

洪岷生

吴克西

等 编译

C语言及其开发工具

——调试器及运行库

(下)

WILEY
MICROSOFT

C语言及其开发工具

——调试器及运行库

(下册)

洪岷生 吴克西等 编译

厦门大学出版社

1988·3

责任校对：陈喜乐

C语言及其开发工具
——调试器及运行库
（下册）

洪岷生 吴克西等 编译

*

厦门大学出版社出版发行
三 明 市 印 刷 厂 印 刷

•

开本787×1092 1/16 22.625印张 550千字
1988年5月第1版 1988年5月第1次印刷
印数：0001—1000册

ISBN 7-5615-0057-9

N·1 定价 4.50 元

第四篇 Microsoft C 运行库

第一部分 概述(IV-105)

第一章 引言(IV-105)

第二章 运行库例行程序的使用(IV-107)

2.1 引言(IV-107)

2.2 函数和宏(IV-107)

2.3 包含文件(IV-108)

2.4 函数说明(IV-109)

2.5 参数类型检查(IV-109)

2.6 错误处理(IV-110)

2.7 文件名和路径名(IV-111)

2.8 二进制方式和文本方式(IV-113)

2.9 MS-DOS 的版本影响(IV-113)

2.10 浮点支持(IV-114)

2.11 巨型数组(IV-115)

第三章 全程变量和标准类型(IV-116)

3.1 引言(IV-116)

3.2 _ambiksiz(IV-116)

3.3 daylight, timezone, tzname(IV-116)

3.4 _doserrno, errno, sys_errlist, ssynerr(IV-117)

3.5 _fmode(IV-118)

3.6 _osmajor, _osminor(IV-118)

3.7 _environ, _psp(IV-118)

3.8 标准类型(IV-119)

第四章 运行库例行程序的分类(IV-120)

4.1 引言(IV-120)

4.2 缓冲操作(IV-120)

4.3 字符的分类与转换(IV-120)

4.4 数据转换(IV-121)

4.5 目录控制(IV-122)

4.6 文件处理(IV-122)

4.7 输入和输出(IV-123)

4.7.1 流式 I/O 例行程序(IV-124)

4.7.2 低级 I/O 例行程序(IV-128)

4.7.3 控制台和端口 I/O 例行程序(IV-130)

4.8 数学函数(IV-130)

4.9 内存分配(IV-132)

4.10 MS-DOS 接口(IV-133)

4.11 进程控制(IV-134)

4.12 查找与排序(IV-136)

4.13 字符串操作(IV-136)

4.14 时间函数(IV-137)

4.15 可变长的参数表(IV-138)

4.16 其他例行程序(IV-138)

第五章 包含文件(IV-140)

5.1 引言(IV-140)

5.2 assert.h(IV-140)

5.3 conio.h(IV-141)

5.4 ctype.h(IV-141)

5.5 direct.h(IV-141)

5.6 dos.h(IV-141)

5.7 errno.h(IV-142)

5.8 fcntl.h(IV-142)

5.9 float.h(IV-142)

5.10 io.h(IV-142)

5.11 limits.h(IV-143)

5.12 malloc.h(IV-143)

5.13 math.h(IV-143)

5.14 memory.h(IV-144)

5.15 process.h(IV-144)

5.16 search.h(IV-144)

5.17 setjmp.h(IV-144)

5.18 share.h(IV-144)

5.19 signal.h(IV-145)

5.20 stdarg.h(IV-145)

5.21 stddef.h(IV-145)

5.22 stdio.h(IV-145)

5.23 stdlib.h(IV-146)

5.24 string.h(IV-147)

5.25 sys\locking.h(IV-147)

5.26 sys\stat.h(IV-147)

5.27 sys\timeb.h(IV-147)

5.28 sys\types.h(IV-147)

5.29 sys\utime.h(IV-148)

5.30 time.h(IV-148)

5.31 varargs.h(IV-148)

5.32 v2tov3.h(V-148)

第二部分 参考(V-149)

附录 A 错误信息

A.1 引言.....(V-341)

A.2 errno 值.....(V-341)

A.3 数学错误(V-342)

附录 B MS-DOS 和 XENIX 的公用库

B.1 引言(V-344)

B.2 公用的库例行程序(V-344)

B.3 公用的全程变量(V-346)

B.4 公用的包含文件(V-346)

B.5 公用例行程序间的差别(V-347)

附录 C 运行库例行程序名索引

目 录

第三篇 窗口符号调试器 CodeView

第一章 概述.....(I - 1)

第二章 启动.....(II - 2)

2.1 引言.....(II - 2)

2.2 启动示范程序.....(II - 2)

2.3 准备调试用的 C 程序.....(II - 2)

2.3.1 编写 C 源程序.....(II - 2)

2.3.2 编译源文件.....(II - 3)

2.3.3 连接目标文件.....(II - 4)

2.4 启动调试器.....(II - 4)

2.5 调试器选项.....(II - 5)

2.5.1 设置黑白显示方式.....(II - 6)

2.5.2 规定初启命令.....(II - 6)

2.5.3 设置屏幕切换方式.....(II - 7)

2.5.4 设置窗口方式或顺序方式.....(II - 8)

2.5.5 关闭鼠标.....(II - 9)

2.5.6 设置 43 行显示方式.....(II - 9)

2.5.7 使用两个显示器.....(II - 9)

2.5.8 取消某些 IBM 专用功能.....(II - 9)

2.5.9 设置某些 IBM 专用功能.....(II - 10)

2.6 调试汇编语言程序.....(II - 10)

第三章 CodeView 显示.....(III - 11)

3.1 引言.....(III - 11)

3.2 窗口方式.....(III - 11)

3.2.1 键盘操作.....(III - 12)

3.2.1.1 移动光标.....(III - 13)

3.2.1.2 改变屏幕.....(III - 13)

3.2.1.3 控制程序执行.....(III - 13)

3.2.1.4 选择菜单.....(III - 14)

3.2.2 鼠标操作.....(III - 15)

3.2.2.1 改变屏幕.....(III - 15)

3.2.2.2 控制程序执行.....(III - 15)

3.2.2.3 选择菜单.....(III - 16)

3.2.3 菜单.....(III - 17)

3.2.3.1 File 菜单.....(III - 17)

3.2.3.2 Search 菜单.....(III - 17)

3.2.3.3 View 菜单.....(III - 18)

3.2.3.4 Run 菜单.....(III - 18)

3.2.3.5 Watch 菜单.....(III - 18)

3.2.3.6 Options 菜单.....(III - 19)

3.2.3.7 Calls 菜单.....(III - 20)

3.2.4 求助系统.....(III - 21)

3.3 顺序方式.....(III - 21)

第四章 对话命令.....(IV - 23)

4.1 引言.....(IV - 23)

4.2 命令和参数的输入.....(IV - 23)

4.2.1 专用键.....(IV - 23)

4.2.2 命令缓存.....(IV - 23)

4.3 命令和参数的格式.....(IV - 24)

4.4 C 表达式.....(IV - 24)

4.4.1 标识符.....(IV - 25)

4.4.2 常数.....(IV - 26)

4.4.3 寄存器.....(IV - 26)

4.4.4 地址.....(IV - 27)

4.4.5 地址范围.....(IV - 28)

4.4.6 行号.....(IV - 28)

4.4.7 字符串.....(IV - 29)

第五章 执行代码.....(V - 30)

5.1 引言.....(V - 30)

5.2 跟踪命令.....(V - 30)

5.3 程序步命令.....(V - 32)

5.4 执行命令.....(V - 33)

5.5 慢执行命令.....(V - 35)

5.6 重启动命令.....(V - 36)

第六章 检查数据和表达式.....(VI - 37)

6.1 引言.....(VI - 37)

6.2 表达式显示命令.....(VI - 37)

6.3	符号检查命令	(Ⅱ-41)
6.4	内存显示命令	(Ⅱ-44)
6.4.1	内存显示命令	(Ⅱ-44)
6.4.2	内存字节显示命令	(Ⅱ-45)
6.4.3	内存 ASCII 显示命令	(Ⅱ-45)
6.4.4	内存整数显示命令	(Ⅱ-46)
6.4.5	内存无符号整数显示命令	(Ⅱ-46)
6.4.6	内存字显示命令	(Ⅱ-46)
6.4.7	内存双字显示命令	(Ⅱ-47)
6.4.8	内存短实数显示命令	(Ⅱ-47)
6.4.9	内存长实数显示命令	(Ⅱ-48)
6.4.10	内存10字节实数显示命令	(Ⅱ-48)
6.5	寄存器命令	(Ⅱ-48)
6.6	8087命令	(Ⅱ-50)
第七章	断点	(Ⅱ-52)
7.1	引言	(Ⅱ-52)
7.2	断点设置命令	(Ⅱ-52)
7.3	断点清除命令	(Ⅱ-54)
7.4	断点禁止命令	(Ⅱ-54)
7.5	断点恢复命令	(Ⅱ-55)
7.6	断点列表命令	(Ⅱ-56)
第八章	监视语句	(Ⅱ-57)
8.1	引言	(Ⅱ-57)
8.2	设置表达式监视和内存监视语句	(Ⅱ-57)
8.3	设置监视点	(Ⅱ-59)
8.4	设置跟踪点	(Ⅱ-60)
8.5	删除监视语句	(Ⅱ-62)
8.6	列出监视点和跟踪点	(Ⅱ-63)
第九章	检查代码	(Ⅱ-65)
9.1	引言	(Ⅱ-65)
9.2	方式设置命令	(Ⅱ-65)
9.3	反汇编命令	(Ⅱ-66)
9.4	查看命令	(Ⅱ-67)
9.5	当前位置命令	(Ⅱ-69)
9.6	堆栈跟踪命令	(Ⅱ-70)
第十章	修改代码和数据	(Ⅱ-72)
10.1	引言	(Ⅱ-72)
10.2	汇编命令	(Ⅱ-72)
10.3	输入命令	(Ⅱ-73)
10.3.1	输入命令	(Ⅱ-75)
10.3.2	字节输入命令	(Ⅱ-76)
10.3.3	字输入命令	(Ⅱ-76)

10.3.4	整数输入命令	(Ⅱ-76)
10.3.5	无符号整数输入命令	(Ⅱ-77)
10.3.6	字输入命令	(Ⅱ-77)
10.3.7	双字输入命令	(Ⅱ-78)
10.3.8	短实数输入命令	(Ⅱ-78)
10.3.9	长实数输入命令	(Ⅱ-79)
10.3.10	10字节实数输入命令	(Ⅱ-79)
10.4	寄存器命令	(Ⅱ-79)
第十一章	系统控制命令	(Ⅱ-82)
11.1	引言	(Ⅱ-82)
11.2	求助命令	(Ⅱ-82)
11.3	退出命令	(Ⅱ-82)
11.4	基数设置命令	(Ⅱ-83)
11.5	屏幕重画命令	(Ⅱ-84)
11.6	屏幕切换命令	(Ⅱ-84)
11.7	搜索命令	(Ⅱ-85)
11.8	外壳调用命令	(Ⅱ-86)
11.9	制表符设置命令	(Ⅱ-88)
11.10	重定向命令	(Ⅱ-88)
11.10.1	CodeView输入重定向	(Ⅱ-88)
11.10.2	CodeView输出重定向	(Ⅱ-89)
11.10.3	CodeView输入输出重定向	(Ⅱ-90)
11.10.4	与重定向有关的命令	(Ⅱ-90)
11.10.4.1	注释命令	(Ⅱ-90)
11.10.4.2	延迟命令	(Ⅱ-91)
11.10.4.3	暂停命令	(Ⅱ-91)
附录A	命令和方式小结	
A.1	引言	(Ⅱ-93)
A.2	方式	(Ⅱ-93)
A.3	选项	(Ⅱ-93)
A.4	窗口命令	(Ⅱ-94)
A.5	对话命令	(Ⅱ-95)
A.6	格式化输出	(Ⅱ-97)
附录B	正规表达式	
B.1	引言	(Ⅱ-98)
B.2	正规表达式中的特殊字符	(Ⅱ-98)
B.3	特殊字符的查找	(Ⅱ-98)
B.4	句点的使用	(Ⅱ-98)
B.5	方括号的使用	(Ⅱ-99)
B.6	星号的使用	(Ⅱ-100)
B.7	行首和行尾的匹配	(Ⅱ-100)
附录C	错误信息	

第三篇 窗口符号调试器 CodeView

第一章 概 述

Microsoft CodeView 调试器是用来对在 Microsoft C 编译器上开发的可执行文件进行调试的程序。

CodeView 调试器可以显示和执行程序代码，控制程序流程，检查和修改内存的值。它的窗口界面使调试更容易进行。用户可以在第一个窗口观察源程序代码，在第二个窗口观察命令和响应，在第三个窗口观察寄存器和标志位，在第四个窗口观察变量或表达式的值。而且，对于全程变量或局部变量的值，可以个别地检查或者结合表达式中的其他变量一起检查。

窗口界面是专为 IBM PC 机和 IBM 兼容机设计的。但是，非 IBM 兼容机通过顺序界面也可以使用 CodeView 调试器。任何在窗口界面执行的调试操作也可以在顺序界面执行。

CodeView 调试器可以通过地址、符号或行号来访问程序的内存位置，因此便于找到某一特定代码的具体位置然后加以调试。用户既可在调试器的汇编语言方式的源程序级上调试程序，也可在调试器的汇编语言方式的机器代码级上检查代码。

CodeView 调试器的命令可以从键盘上输入或者通过鼠标器输入（后者仅适用于窗口界面）。用户一旦掌握了调试器的各种命令，就可以同时使用鼠标器和键盘输入来提高工作效率。

CodeView 调试器易学易用，它的命令很有逻辑性，容易被人理解，特别是对那些熟悉 Microsoft 符号调试器 SYMDEB 和 MS-DOS 实用程序 DEBUG 的程序员来讲更是如此。CodeView 用户界面继承其前辈的某些特性，但也扩充了一些强有力的新特性，如弹出菜单、多窗口显示、鼠标支持及单键命令。

本书采用如下所示的命令行格式约定：

DB【地址 | 范围】

其中，DB 是内存字节显示命令，用双方括号（[]）括起来的项表示任选项（即可有可无项），例如可只给出 DB 命令而不给出任何地址或范围参数。垂直线（|）表示在其左右的两项中只能规定一项，例如只规定地址或者只规定范围。

第二章 启动

2.1 引言

CodeView 调试器的启动只需简单几步。首先为需要调试的程序准备一个特别格式的可执行文件，然后调用调试器。此外还可规定一些选项，以影响调试器的操作。

本章叙述一个存放在表演盘上的示范程序，它是一个关于 CodeView 调试器的入门介绍。本章也叙述如何编译和连接一个程序，以便生成一个 CodeView 格式的可执行文件，以及如何把一个程序装入调试器。同时还简要介绍如何使用调试器来调试汇编语言程序。

2.2 启动示范程序

在表演盘上提供的示范程序可以显示 CodeView 调试器的许多功能。读者在进一步阅读本篇之前最好试一下这个示范程序。

只要把表演盘插入软盘驱动器，然后键入 SAMPLE，就可以启动示范程序。这条命令执行批文件 SAMPLE.BAT，它首先显示一些有关示范程序的目的说明，然后自动把一个程序装入调试器。在整个示范过程中，注释将告诉你发生了什么以及为什么发生。同时不断地要求你按任意一键，以使示范过程继续下去，这样你就可以控制示范过程的进展速度。

注意，如果你想在示范程序结束之前退出，只要按一下 CONTROL-C 或 CONTROL-BREAK 组合键就行。如果仍在批文件中，那么将出现一条提示信息问你是否要终止执行，这时键入 Y 就终止执行。如果调试器已经开始运行，在 CodeView 提示符 (>) 后面将出现 break 这个字，这时键入 Q (退出命令) 就可终止程序并返回到 MS-DOS 操作系统。

2.3 准备调试用的 C 程序

可执行文件必须具有特别格式以便调试器使用。这个特别格式的文件除了包含可执行代码以外，还包含行号信息和一个符号表。在编译时必须使用正确的选项，才能把这些附加的信息放在目标文件中，然后在连接时再把它们放入可执行文件中。

当调试一个不包含这些附加信息的可执行文件时，调试器将无法对符号加以解释，也无法使代码地址和源程序行号发生联系。这时虽然仍可在汇编方式下调试程序，但是调试器的许多最有力的功能将无法使用。

程序必须用 Microsoft C 编译器提供的两个编译控制程序 (MSC 和 CL) 之一进行编译，而且必须用 Microsoft 复盖连接程序 (LINK) 加以连接 (见第二篇“Microsoft C 编译器”) 编译和连接 CodeView 调试器所需程序的特殊步骤在下面几节中叙述。

2.3.1 编写 C 源程序

任何合法的 C 源程序都可编译成可执行文件并在 CodeView 调试器上进行调试。但是, 某些编程的具体作法可能增加调试困难, 因此必须知道这些具体作法是如何影响调试的。

如果源程序的每一行只含有一个语句, 那么程序就容易调试。例如, 下列是一行合法的 C 程序:

code = buffer[count]; if (code-- '\n') ++lines; 这行程序实际上是由三个分开的 C 语句组成。当把它们放在同一行上, 在调试时就无法个别存取这些语句。因此, 既不能在 ++lines 语句上设置断点, 也不能只执行到 if (code-- '/n') 语句。同样的程序如按下列形式编写就容易调试:

```
code = buffer[ count ];  
if ( code-- '\n' )  
    ++lines;
```

这样做也使程序易读, 通常被认为是好的编程作法。

对调试器来说, 宏是不好调试的。对大多数简单的宏, 调试一般没有什么问题, 但是对带有副作用的复杂的宏, 也许首先需要把它们展开, 写成通常的源程序语句, 经调试后再把它们改写成宏。

C 语言允许把源程序分开放在一些包含文件中, 然后在源文件中用 #include 伪指令将它们读进来。但是, 调试器不能调试包含文件中的源程序。因此, 最好使用只含有宏的包含文件。在 C 程序中使用标准代码的最好办法是分开编写和编译各种库模块, 然后将生成的目标模块跟用户程序连接。CodeView 调试器支持这种作法。

2.3.2 编译源文件

当编译一个需要调试的程序的源文件时, 在回答 MSC 提示或者在 MSC 或 CL 命令行上, 必须规定 /Zi 选项。/Zi 选项指示编译器在目标文件中包含行号和符号信息。

如果有些模块不需要完整的符号信息, 那么可用 /Zd 选项代替 /Zi 选项来编译这些模块。/Zd 选项在目标文件中只放很少的符号信息, 因此它可以节省磁盘和内存的空间。例如, 有个程序由五个模块组成, 其中只有一个模块需要调试, 那么可以用 /Zi 选项编译这个模块, 而用 /Zd 选项编译其他四个模块。这样得到的可执行文件将大大小于全部采用 /Zi 选项编译得到的文件。对于用 /Zd 选项编译的模块, 在调试时仍可以检查它的全程变量和观察它的源程序行, 但不能检查它的局部变量。

此外, 可能还需要规定 /Od 选项来取消优化。优化的代码可能进行重排以获得最高效率, 结果也许造成程序指令不能跟源程序行很好地对应。但经调试后, 便可采用任意优化级来编译一个程序, 从而获得其最终优化版本。

注意, 程序只有编译成功才能进行调试。调试器并不能帮助改正语法错误或编译错误。用户可能需要参考源程序清单和有关的参考书来改正源程序, 直到编译成功为止。接着才用调试器确认程序是否存在逻辑错误。如果发现错误, 则须在源程序上改正这些错误并重新编译源程序。

例子

```
MSC COUNT /Zi /Od,
```

这个例子编译源文件 COUNT.C 并生成目标文件 COUNT.OBJ。此目标文件包含行

号信息、一个符号表以及未优化的目标代码。

2.3.3 连接目标文件

当连接一个给调试器准备的目标文件时，必须规定/**CODEVIEW** 选项（可缩写成/**CO**）。这个选项指示连接器把源程序的符号信息放进可执行文件中。

调试器本身不再需要其他任何选项，但是如果程序本身需要的话，也可以使用其他的选项，如/**MAP** 或 /**PAUSE**选项。但是，注意不要使用/**EXEPACK** 选项，因为它会取消在执行文件上生成符号信息。**EXEPACK** 实用程序也会取消符号信息。调试器在启动时如果检测到一个压缩文件，它将产生一个警告信息。这时虽然仍可用汇编方式调试程序，但是没有任何符号信息可供使用。

采用/**CODEVIEW** 选项准备的可执行文件虽然象其他的可执行文件一样可在 **MS-DOS** 命令级上执行，但是由于它带有附加的符号信息，所以它的程序较长。因此，程序经调试后，在生成该程序的最终版本时，不要再用/**CODEVIEW** 连接选项，以便获得最短的程序。

例子

```
LINK /CO COUNT;  
CL /Zi /Od COUNT.C
```

第一个例子连接目标文件 **COUNT.OBJ**，生成一个包含有 **CodeView** 调试器所需的符号和行号信息的可执行文件 **COUNT.EXE**。

第二个例子用 **CL** 命令在一次连续操作中对程序进行编译和连接并生成一个可执行文件。用 **CL** 命令时不需要规定 /**CODEVIEW** 选项，因为规定 /**Zi** 编译选项就等于自动规定 /**CO** 连接选项。

2.4 启动调试器

在启动调试器之前，必须确认它所需的文件都已存放在合适的位置上。对于调试 **C** 源程序，推荐采用如下文件组织：

文件	位置
CV.EXE	CodeView 程序文件可以放在当前目录或者放在 PATH 命令所规定的任何一个目录（如 \ BIN 目录）中。
CV.HLP	如果用户在调试过程中需要求助信息，就应把这个文件放在当前目录或者放在 PATH 命令所规定的任何一个目录（如 \ BIN 目录）中。如果用户提出求助而调试器不能找到这个求助文件，它将显示一个错误信息，但这不影响程序调试。
可执行文件	需要调试的可执行文件必须放在当前目录或者放在命令行上规定的驱动器和目录中。如果调试器找不到这个可执行文件，它将显示一个错误信息并终止执行。
源文件	源文件通常应放在当前目录中。但是如果在编译时给出源文件的目录说明，那么该说明将作为符号信息的一部分存放在可执行文件中。例如，编译时如果使用命令行 MSC DEMO ，那么调试器将认为源文件在当前目录中。但是，编译时如果使用命令行 MSC/C/DEMO ，那么调试器将认为源文件在 /C 目录中。如果调试器不能在可执行文件规定的目录中找到源文件，那么它将提示用户输入一个新的目录名。这时可输入一个新的目录名，或者只按回车键表示本模块不需要使用源文件。如果没有规定源文件，那么只能用汇编方式进

行调试。

如果有关的文件都已存放在正确的目录中，就可以在 MS-DOS 命令提示符后面输入 CodeView 命令行。此命令行的形式如下：

CV【选项】可执行文件名【参数】

此命令行可以带2·5节所述的一个或多个选项，也可不带任何选项。调试器装入的可执行文件必须具有 .EXE 或 .COM 的扩展名。如果试图装入一个非可执行的文件，那么将出现下列信息：

Not an executable file

包含 CodeView 符号信息的 C 程序和汇编语言程序总是带 .EXE 扩展名。带 .COM 扩展名的文件可以在汇编方式下调试，但它们不包含符号信息。采用复盖的程序不能用 CodeView 调试器调试。

在命令行上传给可执行文件的参数是任选的。如果被调试程序并不接受命令行参数，那么就不需要规定任何参数。

如果给出一个不带扩展名的文件名，调试器就把给出的名字作为基本名再加上 .EXE 扩展名来查找一个可执行文件。如果规定的文件不是 CodeView 格式的文件，那么调试器在启动后将直接进入汇编方式，同时显示下列信息：

No symbolic information

在启动调试器时必须规定一个可执行文件，如果忽略了该文件，调试器将显示一条命令格式的提示信息。

如果给出的命令行是有效的，调试器将装入可执行程序 and 源文件，处理地址数据，并显示 CodeView 屏幕信息。最初的显示或是窗口方式或是顺序方式，这取决于用户规定的选项和所用的计算机型号。例如，若想调试 SIEVE.EXE 程序，可用如下命令行启动调试器：

CV SIEVE

如果这个命令行是在 IBM PC 上给出的，则调试器将自动选择窗口方式，否则调试器一般会选择顺序方式。但是，用户可用2·5节所述的 CodeView 选项来取代这个缺省初启方式。

对于 C 程序，调试器在启动后就会停在程序之前的 C 初启代码的入口位置。在源方式下，可以输入一条执行代码命令（如跟踪命令或程序步命令）来自动执行 C 的初启代码，然后停在程序的开头。这时就可开始按第三章到第十一章所述的那样来调试程序。

2·5 调试器选项

通过在命令行上规定一些选项，可以改变调试器的初始状态。

选项是由一个正斜杠 (/) 或一个破折号 (—) 打头的一串字符序列。如果文件名含有破折号，那么在用调试器调试该文件之前必须给它重新命名，以免调试器将该文件名中的破折号解释为选项指示符。命令行可带多个选项，各选项必须带有选项指示符，且项与项之间必须用空格隔开。

下面是关于采用选项的一些建议。如果有一个以上条件成立，那么就可以采用一个以上

的选项（按任意次序）。如果不满足任何条件，那么就不要再采用任何选项。

条件	选项
• IBM 兼容机，且想用窗口方式。	/W
• 配双色显示器和彩色图形适配器（CGA），且为IBMPC机或IBM兼容机。	/B
• 调试一个图形程序且想看到输出屏幕。	/S
• 调试一个使用多视频页显示的程序且想看到输出屏幕。	/S
• 非IBM兼容机，且想看到输出屏幕。	/S
• IBM兼容机，调试一个非图形或非多视频页显示的程序，且想看到输出屏幕。	/F
• IBMPC机，但想按顺序方式调试。	/T
• 系统装有鼠标器，但调试中不想用它。	/M
• 带增强图形适配器（EGA）的IBMPC机或IBM兼容机，且想要43行显示。	/43
• 想要调试器在启动时自动执行一系列命令。	/C命令序列

假设有一台 IBM 兼容机，它带有一个彩色图形适配器和一个双色显示器。需调试的程序叫 GRAPHIX.EXE，它在图形方式下作图。在调试过程中希望能看到输出屏幕。最后希望启动调试器若干次而无须记住所有的选项，且希望调试器每次都能自动执行C的初启代码。这时可以建立一个叫GRAFBUG.BAT的批文件，它包括下列一行命令：

```
CV /W /B /S /C Gmain GRAPHIX
```

2.5.1 到2.5.6节将更详细叙述Code View的选项。

2.5.1 设置黑白显示方式

选项

/B | -B

调试器在启动时检查计算机所带的显示适配器的类型，如果它检测到单色适配器(MA)，就按双色显示，如果它检测到彩色图形适配器(CGA)，就按彩色显示。而/B选项强制调试器都按双色显示（即使有CGA也一样）。

带CGA的双色显示器（通常为绿色和黑色，或者琥珀色和黑色）通常试图用不同的交错图案或不同的灰度来显示各种颜色。如果用/B选项强制显示器按黑白显示，屏幕显示就更易读。这时，多数双色显示器仍有四种颜色区分：背景（黑色）、正常、高亮度和反白。

例子

```
CV /B COUNT COUNT.TXT
```

这个例子使调试器按黑白显示方式启动。这是单色显示器唯一可用的方式，对于一个带有CGA的双色显示器，采用这个方式会使屏幕显示更易读。

2.5.2 规定初启命令

选项

/C命令序列 | -C命令序列

/C选项允许规定一个或多个调试器在初启时将自动执行的命令，各命令项之间用分号隔开。如果这些命令项带有要用空格分隔的参数，那么整个选项就必须用双引号括起来。否则，调试器将把各个参数解释成调试器命令行的参数，而不是解释成调试器命令的参数。注意，任何使用小于符号(<)和大于符号(>)的初启命令选项，即使它本身不需要空格也

必须用双引号括起来。因为这样才能保证这些重定向符将由调试器自己解释而不由 MS-DOS 解释。

例子

```
CV /CGmain COUNT.EXE COUNT.TXT
```

```
CV "/CS-, N16, G countwords, D buffer L100" COUNT.EXE COUNT.TXT
```

```
CV "/C<INPUT.FIL" COUNT.EXE COUNT.TXT
```

第一个例子使调试器装入 COUNT.EXE 作为可执行文件，装入 COUNT.TXT 作为参数，且在初启时用命令 Gmain 执行 C 的初启代码。由于该命令 (G) 和其参数 (main) 之间不需要空格，所以该选项不用双引号括起来。

第二个例子装入同一个文件且带同样的参数，但命令项却多得多。调试器从汇编方式 (S-) 开始，采用 16 进制 (N16)，执行到过程 countwords (G countwords)，然后从变量 buffer 开始显示 100 个字节的内存单元的内容 (D buffer L100)。由于若干个命令使用了空格，所以整个选项用双引号括起来。

第三个例子装入同样的文件和参数，但初启命令指示调试器不从键盘而从文件 INPUT.FIL 中接收输入。虽然该选项不包含任何空格，但它必须用双引号括起来，使得小于符号 (<) 由调试器自己解释而与 MS-DOS 无关。

2.5.3 设置屏幕切换方式

```
/F | -F
```

```
/S | -S
```

调试器允许在作为程序输出的输出屏幕和作为调试显示的调试屏幕之间迅速切换。调试器可以用两种方式来处理这种屏幕切换：屏幕换页或屏幕交换。在命令行上可以用 /F 选项（屏幕换页）或 /S 选项（屏幕交换）来选择这两种方式之一。

对于非 IBM PC 机，其缺省方式是无屏幕切换。因此如果不规定任何方式，屏幕切换命令将不起作用。对于带图形适配器的 IBM PC 机，其缺省方式是屏幕换页，而对于带单色适配器的 IBM PC 机，其缺省方式是屏幕交换。

屏幕换页用图形适配器 (CGA 或 EGA) 上的若干个视频显示页来存放各个屏幕正文，视频显示页是一些专用的内存缓冲区，它们专门用作多屏幕的显示输出。这种方式比屏幕交换方式速度更快且使用较少内存。但是，屏幕换页不能用于单色适配器 (MA)，或者用来调试产生图形输出或使用多个视频显示页的程序。此外，调试器的屏幕换页方式只能在 IBM PC 机和 IBM 兼容机上使用。

屏幕交换虽然没有屏幕换页的那些限制，但是它的速度明显下降且它需要更多的内存。在屏幕交换方式下，调试器在内存中设立一个缓冲区，用它存放当前暂时不用的屏幕正文。当用户要看另一个屏幕时，调试器就将显示缓冲区的屏幕正文跟内存缓冲区的屏幕正文进行交换。

当使用屏幕交换时，内存缓冲区的尺寸是 4K（对于 MA）或者是 16K（对于 CGA 或 EGA）。因此，调试器所用的内存空间便增加了该缓冲区的尺寸。

表 2.1 表示在各种配置下缺省设定的屏幕切换方式（交换或换页）和显示方式（顺序或窗口）。显示方式将在 2.5.4 节中讨论。

表2.1 缺省设定的屏幕切换方式和显示方式

计 算 机	显示适配器	缺省方式	其 他 方 式
IBM	CGA或EGA	/F/W	/S(使用多个显示页或图形的程序的屏幕交换); /T(顺序方式)
IBM兼容机	CGA或EGA	/T	/W(窗口方式); /F(文本程序的屏幕换页)或/S(使用多个显示页或图形的程序屏幕交换)
IBM	MA	/S/W	/T(顺序方式)
IBM兼容机	MA	/T	/W(窗口方式); /S(屏幕交换)
非兼容机	任意	/T	/S(屏幕交换)

如果用户不能肯定自己的计算机是否跟IBM完全兼容, 可以进行试验。如果计算机的基本输入输出系统(BIOS)不够兼容, 那么调试器可能不能用/F选项。

如果对单色适配器规定 /F 选项, 调试器将忽略这个选项而采用屏幕交换方式。如果试图用屏幕换页方式来调试一个产生图形输出或使用多个视频显示页的程序, 其结果将难于意料; 因此必须采用/S选项重新启动调试器。

例子

```
CV /F COUNT COUNT.TXT
CV /S GRAFIX
```

第一个例子按屏幕换页方式启动调试器。对于一台IBM兼容机, 如果希望取代缺省设定的无屏幕切换方式, 且选择内存使用较少和切换速度较快的屏幕换页方式, 就可以用这条命令。对于一台IBM PC机, 不需要规定/F选项, 因为屏幕换页就是它的缺省方式。

第二个例子按屏幕交换方式启动调试器。如果程序使用了图形态, 就得用这条命令。

2.5.4 选择窗口方式或顺序方式

选项

```
/T | -T
/W | -W
```

CodeView 调试器可以在窗口方式或顺序方式下操作。窗口方式可以显示多达四个窗口, 使用户可同时看到调试过程的不同侧面。窗口方式要求是IBM或IBM兼容的计算机。顺序方式适用任何计算机, 且适用重定向命令, 这时调试信息在屏幕上顺序显示。

第三章“CodeView 显示”将详细讨论每种方式的特点。如果用户不能肯定自己的计算机是否跟IBM完全兼容, 可以进行试验。如果计算机的BIOS不够兼容, 那么可能不能使用窗口方式。虽然窗口方式要方便得多, 但是在窗口方式下任何可行的调试操作在顺序方式下同样可行。

例子

```
CV /W SIEVE
CV /T SIEVE
```

第一个例子按窗口方式启动调试器。对于IBM兼容机, 最好用/W选项, 因为缺省 设定的顺序方式对于大多数调试任务来说使用较不方便。

第二个例子按顺序方式启动调试器。IBM PC机如果有特殊理由需要使用顺序方式，就可以用/T选项。例如，当把调试输出重定向到一台远程终端时，采用顺序方式一般效果较好。

2.5.5 关闭鼠标

选项

/M | -M

如果系统装有鼠标器，那么可以用 /M 选项告诉调试器忽略它。当调试一个使用非Microsoft 鼠标器的程序时，可能需要这个选项。如果用/M选项，被调试的程序仍可使用鼠标器，但调试器本身不能使用它。

对于Microsoft 鼠标器如果用Microsoft Mouse 5.02以前的版本，程序和调试器在鼠标器使用上也会发生冲突。Microsoft C编译器软件包带有最新版的鼠标器驱动程序（MOUSE.EXE）。如用它替换旧版驱动程序，那么在调试中，调试器和被调试的程序都可以使用鼠标器。关于MOUSE.SYS的安装，请见“Microsoft 鼠标器用户指南”。

2.5.6 设置43行显示方式

选项

/43 | -43

对于增强的图形适配器（EGA），可以用/43选项选择43行×80列的文本方式。一般图形适配器（CGA）和单色适配器（MA）没有这种方式。如果调试器没有检测到EGA，它将忽略这个选项。

EGA的43行方式的操作跟EGA、CGA和MA的缺省设定的25行×80列方式是一致的。它的优点是调试器可以在屏幕上同时显示更多的信息，其缺点是文字更小，因此较难读。

例子

CV /43 COUNT COUNT.TXT

这个例子按43行方式启动调试器。但是如果没有硬件（EGA适配器和显示器）支持，调试器就会忽略这个选项。

2.5.7 使用两个显示器

选项

/2 | -2

如果机器上有两个显示适配器和两个显示器，那么用/2选项可以使程序在缺省设定的现行适配器和显示器上进行显示，同时使调试器在另一个适配器和显示器上进行显示。

假设机器上有一个彩色图形适配器（CGA）和一个单色适配器（MA），而且把CGA设置为现行适配器。这时可以调试一个图形程序，让程序在图形显示器上显示图形而让调试器在单色显示器上显示调试信息。

如果用/2选项，在调试显示器上将不能使用鼠标器。

2.5.8 取消某些IBM专用功能

选项

/D | -D

/D选项取消某些IBM专用功能，其中包括非屏蔽中断陷阱和8259屏蔽。对于不

些功能的IBM兼容机,如IBM PC袖珍机、Tandy 1000和AT&T 6300 Plus,必须用这个选项。如果用/D选项,CONTROL-C和CONTROL-BREAK都不起作用,且调试器缺省设置成顺序方式。但是/W选项可以和/D选项一起使用,使调试器设置成窗口方式。

2.5.9 设置某些IBM专用功能

选项

/I | -I

/I 选项强制调试器处理某些IBM专用功能,其中包括非屏蔽中断陷阱和8259屏蔽。对于调试器识别为非IBM兼容机的计算机,如Eagle PC等,这个选项可以使这些计算机处理CONTROL-C和CONTROL-BREAK。这时调试器缺省设置成窗口方式,因此不必规定/W选项。

2.6 调试汇编语言程序

CodeView 调试器可以调试用Microsoft (或IBM) 宏汇编器开发的程序。但调试器在调试这些汇编语言程序时有下列一些限制:

1. 由于宏汇编器产生的目标文件不带行号,所以汇编语言程序不能在源方式下调试。源文件可以装入并在源方式下查看,但必须切换到汇编方式下才能调试。
2. 由于宏汇编器不把每个变量的类型长度信息写在目标文件上,所以CodeView 表达式只能用常数而不能用变量。这表示对汇编语言程序不能使用表达式显示命令、监视点命令、表达式监视命令和表达式跟踪点命令,但是可以使用内存显示命令、内存监视命令和内存跟踪点命令。
3. 符号名在汇编时通常被转换成大写字母。如果在源程序中使用了小写的符号名,那么在Options菜单上应关闭Case Sense选项。
4. 仅当使用Microsoft 的调用约定(见第二篇第十章)时,在Calls菜单上(或者用堆栈跟踪命令)才能显示过程。与此类似,仅当局部变量按Microsoft 高级语言的方式加以存储,它们才能被存取。
5. 若要查看全程符号,必须规定模块名。

调试器可以使用.EXE或.COM的文件,但是只能在.EXE文件上观察符号信息。汇编和调试.EXE文件的过程小结如下:

1. 在源文件中,把想要让调试器引用的所有符号,如标号和变量,都说明成公用符号。
2. 跟通常一样进行汇编,不要求任何特殊选项,而且可以使用所有的汇编选项。
3. 用Microsoft C编译器(4.0及以后版本)所带的连接器进行连接。不要使用Microsoft 宏汇编器(4.0及以前版本)所带的连接器。在连接时必须用/CODEVIEW选项。
4. 在汇编方式下进行调试(这是调试器在启动时未能找到行号信息情况下的缺省方式)。这时虽然不能在源方式下进行调试,但是可以把源文件装入显示窗口并在源方式下进行观察。这样对查看宏定义和注释尤其方便。对于在源文件中说明为公用符号的任何标号或变量,可以用符号名代替地址来显示和引用它们。

上述过程也可用来调试C程序中调用的C库例行程序和汇编语言模块。