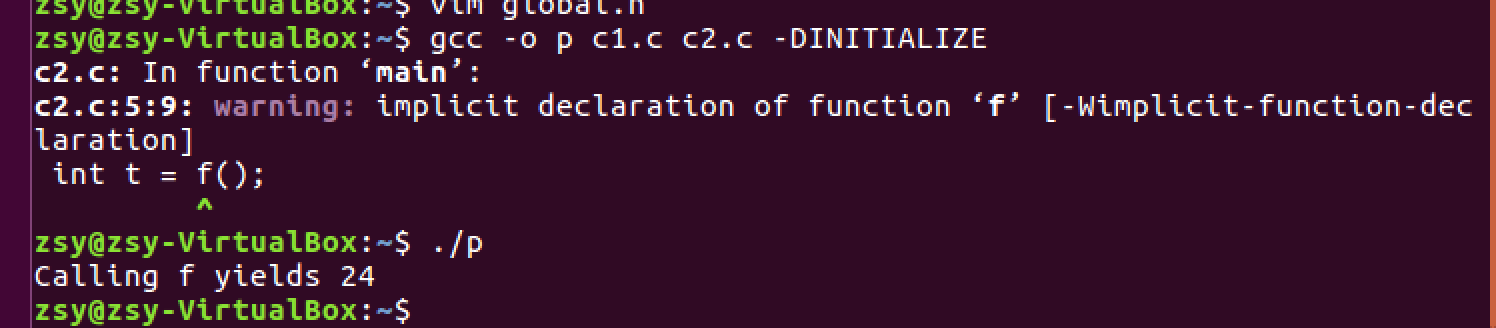
**但是**命令行输出：g被多次定义。因为出现了多个强信号，而在编译时，函数和已初始化的全局变量是强符号，未初始化的全局变量是弱符号，遵从三个规则：  
规则1：不允许有多个强符号。  
规则2：如果有一个强符号和多个弱符号，那么选择强符号。  
规则3：如果有多个弱符号,那么从这些弱符号中任意选择一个。

违反规则1，所以报错。

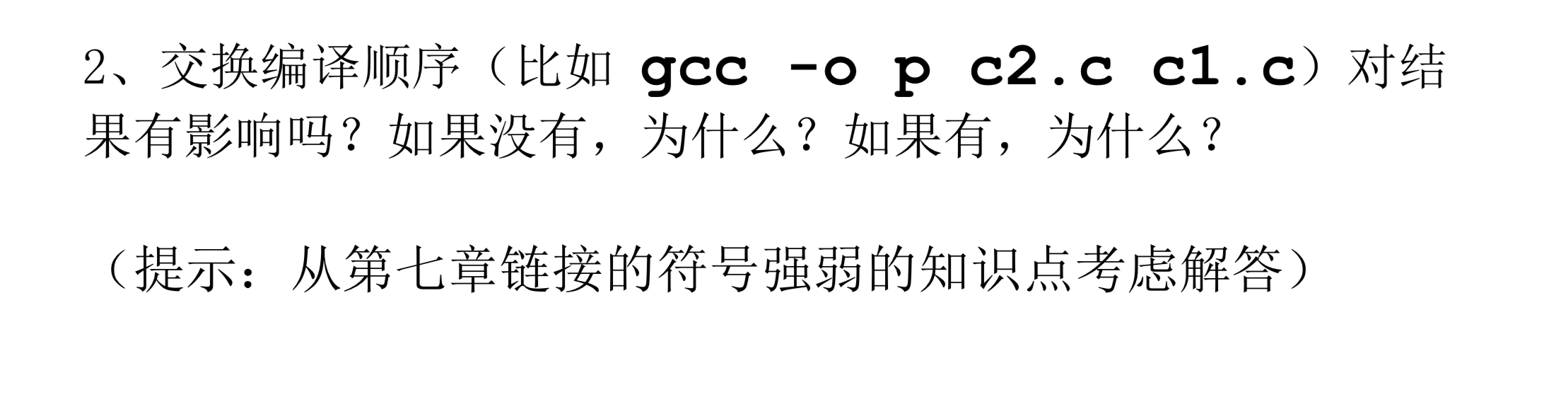
更改方法：

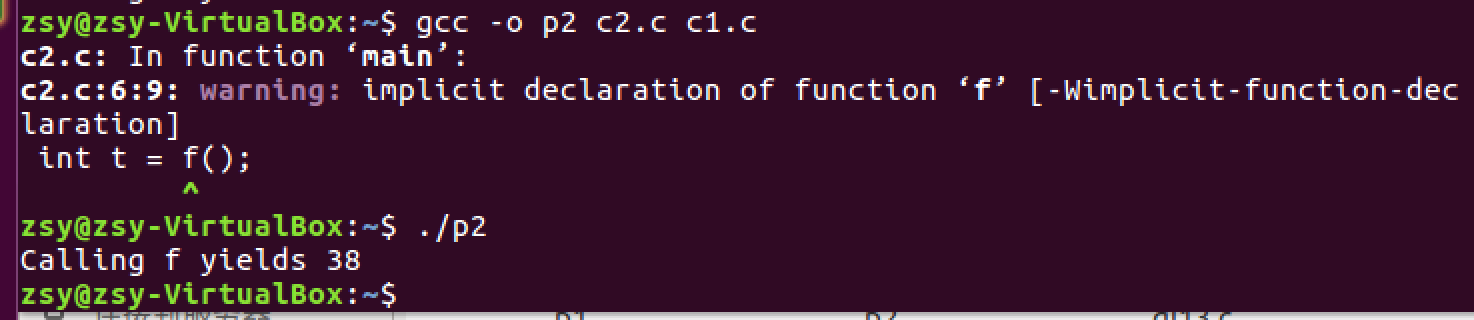


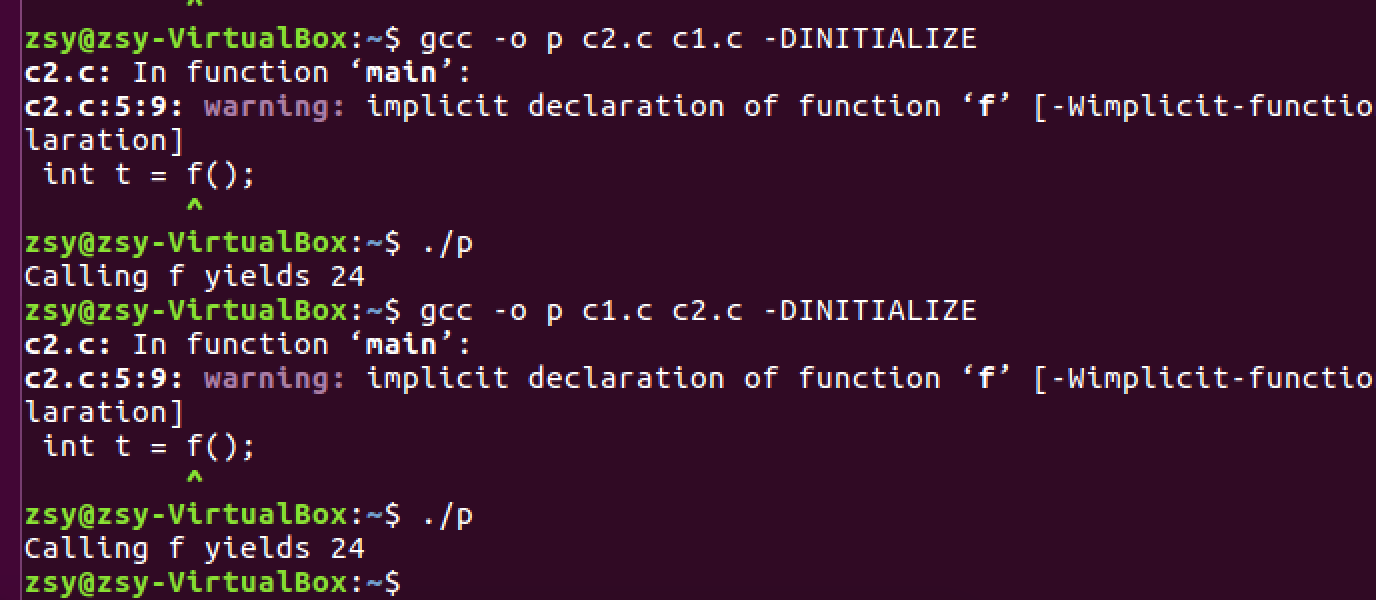
在g前面加一个static，让它变成局部符号，而强弱符号等规则是针对全局符号的，所以就躲避了规则。



就能得到结果了。





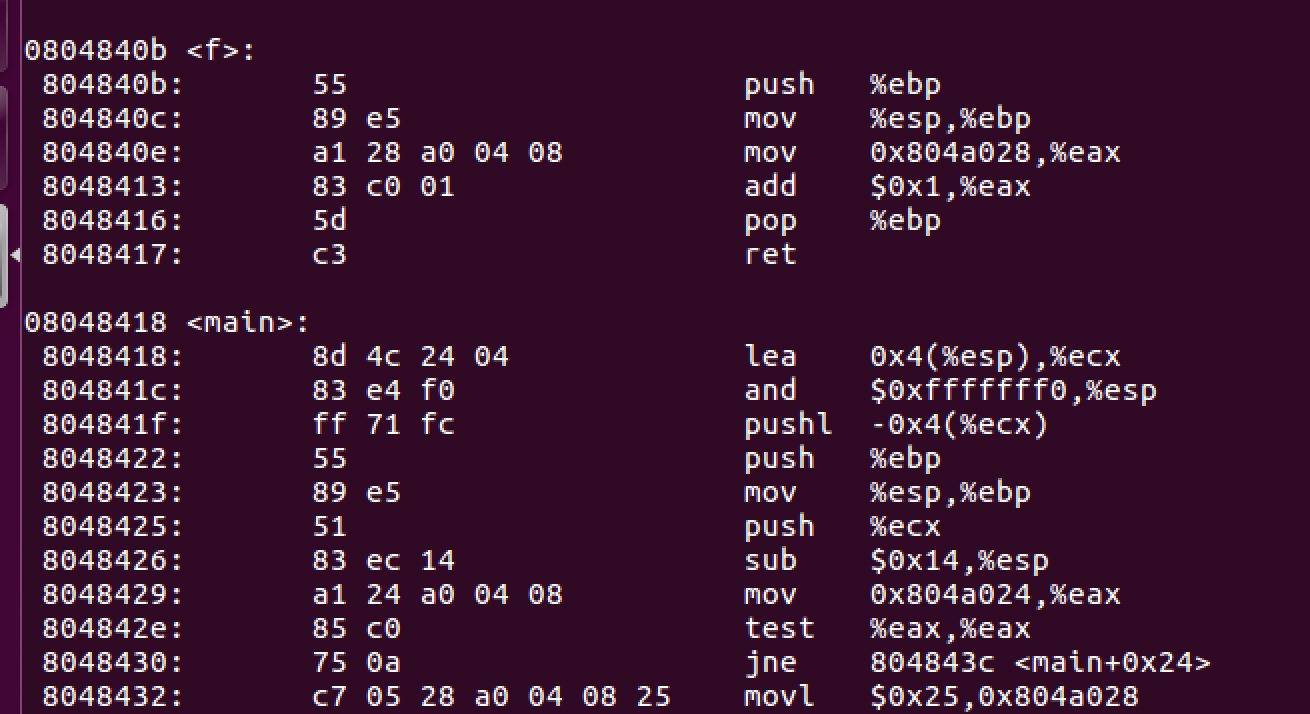


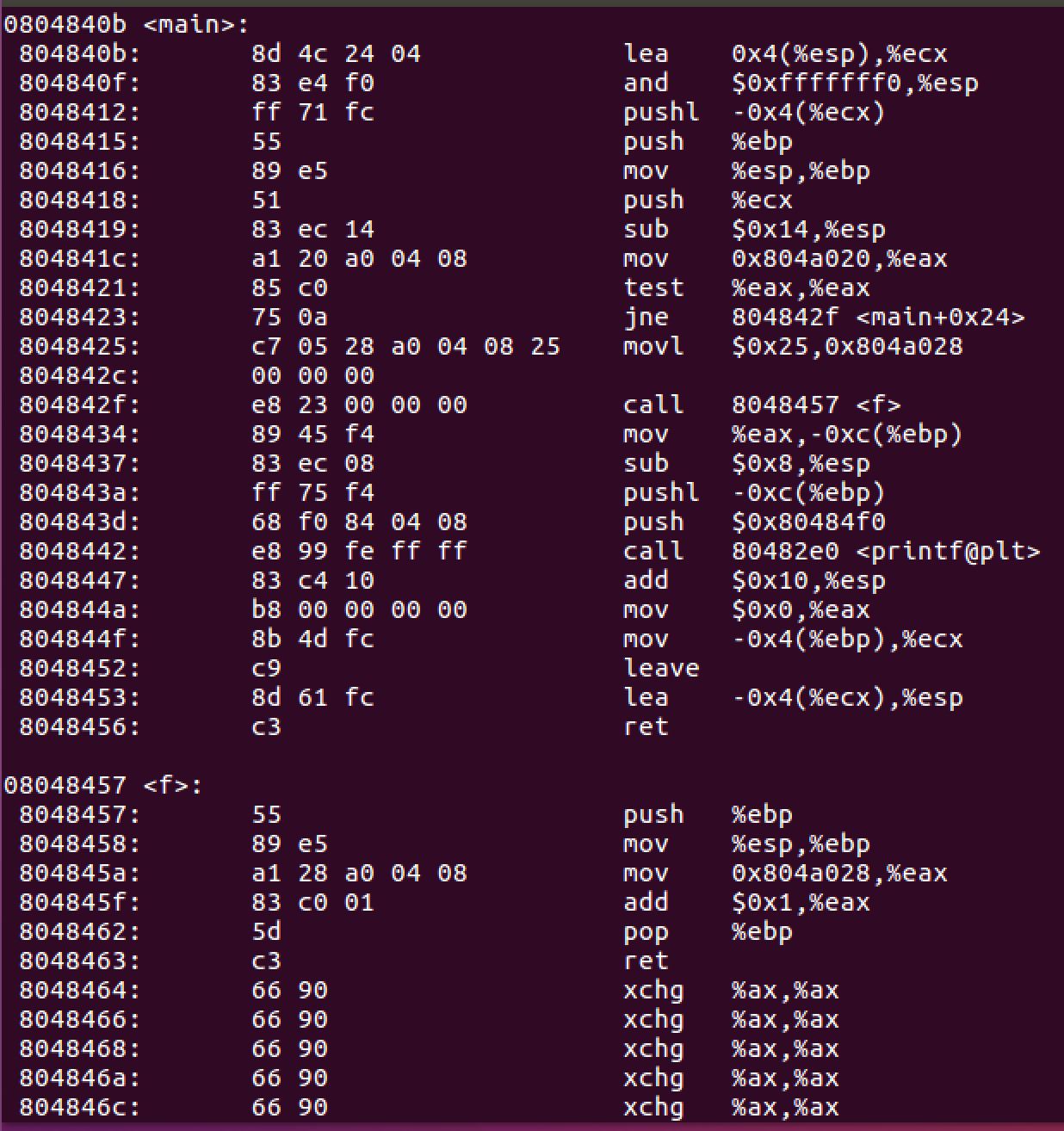
输出不变，没有影响

需要把编译顺序和链接.a文件的顺序关系区分开。

编译只要把函数的声明在有main函数的文件里面包含就行（通过include头文件或者extern declare）。这个和编译顺序没关系。

不过若编译器在命令行中从左向右顺序读取.o文件，则它们的出现顺序有限制：含有某函数定义的文件必须出现在含有调用该函数的文件之后**，但GCC无此限制。**





影响可能就是函数的地址改变了……