Microsoft C 5.0

Quick C 使 用 说 明

译 校 者 序 言

翻 译：宗丽苹 吴 倩 邦继明 鲁 倩 王 越 徐砚祥 陈 林 彭怀宇
王淑平 何 骏 郭瑞阳 萧燕林

Microsoft C 5.0优化编译语言系统是美国87年国家标准，具有效率高、功能强的特点。为了迎合广大热心者的要求，配合国内用户高水平开发工作，由中国科学院386微机研究，开发组直接组织编译了此书共九册。虽然译校者大多是软件专业的研究生及工作多年的高资历工作人员，但由于印排仓促，审校时间紧迫，难免会有这样或那样的错误。望读者能给以谅解与指正。

导　言

欢迎使用 Microsoft 快速 C 编译器！你已经购买了能够用 C 语言进行工作的最有效的软件包之一，它是程序设计初学者以及有经验的软件开发者理想的一组工具。

Microsoft 快速 C 编译器是集效率、可移植性、灵活性于一体并提供完善的开发环境的 C程序设计语言的完全实现。它包括如下一些特性：

在一个集成化程序设计环境中获得所有你需要的工具

Microsoft 快速 C 编译器在单个软件包中提供你所需要的程序开发工具：集成程序编辑器编译器和调试程序（第 8 章）用 Microsoft 快速 C 编译器，你可以不用退出程序设计环境而对程序进行编译。运行、调试、编辑和重编译快速 C 编译器可以建立在内存中执行的程序，也可以建立在库加上可执行文件中使用的独立的目标文件，以便今后在DOS下运行。

因为大多数标准 C 库函数建立在程序设计环境中，因此通常没有必要和外部库连接。任何在环境中没有建立的标准 C 库例行程序可以装入称为快速库的用户可配置库中。

使用 Wordstar 兼容编辑器快速学习

学习快速 C 编辑器是一件简单的事情，如果你熟悉 MicroPro WordStar 程序，那么，你已经知道了大部分快速 C 编辑命令，因为它们和相应的 WordStar 命令完全一样。（7.1.1描述快速 C 编辑键）。

使用集成化调试程序快速找错

快速 C 包含许多专用调试程序的排错能力，它们使得以程序逻辑跟踪错误变得很容易。使用快速 C 调试程序，你可以：

- 一行一行地走过程序执行
- 设置多个断点
- 在程序执行过程中，在独立的窗口显示变量的值
- 在程序显示和程序输出显示之间转换
- 在源程序中查找特定的函数

§ 8.2— § 8.24 给出使用这些调试实用程序的方法，即使这些特性还不够，你还可以使用和 Microsoft C 优化编译器以及其它 Microsoft 语言产品一起提供的 Microsoft Code View 窗口调试程序。

在你的手头很方便地获取帮助

你有关于 C 语句语法方面的问题吗？你有关于库例行程序参数类型方面的问题吗？你有关于快速 C 编辑器键方面的问题吗？用快速 C 的上下文相关的连联帮助实用程序可以很快

地得到答案，浏览一下题目的菜单，或者简单地照亮(highligf)某一函数名并按键，你甚至不需要离开编辑程序，信息便出现在屏幕顶端的一个单独的窗口中。

图形库给程序添加直观的紧缩

和快速C库一起提供的图形库使你能够写出复杂的图形应用程序，该库支持所有的IBM及其兼容机显示适配器，包括IBM视频图形陈列（VGA），单色显示适配器（MDA），彩色／图形适配器（CGA），以及增强型图形适配器（EGA），有关图形库的信息参见第四章。

快速编写程序，然后用Microsoft C优化编译器进行优化

由于Microsoft快速C编译器和Microsoft C优化编译器在源文件和目标文件一级是完全兼容的，当你需要增加Microsoft C优化编译器提供的灵活性和效率时，保证你可以顺利通过，如果你是在用Microsoft C优化编译器开发程序，你可以使用Microsoft快速C编译器加快程序开发的早期进程，然后再用Microsoft C优化编译器对程序性能进行精细的调整。

用和OS/2兼容的用戶接口快速启动

Microsoft快速C编译器的用戶接口很容易学会。你只需用键盘，任选鼠标器或两者简单地从菜单中选择命令，或者不用打开菜单就按一些特殊键执行命令。看来快速C用戶接口和工作起来都很象新Microsoft显示管理程序与OS/2操作系统的接口。

自动维护大程序

具有大量模块的大型程序是用Microsoft快速C编译器自动进行维护的，在你编辑构成某一程序的模块时，将它们加到“程序模块表”中，在你重新建立该程序时，快速C只是重新编译那些自上次程序建立以来修改过的模块，以此节省你的时间。

程序模块表保存在一个与MAKE程序维护实用程序兼容的文件中；因而，程序可以在快速C程序设计环境外自动更新。

利用下面这些附加特性取得所想要的结果

Microsoft快速C提供下面这些附加特性：

- 一个完善的实用程序工具包，你几乎能满足任何程序设计的要求，该工具包包括QCL编译程序／连接程序驱动器；
  Microsoft覆盖连接程序LINK；Microsoft库管理程序LIB；以及Microsoft程序维护实用程序MAKE。
- 使用四种内存模型中的任何一种简化内存管理。QCL编译程序／连接程序驱动器支持小内存、中内存、紧致内存和大内存模型。
- 出类拔萃的文档参考资料。文档资料包括完整的C程序设计语言和标准C库例行程序的参考资料，其中包括数百个完整的程序例子。

系统所需配置

Microsoft 快速 C 编译器至少需要下述一些设备配置：

- 一台运行 MS-DOS 或 PC-DOS2.1 版本或其后版本的 IBM 个人计算机或者严格与其兼容的机器。
- 两个软盘驱动器或者一个软盘驱动器和一张硬盘。
- 448k 可用 RAM 内存。

使用本手册

Microsoft 快速 C 编译器是为一广范围的用户设计的，从对程序设计感到陌生的初学者到能够熟练使用 C 语言的有经验的程序开发者。手册的不同部分满足不同需要的用户。

下面的各段解释本手册是如何组织的，给不同类型的用户建议不同的途径，并描述在该软件包中提供的其它手册。

本手册是如何组织的

手册的第一部分“启动”，解释如何装配 Microsoft 快速 C 编译器，向你介绍快速 C 程序设计环境，并为熟悉诸如 BASIC 或 Pascal 语言的用户描述 C 语言的重要特性，该部分还告诉你如何使用由 Microsoft 快速 C 编译器提供的图形库。

第二部分，“快速 C 程序设计环境”，解释如何在快速 C 程序设计环境内对程序进行编辑，编译和调试。

本手册的第三部分，“快速 C 工具包”，解释如何用 QCL 及 LINK 程序编译和连接程序；如何用 LINK 建立快速库，如何用 LIB 实用程序建立独立的库；以及如何用 MAKE 实用程序对程序进行维护。

手册的附录部分包含你可能发现会有用的附加信息，诸如：

- ASCII 字符码表
- 快速 C 内存模型描述
- Microsoft C 和 Microsoft 宏汇编程序（MASM）之间的接口描述
- 错误信息表

本手册中简便快速的参考卡片列出快速 C 编辑和调试键及其键序列。

阅读手册的哪些内容

所有用户都应该阅读第一章，“装配并启动快速 C”，得到装配编译器软件的指示。如果你想用编译器做一些介绍性的手头练习，你会想浏览 §1.7.4 中描述的简短样例对话。

下面要干什么取决于你的特殊需求，如下所述：

对程序设计和 C 语言陌生的用户

如果你正在同时学习程序设计和 C 语言，参考本导言部分“学习资料”中所列的教材。

对 C 语言不熟悉的程序员

如果你有一些程序设计背景，但是以前从来没有用 C 语言写过程序，参考第 3 章，“C 快速启动”，获取一些关于 C 的特性及 C 和其它如 BASIC 和 Pascl 语言之间的差别的信息。

想要用快速 C 图形编制程序的用户

阅读第四章，“图形快速启动”，获取用 Microsoft 快速 C 编译器提供的图形函数编写程序的方法。

对集成化程序设计环境陌生的用户

阅读第五章，“快速 C 程序设计环境简介，”获取快速 C 程序设计环境组成部分的概况，然后阅读第二部分，“快速 C 程序设计环境”，学习如何使用建立在快速 C 环境中的编辑器、编译器以及调试程序。

对快速 C 提供的其它工具感兴趣的用户

阅读第三部分，“快速 C 工具包”，了解 QCL 编译程序／连接程序驱动器，LINK 连接程序，LIB 库管理程序以及 MAKE 程序维护实用程序。

你可以在快速 C 中得到的其它手册

Microsoft 快速 C 编译器软件包中还提供了两本附加手册：

- 《Microsoft C 语言参考手册》，它描述了 C 语言的 Microsoft ANSI-兼容的实现
- 《Microsoft C 运行库参考手册》，它描述了 Microsoft 快速 C 编译器提供的 350 多个库例行程序和 34 个内含文件。

记号约定

本手册使用如下这些记号：

约定的例子	约定的描述
Example（例子）	左列所示的字样用来模拟将要打印到屏幕上或通过打印机印出的信息的外形
placeholders（定位）	斜体字是命令行中的定位式你必须提供的信息的类型任选项说明。斜体字有时还在正文中用来表示强调。
DOS NAMES（DOS 名），**FILE NAMES**（文件名），和 **MACROS**（宏）	黑体大写字母用来表示可执行文件的名字以及由和产品一起提供的文件名，表示环境变量，表示替换常数，表示宏，表示寄存器、这些命令包括 DOS 内部命令，如 SET，也包括和快速 C 软件包一起提供的文件名。
Reserved words（保留字）	黑体表示正文必须按所示的字体打出。通常用黑体表示的正文包括运算符，关键字，库函数，命令，任选项和预处

	理程序伪指令，例子包括C关键字int和函数名fopen。
〔任选项〕	中括号括起来的为命令行中的任选域及任选项说明。
{选择1|选择2}	大括号和一竖线表示你可以在两项式或多项之中选择。选择用大括号括起来，用竖线将它们分开。
“术语定义”	引号使正文中定义的术语更明显，例如，术语“far（段外）”第一次定义时出现在引号中，引号还使正文中的命令行提示符更明显清晰。 有一些C结构需要引号。语言所需的引号为" "而不是“ ”。
重复元素……	跟在某一项目后面三点（也称为“省略号”）表示可以键入更多相同形式的项目。 语法行和程序例子中的一列是表示程序的一部分被省略。
键名	小的大写字母用来表示你必须要按的键的名字。键序列用通过加号（+）隔开来的小的大写字母表示。例如包括ENTER和CTRL+C。 本手册中键的名字对应于IBM个人计算机键板上打印出的名字。如果你使用的是其它机器，这些键的名字可能稍微有些不一样。主键盘右边的数字键盘上的光标移动组键（有时也称为“箭头”）称做方向键。每个单独的方向键或者指键板上箭头的方向，或者指键板上的名字（前者如LEFT,RIGHT,UP,DOWN,后者如PGTP和PGDN)。 回车键表示为ENTER。

学习资料

你或许想用下面所列的书籍进一步开发C。它按困难度递增的顺序列出，从开始到高级。

Hancock,Les,and Morris Krieger.*The C Primer*,*2d ed*.New Yori:McGraw-Hill,1985 (A beginner's guide to programming in the C language.)

Schilat,Herbert.*C Made Easy*.Berkeley,CA:Osborne/McGaw-IIill,1985.(A good introduction to D fol the reader who already knows BASIC.)

Waite,Mitchell,Stphine Prata,and Donald Martin.*C Primer plus*.Indianapolis,IN: Howard W.Sams,Inc.,1984.(The best-sclling introduction to the C language.)

Plum,Thomas.*Learning to Program in C*.Cardiff,New Jersey:Plum Hall,Inc.,1983 (A widely used introductory college text on computer programming using the C language.)

Kochan,Stephen. *Programming in C*.Hasbrouck Heights, NJ:Haydcn Book Company, Inc, 1983. (A comprehensive introduction to C with some emphasis on the UNIX environment.)

Harbison, Samuel P. and Guy L. Steele, Jr. *C: A Reference Manual, 2d ed.* Englewood Cliffs, NJ: Prentice-Hall, Inc., 1987. (An outstanding reference to the C language. The second edition incorporats the ANSI standard.)

Kernighan, Brian W., and Dennis M. Ritchie. *The C Programming Language.* Englewood Cliffs, NJ: Prentice-Hall, Inc., 1978. (The origi-nal, classic C book, known to insiders as "K & R" or as "the white book." Useful after you have learned C.)

Tondo, Clovis L., and Scott E. Gimple. *The C Answer Book.* Englewood Cliffs, NJ: Prentice-Hall, Inc., 1985. (A collection of answers to the exercises in K & R. A companion volume to K & R.)

Jaesche, Rex. *Solutions in C.* Reading, Massachusetts: AddisonWesley, 986. (A useful collecton of C programming tips.)

Ward, Robert. *Debugging C.* Indianapolis, Indiana: Que Corporation, 1986. (A guide to the techniques of debuggng C programs.)

Schustack, Steve. *Variations in C.* Redmond, Washington: Microsoft Press, 1988. (A guide to programming business applications in C.)*

Hansen, Augie. *Proficent C.* Redmond, Washington: Microsoft Press, 1987. (A guide to advanced C programming in the MS-DOS environment.)*

从 Microsoft 获取帮助

如果你感到在软件中发现了问题，请用本手册后面的 Product Assisfance Request（产品辅助要求）报告该问题。

如果你对本产品的手册有什么注释或建议，请使用 Documenfafion Feedback carc（文档反馈卡片）。

第一部分 启 动

手册的这部分将使你开始用 Microsoft 快速 C 编译器进行程序设计。

第一章说明如何用 SETUP 程序装配编译器以及在快速 C 程序设计环境中如何启动，如何退出；同时还描述了一个样例对话，在其中你可以用快速 C 编译器装入，编译并运行程序。

第二章描述快速 C 程序设计环境和它里面常用的操作，这章是为那些在详细阅读菜单和命令之前就想试验使用快速 C 的用户设计的。

第三章向那些熟悉诸如 Pascal 或 BASIC 语言的用户介绍 C 程序设计语言。

第四章介绍如何用图形库例行程序编写绘图程序。

第一章 装配并启动快速C

本章告诉你如何装配并启动快速C编译程序。在你使用编译程序之前，要确保：

1. 制作产品（product）备份盘（参见1.1节）
2. 核对你的磁盘内容（参见1.2节）
3. 阅读你的产品分布盘（product distribution disk）的工作拷贝上的READMEDOC文件，了解本手册打印后对软件所做的改变和增添。
4. 运行SETUP程序，装配软件（参见1.3.1.1—1.3.1.2及1.3.2.1—1.3.2.2等章节）。
5. 建立DOS环境，使快速C能够找到它所需要的文件。

本章提到几个DOS过程。特别地，DOS SET和PATH命令用来给DOS环境变量赋值。如果你对所提到的DOS过程不熟悉，请查阅DOS用户指南的有关说明。

注：本手册中，术语“DOS”既指MS—DOS，也指PC—DOS。

1.1 制作磁盘备份

当你从软件包中取出磁盘后，你应当做的第一件事情就是用DOS COPY命令或者DISKCOPY实用程序制作工作拷贝。将原来的磁盘保存起来，以便将来再制作工作拷贝。

1.2 核对磁盘内容

当你第一次打开你的编译程序软件包时，你可能想证实你确实拥有完整的一组软件，在你编译程序软件包中，有产品分布盘，其中包含着命名为PACKING.LST的文件，该文件列出并描述组成Microsoft快速C软件包的文件和手册。

为适应可能会加到Microsoft快速C编译程序中的新特性，某些文件或程序可以转移到本手册所描述的磁盘以外的磁盘上去，如果你找不到你所需要的文件或者程序。请查看PACKING.LST文件找出该文件或程序在哪个分布盘上。

1.3装配快速C

用你的1号(#1)库分布盘工作拷贝上提供的SETUP程序装配快速C软件，后面的部分给出关于如下内容的信息：

1. 当SETUP装配编译程序软件时真正做了些什么
2. 怎样在硬盘以及软盘系统中运行SETUP以装配编译程序
3. 如何建立DOS环境使得快速C能够找到它所需要的文件

注：如果你还有 Microsoft C 优化编译器的 5.0 版本，跳过这里所描述的装配过程。运行提供 Microsoft C 优化编译器的 SETUP 版本。版本 5.0 SETUP 同时装配 Microsoft C 优化编译程序和 Microsoft 快速 C 编译程序。参看“Microsoft C 优化编译器用户指南”第二章的说明。

.3.1 在硬盘系统中装配

1.3.1.1—1.3.1.6 几节给出了在硬盘系统中装配 Microsoft 快速 C 编译程序的指示。

3.1.1 在硬盘系统中 SETUP 做些什么

当你在硬盘系统中装配快速 C 时，SETUP 程序完成如下一些操作：

- 将所有那些你所需要的文件复制到你所指定的目录或者磁盘中。
- 为你在 SETUP 命令行中指定的每一内存模型／数学软件包建立独立的库，许多标准 C 库例行程序建立在快速 C 程序设计环境中。不过，在某些情况下，快速 C 编译程序在 SETUP 建立的库中查找在其它地方找不到的例行程序。如果你在快速 C 程序设计环境外面建立程序也要使用这些库。
- 建立一个命名为 NEW—VARS.BAT 的批文件，给 DOS 环境变量置值，使得快速 C 能够找到它所需要的文件。
- 建立一个命名为 NEW—CONF.SYS 的文件，该文件包含你的 CONFIG.SYS 文件中的 files 以及 buffer 参数的合适的设置。

.3.1.2 在硬盘系统中运行 SETUP

当你准备好运行 SETUP 时，将你的 1 号（#1）库分布盘的工作拷贝插入 A 驱动器中，然后键入如下形式的命令行：

SETUP H C:\dest {s|M|C|L}……{EM|87}……[GR] [\bin] [\incl] [lib]

警告

你给 SETUP 的自变量中有几个都是目录名。如果你所指定的目录不存在，则 SETUP 自动建立该目录。

SETUP 在用相同的名字装配新文件时，将新文件重写到原来的文件中，除非你不关心该目录名下文件的重写问题或者你知道该目录下没有与编译程序文件同名的文件，否则，不要给 SETUP 已有目录的名字。

上述所示的 SETUP 命令行中的每一项（或自变量）给 SETUP 关于你想如何装配快速 C 编译程序的信息。任选自变量于中括号（[ ]）中，如果你省略某一任选自变量，SETUP 将使用适当的缺省值。有些任选自变量允许你键入问号（？）来选择缺省值。

下面是你可以在 SETUP 命令行中键入的自变量，当程序分布盘还在 A 驱动器中时，键入不带自变量的 SETUP，你将会得到这些自变量的一个简单的联机解释。

自变量	意义
H	键入 H 告诉 SETUP 你在硬盘系统中装配。
C:\dest	键入 C:告诉 SETUP 你在 C 驱动器中装配，后面紧跟着供装配用的“目

的地”目录的名字（\dest）。SETUP将你在命令行中给出的所有其它名字作为该目录的子目录。

S,M,C,L　　键入这些字符中的一个或几个，相互间用空格隔开，告诉SETUP你将使用哪一种内存模型，键入M，因为中内存模型是快速C程序设计环境的缺省值。如果你将在快速C程序设计环境外面开发程序（使用第九章，“编译和连接程序”中讨论的QCL命令），键入S，因为小内存模型是QCL命令的缺省值，另外，如果你要使用紧致内存模型或大内存模型，键入C（紧致内存模型）或L（大内存模型）。

SETUP用你所给的内存模型自变量和浮点数学自变量（参见下面的内容）决定建立哪些库。组合库可以加快程序的创建过程，不过，因为组合库很大，你应该只说明那些你知道要用的内存模型（如果你要用一个以上的内存模型，你或许也想用非组合库，非组合库不象组合库那样占有磁盘空间，关于进一步信息，参阅1.4节）。

EM,87　　键入二者或其中之一告诉SETUP你想如何处理你的程序中的源点数学运算。

通常，键入EM告诉快速C你将使用一个符点仿真器而不是80287协同处理器（如果你需要关于处理程序中源点数学的更进一步的信息，参阅§9.3.5节）。如果你开发的是完成科学、数学或财务的实数计算程序，键入87，运行将会在有协同处理器的系统中进行，键入87将会建立一个附加库，该库需要大量的磁盘空间。不过，用该库创建程序时，程序将会更小和更快。

GR　　如果你想在SETUP所建立的库中建立快速C图形函数，键入GR（图形函数在本手册的第四章和Microsoft C运行库参考手册中均有描述）。任选这项将使库增大约50k。如果你不给出任选这项，图形函数便不包含在内。如果你的程序使用快速C图形函数，快速C必须要找到命名为GRAPHICS.LIB的库。关于更进一步的信息，参见1.5节。

\bin　　键入你想将包括编译程序、连接程序和其它实用程序在内的编译程序可执行文件装入其中的dest子目录，如果你省略该自变量或者用（？）代替，SETUP将使用缺省值\BIN作为子目录。

\incl　　键入你将包含文件装入其中的dest子目录。如果你省略该自变量或用问号（？）代替，SETUP将用缺省值\INCLUDE作为子目录。

\lib　　键入你想将库文件装入其中的dest子目录，如果你省略该自变量或者用问号（？）代替，SETUP将用缺省值\LIB作为子目录。

SETUP将自动建立如下子目录：

- 编译程序在编译过程中存放临时文件的命名为\TMP的子目录。
- 命名为\SAMPLE的可执行文件目录的子目录，SETUP将说明程序装配在其中。

例

```
SETUP H C:\ M EM? ? ?
```

上述命令行告诉 SETUP 将编译程序软件装配到硬盘驱动器 C 根目录（\）下的缺省子目录下。

对于编译程序和实用程序可执行文件，其缺省子目录为\BIN，对于内含文件，其缺省子目录为\INCLUDE，对于库文件，其缺省子目录为\LIB。SETUP 将为中内存模型和浮点数学包仿真器建立适当的库，并将此库命名为 MLIBCE.LIB。

SETUP H C:\QC SMCL EM GR 87 \BINDIR \INC \LIBS

上述命令行告诉 SETUP 将编译程序软件装配到硬盘驱动器 C 中目录\QC 的给定子目录可执行文件装配到\QC\BINDIR 下；内含文件装配到\QC\INC 下；库文件装配到\QC\LIBS 下；说明文件装配到\QC\BINDIR\SAMPLE 目录下。为所有有效内存模型和浮点数学软件包建库。（一共有 8 个库）每个库均包含 Microsoft C 图形函数。

1.3.1.3 *在硬盘系统中建库*

当你按下 ENTER 键后，SETUP 开始建立并装配库。当需要换驱动器 A 中的硬盘时，SETUP 会给你提示。SETUP 读取你在命令行中给出的每个内存模型自变量（S，M，C，L）和浮点数学自变量（EM，87）并为两者的每种可能组合各建立一个库。

SETUP 还根据你所选择的内存模型以及浮点数学软件包给各个库命名，各缺省库的名字具有如下所示的形式：

{S|M|C|L}LIBC{E|7}.LIB

库名的第一个字符和内存模型自变量相同：S 为小内存模型（缺省模型），M 为中内存 C 为紧致内存模型，L 为大内存模型。基本名的最后一个字符(基本名为.LIB 扩展名前面的部分）根据浮点数学自变量决定：如果所给的自变量为 EM，则末字符为 E；如果所给的自变量为87，则末字符为 7 。

例如，名字为 MLIBCE.LIB 的库支持中内存模型、浮点数学软件包仿真器，它是快速 C 程序统计环境的缺省值。命名为 SLIBC7.LIB 的库支持小内存模型、8087/80287浮点数学包，仅当程序运行于有数学协同处理器的机器上时，才可以使用。

注：为讨论简单起见，本手册余下的部分将使用缺省名来标识支持内存模型和数学包的特殊组合库。

1.3.1.4 *删除库的某些成分*

SETUP 显示如下的附加提示信息：

setup no longer needs the library sub-components and you do not normally need them to compile and link your C program. Do you want to delete them? [y/n]

如果你使用内存模型和浮点数学色组合而不是那些你在 SETUP 命令行中所给出的参数值，你可能想要保留非组合库。然而，如果你只使用组合库所支持的模式和数学色，你可以删除非组合库。键入 Y 或者y删除非组合库，键入 N 或者 n 保留非组合库。

1.3.1.5 *完善装配*

SETUP执行若干附加步骤，包括创建NEW-VARS.BAT文件和NEW-CONF.SYS 文件。

1.3.1.6 *在硬盘系统中建立DOS环境*

装配的最后一步是建立DOS环境，使快速C 编译程序能够找到它所需要的文件。

SETUP自动建立一个命名为NEW_VARS.BAT的批文件。用 NEW-VARS.BAT 设置你的DOS环境变量，使编译程序和连接程序能够找到它所需要的文件。

如果愿意，你可以在NEW-VARS.BAT文件里在你的 AUTOEXEC.BAT文件中加上 SET 命令，使你的系统每次启动时都能正确地建立DOS环境。

除了NEW_VARS.BAT文件以外，SETUP还建立一个命名为 NEW_CONF.SYS 的文件。该文件为Microsoft快速C 编译程序设置合适的files及 buffers 参数值。你可以用 NEW_CONF.SYS文件取代你原有的CONFIG.SYS 文件，也可以从 NEW_CONF.SYS 文件中将 buffers 和 files设置复制到你原有的CONFIG.SYS文件中。

SETUP将NEW_VARS.BAT文件及 NEW-CONF.SYS 文件装配到你装配快速C 可执行文件的 dest子目录下（缺省值为dest\BIN\SAMPLE）。

1.3.2 在软盘系统中装配

1.3.2.1—1.3.2.4节给出在软盘系统中装配Microsoft 快速C 编译程序的步骤。注意，这些步骤假定你将在A驱动器中运行SETUP。如果你愿意在B驱动器下运行，只需简单地将下面步骤中的“驱动器B”换成“驱动器A”，将“驱动器A”换成“驱动器B”即可。

1.3.2.1 在软盘系统中SETUP做些什么

当你在软盘系统中装配快速C 时，SETUP程序为你在SETUP命令行中指定的每个内在模型/数学色组合建立一个独立的库。许多标准C 库例行程序建立在快速C 程序设计环境中。但是，在某些情况下，快速C 编译程序在SETUP建立的库中查找在其它地方找不到的例行程序。另外，如果你在快速C 程序设计环境外面建立程序时也要使用这些库。

1.3.2.2 在软盘系统中运行 SETUP

在你在软盘系统中运行 SETUP 之前，先决定为保存 SETUP 建立的库需要多少格式化空盘。你需要为你将使用的每个内存模型准备一张格式化空盘，另加一张“零用(scratch)”盘。如果你只将使用快速C 程序设计环境的缺省内存模型（中内存模型），你所需要的只是两片盘。SETUP 将它建立的每个库放入一片单独的软盘中。

当你将所需数目的盘格式化以后，将你的 1 号库分布盘的工作拷贝插入A驱动器中。然后键入如下形式的命令行：

SETUP F B: {S|M|C|L}…{EM|87}[GR]

SETUP 命令行中的每一项（或"自变量"）给 SETUP 一些你想如何装配快速C 编译程序的信息。任选自变量置于中括号（[ ]）中，如果你省略某一任选自变量，SETUP 将使用适当的缺省值。有些任选自变量允许你键入问号（？）来选取缺省值。

下面是你可以在命令行中键入的自变量。当程序分布盘还在A驱动器中时，键入不带参数的 SETUP，你将会得到这些自变量的一个简短的联机解释。

自变量	意义
F	告诉 SETUP 你是在软盘系统中装配。
B:	键入 B:告诉 SETUP 你在驱动器B中的磁盘上建库。
S, M, C, L	键入这些字符中的一个或几个，相互间用空格隔开，告诉SETUP 你将使用哪一种内存模型。键入M，因为中内存模型是快速C 程序设计环境的缺省值。如果你将在快速C 程序设计环境外开发程序（使用第九章，“编译和连接程序”中讨论的QCL 命令），键入

S，因为小内存模型是QCL命令的缺省值。另外，如果你要使用紧致内存模型或大内存模型，键入C（紧凑内存模型）或L（大内存模型）。SETUP用你所给的内存模型参数和浮点数学参数（参见下面的内容）决定建立哪些库。组合库可以加快程序的创建过程，不过，因为组合库很大，你应该只说明那些你知道要用的内存模型。（如果你要用一个以上的内存模型，你或许也想用非组合库，非组合库不象组合库那样占有磁盘空间。关于更进一步的信息，参阅1.4节）

EM，87

键入二者或其中之一告诉SETUP你想如何处理你的程序中的浮点数学运算。

通常，键入EM告诉快速C你将使用一个符点仿真器而不是8087或80287协同处理器。（如果你需要关于处理程序中浮点数学的更进一步的信息，参阅9.3.5节）如果你开发的是完成科学、数学或财务方面的实数计算程序，键入87，运行将在有协同处理器的系统中进行。键入87将建立一个附加库，该库需要大量的磁盘空间。不过，用该库创建程序时，程序将会更小更快。

GR

如果你想在SETUP所建立的库中建立快速C图形函数，键入GR（图形函数在本手册的第四章和Microsoft C运行库参考手册中均有描述）。这任选项将使库增大约50K。如果你不给出任选这项，图形函数便不包含在内。如果你的程序使用快速C图形函数，快速C必须要能找到命名为GRAPHICS.LIB的库关于更进一步的信息，请参阅1.5节。

例

SETUP FB: M EM

上述命令行告诉SETUP在B驱动器中的软盘上为中内存模型和浮点仿真器建立适当的库，MLIBCE.LIB。

1.3.2.3 *在软盘系统中建库*

当你按下ENTER键后，SETUP开始建立并装配库。SETUP在每片软盘上装配一个库，它会提示你何时在A驱动器和B驱动器中换上所需要的库分布盘，何时换上你要在它上面装配新库的软盘，何时换上暂用盘。

SETUP读取你在命令行中给出的每个内存模型自变量（S,M,C或L）和浮点数学自变量（EM或87），并为两者的每种可能组合各建一个库。SETUP如同建库那样显示它正在建造的库名。

SETUP还为根据你选择的内存模式和浮点数学软件包所建的库命名。每个缺省库名形式如下：

{S|M|C|L}LIBC{E|7}.LIB

库名的第一个字符和内存模型自变量一样：S为小内存模型（缺省模型），M为中内存模型，C为紧致内存模型，L为大内存模型。基本名（.LIB扩展名前面的一部分名字）的最后一个字符由符点数学自变量决定：如果所给的自变量为EM，则末字符为E；如果所给的

参数为87，则末字符为 7 。

例如，命名为 MLIBCE.LIB的库支持中内存模型和仿真器浮点数学包，它为快速C程序设计环境的缺省库。名字为 SLIBC7.LIB 的库支持小内存模型和8087/80287 浮点数学包，该库仅被运行在有数学协同处理器的机器上的程序使用。

注：为讨论简单起见，本手册的余下部分将使用缺省库名来标识支持内存模型和数学包特别组合的库。

1.3.2.4 *在软盘系统中建立 DOS 环境*

装配的最后一步是建立 DOS 环境，使快速 C 能够找到它所需要的文件。键入如下形式的 DOS 命令：

```
SET PATH = A:;
SET LIB = A:;
SET INCLUDE = A:\INCLUDE;
```

这些命令告诉快速 C 在驱动器 A 中磁盘的根目录下查找可执行文件和库文件，在驱动器 A 中磁盘的\INCLUDE 目录下查找内含文件，这些设置是一直保持到你下次重新启动系统之前。你可以添加这些命令到你的 AUTOEXEC.BAT文件，使得当你重新启动系统时，它们可以自动地执行。

在你运行快速 C 以前，把以下值打入你的 CONFIG.SYS 文件中，然后重新启动你的系统：

```
files = 15
buffers = 10
```

1.4 使用非组合库

由于用非组合库创建程序时需要更多的时间，因此，SETUP程序建立组合库。如果你使用许多不同的内存模型和浮点数学包组合，你或许不想用完为存放所有那些你需要的组合库所要求的磁盘空间。这时，你可以使用库的各个组成成分。注意，使用非组合库有如下一些缺点：

- 一般地，你必须键入更长的命令行来编译和连接你的程序。
- 你必须告诉连接程序不使用它通常使用的库，并确切地给出适当的非组合库的名字。
- 用非组合库连接比用组合库连接需要更长的时间。

下面是和 Microsoft 快速C 编译程序一起提供的非组合库（其中 m 表示合适的内存模型）。

库	用途
mLIBC.LIB	标准运行库；包含除需要浮点支持的数学例行程序外的所有包括在 Microsoft C 运行库中的例行程序。
mLIBFP.LIB	浮点数学库；在你的程序使用 EM.LIB 或87.LIB 时需要的库。
EM.LIB	独立于模型的浮点仿真器；用来执行浮点运算。