

VC++与 Matlab 混合编程的快速实现

刘建伟

吉林大学朝阳校区电子科学与工程学院电子信息工程系 (长春 130026)

摘要: 许多工程软件需要用到复杂的数学算法。VC++能够形成各种用户界面, 并可以直接与系统及底层硬件交换数据。因此研究 VC++和 Matlab 的混合编程具有很大的实际意义。该文讨论了如何使用 Matlab 的 Compiler 将*.m 函数编译为动态链接库 DLL, 提供给 VC++ 调用的方法, 提供了一种 VC++与 Matlab 混合编程的快速实现。

关键词: Matlab VC++ MCC Compiler 动态链接库 DLL

Matlab 作为当今世界上应用最为广泛的数学软件, 具有非常强大的数值计算、数据分析处理、系统分析、图形显示甚至符号运算的功能。已经在如生物工程, 图像处理, 语音处理, 雷达探空, 声纳探水, 地震探地, 以及控制论, 系统论等各个领域得到广泛的应用。它是一个完整的数学平台, 在这个平台上, 用户只需寥寥数语就可以完成十分复杂的功能, 大大提高了工程分析计算、图像处理的效率。但是 Matlab 强大的功能只能在它所提供的平台上才能使用, 即用户必须在安装 Matlab 系统的机器上才能执行*.m 文件。这样当用户需要将在 Matlab 下已开发完毕的复杂算法应用到高级语言开发环境下时就带来了问题, 是将现成的东西集成高级语言开发的程序中呢? 还是用高级语言再重新实现一遍? 显然大家所期望的是减少工作量, 最好能将 Matlab 下开发好的程序或函数直接应用到高级语言开发的程序中, 这就是 Matlab 与高级语言的混合编程问题。

VC++是当前 Windows 下的主要编程语言之一, 它能方便地形成应用软件中所需要的各种用户界面和工具并直接与系统及底层硬件交换数据。用 VC++开发的系统具有容易维护升级、界面友好、代码效率高、执行速度快等一系列优点, 因此已成为目前应用软件开发中广泛采用的主要工具之一

因此, 如何将 Matlab 与 VC++进行有效的混合编程广受工程科研人员的关注。解决这个问题, 不仅能更好地发挥 Matlab 强大的功能, 还能快速地进行软件开发, 尤其是当软件开发中需要实现复杂的数学算法、图形处理时尤为迫切。研究 VC++与 Matlab 的混合编程具有很大的实际意义。

MATLAB 与 Visual C++结合的方法有多种, 但其实用程度与范围不同。常用的方法有: (1) 将 Matlab 程序编译成 C/C++源文件并嵌入 VC++; (2) 在 C / C++程序中利用 Matlab engine 调用 Matlab 函数; (3) 在 C/C++程序中直接使用 Matlab C/C++ Math Library; (4)将.m 文件编译成*.dll 文件嵌入到 VC++的程序中。基于实用性、易用性的考虑, 本文主要讨论第四种方法。另外随着 Matlab 版本的升级, 原来的 math library 已经并入了 Compiler 之中, 是否是 mathworks 比较推崇 DLL 的方式呢? 本文通过一个简单的例子来说明 Matlab 和 VC++混编的具体过程 (环境为 Matlab6.5&VC++6.0)。

1 Matlab 编译环境的设置

为了能够使用 Matlab 的 Compiler 将*.m 函数文件编译为动态链接库 DLL, 必须对 Matlab 的环境做出相应的配置。具体做法如下:

在 Matlab 的 Command Window 下输入命令 **mbuild -setup**, 并根据 Matlab 的提示选择合适的编译器, 笔者使用的是 Microsoft Visual C++. 过程如下

```
>> mbuild -setup
```

```
Please choose your compiler for building standalone MATLAB applications:
```

Would you like mbuild to locate installed compilers [y]/n?

Select a compiler:

[1] Lcc C version 2.4 in D:\MATLAB6P5\sys\lcc

[2] Microsoft Visual C/C++ version 7.0 in C:\Program Files\Microsoft Visual Studio .NET

[3] Microsoft Visual C/C++ version 6.0 in D:\Microsoft Visual Studio

[0] None

Compiler: 3

Please verify your choices:

Compiler: Microsoft Visual C/C++ 6.0

Location: D:\Microsoft Visual Studio

Are these correct?(y)/n):

然后,在 Command Window 中输入命令 **cd(prefdir);mccsavepath**;这是为了后面使用 Matlab 在 VC++ IDE 中的 Add-in 作准备。然后输入命令 **mex -setup**, 以类似的方法配置编译器。至此, Matlab 的配置完成。如有疑问, 可以参阅 Matlab 的帮助 Matlab Compiler\Getting Started\Installation and configuration 一节。

2 创建动态链接库 (*.DLL)

创建动态链接库主要有两种方法: 一种是使用 Matlab 为 VC++ IDE 提供的 Add-in。这种方法比较简单, 方便快捷, 只要在 VC++ 中创建工程的时候选择 Matlab Project Wizard, 并且在接下来的 Step 1 中的 Visual Matlab Application Type 选择 Shared M-DLL 就可以了。接下来就是添加 *.m 文件, 进行编译了。另一种方法就是使用 Matlab 的 **mcc** 命令将 *.m 文件编译为动态链接库 (*.DLL)。因为 Add-in 也是调用 Compiler 的命令 **mcc** 进行编译工作的, 而且有时候这个 Add-in 还会出现不能使用的情况, 因此这里主要讨论使用 **mcc** 命令的方法。

关于 **mcc** 有很多参数可以使用, 而且有多种用法, 具体的可以参见 Matlab 的帮助。这里我们使用命令 **mcc -B csglsharedlib:youlibname function1 funcion2 ...** 来创建我们所需要的动态链接库 DLL 文件。其中 参数 **-B** 表示使用的是 Bundle Files 作为参数, **csglsharedlib** 是 **-B** 的参数, 意思是生成使用 Matlab 图形库的 C 共享动态链接库, **youlibname** 是你想要生成的动态链接库 DLL 的文件名, 注意 **csglsharedlib** 与 **youlibname** 之间有一个冒号 “:” 连接, **function1** 为你想要加入到动态链接库 DLL 中的函数名, 如果有多个函数, 各个函数名用空格隔开。

在我们的例子中, 首先创建所需的 Matlab 函数文件 **myfun.m**。

```
function y=myfun(timeend)
```

```
t=0:0.001:timeend;
```

```
y=sin(2*pi*50*t);
```

```
plot(y);
```

然后使用命令 **mcc -B csglsharedlib:mylib myfun** 生成动态链接库 DLL。Matlab 会生成一系列文件, 其中 **mylib.h mylib.lib mylib.dll** 以及相同目录下的 **bin** 下的 **FigureMenuBar.fig** 和 **FigureToolBar.fig** 是我们在以后所需要使用的。

3 在 VC++中使用 Matlab 生成的动态链接库 (*.DLL)

为了能在 VC++中使用上面生成的*.DLL,首先要对 VC++的环境进行一些设置:

(1) 设置 Include 和 Library 目录

在 VC++ IDE 中选择 Tools->Options->Directories。

在 Show directoris for:中选择 Include files, 添加如下两个目录:

```
<Matlab>\extern\include\  
<Matlab>\extern\include\cpp
```

在 Show directoris for:中选择 Library files, 添加如下两个目录:

```
<Matlab>\extern\lib\win32\  
<Matlab>\extern\lib\win32\microsof\msvc6
```

这里假设<Matlab>为你的 Matlab 的安装目录。

这些操作只需要一次, VC++ IDE 就会自动记录。自动应用到每一个工程 (Project)。

(2) 工程(project)本身的一些设置

在 VC++ IDE 中选择 Project->Setting->C/C++

在 Category 中选择 Code Generation, 在 Use run-time library 中选择 Multithreaded DLL。

在 Category 中选择 Precompiled Headers,选择 Automatic use of precompiled headers, 并且在 Through header 下面添加 stdafx.h。

在 Category 中选择 Preprocessor, 在 preprocessor definitions 中添加 MSVC,MSWIND,IBMPC。

在 VC++ IDE 中选择 Project->Settings->Link

在 Categories 中选择 Input, 在 Ignore libraries:中填入: msvcrt.lib。

(3) 为工程添加相应的文件

把刚才生成的 mylib.h mylib.dll mylib.lib 和<Matlab>\extern\lib\win32 下的 libmatpm.lib, <Matlab>\extern\lib\win32\microsof\msvc6 下的所有*.lib 文件复制到 VC++工程的文件夹下, 并且用 add files to project ...添加的工程中。

接下来就可以使用刚才生成的动态链接库 mylib.dll 了。首先要在使用 DLL 中函数的源文件中加上头文件 mylib.h。在使用之前还需要对 mylib.dll 进注册, 使用完毕之后需要对其进行释放。注册使用函数 mylibInitialllize();释放使用函数 mylibTerminate()。函数名的规则就是 youlibname+Initialize(Terminate)。这两函数在 mylib.h 中可以找到。mylibInitialllize()不仅注册了刚才生成的 mylib.dll, 同时它也注册了 Matlab 本身的一些动态链接库 (*.DLL); 同理 mylibTerminate()不仅释放了刚才生成的 mylib.dll, 同时它也释放了 Matlab 本身的一些动态链接库 (*.DLL)。因此在注册了 mylib.dll 之后, 我们也可以使用 Matlab 的数学库中的函数, 例如 mlfabs(), mlfAssign()等等。

在使用 mylib.dll 的时候, 存在着一个 VC++和 mylib.dll 中的函数之间数据交互的问题。由于 mylib.dll 中的函数所接受的参数是 mxArray 型数据。因此我们要使用 Matlab 为我们提供的 External Interfaces/ API 函数来创建 mxArray 型数据,大量的 API 函数可以参见 Matlab 的帮助文件 Matlab\External Interfaces/API Reference\C MX-Functionsx 小节。还有一个需要注意的问题就是原来的函数名 myfun 变成了 mlfMyfun。可以在 mylib.h 中看到。

下面就是一个实例

```
//包含头文件
```

```
#include "mylib.h"
```

.

```

.
.
//注册动态链接库
mylibInitialize();
double x=0.04;
//创建 mxArray 型数据并对其赋值
mxArray *t;

t=mxCreateDoubleMatrix(1,1,mxREAL);
memcpy(mxGetPr(t),&x,sizeof(double));
//或者使用 t=mxCreateDoubleScalar(x)
//或者使用 Matlab 数学库中的函数 mlfAssign()等

//使用 mylib.dll 中的函数
mlfMyfun(t);
//销毁 t, 释放内存
mxDestroyArray(t);
//注销动态链接库
mylibTerminate();

```

4 程序的发布

在程序完成之后,就可以进行打包发布了。需要注意的问题就是发布的时候除了生成的 mylib.dll 和 FigureMenuBar.fig、FigureToolBar.fig, 还需要加上 Matlab 的一些动态链接库。这样发布的程序就可以完全脱离 Matlab 的环境而独立运行了。如何得到这些动态链接库呢? 将 <matlab>\extern\lib\win32\ 下的 mglinstaller.exe 解压缩, 会得到 mglarchive.exe, 再将 mglarchive.exe 解压缩, 在生成的目录 bin\win32\下就是程序独立运行所需要的动态链接库文件了。

从以上的讨论中可以清楚的看到, 使用 Matlab 生成动态链接库 (*.DLL) 实现 VC++ 和 Matlab 的混编的确以它的方便, 易用成为 VC++ 和 Matlab 混编的一种快速实现方法。另外, Matlab 的 Compiler 也具有一定的局限性: 原则上在混编的过程中只能使用 Matlab 数学库中的函数和图形库中的部分函数, mathworks 提供的文档上面讲到, Matlab 的图形库只有在用 mcc 生成可执行程序 (*.exe) 时才能引用。在 VC++ 的程序中直接使用 Matlab 图形库中的函数 (例如 mlfPlot) 是不支持的。另外, 在使用 mcc 生成 DLL 的时候, Matlab 图形库中的函数也不是全部都测试过。还有对于一些工具箱中的函数, 如果用到一些数学库之外的函数, 就有可能出现错误。即使编译通过, 在运行的时候也有可能出错。但是, 随着 Matlab 的不断升级, mathworks 推出了 Matlab COM Builder 创建 COM 组件, 对这种不兼容性有了一定的改进, 使得 Matlab 和高级语言的混合编程变得越来越实用, 越来越简单。

参考文献:

- [1] Matlab Compiler
- [2] Matlab COM Builder

作者简介:

刘建伟, 男, 生于 1981 年, 河南省郑州市人, 在校本科生。