

# 编程实现为 Windows 95 加口令

张仁惠 (上海交通大学内燃动力与汽车工程系 200030)

**摘要:**本文介绍了一种通过编程实现为 Windows 95 加上口令的方法,可以达到限制非法用户使用 Windows 95 的目的,增强了微机数据的安全性。

**关键词:**口令 Windows 95 屏蔽功能键

Windows 95 是目前微机上广泛使用的操作系统,与 Windows NT 相比,其缺少安全性措施。为了增强 Windows 95 的安全性,可以为 Windows 95 加上口令。在 Windows 95 启动后立刻要求用户输入口令,若口令正确,则正常进入 Windows 95;否则,若口令输入三次以上仍不正确,则认为此用户为非法用户,口令程序自动关闭计算机。

## 1. 口令程序的要求

为了真正达到限制非法用户使用 Windows 95 的目的,Windows 95 的口令程序需要满足如下几个条件:

- 启动 Windows 95 即立即运行此口令程序,要求用户输入口令;
- 能够屏蔽所有的功能键,包括热启动组合键、ESC 键、Windows 95 专用键等所有的功能键;
- 屏蔽用户对桌面上的程序的操作,即用户不能点击桌面上的任何图标以启动其他应用程序;
- 隐藏 Windows 95 的任务栏,使用户无法点击“开始”菜单;
- 用户输入正确口令后,恢复功能键,显示任务栏及桌面。

## 2. 程序实现原理

本程序主要用到几个 Windows API 函数:

SystemParametersInfo、FindWindow、SetWindowPos、ExitWindowsEx。

这几个函数的具体参数请查有关 API 参考手册。SystemParametersInfo 函数用来实现对功能键的屏蔽及恢复,FindWindow 函数用来获得任务栏窗口的句柄,SetWindowPos 函数实现任务栏窗口的隐藏及显示,ExitWindowsEx 函数用来实现关机操作。

## 3. 口令程序的具体实现

笔者的程序是在 C++ Builder 3.0 环境下实现的,但其原理对于 Borland C++、Visucl C++ 或 Visual Basic

环境同样适用。

运行 C++ Builder 3,新建一个 project,命名为 password。

设置 Form1 的属性:BorderIcons = [];BorderStyle = bsNone; FormStyle = fsStayOnTop; WindowState = wsMaximized。这几条属性设置使 Form1 成为一个无标题条、无边框、运行时最大化的永远可见的窗口。这样就可以使用户无法关闭、移动此窗口,同时由于窗口最大化,使用户无法点击桌面上的图标。

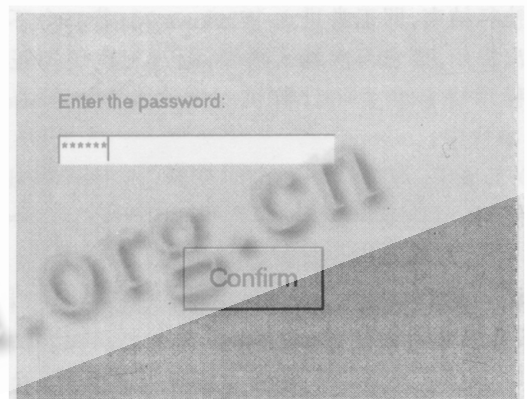


图 1 口令程序窗口

增加一个 Label 控件 Label1,其 Caption 设为“Enter the password:”。增加一个 Edit 控件 Edit1,设置其属性 PasswordChar = '\*',即用字符“\*”来代替显示输入的口令字符。再放置一个 Button 控件 Button1,其 Caption 设为“Confirm”。

至此,窗体的外观设计结束,下面为其添加代码。(具体代码见程序清单)

为了使口令程序能够屏蔽功能键并隐藏任务栏,首先预定义三个常数:

```
# define SPI _ ACTION 97
# define SPI _ HIDEWINDOW 0x80
# define SPI _ SHOWWINDOW 0x40
```

然后在 Form1 的 OnShow 事件中添加如下代码:

```
SystemParametersInfo ( SPI _ ACTION, true,
(PVOID)1, 1); //屏蔽键盘的所有功能键
```

```
HWND m _ wnd = FindWindow ( " Shell _ tray-
wnd", "" ); //获得任务栏窗口句柄
```

```
SetWindowPos ( m _ wnd, 0, 0, 0, 0, 0, SPI _
HIDEWINDOW); //隐藏任务栏窗口
```

同理在用户输入正确口令后,为使口令程序恢复功能键并显示任务栏,在 Form1 的 OnClose 事件中添加如下代码:

```
SystemParametersInfo ( API _ ACTION, false,
(PVOID)1, 1); //恢复被屏蔽的功能键
```

```
HWND m _ wnd = FindWindow ( " Shell _ tray-
wnd", "" ); //获得任务栏窗口句柄
```

```
SetWindowPos ( m _ wnd, 0, 0, 0, 0, 0, 0, SPI _
SHOWWINDOW); //显示任务栏窗口
```

下面的代码实现对口令的判断,若口令正确,则口令程序执行结束,即正常进入 Windows 95;若口令不正确,则给出提示,要求再次输入口令,超过三次仍不正确,则口令程序将自动关闭计算机。代码添加在 Button1 的 Click 事件中:

```
DWORD tmp; //声明一个临时变量,供 ExitWindowsEx
函数使用
```

```
times + + ; //输入口令的次数加 1
```

```
if(Edit1 ->Text == "zhangrenhui") //判断口令是否正
确,这里假定口令为"zhangrenhui"
```

```
{
Application -> MessageBox ( "OK, Pass!", "Infoma-
tion", MB _ OK);
```

```
Close(); //若口令正确则结束本程序
```

```
{
else
```

```
{
Application -> MessageBox ( " Incorrect password!
Try again.", "Information", MB _ OK);
```

```
if(times == 3) //若输入三次仍不正确,则关机
```

```
{
Application -> MessageBox("You have jnot right to
use this Windows 95! /n The computer will be shut
```

```
down.", "Information", MB _ OK);
```

```
ExitWindowsEx(EWX _ SHUTDOWN, tmp); //自
动关机
```

```
}
```

```
Edit1 -> SetFocus(); //使口令输入框获得输入焦点
```

```
}
```

编译运行,生成 password.exe, 口令程序编制完成

#### 4. 实现口令程序在 Windows 95 启动后立即执行

通过修改 Windows 95 注册表来实现。运行 regedit.exe, 找到键名:

```
_ _ HKEY _ LOCAL _ MACHINE/SOFTWARE/Mi-
crosoft/Windows/CurrentVersion/Run 在 Run 下新建一个
字符串项,使命为"password",其值设为 password.exe 的
路径,如笔者设为"d:/zrh/password.exe"。重新启动机
器后,则口令程序生效。
```

如果用户不知道口令,则无法使用 Windows 95,无法修改注册表,也就无法阻止口令程序的自动运行,因此也就无法绕开口令的输入。

#### 5. 源程序清单

(只附 Unit1.cpp 文件,其他文件略。)

```
// - - - - -
```

```
# include < vcl . h >
```

```
# pragma hdrstop
```

```
# include "Unit1 . h"
```

```
# define SPI _ ACTION 97
```

```
# define SPI _ HIDEWINDOW 0x80
```

```
# define SPI _ SHOWWINDOW 0x40
```

```
// - - - - -
```

```
# pragma package ( smart _ init)
```

```
# pragma resource " * . dfm"
```

```
TForm1 * Form1;
```

```
static int times; //times 变量用来记录输入口令的次数
```

```
// - - - - -
```

```
_ _fastcall TForm1 : TForm1 ( TComponent * Owner)
```

```
: TForm ( Owner)
```

```
{
```

```
}
```

```
// - - - - -
```

```
void _fastcall TForm1 : FormShow ( TObject * Sender)
```

```
{
```

```

SystemParametersInfo(SPI _ ACTION, true, (PVOID)1,
1); //屏蔽键盘的所有功能键
HWND m _ wnd = FindWindow("Shell _ traywnd", ""); //
获得任务栏窗口句柄
SetWindowPos(m _ wnd, 0, 0, 0, 0, 0, SPI _ HIDEWIN-
DOW); //隐藏任务栏窗口
times = 0; //设置 times 变量初值
}
//-----
void __fastcall TForm1::FormClose(TObject * Sender,
TCloseAction & Action) {
SystemParametersInfo(SPI _ ACTION, false, (PVOID)1,
1); //恢复被屏蔽的功能键
HWND m _ wnd = FindWindow("Shell _ traywnd", ""); //
获得任务栏窗口句柄
SetWindowPos(m _ wnd, 0, 0, 0, 0, 0, SPI _ SHOWWIN-
DOW); //显示任务栏窗口
}
//-----
void __fastcall TForm1::Button1Click(TObject * Sender)
{
DWORD tmp;
times ++ ;
if(Edit1 ->Text == "zhangrenhui") //判断口令是否正确

```

```

}
Application -> MessageBox("OK, Pass!", "Informa-
tion", MB _ OK);
Close(); //若口令正确则结束本程序
}
else
{
Application -> MessageBox("Incorrect password! Try
again.", "Information", MB _ OK);
if(times == 3) //若输入三次仍不正确,则关机
{
Application -> MessageBox("You have not right to
use this Windows 95! /n The computer will be shut
down.", "Information", MB _ OK);
ExitWindowsEx(EWX _ SHUTDOWN, tmp); //自动关机
}
Edit1 -> SetFocus();
}
//-----

```

### 参考文献

- [1] 跨越 C++ Builder, 李智慧等, 四川大学出版社, 1998.7

(来稿时间: 1998年11月)