

C 多态技术的实现和反思 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/253/2021_2022_C___E5_A4_9A_E6_80_81_E6_c97_253838.htm 别 面向对象技术最早出现于1960年代的Simula 67系统，并且在1970年代保罗阿托实验室开发的Smalltalk系统中发展成熟。然而对于大部分程序员来说，C 是第一个可用的面向对象程序设计语言。因此，我们关于面向对象的很多概念和思想直接来自于C。但是，C 在实现面向对象中关键的多态性时，选择了与Smalltalk完全不同的方案。其结果是，尽管在表面上两者都实现了相似的多态性，但是在实践中却有着巨大的区别。具体的说，C 的多态性实现更加高效，但是并不适用于所有场合。很多经验不足的C 开发者不明白这个道理，在不合适的场合强行使用C 的多态性机制，落入削足适履的陷阱而不能自拔。本文将详细探讨C 多态性技术的局限性及解决的办法。两种不同虚方法调用实现技术 C 的多态性是C 实现面向对象技术的基础。具体的说，通过一个指向基类的指针调用虚成员函数的时候，运行时系统将能够根据指针所指向的实际对象调用恰当的成员函数实现。如下所示：

```
class Base { public: virtual void vmf() { ... } }.  
class Derived : public Base { public: virtual void vmf() { ... } }.  
Base* p = new Base(). p->vmf(). // 这里调用Base::vmf  
p = new Derived(). p->vmf(). // 这里调用 // Derived::vmf ...  
... 请注意代码中突出注释的两行，虽然其表面语法完全相同，但是却分别调用了不同的函数实现。所谓的“多态”即就此而言。这些知识是每一个C 开发者都熟知的。现在我们假设自己是语言的实现者，我们应当如何来实现这种多态性？稍加思考，我
```

们不难得到一个基本的思路。多态性的实现要求我们增加一个间接层，在这个间接层中拦截对于方法的调用，然后根据指针所指向的实际对象调用相应的方法实现。在这个过程中我们人为增加的这个间接层非常重要，它需要完成以下几项工作：1. 获知方法调用的全部信息，包括被调用的是哪个方法，传入的实际参数有哪些。2. 获知调用发生时指针（引用）所指向的实际对象。3. 根据第1、2步获得的信息，找到合适的方法实现代码，执行调用。这里的关键在于如何在第3步中找到合适的方法实现代码。由于多态性是就对象而言的，因此我们在设计时要把合适的方法实现代码与对象绑定到一起。也就是说，必须在对象级别实现一个查找表结构，根据1、2步获得的对象和方法信息，在这个查找表中找到实际的方法代码地址，并加以调用。现在问题变成了，我们应当根据什么信息进行方法查找。对于这个问题有两个不同的解决思路，一个是根据名称进行查找，另一个是根据位置进行查找。粗看上去这两种思路似乎没什么大的差别，但是在实践中，这两种不同的实现思路导致了巨大的差别。下面我们详细地加以考察。在Smalltalk、Python、Ruby等动态面向对象语言中，实际方法的查找是根据方法名称进行的，其查找表结构如下：由于这种查找表根据方法的名称进行方法查找，因此在查找过程中涉及字符串比较，效率较差。但是这种查找表有一个突出的优点，就是有效空间利用率高。为了说明这一点，我们假设一个基类Base中有100个方法可供派生类改写（因此所有Base对象所共享的方法查找表有100项），而它的一个派生类Derived仅仅只打算改写其中5个方法，那么Derived类对象的方法查找表只需要5项。当一个方法调用发

生的时候，runtime根据被调用的方法名称在这个长度为5的方法查找表中进行字符串查找，如果发现该方法在查找表中，则执行调用，否则将调用转寄（forward）给Base类执行。这是虚方法调用的标准行为。当派生类实际改写的方法数量很少的时候，可以将查找表安排成线性表，查找时顺序比较，这种情况下有效空间利用率达到100%。如果派生类实际改写的方法数量较多，那么可以采用散列表，如果采用合理的散列函数，同样可以在空间利用率很高（一般可接近75%）的情况下实现方法的快速查找。应当注意到，由于编译器可以很容易地获得所有被改写方法的名称，因此可以执行标准的gperf算法获得最优的散列函数。事实上，我们还可以这样理解这种方案的优势，把表中每一项的“方法名”项视为“方法地址”项的描述信息，因此可以认为这种方案中的方法查找表携带自描述信息（或者称为元数据）。基于这种携带自描述信息的数据结构，可以实现丰富多彩的扩展功能，比如在运行时插入新的方法，或者用户层次上的方法调用截获等。因此，我们可以说这一方案的适用面广，强大灵活，但在执行效率上并非最优。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