H10 PB18000290 胡毅翔

1.阅读[C++:94—类继承（菱形继承、虚继承(virtual虚基类)）](https://blog.csdn.net/qq_41453285/article/details/104442356)，请自行编写简单的具有菱形继承的C++程序（B和C从A继承，D从B、C继承），

1）练习用nm和demangle的方 法分析程序中的2个名字的改编；

g++ class.cpp

得到a.out

nm a.out

部分输出：

0000000000000928 W \_ZN1B2pbEv

000000000000094a W \_ZN1C2pcEv

000000000000096c W \_ZN1D2pdEv

nm --demangle a.out

部分输出：

0000000000000928 W B::pb()

000000000000094a W C::pc()

000000000000096c W D::pd()

分析：

\_Z mangled符号的前缀

N nested names的前缀

1B,2pb等：<length, id>

E 结束标志

v 参数类型

故\_ZN1B2pbEv表示B::pb()。

试分析在虚继承和非虚继承下的对象布局和方法表布局，并给出你的分析方法。

2.

1. 试从产生的汇编代码总结gcc处理inline的特征，内联是在编译的什么阶段被处理？

在未选择优化时，gcc按照对普通函数的处理方式调用f，而没有展开。在-O2优化中对内联函数进行了展开。内联是在编译的链接阶段被处理。

2）试说明编译器进行了哪些优化而得到带-O2选项生成的汇编码。

对f()进行了展开。展开后删除了f()中的死代码。运用了复写传播，减少了复写语句。公共子表达式a也用1进行了替换。

3）如果将inline.h第1行的static去掉，执行gcc inline.c inline1.c，产生如下错误，试说明原因，并指出这是在编译的哪个阶段产生的错误。

这是在链接的阶段产生的错误。

原因是在inline.c和inline1.c都include了inline.h，但没有使用static对函数f()的作用域加以限制，展开后，两个f()函数的作用域发生了重叠。故在链接inline.o和inline1.o时，报错出现f的重复定义。