

实现 Visual Basic 4.0 与 Fortran 90 的接口

张丽萍 (大连海事大学计算中心 116026)

摘要:本文介绍了如何通过 DLL 实现 Visual Basic 应用程序调用 Fortran 程序的技术,并给出了程序设计实例。

关键词:DLL 动态链接库 过程 声明

一、引言

Fortran 语言是世界上第一个被正式推广使用的高级语言,历时四十多年的历史但仍经久不衰,它始终是数值计算领域所使用的主要语言。随着时间的推移, Fortran 语言也不断地充实、完善。九十年代初期,采用新的美国国家标准的 Fortran 90 已推出使用,并提供了与当今流行的 Windows 环境的接口。众所周知,Windows 操作系统为微机用户提供了一个直观的、图形丰富的工作环境,Windows 应用程序提供的图形用户界面,使应用程序更易于实现和使用。目前在 Visual Basic、Visual C++、Borland C++ 等面向对象的程序设计语言中都可以实现 Fortran 过程调用,这样使得应用程序既保留了 Fortran 擅长于工程与科学计算的特点,又结合了 Windows 环境 GUI 界面的用户友好性,使二者相得益彰。

本文将结合实例介绍如何利用 DLL 技术实现 Visual Basic 应用程序调用 Fortran 程序。本文的实例在 Visual Basic 4.0 及 Lahey Fortran 90 Ver2.01 通过测试,能够正确运行。

二、DLL 概述

Windows 提供了数百个用户程序可以调用的完成诸如窗口、菜单、控制等功能进行画图等操作的函数。Windows 程序都调用了许多这些函数。用户编写的 Visual Basic 程序也可以直接调用它们。对所有 Windows 函数的调用,都是通过 Windows 的 DLL 机制来实现的。动态链接库 DLL(Dynamic Link Library)是一个加载到内存中并且根据需要与用户程序进行连接的代码库。Windows 有许多的 DLL,以提供 Windows 程序能够调用的全部函数。用户也可以用 C、Fortran、Pascal 等语言编写生成相应的 DLL 文件。

实现 Visual Basic 调用 Fortran 过程(子程序与函数),首先要创建该过程的 DLL 文件。Fortran 90 能够建立可以让 Visual Basic 应用程序调用的 DLL,这个 DLL 文件就是把一组将由 Visual Basic 调用的 Fortran 子程序和(/或)函数打包在一起的库文件,它自己不能运行,只

是处于一种可执行的方式下,当 Visual Basic 调用 Fortran 过程时,执行该 DLL 文件。

三、Visual Basic 调用 Fortran 90 过程

Visual Basic 应用程序调用 Fortran 子程序或函数可按以下几步进行:

1. 声明 Fortran 过程

在 Fortran 程序中,对将被调用的子程序和函数,要予以声明,即在程序中加入 DLL-EXPORT 语句,以标识这些子程序和函数在所生成的 DLL 文件中是可用的。DLL-EXPORT 语法如下:

DLL-EXPORT namelist

namelist 是一组用逗号分隔的子程序和(或)函数名,所列的子程序和函数应该是在同一程序范围之中的。DLL-EXPORT 语句用于 Fortran 程序中,用来指出 DLL 文件中所包含的可以调用的函数或子程序。

2. 创建 DLL 文件

编译已声明的 Fortran 程序,若编译无误则生成 DLL 文件。编译时要增加一些选项。语句如下:

LF90 source - filename - WIN - DLL - ML MSVB

LF90 是 Fortran 的编译命令,source - filename 是将要编译的 Fortran 程序名,该程序中包含若干已声明的子程序和函数。- WIN、- DLL、- ML、MSVB 是编译时增加的选项。- WIN 表示生成 Windows 的应用程序,- DLL 指明创建一个动态链接库,- ML 为产生与其他语言兼容的代码。例如 Fortran 程序 test.f90,经过编译后产生 test.dll 文件:

LF90 test.f90 - WIN - DLL - ML MSVB

有一点需要说明, Fortran 语句中大小写无关,这里的大写仅表示它们是 Fortran 的保留字。而在 Visual Basic 中大小写有关,将在后面说明。

将 Fortran 90 提供的 LF90WIOD.DLL 文件和编译后产生的 DLL 文件置于 Visual Basic 应用程序可以存取的路径上。如果未放在正确的目录下, Visual Basic 应用程序执行时将显示"Error loading DLL"错误信息。

3. 在 Visual Basic 应用程序中声明 Fortran 过程

启动 Visual Basic 4.0 后, 创建一个新的或打开已存在的 Project, 在它的 Form 或 Module 层的 General Declaration(通用声明)中, 就可以对被调用的 Fortran 过程进行说明。语句如下:

```
Private Declare Sub subroutine - name Lib "libname"
[(arglist)]
```

```
Private Declare Function function - name Lib "libname" [(arglist)]
```

Private Declare 为 Visual Basic 的声明语句, 对子程序或函数的调用, 分别用 Sub 和 Function 关键字表示。subroutine - name 和 function - name 与步骤 1 的 namelist 有关, 要注意它们的大小写一致。libname 要求给出外部子程序和函数所在 DLL 文件, 例如 test.dll。

4. 在 Visual Basic 应用程序中调用 DLL

使用 Visual Basic 中的 Call 语句实现调用 Fortran 过程。调用语句如下:

```
Call subroutine - name(arglist)
```

```
Call function - name(arglist)
```

5. 执行 Visual Basic 应用程序

在 Visual Basic 下, 可以按下 F5 来执行这个应用程序, 或按 F8 一步步地跟踪执行。执行过程中 DOS 屏幕与 Windows 窗口出现切换。

以上是 Visual Basic 应用程序调用 Fortran 过程的基本步骤。

四、实现 Visual Basic 调用 Fortran 的实例

为了突出说明 Fortran 的调用过程, 我们在程序功能上设计得很简单, 但实现的原理是一致的。程序说明如下:

TEST.F90 是 Fortran 程序, 其中 my-sub1 是一个子程序并被声明。子程序 my-sub1(a, b, c) 的主要功能是从 Visual Basic 接受三个参数, 改变其值并将结果返回 Visual Basic 调用者。

TESTVB.VBP 是 Visual Basic 应用程序, 包括一个 Form, 名为 TESTVB.FRM。Form 设计成有两个按钮: 一个按下为开始调用 Fortran 子程序, 另一个按下则停止调用。

程序代码如下:

TEST.F90 原代码:

```
subroutine my-sub1(a, b, c)
dll-export my-sub1 ! 说明被调用的子程序
real :: a, b, c
character (len=1) :: n
print
a = 2 * a
b = 2 * b
```

```
c = 2 * c
print *, " Press any key and Enter to continue..."
read (*, *) n
end
```

编译 test.f90 :

LF90 test.f90 - WIN - DLL - ML MSVB

编译后产生 test.dll 文件。

TESTVB.FRM 源代码:

```
Private Declare Sub my-sub Lib "test" (x As single,
y As Single, z As Any)
Private Sub Command1-Click()
Dim z As Single
x = 10
y = 30
z = 4
Hide
Call my-sub1(x, y, z)
Show
response = MsgBox("x is " + Str$(x) +
"." + "y is " + Str$(y) + "." + "z is "
+ Str$(z) + ".", vbOKonly + vbApplicationModal, "Lahey Fortran 90 ")
End Sub
Private Sub Command2-Click()
End
End Sub
Sub Form-Load()
End Sub
```

五、结束语

Visual Basic 的可视化开发环境, 让用户可以快速地开发出 Windows 应用程序。实现 Fortran 与 Visual Basic 的接口, 大大地改观了 Fortran 应用程序的图形界面, 为用户分析数据提供了一个清晰、直观、易于实现的画面。笔者在实现 Visual Basic 与 Fortran 接口的过程中, 对其数据传递、DOS 屏幕与 Windows 界面的切换也有所探讨, 限于版面, 不再赘述。

参考文献

- [1] Lahey Fortran 90 user guide
- [2] Lahey Fortran 90 Language Reference
- [3] Visual Basic Programming For Windows
- [4] <http://www.lahey.com>

(来稿时间: 1998 年 4 月)