

C 虚函数表深入剖析计算机等级考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/646/2021_2022_C___E8_99_9A_E5_87_BD_E6_c97_646112.htm C 中的虚函数的作用主要是实现了多态的机制。关于多态，简而言之就是用父类型别的指针指向其子类的实例，然后通过父类的指针调用实际子类的成员函数。这种技术可以让父类的指针有“多种形态”，这是一种泛型技术。所谓泛型技术，说白了就是试图使用不变的代码来实现可变的算法。比如：模板技术，RTTI技术，虚函数技术，要么是试图做到在编译时决议，要么试图做到运行时决议。关于虚函数的使用方法，我在这里不做过多的阐述。大家可以看看相关的C的书籍。在这篇文章中，我只想从虚函数的实现机制上面为大家一个清晰的剖析。当然，相同的文章在网上也出现过一些了，但我总感觉这些文章不是很容易阅读，大段大段的代码，没有图片，没有详细的说明，没有比较，没有举一反三。不利于学习和阅读，所以这是我想写下这篇文章的原因。也希望大家多给我提意见。言归正传，让我们一起进入虚函数的世界。虚函数表对C了解的人都应该知道虚函数（Virtual Function）是通过一张虚函数表（Virtual Table）来实现的。简称为V-Table。在这个表中，主是要一个类的虚函数的地址表，这张表解决了继承、覆盖的问题，保证其容真实反应实际的函数。这样，在有虚函数的类的实例中这个表被分配在了这个实例的内存中，所以，当我们用父类的指针来操作一个子类的时候，这张虚函数表就显得由为重要了，它就像一个地图一样，指明了实际所应该调用的函数。这里我们着重看一下这张虚函数表。在C的

标准规格说明书中说到，编译器必需要保证虚函数表的指针存在于对象实例中最前面的位置（这是为了保证正确取到虚函数的偏移量）。这意味着我们通过对象实例的地址得到这张虚函数表，然后就可以遍历其中函数指针，并调用相应的函数。听我扯了那么多，我可以感觉出来你现在可能比以前更加晕头转向了。没关系，下面就是实际的例子，相信聪明的你一看就明白了。假设我们有这样的一个类：`class Base { public: virtual void f() { cout << 100Test` 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com