文章编号:1005-6033(2006)06-0228-03 收稿日期:2005-11-24

基于 SDK 的软件设计方法

李永红,程耀瑜,隋海洋

(中北大学七系,山西太原,030051)

摘 要:对现在一些公司所提供的 SDK 软件包开发程序进行了拆分,清晰地剖析出所需开发的主干内容,从而能有效地提高了软件开发的效率。

关键词:SDK 软件包;API 函数;窗口函数

中图分类号:TP31 文献标识码:A

1 问题的提出

当我们在做软件的二次开发时,一些厂家所提供的软件开发包程序,往往是 Windows 环境下的 C 语言 API 编程,这些 SDK 示例,对于部分软件从业人员来说比较生疏,现在大部分编程从业者或初学者都知道 MFC,却对 SDK 不甚了解,当他们拿到 SDK 示例程序时常常会手足无措,笔者从实际工作中遇到的一个示例出发,对 SDK 示例程序的框架结构及主要函数做了介绍,从而帮助程序员快速地掌握 Windows 环境下的 C 语言 API 编程,从看似复杂的 SDK 示例程序中找到主干核心,从而节约开发时间,提高工作效率。

2 SDK 说明

SDK 就是 Software Development Kit 的缩写,中文意思就是"软件开发工具包"。这是一个覆盖面相当广泛的名词,辅助开发某一类软件的相关文档、范例和工具的集合都可以叫做"SDK"。我们只讨论广义 SDK 的一个子集,即开发 Windows 平台下的应用程序所使用的 SDK。为了认识 SDK,我们先来了解几个概念:API,DLL(动态链接库),导入库等。"API"也就是 Application Programming Interface,其实就是操作系统留给应用程序的一个调用接口,应用程序通过调用操作系统的 API 而使操作系统去执行应用程序的命令。DLL 即 Dynamic Link Library(动态链接库)。我们经常会看到一些 .dll 格式的文件,这些文件就是动态链接库文件,其实也是一种可执行文件格式。跟 .exe 文件不同的是 .dll 文件不能直接执行,他们通常由 .exe 在执行时装入,内含有一些资源以及可执行代码等。其实 Windows 的三大模块就是以 DLL 的形式提供的(Kernel32.dll,User32.dll,GDI32.dll),里面含有 API 函数的执行代码。为了使用 DLL 中的 API 函数,我们必须要有 API 函数的声明(.H)和其导入库(.LIB),函数的原型声明头文件是对函数进行函数声明的文件,导入库是为了在 DLL 中找到 API 的入口点而使用的。

为了使用 API 函数,就要有与 API 所对应的 .H 和 .LIB 文件,而 SDK 正是提供了一整套开发 Windows 应用程序所需的相关文件、范例和工具的"工具包"。由于 SDK 包含了使用 API 的必需资料,所以人们也常把仅使用 API 来编写 Windows 应用程序的开发方式叫做"SDK 编程"。而 API 和 SDK 是开发 Windows 应用程序所必需的东西,所以其他编程框架和类库都是建立在它们之上的,比如 VCL 和 MFC,虽然他们比起"SDK 编程"来有着更高的抽象度,但这丝毫不妨碍它们在需要的时候随时直接调用 API 函数。

3 SDK 与 MFC 编码时的区别

用 VC 编写 Windows 程序有两种:一是 Windwos C 方式(SDK),二是 C++方式:即对 SDK 函数进行包装,如 VC 的 MFC,BCB 的 OWL 等。SDK 编程就是直接调用 Windows 的 API 进行编程,平时人们常说"用 SDK 写程序"就是指用 Windows 的 API 函数来写程序,API 由成千上万个 API 函数组成。而 MFC 是 API 的封装,结合面向对象程序设计的继承性和多态性组成一个个的"类",共由 100 多个类组成。在实际使用中,MFC 比 SDK 方便。

SDK 与 MFC 编码时的区别为,SDK 编码时包括程序框架(比如消息处理器、主窗口消息回调函数等)在内的代码全为自己编制,但所利用的 API 等接口全来自 SDK 的头文件和库文件。有一种说法说"SDK 只是 MFC 的一个真子集"是错误的,因为有部分 API 函数 MFC 没有封装,不过这些函数你在 MFC 程序中都可以使用。MFC 主要封装的是界面、文件、Winlnet 和线程等函数。SDK 是基于 C 语言的,而 MFC 是基于 C++的,这是最根本的区别。MFC 除了封装 API,最重要的是它的体系结构,它所使用的 Doc/View 结构是 SDK 中没有的,这种架构是比较特殊的。尽管 Doc/View 结构不是微软的发明,但它是 MFC 的特色。MFC 最初是由微软设计,专供 VC++用的,但是 C++ Builder 也支持,C++ Builder 自己也有一个类库,叫 VCL。以前的 Borland C++使用 OWL 类库。微软开发 Windows 时使用的是 C 语言,所以最初设计 SDK 时并没有考虑 C++。

4 解读程序核心

通常编程人员手中的 SDK 程序是一个比较复杂,内容繁多,而且内部调用了很多文件和库函数的程序,为了提高效率,我们应该精简程序,找到程序中的主框架,指出需要我们引用的库函数所在位置和需要我们对程序进行扩展开发的内容部分。

Windows 程序的入口点总是 WinMain 函数,在 WinMain 函数中除了

第一作者简介:刘洪涛,男,1975 年生,现为太原理工大学计算机与软件学院 2003 级在读硕士研究生,山西省太原市迎泽西大街 79 号,030024.

Establishing the BBS System of the College Website by Using ASP+ACCESS

LIU Hong-tao

ABSTRACT: This paper introduces the general concept and scheme of designing the BBS system of the college website by adopting the JSP, and expounds in detail the key techniques of this system.

KEY WORDS: college websites; BBS system; JSP; ACCESS

上面所列程序主干内容之外,在调用 RegisterClass 函数为窗口注册一个窗口类之后,通常会有一些硬件初始化的库函数的调用,例如一些控制硬件的程序需要调用含有重要的硬件信息的.ini 文件。在用来实现 消息循环 "的 GetMessage 函数后会有一些控制硬件结束操作的库函数调用。

Windows 程序在创建一个窗口之前,必须调用 RegisterClass 函数为窗口注册一个窗口类。为调用 RegisterClass 函数,需要先定义一个 WNDCLASS 结构,然后初始化该结构的 10 个域,作为参数传给 RegisterClass。在 WNDCLASS 结构中最重要的两个域是第二个和最后一个。第二个域(lpfnWndProc)是所有基于该窗口类创建的窗口所使用的窗口函数的入口地址(例如:wndclass.lpfnWndProc=WndProc);最后一个域是该窗口类的名称,正是通过这个域将 窗口 "与其响应的 窗口类 "联系起来。窗口类定义了窗口的一般特征,而窗口与显示关系比较密切的一些细节特征尚未指定。因此,给予同一窗口类可以创建多个不完全相同的窗口,这一工作在 CreareWindow 函数中完成。把已经创建的窗口显示出来,调用 API 函数 ShowWindow(hwnd,nCmdShow);调用 UpdateWindow()之后,窗口就显示完毕。

接下来程序必须准备读入用户通过鼠标或键盘输入的信息。我们知道,Windows 是基于消息驱动机制的,Windows 会为当前运行中的每一个 Windows 程序维护一个 消息队列 "。在发生输入事件后,Windows 将事件转换为对应的 消息 ",并将消息放入相应程序的消息队列中。然后,程序从自己的消息队列中按一定的次序读取消息并发送给相应的窗口函数。代码:

```
while(GetMessage(&msg,NULL,0,0))
        {
              TranslateMessage(&msg);
              DispatchMessage(&msg);
         }
```

即用来实现 消息循环 "的。

GetMessage()从消息队列中取出一个消息,TranslateMessage(&msg);用于将 msg 结构传给 Windows 以进行一些键盘转换,DispatchMessage(&msg);再一次将 msg 结构传给 Windows。然后由 Windows 将该消息发送给适当的窗口函数,让后者对消息进行相应的处理。也就是说,是 Windows 调用窗口函数,而不是由程序员调用。在程序中,这个窗口函数就是 WndProc。由语句 wndclass.lpfnWndProc=WndProc 指定,在结束 DispatchMessage 调用的处理之后,流程又由 Windows 回到程序,并且接着从下一个 GetMessage 调用开始继续消息循环。

以上我们所讨论的从某种意义上说只是一些外围的准备性工作,真正实质性的工作在于对消息的处理,这些在窗口函数中实现。窗口函数中定义了对各种消息的响应动作,包括内容的显示、对用户输入的响应等。可以说,窗口函数是窗口最重要的部分,是窗口的精华所在。只要不和其他窗口函数的名称发生冲突,窗口函数的名称可以任取。在程序中,窗口函数即是 WndProc。一个窗口过程总是通过 RegisterClass 函数调用与特定的窗口类相关联;而通过 CreateWindow 函数调用,给予同一个窗口类可以创建多个窗口。

窗口函数负责处理窗口所响应的所有信息。对硬件的操作的消息也是相应地通过该窗口函数来实现的,通常响应窗口控件消息的函数会出现在窗口函数中的 case WM_COMMAND:分支消息响应下,该命令下的函数通常包含了我们所要用到的硬件实现功能的大部分内容。

值得注意的是,窗口函数所使用的调用方式为 CALLBACK。CALLBACK 函数是指由程序员定义、编程实现,而由 Windows 系统调用的函数。这正是 Windows 消息驱动的重要表现之一。

窗口函数接收的每一个消息都是用一个数值来标识的,也就是窗口函数的 message 参数。在 WINUSER.H 中为每一个消息参数定义了一个以 WM_("WindowsMessage,即 窗口消息 ")为前缀的标识符。

一般使用 switch 和 case 结构来确定窗口函数接收什么样的消息,以及如何适当地处理这些消息。窗口函数在处理完消息以后,必须返回 0 值。凡是窗口函数不予处理的消息应当传给名为 DefWindowProc 的函数加以处理,return DefWindowProc(hwnd,message,wParam,lParam)。请注意,凡是窗口函数不予处理的消息都应当交 Windows 默认的消息处理函数 DefWindowProc 加以处理。

Windows 程序的核心结构可归结如下:

```
#include <windows.h>
int WINAPI WinMain (HINSTANCE hInstance,HINSTANCE hPrevInstance,LPSTR lpCmdLine, int nCmdShow)
{
HWND hwnd;
MSG msg;
WNDCLASS wndclass;
wndclass.style= CS_HREDRAW |CS_VREDRAW;
wndclass.lpfnWndProc= WndProc;
wndclass.cbClsExtra= 0;
wndclass.cbWndExtra= 0;
wndclass.hInstance = hInstance;
wndclass.hIcon=LoadIcon( NULL, IDI_APPLICATION);
wndclass.hCursor=LoadCursor( NULL,IDC_ARROW );
wndclasshbrBackground=(HBRUSH)GetStockObject(WHITE_BRUSH);
        wndclass.lpszMenuName= NULL;
        wndclass.lpszClassName= szClassName;
        if( ! RegisterClass( &wndclass))
        {
        }
……
        hwnd = CreateWindow( );

        ShowWindow( hwnd, nCmdShow);
        UpdateWindow( hwnd);

        while( GetMessage( &msg, NULL, 0, 0))
        {
              TranslateMessage( &msg);
              DispatchMessage( &msg);
        }
……
        return msg.wParam;
}

LRESULT CALLBACK WndProc ( HWND hwnd, UINT message, WPARAM wParam, LPARAM lParam)
{
        switch( message)
        {
        case WM_CREATE:
case WM_TIMER:
case WM_COMMAND:
        case WM_PAINT:

        case WM_DESTROY:
              PostQuitMessage( 0);
              return 0;
        }
        return DefWindowProc( hwnd, message, wParam, lParam);
}
```

Windows.h 是最主要的头文件,它包含了许多其他的 Windows 头文件,其中比较重要的头文件如下:

WINDEF.H 用于定义基本数据类型;

WINBASE.H 用于包含内核函数;

文章编号:1005-6033(2006)06-0230-02　收稿日期:2005-11-25

如何突破系统守护的大门

关清占

(太原理工大学,山西太原,030024)

摘　要:介绍了一些消除密码进入操作系统的方法,包括清除 sam 文件、增加用户、使用专用程序破解、物理移除等方法。

关键词:密码设置;密码破解;sam 文件;操作系统
中图分类号:TP309　文献标识码:A

大家使用电脑经历最多的就是设置各种密码,为了确保自己的资料安全,电脑不被轻易进入,设置了重重密码,诸如开机密码、系统登录密码、软件保护密码等等,但是这样就安全了吗?本文以如何进入系统大门为例,为大家解开其中的秘密,操作系统就以 Windows2000/NT/XP 系统为例。

1　釜底抽薪——清除 sam 文件

由于 Windows2000/NT/XP 都是建立在 NT 技术核心基础上的,NT 系列的系统账户信息是存在%systemroot%\system32\config\sam 这个注册表文件里的,也就是说这个文件存放着账户的口令,每次登录的时候系统都和它进行自检匹配,符合就通过,不符合则不允许进入。如果系统里没有重要的账户,或者账户比较少,用删除%systemroot%\system32\config\sam 的方法是比较简单的。无论是对于 FAT32 还是 NTFS 格式分区都有效,Win2000 专业版和服务版都能通过进入,不过因为系统会还原为只有 administrator(密码为空)和 guest 两个账户,所以有些程序因为它们所依赖的账户丢失了,如 IIS 和 vmware 就不能启动了。

把 sam 文件改名或者删除,但是如何进入 DOS 访问系统分区呢?如果是 FAT32 和 FAT 分区,使用 Win98 启动盘就行了。如果是 NTFS 分区,可以使用 winternal 的 NTFS forDOS,NTFS for98 或者是支持 NTFS 的启动光盘,再或者挂到其他 Win2000,Linux 等机器上,不提倡重新安装系统。对于 WinXP 和 Win2003 来说删除 sam 文件会导致系统的致命缺陷而重新启动计算机。

2　暗度陈仓——增加用户

利用系统自带的 NET 命令。在 WindowsXP 中提供了"netuser"命令,该命令可以添加、修改用户账户信息,其语法格式为:

netuser[UserName[Password|*][options]][/domain]
netuser[UserName{Password|*}/add[options][/domain]
netuser[UserName[/delete][/domain]]

以恢复本地用户"tyutgqz"口令为例,来说明解决忘记登录密码的步骤:

(1)重新启动计算机,在启动画面出现后马上按下 F8 键,选择"带命令行的安全模式"。

(2)运行过程结束时,系统列出了系统超级用户"administrator"和本地用户"tyutgqz"的选择菜单,鼠标单击"administrator",进入命令行模式。

(3)键入命令:"netuserwinxpgqz123456/add",增加一个名为"winxpgqz"的用户名,该用户名密码为"123456";然后我们把该用户提升为管理员权限:"netlocalgroupadministratorswinxpgqz/add";这样"winxpgqz"就拥有和"administrator"一样的权限了。

(4)重新启动计算机,选择正常模式下登录新建用户 winxpgqz,用更改后的口令"123456"登录就可以了。

不过此法仅适用于采用 FAT32 分区安装,且用户账户不是汉字名称的 WinXP,经过实验对于采用 NTFS 格式的 WinXP 系统在安全模式登录时,也要输入登录口令,而不会像前者出现的点击进入命令行,这也体现了 NTFS 格式比 FAT 和 FAT32 格式的安全性吧。使用此方法有一定的局限性。

3　借力打力——专用程序破解

进入 Windows 管理员系统还有一个简单解决方法,就是使用

WINUSER.H 用于包含用户接口函数;
WINGDI.H 用于包含图形设备接口函数;
WINNT.H 用于定义支持 Unicode 编码的基本数据类型。

一般厂家提供的 SDK 程序的重要的控制函数都会在窗口函数中调用。在这部分核心部分中大部分的程序内容对我们来说是多余的,例如其中很多的 wsprintf()等,所以我们在读程序的时候可以将没用的部分去掉,使程序看起来更精炼,易读性更强。要实现在别的程序上调用厂家所提供的函数库函数需要事先做一些设置,如将.dll 文件加入所建工程中,并将与 dll 相对应的.h 文件和.lib 文件添加到你所创建的工程中。

参考文献

[1]　朱磊,周彬.Windows 下的 C/C++高级编程[M].北京:人民邮电出版社,2002.

(责任编辑:胡建平)

第一作者简介:李永红,男,1979 年 2 月生,现为中北大学信息工程系 2003 级硕士研究生,山西省太原市中北大学七系,030051.

A Method of Software Design Based on SDK

LI Yong-hong, CHENG Yao-yu, SUI Hai-yang

ABSTRACT: This paper makes resolution of the developing program of the SDK software package provided by some companies at present, and dissects the needed main contents clearly for the purpose of increasing the efficiency of the development of the software effectively.

KEY WORDS: SDK software package; API function; windows function