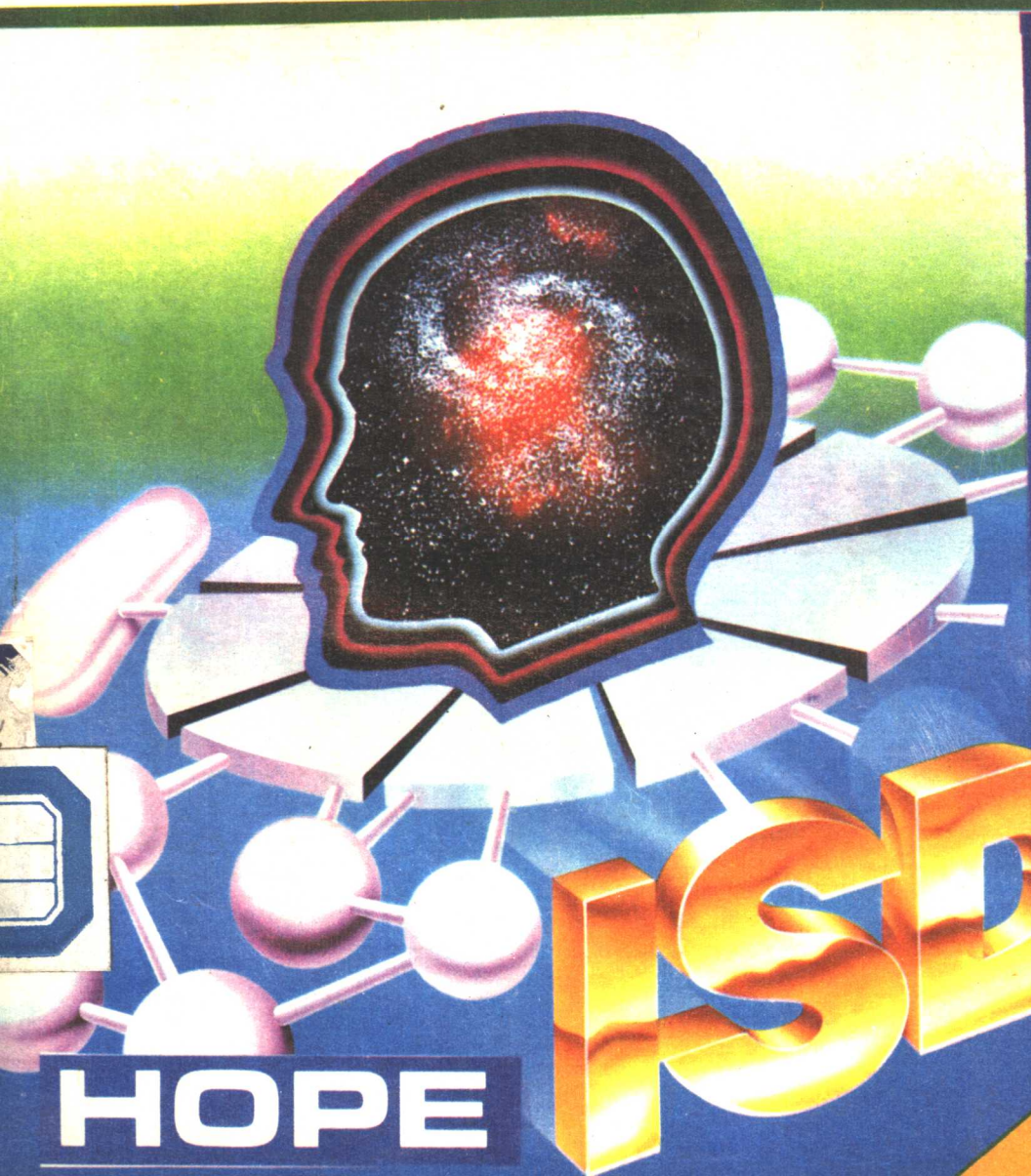




(共三册)

5.0 版

Microsoft FORTRAN 调试工具CodeView 使用指南



北京希望电脑公司

HOPE

3

Microsoft FORTRAN 5.0

调试工具 CodeView 使用指南 (共三册)

王 乃 文 编译

- 强大的图形函数包
- 混合语言编程
- 支持源级调试(CodeView)
- 支持 OS/2 性能
- 更丰富的控制结构
- 更好的可移植性

Microsoft FORTRAN 5.0 必备资料:

《Microsoft FORTRAN 5.0 程序员参考手册》

《Microsoft FORTRAN 5.0 用户手册》

《Microsoft FORTRAN 5.0 调试工具 CodeView 使用指南》

3

中国科学院希望高级电脑技术公司

一九九一年八月

版权所有

不许翻印

违者必究

■ 北京市新闻出版局

准印证号 3314 — 90314

■ 订购单位：北京 8721 信箱资料部

■ 电 话：2562329

■ 电 传：01—2561057

■ 电 挂：0755

■ 地 址：海淀影剧院北侧

■ 乘 车：320、332、302路海淀黄庄下车

■ 办公地点：公司大楼 101 房间

前 言

Microsoft FORTRAN 5.0 是运行在各种微型计算机上的最新 FORTRAN 版本,它吸取了其它语言的优点,克服了一般 FORTRAN 语言数据结构种类少、语句的格式呆板、控制结构不灵活、编译器的灵活性差和不支持图形等缺点,在与已经普及的 Microsoft FORTRAN 语言兼容的基础上增进和增加了若干重要的功能,如:

- 第一次提供了 FORTRAN 图形库;库中包括了丰富的绘图函数和字体输出函数。
- 支持所有 IBMSAA 扩展,包含许多 VAX 扩展。SAA 和 VAX 上的 FORTRAN 应用程序可以方便地移植到 MS-DOS 计算机。
- 支持 OS/2 系统,包括动态连接库与多流执行。程序可以在 DOS 和 OS/2 下运行,具有很好的灵活性。
- 允许对源程序进行条件编译。
- 提供了丰富的接口,可以方便地进行混合语言编程。
- 提供了某些新的数据结构,如复合数据类型(结构与记录)。
- 提供了若干新的控制结构,如 SELECT CASE 分支语句和 DO WHILE 循环语句等。
- 实用程序 CodeView 提供 FORTRAN 源程序的源级调试。
- 此外,还增加了许多新的函数与过程,如 EPSILON 和 HUGE 等。

Microsoft FORTRAN 5.0 的参考资料包括《Microsoft FORTRAN 5.0 程序员参考手册》、《Microsoft FORTRAN 5.0 用户手册》和《Microsoft FORTRAN 5.0 调试工具 CodeView 使用指南》。

《Microsoft FORTRAN 5.0 程序员参考手册》介绍 FORTRAN 语言的基本元素、程序结构和输入/输出系统,详细讲述了每条语句的用法,说明了内部函数和附加过程;此外,还详细解释了语言的每个元命令及编译与连接选项的用法。

《Microsoft FORTRAN 5.0 高级程序设计》讨论 FORTRAN 程序设计的高级技术,包括浮点运算、内存模式、OS/2 下的编程、混合语言编程、语言之间的数据传递、记录结构和程序优化等,该书着重介绍 Microsoft FORTRAN 5.0 图形库,包括每个绘图函数、字体函数的使用方法,并以示例相辅。

《Microsoft FORTRAN 5.0 调试工具 CodeView 使用指南》介绍调试 FORTRAN 源程序的工具 CodeView 的命令和使用方法。利用 CodeView 能监视和修改 FORTRAN 的变量,在源程序行上设置断点,单步或多步跟踪源程序的执行等等。CodeView 能极大地提高 FORTRAN 的编程效率。

目 录

前言

第一章 Microsoft FORTRAN 5.0 CodeView 简介	1
1.1 引言	1
1.2 概况	1
1.3 关于本手册	2
1.4 记号约定	3
1.5 定义	4
1.6 掌握源程序级特征	5
第二章 准备启动	7
2.1 引言	7
2.2 启动演示程序	7
2.3 为 CodeView 调试程序准备 FORTRAN 程序	7
2.3.1 编写 FORTRAN 源代码	7
2.3.2 为 CodeView 调试程序编译源文件	8
2.3.3 为 CodeView 调试程序连接目标文件	8
2.4 启动 CodeView 调试程序	9
2.5 使用 CodeView 选择项	10
2.5.1 启动黑白显示	12
2.5.2 说明启动命令	12
2.5.3 设置屏幕切换方式	13
2.5.4 启用窗口方式或顺序方式	14
2.5.5 关闭鼠标器	15
2.5.6 处理中断陷阱	15
2.5.7 使用增强型图形适配器的 43 行显示方式	16
2.5.8 使用两个显示器	16
2.6 将 CodeView 调试程序用于宏汇编程序	16
第三章 CodeView 显示	18
3.1 引言	18
3.2 使用窗口方式	18
3.2.1 用键盘执行窗口命令	19
3.2.1.1 用键盘命令移动光标	19
3.2.1.2 用键盘命令修改屏幕	20
3.2.1.3 用键盘命令控制程序执行	21
3.2.1.4 用键盘选取菜单	22
3.2.2 用鼠标器执行窗口命令	22
3.2.2.1 用鼠标器修改屏幕	23

3.2.2.2	用鼠标器控制程序的执行	23
3.2.2.3	用鼠标器选择菜单	24
3.2.3	使用菜单选择项	25
3.2.3.1	使用 File 菜单	25
3.2.3.2	使用 Search 菜单	26
3.2.3.3	使用 View 菜单	28
3.2.3.4	使用 Run 菜单	29
3.2.3.5	使用 Watch 菜单	29
3.2.3.6	使用 Options 菜单	31
3.2.3.7	使用 Calls 菜单	33
3.2.4	使用帮助(help)系统	34
3.3	使用顺序方式	35
第四章	使用会话命令	37
4.1	引言	37
4.2	输入命令和参数	37
4.2.1	使用特殊键	37
4.2.2	使用命令缓冲区	38
4.3	CodeView 命令和参数的格式	38
4.4	表达式	38
4.4.1	符号	40
4.4.2	常数	41
4.4.3	寄存器	41
4.4.4	地址	42
4.4.5	范围地址	43
4.4.6	行号	43
4.4.7	串	44
4.4.8	切换表达式求值程序	44
第五章	执行代码	46
5.1	引言	46
5.2	跟踪命令(Trace)	46
5.3	程序步命令	48
5.4	运行命令(Go)	49
5.5	执行命令(Execute)	51
5.6	重新启动命令(Restart)	52
第六章	检查数据和表达式	53
6.1	引言	53
6.2	显示表达式命令	53
6.3	检查符号命令	56
6.4	转贮命令	58
6.4.1	转贮	59

6.4.2	字节转贮	59
6.4.3	ASCII 转贮	60
6.4.4	整数转贮	60
6.4.5	无符号整数转贮	61
6.4.6	字转贮	61
6.4.7	双字转贮	62
6.4.8	短实数转贮	62
6.4.9	长实数转贮	62
6.4.10	十字节实数转贮	63
6.5	寄存器命令	63
6.6	8087 命令	64
第七章	管理断点	67
7.1	引言	67
7.2	断点设置命令	67
7.3	断点清除命令	69
7.4	断点屏蔽命令	69
7.5	断点恢复命令	70
7.6	断点列表命令	71
第八章	管理观察语句	72
8.1	引言	72
8.2	设置观察表达式和观察内存语句	72
8.3	设置观察点	75
8.4	设置跟踪点	76
8.5	删除观察语句	79
8.6	列出观察点和跟踪点	80
第九章	执行代码	81
9.1	引言	81
9.2	设置方式命令	81
9.3	反汇编命令	82
9.4	查看命令	84
9.5	当前位置命令	85
9.6	栈跟踪命令	86
第十章	修改代码或数据	88
10.1	引言	88
10.2	汇编命令	88
10.3	输入命令	90
10.3.1	输入命令	92
10.3.2	输入字节命令	92
10.3.3	输入 ASCII 命令	93
10.3.4	输入整数命令	93

10.3.5	输入无符号整数命令	93
10.3.6	输入字命令	94
10.3.7	输入双字命令	94
10.3.8	输入短实数命令	95
10.3.9	输入长实数命令	95
10.3.10	输入十字节实数命令	95
10.4	寄存器命令	96
第十一章	系统控制命令的使用	98
11.1	引言	98
11.2	帮助命令	98
11.3	退出命令	98
11.4	基数命令	99
11.5	重画命令	100
11.6	屏幕切命令	100
11.7	检索命令	101
11.8	外壳转义命令	102
11.9	制表符(Tab)设置命令	104
11.10	重定向命令	104
11.10.1	重定向 CodeView 输入	104
11.10.2	重定向 CodeView 输出	105
11.10.3	重定向 CodeView 输入和输出	106
11.10.4	用于重定向的命令	106
11.10.4.1	注释命令	106
11.10.4.2	延迟命令	107
11.10.4.3	暂停命令	107
附录 A	命令和方式汇总	109
附录 B	正则表达式	115
附录 C	错误信息	118
词汇表		

第一章 Microsoft FORTRAN 5.0 CodeView 简介

1.1 引言

Microsoft CodeView 调试程序可以帮助你测试用 Microsoft FORTRAN 优化编译程序开发的可执行文件。本章介绍 Code View 调试程序,并概述本手册的内容以及符号使用上的约定。

1.2 概况

重要内容

要使用 CodeView 调试程序,并不要求精通汇编语言。它是高级语言程序的强有力的全特征的调试工具。调试程序具备全部汇编特征,但这些特征全部都是可选的。在 FORTRAN 源程序一级运行 CodeView 调试程序时,如果使用 FORTRAN 表达式与 CodeView 通讯,则调试程序以十进制形式显示 FOTRAN 变量的值和表达式的值。

假如只想使用调试程序的源程序级特征,则可参阅 1.6 节“掌握源程序级特征”的一些有用建议。

CodeView 调试程序可显示和执行程序代码、控制程序流程并检查或更改内存单元的值。它的窗口界面方便了设计。可以在一个窗口中浏览源代码,在另一窗口里看命令和响应信息,在第三个窗口里看寄存器和标志位,在第四个窗口里看变量或表达式的值。可以检查全局及局部变量值(自身值或与表达式中其它变量结合的值)。

窗口界面是为 IBM 个人计算机及其兼容机设计的。不过,通过顺序界面,你也可在非 IBM 兼容机上使用 CodeView 调试程序。

利用窗口界面执行的任何调试操作,也可利用顺序界面来做。

CodeView 调试程序通过地址、符号或引用行号可存取程序地址。这样就方便了对特定代码部分的定位和调试。

CodeView 的命令可通过键盘输入,或者(在多数场合下)用 Microsoft 鼠标器输入(只用于窗口界面)。一旦掌握了命令的用法,即可使用鼠标器和键盘高效率地工作。鼠标器不是必不可少的,因为所有命令均可从键盘输入。

注意

CodeView 调试程序是专门为 Microsoft 鼠标器设计的。许多厂家宣称自己的定点设备与 Microsoft 鼠标器兼容。如果它们确是充分兼容的,那么,CodeView 调试程序就可以在这些设备上使用。

CodeView 调试程序易学易用。它的命令符合人们的思维习惯,因此容易理解,尤其对于那些熟悉 Microsoft 符号调试实用程序(SYMDEB)或 DEBUG 实用程序(DOS 提供)的程序员更是这样。CodeView 用户界面和早些时候的同类产品具有共同特征,但又增加了强有力的新特征,如上托菜单、多窗口、鼠标器支持和单键命令。

1.3 关于本手册

本手册解释怎样用 CodeView 调试程序检查程序和寻找程序中的错误。它是《Microsoft FORTRAN 5.0 用户手册》和《Microsoft FORTRAN 5.0 程序员参考手册》的姊妹篇。有关开发 FORTRAN 程序的内容,请参考这些手册。

虽然手册中着重叙述了用源代码调试 FORTRAN 程序,但 CodeView 调试器也可调试汇编语言程序或在汇编语言级调试 FORTRAN 程序。如果不熟悉汇编语言程序设计,可能就不熟悉调试程序的某些特点(特别是汇编方式和寄存器窗口)。

如前所述,CodeView 的一切与汇编语言有关的特征都是可选的。可以忽略汇编方式特征,而在 FORTRAN 源程序方式下调试程序。如果想更详细地了解汇编语言程序设计,则应该考虑购买 Microsoft 宏汇编程序,并阅读它的手册和汇编语言程序设计的教学书籍。

下面列出的内容告诉你怎样查找有关 CodeView 调试程序各方面的信息:

对下列信息	请参见
按照CodeView要求的特殊格式编译和连接FORTRAN程序并用各种命令行选择项调用调试程序	第二章 “准备启动”
使用CodeView显示中的各种元素,包括窗口、上托菜单和鼠标器	第三章 “CodeView显示”
说明会话命令的各种参数,及使用CodeView操作符建立表达式。	第四章 “使用会话命令”
执行整个或部分程序	第五章 “执行代码”
测试表达式的值或检查不同长度的数据	第六章 “检查数据和表达式”
设置、允许、禁止、清除和列出断点	第七章 “管理断点”
建立观察语句并管理观察窗口	第八章 “管理观察语句”
检查代码和跟踪函数、过程或子例程。	第九章 “检查代码”
修改内存数据或代码	第十章 “使用系统控制命令”
控制CodeView调试程序的操作	第十一章 “使用系统控制命令”

除上述信息外,附录中还包括下列信息:

对下列信息	请参见
CodeView方式、命令和菜单的汇总。	附录A “命令和方式汇总”
怎样使用正规式在源文件中查找可变正文字符串	附录B “正规表达式”
CodeView错误信息表	附录C “错误信息”
手册中术语的定义	词汇表

重要内容

在 README.DOC 文件中可能有关于 CodeView 调试程序的附加信息。在本手册印刷后, 如果资料有增添或程序有修改, 该文件中都会有记载。

1.4 记号约定

下列的记号约定适用于整个手册:

约定举例	约定描述
	KEYWORDS和其它概念 黑体大写字母用作文件、目录、寄存器、环境变量、FORTRAN 关键字和FORTRAN内部函数的名字。在DOS级输入的命令也采用大写字母。这些命令既包括程序名, 如FL和LINK, 又包括DOS内部命令, 如SET。在实际输入DOS命令、FORTRAN关键字和内部函数时, 不必用大写字母。
	黑体有时指示正文必须按所示输入。必须按所示输入的正文包括编译程序可选项和FORTRAN运算符。
	+ INT 4 CONTINUE FUNCTION IF /Zi
Program	垂直省略号用在程序举例中, 表示省略程序的一部分。 例如下列三个语句。省略点号表示介于程序行之间有语句, 但没写出来。
	COUNT = 0 . . . PASS=PASS+1 . . . COUNT = 0
Fragment (垂直省略号)	
[选择项]	方括号之间是命令行中的选择项字段和选择项语法。考虑下列命令行语法: R[寄存器][=]值]双方括弧指示你可以输入一个寄存器和一个值。第二组方括弧中的等号(=)表示你可以在值前放一个等号, 但只有给定了值才行。
[选择1 选择2] (垂线)	垂线同方括号配合使用, 表示你可以输入垂线两侧任一项, 如下列语法所示: DB [地址 范围] 垂线表示在字节转贮命令(DB)之后可以说明地址或范围。

因为这两项是在方括弧内，所以使用该命令时可省略参数。
垂线同花括弧配合使用，表示你必须输入垂线两侧中的一
项，如下列语法所示：

W [类型] {地址|范围}

“Quotation marks” (双引

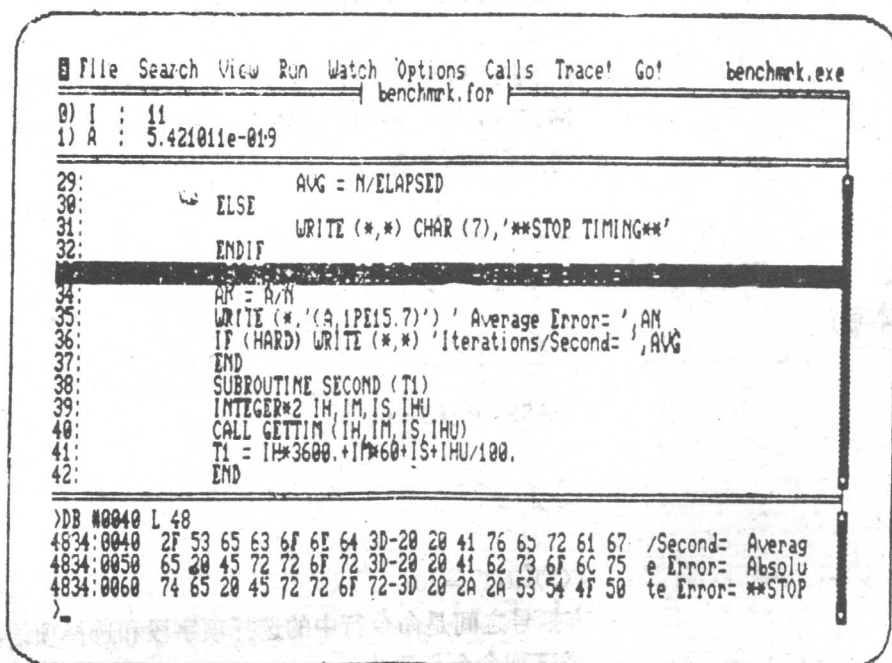
双引号用不标记在文本中定义的术语。譬如，术语“高亮
部号 “ ”) KEY NAMES (细体 分” 第一次定义时
出现在双引号内。

大写字母)

细体大写字母用来表示键名和键序列名，如ENTER、
CONTROL-C、ALT-F。

屏幕示例

屏幕示例用黑白体表示。如果你使用单色显示器或在
CodeView命令行中使用/B选择项(见第二章中2.5.1节“准
备启动”)，那么你的屏幕看上去就和图中一样。图1.1是
一个示例。如果在彩色方式下使用彩显，屏幕就会稍有不同。



The screenshot shows the CodeView window for the executable 'benchmark.exe'. The menu bar includes File, Search, View, Run, Watch, Options, Calls, Trace, and Go!. The main window is divided into two panes. The top pane displays assembly code with line numbers 29 through 42. The code includes instructions like 'AVG = N/ELAPSED', 'WRITE (*,*) CHAR (7), '**STOP TIMING**'', 'ENDIF', 'AV = A/N', 'WRITE (*,*) ((A,1PE15.7))', 'Average Error= ', 'AN', 'IF (HARD) WRITE (*,*) 'Iterations/Second= ', 'AVG', 'END', 'SUBROUTINE SECOND (T1)', 'INTEGER*2 IH, IM, IS, IJU', 'CALL GETTIM (IH, IM, IS, IJU)', 'T1 = IH*3600. + IM*60. + IS + IJU/100.', and 'END'. The bottom pane shows register values for DB #0040 L 48, with addresses 4834:0040, 4834:0050, and 4834:0060. The register values are displayed in a hexadecimal and decimal format, with labels like '/Second= Averag', 'e Error= Absolu', and 'te Error= **STOP'.

```
File Search View Run Watch Options Calls Trace! Go! benchmark.exe
benchmark.for
0) I : 11
1) A : 5.421011e-019

29:          AVG = N/ELAPSED
30:          ELSE
31:          WRITE (*,*) CHAR (7), '**STOP TIMING**'
32:          ENDIF
33:
34:          AV = A/N
35:          WRITE (*,*) ((A,1PE15.7)) ' Average Error= ', AN
36:          IF (HARD) WRITE (*,*) 'Iterations/Second= ', AVG
37:          END
38:          SUBROUTINE SECOND (T1)
39:          INTEGER*2 IH, IM, IS, IJU
40:          CALL GETTIM (IH, IM, IS, IJU)
41:          T1 = IH*3600. + IM*60. + IS + IJU/100.
42:          END

>DB #0040 L 48
4834:0040 2F 53 65 63 6F 6E 64 3D 20 20 41 76 65 72 61 67 /Second= Averag
4834:0050 65 20 45 72 72 6F 72 3D 20 20 41 62 73 6F 6C 75 e Error= Absolu
4834:0060 74 65 20 45 72 72 6F 72 3D 20 20 2A 53 54 4F 50 te Error= **STOP
>
```

图 1.1 CodeView 窗口方式下的屏幕

1.5 定义

CodeView 既和源程序级程序打交道又和汇编语言级程序打交道。手册中有几个术语，根据讨论的是源程序特征还是汇编语言特征，含义稍有不同。手册中尽可能不使用依赖语言的术语，而使用等价的普通术语。

现列出手册中有特殊用途或依赖语言的术语：

术语	含 义
DOS	用在Microsoft手册中的一个术语，用来包括MS-DOS和PC-DOS（当指出其中之一的独特特征时例外）。
符号	是一个名字，象通常所描述的那样，它表示内存中一个地址。在FORTRAN中，变量名和子程序名(子例程或函数)都是符号。
标号	在FORTRAN中，它是一个数，用来指示源代码中的一行(例如，标识DO循环的结尾或GOTO语句的目标)。在汇编语言中，标号是符号，用来指出代码段中一个地址。CodeView调试程序不识别FORTRAN行标号。除非另有说明，否则本手册中的标号一律指汇编语言标号。
例程	在程序中某个地方调用的任何代码段。例程执行完后，程序控制返回到调用语句的下一条语句。FORTRAN中，子程序(子例程或函数)是例程。内部函数也是例程(调用发生在汇编语言级到FORTRAN库例程之间)。汇编语言例程也叫做“过程”。
调用	暂时将程序控制转移到例程的语句，如上所述。
CodeView格式说明符	单字母代码。可能连在表达式中，控制表达式的值显示的格式。这些说明符来自标准C语言库的函数printf。

1.6 掌握源程序级特征

CodeView 调试程序具有众多特点，这些特点在本手册中都有详细叙述。如果你对源程序级调试程序不熟悉，可能就不知道这些特征。如果你只对掌握源程序级 CodeView 调试程序感兴趣，就可以忽略不需要的特征。假设你可以使用 CodeView 窗口方式，推荐下面的使用步骤：

1. 运行在磁盘上提供的源程序。这将使你对 CodeView 调试程序的功能和灵活性有一个总体印象。

2. 当你亲自探索使用 CodeView 调试程序时，只使用菜单命令。它们的用法不言自明，尤其使用鼠标器时更是如此。菜单命令在第三章“CodeView 显示”中叙述。

3. 第一次阅读手册时，不要顾虑会话命令或存储器级和汇编级特征。特别要把注意力集中在描述如何使用带表达式的 CodeView 命令的文本上。在源程序级，你在 CodeView 调试程序上要做的绝大部分工作涉及到 FORTRAN 表达式和变量名。

4. 摸索调试演示程序或你熟翻的另一个程序。掌握怎样设置或删除断点(它们能停止程序的执行)。使用 GO、Program Step (程序步)和 Trace (跟踪) 命令执行程序的一部分。在观察窗口显示一些变量。当执行程序不同部分时，观察变量的变化。在循环内设置断点，并观察每次重复执行时循环内的变量是如何改变的，这样做是特别有益的。这些功能是在本手册的第四、五和七章中解释。

5. 一旦你掌握了断点，就去学习如何使用观察点和跟踪点(只有当表达式变为非零或

改变时，它们才停止程序的执行)。观察语句在第八章中叙述。

6. 当你进一步熟悉了 CodeView 调试程序后，就可摸索使用显示表达式会话命令。该命令用问号(?)在 CodeView 命令行输入。显示表达式命令在 6.2 节叙述。

7. 然后进入到 Dump (转贮) 命令。学习变量怎样存入内存和各种 Dump 命令是怎样和某些变量类型相对应的。摸索使用 CodeView 格式描述符。这些知识即使在源程序级也是有价值的，因为它允许你同时浏览数组中的所有元素。转贮命令和格式描述符在第六章中叙述。

8. 学习与你经常使用的菜单命令相对应的会话命令。它们通常比使用菜单命令功能更强，花的时间更少。会话命令在第四章中叙述。

第二章 准备启动

2.1 引言

准备启动 CodeView 调试程序要求几个简单的步骤。对打算调试的程序，必须准备一个特殊格式的可执行文件。你也许还想选用影响调试程序运行的选择项。

本章叙述一个基于磁盘的演示程序，它向你说明怎样为调试准备文件，并向你介绍 CodeView 调试程序。本章还叙述怎样编译和连接 FORTRAN 程序以产生 CodeView 格式的可执行文件以及怎样将程序装入 CodeView 调试程序。本章还对使用汇编语言程序下的调试程序进行了简单的讨论。

2.2 启动演示程序

装在“掌握 CodeView”磁盘中的演示程序说明了 CodeView 调试程序的许多特征。你在进一步阅读手册之前可能就想试一试演示程序了。

为了启动演示程序，将“掌握 CodeView”磁盘插入软盘驱动器，转到该驱动器，并输入 SAMPLE。

2.3 为 CodeView 调试程序准备 FORTRAN 程序

为了用在 CodeView 调试程序上，可执行文件必须是特殊格式的。除可执行代码之外，特殊格式文件包括行号信息和一个符号表。在编译过程中，必须使用正确的选择项将这些附加信息放入目标文件中，然后在连接过程中放入可执行文件。

假如试图调试未包含这些附加信息的可执行文件，调试程序就不能翻译符号或将代码地址和源程序行号联系起来。虽然仍可在汇编方式下调试程序，但 CodeView 调试程序的许多强有力的特征将不能使用。

对于用 Microsoft FORTRAN 编译程序 4.0 版之前的版本编译过的程序，以上限制也有效。为了使早期的 FORTRAN 程序也能充分利用 CodeView 调试程序，你必须用 Microsoft FORTRAN 优化编译程序(4.0 或更新版本)和 Microsoft 覆盖连接程序(3.5 或更新版本)对它们进行重新编译、连接。可以使用 FL 命令(在《Microsoft FORTRAN 5.0 用户手册》中叙述)编译和连接程序。有关编译和连接的完整命令，请参阅用户指南。对 CodeView 调试程序，编译和连接要求的特殊步骤在本手册的 2.3.1-2.3.3 节。

2.3.1 编写 FORTRAN 源代码

FORTRAN 中任何合法源代码均可编译成可执行文件，并用 CodeView 调试程序进行调试。但某些编程策略使调试更加困难。你对此应该有所了解。

例如，FORTRAN 语言允许你将代码放在分开的嵌入文件中，然后用元命令 \$INCLUDE 将这些文件读入源文件中。不过，你不能用 CodeView 调试程序调试嵌入文件中的源代码。较好的办法就是写且编译分开的库模块，然后将生成的目标文件和你的程序连接起来。CodeView 调试程序是支持这一技术的，并且当你跟踪进入包含在分别编译的文件中的例程

时, 它将自动显示正确的源文件。

2.3.2 为 CodeView 调试程序编译源文件

注意

FL 控制程序将接收以前向斜线(/)或破折号(-)打头的命令行选择项。为简便起见, 手册中描述选择项时只列出斜线, 但你也可使用破折号。虽然文件名可以用大写也可以用小写, 但 FL 命令中选择项的字母大小写是很重要的。你必须严格按照所示字母输入。

当编译要调试的程序时, 必须在 FL 命令中说明 /Zi 选择项。/Zi 选择项指示编译程序将行号和符号信息放到目标文件中。

如果不需要某些模块的完整符号信息, 可用 /Zd 选择项取代 /Zi 编译那些模块。/Za 选择项写到目标文件中的符号信息较少, 所以使用该选择项可节省磁盘空间和内存。譬如, 你在一个包含五个模块的程序上工作, 但只需要调试其中的一个模块, 则可以用 /Zi 命令编译该模块, 而用 /Zd 命令编译其它模块, 然后将它们连接起来生成一个可执行文件。(关于连接分开编译模块的信息, 见《Microsoft FORTRAN 5.0 用户指南》的第四章。)如果用 /Zi 编译了所有模块, 可执行文件将会大得多。可以检查用 /Zd 选择项编译的模块中的全局变量并看其中的源程序行, 但不能检查局部变量。

此外, 你可以说明 /Od 选择项来关闭优化。优化的代码可以重新安排以获得更高效率, 因此, 程序中的指令可能不是和源程序行一一对应。变量, 尤其循环中的变量可暂时赋给寄存器, 或完全不被引用, 所以观察和跟踪命令显示的值可能不合适。调试之后, 可以用自己乐意的优化级编译最终的程序。

对于一个程序, 只有在成功地编译之后, 才能调试它。CodeView 调试程序不会协助你更正语法错误或编译错误。应该参考源程序列表和合适的参考书来更正代码, 直到编译成功, 然后用调试程序定位程序中的逻辑错误。最后, 你可以更正源代码中的这些错误并重新编译。

注意

CodeView 调试程序对于用 Microsoft 库管理程序(LIB)处理过的模块并不适用, 即使它们用 /Zi 选择项编译过也不行。LIB 假设它所处理的模块是已经调试过的, 所以便清除了所有调试信息。

■ 例子

```
FL /Zi /Od EXAMPLE.FOR
```

这个例子编译源文件 EXAMPLE.FOR, 产生一个叫做 EXAMPLE.OBJ 的目标文件。目标文件包含行号信息、符号表和未优化的目标代码。然后 FL 自动调用连接程序, 形成一个可执行文件 COUNT.EXE, 它包含 CodeView 调试程序所需的符号和行号信息。

2.3.3 为 CodeView 调试程序连接目标文件

如果是为了使用 CodeView 调试程序而分开地使用 LINK (而不通过 FL 来调用)去连接一个目标文件或一些文件, 则应该说明 /CODEVIEW 选择项(可缩写为 /CO)。这指示连接程序将符号和源行号地址放入可执行文件中。注意, 当使用 FL 命令时, 不必说明 /CODEVIEW 连接选择项, 因为在说明 /Zi 编译选择项时, 程序自动提供该选择项。

除以上选择项外, 调试程序不再需要其它选择项, 但假如你的程序需要的话, 也可使

用其它选择项(参考《Microsoft FORTRAN 5.0 用户手册》)。譬如, 用户可用/ MAP 或 / PAUSE 选择项。

尽管用/ CODEVIEW 选择项准备的可执行文件可以象其它任何可执行文件一样从 DOS 命令行执行, 但因为其中包含了附加的符号信息, 所以文件比较大。为了最大限度地压缩程序, 在完成对程序的调试后, 则可以不用/ Zi 选择项重新编译、连接, 形成最后版本。

■ 例子

LINK /CO EXAMPLE

这个例子连接目标文件 EXAMPLE.OBJ 形成可执行文件 EXAMPLE.EXE, 它含有 CodeView 调试程序所需要的符号和行号信息。

FL /Zi EXAMPLE

此例用 FL 命令连接目标文件 EXAMPLE.OBJ 形成可执行文件 EXAMPLE.EXE。(注意, 待连接文件扩展名已被省略了, 因此跳过了编译。)

2.4 启动 CodeView 调试程序

文件	位 置
CV.EXE	CodeView程序文件可在当前目录, 或在PATH命令可存取的任何目录上。譬如, 如果使用分布盘中提供的SETUP程序在硬盘上准备FORTRAN编译程序文件, 则可以将CV.EXE放在\BIN目录下。
CV.HLP	如果想要在会话过程中使用联机帮助, 就应该将这个文件放在当前目录或用PATH命令可存取的任何目录里。譬如, 如果用分布盘中提供的SETUP程序在硬盘上准备FORTRAN编译程序文件, 可以把CV.HLP放到\BIN目录下。假如CodeView调试程序找不到帮助文件, 而你又用了一条帮助命令, 这时你仍可使用调试程序, 不过你将得到一条错误信息。
Program.EXE	你要调试的Program这个可执行文件必须放在当前目录下, 或放在启动文件说明时指定的驱动器和目录中。如果找不到可执行文件, CodeView调试程序就会显示错误信息并拒绝启动。
Program.FOR	源文件一般放在当前目录下。如果在编译时, 给出了源文件的所在目录, 则该信息将作为符号信息的一部分存放在可执行文件中。例如, 如果你用命令行 FL -Zi -Od DEMO.FOR 编译时, CodeView将认为源文件放在当前目录下。但是, 如果用命令行 FL -Zi -Od\SOURCE\DEMO.FOR 编译, 则调试程序认为源文件放在\SOURCE目录下。如果在可执行文件说明的目录下找不到源文件(一般是当前目录), 程序会提示你输入一个新目录。你可以输入一个新目录, 也可以按下ENTER键表示该模块用不到源文件。如果没有源文件, 你必须在汇编方式下调试。