

利用 VB 调用其他应用程序的技巧

孟 辉 (同济大学经济管理学院 200092)

在开发应用程序过程中,除了要根据具体需要开发自己的功能模块以外,可能还会用到一些其他的应用程序,如字处理程序 Notepad、画图工具 Paintbrush 等,在 VB 设计中,人们常常会用 Shell 函数进行调用,但这种方式有一个缺点,就是在重复调用时,同样的应用程序可能会重复运行,造成计算机资源的极大浪费,以 Notepad 为例,重复调用会同时运行多个 Notepad 文档。尽管我们可以用一个变量保存 Shell 运行程序的任务标识,但其他程序退出后,此变量不会发生相应变化,因此,我们不能跟踪其他应用程序是否在运行,为了克服这个缺点,笔者在实践中利用 WaitForSingleObject 函数监控应用程序的运行,收到了良好的效果。

WaitForSingleObject 是 Windows 中的一个函数,它的功能是检查指定对象的当前状态,若该对象的状态为发信号状态,则调用线程进入一种有效的等待状态。利用此函数和其他函数,我们可以在调用其他应用程序之前最小化调用界面,并强制它在其他应用程序运行过程中等待,直到其他程序退出后才能恢复,这样通过调用界面,用户只能一次调用一个程序,并且在程序退出后才能得到调用界面的控制权。

为了实现这一点,需要在调用界面的一个 Module 中定义一个过程,此过程由 CreateProcess、WaitForSingleObject 和 CloseHandle 函数组成,其作用是利用 CreateProcess 函数调用其他程序,利用 WaitForSingleObject 函数监控此程序的运行状态,利用 CloseHandle 函数结束程序。下面是 Module 中的具体代码,笔者对函数中用到的函数和有关的数据结构中的成员进行简要的描述。

Private Type STARTUPINFO 'CreateProcess 函数使用 Startupinfo 结构指定的主“窗口的特性

cb As Long '以字节为单位指定该结构的大小

lpReserved As String '保留

lpDesktop As String '若为非 Null,则该成员指出在其中启动过程的桌面名

lpTitle As String '若为 Null,用可执行文件的名称

代替窗口标题

dwX As Long '确定窗口位置

dwY As Long

dwXSize As Long '确定窗口宽度和高度

dwYSize As Long

dwXCountChars As Long '指定屏幕缓冲区的宽度和高度

dwYCountChars As Long

dwFillAttribute As Long '指定初始的文字和背景

dwFlags As Long '确定过程创建窗口时,是否用 Startup 结构中的某一成员

wShowWindow As Integer '窗口显示方式

cbReserved2 As Integer '保留,必须置为 0

lpReserved2 As Byte '保留,必须置为 Null

hStdInput As Long '指定过程的标准输入句柄

hStdOutput As Long '指定过程的标准输出句柄

hStdError As Long '指定过程的标准错误句柄

End Type

Type PROCESS_INFORMATION 'CreatProcess 函数使用关于一个新创建的过程及其原始

'线程的信息填充 PROCESS_INFORMATION

hProcess As Long '返回一个新创建的过程和句柄

hThread As Long '返回一个新创建的过程的原线程句柄

dwProcessId As Long '返回一个全局过程标识符,用来标识一个过程

dwThreadId As Long '返回一个全局线程标识符,用来标识一个线程

End Type

Public Const NORMAL-PRIORITY-CLASS = &H20 '定义一个优先级常量

Public Const INFINITE = &HFFFF '定义一个无限等待时间常量

```
Declare Function CreateProcess Lib "Kernel32" Alias "CreateProcessA" (ByVal lpApplicationName As Long
```

```
    ByVal lpCommandLine As String, ByVal lpProcessAttributes As Long, ByVal
```

```
    lpThreadAttributes As Long, ByVal bInheritHandles As Long,
```

```
    ByVal dwCreationFlags As Long, ByVal lpEnvironment As Long,
```

```
    ByVal lpCurrentDirectory As String, lpStartupInfo As STARTUPINFO,
```

```
    lpProcessInformation As PROCESS_INFORMATION) As Long
```

'CreateProcess 将创建一个新进程,此函数的具体参数定义如下:

'LpApplicationName 指定一个 Null 结束的字符串,字符串指定了要执行模块的全路径的文件名

'lpCommandLine 指定一个 Null 结束的字符串,字符串为将要执行的应用程序指定命令行

'lpProcessAttributes 指向一个 Security-Attributes 结构,此结构为所创建的进程指定了安全属性,若为 Null,则创建此进程时,用一个缺省的安全描述符

'lpThreadAttributes 指向一个 Security-Attributes 结构,此结构为所创建的主线程指定安全属性

'ByVal bInheritHandles 指出进程是否从调用进程中继承句柄

'dwCreationFlags 指定控制优先类和进程创建的附加标志,可以控制优先级

'ByVal lpEnvironment 指定一个用于此新进程的环境块,若为 Null 则新进程使用调用进程的环境

'lpCurrentDirectory 此字符串为新进程指定当前驱动器和目录

'lpStartupInfo 指向一个 StartInfo 数据结构,此结构指定如何显示此进程的主窗口

'lpProcessInformation 指向一个 Process-Information 数据结构,它接收有关新进程的标识信息

```
Declare Function WaitForSingleObject Lib "Kernel32" (ByVal hObject As Long, ByVal dwMilliseconds As Long) As
```

Long

'WaitForSingleObject 将检查 hObject 指向对象的当前状态,若它为不发信号状态,则调用线程进入'一种等待状态,在等待一个等待条件被满足时,此线程战胜极少的处理器时间,其中 hObject' 是 CreateProcess 函数返回的句柄,当过程结束时,此过程对象的状态为发信号状态;dwMilliseconds' 以毫秒为单位指定限定时间,或为 Infinite,则此函数的限时间为无限。

```
Declare Function CloseHandle Lib "Kernel32" (ByVal hObject As Long) As Long '关闭一个打开对象的句柄
```

```
Public Sub ExecStr (CmdStr $) '定义一个调用函数
```

```
    Dim myproc As PROCESS_INFORMATION
```

```
    Dim mystart As STARTUPINFO
```

```
    Dim myststr As String
```

```
    Dim ret, x As Double '定义返回值变量
```

```
    Dim As Long '定义返回值变量
```

```
    myststart.cb = Len(myststart) '初始化 STARTUPIN-
```

FO 结构

```
    ret = CreateProcess(0 &, CmdStr $, 0&, 0&, 1&, NORMAL-PRIORITY-CLASS, 0&, myststr, myststart, myproc) '启动程序
```

```
    x = WaitForSingleObject (myproc.hProcess, INFINITE) '等待应用程序结束
```

```
    y = CloseHandle(myproc.hProcess) '关闭句柄
```

```
End Sub
```

在定义完有关的函数和变量后,就可以在调用界面中利用菜单或按钮等调用其他程序了,下面是一个在 MainForm 中利用 Cmd1 按钮调用的例子:

```
Private Sub Cmd1_Click()
```

```
    MainFrm.WindowState = 1 '最小化
```

```
    ExceCmd("Notepad.exe")
```

```
    Mainfrm.WindowState = 0 '恢复窗口
```

```
End Sub
```

以上程序和代码均在 VB5.0 上运行通过。

(来稿时间:1998年6月)