****二、Windows钩子程序的编制****

编制Windows的钩子程序，需要用到几个SDK中的API函数。下面列出这几个函数的原型及说明：

HHOOK SetWindowsHookEx( int idHook, HOOK\_PROC lpfn, HINSTANCE hMod,DWORD dwThreadID);

参数说明：   
idHook ：钩子的类型  
lpfn ：钩子处理函数地址  
hMod ：包含钩子函数的模块句柄  
dwThreadID ：钩子的监控线程

函数说明：  
函数将在系统中挂上一个由idHook指定类型的钩子，监控并处理相应的特定消息。

BOOL UnhookWindowsHookEx( HHOOK hhk );

函数说明：函数将撤销由hhk指定的钩子。

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

函数说明：函数将消息向下传递，下一个钩子处理将截获这一消息。

由于钩子的处理涉及到模块及进程间的数据地址问题，一般处理是把钩子整合到一个动态链接库（DLL）中，并设立一个全局数据共享数据段，以存贮一些全局变量，保留上次钩子消息事件发生时的状态。全局共享数据段可以用如下的格式定义：

#pragma data\_seg("PublicData")

HHOOK hhook=NULL; //全局共享数据

#pragma data\_seg()

在本文所附带的范例程序中，演示了如何编制一个鼠标钩子（WH\_MOUSE）程序。这个程序监视了Windows系统的鼠标消息，在监控期间，程序可以用户单击鼠标左键的次数。其它类型的钩子程序的编写过程与范例程序类似。

****三、范例程序的建立与代码分析****

正如上面所说的，建立钩子程序时需要把钩子处理整合到动态链接库中，所以例程中需要建立两个Project。

****1、建立钩子处理动态链接库：****

（1）选择MFC AppWizard(DLL)创建一个新Project，命名为"Spy";

（2）选择MFC Extension DLL类型

（3）创建一个新的头文件，命名为"Hook.h"，修改它的代码如下

extern "C" LRESULT CALLBACK MouseProc(int code,

WPARAM wParam,LPARAM lParam); //钩子处理函数

extern "C" BOOL WINAPI StartHook(); //启动钩子函数

extern "C" BOOL WINAPI StopHook(); //撤销钩子函数

extern "C" int WINAPI GetResult(); //取得鼠标单击次数的函数

（4）修改Spy.cpp文件代码如下（黑体部分为添加内容）

#include "stdafx.h"

#include <afxdllx.h>

#include "spyhook.h"

……//省略部分机器生成代码

#pragma data\_seg("PublicData") //定义全局数据段

HHOOK hhook=NULL; //钩子句柄

HINSTANCE pInstance=NULL; //钩子模块句柄

UINT MouseClick=0; //记录鼠标单击次数的变量

#pragma data\_seg()

……//省略部分机器生成代码

extern "C" int APIENTRY

DllMain(HINSTANCE hInstance, DWORD dwReason, LPVOID lpReserved)

{ if (dwReason == DLL\_PROCESS\_ATTACH)

{ ……//省略部分机器生成代码

new CDynLinkLibrary(SpyDLL);

pInstance=hInstance; //取得模块句柄

}

else if (dwReason == DLL\_PROCESS\_DETACH)

{ TRACE0("SPY.DLL Terminating!\n");

AfxTermExtensionModule(SpyDLL);

}

return 1;

}

extern "C" LRESULT CALLBACK MouseProc(int code,WPARAM wParam,

LPARAM lParam) //钩子处理函数

{ if (code < 0) //若code<0，直接调用CallNextHookEx返回

return CallNextHookEx(hhook, code, wParam, lParam);

if(wParam==WM\_LBUTTONDOWN)

{ MouseClick++; //记录鼠标单击次数

}

return CallNextHookEx(hhook, code, wParam,lParam);

}

extern "C" BOOL WINAPI StartHook() //启动钩子函数

{ hhook=SetWindowsHookEx(WH\_MOUSE,MouseProc,pInstance,0); //挂上钩子

if(hhook!=NULL)

return TRUE;

else return FALSE;

}

extern "C" BOOL WINAPI StopHook() //撤销钩子函数

{ return UnhookWindowsHookEx(hhook); //撤销钩子

}

extern "C" int WINAPI GetResult() //返回鼠标单击次数

{ return MouseClick;

}

（5）修改Spy.def文件如下

LIBRARY "SPY"

DEs criptION 'SPY Windows Dynamic Link Library'

EXPORTS

StartHook @1

StopHook @2

GetResult @3

（6）编译Project，生成Spy.dll文件和Spy.Lib文件

****2、建立使用钩子的应用程序****

生成一个单文档的可执行文件（EXE）的Project   
修改资源中的主菜单，增加一个菜单项"监控"，下有三个子菜单项，分别为"启动"，"撤销"，"取出"   
在Project中加入Spy.Lib文件和Hook.h文件   
分别修改"启动"，"撤销"，"取出"菜单项的Command响应函数如下：

#include "hook.h"

……//省略部分机器生成代码

void CMainFrame::OnStartSpy() //"启动"菜单项的响应函数

{ StartHook();

}

void CMainFrame::OnReleaseSpy() //"撤销"菜单项的响应函数

{ StopHook();

}

void CMainFrame::OnGet() //"取出"菜单项的响应函数

{ int Result=GetResult();

char buffer[40];

wsprintf(buffer,"在程序运行期间，你共单击鼠标%d次",Result);

::MessageBox(this->m\_hWnd,buffer,"Message",MB\_OK);

}

编译这个Project，并把Spy.dll放到生成的可执行文件的目录下，便可运行程序。运行时，选择"监控"菜单中的"启动"菜单项，钩子便开始工作，监视鼠标的活动情况；选择"撤销"菜单项，系统便撤销钩子；选择"取出"菜单项，程序便报告在监控期间，用户单击鼠标左键的次数。