

计算机二级C 辅导:C 回调函数用法计算机等级考试 PDF转换
可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/616/2021_2022__E8_AE_A1_

[E7_AE_97_E6_9C_BA_E4_c97_616941.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 2009年下半年全国计算机二级考试报名时间从6月1日起已经开始报名。详情点击

：2009年下半年全国计算机等级考试各地报名点汇总。2009年下半年全国计算机二级考试时间是2009年9月19日至23日。更多优质资料尽在百考试题论坛 百考试题在线题库。 一回调函数 我们经常C 设计时通过使用回调函数可以使有些应用（如定时器事件回调处理、用回调函数记录某操作进度等）变得非常方便和符合逻辑，那么它的内在机制如何呢，怎么定义呢?它和其它函数（比如钩子函数）有何不同呢？使用回调函数实际上就是在调用某个函数（通常是API函数）时，将自己的一个函数（这个函数为回调函数）的地址作为参数传递给那个函数。而那个函数在需要的时候，利用传递的地址调用回调函数，这时你可以利用这个机会在回调函数中处理消息或完成一定的操作。至于如何定义回调函数，跟具体使用的API函数有关，一般在帮助中有说明回调函数的参数和返回值等。C 中一般要求在回调函数前加CALLBACK（相当于FAR PASCAL），这主要是说明该函数的调用方式。至于钩子函数，只是回调函数的一个特例。习惯上把

与SetWindowsHookEx函数一起使用的回调函数称为钩子函数。也有人把利用VirtualQueryEx安装的函数称为钩子函数，不过这种叫法不太流行。也可以这样，更容易理解：回调函数就好像是一个中断处理函数，系统在符合你设定的条件时自动调用。为此，你需要做三件事：1. 声明；2. 定义；3. 设置触发条件，就是在你的函数中把你的回调函数名称转化为地址作为一个参数，以便于系统调用。声明和定义时应注意：回调函数由系统调用，所以可以认为它属于WINDOWS系统，不要把它当作你的某个类的成员函数。二回调函数、消息和事件例程调用(calling)机制从汇编时代起已经大量使用：准备一段现成的代码，调用者可以随时跳转至此段代码的起始地址，执行完后再返回跳转时的后续地址。CPU为此准备了现成的调用指令，调用时可以压栈保护现场，调用结束后从堆栈中弹出现场地址，以便自动返回。借堆栈保护现场真是一项绝妙的发明，它使调用者和被调者可以互不相识，于是才有了后来的函数和构件。此调用机制并非完美。回调函数就是一例。函数之类本是为调用者准备的美餐，其烹制者应对食客了如指掌，但实情并非如此。例如，写一个快速排序函数供他人调用，其中必包含比较大小。麻烦来了：此时并不知要比较的是何类数据--整数、浮点数、字符串？于是只好为每类数据制作一个不同的排序函数。更通行的办法是在函数参数中列一个回调函数地址，并通知调用者：君需自己准备一个比较函数，其中包含两个指针类参数，函数要比较此二指针所指数据之大小，并由函数返回值说明比较结果。排序函数借此调用者提供的函数来比较大小，借指针传递参数，可以全然不管所比较的数据类型。被调用者回头调用调

用者的函数（够咬嘴的），故称其为回调（callback）。回调函数使程序结构乱了许多。Windows API 函数集中有不少回调函数，尽管有详尽说明，仍使初学者一头雾水。恐怕这也是无奈之举。无论何种事物，能以树形结构单向描述毕竟让人舒服些。如果某家族中孙辈又是某祖辈的祖辈，恐怕无人能理清其中的头绪。但数据处理之复杂往往需要构成网状结构，非简单的客户/服务器关系能穷尽。Windows 系统还包含着另一种更为广泛的回调机制，即消息机制。消息本是 Windows 的基本控制手段，乍看与函数调用无关，其实是一种变相的函数调用。发送消息的目的是通知收方运行一段预先准备好的代码，相当于调用一个函数。消息所附带的 WParam 和 LParam 相当于函数的参数，只不过比普通参数更通用一些。应用程序可以主动发送消息，更多情况下是坐等 Windows 发送消息。一旦消息进入所属消息队列，便检感兴趣的那些，跳转去执行相应的消息处理代码。操作系统本是为应用程序服务，由应用程序来调用。而应用程序一旦启动，却要反过来等待操作系统的调用。这分明也是一种回调，或者说是一种广义回调。其实，应用程序之间也可以形成这种回调。假如进程 B 收到进程 A 发来的消息，启动了一段代码，其中又向进程 A 发送消息，这就形成了回调。这种回调比较隐蔽，弄不好会搞成递归调用，若缺少终止条件，将会循环不已，直至把程序搞垮。若是故意编写成此递归调用，并设好终止条件，倒是很有意思。但这种程序结构太隐蔽，除非十分必要，还是不用为好。利用消息也可以构成狭义回调。上面所举排序函数一例，可以把回调函数地址换成窗口 handle。如此，当需要比较数据大小时，不是去调用回调函数

，而是借 API 函数 SendMessage 向指定窗口发送消息。收到消息方负责比较数据大小，把比较结果通过消息本身的返回值传给消息发送方。所实现的功能与回调函数并无不同。当然，此例中改为消息纯属画蛇添脚，反倒把程序搞得很慢。但在其他情况下并非总是如此，特别是需要异步调用时，发送消息是一种不错的选择。假如回调函数中包含文件处理之类的低速处理，调用方等不得，需要把同步调用改为异步调用，去启动一个单独的线程，然后马上执行后续代码，其余的事让线程慢慢去做。一个替代办法是借 API 函数 PostMessage 发送一个异步消息，然后立即执行后续代码。这要比自己搞个线程省事许多，而且更安全。如今我们是活在一个 object 时代。只要与编程有关，无论何事都离不开 object。但 object 并未消除回调，反而把它发扬光大，弄得到处都是，只不过大都以事件（event）的身份出现，镶嵌在某个结构之中，显得更正统，更容易被人接受。应用程序要使用某个构件，总要先弄清构件的属性、方法和事件，然后给构件属性赋值，在适当的时候调用适当的构件方法，还要给事件编写处理例程，以备构件代码来调用。何谓事件？它不过是一个指向事件例程的地址，与回调函数地址没什么区别。不过，此种回调方式比传统回调函数要高明许多。首先，它把让人不太舒服的回调函数变成一种自然而然的处理例程，使编程者顿觉气顺。再者，地址是一个危险的东西，用好了可使程序加速，用不好处处是陷阱，程序随时都会崩溃。现代编程方式总是想法把地址隐藏起来（隐藏比较彻底的如 VB 和 Java），其代价是降低了程序效率。事件例程（？）使编程者无需直接操作地址，但并不会使程序减速。（例程似乎是进程的台湾翻

译。) 三精妙比喻:回调函数还真有点像您随身带的BP机:告诉别人号码,在它有事情时Call您。回调用于层间协作,上层将本层函数安装在下层,这个函数就是回调,而下层在一定条件下触发回调,例如作为一个驱动,是一个底层,他在收到一个数据时,除了完成本层的处理工作外,还将进行回调,将这个数据交给上层应用层来做进一步处理,这在分层的数据通信中很普遍。其实回调和API非常接近,他们的共性都是跨层调用的函数。但区别是API是低层提供给高层的调用,一般这个函数对高层都是已知的;而回调正好相反,他是高层提供给底层的调用,对于低层他是未知的,必须由高层进行安装,这个安装函数其实就是一个低层提供的API,安装后低层不知道这个回调的名字,但它通过一个函数指针来保存这个回调,在需要调用时,只需引用这个函数指针和相关的参数指针。其实:回调就是该函数写在高层,低层通过一个函数指针保存这个函数,在某个事件的触发下,低层通过该函数指针调用高层那个函数。

四调用方式 软件模块之间总是存在着一定的接口,从调用方式上,可以把他们分为三类:同步调用、回调和异步调用。同步调用是一种阻塞式调用,调用方要等待对方执行完毕才返回,它是一种单向调用;回调是一种双向调用模式,也就是说,被调用方在接口被调用时也会调用对方的接口;异步调用是一种类似消息或事件的机制,不过它的调用方向刚好相反,接口的服务在收到某种讯息或发生某种事件时,会主动通知客户方(即调用客户方的接口)。回调和异步调用的关系非常紧密,通常我们使用回调来实现异步消息的注册,通过异步调用来实现消息的通知。同步调用是三者当中最简单的,而回调又常常是异

步调用的基础。对于不同类型的语言（如结构化语言和对象语言）、平台（Win32、JDK）或构架（CORBA、DCOM、WebService），客户和服务的交互除了同步方式以外，都需要具备一定的异步通知机制，让服务方（或接口提供方）在某些情况下能够主动通知客户，而回调是实现异步的一个最简捷的途径。对于一般的结构化语言，可以通过回调函数来实现回调。回调函数也是一个函数或过程，不过它是一个由调用方自己实现，供被调用方使用的特殊函数。在面向对象的语言中，回调则是通过接口或抽象类来实现的，我们把实现这种接口的类成为回调类，回调类的对象成为回调对象。对于象C或Object Pascal这些兼容了过程特性的对象语言，不仅提供了回调对象、回调方法等特性，也能兼容过程语言的回调函数机制。Windows平台的消息机制也可以看作是回调的一种应用，我们通过系统提供的接口注册消息处理函数（即回调函数），从而实现接收、处理消息的目的。由于Windows平台的API是用C语言来构建的，我们可以认为它也是回调函数的一个特例。对于分布式组件代理体系CORBA，异步处理有多种方式，如回调、事件服务、通知服务等。事件服务和通知服务是CORBA用来处理异步消息的标准服务，他们主要负责消息的处理、派发、维护等工作。对一些简单的异步处理过程，我们可以通过回调机制来实现。下面我们集中比较具有代表性的语言（C、Object Pascal）和架构（CORBA）来分析回调的实现方式、具体作用等。过程语言中的回调（C）(1)函数指针 回调在C语言中是通过函数指针来实现的,通过将回调函数的地址传给被调函数从而实现回调。因此，要实现回调，必须首先定义函数指针，请看下面的

例子： `void Func(char *s); // 函数原型 void (*pFunc) (char *).` // 函数指针可以看出，函数的定义和函数指针的定义非常类似。一般的化，为了简化函数指针类型的变量定义，提高程序的可读性，我们需要把函数指针类型自定义一下。 `typedef void(*pcb)(char *).` 回调函数可以象普通函数一样被程序调用?
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