

计算机二级C实例编程:C\_C 中回调函数初探计算机二级考试  
PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/644/2021\\_2022\\_\\_E8\\_AE\\_A1\\_](https://www.100test.com/kao_ti2020/644/2021_2022__E8_AE_A1_)

[E7\\_AE\\_97\\_E6\\_9C\\_BA\\_E4\\_c97\\_644881.htm](#) 编辑特别推荐: 全国计算机等级考试（等考）指定教材 全国计算机等级考试学习视频 全国计算机等级考试网上辅导招生 全国计算机等级考试时间及科目预告 百考试题教育全国计算机等级考试在线测试平台 全国计算机等级考试资料下载 全国计算机等级考试论坛 计算机等级考试四级应用题解析汇总 简介 对于很多初学者来说，往往觉得回调函数很神秘，很想知道回调函数的工作原理。本文将要解释什么是回调函数、它们有什么好处、为什么要使用它们等等问题，在开始之前，假设你已经熟知了函数指针。什么是回调函数？简而言之，回调函数就是一个通过函数指针调用的函数。如果你把函数的指针（地址）作为参数传递给另一个函数，当这个指针被用为调用它所指向的函数时，我们就说这是回调函数。为什么要使用回调函数？因为可以把调用者与被调用者分开。调用者不关心谁是被调用者，所有它需知道的，只是存在一个具有某种特定原型、某些限制条件（如返回值为int）的被调用函数。如果想知道回调函数在实际中有什么作用，先假设有这样一种情况，我们要编写一个库，它提供了某些排序算法的实现，如冒泡排序、快速排序、shell排序、shake排序等等，但为使库更加通用，不想在函数中嵌入排序逻辑，而让使用者来实现相应的逻辑；或者，想让库可用于多种数据类型（int、float、string），此时，该怎么办呢？可以使用函数指针，并进行回调。回调可用于通知机制，例如，有时要在程序中设置一个计时

器，每到一定时间，程序会得到相应的通知，但通知机制的实现者对我们的程序一无所知。而此时，就需有一个特定原型的函数指针，用这个指针来进行回调，来通知我们的程序事件已经发生。实际上，SetTimer() API使用了一个回调函数来通知计时器，而且，万一没有提供回调函数，它还会把一个消息发往程序的消息队列。另一个使用回调机制的API函数是EnumWindow()，它枚举屏幕上所有的顶层窗口，为每个窗口调用一个程序提供的函数，并传递窗口的处理程序。如果被调用者返回一个值，就继续进行迭代，否则，退出。EnumWindow()并不关心被调用者在何处，也不关心被调用者用它传递的处理程序做了什么，它只关心返回值，因为基于返回值，它将继续执行或退出。不管怎么说，回调函数是继续自C语言的，因而，在C中，应只在与C代码建立接口，或与已有的回调接口打交道时，才使用回调函数。除了上述情况，在C中应使用虚拟方法或函数符（functor），而不是回调函数。一个简单的回调函数实现下面创建了一个sort.dll的动态链接库，它导出了一个名为CompareFunction的类型

```
--typedef int (__stdcall *CompareFunction)(const byte*, const byte*), 它就是回调函数的类型。另外，它也导出了两个方法：Bubblesort()和Quicksort()，这两个方法原型相同，但实现了不同的排序算法。 void DLLDIR __stdcall Bubblesort(byte* array,int size,int elem_size,CompareFunction cmpFunc). // cmpFunc是函数指针 void DLLDIR __stdcall Quicksort(byte* array,int size,int elem_size,CompareFunction cmpFunc). 这两个函数接受以下参数： byte * array：指向元素数组的指针（任意类型）。 int size：数组中元素的个数。 int elem_size：数组中
```

一个元素的大小，以字节为单位。 CompareFunction cmpFunc  
：带有上述原型的指向回调函数的指针。 这两个函数的会对  
数组进行某种排序，但每次都需决定两个元素哪个排在前面  
，而函数中有一个回调函数，其地址是作为一个参数传递进  
来的。对编写者来说，不必介意函数在何处实现，或它怎样  
被实现的，所需在意的只是两个用于比较的元素的地址，并  
返回以下的某个值（库的编写者和使用者都必须遵守这个约  
定）： -1：如果第一个元素较小，那它在已排序好的数组中  
，应该排在第二个元素前面。 0：如果两个元素相等，那么  
它们的相对位置并不重要，在已排序好的数组中，谁在前面  
都无所谓。 1：如果第一个元素较大，那在已排序好的数组  
中，它应该排第二个元素后面。 基于以上约定，函  
数Bubblesort()的实现如下， Quicksort()就稍微复杂一点： void  
DLLDIR \_\_stdcall Bubblesort(byte\* array,int size,int  
elem\_size,CompareFunction cmpFunc) { for(int i=0. i lt. size-1. j ) {  
//回调比较函数 if(1 == (\*cmpFunc)(array j\*elem\_size,array (j  
1)\*elem\_size)) { //两个相比较的元素相交换 byte\* temp = new  
byte[elem\_size]. memcpy(temp, array j\*elem\_size, elem\_size).  
memcpy(array j\*elem\_size,array (j 1)\*elem\_size,elem\_size).  
memcpy(array (j 1)\*elem\_size, temp, elem\_size). 0delete [] temp. } }  
} } 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)