C++的内部实现

# C++的主要特征

C++三个主要特性：封装、继承、多态。

C++主要涉及到的概念有类，类普通成员函数，类的虚函数，类的静态函数，类的普通成员变量，类的静态变量；以及模板，模板特化，模板实例化；函数重载；类继承；

# 案例A

## 现象

在一个项目中不同的源文件定义相同的类（类名和部分成员函数相同），编译不报错，但是调用时不一定是调用的

调用文件内声明的那个函数。也即，本文件中的那个类可能并不是实际使用的那个类。

演示如下：

现有三个文件main.cpp, exampleA.cpp, exampleB.cpp。内容如下：

main.cpp

----------------------------------------------------------------

#include <iostream>

using namespace std;

class Example

{

public:

void Print()

{

cout << "Example::Print main.cpp" << endl;

}

};

void FunExampleMain()

{

Example i;

i.Print();

}

void FunExampleA();

void FunExampleB();

int main()

{

FunExampleMain();

FunExampleA();

FunExampleB();

return 0;

}

----------------------------------------------------------------

exampleA.cpp

----------------------------------------------------------------

#include <iostream>

using namespace std;

class Example

{

public:

void Print()

{

cout << "Example::Print exampleA.cpp" << endl;

}

};

void FunExampleA()

{

Example i;

i.Print();

}

----------------------------------------------------------------

exampleB.cpp

----------------------------------------------------------------

#include <iostream>

using namespace std;

class Example

{

public:

void Print()

{

cout << "Example::Print exampleB.cpp" << endl;

}

};

void FunExampleB()

{

Example i;

i.Print();

}

在Linux上进行编译，对应的中间文件分别为

main.o exampleA.o exampleB.o

输出为：

Example::Print exampleA.cpp

Example::Print exampleA.cpp

Example::Print exampleA.cpp

我们期待的输出应该是每个文件中的输出应不同，但实际结果却是所有的输出均相同。

即所有文件中对Example::Print的调用最终调用的是exampleA.cpp对应的函数。并且在GCC上，修改编译（准确而言是链接）的顺序后，相应的真实调用也会修改（总是调用第一个输入文件对应的函数）。

## 分析

首先观察各个文件编译后的结果。

main.o

xjl@ubuntu:~$ nm main.o

U \_\_cxa\_atexit

U \_\_dso\_handle

0000000000000092 t \_GLOBAL\_\_sub\_I\_\_Z14FunExampleMainv

000000000000003a T main

U \_\_stack\_chk\_fail

U \_Z11FunExampleAv

U \_Z11FunExampleBv

0000000000000000 T \_Z14FunExampleMainv

0000000000000054 t \_Z41\_\_static\_initialization\_and\_destruction\_0ii

0000000000000000 W \_ZN7Example5PrintEv

U \_ZNSolsEPFRSoS\_E

U \_ZNSt8ios\_base4InitC1Ev

U \_ZNSt8ios\_base4InitD1Ev

U \_ZSt4cout

U \_ZSt4endlIcSt11char\_traitsIcEERSt13basic\_ostreamIT\_T0\_ES6\_

0000000000000000 b \_ZStL8\_\_ioinit

U \_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc

exampleA.o

xjl@ubuntu:~$ nm exampleA.o

U \_\_cxa\_atexit

U \_\_dso\_handle

0000000000000078 t \_GLOBAL\_\_sub\_I\_\_Z11FunExampleAv

U \_\_stack\_chk\_fail

0000000000000000 T \_Z11FunExampleAv

000000000000003a t \_Z41\_\_static\_initialization\_and\_destruction\_0ii

0000000000000000 W \_ZN7Example5PrintEv

U \_ZNSolsEPFRSoS\_E

U \_ZNSt8ios\_base4InitC1Ev

U \_ZNSt8ios\_base4InitD1Ev

U \_ZSt4cout

U \_ZSt4endlIcSt11char\_traitsIcEERSt13basic\_ostreamIT\_T0\_ES6\_

0000000000000000 b \_ZStL8\_\_ioinit

U \_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc

exampleB.o