

用 DLL 连接 Delphi、VC++ 编制的不同程序

蔡一兵 (总参第六十一所八室 100039)

赵红宇 (解放军后勤管理学院 100039)

摘要:本文阐述了动态连接库的基本应用,给出了 VC++ 建立 DLL 及在 Delphi 中调用 VC++ 创建的 DLL 函数的编程实例,实现了通过 DLL 来连接 Delphi、VC++ 编制的不同应用程序的目的。

关键词:动态连接库 VC++ 编程 Delphi 编程

动态连接库(Dynamic Link Library)是一个库中函数的集合,是 Windows 操作系统的重要组成部分,Windows 使用 DLL 来支持公共函数调用。使用 DLL 的优点在于:(1)如果不同程序使用相同的 DLL,则只需将 DLL 在内存中装载一次,这样可节省系统内存。(2)对于一个大型的、不断更新的应用程序,可以提供 DLL 的不同版本,代替当前的 DLL。如果 DLL 中的子例程有相同的参数,可以用新版本的 DLL 运行程序,而不需重新编译应用程序。(3)使用 DLL 存储资源。如建立 DLL 的不同版本来保存不同语言的字符串,然后在运行时改变语言;或可以准备图标与位图库,然后在不同的应用程序中使用。

还有一个主要的优点在于: DLL 独立于编程语言。大多数 WINDOWS 编程环境都允许程序员调用存储在 DLL 中的子例程。这就意味着可以用 C++、VISUAL BASIC、POWER BUILD、JAVA、Delphi、汇编语言等来建立 DLL,然后从用不同语言编制的 Windows 应用程序中调用它。这样,通过 DLL 即可很方便地实现不同语言编制的 Windows 应用程序之间的连接问题。

笔者曾经遇到以下问题:主应用程序使用 Delphi 编制,设备的驱动程序使用 VC++ 编制,如何将不同编程环境下的应用程序连接起来?解决方法是:将 VC++ 编制的程序创建为 DLL,然后在 Delphi 编制的应用程序中调用该 DLL。下面用一实例来说明。

1. VC++ 创建 DLL

在 VC++ 中创建 DLL 项目最简单的方法是用 AppWizard 来创建一个新的项目,可以选择 DLL 项目类型。在启动 VC 之后,选择 FILE - NEW - MFC APPLICATION (DLL),然后选择 REGULAR DDL USING SHARED MFC DLL,最后选择 FINISH。这样,AppWizard 将自动创建一个包含基本要素的 DLL 框架,接下来只需手工加入自己的代码。

打开 *.CPP 文件,对需要编制的 DLL 函数声明如下:

```
char * WINAPI test1(char * input)
{ static char x[8];
  int i;
  for (i=0;i<8;i++)
  { x[i]=*(input+i); }
 strupr(x);
  char * d=x;
  return(d);
}
```

以上声明了一个 test1() 的函数,输入参数为一个字符指针,函数的返回结果为一个字符指针。该函数完成将输入的小写字母转换为大写字母的功能。

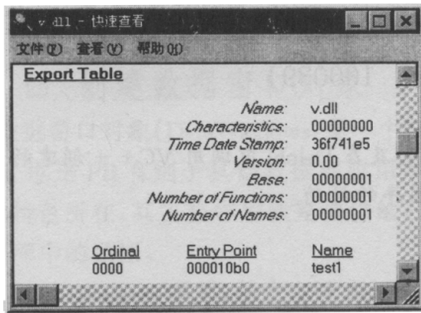
接下来打开对应的 *.DEF 文件,在 EXPORTS 下输入需要在 DLL 中显示的函数名。*.DEF 文件给每一个库以一个名字和一个描述,然后是函数列表。文件内容如下:

```
LIBRARY "v"
DESCRIPTION 'v Windows Dynamic Link Library'
EXPORTS
; Explicit exports can go here
test1
```

接下来在 VC 编译环境中点击 BUILD - BATCH BUILE,选择 WIN32 DEBUG,则生成调试版本的 *.DLL;选择 WIN32 RELEASE,则生成发行版本的 *.DLL。

如果正在运行 Windows 95 或 WIN NT,且安装了可执行文件的查看器(在系统中作为选项出现),只需要在资源管理器中选中 *.DLL,点击鼠标右键,并选择快速查看命令。显示出的查看器会列出对每个可执行文件

可以使用的一些低级技术信息。我们现在感兴趣的是 EXPORT TABLE 部分,如下图所示。函数的内部名称与它在 VC++ 源代码文件中的名称是一致的。



2. 在 Delphi 中调用 VC++ 创建的 DLL 函数

为了调用存储在 DLL 中的函数,需要对它们进行声明。一旦声明之后,就可以象调用其他任何函数一样,在应用程序代码中调用它了。有关调用语法并没有任何特别之处,它只是一般的函数或过程调用。

下面是 Delphi 中被调用函数的使用声明:

```
function test1(A:pchar) : pchar ; stdcall ; external 'v.dll' ;
```

声明主要包括两部分:(1)external 'v.dll'。该部分定义了 DLL 名称,名称应包括 DLL 扩展名;(2)test1(A:pchar)。该部分定义 DLL 函数自己的名称。

因为在 VC++ 中定义 test1()函数时,指定函数的返回结果为字符指针类型,所以我们对该函数在 Delphi 中进行声明时,必须把该函数的返回结果类型同样声明为字符指针类型。Delphi 中的字符指针类型为 pchar。

示范例程如下:

```
procedure TForm1.Button1Click(Sender: TObject);
var
get : pchar ; cinput : array[0..8] of Char ;
begin
cinput := 'abcdefgh'; {将 8 个小写字母赋与字符数组}
get := cinput;
edit1.text := test1(get); {调用 v.dll 中 test1()函数,将小写字母转化为大写字母}
```

end;

3. 加载 DLL

在应用程序开始运行时,它就开始寻找与它连接的 DLL 库,并把它映射到进程的内存空间。为找到 DLL,操作系统需要查找如下位置:(1)EXE 文件运行的目录;(2)进程所在的当前目录;(3)Windows 的系统目录;(4)PATH 环境变量所指定目录。

如果 DLL 没被找到,应用程序会显示一个对话框,说明没有找到 DLL,并显示寻找路径,最后进程结束。如果找到 DLL,它就被映射到进程的内存空间,直到进程结束。你的应用程序现在可以任意调用 DLL 中包含的函数。所以在发布你的应用程序时,一定要确保你的应用程序用到的 DLL 文件能被操作系统找到。

将 v.dll 文件放到示范例程的所在的子目录下,运行结果如下所示:



4. 小结

Windows 使用 DLL 来支持公共函数调用。DLL 为不同编程环境下的应用程序之间的连接提供了一个较好的解决方法。该方法具有节省内存,程序更新升级方便快捷等优点。本文提供了一个用 DLL 连接 Delphi、VC++ 编制的不同程序的范例。

参考文献

- [1] “DELPHI3 从入门到精通”, Marco Cantu 著,电子工业出版社,1998.4
- [2] “VISUAL C++ 5 开发人员指南”, David Bennett 等著,机械工业出版社,1998.7
- [3] “Windows NT 高级编程指南”, Jeffrey Richter 著,清华大学出版社,1994.12

(来稿时间:1999年3月)