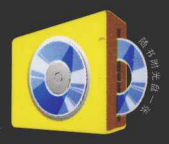
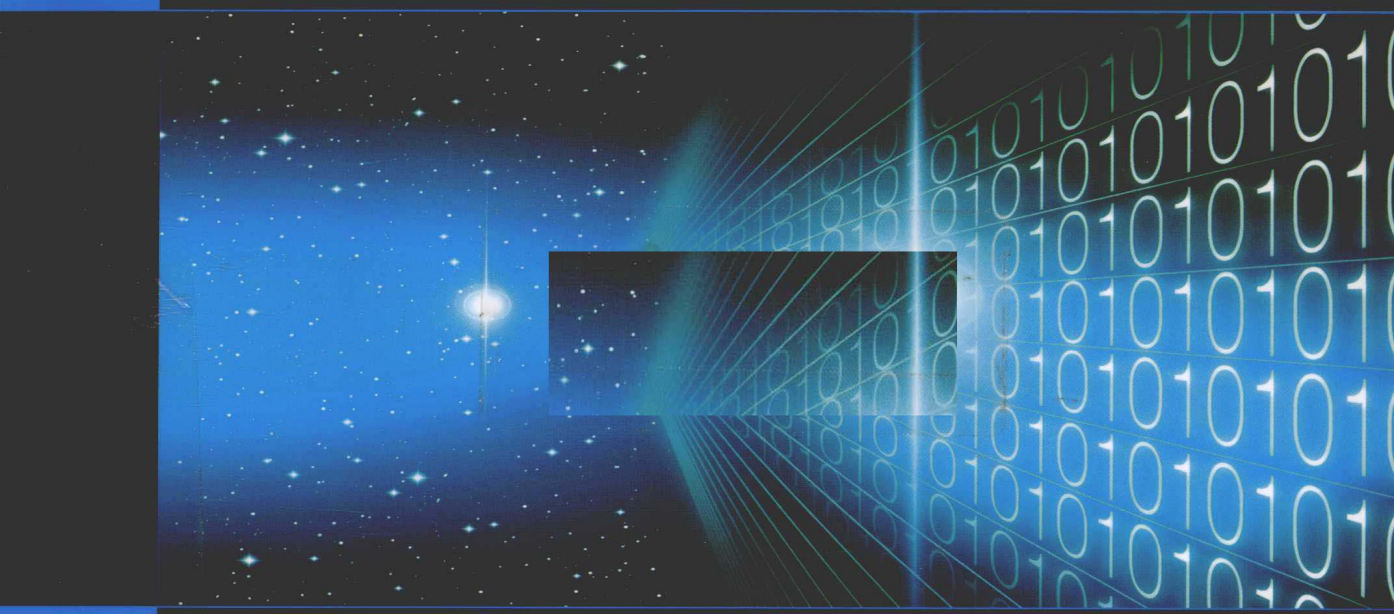
MATLAB
BIANYI CHENGXU
HE WAIBU JIEKOU



MATLAB 编译程序和外部接口

董振海　编著



國防工業出版社
National Defense Industry Press

MATLAB 编译程序和外部接口

董振海　编著

国防工业出版社

·北京·

前　言

MATLAB 编译程序是一个不小的题目，且是一个不可缺少的工具，为什么就没有一本完整而系统的关于它的书呢？我想做，我想让还不十分了解编译程序的人们知道编译的全过程，知道它所包含的所有功能以及编译程序外延的事情，以便他们可以根据不同的 MATLAB 源程序设计个性化的编译过程。同时我也在关注着 MATLAB 与 VB、C/C + + 等混合编程的问题，即 MATLAB 的外部接口。将 MATLAB 编译程序与外部接口放在一本书中，作为姊妹篇，这恐怕是一种天然的渊源。因为编译程序本身具有连接外部程序的功能，利用编译程序就可以直接将 MATLAB 与 C/C + + 程序连接在一起，形成独立应用程序。

那么你知道为什么 MATLAB 与 C 或 C + + 有天然的接口吗？

答案是：将 MATLAB 源程序编译成目标程序，分成两步。第一步编译 MATLAB 源程序成中间语言，即 C 或 C + + 代码；第二步将 C 或 C + + 代码编译成目标代码。

你知道服务器程序与客户程序不是网络概念，而是不同语言的程序连接时所扮演的“一仆一主”角色吗？你知道使用 MATLAB 的语法形式就能在 MATLAB 中直接操作 Java 的类和对象是为什么吗？你知道在 MATLAB 与 VB 的混合程序中是 MATLAB 程序控制 VB 程序，还是 VB 程序控制 MATLAB 程序呢？你知道在不打开 Excel 的情况下，用 MATLAB 可以建立工作簿、添加工作表、从工作表读写数据吗？这些答案尽在本书中。这些也正彰显出 MATLAB 外部接口的魅力。

MATLAB 这几年在中国的应用和普及日益扩展，它非常强的计算功能和适用于多学科、多领域的优势，是其他的编程语言所不能比拟的。将 MATLAB 作为计算引擎，再利用其他语言的长处写适当的子程序，或者使用已有的其他语言程序作为辅助，这岂不是扩大了 MATLAB 的能力！MATLAB 的外部接口就是给它增加了几只有力的“臂膀”。

这本书的特色可以概括为：内容全面，阐述细致，实例众多，理论与实际相结合。

本书中绝大部分的例子都经实实在在的编译、运行。编译、运行的过程和结果以及图形都尽量完整。对于实际编译和运行中，原始资料叙述不详和错误的地方，都加以说明和纠正。运行中出现的问题及改正的方法，也都有描述。力求让别人多从实例中更好、更快地学到东西。程序例子中用到的函数，其参数和用法都一一详细说明，对读懂程序非常有用。

这本书开门见山直接切入主题，没有专门的章节写 MATLAB 的基础内容。书中对个别用到的 MATLAB 编程知识，做了简单介绍，这是远远不够的。有关 MATLAB 编程的诸多问题，请参看我所编写的《精通 MATLAB 7 编程与数据库应用》。

受知识和经历的局限，书中难免有错误和不当之处，请读者朋友予以批评指正。

编　者

2010 年 9 月

目 录

上篇 MATLAB 编译程序

下篇 MATLAB 外部接口

上篇　MATLAB 编译程序

引　言

有人问我：

编译程序（有的人称为编译器）有什么用？

我立即回答：

能将编程语言写成的源程序编译成可以执行的目标程序。

这我知道。除此之外它还能做什么？

我明白了，他想问的不是“编译”的笼统概念，而是想知道有关 MATLAB 编译程序的诸多事情。

于是，我告诉他：

MATLAB 编译程序有很多选项，用不同的选项主要可以做到：

（1）生成独立应用程序，或别的软件组件（如共享库、COM 对象等）。

（2）可选择不同的中间语言的编译程序（如 C 或C ++ 编译程序）。

（3）可以输出追踪和查错信息，便于改正源程序的错误。

（4）可以只生成中间语言（C 或 C ++）程序，或者连接需要的库文件做成完整的独立应用程序。

（5）输出整个的编译信息，用于查找编译过程的问题。

（6）选择最终的输出，避免不必要的资源耗费。

等等。

当然，这仅仅是编译过程（或编译程序本身）所要完成的工作。还有编译程序外延的一些事情：

（1）如何将目标程序部署在没有安装 MATLAB 的机器上，需要准备哪些文件。

（2）怎样安装 MATLAB 编译程序，怎样安装 C 或 C ++ 编译程序。

MATLAB 编译程序版本 4（R14）适用于 MATLAB 7.0，MATLAB 编译程序版本 4 兼容以前的版本。以前的版本能够编译的 M 文件，这个版本也可以编译。

编译程序版本 4 生成的 API（应用编程接口）与以前的版本生成的不同。如果要开发软件组件，需要调整为新的 API。

MATLAB 编译程序版本 4 能够将 M 文件作为输入，生成独立应用程序或软件组件。生成的这些应用程序和软件组件只能用于特定的平台。MATLAB 编译程序能够生成如下类型的应用程序或软件组件：

1)独立的应用程序

独立的应用程序运行时不需要 MATLAB,即使 MATLAB 没有安装在用户的系统上,它们也能够运行。

2)C 和 C ++共享库(在 Microsoft Windows 中叫动态链接库(DLL))

即使 MATLAB 没有安装在用户的系统上,它们也能被使用。

3)Excel add-ins

需要 MATLAB Builder for Excel。

4)COM objects

需要 MATLAB Builder for COM。

MATLAB 编译程序支持所有的 MATLAB 语言的功能,包括对象。

MATLAB 解释程序与编译程序的区别:MATLAB 解释程序是 MATLAB 的一个应用程序,它接受 MATLAB 命令,执行 M 和 MEX 文件。当你使用 MATLAB,在命令窗口写入命令或运行函数时,就在使用 MATLAB 的解释程序。它解释命令,执行命令赋予的功能,有时将结果输出在命令窗口。

MATLAB 编译程序是一个单独的产品,它接受 M 文件作为输入,且生成独立的应用程序或软件组件。调用 MATLAB 编译程序,用 mcc 命令。

第1章　MATLAB编译程序的有关命令、附注函数

1.1　编译程序的有关命令

此一节有四个命令：mcc，buildmcr，mbuild，isdeployed。

1.1.1　mcc

mcc调用MATLAB编译程序。

1. 语法形式

```
mcc[-options] mfile1 [mfile2 ... mfileN] [C/C++file1 ... C/C++fileN]
```

说明：

mcc是调用MATLAB编译程序的命令，能够从MATLAB命令提示符、DOS命令行或UNIX命令行发出这个命令。

mcc为MATLAB环境之外准备部署的M文件。用C或C++生成封装文件，并选择性地建立独立二进制文件。写结果文件到当前目录(此为默认选择)。

如果在命令行指定了多于一个的M文件，编译程序为每个M文件生成一个C或C++函数。如果指定了C或目标文件，它们和生成的C文件一起被传送给mbuild。

2. 参数说明

-options	选项，皆以"-"为先导，可选参数。注意，同一个字母的大小写是不同的选项。
mfileN	被编译的M文件，1到N个。
C/C++fileN	被编译的C/C++文件，1到N个。这些文件与编译生成的C/C++文件一起被传送给mbuild。

3. 命令选项

1)-a

加一个文件到CTF档案。如：

```
-a filename
```

将指定的文件filename直接加到CTF档案，允许多个-a选项。编译程序在MATLAB路径寻找这些文件，因此，指定完全的路径名是可以的。这些文件不被传送到mbuild，因此可包含数据文件。

2)-b

生成Excel兼容的公式函数(Formula Function)。

生成一个VB文件(.bas)，文件包含Microsoft Excel公式函数与编译程序生成的COM对象的接口。当引入到工作簿VB代码时，代码允许这个MATLAB函数作为单元格公式

函数。要求 MATLAB 的 Excel 生成器。

3)-B 指定捆绑文件

用指定捆绑文件 filename 的内容代替 mcc 命令行的文件本身,形式如下:

```
-B filename[:<a1>,<a2>,...,<an>]
```

例如:　　mcc -B 'ccom:addin,addin,1.0' main.m

捆绑文件 filename 应当只含有 mcc 命令行选项和相应的参数,以及别的文件名。这个文件可以包含另外的-B 选项。捆绑文件能够包含接受名字和版本号的编译程序选项的替换参数。(参见表 5.2。)

4)-c 只生成 C 代码

与宏选项一起使用时,生成 C 代码但不调用 mbuild,即不生成独立应用程序。这个选项等价于放在 mcc 命令行末尾的-T codegen。

5)-d 输出指定目录

输出到一个指定目录,用

```
-d directory
```

将来自编译程序的输出文件放到-d 选项指定的目录。

6)-f 指定选项文件

由指定选项文件代替默认选项文件,形如:

```
-f filename
```

调用 mbuild 的时候,指定 filename 作为选项文件。这个选项允许对 MATLAB 编译程序的不同需求使用不同的 ANSI(美国国家标准学会)编译程序。选项的指定文件直接传送给 mbuild。

注:建议使用 mbuild-setup,预先设置选项文件,这样可以不用-f。

7)-g 生成追踪信息

包含对封装的查错符号信息。

8)-G 仅用于查错

选项使 mbuild 用适当的 C/C ++ 编译程序的查错选项调用 C/C ++ 编译程序。如果想用查错程序对独立应用程序查错,应当指定-G。

9)-I 加目录到路径

每个-I 选项加一个目录到当前搜索路径的末尾,例如:

```
-I <directory1> -I <directory2>
```

将建立搜索路径,以便为寻找 M 文件首先搜索 directory1,再搜索 directory2。在独立编译中 MATLAB 路径不可用时,这个选项很重要。

10)-l 生成函数库

建立函数库的宏。这个选项对命令行的每个 M 文件生成一个库封装函数,并调用 C 编译程序建立一个共享库。库名是命令行的第一个 M 文件名。这个宏等价于

```
-W lib:<string> -T link:lib
```

11)-m 生成独立应用程序

也是一个宏,产生独立应用程序。它等价于

```
-W main -T link:exe
```

12)-M 直接传送

指定编译时选项,用

```
-M string
```

它直接传送 string 到 mbuild。这为指定编译时选项提供了有用的方法。例如:

```
-M "-Dmacro=value"
```

注:如果出现多个-M 选项,只使用最右边的一个。

13)-N 清除路径

有效清除所有目录的路径,除下列核心目录外(这个列表有时会有变化):

```
<matlabroot>/toolbox/matlab
<matlabroot>/toolbox/local
<matlabroot>/toolbox/compiler
```

编译时出现在 MATLAB 路径中的上述目录的所有子目录也在不清除之列。在命令行使用选项-N,也允许替换原始路径的目录,而保持被包含目录的相对顺序,当然,也包括出现在原始路径中被包含目录的所有子目录。

14)-o 指定可执行程序

指定最后的可执行程序的名字和位置(仅限独立应用程序),以这样的形式:

```
-o outputfile
```

指定编译程序的最后可执行输出文件的名字。有些情况下,要对指定的名字加扩展名(例如,对 Windows 独立应用程序要加.exe)。

15)-p 加目录到路径

加目录到编译路径,按照在 MATLAB 路径中寻找的顺序加到适当的位置。

```
-p directory
```

这里的 directory 是被加的目录。如果 directory 不是绝对路径,就认为它是在当前工作目录下。加这些目录的规则是:

(1)如果-p 后面的目录是在原始 MATLAB 路径中,那么这个目录以及它的所有出现在原始路径中的子目录被加到编译路径,并且按照在 MATLAB 路径中寻找的顺序。

(2)如果-p 后面的目录不在原始 MATLAB 路径中,这个目录不被加到编译路径(但可以使用-I 选项加它)。

如果-I 选项能够被用来加一个目录(要求-N 选项),且这个目录已经在 MATLAB 路径中,它将被加到编译路径(按照在 MATLAB 路径中寻找的顺序),就像使用-p 选项一样。否则,这个目录被加到编译路径的头上,就像正常使用-I 选项一样。

16)-R 运行时

提供 MCR 运行时选项,形如

```
-R option
```

提供下列运行时选项,见表 1.1。

表 1.1 运行时选项

选项	描 述
-nojvm	不使用 Java Virtual Machine (JVM)
-nojit	不使用 MATLAB JIT(二进制代码的生成用来加速 M 文件执行)

注:-R 选项仅仅用于独立应用程序。为了在别的 MATLAB 编译中代替 MCR 选项,使用 clInitializeApplication 和 mclTerminateApplication 函数(更详细的信息参看 8.1.3 节)。

下面是有效使用的-R 选项:

```
mcc -m -R "-nojvm -nojit" -v foo.m
mcc -m -R "-nojvm" -v -R "-nojit" foo.m
mcc -m -R -nojvm -R -nojit foo.m
mcc -m -R -nojvm -v foo.m
mcc -m -R -nojvm -R -nojit foo.m
```

注:nojvm 或 nojit 前面的横线必不可少。

而这个例子是非法的:

```
mcc -m -R -nojvm -nojit foo.m
```

17)-T 指定目标程序级

指定输出的目标程序级和类型,用如下形式:

```
-T target
```

定义输出类型。有效的 target 值见表 1.2。

表 1.2 有效的 target 值

target 值	描 述
codegen	生成 C/C ++ 封装文件,默认值为 codegen
compile:exe	具有 codegen 的功能,再加上编译 C/C ++ 文件成为适合于连接到独立可执行程序的目标形式
compile:lib	具有 codegen 的功能,再加上编译 C/C ++ 文件成为适合于连接到共享库/ DLL 的目标形式
link:exe	与 compile:exe 一样,还能连接目标文件到独立可执行程序
link:lib	与 compile:lib 一样,还能连接目标文件到共享库/DLL
注:exe 使用 mbuild 生成可执行程序,lib 使用 mbuild 生成共享库。	

18)-v(Verbose)编译信息

显示编译阶段的信息,包括:

(1)编译程序的版本号。

(2)被处理的源文件名。

(3)被建立的输出文件名。

(4)mbuild 的调用。

-v 传送它的选项给 mbuild,并显示有关 mbuild 的信息。

19)-w 警告信息

显示警告信息,用语法形式

```
-w option[:<msg>]
```

控制警告信息的显示。表 1.3 列出了有效的警告选项。

表 1.3 有效的警告选项

句法形式	描 述
-w list	形成一个表,表中是使用了 nable, disable 和 error 选项的警告信息字串(<string>)。参考"第 10 章错误和警告信息"