**C++编译器多态实现原理专题讲座**

written by王保明

1. 多态基础

|  |
| --- |
| **多态的实现效果**  多态：同样的调用语句有多种不同的表现形态；  **多态实现的三个条件**  有继承、有virtual重写、有父类指针（引用）指向子类对象。  **多态的C++实现**  virtual关键字，告诉编译器这个函数要支持多态；不要根据指针类型判断如何调用；而是要根据指针所指向的实际对象类型来判断如何调用 **多态的理论基础**  动态联编PK静态联编。根据实际的对象类型来判断重写函数的调用。  **多态的重要意义**  设计模式的基础。  **实现多态的理论基础**  函数指针做函数参数  铁律10： C函数指针是C++至高无上的荣耀。C函数指针一般有两种用法（正、反）。 |
| C++中多态的实现原理  当类中声明虚函数时，编译器会在类中生成一个虚函数表  虚函数表是一个存储类成员函数指针的数据结构  虚函数表是由编译器自动生成与维护的  virtual成员函数会被编译器放入虚函数表中  存在虚函数时，每个对象中都有一个指向虚函数表的指针(vptr指针) |
|  |
|  |
| 说明1：  通过虚函数表指针VPTR调用重写函数是在程序运行时进行的，因此需要通过寻址操作才能确定真正应该调用的函数。而普通成员函数是在编译时就确定了调用的函数。在效率上，虚函数的效率要低很多。  说明2：  出于效率考虑，没有必要将所有成员函数都声明为虚函数 |
|  |

2、**构造函数中能调用虚函数，实现多态吗？why？**

|  |
| --- |
| 1）**对象中的VPTR指针什么时候被初始化？**  对象在创建的时,由编译器对VPTR指针进行初始化  只有当对象的构造完全结束后VPTR的指向才最终确定  父类对象的VPTR指向父类虚函数表  子类对象的VPTR指向子类虚函数表 |
| 2）分析过程  画图分析  3）结论：构造函数中调用多态函数,不能实现多态。 |
| 4）如何证明vptr指针的存在哪? |
|  |