

动态链接库实现程序跨平台技术

魏会贤, 王培福, 韩仲祥, 陶晓燕

(空军工程大学 电讯工程学院, 陕西 西安 710077)

摘要: 讨论了 Win32 下用 C 语言创建动态库的原则; 阐述了 CB4、VC6 编译系统形成自定义动态库的方法; 提出了如何在 Authorware4、CB4、VB6 使用自定义动态库的思路。

关键词: 动态库; CB; Authorware

中图分类号: TP311.54 **文献标识码:** A **文章编号:** 1009-3516(2000)02-0052-04

动态链接库(Dynamic Link Libraries)是具有 .DLL 扩展名的文件, 是实现模块化程序设计的一种方法。它具有有效的程序重复使用、代码共享、代码分段、国际化和有效利用 Windows 资源等优点, 因此, 很多软件开发平台都支持动态库的应用。

本文较详细地阐述了如何用 CB4 编译系统通过自定义的动态库, 将 Powerpoint 可执行程序应用于 Authorware4 中实现程序跨平台使用的方法, 并给出了较完整的程序。

1 动态库的创建

1.1 Win32 动态库编程

动态库可以用 CB4 或 VC6 等编译系统进行创建, 为了使动态库能在不同平台中使用, 应该遵循 Win32 编程原则, 主要是:

- ① 动态库中应用的输入输出避免使用诸如“-export”、“-import”等关键字;
- ② 对数据类型, 必要时使用 Windows 数据类型, 以便在跨平台时被正确地转换;
- ③ 注意平台间的数据类型声明的对应关系。

1.2 输入输出函数

动态库中的函数主要有两种基本类型:

- ① 从动态库内调用的函数和作为动态库中类成员的函数;
- ② 从动态库处调用的函数。

从动态库内(Internal)调用的函数不需要特殊的管理, 它可以被动态库中其它的函数调用, 但不能从动态库之外调用。

另一类函数可以从动态库之外调用, 就是那些被动态库输出的函数, 它们既可以在动态库内部使用, 也可以被动态库外部的应用调用, 类似于类中 Public 型函数。

为了输出一个应用, 在建立动态库时, 保证头文件的正确是一个关键环节。

1.3 创建头文件

动态库中应用的输出, 必须使用关键字(变址符): --declspec(dllexport), 像这样: extern “C” void --declspec(dllexport) ShowPowerpoint(void); 为了在应用程序中调用这个应用, 要求输入此应用, 即创建动态库头文件时必须使用关键字(变址符)--declspec(dllimport), 像这样: extern “C” void --declspec(dllimport) ShowPowerpoint(void)。这里 extern “C” 被用来防止函数名字的错误, 任何时候使用动态库中独立的函数

时,就要用 extern “C”关键字,这将使函数可被任何应用调用。

由上可见,在建立动态库时,为了使某一个应用能被外部应用调用,要求输出这个应用;要使用此应用,即建立调用应用的可执行文件时,要求输入此应用,因此,就形成了两个头文件:一个应用于建立,一个应用于调用,这样就为建立和应用带来诸多不便。而使用本文给出的头文件,可以解决建立和应用调用时两个头文件带来的不便。头文件(假定头文件名为 Mydll.h)如下:

```
#ifndef _MYDLL_H
#define _MYDLL_H
#ifdef __MYDLL__
#define Dll_Func __declspec(dllimport) //win32 Keys
#else
#define Dll_Func __declspec(dllexport)
#endif
extern "C" {
    Dll_Func void __stdcall ShowPowerpoint(void);
}
#endif
```

1.4 创建程序文件及编译动态库

1.4.1 在 CB4 中实现 PowerPoint 的链接编译动态库的方法

运行 CB4 集成开发系统,单击 File|New 菜单,在 New 选项卡中单击 DLL 图标,单击 Ok,可产生动态库的程序文件,将此文件存为 Mydll.cpp。

单击 File|New 菜单,在 New 选项卡中,单击 Text 图标,单击 Ok,将此文件存为 Mydll.h。在头文件加入函数的声明及编译指令控制。这里,使用我们上述给出的完整头文件,头文件中函数 ShowPowerpoint 就可以从动态库外调用。

单击 File|NewForm 加入一个窗体,程序文件命名为 Mydll-Form1.cpp,头文件命名为 Mydll-Form1.h。在窗体上放入一个 OLE 控件,双击该控件,弹出插入对象窗口,点击“由文件创建”单选按钮,在文件栏写入要使用的 PowerPoint 文件或单击“浏览”按钮选定对象;“链接”复选框指定被选对象是以嵌入或链接方式与 OLE 控件相关,这里选定“链接”复选框。最后,单击“确定”按钮完成 OLE 控件与对象的相关。

以下为 Mydll.dll 的程序文件 Mydll.cpp 的主要部分:

```
#include <vcl.h>
#define __Mydll__
#include "Mydll.h" //Include the head file
#include "Mydll_Form1.h" //Include the form head file
#pragma hdrstop
Dll_Func void __stdcall ShowPowerpoint(void)
{
    Form1=new TForm1(NULL); //Pointer, Dynamic Show
    Form1->ShowModal(); //Modal Mode
    delete Form1; //Delete Pointer
}
```

//实现调用 ShowPowerpoint 函数产生 Form1 窗体时,Powerpoint 文件自动执行

```
void __fastcall TForm1::FormActivate(TObject *Sender)
{
    Form1->OleContainer1->Do Verb(); //Ole Primary Verb
}
```

最后,单击 Project|Build Mydll 就可以创建成链接有 Powerpoint 程序的 Mydll.dll 动态库了。

1.4.2 在 VC6 中创建及编译动态库的方法

使用 VC6 中 MFC APPWizard 也能创建动态链接库,且获得实现动态链接库基本功能的初始代码。

MFC 初始代码包括 C++ 程序文件、资源文件和工程文件。

创建动态链接库的过程有三步：

- ①根据 MFC APPWizard 提供的选项逐步定制应用程序；
- ②编译并链接这些初始代码；
- ③在向导创建的初始代码中添加自己的代码，以实现具体功能。

应该指出的是，动态库中头文件可以使用上面我们给出的头文件，以实现动态库的跨平台使用。

2 动态库的应用

2.1 在 CB4 中应用

CB4 中应用动态库，可以通过两种途径：静态应用和动态应用。

2.1.1 静态应用

生成一个工程，单击 Project|Add to Project 菜单，在文件名选项中输入 Mydll.lib 输入库，单击“打开”，将此输入库加入工程；在工程的头文件中包含动态库的头文件，既将 #include “Mydll.h”加入到工程的头文件中，然后就可以在事件中使用动态库中的函数 ShowPowerpoint 了。像这样：

```
ShowPowerpoint();
```

这里，Mydll.lib 是对应于 Mydll.dll 的输入库，它可以利用 CB4 中提供的可执行程序 Implib.exe 生成，方法是：

从 Windows 中打开一个 MS-DOS 窗口并转到 DLL 所在的目录，在 MS-DOS 提示符下输入：Implib Mydll.lib Mydll.dll。

这样就会生成了一个对应 Mydll.dll 名为 Mydll.lib 的输入库。

这里要特别指出的是，使用第三方提供的动态库静态用于 CB4 编译系统时，也要用 Implib.exe 以上面的方法生成输入库，而不能用其他如 VC 中生成输入库的程序来生成输入库。

静态使用时，一旦静态库被链入工程，将一直保留在内存中，直到应用程序结束。

2.1.2 动态应用

从动态装入的动态库中调用函数时，要使用 Windows API 函数 LoadLibrary、GetProcAddress、FreeLibrary。先建立一个指向动态库中的函数的指针，然后使用 GetProcAddress 取得函数的指针，最后释放动态库。如下为使用 Mydll.dll 中函数 ShowPowerpoint 的过程。

```
// Load the dll
HINSTANCE dllinstance=LoadLibrary("Mydll.dll"); //Get the handle
// Get the address of the function
ShowPowerpoint=(void _stdcall * )
                GetProcAddress(dllinstance, "ShowPowerpoint");
// Call the function
ShowPowerpoint();
// Unload the dll
FreeLibrary(dllinstance);
```

从动态装入的动态库中调用函数可以体现动态库的各种优点。

2.2 在 Authorware4 中应用

在 Authorware4 中使用动态库应在计算图标中操作。

进入 Authorware4 环境中，将主程序命名为 ApplicationMydll，在主流线上放置一个计算图标，然后双击计算图标，打开图标计算窗口。

单击 Window|Function 菜单，在 Category 下拉菜单中找到 ApplicationMydll 并单击，再单击 Load 按钮，将出现一个动态库选择窗口。

在动态库选择窗口的“文件名”栏输入要使用的动态库名（如 CB4 中生成的 Mydll.dll）或单击文件列表中该动态库图标，然后单击“打开”按钮，将出现一个此动态库的函数选择窗口。

在动态库的函数选择窗口的“Function Name”栏中输入要调用的函数名(如 ShowPowerpoint);在“Arguments”中输入此函数的参数类型(对应 ShowPowerpoint 的函数应为: void),如有多个参数时,参数之间要用“,”号隔开;在“Return”中输入函数返回的类型(对应 ShowPowerpoint 的函数应为: void),之后单击“Done”按钮。

在动态库选择窗口,单击“Paste”按钮,这时函数 ShowPowerpoint()就会出现在计算窗口,在这个函数中写入实参即可,本函数无实参。

最后,便可运行此程序,实现在 Authorware4 中通过动态库使用 Powerpoint 文件。

2.3 在 VB6 中应用

在 VB6 中使用动态库时,首先要在声明窗体变量的地方声明使用的动态库及函数,像这样:

```
Private Declare Function ShowPowerpoint Lib "Mydll.dll" () As Byte
```

然后,就可以在自定义函数或在其他事件中使用这个函数,下面是用自定义函数,来调用 Mydll.dll 中函数 ShowPowerpoint 的方法:

```
Private Static Function VB-ShowPowerpoint()  
    ShowPowerpoint()  
End Function
```

实际上,利用 VB 中自有的 OLE 控件,也可以实现 Powerpoint 文件的使用。这里只是一个使用自定义动态库实现程序跨平台使用的一个思路。

3 结束语

为满足软件开发者的不同需求,各种各样的软件开发平台相继被开发出来并得到广泛的应用,每一种软件开发平台都侧重于某些方面,但作为软件开发者所使用的某一种开发平台,有时须利用其他软件开发平台所开发出来的成果,也就是本文所提到的软件跨平台使用。利用动态链接库来实现是一种重要方法。本文给出的方法和思路,对其他支持动态库的开发平台利用动态库实现软件跨平台使用也具有一定的参考价值。

参 考 文 献

- [1] 刘柄文. Visual Basic 4.0 for Windows95 程序设计基础[M]. 北京:人民邮电出版社,1998.
- [2] 梁书斌. Visual C++ 6.0 高级编程[M]. 北京:清华大学出版社,1999.
- [3] 马 骁. 精通 Authorware 4.0[M]. 北京:宇航出版社,1998.
- [4] 龙启铭. Borland C++ Builder 高级编程技术[M]. 北京:北京希望电脑公司,1998.

Application of Programs in Different Platforms with Dynamic-Link Libraries

WEI Hui-xian, WANG Pei-fu, HAN Zhong-xiang, TAO Xiao-yan
(The Telecommunication Engineering Institute, AFEU., Xi'an 710077, China)

Abstract: This paper discusses the principle of writing Dynamic-Link Libraries in C language under Win32 surrounding. A method of forming Dynamic-Link Libraries with CB4, VC6 compiling system is presented. It also expounds the use of Dynamic-Link Libraries in program of Authorware and VB6.

Key words: dynamic-link libraries; CB; authorware