

Microsoft CodeView

面向窗口的调试器 使用手册

**Version
4.0**



**Create the fastest FORTRAN programs!
Fully GSA-certified FORTRAN 77.**

黄如福 邵辣子 阎纪明 译

译者的话

《Microsoft Codeview面向窗口的调试器》一书是美国Microsoft公司1987年出版的面向窗口调试器的使用手册。该调试器相对以前各版增加了很多强有力的调试功能；相对其他各种调试程序，它也是目前最好的一种调试器。为此我们组织力量在短期内将它译出，供大家使用。由于时间仓促，翻译欠妥之处，请提出批评意见。

在翻译中，林春哲同志给予了很大的帮助，胡梅同志也参与了工作。

另外，我们还可以优惠向您提供该软件。
联系地址：北京安定门外，中国建筑科学研究院计算中心阎纪明。

1988年5月

面向窗口的调试器

第一章 Microsoft CodeView调试器简介

1.1 引言.....	(1)
1.2 综述.....	(1)
1.3 关于本手册的情况.....	(3)
1.4 符号约定.....	(4)
1.5 定义.....	(8)
1.6 原程序级的功能.....	(9)

第二章 启动调试器

2.1 引言.....	(12)
2.2 启动演示文件.....	(13)
2.3 为C-V调试器准备FORTRAN程序.....	(13)
2.4 启动C-V调试器.....	(17)
2.5 C-V任选项的使用.....	(21)
2.6 与宏汇编一起使用C-V调试器.....	(31)

第三章 C-V的显示

3.1 引言.....	(35)
3.2 使用窗口方式.....	(35)
3.3 使用顺序方式.....	(65)

第四章 使用会话命令

4.1 引言.....	(68)
-------------	------

4.2 键入命令和参数.....	(69)
4.3 C-V命令及参数的格式.....	(71)
4.4 表达式.....	(72)

第五章 执行代码

5.1 引言.....	(84)
5.2 Trace (追踪) 命令	(86)
5.3 Program Step (程序步进) 命令	(88)
5.4 Go (前进) 命令.....	(91)
5.5 Execute (执行) 命令	(94)
5.6 Restart (重新启动) 命令	(95)

第六章 检查数据和表达式

6.1 引言.....	(98)
6.2 显示表达式命令.....	(99)
6.3 检查符号命令.....	(104)
6.4 内存显示命令.....	(107)
6.5 寄存器命令.....	(117)
6.6 8087命令.....	(119)

第七章 断点管理

7.1 引言.....	(122)
7.2 设置断点命令.....	(123)
7.3 清除断点命令.....	(126)
7.4 屏蔽断点命令.....	(127)
7.5 启动断点命令.....	(128)
7.6 断点列表命令.....	(129)

第八章 观察语句的管理

8.1 引言	(131)
8.2 设置观察表达式和观察内存表达式	(133)
8.3 设置观察点	(137)
8.4 设置追踪点	(140)
8.5 删除观察语句	(144)
8.6 列出观察点和追踪点	(145)

第九章 代码检查

9.1 引言	(147)
9.2 设置方式命令	(147)
9.3 反汇编命令	(150)
9.4 显示命令	(153)
9.5 当前地址命令	(156)
9.6 栈追踪命令	(157)

第十章 修改代码或数据

10.1 引言	(160)
10.2 汇编命令	(161)
10.3 键入命令	(164)
10.4 寄存器命令	(174)

第十一章 系统控制命令的使用

11.1 引言	(178)
11.2 help命令	(179)
11.3 退出命令	(180)

11.4 置基数命令	(180)
11.5 重画屏幕命令	(182)
11.6 屏幕转换命令	(183)
11.7 查寻命令	(184)
11.8 Shell调用命令	(187)
11.9 置制表符命令	(189)
11.10 重定向命令	(190)

附 录

A. 命令和方式总结	(197)
B. 规则表达式	(205)
C. 错误信息	(211)
D. 词汇表	(220)

第一章 Microsoft CodeView

调试器简介

1.1 引言

Microsoft公司的调试器是一种帮助你测试用Microsoft公司 FORTRAN 优化编译程序开发的可执行文件的调试程序。这一章将向你介绍调试器、手册摘要和使用上的约定。

1.2 综 述

重 点

使用调试程序,你不必有汇编语言的经验,它为高级程序员提供了一个强有力的全貌的调试工具。调试程序包含了全部汇编语言的特性,但这些是完全可选择的。当在FORTRAN语言源程序级操作调试程序时,你与调试程序的通讯使用的是FORTRAN表达式,调试程序显示的是FORTRAN变量和表达式值的十进制数格式。

如果你仅只希望使用调试程序的源程序级,你在§1.6节中将会得到一些帮助性的建议。

调试程序能够显示和执行程序代码,控制程序的流向,检查或改变内存中的值。调试程序的窗口界面使得调试工作

变得很容易。在一个窗口上你能看到源码，另一个上你能看到命令和返回信息，在第三个窗口上你能看到寄存器和标志位，在第四个窗口上，你能看到变量或表达式的值。你可以通过变量本身或它在表达式中与其它变量结合起来检查全局或局部变量的值。

窗口界面是为IBM-PC及其兼容机设计的。然而，你同样能用顺序界面将调试程序用于非IBM兼容机。用窗口界面能够执行的任何调试操作，其在顺序界面上同样也能执行。

调试程序还能通过地址、符号或行号来存取程序。这使得程序易于定位和调试某一指定的代码段。你能在源程序级调试程序，也能在代码级调试程序。

调试命令可以从键盘键入，通常情况下也可以用Microsoft公司的鼠标来输入。一旦你学会了那些命令，使用鼠标或键盘命令都能最有效地进行工作。鼠标不是必备的，所有的命令都能从键盘输入。

注意：

调试程序是为Microsoft公司的鼠标专门设计的。许多制造商大肆宣传他们的设备与Microsoft公司的鼠标是兼容的。如果他们是严格兼容的，那么调试程序可以在这些设备上工作。

调试程序易学易用。其命令具有逻辑性，易于学懂。尤其对那些熟悉SYMDEB和DEBUG程序的程序员。调试程序的用户接口与其旧版本具有某些共性，但增加了许多强有力的新功能。例如，弹出菜单、多窗口、鼠标和单键命令。

1.3 关于本手册的情况

本手册解释如何使用调试程序进行检查程序和确定程序中的错误。本手册应结合“FORTRAN 编译用户指南”和“FORTRAN编译语言参考”一起使用。

尽管本手册注重于使用源代码调试FORTRAN程序,它同样可以调试汇编语言,或者能够调试汇编语言级的FORTRAN程序。如果你不熟悉汇编语言程序设计,那么你对调试程序的某些内容可能也不熟悉,如汇编方式和寄存器窗口。

象早已说明的那样,调试程序涉及汇编语言的内容,你可以不管它,而只用FORTRAN语言调试你的程序。若你想学习这方面的知识,你可以购买宏汇编和有关的教材。

下面的清单告诉你在调试程序中,如何寻找你需要的信息。

关于这方面的信息	所在章节
(1) 编译和连接FORTRAN程序时, 指出调试程序的格式要求, 用调试器和命令选择起动程序。	第二章
(2) 使用窗口,弹出菜单和鼠标 显示有关内容。	第三章
(3) 给定会话命令中的变量和操作符去建立表达式。	第四章
(4) 执行部分或全部程序。	第五章
(5) 测试表达式的值,或检测不	第六章

同长度的数据。

- | | |
|--------------------------|------|
| (6) 设置、启动、屏蔽、清除断点和列断点清单。 | 第七章 |
| (7) 建立观察语句及管理观察窗口。 | 第八章 |
| (8) 执行代码并追踪函数、过程或子程序调用。 | 第九章 |
| (9) 修改内存的数据或代码 | 第十章 |
| (10) 控制调试器的操作 | 第十一章 |

除上述信息外，附录中还含有如下内容：

- | 关于这方面的信息 | 所在章节 |
|--------------------------------|------|
| (1) 方式、命令、菜单小结 | 附录 A |
| (2) 如何使用规则表达式，在源文件中去寻找可变文稿字符串。 | 附录 B |
| (3) 错误信息清单 | 附录 C |
| (4) 手册中所使用的术语定义。 | 附录 D |

注意

在README.DOC文件中，有许多关于调试程序的增补信息。这些信息是该手册出版以后，追补的一些修改资料。

1.4 符号约定

本手册一律使用如下符号约定：

约	定	约	定	描	述
---	---	---	---	---	---

黑体字、大写

醒目的大写字母用于文件名、目录名、寄存器名、环境变量名, FORTRAN关键字及FORTRAN内部函数。在DOS下键入的命令同样是大写的。这些命令包括建立DOS命令。例如程序名FL和LINK。当你在实际输入DOS命令或FORTRAN关键字和内部函数时,并不要求你使用大写字母。有时文稿中的醒目类型必须按规定的那样键入。如FORTRAN编译时的选择项和FORTRAN的操作符。

```
+                INT 4
CONTINUE        FUNCTION
IF              /Zi
```

斜体字

斜体字安置在命令行和选择说明中。一个斜体字表示一个必须出现在指定位置上的可变项。仔细想想基数命令Nnumber的句法。注意,斜体字number说明,它表示基数(N)命令的一般格式。在实际命令中,使用者为number提供了一个特定的数据。

偶尔,斜体字可能也用于强调文稿中的特殊词句。

示范

示范以一种特定的非比例的正楷字显示出来。如果一个命令会产生输出,那么,你所输入的命令将以醒目的式方显示出来,而通常情况下,调

试程序将要输出的内容以不醒目的方式输出，下面便是一例：

```
>RAX
AX 0041
: 43
>
```

程序

:

片断 (Fragment)

垂直省略点用于程序范例。它表示有部分程序省略了。例如，在下面的摘录中，显示了三个语句。在两个语句之间的省略点，表示了有程序行占用，而未表示出来：

```
COUNT = 0
:
PASS = PASS + 1
:
COUNT = 0
```

【任选项】

在命令行和选择句法中双方括号括起来的选择区域，考虑下面的命令行句法：

```
R [register] [ [= ] value]
```

双括号所括起来的置换表示你可以输入一个寄存器和一个值。第二个方括号中的等号说明你在value前面可以放置一个等号(=)，但它仅只在你说明一个value时才使用。

【选择₁ | 选择₂】

与双括号相关联的垂直线，表示你可以输入垂直线任意一边的内容。

例如下面的句法:

DB [address | range]

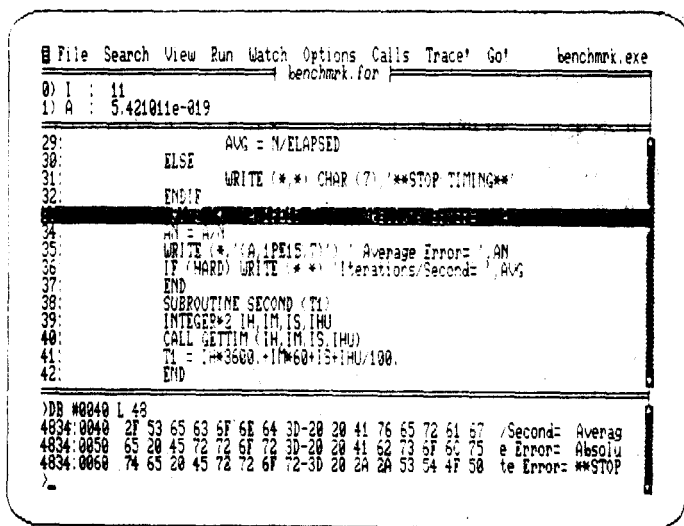
跟随在显示内存字节(DB)命令后的垂直线表示,你可以说明address或range。尽管双括号中有两种选择,你同样也可以不为此命令给出变量。

与花括号相关联的垂直线,表示了你必须选择垂直线任意一边的内容作为输入。例如下面的句法:

W [type] {address | range}

“引号”

引号放在稿文中术语定义的外面。例如,术语“highlight”出现在



The screenshot shows a debugger window titled 'benchmrk.exe'. The menu bar includes File, Search, View, Run, Watch, Options, Calls, Trace, Go. The main window is divided into two panes. The top pane shows assembly code for 'benchmrk.for' with line numbers 29 through 42. The code includes instructions like 'AUG = M/ELAPSED', 'WRITE (*,*) CHAR (7) /**STOP TIMING**', 'END IF', 'SUBROUTINE SECOND (T1)', 'INTEG*2 IM,IM,IS,IMU', 'CALL GETTIM (IM,IM,IS,IMU)', and 'T1 = IM*3600.+IM*60+IS+IMU/100.'. The bottom pane shows a memory dump starting with 'DB #0040 L 48'. It displays hexadecimal addresses (4834:0040, 4834:0050, 4834:0060) and their corresponding byte values. To the right of the memory dump, there is a summary of statistics: '/Second= Averag', 'e Error= Absolu', and 'te Error= **STOP'.

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

29:             AUG = M/ELAPSED
30:             ELSE
31:             WRITE (*,*) CHAR (7) /**STOP TIMING**
32:             END IF
33:             SUBROUTINE SECOND (T1)
34:             IM = 0
35:             WRITE (*,*) (A,1PE15.7V) Average Error= IM
36:             IF (NARD) WRITE (*,*) Iterations/Second= AVG
37:             END
38:             SUBROUTINE SECOND (T1)
39:             INTEG*2 IM,IM,IS,IMU
40:             CALL GETTIM (IM,IM,IS,IMU)
41:             T1 = IM*3600.+IM*60+IS+IMU/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 2A 2A 53 54 4F 50 te Error= **STOP
>
```

图 1.1 C-V调试器在窗口方式下

第一次定义的引号中。

键 名

小的大写字母用于键名和组合键，如，ENTER、CONTROL-C和ALT-F。

屏幕示例

屏幕示例是以黑白方式给出的。如果你有一个单色监视器或者你在调试程序的命令行使用了/B的选择，那么你的屏幕也会是黑白的（见第二章第2.5.1节，），图1.1显示了一个例子。如果你在彩色方式下使用彩色监视器，屏幕将会明显的不同。

1.5 定 义

调试器处理源程序和汇编语言两种程序。用在这本手册中的几个术语，根据所讨论的源程序和汇编语言的不同而具有明显不同的含意。无论哪一种，本手册将使用一个等价的全称术语，而不是与语言相对应的不同术语。

在手册中专用或依赖于语言的术语列表如下：

术	语	含	意
DOS		用于Microsoft公司手册中的术语包括MS-DOS和PC-DOS两种，除非唯一的情况，才是指其中一种。	
符号 (Symbol)		名字，常常描写定位于内存中任意位置的标记。在Fortran中，变量名和子程序 (Subroutines或functions) 名都是符号。	

标号 (Label)

在FORTRAN程序中,一个数据表示了源程序代码所在行的标号例如, DO循环的终端语句, 或GOTO语句的目标)。在汇编语言中, 标号是一个代码所在段的地址标记符。

调试程序并不承认FORTRAN语言的行标号。除非另外说明, 手册中所有的参考标号都是引用汇编语言的标号。

例行程序
(Routine)

例行程序是从程序中别的任何地方调用的任意代码块。当执行完例行程序后, 程序控制返回到调用语句的下一语句。在FORTRAN中, 子程序(Subroutine或 Function)是例行子程序。内部函数同样也是例行子程序(调用FORTRAN程序库里的汇编语言), 汇编语言的例行子程序还做“过程”。

调用 (Call)

临时转移程序的控制到例行子程序的语句。如上所述。

调试程序格式说明符

可将一个字母附加在一种表达式的后面, 由此去控制如何显示表达值。这些说明符来自Printf, C语言标准库的一种功能。

1.6 源程序级的功能

调试程序有许多功能, 所有的功能将在本手册中进行详

细的描述。如果你不熟悉源程序级的调试程序，这些功能对你来说会是新的。如果你仅只对源程序级使用汇编程序感兴趣，你可以忽略那些你不需要的功能。假定你会使用调试程序的窗口方式，下面告诉你如何前进的一些建议。

1. 通过运行指定盘上提供的一个示例，它将给你调试程序的能力和灵活性的总体概念。

2. 当你自己开始具有调试程序的经验时，可仅只使用菜单命令。尤其在你使用鼠标时，你很快会明白它们的使用法。在第三章，“显示”中描述菜单命令。

3. 当你首次阅读这一手册时，不要急于知道会话命令或内存级和汇编语言的功能。要特别注意稿文中对具有表达式的调试程序命令如何描述。你的工作在源程序级使用调试程序，大多应注意FORTRAN表达式和变量名。

4. 具有示例程序或你懂得别的程序关于这方面的经验。你可以学习如何设置和删除断点，停止程序的执行。使用Go，程序步进（Step），和追踪（Trace）命令去执行你的程序。在观察窗口显示一些变量。你执行不同的程序部分，观察变量的变化。尤其可以设置一个断点在循环体内观察循环体内每次迭代时变量如何变动的。这些任务将在4、5和7章中解释。

5. 一旦你精通了断点时，你可以学习如何使用观察点和跟踪点，仅当表达式变成非零和发生变化时，停止程序的执行。观察语句在第8章中描叙。

6. 当你变得比较熟悉调试程序时，试着使用显示表达式的会话命令，它是在调试程序的命令行输入一个问号（?）来实现的。显示表达式命令是在6.2节中描叙。

7. 现在可以上升到显示内存命令了。学习变量如何存

贮在内存和各种显示内存命令如何对应某些变量类型。需要有格式说明符 (format specifier) 的经验。即使是在源程序级, 这种知识也是值得的, 因为它允许你在一个数组中同时寻找所有的元素。显示内存命令和格式说明符在第 6 章中解释。

8. 学习你经常使用的与菜单命令相对应的会话命令。通常它们具有更强的功能。另外, 有时它花的时间比菜单命令少。会话命令在第 4 章中描述。