

Windows 动态连接库的建立和使用

虞 歌 (杭州师范学院)

动态连接库(Dynamic Link Libraries, 简称 DLL)是一个在运行时进行连接的、可执行的代码或数据模块,是组成 Windows 系统的最重要的元素。建立动态连接库的出发点是让不同的 Windows 应用程序能共享库中的代码和数据,这样就可以充分利用系统资源,并且能够建立供所有程序使用的标准库。Windows 本身通过动态连接库为所有的 Windows 应用程序提供 Window API 函数和资源,Windows 本身使用的所有库也都是动态连接库。因此了解动态连接库的建立和使用非常重要。

一、什么是动态连接库

1. 动态连接库的定义

动态连接库可以视为可执行文件的特殊形式,如 Windows 中的三大动态连接库 KRNLX86.EXE、USER.EXE、GDI.EXE,但是动态连接库也可以包含程序以外的数据和资源。广义地说,在 Windows 中可以将动态连接库视为操作系统的扩展,主要目的是提供函数或资源让其他的程序使用,以便扩展系统的功能。

2. 动态连接和静态连接的区别

所谓静态连接,就是将程序中被调用但却不存在程序内的函数,通过连接程序查找函数库,直到找到函数所在的位置,最后再将该函数复制一份,放到 EXE 文件内的过程。动态连接的过程就有所不同,当然此时连接程序仍然要解决编译后目标文件中那些外部函数调用。通过查找函数库,以找到这些外部函数。但是连接程序不会象静态连接那样,进行函数代码复制工作,而是将输入库中对应于函数的有关“输入记录”复制到 EXE 文件中。“输入记录”记载着所需的函数在哪一个动态连接库中以及运行时如何调用等信息。当运行 EXE 文件时,Windows 会检查程序中是否存在任何的动态连接,对于找到的每一个动态连接,Windows 会装入这个动态连接

库,然后按“输入记录”提供的函数在动态连接库中的正确位置解决外部调用。因此,动态连接要分成两个阶段进行,首先要为连接程序在 EXE 文件中填入必要的信息,然后在每一次运行这个程序时,Windows 还要解决外部调用的寻址问题。

传统的 .LIB 文件只包含静态连接所需的函数代码,为程序编译时复制函数代码提供来源,一般称为目标函数库。输入函数库是一种特殊的函数库文件,通常也用 .LIB 作为扩展名,但其中没有任何程序代码。

函数库形态	使用时间	函数库名称(例)	函数名称(例)
目标函数库	连接期间	CWS.LIB	strlen
输入函数库	连接期间	IMPORT.LIB	TextOut
动态连接库	运行期间	GDI.EXE	TextOut

如上表所示,在 Borland C / C++ 中当应用程序调用 strlen 函数时,连接程序会从 CWS.LIB 库中读出目标代码,复制到应用程序的 EXE 文件中。如果应用程序调用 GDI 函数 TextOut,连接程序会由 IMPORT.LIB 文件读取该函数的有关信息到 EXE 文件中,等到运行时调用 TextOut 时,Windows 会利用连接时由 IMPORT.LIB 提供的信息,在 GDI.EXE 这个动态连接库中找到 TextOut 函数的实际位置并运行之。

3. 动态连接库和应用程序的区别

Windows 中所定义动态连接库在程序形式上和应用程序非常相近,它们具有相同的文件格式。和应用程序一样,Windows 也将动态连接库视为一个“执行模块”。那么,它们的差别到底在那里呢?通常动态连接库并不能直接运行,也不象一般的 Windows 程序必须有一个消息循环随时准备接收消息,动态连接库是不接收任何消息的。只有在别的程序调用了动态连接库中的某个函数以后,动态连接库才会发生作用。其次就是动态

连接库没有自己的堆栈,所以当应用程序调用动态连接库中的函数时,动态连接库可以使用自身的数据,但是必须使用调用它的应用程序的堆栈,作为暂时性的栖身之处。

4.使用动态连接库的优点

在开发软件时,使用动态连接方式来取代静态连接方式,一般有如下好处:应用程序编译连接生成的可执行文件比较小,应用程序所需使用的内存开销比较少,应用程序的修改更新比较容易。但是动态连接方式也并不是全无缺点,因为没有了动态连接库,应用程序就无法独立运行,所以动态连接库的存放位置一定要设置正确。

5.Windows 如何寻找动态连接库

要让 Windows 能够找到动态连接库,动态连接库必须放在下列这些目录中的任何一个里:当前目录,Windows 所在的目录(包含 WIN.COM 的目录),Windows 的系统目录(包含 KRNLX86.EXE 的目录),位于 PATH 环境变量中的目录。

6.查看动态连接库的内容

动态连接库和应用程序使用相同的文件格式,因此可以利用一些工具软件,如 Borland 公司的 TDUMP.EXE 或 Microsoft 公司的 EXEHDR.EXE 来查看动态连接库中内容。EXEHDR 运行后,一般会显示下面这些信息:旧的 MS-DOS 可执行文件头信息,新的可执行文件头信息,模块中程序段和数据段的列表,模块中定义的输出函数,每个段的复位信息。

二、开发和使用动态连接库

动态连接库可以用任何支持 Windows 开发的高级语言编写。如可以用 Turbo Pascal 建立动态连接库供自身和别的 Windows 程序使用;在 Turbo Pascal 程序中也可以使用在 Borland C/C++、Microsoft C/C++ 或别的能进行 Windows 编程的平台中建立的动态连接库。下面我们用 Turbo Pascal for Windows 来建立一个简单的动态连接库,看看动态连接库是如何工作的。

1.建立一个动态连接库

这个动态连接库只包含一个函数,用于求两个整数中的最大值。注意,下列程序应以 DEMODLL.PAS 名字保存文件并编译生成 DEMODLL.DLL。

```
{Library 表示程序是一个动态连接库}
Library DemoDLL;
{Export 说明函数将输出到其他模块中}
Function Max(x, y : Integer) : Integer; Export;
Begin
  If (x > y) then
    Max := x
  Else
    Max := y;
End;
{说明输出的函数并赋一索引值}
Exports Max Index 1;
Begin
End.
```

2.将动态连接库中的函数输入到一个单元中

为了在一个 Turbo Pascal 程序中使用动态连接库,首先必须建立一个单元用来获取对动态连接库中函数的访问权。注意,下列程序应以 DEMOUNIT.PAS 名字保存文件并编译生成 DEMOUNIT.TPU。

```
Unit DemoUnit;
Interface
Function Max(x, y : Integer) : Integer;
Implementation
{说明函数来自外部模块,通过引用 DLL 库中的索引值来标识需要的函数}
Function Max(x,y:Integer):Integer;External'Demo
DLL' Index 1;
End.
```

3.在应用程序中使用动态连接库中的函数

剩下的过程相当简单,只要在一个 Uses 语句中列出一个包含动态连接库信息的单元就可以了。

```
Program Demo;
Uses WinCrt, DemoUnit;
Var
  x, y : Integer;
Begin
  x := 45;
  y := 57;
  WriteLn('Integer number #1 is ', x);
  WriteLn('Integer number #2 is ', y);
  WriteLn('Max integer number is ', Max(x, y));
End.
```