

Quick BASIC 4.0

学习和使用手册

薛冰 朱望苏 等译

金啸东 等校



北京科海高技术集团公司培训中心

引 言

欢迎使用 Microsoft Quick BASIC 4.0 版。Quick BASIC 将一个完整的程序设计环境和最强有力的 BASIC 语言相结合，为您提供了快速开发高效、简洁程序的一切条件。在 Quick BASIC 环境中，可以任一方式将程序的编辑，运行和调试混为一体，由于所有的程序设计工具总是就绪可立即运行，所以用 Quick BASIC 能节约时间编出较好的程序。

- 编写程序时可立即执行代码

- Quick BASIC立即把输入的指令行转换为可执行代码，没有耽误时间和中断思路的单独“编译步骤”。只要按一个键，不论程序的完整程度如何，都可运行程序，如程序运行正确，则只需要通过四个步骤，就可将 Quick BASIC 程序转换为独立的 DOS 下的可执行程序。

- 即时帮助和语法检查

联机帮助 Help 为每个语句提供语法检查，同样也为菜单命令、快速键和 ASCII 码提供语法检查，Quick BASIC 的“smart editor”自动检查每一行语句的语法，所以在编程时就可以改正错误。

- 即时调试

Quick BASIC 中，没有单独的调试步骤，您可在编辑时调试它。可使用三种不同的方式跟踪整个程序，并在一个特别的窗口中显示变量值变化。多重断点和观察点使得改变和检查程度很容易。

- 多重模块

将大程序在逻辑上分解成多个部分，可使程序的编写和调试简化。Quick BASIC 使得创建多模块程序变得容易，因为它能同时将所有模块保存在内存中，不但是程序模块，还包括各模块中的过程。

- 多重编辑窗口

Quick BASIC 多重编辑窗口允许同时观察程序的各个部分。调试时，使用多重编辑窗口跟踪各个过程能保持执行逻辑的清晰。

- 完全图形支持

Quick BASIC 完全支持 PS/2 的图形模式：VGA 和 MCGA。同时也支持 CGA 和 EGA 以及单色模式。现在 Quick BASIC 支持 Hercules® 图形卡。

除了程序设计环境，Quick BASIC 在扩展 BASIC 语言方面还包括以下这些强大的功能。

- Quick BASIC支持二进制文件，所以程序可以创建和操作任何形式的文件。

- 用户定义的数据类型简化了随机访问文件的输入和输出，并帮助您用数字变量和字符串构造更加复杂的数据结构。

- Quick BASIC子程序和函数可以递归，允许您编写出简洁、强有力的程序。

- 长整数（32位）和8087/80287协处理器使您做快速、精确的数字运算。

- 象 SELECT CASE 和 IF...THEN...ELSE 这样的复杂控制流程语句，改进了程序的逻辑流程。

目 录

第一章 安装 Quick BASIC	(1)	2.2.5 建立可执行程序	(17)
1.1 备份 Quick BASIC 盘	(1)	2.2.6 在 DOS 下运行程序	(17)
1.2 盘目录	(1)	2.2.7 该版本 Quick BASIC 的改进 之处	(18)
1.3 在硬盘上安装 Quick BASIC	(2)	2.2.7.1 环境的改进	(16)
1.4 在软盘上安装 Quick BASIC	(2)	2.2.7.2 新的语言特性	(16)
1.5 使用 Mouse	(4)	2.2.8 下一步	(19)
1.6 使用协处理器	(4)	2.3 假定用户已使用过 BASIC 编译器	(19)
1.7 置 DOS 参数	(5)	2.3.1 在 DOS 命令行上编译与链接 处理	(19)
1.7.1 PATH 命令	(5)	2.3.2 下一步	(20)
1.7.2 LIB 命令	(5)	2.4 Quick BASIC 用户须知 ...	(20)
1.7.3 No87 命令	(6)	2.4.1 为什么说 Quick BASIC 比其它 BASIC 产品好?	(20)
第二章 初识 Quick BASIC	(7)	2.4.1.1 用户所希望的开发环境 ...	(20)
2.1 作为 BASIC 程序员	(7)	2.4.1.2 新的语言特性	(22)
2.1.1 Quick BASIC 的启动	(8)	2.4.2 qb 命令	(23)
2.1.2 装入程序	(8)	2.4.3 制定 Quick BASIC 屏幕	(24)
2.1.3 运行程序	(9)	2.4.4 QB.INI 文件	(25)
2.1.4 Quick BASIC 中解释、编译、 运行的比较	(10)	2.4.5 从 Quick BASIC 获得帮助 ...	(25)
2.1.5 在盘上建立一个可执行文件 ...	(11)	2.4.6 Quick BASIC 文件命令规则	(26)
2.1.6 退出 Quick BASIC	(12)	2.4.6.1 路径名	(26)
2.1.7 在 DOS 下运行程序	(12)	2.4.6.2 大、小写字母	(26)
2.1.8 Quick BASIC 与 BASIC 的 区别	(12)	2.4.6.3 文件扩展名	(27)
2.1.8.1 环境增强	(12)	2.4.6.4 定义文件名	(27)
2.1.8.2 新语言特点	(13)	第三章 进入 Quick BASIC 环境 ...	(28)
2.1.9 下一步	(13)	3.1 Quick BASIC 屏幕	(28)
2.2 如果用户已使用过 Quick BASIC V2.0 或 V3.0	(14)	3.2 打开菜单并选择命令	(29)
2.2.1 启动 Quick BASIC	(14)	3.2.1 键盘技能	(30)
2.2.2 装入 Quick BASIC 程序	(14)	3.2.2 鼠标器的技能	(31)
2.2.3 子程序放在何处?	(15)	3.2.3 快速键的使用	(31)
2.2.4 运行 Quick BASIC 程序	(16)		

3.3 使用对话框.....	(33)	5.2.3 关闭语法检查	(55)
3.4 窗口的使用.....	(35)	5.3 插入与重叠输入.....	(56)
3.4.1 窗口类型	(35)	5.4 选择文本.....	(56)
3.4.2 观察窗口的划分	(36)	5.5 删除和插入文本.....	(57)
3.4.3 改变活动窗口	(36)	5.5.1 使用剪贴板	(57)
3.4.4 改变窗口尺寸	(37)	5.5.2 clear, Cut, Copy 和 paste 命令	(57)
3.4.5 在活动窗口内滚动	(37)	5.6 恢复当前行.....	(57)
3.4.6 立即窗口	(37)	5.7 搬移和拷贝文本.....	(58)
3.5 临时返回 DOS (DOS shell).....	(38)	5.8 查找和替换.....	(58)
3.6 退出 Quick BASIC	(38)	5.8.1 查找文本	(59)
第四章 源文件的管理	(39)	5.8.1.1 查找	(59)
4.1 Quick BASIC 中的文件分类	(39)	5.8.1.2 被选中的文本	(60)
4.2 程序.....	(41)	5.8.1.3 重复上一个查找	(61)
4.2.1 装入并列程序	(41)	5.8.1.4 标号	(61)
4.2.1.1 指定一个文件	(42)	5.8.2 置换被选中的文本	(61)
4.2.1.2 列出目录内容	(42)	5.9 在程序中运用位置标记.....	(63)
4.2.2 保存文件	(43)	5.10 输入特殊字符	(63)
4.3 模块	(44)	5.10.1 输入扩展 ASCII 码字符	(63)
4.3.1 建立和装入模块	(44)	5.10.2 输入控制序列字符	(63)
4.3.2 显示多个模块	(46)	5.11 缩进	(63)
4.3.3 保存多个模块	(46)	5.12 连接行	(64)
4.4 改变主模块.....	(46)	5.13 从其它文件中拷贝文本	(64)
4.4.1 .MAK 文件	(47)	5.13.1 拷贝一个完整文件	(65)
4.4.2 从程序中除去模块	(48)	5.13.2 拷贝文件的一部份	(65)
4.5 Include 文件	(48)	5.14 编辑命令总结	(65)
4.5.1 建立和装入 Include 文件	(49)	第六章 建立和运行 Quick BASIC	
4.5.2 观察和编辑 Include 文件	(49)	程序	(68)
4.5.3 嵌套 Include 文件	(50)	6.1 编写一个程序.....	(68)
4.6 文献.....	(51)	6.1.1 建立一个主模块	(68)
4.7 合并两个文件.....	(52)	6.1.2 输入程序语句	(69)
4.8 打印文件.....	(53)	6.1.3 在内存里编码和运行	(70)
第五章 编辑	(54)	6.1.4 返回输出屏幕	(70)
5.1 输入文本.....	(54)	6.1.5 在立即窗口操作	(71)
5.2 Smart 编辑器的特性	(54)	6.1.5.1 立即窗口的特点	(71)
5.2.1 什么时候启用 Smart 编辑器?	(54)	6.1.5.2 立即窗口中允许的语句	(72)
5.2.2 自动语法检查	(55)	6.1.5.3 计算	(72)
		6.1.5.4 测试屏幕输出	(72)
		6.1.5.5 调用过程	(72)

6.1.5.6 改变运行程序中的变量值	7.3.4.3 观察点	(88)
..... (72)	7.3.4.4 观察表达式	(89)
6.1.5.7 模拟运行时间错误	7.3.5 控制执行	(90)
(73)	7.3.5.1 继续(F5)	(90)
6.1.6 带有行命令参数的程序	7.3.5.2 执行到光标处(F7)	(91)
(73)	7.3.5.3 设置下一个语句	(91)
6.2 在 Quick BASIC 内部建立可 执行文件	7.3.5.4 下一个语句	(91)
(73)	7.3.5.5 重新启动	(91)
6.2.1 Quick 库与可执行文件	7.4 高级调试	(91)
(75)	7.4.1 calls 菜单	(91)
6.2.2 关于运行时模块的考虑	7.4.2 与 Code View 调试程序的兼容性	(92)
(75)	第八章 Quick 库	(94)
6.2.2.1 使用运行时模块	8.1 库的类型	(94)
(75)	8.2 Quick 库的优点	(94)
6.2.2.2 建立独立的程序	8.3 建一个 Quick 库	(95)
(75)	8.3.1 创建一个 Quick 库所需的文件	(96)
6.2.3 运行时可执行文件中的错误检查...	8.3.2 产生一个 Quick 库	(96)
(76)	8.3.3 在 QB 环境中 一个 Quick 库...	(96)
6.2.4 可执行文件中的浮点运算	8.4 Quick 库的使用	(98)
(76)	8.4.1 装入一个 Quick 库	(98)
6.3 在程序中使用 SUB 和 FUNCTION 过程	8.4.2 Quick 库中的浮点运算	(98)
(77)	8.4.3 查看 Quick 库的内容	(99)
6.3.1 建立 SUB 或 FUNCTION 过程	8.5 系统提供的库	(99)
(77)	8.6 .QLB 文件扩展名	(99)
6.3.2 过程的保存与命名	8.7 在命令行上建库	(99)
(79)	8.8 在 Quick 库中使用其它语言 的例行程序	(100)
6.3.3 观察与编辑模块与过程	8.9 Quick 库的内存考虑	(101)
(79)	8.10 创建紧缩的可执行文件 ...	(101)
6.3.4 过程的自动说明	第九章 在 DOS 下编译和链接	(102)
(80)	9.1 BC.link.LIB	(102)
6.4 将模块合并到程序中	9.2 使用 BC 的优点	(102)
(80)	9.3 编译和连接过程	(102)
第七章 程序调试	9.4 用 bc 命令编译	(103)
(82)	9.4.1 定义文件名	(103)
7.1 Quick BASIC 的调试手段	9.4.1.1 大写和小写字母	(104)
..... (82)		
7.2 排错		
(83)		
7.3 Quick BASIC 调试工具特征		
(84)		
7.3.1 调试术语和概念		
(84)		
7.3.2 Debug 菜单		
(85)		
7.3.3 跟踪运行的程序		
(85)		
7.3.3.1 单步		
(86)		
7.3.3.2 过程单步		
(86)		
7.3.3.3 动态跟踪		
(86)		
7.3.3.4 History on		
(86)		
7.3.3.5 History Back		
(86)		
7.3.3.6 History Forward		
(86)		
7.3.4 断点、观察点、观察表达式 ...		
(86)		
7.3.4.1 观察窗口		
(87)		
7.3.4.2 断点		
(88)		

9.4.1.2 文件扩展名	(104)
9.4.1.3 路径名	(104)
9.4.2 使用 bc 命令选择项	(104)
9.5 连接	(106)
9.5.1 连接程序的缺省参数	(107)
9.5.2 给连接程序定义文件名	(108)
9.5.3 给连接程序定义库	(108)
9.5.4 连接程序需要的内存	(108)
9.5.5 使用连接程序的选择项	(109)
9.5.5.1 查看选择表	(110)
9.5.5.2 连接期间暂停	(110)
9.5.5.3 显示连接过程信息	(110)
9.5.5.4 消除连接程序提示	(111)
9.5.5.5 建立 Quick 库	(111)
9.5.5.6 紧缩可执行文件	(111)
9.5.5.7 不能紧缩段	(111)
9.5.5.8 忽略一般的 BASIC 库	(111)
9.5.5.9 设置段的最大数目	(112)
9.5.5.10 建立映象文件	(112)
9.5.5.11 映象文件中包含的行数	(113)
9.5.5.12 紧缩邻接段	(113)
9.5.5.13 使用 Code View 调试程序	(114)
9.5.5.14 区别情况	(114)
9.5.6 其它 Link 命令行选择项	(114)
9.6 管理独立库	(115)
9.6.1 执行 LIB 命令	(115)
9.6.2 LIB 的通常应答	(116)
9.6.3 命令符号	(116)
9.6.4 交叉引用列表文件	(118)
9.6.5 设置库页面的长库	(118)

附录 A BASIC 程序到 Quick

BASIC 的转换	(119)
A.1 源文件格式	(119)
A.2 磁带 BASIC	(119)
A.3 在 Quick BASIC 中禁止使 用的语句与函数	(119)
A.4 需作修改的语句	(119)

附录 B 本版本与低版本的区别 ... (121)

B.1 新的特性

B.1.1 用户定义的类型

B.1.2 IEEE 格式与算术协处理器的 支持

B.1.2.1 IEEE 格式数的范围

B.1.2.2 PRINT USING 与 IEEE 格式数

B.1.2.3 采用 /mbf 选择重新编译 原有程序

B.1.2.4 文件与程序的转换

B.1.3 在线帮助

B.1.4 长整数 (32 位)

B.1.5 定长串

B.1.6 语法检查

B.1.7 二进制文件的输入/输出 ... (125)

B.1.8 FUNCTION 过程

B.1.9 代码观察调试器支持

B.1.10 与其它语言的兼容性

B.1.11 多个模块同时驻留内存

B.1.12 与 Prokey™, Sidekick® 和 Superkey 的 兼容性

B.1.13 插入与重写方式

B.1.14 Wordstar 类型的键盘 命令

B.1.15 递归

B.1.16 在分开编译过程中的 错误列表输出

B.1.17 在分开编译过程中汇编 语言列表输出

B.2 与低版本在程序设计环境 上的不同

B.2.1 选择命令与选择项

B.2.2 窗口

B.2.3 新的菜单

B.2.4 菜单命令

B.2.5 编辑键变化

B.3 与前版本在编译与调试上的

区别	(129)	C.5 名引用与值引用方式传递数据	(153)
B.3.1 命令行区别	(129)	C.5.1 BASIC 参变量	(153)
B.3.2 分开编译差别	(130)	C.5.2 C 参变量	(154)
B.3.3 用户库和 BUILDLIB	(130)	C.6 在混合语言程序设计中的数	
B.3.4 对包含文件的限制	(130)	值与串数据	(154)
B.3.5 调试	(131)	C.6.1 整型数与实型数	(154)
B.4 对于 BASIC 语言方面的变化	(131)	C.6.2 字符串	(155)
B.5 文件兼容性	(135)	C.6.2.1 串格式	(155)
附录 C 调用 C 语言子程序和汇编		C.6.2.2 BASIC 中的传送	(156)
语言子程序	(136)	C.6.2.3 BASIC 中对 C 的传送	(157)
C.1 混合语言程序的组织	(136)	C.6.2.4 C 中的传送	(158)
C.2 混合语言程序设计要素	(137)	C.7 特殊的数据类型	(158)
C.2.1 构造混合语言调用	(137)	C.7.1 数组	(158)
C.2.2 命名规则要求	(138)	C.7.1.1 由 BASIC 传送数组	(159)
C.2.3 调用规则要求	(140)	C.7.1.2 数组索引与说明	(159)
C.2.4 参数传递要求	(140)	C.7.2 结构与用户定义类型的存贮	(160)
C.2.5 编译与连接	(141)	C.8 指针、地址变址与公用块	(161)
C.2.5.1 用适当的存贮模式进行编译	(141)	C.8.1 共用块	(161)
C.2.5.2 与语言库相连接	(142)	C.8.2 使用可变参数	(162)
C.3 BASIC 调用 C	(142)	C.9 B-OnExit 例程	(164)
C.3.1 BASIC 与其它语言的接口	(142)	C.10 汇编语言与 BASIC 间的	
C.3.1.1 DECLARE 语句	(142)	接口	(164)
C.3.1.2 利用 ALIAS	(143)	C.10.1 编写汇编语言过程	(164)
C.3.1.3 使用参数表	(144)	C.10.1.1 建立过程	(165)
C.3.2 其它的 BASIC 接口	(144)	C.10.1.2 进入过程	(165)
C.3.3 BASIC 调用 C	(145)	C.10.1.3 分配局部数据	(165)
C.3.4 关于由 BASIC 发出调用的限制	(147)	C.10.1.4 保护寄存器值	(166)
C.3.4.1 存贮分配	(147)	C.10.1.5 参数存取访问	(167)
C.3.4.2 不相兼容的函数	(148)	C.10.1.6 返回值 (可为任意的)	(168)
C.3.4.3 分配串空间	(148)	C.10.1.7 非 2 字节和 4 字节整数的	
C.3.4.4 执行 BASIC 文件输入/输出	(149)	返回值	(168)
C.3.4.5 事件与出错	(149)	C.10.1.8 退出汇编过程	(169)
C.4 C 对 BASIC 的调用	(149)	C.10.2 来自 BASIC 的调用	(170)
C.4.1 C 对其它语言的接口	(149)	C.10.3 在 BASIC 发来的调用中使用	
C.4.2 C 对 BASIC 的调用	(151)	CDECL	(172)

C.10.4 Microsoft® 段模式 ...	(172)	D.3 调试命令	(178)
附录 D 命令概述	(174)	D.4 编辑命令	(178)
D.1 使用选择项	(174)	D.5 窗口命令	(180)
D.2 菜单命令和快速键	(175)		

第一章 安装 Quick BASIC

本章描述了如何安装 Quick BASIC。

当读完本章，你就会知道怎样：

- 备份 Quick BASIC 盘
- 在硬盘或软盘上安装 Quick BASIC
- 设置环境变量，以便 Quick BASIC 查找它所需的文件

如果不熟悉本章所述的 DOS 操作，可查阅 DOS 文件指南。

1.1 备份 Quick BASIC 盘

为防止 Quick BASIC 原盘被破坏，可用 DOS 中的 DISKCOPY 命令复制 Quick BASIC 的所有盘。如果复制盘被破坏了，则再用原盘来复制。

1.2 盘目录

表 1.1 描述了 Quick BASIC 所提供原文件。在原盘中还附带有一些文件，如 BASIC 程序（带 BAS 扩展名的文件）以及在本手册印刷后添加的一些文件。在 1 号盘的 PACKING.LST 文件中列有 Quick BASIC 包含的所有文件，以及它们所在的盘号。

表 1.1 使用 Quick BASIC 时所需文件

文件名	描述	所做工作
QB.EXE	Quick BASIC 开发环境	在内存建立、调试和运行 BASIC 程序，建立独立的可执行程序，建 Quick 库。
BC.EXE	由 Run 菜单的文件命令来调用的 BASIC 命令行编译器	用 Quick BASIC 及 DOS 来建立可执行程序；建 Quick 库。
SETUP.BAT	Quick BASIC 安装程序	安装 Quick BASIC
QB.QLB	Quick 库，包含 DOS 中断后援程序 (Support routines)	---
QB.LIB	独立库，包含 DOS 中断后援程序	建 Quick 库
QB.BI	包含文件，与 QB.QLB 和 QB.LIB 一起使用	参见第八章“Quick libraries”介绍。
README.DOC	有用的重要信息在该说明书的手面均被打印出来	---
PACKING.LIS	列出 Quick BASIC 各张盘上全部文件	---
BRUN40.EXE	Quick BASIC 立即运行模块	运行用 BRUN40.LIB 生成的可执行 BASIC 程序。

表 1.1 (续)

文件 名	描 述	所做工作
QBLB40.LIB	Quick 库运行支持	建 Quick 库
BCOM40.LIB	Quick BASIC 立即运行模块库	由 Quick BASIC 及 DOS 生成可执行程序
BCOM40.LIB	Quick BASIC 交互式运行库	在 Quick BASIC 在和 DOS 下生成可执行程序。运行此库生成的程序无需 BRUN40.EXE
LINK.EXE	链接程序	在 Quick BASIC 和 DOS 下建可执行程序及 Quick 库
LIB.EXE	库管理程序	建独立库
NOCOM.OBJ	目标文件 (Object file), 用于链接无通讯要求的程序	使无通讯要求的程序长度达到最小。
MOUSE.COM	Quick BASIC 所用的鼠标器 (MOUSE) 驱动程序	使用鼠标器
DEM01.BAS	样本程序	第二章中“初识
DEM02.BAS	——	Quick BASIC”中
DEM03.BAS	——	Quick BASIC 的应用
QB.HLP	Quick BASIC 联机帮助文件	用于联机 (on-line) 帮助

1.3 在硬盘上安装 Quick BASIC

用 Quick BASIC 1 号盘上 SETUP.BAT 把 Quick BASIC 拷入硬盘。

不用 SETUP.BAT 也能安装 Quick BASIC, 即根据你使用 Quick BASIC 的需求来确定安装表 1.1 中的相应文件。

一旦确定你所需要文件后, 则要做下列事之一:

- 把所有文件拷贝到硬盘上的一个目录下。
- 拷贝不同类型的文件在不同的目录下, 并置 DOS 环境变量 (environment variables) 以便 DOS 及编译程序知道在何地找到所要文件。

第二种方法可使硬盘的编排方式较为整洁。见 1.7 节“置 DOS 变量”中的详细介绍。

1.4 在软盘上安装 Quick BASIC

要在软盘上安装 Quick BASIC, 可用 DOS 中的 Copy 命令把原盘上的文件拷贝到自己的盘上。参考表 1.1 来决定你所需的文件。

采用双软盘驱动器系统时, 建议用户按表 1.2 列出的文件配置, 将 Quick BASIC 拷在四张格式化过的软盘上。

表 1.2 双软盘驱动器上的软盘文件装配

盘 号	文 件
1	COMMAND.COM AUTOEXEC.BAT (可选) CONFIG.SYS (可选) QB.EXE QB.HLP MOUSE.COM
2	BRUN40.EXE BRUN440.LIB BQLB40.LIB LINK.EXE LIB.EXE BC.EXE
3	BCOM40.LIB LINK.EXE QB.LIB QB.BI*
4	BASIC文件 QB.QLB (可选) + Quick库

*: 该文件只有当程序使用 QB.LIB 进行 DOS 中断处理时才需要。见第 8 章“Quick 库”详细介绍。

*: Quick 库只有当程序用到 DOS 中断时才需要。见八章“Quick”详细介绍。

使用以上盘配置时，必须记住将 QB.EXE 放在 A 盘中，从 B 盘调用 Quick BASIC。这样便可保证你的程序能找到它所需的任何库，并且可执行文件将被写到 B 盘上。由 B 盘来调用 Quick BASIC 则要把 1 号盘放在 A 驱动器中，4 号盘放在 B 驱动器中。

1. 键入 b: 及回车 (ENTER) 使 B 盘为当前盘；

2. 键入 a: q b 引导 Quick BASIC；

在绝大多数时间里你只需 1 号盘和 4 号盘。下列一些情况中将需用 2 号盘和 3 号盘：

所 做 工 作	驱 动 器 配 置
在内存中写或运行程序需要 Quick 库时	放 1 号盘在 A 驱动器中，4 号盘在 B 驱动器中
在内存中写或运行程序需要 Quick 库时	放 1 号盘在 A 驱动器中，4 号盘在 B 驱动器中，且 4 号盘上必须含有 BASIC 源程序及它所需的 Quick 库

建 Quick 库或利用 EXE Requiring 建可执行程序由 Run 菜单的 MAKE EXE 文件命令来选择 BRUN40.EXE	放 1 号盘在 A 驱动器中, 4 号盘在 B 驱动器中。当出现“文件名没找到”的提示时, 再插 2 号盘到 A 驱动器中, 然后键入“a”并按回车。如果程序需用 Quick 库, 则必须把那张库盘插在 B 驱动器中。
由 RUN 菜单的 MAKE EXE File 命令来选择 Stand-Alone EXE File, 再利用它来建一个可执行程序	放 1 号盘在 A 驱动器中, 4 号盘在 B 驱动器中。当出现“文件名没找到”的提示时, 插入相应的盘在 A 驱动器中, 然后键入“a”并按回车。(第一个提示所对应的是 2 号盘, 即含有 BC.EXE 和 LINK.EXE 文件, 而下一个提示所对应的盘为 3 号盘, 即含有 BCOM40.LIB 文件)。最后提示的为含有 QB.EXE 文件的盘。

通过使用 Quick 库的程序来建独立的可执行文件是不可能的。如遇到要建独立的可执行文件这样一个问题时, 可通过把原始模块装入到 Quick BASIC 中去解决。

注: 如果你用的是单驱动器系统或 3.5 英寸的盘, 可查看 1 号盘上 README.DOS 文件关于盘配置的建议。

1.5 使用 MOUSE

运行 Quick BASIC 时既可使用 MOUSE, 或不用 MOUSE, 也可将 MOUSE 技术与键盘命令一起混合使用。

Microsoft Quick BASIC 在设计上是满足 Microsoft MOUSE 的要求。尽管许多厂商声称他们的指示设备 (Pointing devices) 是与 Microsoft MOUSE 相兼容的, 但只有在这些指示设备与 Microsoft MOUSE 功能完全一致时, Quick BASIC 才能与它们一起使用。因此在具体使用时要检查一下 MOUSE 制造厂家。

Quick BASIC 可与 MOUSE.COM 或 MOUSE.SYS6.00 版或更高版本一起使用。如果程序使用 MOUSE 功能调用, 那么就必须用 MOUSE.COM6.11 版, 该软件在 Quick BASIC3 号盘上提供。SETUP 程序自动地把该文件拷贝到硬盘上。如果没用 SETUP, 而又有一个 MOUSE, 则应拷贝与该 MOUSE 产品一起提供的 MOUSE.COM 对应版本文件。

1.6 使用协处理器

8087 和 80287 协处理器芯片提供了快速、精确的“浮点运算”功能。浮点运算是指非整数的算术运算, 如: $4.567 / 23.4^2$ 就是一个浮点运算。

如果有协处理器芯片 Microsoft Quick BASIC 就会自动的使用该器处理。否则, Quick BASIC 就采用软件程序来仿效处理器的功能。仿效协处理器导致执行速度比处理器芯片要慢些, 但具有与使用协处理器芯片相同的精确度。

注: 设置 NO87 参数可使 Quick BASIC 忽略现有的 8087 或 80287 协处理芯片。见 1.7.3 节“NO87 参数”。

使用协处理器（或采用软件来仿效）进行浮点运算通常比 BASIC 或 Quick BASIC 早先版本具有更高的精确度。

1.7 置 DOS 参数

如果你不想把所有 QB 文件存放在一个目录下，可把它们放在不同的目录下，然后用 DOS 里的 PATH 及 SET 命令来确定环境变量，以使 QB 知道去何处找它所需的文件。

设置环境变量对运行 QB 来说是有助的。若不设置变量，则 QB 只在当前目录下查找所需文件，如果找不到则显示“文件没找到，输入 PATH:”

这里是 QB 所需的文件：

文件类型和扩展名	描 述
可 执 行 文 件 (.EXE)	程序作为可执行文件处理。当在编程环境中运行时要包括 QB.EXE 文件。如果正在编译、连接独立程序或正在建库，则还要有 BC.EXE 连接程序、BRUN40.EXE 交替运行方式和 LIB.EXE 库管理程序。
库文件 (.LIB)	当用 .LIB、.EXE 连接时，需要 BRUN40.LIB、BCOM40.LIB、BQLB40.LIB

一旦设置了环境变量，则它就一直保持所设置的功能直到重新给它设置一个不同的值或重新启动或关机为止。1.7.1 和 1.7.2 两节简要介绍了两个最常用的命令—PATH 和 LIB，并说明了如何设置它们的值。如果有 8087 或 80287 协处理芯片，则为使程序忽略该片，可阅读 1.7.3 那关于 NO87 变量的设置。

1.7.1 PATH 命令

PATH 命令告诉编译程序在那儿找到可执行的文件。QB.EXE 和 BC.EXE 两程序均需查看 PATH 变量。

在 DOS 下用 PATH 命令来确定 PATH 变量。输入命令，一个空格或等号以及一至多个用分号(;) 隔开的路径说明。如下设置：

```
PATH=C: \BIN; C:\QB
```

它告诉了编译程序先到 C: 盘上的 \BIN 目录中查找，然后再到 C: 盘上 \QB 目录中查找那些可执行文件。

用该例所置的 PATH，你可把 QB.EXE 放在 \BIN 目录中，然后就可在任何目录下启动 Quick BASIC。

③

注：如果用户有 DOS3.0 版或更高版本，就可与 SET 命令一起来设定 PATH 变量；如下所示：

```
SET PATH=C: \QB
```

如果所用的 DOS 版本是 3.0 以前的，则不能与 SET 命令一起来设置 PATH 变量，而必须用 PATH 命令。

1.7.2 LIB 命令

LIB 命令告诉 QB 在何处可找到 qb 命令行上所指定的 Quick 库，以及告诉 LINK.EXE 在何处找到它所需的库文件。可用 SET 命令来设定 LIB 变量。如下例中的命令就告诉链接程序到 C: 盘上的 \LIB 目录中查找所有库文件:

```
SET LIB=C: \LIB
```

1.7.3 NO87 命令

如 1.6 节所述，若安装了一个 8087 / 80287 协处理器，则在程序运行时就要用到。然而你可通过设置 NO87 变量使程序仿效协处理器的功能，而不需使用协处理器本身。以下给出了设置 NO87 变量的两种方法:

```
SET NO87=USE of Coprocessor Suppressed
```

```
SET NO87= (一至多个空格)
```

当用程序来代替现有的 8087 / 80287 执行时，第一种设置就使得屏幕上显示出“取消协处理器的使用”信息，而第二种设置就不显示该信息。

设置 NO87 等于空值，则为强行关掉协处理器仿真。如下例:

```
SETNO87=
```

注意: 该例中等号 (=) 后面无空格。

第二章 初识 Quick BASIC

用户可根据自己以前用 BASIC 编程的经历, 为自己选择一种掌握如何使用 Quick BASIC 的途径。

经 验	所需使用章节
没有 BASIC 编程经验, 或有其它解释 BASIC 编程经验但未用过 Quick BASIC	2.1 节“假如你是个 BASIC 程序员”2.4 节“Quick BASIC 用户不可缺少的知识”
有用其它 Quick BASIC 版本编程的经验	2.2 节“假如你已使用过 2.0 或 3.0 版的 Quick BASIC”2.4 节“Quick BASIC 用户须知”

当读完本章, 你就会知道怎样:

- 启动 Quick BASIC
- 装载 Quick BASIC 程序
- 在内存中运行 Quick BASIC 程序
- 在盘上建一个可执行文件 (即带有 .EXE 扩展名的文件)
- 退出 Quick BASIC

2.1 作为 BASIC 程序员

如果用户从未用 BASIC 编过程序, 或者是个解释 BASIC 或 BASIC 程序员, 但从未使用过 Quick BASIC, 则从此入手。如果用户已具备用 Quick BASIC 的其它版本编程的经验, 则可从 2.2 节继续。如果具有 Microsoft 编译 BASIC 的丰富经验, 用户就可把本节与下一节作为 Quick BASIC 程序开发环境介绍来阅览或直接阅读 2.3 节。

在本节练习开始之前, 应按照第一章所述把 Quick BASIC 安装好, 同时还需一个 BASIC 程序。可先从你自己的 BASIC 程序或 Quick BASIC 3 号盘上所提供的 DEM01.BAS 程序着手。

首先试一个很简单的程序。某些语句和功能在 Quick BASIC 中运行与在 BASIC 中运行有一点不同, 所以如果用自己的程序, 则在运行前要先参考附录 A: “BASIC 程序到 Quick BASIC 的转换”。所选的程序应产生一些输出, 以便知道它否运行正常。

注: 如果使用自己的 BASICA 程序, 则必须保证程序是 ASCII 格式。因为 Quick BASIC 不能装入二进制形式的 BASICA 程序。

如果不清楚 BASICA 程序是否是 ASCII 格式, 则可用 DOS 中的 TYPE 命令来查看一下。如果显示出的与它在 BASIC 中所显示的形式一样, 那么该程序就是用 ASCII 形式保存的。而用二进制形式保存的程序, 在用 TYPE 显示时, 屏幕上就会出现许多非法字符。

如果程序是二进制形式, 则在运行 Quick BASIC 前要先做以下几步:

1. 键入: basic 启动 BASICA
2. 键入: Load "sample" 装入程序 (假定程序名为 sample.BAS)
3. 键入: save "sample", a 重新用 ASCII 形式来保存文件
4. 键入: system 退出 BASICA

2.1.1 Quick BASIC 的启动

如果使用的是彩色监视器或带有单色适配器的监视器, 则启动 Quick BASIC 时需键入: qb 及按 ENTER (回车键)。

Quick BASIC 能测出显示适配器的类型, 但并不知道监视器的类型。如果用的是 CGA 黑白监视器, 则启动 Quick BASIC 需键入: qb / b 及按 ENTER。

如果 Quick BASIC 不能启动, 则必须把 QB.EXE 文件放在当前目录中或用 PATH 变量指明在哪个目录中, 然后再试一次。

当 Quick BASIC 启动后, 你就可装入第一个程序了。

2.1.2 装入程序

在运行一个程序前, 必须先把程序装入 Quick BASIC。下面就是装入一个程序的步骤:

1. 按 ALT 键, 则在屏幕左上角的 "File" 字就变成高亮度了, 即已选中。注意: 不要一直按 ALT 键。当使用菜单工作时, 在进行第二步前你可按下和松开 ALT 键。
2. 按下 F 键以显示 File 菜单中的所有命令 (如图 2.1 所示)。见第三章 "进入 Quick BASIC 环境" 关于菜单的详细说明。

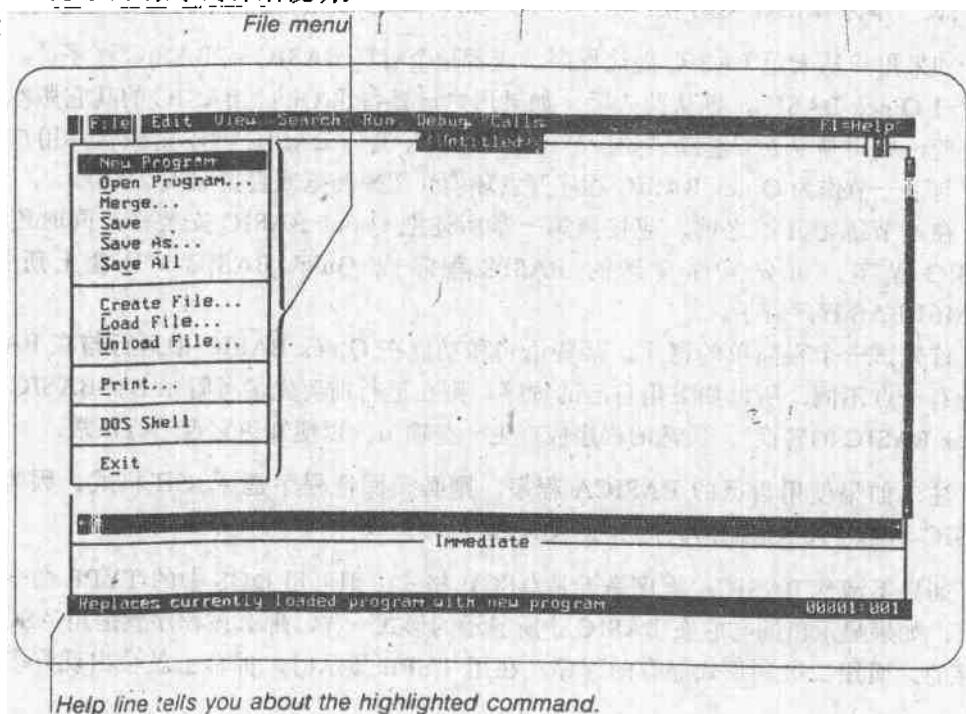


图.2.1 文件菜单

3. 按下 O 键以选择 File 菜单的打开程序命令。

这就引出了打开程序“对话框”，如图 2.2 所示。对话框就是输入一个命令所需的信息，此时打开程序命令所需的信息就是你所要装入文件的名字。

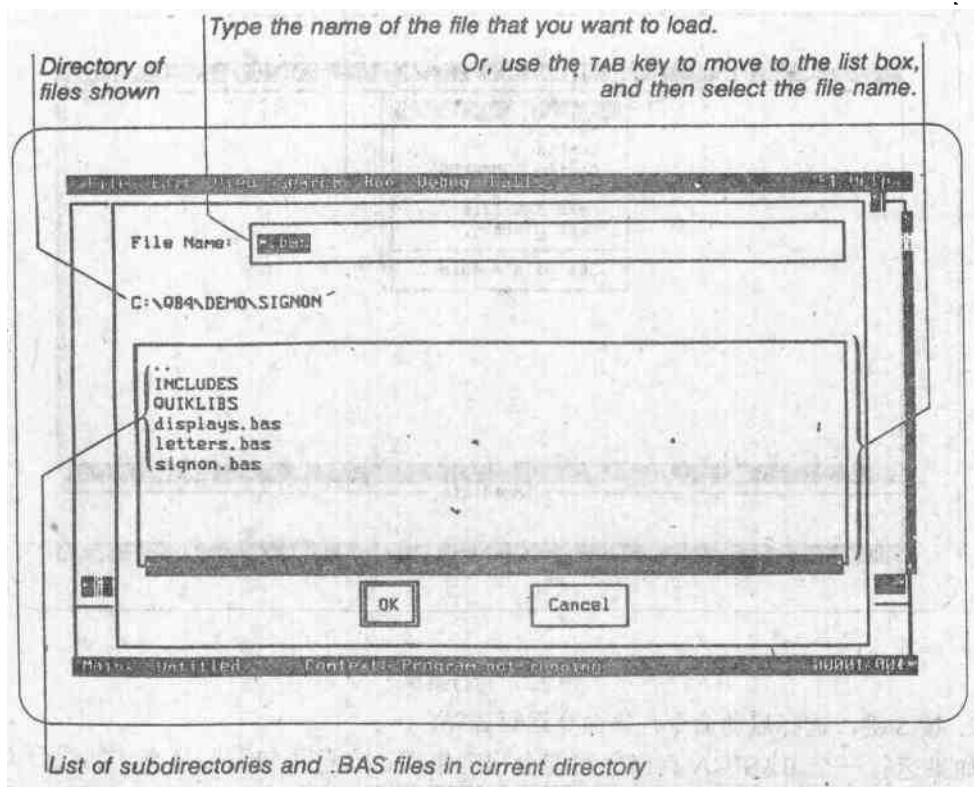


图 2.2 打开程序对话框

屏幕上所示的可能与图 2.2 不同，这是因为用户盘上有不同的文件及目录，见第四章“管理源文件”关于打开程序对话框的详细介绍。

4. 键入装入文件的文件名（如果文件在另一个目录中还要键入文件的目录路径）以及回车键，这样就能在对话框中 File Name 的下面看到你所键入的文件名。

5. 如果装入的是个 BASIC 程序，则可能看到一个错误信息，这多半是因为该程序是用二进制形式保存的。如果看到错误信息，则按 ESC 键清除屏幕上的错误信息，然后重复 1—4 步，但这时要装入 Quick BASIC3 号盘上所提供的 DEM01.BAS 程序。

2.1.3 进行程序

现在程序已装好，下面就可准备运行了。本节介绍如何在 Quick BASIC 环境内运行程序。使用 Quick BASIC 的过程——“解释执行”——与运行 BASICA 程序很相似。以下步骤就是运行 Quick BASIC 程序的步骤：

1. 按 ALT-R 键，打开运行菜单。如图 2.3 所示：