

C 中的指针(二)函数指针 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/134/2021\\_2022\\_C\\_\\_\\_E4\\_B8\\_AD\\_E7\\_9A\\_84\\_E6\\_c97\\_134378.htm](https://www.100test.com/kao_ti2020/134/2021_2022_C___E4_B8_AD_E7_9A_84_E6_c97_134378.htm) 先说一下C式的函数指针。这种函数指针的应用十分广泛。对于任何函数 `void print(string s)`，它的指针这样定义：`void (*pfun)(string) = NULL`。`pfun = amp.c.bool(C::*p)() = amp.C::test`。在VS2003里右边可以省去 `C::`，到了VS2005语法更严格了，被禁止了。这里的成员函数指针对非静态函数有效。静态函数不依赖于任何object，它的表示方法和C的一样。对于非静态成员函数的指针的继承关系是这样的：`upcast`合法，`downcast`不合法。这样的到的指针永远是安全的。非静态成员函数指针在实际程序中的应用很多。一个典型的例子是用来写state machine(状态机器？)。例如程序在控制一个机器人的初始化阶段。整个初始化需要三个函数：1。初始化机器人的身子，2。初始化机器人的左手，3。初始化机器人的右手。这样我们在state machine中用两个成员函数指针分别指向当前的状态和下一个状态 `bool(CStateMachine::*m_pCurrentState)`，`bool(CStateMachine::*m_pNextState)`。。一开始永远叫`Start()`  
`CStateMachine::CStateMachine { m_pCurrentState = CStateMachine::Start. }`然后在每一个State里面管理状态变化：  
`bool CStateMachine::Start() { ..... m_pNextState = CStateMachine::InitializeLeftHand(). }` 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](http://www.100test.com)