

(共三册)

Microsoft C 5.0

技 术 丛 书

(中册)

■ Code View 调试器使用说明

和 实 用 程 序

■ 混合语言程序设计指南

■ Quick C 使用说明

北京联想计算机集团公司

一九九〇年四月

译 校 者 序 言

翻 译: 宗丽苹 吴 倩 邦继明 鲁 倩 王 越 徐砥祥 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。

参与本书印排工作的除中国科学院软件所、计算所、自动化所之外,还有祥云电脑公司、海声软件开发公司,故上述单位共有此中文资料版权。任何其它单位不得随意翻印此书。

目 录

简 介

1.1 CodeView 调试器的新特色.....	(1)
1.2 关于本手册.....	(2)
1.3 符号的约定.....	(3)

第一部分 CodeView 调试器

第一章 启动.....	(7)
1.1 限制	(7)
1.2 准备 CodeView 要用的程序.....	(7)
1.2.1 编程 考虑.....	(8)
1.2.2 CodeView 编译任选项.....	(8)
1.2.3 CodeView 连接任选项.....	(9)
1.2.4 准备 C 程序	(9)
1.2.5 准备 Fortran 程序.....	(10)
1.2.6 准备 Basic 程序.....	(11)
1.2.7 准备 Pascal 程序.....	(12)
1.2.8 准备汇编语言 程序.....	(13)
1.3 启动 CodeView 调试器	(15)
1.4 CodeView 但选项的使用.....	(17)
1.4.1 使用双视屏适配器.....	(18)
1.4.2 使用增强型图形适配器的43行显示 方式.....	(18)
1.4.3 使用黑白屏幕 显示.....	(19)
1.4.4 给定启动 命令.....	(19)
1.4.5 管理中断 陷阱.....	(20)
1.4.6 使用扩展型存 贮器.....	(20)
1.4.7 设定屏幕切换 方式.....	(20)
1.4.8 关闭鼠 标.....	(22)
1.4.9 扩展 EGA 兼容性.....	(22)
1.4.10 启用 窗口或顺序方式.....	(23)
1.5 调试大程序	(23)
1.6 和汇编程序的旧版本一起 工作.....	(23)

第二章 CodeView 的显示	(25)
2.1 启用窗口方式	(25)
2.1.1 通过键盘执行窗口命令	(26)
2.1.1.1 用键盘命令移动光标	(26)
2.1.1.2 用键盘命令改变屏幕	(27)
2.1.1.3 用键盘命令控制程序执行	(27)
2.1.1.4 用键盘从菜单中选择	(28)
2.1.2 用鼠标执行窗口命令	(29)
2.2.2.1 用鼠标改变屏幕	(29)
2.2.2.2 用鼠标控制程序执行	(30)
2.2.2.3 用鼠标从菜单中选择	(31)
2.1.3 使用菜单选择项	(32)
2.1.3.1 使用 File 菜单	(32)
2.1.3.2 使用 View 菜单	(33)
2.1.3.3 使用 Search 菜单	(34)
2.1.3.4 使用 Run 菜单	(35)
2.1.3.5 使用 Watch 菜单	(36)
2.1.3.6 使用 Options 菜单	(37)
2.1.3.7 使用 Language 菜单	(38)
2.1.3.8 使用 Calls 菜单	(38)
2.1.3.9 使用 Help 菜单	(39)
2.1.4 使用 Help 系统	(39)
2.2 使用顺序方式	(39)
第三章 使用会话命令	(41)
3.1 键入命令和参数	(41)
3.1.1 使用特殊键	(41)
3.1.2 使用命令缓冲区	(41)
3.2 CodeView 的命令和参数的格式	(42)
第四章 CodeView 表达式	(43)
4.1 C 表达式	(43)
4.1.1 标识符	(43)
4.1.2 常数	(44)
4.1.3 C 串	(45)
4.2 FORTRAN 表达式	(45)
4.2.1 FORTRAN 符号	(46)
4.2.2 FORTRAN 常数	(46)

4.2.3	FORTTRAN 串	(47)
4.2.4	FORTTRAN 内部函数	(47)
4.3	BASIC 表达式	(48)
4.3.1	BASIC 符号	(49)
4.3.2	BASIC 常数	(49)
4.3.3	BASIC 串	(51)
4.3.4	BASIC 内部函数	(51)
4.4	Pascal 表达式	(51)
4.4.1	Pascal 标识符	(52)
4.4.2	Pascal 常数	(52)
4.4.3	Pascal 串	(53)
4.4.4	Pascal 内部函数	(53)
4.5	汇编表达式	(54)
4.6	行标号	(55)
4.7	寄存器和地址	(55)
4.7.1	寄存器	(56)
4.7.2	地址	(56)
4.7.3	地址域	(57)
4.8	存储器操作	(57)
4.8.1	存取字节 (BY)	(58)
4.8.2	存取字 (WO)	(58)
4.8.3	存取双字 (DW)	(59)
4.9	转换表达式识别器	(59)
第五章	执行代码	(61)
5.1	Trace (追踪) 命令	(61)
5.2	Program Step (程序步) 命令	(63)
5.3	Go (走) 命令	(65)
5.4	Execute (执行) 命令	(67)
5.5	Restart (重启) 命令	(68)
第六章	检查数据和表达式	(69)
6.1	表达式显示命令	(69)
6.2	符号检查命令	(76)
6.3	内存显示命令	(80)
6.3.1	内存显示	(82)
6.3.2	内存字节显示	(82)
6.3.3	内存 ASCII 字符显示	(82)
6.3.4	内存整数显示	(83)

6.3.5	内存无符号整数显示	(83)
6.3.6	内存字显示	(84)
6.3.7	内存双字显示	(84)
6.3.8	内存短实数显示	(85)
6.3.9	内存长实数显示	(85)
6.3.10	内存10字节实数显示	(85)
6.4	存储器比较命令	(86)
6.5	搜索存储器查寻命令	(87)
6.6	端口输入命令	(88)
6.7	寄存器命令	(88)
6.8	8087命令	(90)
第七章	断点管理	(92)
7.1	Breakpoint Set (断点设置)命令	(92)
7.2	Breakpoint Clear (断点清除)命令	(94)
7.3	Breakpoint Disable (断点屏蔽)命令	(95)
7.4	Breakpoint Enable (断点启动)命令	(95)
7.5	Breakpoint List (断点列表)命令	(96)
第八章	Watch (观察) 语句的管理	(98)
8.1	设置表达式观察和内存观察语句	(99)
8.2	设置观察点	(101)
8.3	设置追踪点	(103)
8.4	删除观察语句	(106)
8.5	列出观察点和追踪点	(107)
8.6	C的实例	(108)
8.7	FORTRAN的实例	(108)
8.8	Pascal的实例	(109)
8.9	汇编的实例	(110)
第九章	检查代码	(112)
9.1	Set Mode (方式设置)命令	(112)
9.2	Unassemble (逆汇编)命令	(113)
9.3	View (查看)命令	(115)
9.4	Current Location (当前定位)命令	(117)
9.5	Stack Trace (栈追踪)命令	(118)
第十章	代码和数据的修改	(121)
10.1	汇编命令	(121)

10.2	键入命令	(123)
10.2.1	键入命令	(126)
10.2.2	键入字节命令	(126)
10.2.3	键入 ASCII 字符命令	(126)
10.2.4	键入整数命令	(127)
10.2.5	键入无符号整数命令	(127)
10.2.6	键入字命令	(128)
10.2.7	键入双字命令	(128)
10.2.8	键入短实数命令	(129)
10.2.9	键入长实数命令	(129)
10.2.10	键入 10 字节实数命令	(129)
10.3	内存存入命令	(130)
10.4	内存转移命令	(131)
10.5	端口输出命令	(131)
10.6	寄存器 (Register) 命令	(132)
第十一章	系统控制命令的使用	(135)
11.1	帮助命令	(135)
11.2	退出命令	(135)
11.3	置基数命令	(136)
11.4	重新画屏命令	(137)
11.5	屏幕转换命令	(138)
11.6	查寻命令	(138)
11.7	Shell 调用 (Shell Escape) 命令	(140)
11.8	置制表符命令	(141)
11.9	选择项命令	(142)
11.10	重定向命令	(143)
11.10.1	Code View 输入重定向	(144)
11.10.2	Code View 输出重定向	(144)
11.10.3	Code View 输入输出重定向	(145)
11.10.4	与重定向有关的命令	(145)
11.10.4.1	注解命令	(145)
11.10.4.2	延迟命令	(146)
11.10.4.3	暂停命令	(146)

第二部分 实用程序

第十二章	使用 LINK 连接目标文件	(151)
12.1	指定连接文件	(151)

12.1.1	指定文件名	(151)
12.1.2	使用 LINK 命令行进行连接	(152)
12.1.3	使用 LINK 提示信息进行连接	(153)
12.1.4	使用回答文件进行连接	(154)
12.1.5	LINK 如何查寻库文件	(155)
12.1.6	LINK 的内存要求	(156)
12.2	指定连接器的可选项	(157)
12.2.1	介绍可选项表 (/HE)	(157)
12.2.2	连接时的暂停 (/PAU)	(158)
12.2.3	显示连接器的运行信息 (/Z)	(158)
12.2.4	压缩可执行文件 (/E)	(159)
12.2.5	公共符号列表 (/M)	(159)
12.2.6	映射文件中包含行号 (/LI)	(160)
12.2.7	保持区分大小写 (/NOI)	(160)
12.2.8	忽略缺省库 (/NOD)	(160)
12.2.9	控制栈大小 (/ST)	(160)
12.2.10	设置最大分配空间 (/CP)	(161)
12.2.11	设置段数最大值 (/SE)	(161)
12.2.12	设置覆盖中断 (/O)	(162)
12.2.13	段的排序 (/DO)	(162)
12.2.14	控制数据的装入 (/DS)	(162)
12.2.15	控制可执行文件的装入 (/HZ)	(163)
12.2.16	保持兼容性 (/NOG)	(163)
12.2.17	为调试作准备 (/CO)	(163)
12.2.18	以批方式运行 (/B)	(163)
12.2.19	优化长调用 (/F)	(164)
12.2.20	屏蔽长调用的优化 (/NOF)	(165)
12.2.21	压缩相邻段 (/PAC)	(165)
12.2.22	屏蔽段的压缩 (/NOP)	(166)
12.2.23	指定 Quick Language 的用户库	(166)
12.3	通过 LINK 环境变量选择可选项	(166)
12.4	连接器操作	(167)
12.4.1	段的对齐	(167)
12.4.2	构等号	(168)
12.4.3	段的顺序	(168)
12.4.4	段的结合	(168)
12.4.5	组	(168)
12.4.6	修正	(169)
12.5	覆盖的使用	(169)

12.5.1	覆盖的限制	(170)
12.5.2	覆盖管理器的提示	(170)
第十三章	使用LIB 管理库	(172)
13.1	库的管理	(172)
13.1.1	使用LIB 命令行 管理库	(172)
13.1.1.1	指定库文件	(173)
13.1.1.2	指定页尺寸	(173)
13.1.1.3	LIB 命令域	(173)
13.1.1.4	建立交叉参照列表	(173)
13.1.1.5	指定输出库	(174)
13.1.2	使用LIB 提示 管理库	(175)
13.1.2.1	行的扩充	(176)
13.1.2.2	对提示的缺省回答	(176)
13.1.3	使用回答文件管理库	(176)
13.1.4	终止LIB 命令 的执行	(177)
13.2	使用LIB 执行库管理任务	(177)
13.2.1	建立库文件	(177)
13.2.2	修改库文件	(178)
13.2.3	添加库文件	(178)
13.2.4	删除库文件	(178)
13.2.5	替换库文件	(178)
13.2.6	复制库文件	(179)
13.2.7	移动库文件 (选取)	(179)
13.2.8	结合库	(179)
13.2.9	建立交叉参照列表	(179)
13.2.10	完成一致性 检查	(179)
13.2.11	设置库的页 尺寸	(180)
第十四章	用MAKE 维护程序	(181)
14.1	使用MAKE	(181)
14.2	建立MAKE 描述 文件	(181)
14.3	自动程序开发	(184)
14.4	运行MAKE	(185)
14.5	指定MAKE 选项	(185)
14.6	宏定义的使用	(186)
14.6.1	定义和使用宏	(186)
14.6.2	嵌套宏定义	(187)
14.6.3	使用特殊的宏定义	(187)

14.7 定义推导规则.....	(188)
第十五章 使用 EXEPACK, EXEMOD, SETENV 及 ERROUT	(190)
15.1 使用 EXEPACK, 压缩可执行文件.....	(190)
15.2 使用 EXEMOD 修改程序标题.....	(191)
15.3 使用 SETENV 扩充 DOS 环境空间.....	(193)
15.4 使用 ERROUT 为错误输出重新定向.....	(194)

附 录

A. 正规表达式	(196)
A.1 引言.....	(196)
A.2 正规表达式中的特殊字符.....	(196)
A.3 特殊字符的查寻.....	(196)
A.4 使用点号.....	(197)
A.5 使用方括号.....	(197)
A.5.1 使用方括号中的减号.....	(197)
A.5.2 使用方括号中的插入符.....	(197)
A.5.3 匹配括号中的括号.....	(198)
A.6 使用星号.....	(198)
A.7 匹配一行的头或尾.....	(198)
B. 出器码	(199)
B.1 MAKE 与出口码.....	(199)
B.2 DOS 批处理文件与出口码.....	(199)
B.3 几个实用程序的出口码.....	(200)
B.3.1 Code View 出口码.....	(200)
B.3.2 LINK 出口码.....	(200)
B.3.3 LIB 出口码.....	(200)
B.3.4 MAKE 出口码.....	(200)
B.3.5 EXEPACK 出口码.....	(200)
B.3.6 EXEMOD 出口码.....	(200)
B.3.7 SETENV 出口码.....	(201)
B.3.8 ERROUT 出口码.....	(201)
C. 错误信息	(201)
C.1 Code View 错误信息.....	(201)
C.2 Link 错误信息.....	(207)

C.3 LIB 错误信息.....(214)

C.4 MAKE 错误信息.....(216)

C.5 EXEPACK 错误信息.....(218)

C.6 EXEMOD错误信息.....(219)

C.7 SETENV 错误信息.....(220)

C.8 ERROUT 错误信息.....(221)

简介

欢迎使用 Microsoft CodeView 调试器和开发用的实用程序。这些都是可执行程序，帮助你开发用 Microsoft BASIC、C、FORTRAN、Pascal 编译器及 Microsoft 宏汇编器写的软件

Microsoft CodeView 调试器是一个强有力的基于窗口的工具，它可使你跟踪程序中的逻辑错误，使你能够在程序真正运行时分析它。CodeView 调试器可显示源代码或汇编代码，指示将要运行的行，动态检查变量值（局部变量或全局变量），切换屏幕以显示程序输出，以及执行其它相关的功能，汇编语言的程序员和高级语言的程序员都可以很容易地学会和使用调试器。

实用程序对软件开发的阶段都很重要，在你用编译器或汇编器生成一个或多个目标文件时，要用 LINK 来产生一个可执行文件。（当一个程序变成可执行文件时，它最终所处的形式是能被 DOS 装入和运行），在 LINK 的过程中，你可能会使用软件库，实用程序 LIB 使你可以生成、检查和维护这些库，用实用程序 MAKE，可在相当大的程度上自动实现编译和连接，MAKE 保持对改变过的源文件的追踪并只执行必要的命令来修改程序。

其它实用程序帮助你维护生成的可执行文件，你可用 EXEPACK 来减小存在磁盘中的文件的大小，可用 EXEMOD 来检查或修改文件的标题。可执行文件的标题显示了栈大小、装入模块大小。以及其它 DOS 每次运行文件时要用的重要项。

最后，你可用实用程序 SETENV 和 ERROUT 来修改你的 DOS 环境。

1.1 CodeView 调试器的新特性

- 多种语言表达式的鉴别

CodeView 有一个插入的语言解释器可鉴别 C、BASIC、FORTRAN 或 Pascal 表达式。

- 386 支持

CodeView 现在可支持专为 386 处理器所写的代码的调试，你可对 386 指令进行译码和汇编，也可查看 386 寄存器。

- 扩展的存储器支持

如果你已扩展了存储器，那么你可以减少大量调试一个程序所需的查存空间，许多过去太大的程序现在可以和 CodeView 调试器一起运行。

- 8087 仿真器支持

如果你没有 8087 协同处理器，你可以连接到一个 Microsoft 仿真程序库上，使用 7 个命令，调试器会显示伪 8087 的寄存器，就好像在你的机器中确实有一个数学协同处理器。

- 重叠和库模块

调试器完全和重叠使用的程序相容，你还可以调试库模块。

● 新的命令

在 CodeView 调试器的命令表中增加了 SYMDEB (符号调试器) 命令 Compare, Fill, Move 和 Input, Output, option 命令提供了更强的输入重定向和启动命令的功能。

1.2 关于本手册

此手册打算作为 Microsoft 语言手册的参考书。它不是一种语言的说明书,除了有些地方需要一些例子之外,而在这种情况下,有代表性地给出了几种语言的实例。

本手册分两部分,最后是附录。第一部分(由 1-11 章组成)解释如何使用 CodeView 调试器来检查和定位程序错误;第二部分(由 12-15 章组成)说明如何使用各个实用程序(包括 LINK、LIB、MAKE、EXEPACK、EXEMOD 和 SETENV)。手册最后的附录讨论 CodeView 和所有实用程序的返回码和错误信息。

下来说明如何在本手册中找到你需要的不同信息。这个表决不是详尽无遗的,而只是作为一个起点,尤其对 CodeView 调试器的新用户来说。

信 息 位 置

检查和定位 程序错误	第一部分:“CodeView 调试器”以第 1 至 11 章的顺序进行描述以帮助你跟踪程序的错误,分析运行中的程序。在此手册的最后部分讨论了返回码和错误信息。
启动一个调试期	第一章,“启动”告诉你如何编译和连接程序从而把它们和调试器一道运行。该章还解释了每个 CodeView 命令行的选择项。
使用 CodeView 界面	第二章,“CodeView 的显示”描述如何使用 CodeView 窗口,弹出菜单和鼠标。
确定 CodeView 命令	第三章,“使用会话命令”给出命令的一般形式。第四章 Code View 表达式描述如何构成命令中用的复合表达式
控制你的程序的执行	第五章,“运行代码”解释用 CodeView 调试器控制程序运行的基础。
查看变量或表达式的值	第七章,管理断点解释如何用断点来暂停程序。
使用实用程序	第六章,“检查数据和表达式”说明如何显示值;第八章“Watch 语句的管理”说明如何在窗口中置变量以观察它们在程序运行中值的改变。
生成可执行文件	第二部分:“实用程序”在第 12 章 15 章描述用于产生和维护可执行文件以及其它任务的各种实用程序。在本手册最后部分的附录中讨论了实用程序的返回码和错误信息。
管理软件库	第十二章“用 LINK 连接目标文件”
有几个模块的自动规划	第十三章“用 LIB 管理库”
使用其它实用程序	第十四章“用 MAKE 进行自动程序开发”
	第十五章“使用 EXEPACK, EXEMOD, SETENV, ERROUT”

为 CodeView 的 Search 命令确定表达式	附录 A: “常规表达 A”
各实用程序返回给 DOS 的代码	附录 B: “返回码”
错误信息表	附录 C: “错误信息”

注意:

关于 CodeView 调试器的其他信息可在文件 README.DOC 中找到。这一文件中包括本手册定稿后, 所有对调试器本身以及使用方法的改动。

在本手册中, “DOS” 用来指 MS-DOS 和 PC-DOS 两者, 除了有时一些值得在意的特性是其中之一特有的。

1.3 符号的约定

下面的符号约定是本手册使用的, 尤其适用于句法显示。

约定的实例 约定的描述

关键项	黑体字母表示那些打算按字面意义使用的专用名或标点符号: 如语言的关键字 (如 IF), Microsoft 产品发布的文件名 (如 LINK) 以及命令行选项 (如 /Zi) 这些项必须准确地按给定的顺序键入才会生效。但是, 使用上档键或下档键并不总有效。例如, 你可以用下列方式调用连接器: 用 LINK, Link 或 Link 的任一种响应 DOS 的提示。
位置标志符	斜体字表明一般性质的信息, 期待你提供一个实际的值。例如考虑 CodeView 置基数命令的语法显示: Nnumber 此语法显示要求你通过输入一个 N 来输入置基数命令, 紧接着是 number 的值例如你可以键入 N8; 但你不可以键入 “number” 字本身。
实例	实例的显示采用一种无比例的铅字从而使它们看起来更象计算机监视器屏幕或打印机的输出。当显示同时包括用户输入和命令输出时, 输入用粗体字表示, 输出用常规的, 非粗体字表示: <pre>>RAX AX 0041 : 43 ></pre>
程序 :	在程序实例中用竖的省略号表示一段程序被省略。例如在下面的选录中, 显示了三个语句, 语句之间的省略号表明中间有程序行存在, 但没有显示。

片段	<pre>Count = 0 : PASS = PASS + 1 : Count = 0</pre>
[[选择项]]	双括号里包含命令行和任选项语法中可以任选的部分。例如在下列命令行语法中: R[[register]][[:]]value]]外面的两对双括号表示你可以键入一个寄存器或一个值。等号“=”表明你应该在值的前面放一个等号,但仅仅是当你给定值时。
[[选择1 选择2]]	竖杠表明你可键入竖杠两边的任何一项,下面的命令行句法说明了竖杠的用法: <pre>DB[[address range]]</pre> 竖杠表示在显示内存字节命令(DB)中,你可以给出address(地址),也可给出range(范围)。因为它们两者都在双括号中,你可以不加参数地使用命令。
定义项	引号表明正文中定义的项,例如“Watch Point”项在它第一次定义时是用引号给出的
键名	白体大写字母用于键名和键组合名,如 ENTER, CONTROL + C, T + F
屏幕示例	屏幕示例以黑白排印,它和你用单色监视器,或在 CodeView 命令行中选用/B 任选项所得到的屏幕相似。(看1.4.3节:“启用黑白屏幕显示”)

第一部分 CodeView 调试器

第一部分解释 CodeView 调试器的使用，在此给出了调试器的命令、显示和接口，其余相关的部分（如错误信息和返回码）在附录中给出。

第一章说明如何生成一个可以和 CodeView 调试器一起运行的 C、FORTRAN、BASIC、Pascal 或汇编等语言编写的程序，本章还解释了如何启动调试器和选择命令行任选项。

第二章讨论了 CodeView 的屏幕显示和接口，包括功能键，键盘命令和鼠标。

第三至十一章描述了如何使用 CodeView 命令和表达式。

