**钩子函数使用小结**

最近在研究一个项目这个项目是一个关于Windows窗口恢复的小软件，旨在方便用户不小心关闭某个窗口后能有一次“后悔”的机会恢复刚关闭的窗口。如何知道某个窗口被关闭了呢？而且在窗口关闭后必须得记录相关信息才能进行窗口的恢复。Windows系统是基于消息机制，因此窗口的关闭动作也是窗口过程接收到相应的关闭消息后系统才将窗口关闭。钩子作为Windows系统的主要特性之一，通过设置“钩子”，应用程序可以在系统级对所有消息、事件进行过滤，访问在正常情况下无法访问的消息。因此，我们可以通过设置钩子来抓取窗口关闭的消息来记录窗口的相关信息以进行恢复。由于多了这么一道处理过程，系统性能会受到一定的影响，所以大家在必要的时候才使用“钩子”，并在使用完毕及时将其删除。

首先让我们看看HOOK函数是怎么安装、调用和删除的。应用程序通常是调用SetWindowsHookEx()函数来进行安装的，其函数的原型如下：

SetWindowsHookEx(Int idHook,HOOKPROClpfn,HINSTANCE hMod,DWORD dwThreadId);

参数说明：

idHook是”钩子”的类型，”钩子”的类型一共有13种。

lpfn指向“钩子”过程的指针。

hMod“钩子”过程所在模块的句柄。

dwThreadId“钩子”相关线程的标识

钩子分为只供单个进程访问的钩子和共所有进程访问的钩子，对于前者可以将钩子过程放在安装钩子的同一线程内，此时SetWindowsHookEx()中的第三个参数为该线程的hInstance；于后者钩子过程应该放在单独的dll文件中，那么SetWindowsHookEx()中的第三个参数就是该动态链接库模块的句柄。钩子过程需要考虑是否把截获的消息传递给下一个钩”过程。如果要传递的话，就要调用CallNestHookEx()函数，其原型如下：

LRESULT CallNextHookEx(HHOOKhhk,int nCode,WPARAM wParam,LPARAM lParam,);

其中hhk为当前”钩子”的句柄，由SetWindowsHookEx()函数返回。NCode为传给”钩子”过程的事件代码。wParam和lParam分别是传给”钩子”过程的wParam值，其具体含义与”钩子”类型有关。

因为钩子会影响到系统性能，特别是对于全局钩子，因此在使用完后记得释放钩子，释放钩子的函数原型如下：

UnhookWindowsHookEx(HHOOKhhk);

函数成功返回TRUE，否则返回FALSE。

以上部分简单地介绍了下钩子的使用过程，接下来将结合我开发的windows窗口恢复软件来具体展示下如何使用钩子，因为这个软件还未做完，因此这里只部分介绍下其中的处理逻辑，但这部分已足以展示整个钩子的使用过程。

首要工作是需要选定安装钩子的类型，窗口关闭事件是通过将WM\_CLOSE消息发送到对应的窗口过程来实现的，因此在这里我选择了WH\_CALLWNDPROC这个钩子，WH\_CALLWNDPROC使你可以监视发送到窗口过程的消息，因此在窗口被真正销毁之前我通过截获WM\_CLOSE消息，并获取该消息的目标窗口的句柄，然后通过句柄获取该窗口的相关信息并记录以方便恢复。

显而易见我们对于每个窗口的关闭消息都需要记录，因此在这里我们需要将钩子设置为全局钩子，在建立工程的时候我们将钩子过程单独放在一个dll工程中，同时SetWindowsHookEx函数的最后一个参数设置为0，表示该钩子是一个全局钩子，该dll将注入到每一个进程进行监控。我们需要建立另一个工程来安装钩子，这个工程里的代码不涉及钩子的核心实现，只是一个实现钩子安装的过程以及一些界面功能的实现。

下图就是关闭一个应用程序窗口后钩子函数进行截获并获取其路径的截图：



