

VB 应用程序中用户自定义动态链接库的关键技术

贾宏宇 (西安交通大学自动控制系 710049)

赵俊峰 (西安交通大学轴承所 710049)

摘要:本文结合作者 Visual Basic 应用程序实际开发经验,介绍了在 Visual Basic 应用程序中使用动态链接库特别是用户自定义的动态链接库时必须正确处理的几个关键技术问题,并给出了一个具体的应用程序实例。

关键词: Visual Basic(VB) 动态链接库(DLL) 编译 重载 压栈

一、动态链接库 VB 应用程序中使用的关键技术

要使 VB 应用程序能正确使用动态链接库中的函数,就必须使 VB 应用程序中所声明的函数名称(非 Alias 引出的别名)和动态链接库中输出的函数名称一致,实参和形参的数据类型一致,实参和形参在堆栈中的排列顺序一致。下面就针对这几个方面加以说明。

1. 函数重载引发的函数名称问题

C++ 与 C 的不同点之一在于函数的重载, C++ 的编译器为了能正确处理函数的重载问题,在生成应用程序(或动态链接库)时,会在用户定义的函数名称的后面加上用于表明函数调用时参数类型的特征字符,故实际生成的函数名称与用户定义的不一样,即使我们在程序中并未重载任何函数, C++ 编译器也会进行同样的处理。可以利用一些调试工具,如 BC++ 3.1 中的

TDump, 来查看这些被实际输出的函数名称,例如后面例子中的“ReadPort”函数以 C++ 风格编译时实际输出的函数名称为“@ReadPort \$ qi”,“WritePort”函数的实际输出函数为“@WritePort \$ qi”。而 C 编译器则不会对函数名称作类似的处理,实际输出的函数名称和用户定义的完全一致。出于对以后调用时方便的考虑,我们可以采用 C 风格的函数定义,即在 C++ 环境中定义函数时,前面加上 extern “C”即可,具体应用见后面的例子。

2. 函数名称前面的下划线问题

大多数的 C/C++ 编译器带有一个编译开关:是否在生成的函数名称前面加下划线。如果选择加下划线,则实际生成的函数名称同用户定义的函数名称相比,前面加了一个下划线字符“-”,导致实际输出的函数名称和用户定义的函数名称的不一致。如果不选这一编译开关,则实际生成的函数名称和用户定义的完全一致(C 风

格)。这一编译开关在 BC++ 3.1 的 IDE 中可以按照下列步骤进行选择:选择 Options 菜单下的 Compiler 菜单项,再选择 Advanced code generation * 菜单子项,在随后出现的对话框中,对 Generate underbars 复选框选择切换。

3. 参数调用约定

在 C/C++ 中,函数调用时的参数压栈形式有两种:C 方式或 Pascal 方式。C 方式以从右至左的顺序将参数压栈,调用者负责清除堆栈;Pascal 方式以从左至右的顺序将参数压栈,被调用者负责清除堆栈。参数的入栈顺序或堆栈的清除不正确,函数的调用肯定会失败。VB 应用程序调用动态链接库中的函数时遵循 Pascal 调用约定,所以为了能使 DLL 中的函数被 VB 应用程序正确调用,应当申明其为 Pascal 调用约定。

三、实例

下面的例子使用 BC++ 3.1 生成了一个端口读写的动态链接库,并在 VB 3.0 的应用程序中利用其中的函数进行了端口 340 的读写。

1. BC++ 源程序

```
① //DLLinVB.
CPP
#include <dos.h>
#include <windows.h>

extern "C"
{
int far -export PASCAL ReadPort(int PortNo);
void far -export PASCAL WritePort(int PortNo, int
Data);
}

int far -export PASCAL ReadPort(int PortNo)
{ return inport(PortNo); }

void far -export PASCAL WritePort(int PortNo, int
Data)
{ output(PortNo, Data); }

int FAR PASCAL LibMain(HINSTANCE, WORD,
WORD, LPSTR)
{ return 1; }
```

```
int FAR PASCAL WEP (int)
{ return 1; }
```

②;DLLinVB.DEF

```
LIBRARY DLLinVB
DESCRIPTION 'The Example of DLL Used in VB'
EXETYPE WINDOWS
CODE PRELOAD MOVEABLE DISCARDABLE
DATA PRELOAD MOVEABLE SINGLE
HEAPSIZE 1024
```

2. VB 3.0 源程序

①DLLinVB.BAS:

```
Declare Sub WritePort Lib "C:\DLLinVB\
DLLinVB.DLL" (ByVal PortNo As Integer, ByVal Value
As Integer)
```

```
Declare Function ReadPort Lib "C:\DLLinVB\
DLLinVB.DLL" (ByVal PortNo As Integer) As Integer
```

②DLLinVB.FRM:

```
Sub Command1-Click ()
```

```
Text1 = Format(ReadPort(&H340))
```

```
End Sub
```

```
Sub Command2-Click ()
```

```
WritePort &H340, Val(Text1)
```

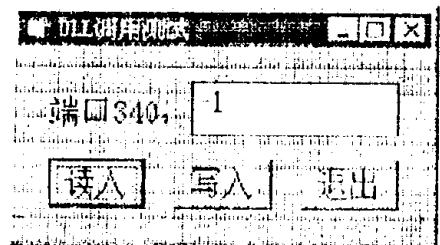
```
End Sub
```

```
Sub Command3-Click ()
```

```
End
```

```
End Sub
```

3. 运行结果



(来稿时间:1997年12月)