

ECLIPSE MV系列机资料

AOS/VS

调试程序和文件编辑程序

用 户 手 册

苏州电子计算机厂情报室

AOS/VS

调 试 程 序 和 文 件 编 辑 程 序
用 户 手 册

093—000246—00

何 炜 青 译

译 者 说 明

本书是根据日本数据通用 (NDG) 公司提供的《AOS/VS调试程序和文件编辑程序用户手册》[Advanced Operating System/Virtual Storage (AOS/VS) Debugger and File Editor User's Manual] 一书译出的。

AOS/VS是ECLIPSE MV系列机所使用的操作系统。调试程序 (DEBUG) 和文件编辑程序 (FED) 是AOS/VS操作系统支持下的系统实用程序 (utility)。通过使用调试程序和文件编辑程序, 用户可动态地调试AOS/VS汇编语言级的程序和编辑AOS/VS磁盘文件。谨在此将这两个实用程序的用户手册译出, 以期对广大用户开发AOS/VS支持下的系统软件有所帮助。

本手册中所涉及的所有操作命令均在教育部的ECLIPSE MV/6000培训样机上进行了验证。翻译中仅对原手册中本人所观察到的明显错误之处进行了更正。

本书承山东海洋学院数据信息研究室主任徐斯同志精心审阅, 并在翻译过程中得到了徐斯同志和本室其他同志的多方帮助。在此一并表示衷心感谢。

由于本人水平所限, 加之时间仓促, 错误和遗漏在所难免, 恳请读者们批评指正。

译 者

1983年6月

前 言

本手册描述AOS/VS操作系统的调试程序 (DEBUG) 和文件编辑程序 (FED) 的用途和操作。使用调试程序和文件编辑程序可调试ECLIPSE—MV/8000汇编语言程序和编辑AOS/VS磁盘文件。

我们是为熟悉ECLIPSE MV/8000汇编语言指令系统和AOS/VS操作系统的程序员而编写本书的。我们也提供一些中等程度的解释和若干实例以便那些相对来说经验较少的程序员也能使用本手册。

推荐的手册：

本手册中所涉及的许多概念的较深刻描述将在DG公司的其他出版物中提供。在某些例子中，你也许需要参考下列手册之一以得到更多的信息。

- AOS/VS连接和库文件编辑用户手册 (AOS/VS Link and Library File Editor User's Manual (093—000245))

- AOS/VS宏汇编参考手册

(AOS/VS Macroassembler (MASM) Reference Manual (093—000242))

- AOS/VS程序员手册

(AOS/VS Programmer's Manual (093—000241))

- AOS/VS和AOS命令行解释程序用户手册

(AOS/VS and AOS Command Line Interpreter (CLI) User's Manual (093—000122))

- ECLIPSE®MV/8000 操作原理

(ECLIPSE®MV/8000 Principles of Operation (014—000648))

读者注意：

我们在本手册中对命令格式有以下约定：arg [, arg] ... \$X

其中 含义

arg 表示必须送入一些宗量。有时我们使用 $\{\begin{smallmatrix} \text{arg1} \\ \text{arg2} \end{smallmatrix}\}$ ，这意味着必须送入这些宗量中的一个。不要送入花括号；花括号仅表示选择一个宗量。

[arg] 可以任选地送入此宗量，不要送入方括号；方括号仅表示其中宗量可有可无。

... 表示可以重复其前面的一项或几项。命令的解释部分会确切地告诉你可以重复的内容。

\$ 表示必须送入换码 (escape) 字符 (ASCII值为33_s)。按ESC键即送入了此字符。DEBUG对escape符的回送是一美元符号。

X 表示必须送入此单个字母。这个字母对DEBUG来说代表一专用的命令。

本手册中所使用的其他约定是：

✓ NEW LINE (换行) 符 (ASCII 值为12_s)

↓ 回车符 (CR) (ASCII值为15_s)

↑ 上箭头 (ASCII值为136_s)

\$ 换码 (ESC) 字符 (ASCII值为33。)

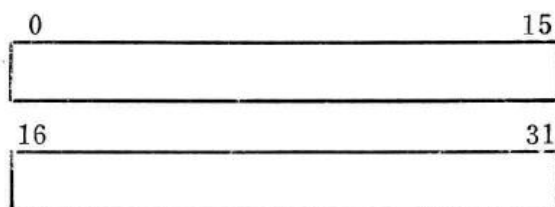
<\$> 美元符号字符 (ASCII值为44。)

CTRL-X 控制字符序列。按住控制键 (CTRL) 后再按另一键 (以X表示)

TAB 水平制表符 (ASCII值为11。)

如果你的键盘不符合ANSI标准, 则↵和↓的意思互换, 即: ↓表示NEW LINE (换行) 而↵表示回车。详见第一章。

我们用下面的格式来表示二进制位段 (bit fields) :



0位是第一位, 我们称之为最有效位 (most significant), 称31位是三十二位中的最无效位 (least significant)。

注意, 我们以16位大小来划分二进制位段, 每一16位段称为一个字。

最后, 我们使用术语“控制台”代表显示 (CRT) 终端和硬拷贝设备。

与DATA GENERAL的联系:

我们欢迎你对本手册和其他DG公司出版物的意见和建议。如果你:

- 有关于本手册的意见, 请使用索引后面的邮资预付的评议表 (Remarks Form)。
- 想要得到其他手册, 请与当地的DG的销售代表处接触。
- 在软件上碰到了问题, 请通知本地的DG公司系统工程师。

此外, 书中黑体字部分是你从控台上超入的信息, 而英文大写字母部分是调试程序输出的信息。

—译者—

目 录

第一章	AOS/VS调试程序引言.....	1
1—1	综述 (overview)	
1—2	DEBUG特性	
1—3	文件编辑 (FED)	
1—4	控制台的区别	
第二章	DEBUG操作过程.....	4
2—1	调试程序的进入	
2—1.1	从 CLI 进入调试程序	
2—1.2	从程序中进入调试程序	
2—2	DEBUG命令	
2—2.1	命令格式	
2—2.2	更正键入错误	
2—2.3	出错响应	
2—3	启动程序执行	
2—4	脱离调试程序	
2—5	一般用途命令	
2—5.1	求助命令	
2—5.2	保存调试会话	
1	打开一个注记文件	
2	关闭一个注记文件	
3	在注记文件中写入注解	
2—5.3	下推入 (pushing) CLI	
第三章	存贮器访问.....	12
3—1	存贮器	
3—1.1	环	
1	AOS/VS的地址	
2	环寄存器 (\$G)	
3	环域运算符 (;)	
3—2	位置计数器和程序计数器	
3—3	检查和修改存贮器	
3—3.1	检查存贮单元	
1	用 / 或 \ 关闭一个存贮单元	
2	以各种显示模式检查存贮器	
3—3.2	修改存贮器单元	

1	合法的键入值	
2	永久性 (permanent) 的存储器修改	
3—4	显示和检索存储器	
3—4.1	显示存储器	
1	显示指令	
3—4.2	检索存储器	
第四章	寄存器访问	24
4—1	检查和修改寄存器	
4—2	机器状态寄存器	
4—2.1	寄存器和进位位	
4—2.2	处理机状态寄存器	
4—2.3	浮点累加器和FPSR	
4—2.4	堆栈寄存器	
4—2.5	程序计数器	
4—3	调试程序寄存器	
4—3.1	基数寄存器	
4—3.2	全局显示模式寄存器	
4—3.3	行进计数寄存器	
4—3.4	环寄存器	
第五章	断点和程序再启动	33
5—1	建立断点	
5—2	显示断点	
5—3	删除断点	
5—4	条件断点	
5—5	行进计数	
5—6	程序再启动	
5—6.1	在一断点处重新启动程序	
5—6.2	在某一位置再启动程序	
第六章	符号识别和再定义	39
6—1	程序符号识别	
6—1.1	禁用一个符号表	
6—1.2	启用一个符号表	
6—2	临时符号	
6—2.1	定义临时符号	
6—2.2	显示临时符号	
6—2.3	删除临时符号	
6—3	专用的调试程序符号	
6—4	检查符号的值	
6—5	地址符号	
6—6	键入符号	

6—6.1	键入符号	
6—7	调试程序的符号显示	
6—7.1	显示符号地址	
6—7.2	以符号模式显示存储器寄存器的内容	
第七章	调试程序表达式	47
7—1	表达式句法	
7—2	运算数	
7—2.1	整型数	
7—2.2	符号	
7—3	运算符	
7—3.1	算术运算符	
7—3.2	逻辑运算符	
7—3.3	关系运算符	
7—3.4	间接运算符	
7—3.5	ASCII字符运算符	
7—3.6	半字压缩运算符	
7—3.7	环域运算符	
7—3.8	运算符的优先顺序	
7—4	专用表达式	
7—4.1	浮点数	
7—4.2	指令	
7—5	检查表达式的值	
第八章	调试程序显示模式	56
8—1	十种显示模式	
8—1.1	数值模式(1)	
8—1.2	数值字模式(2)	
8—1.3	指令模式(3)	
8—1.4	符号模式(4)	
8—1.5	半字模式(5)	
8—1.6	字节指针模式(6)	
8—1.7	ASCII模式(7)	
8—1.8	浮点显示模式(8)	
8—1.9	系统调用模式(8)	
8—1.10	AOS/VS错误信息模式(0)	
8—2	建立显示模式	
8—2.1	带功能键的键盘	
1	局部显示命令	
2	全局显示命令	
8—2.2	不带功能键的键盘	
1	局部显示命令	

2 全局显示命令

8—2.3 全局显示模式寄存器

第九章	调试16位字长的程序.....	69
9—1	累加器	
9—2	堆栈	
9—3	符号的显示	
9—4	覆盖	
第十章	AOS/VS文件编辑程序 (FED)	71
10—1	FED命令	
10—2	FED操作过程	
10—2.1	程序文件和用户数据文件	
1	对前文的寻址	
10—3	AOS/VS系统文件	
附录A	DEBOG错误信息	
附录B	ASCII字符表 (从略)	
附录C	DEBUG命令综述	

插图

图	标题	
2—1	键入 \$H求助命令后的显示	8
3—1	\$S命令	21
5—1	断点逻辑图	36
8—1	功能键	63
8—2	AOS/VS调试程序键盘模板	64
10—1	磁盘上的程序文件	73

表

表	标题	
2—1	一般用途命令	7
3—1	存贮器访问的方法	12
3—2	关闭存贮单元	15
4—1	寄存器命令概述	24
4—2	显示模式值	30
5—1	断点命令综述	33
6—1	符号命令概述	39
6—2	专用的调试程序符号	41
7—1	DEBUG运算符	48
7—2	算术运算符	48
7—3	逻辑运算符	49
7—4	关系运算符	50
7—5	间接运算符	50
7—6	ASCII字符运算符	51
8—1	十个显示模式	56
8—2	各种键盘的显示命令	62
8—3	显示模式寄存器的值	68
10—1	FED命令	71
10—2	FED命令开关	72

第一章 AOS/VS调试程序引言

1—1 综述(overview)

通常，在汇编语言程序开发中最令人生畏的任务乃是对错误的检测、定位和排除，我们称这一过程为调试(debugging)你的程序。

调试过程是在你成功地汇编和连接了程序后进行的(即：汇编(MASM)和连接(LINK)中没出现任何错误)。这时，如果执行你的程序，而程序不做预期的动作，你就必须改正错误(或“bugs”)。(汇编语言程序开发过程的概述可见《AOS/VS宏汇编参考手册》一书)。

有时你可能很容易地检测到程序中的错误，对这种情况，直接改正源程序文件，然后重新汇编和连接之。

如果你不能容易地探测出汇编语言程序中的错误，就可使用DEBUG实用程序。通过使用DEBUG，可以监视并控制程序的执行，也就是说，可以在任意位置停止程序执行，检查或修改存储器寄存器的内容，然后在任意位置继续执行程序。

在定出了程序中错误所在的位置后。离开DEBUG并改正源文件，然后，重新汇编和连接此程序。最后，执行这一新程序文件以确证你的确已改正了有关错误。

1—2 DEBUG特性

下面是AOS/VS调试程序(DEBUG)及其特性的概述表，详尽的说明将在本手册的其他章节给出：

交互式调试 AOS/VS调试程序是一个交互式的程序，即：DEBUG在你送入一个命令后执行此命令

存储器访问
(第三章)

寄存器访问
(第四章)

断点(breakpoint)
(第五章)

程序符号识别
(第六章)

临时符号定义
(第六章)

一次。这使得你在调试会话期间可根据DEBUG对先前所执行命令的响应来作出决定。

AOS/VS调试程序提供了若干条允许你检查和修改存储器的命令。命令的选用项允许你一次观察一至四个字、查找特定的值以及在存储器中步进或步退。

AOS/VS调试程序允许你检查和修改下列ECLIPSE@MV/8000机器状态寄存器的内容：进位位、定点累加器、堆栈寄存器、浮点累加器、程序计数器和处理机状态寄存器。此外，调试程序提供了四条访问DEBUG内部变量(例如：输出基数)的专用寄存器的命令。

DEBUG允许你在程序中设置20₁₀个断点。断点是程序中你希望在此处停止执行的位置。当程序遇到一个断点时，你可以检查和修改存储器和寄存器的内容。

DEBUG自动地识别程序中的全局符号(即：·ENT伪操作语句中说明的符号)。通过使用DEBUG命令，你可以按照自己的意愿启用(enable)和禁用(disable)符号表文件。

DEBUG允许你在调试会话期间定义多达20₁₀个符号。

表达式 (第七章)	当向调试程序传送数值时, 可以输入数学表达式, DEBUG 提供了若干运算符 (例如: 算术、逻辑、关系、间接运算符)。
显示模式 (第八章)	你可以指示DEBUG调试字长为16位的程序 (即: 在16位AOS下开发的但重新连接后在32位字长的AOS/VS操作系统上使用的程序)。DEBUG对16位程序提供全面的支持。
一般用途的命令 (第二章)	AOS/VS调试程序提供使得DEBUG特别便于使用的专用命令。通过使用这些命令, 你可以生成一个子CLI进程并在一个注记文件中保存你的调试会话的付本; 此外, 调试程序还有一种“求助” (help) 命令——它可以给出全部DEBUG命令的信息。

1—3 文件编辑(FED)

AOS/VS调试程序仅可用于可执行的程序文件, 即: PRV类型的文件 (程序文件名通常以.PR后缀结束)。然而, 有一个磁盘文件编辑程序装在AOS/VS DEBUG软件包内, 它被称之为AOS/VS文件编辑程序 (FED)。借助于FED, 你可以检查和修改任何类型的文件——而不仅仅限于可执行的程序文件。

FED命令是DEBUG命令的一个子集。当你编辑一文件时, 此文件并不是正在被执行, 因此, 所有与程序执行、断点和机器状态寄存器有关的命令在FED中均为非法的。

关于AOS/VS FED实用程序的全面说明请见第十章。

1—4 控制台的差别

AOS/VS调试程序支持各种类型的控制

台——既支持CRT, 也支持硬拷贝设备。大多数情况下, DEBUG命令对所有的控制台是一样的。

然而, 某些DEBUG命令因键盘设计的不同而异。更确切地说, 某些DEBUG命令因下述键盘特点而异:

- 键盘是否带有功能键。
- 键盘是否符合ANSI标准。

功能键是位于键盘顶端的空白键, DG的DASHER™ D1 (6052)、D2 (6053)、D3、D4、D5和D200视频显示终端都带有功能键。DASHER D100终端和硬拷贝设备TP₁和TP₂不带功能键。DEBUG显示模式命令因你的键盘是否带有功能键而不同。第八章“建立显示模式”部分对命令的区别提供全面的讨论。

美国国家标准学会 (ANSI) 已制订了一个终端键盘上按键排列的标准。按照这一标准, NEW LINE (换行) 键要大于其他键——由于换行键使用得很频繁。然而, 在某些非ANSI标准的键盘上, 回车 (CR) 键而不是换行键是较大的键。

当你正在检查存储器时, NEW LINE和回车对于调试程序有特定的意义。NEW LINE符仅关闭一个开启的存储单元而回车则除了关闭外还打开下一个单元。第三章中“检查存储器单元”一节和表3—2提供使用NEW LINE和回车的更多信息。

如果你使用的是非ANSI键盘, 则NEW LINE和回车的含义颠倒, 即: 回车仅关闭一个开启的存储单元而NEW LINE除关闭外还打开后继的单元。总之, 大的键 (或是NEW LINE或是回车) 关闭一个开启的单元, 而二者中较小的键关闭一个开启的单元, 同时还打开下一个单元。

在本手册中, 我们用↵代表NEW LINE符, ↓代表回车, 若你使用的是非ANSI键盘, 两者的含义互换。

请注意：DEBUG用终端的设备特性 (device characteristics) 来测定键盘是否具有功能键和是否符合 ANSI 标准。请参考

《AOS/VS和AOS命令行解释程序 (CLI) 用户手册》中关于CHARACTERISTICS命令的说明以得到更多的信息。

第二章 DEBUG操作过程

本章有两大主要部分。第一部分解释进入和脱离AOS/VS调试程序的各种方法，它也给出调试程序命令的一般情况并解释在调试程序中怎样启动程序执行。

第二部分名叫“一般用途的命令”，它解释怎样保存你的调试会话的付本，怎样生成一个子CLI进程以及怎样使用“求助命令”。

2—1 调试程序的进入

要想运行AOS/VS调试程序，必须满足下述一般要求：

- 你的用户说明文件 (user profile) 必须允许你生成一个子进程。
- 你必须保留四个 I/O 通道畅通以便 DEBUG 专用。

假定你满足了这些要求，就可用下述两种方式之一进入调试会话。

1°发一条CLI中的DEBUG命令。

2°在你的汇编语言程序中发 ? DEBUG 系统调用。

以下两段分别解释请求AOS/VS调试实用程序的方法。

2—1.1从CLI进入调试程序

在大多数情况下，你将从AOS/VS命令行解释程序 (CLI) 请求进入调试程序。可直接键入下面的命令：

DEBUG program—file [program—arg] ...
这里：

DEBUG是请求AOS/VS调试程序的CLI命令。

program—file是你希望调试的程序文件的路径名，当键入文件名时，不必带PR后缀。

Program—arg是程序文件执行所要求的一个或多个宗量，并非所有的程序都要求宗量，故此项为选用项。

DEBUG命令无开关，若你的程序使用开关，可将开关紧跟在DEBUG命令行中 Program—file之后。

DEBUG装入你的程序之后，它显示四个累加器和进位位的值，然后显示一个下划线符号，下划线是DEBUG的提示符，例如：

) **DEBUG PROG1**↵

```
AOS/VS Debugger—Rev 001.000.000.000
00000000000 00000000000 00000000000
00000000000 00000000000—
```

紧随Rev后的数字指示正在使用的AOS/VS调试程序软件的版本，第二行的五个数是AC₀、AC₁、AC₂、AC₃和进位位的八进制值。

第三行显示一个下划线符，无论DEBUG何时显示此提示符，都表示它可以接受命令了。

当第一次看到DEBUG提示符时，你的程序还未被开始执行。因此，在开始执行程序之前，你可以设置断点并修改存储器或寄存器。要开始执行你的程序，请使用 \$R 命令。（见本章稍后的“启动程序执行”一节）

可以在CLI PROCESS命令中使用 /DEBUG开关来代替发DEBUG命令。例如：

) **PROCESS/DEBUG/DEFAULT/BLOCK**
/IOC PROG1↵

此命令将PROG1程序装入并开始调试会话。参见《AOS/VS和AOS命令行解释程序 (CLI) 用户手册》以获得关于PROCESS命令和其他PROCESS开关的更多信息。

当进入调试会话后，DEBUG自动地寻找你的程序符号表文件（即：与你的程序同名但带有ST后缀的文件），如果此文件可以使用，则DEBUG在符号识别中使用.ST文件。详见第六章中“程序符号识别”一节。

2—1.2 从程序中进入调试程序

可以从所希望调试的程序中直接进入DEBUG而不是从CLI进入。直接在汇编语言程序中发？DEBUG系统调用即可。和其它系统调用一样，？DEBUG既有错误的也有正常的返回语句。《AOS/VS程序员手册》给出了？DEBUG系统调用的格式。

当你的程序在运行时遇到了？DEBUG系统调用时，控制转给AOS/VS调试程序，DEBUG呈现出四个累加器的值和进位位的值，随后出现DEBUG提示符（下划线）：
AOS/VS Debugger- Rev.001.000.000.000
000000000000 000000000000 000000000000

000000000000 000000000000 000000000000—

无论何时，只要DEBUG显示其提示符，就可以发DEBUG命令。

如果从你的程序中进入DEBUG（即：通过？DEBUG），就必须遵守下面的规则：

- 不可以修改存贮器的内容。
- 不可以设置断点。

所有其它的DEBUG命令均合法。可以检查存贮器的内容，检查和修改各个寄存器的内容。

请注意：程序可能在已处于调试会话时发出？DEBUG系统调用，也就是说：你可以发一条CLI DEBUG命令，然后开始或继续程序的执行，在执行中，程序发出一个？DEBUG系统调用从而控制再次转给调试程序。当以这种方式进入AOS/VS调试程序时，上述的对？DEBUG的限制是无效的。

最后请注意：当用？DEBUG系统调用进入DEBUG时，程序计数器（PC）内容为？

DEBUG的正常返回语句的地址而不是你的程序的启动地址。

2—2 DEBUG命令

在DEBUG显示其提示符（下划线符）之后，你就可以键入指示调试程序去执行某些操作的命令了。

附录C表列出并简洁地描述了所有的DEBUG命令；详细的说明则贯穿本手册。

2—2.1 命令格式

DEBUG命令遵循下述的一般格式：

[arg]...\$X

这里：

arg 是DEBUG命令的一个或多个宗量，并非所有的命令都要求有宗量；但有些命令需要几个宗量。如果用了多个宗量，请用逗号隔开。

\$ 代表escape符（ASCII值为33），通过按键盘上的ESC键送入此字符。DEBUG对ESC键的回送为一美元符号。

X 是A~Z中的单个大写字母。此字母代表一特定的DEBUG命令。

当DEBUG遇到了一个escape符（即：上述格式中的\$），它就等待一单个的命令字符（即：X）。你一键入命令字符后，它就立即执行这个命令。你不必用NEW LINE或回车来结束DEBUG命令行。

DEBUG对escape符的回送为控制台上的美元符号（\$）。下面，在本手册中，我们用\$符号代表escape符而不代表美元符号。当我们要引用美元符号（ASCII值为44）时，就将美元符号以角括号括起来，即：<\$>代表美元符号。请参见前言中关于本手册命令格式约定条目。

虽然大多数命令符合上述的一般命令格式，但打开或关闭存贮器的命令却不然（即：

/, \, ↵, ↓和↑命令)。这些命令的完整说明请见第三章。

2—2.2 更正键入错误

如果当键入一数值或命令时出了一个错误, 你可以改正此错误, 为此只需按:

- RUBOUT 或 DEL [删除 (delete)] 键。
- CTRL—U 命令。

按RUBOUT或DEL键可将紧前面键入的字符去掉。

用CTRL—U命令可删除所键入的整个项。按CTRL—U可删除先前一个提示符以后的所有东西。要键入CTRL—U命令, 可按住CTRL键同时再按U。

你不能将escape符删除。如果出了一个错且已按了ESC键, 可键入NEW LINE 或回车, 调试程序将回送一个问号后跟一提示符, 然后就可键入正确的命令。

2—2.3 出错响应

偶然地, 你会键入了一个调试程序不理解的命令或表达式。DEBUG以下述两种方式之一回答你的错误:

- 若错误导致一个无定义的符号, DEBUG 回答U?
- 对其它所有错误, DEBUG简单地回答一个问号: ?

在DEBUG回答U? 或? 后, 它显示其提示符(下划线符)。通常, 你能容易地找出错误, 对这种情形, 直接使用正确的命令即可。

下面的例子显示了非法的命令和DEBUG对错误的响应。

```
-429,439 $$ 9 是一个非法的八进制数字。
? -
-FILE1 $X DEBUG找不到文件FILE1。
? UNABLE TO OPEN OR ACCESS
SPECIFIED FILENAME
```

-START/ U? 符号START未定义。

-7 \$A? - 不存在7号累加器。

若你不理解为何调试程序回答出错, 你可发 \$? 命令。这个命令指示DEBUG打出它最近遇到的错误的错误诊断信息。命令的格式为:

\$?

这里:

\$ 代表escape (ESC) 符。

? 指示DEBUG显示它最近遇到的错误的错误诊断信息。

下面的例子展示了非法的DEBUG命令和其后使用 \$? 命令的情形:

```
-429,439 $$
```

```
? -$?
```

```
ILLEGAL OCTAL DIGIT IN NOMBER
```

```
-START/ U? -$?
```

```
UNDEFIND SYMBOL
```

```
-7 $A
```

```
? -$?
```

```
UNRECOGNIZED COMMAND FORMAT
```

在例子中, 我们在DEBUG回答出错后键入 \$? 命令, DEBUG然后就显示相应的错误信息。

附录A列出了当使用 \$? 命令时 DEBUG能回答的各种错误信息。

2—3 启动程序执行

当你从CLI进入调试程序时, 程序并未开始执行。为了启动程序执行, 发如下的命令:

```
[address] $R
```

这里:

address 是一可选的数、符号或表达式, 它们代表程序中的一个地址。

\$ 代表escape (ESC) 符。

R 指示DEBUG启动程序执行。

如果没向 \$R命令递交address, 则程序在程序计数器(PC)的当前值处开始执行。

当使用CLI DEBUG命令进入调试程序时，PC的值为你的程序指定的最初的启动地址。若你通过? DEBUG进入，PC的值为? DEBUG正常返回语句的地址。关于“程序计数器”请见第三章。

若要在某地址而不是在PC的当前值处开始程序的执行，可向\$R命令的地址宗量送入一个地址，DEBUG将在提供的地址处启动程序。

在你发出\$R命令之后，程序将一直运行到遇见一个断点或程序结束为止（即：执行一个? RETURN系统调用或通过? TERM结束自身）。如果程序遇见了一个断点，执行就停止，将给出一个DEBUG提示符。在此断点处，可以检查或修改存储器器和寄存器。我们将在第五章中描述断点。

要在一断点处继续程序的执行，请发\$R或\$P命令。当你第一次开始执行时，不可使用\$P命令，\$P命令只能在遇到断点后使用。详见第五章中“程序再启动”一节。

2—4 脱离调试程序

可用下述三种方式之一结束调试会话

- 发\$Z命令。
- 键入CTRL—D。
- 正在调试的程序结束了。

通常，你将通过键入下面的调试程序命令终止调试会话：

\$Z

这里：

\$ 代表escape (ESC) 符。

Z 指示DEBUG终止调试会话。

\$Z命令终止调试程序，并将控制转给父CLI进程。\$Z总是将控制送还给CLI而不管你是如何（是经过CLI DEBUG命令还是? DEBBUG系统调用）进入调试程序的。

虽然\$Z命令是一个离开调试会话的较好手段，但是，CTRL—D也将终止调试程序。

CTRL—D是一个文件尾 (end—of—file) 指示，当从键盘（即：从总称文件@CONSOLE）发CTRL—D时，结束调试会话。要键入CTRL—D，按住CTRL键后再按D。

如果在调试程序中你的程序结束了，则调试会话也结束。也就是说，如果你键入一个\$R或\$P命令（启动或继续程序的执行）且程序在断点前遇到了一个? RETURN系统调用语句，DEBUG将终止。

2—5 一般用途命令

AOS/VS DEBUG 给出几个一般用途的命令，这些命令执行与任何特定DEBUG操作无关的有用功能。我们在这里，即本手册的前部给出。是因为它们在程序调试会话的所有阶段都将很有用处。

表 2—1 描述一般用途命令。

表 2—1 一般用途命令

命 令	功 能
\$C	生成一个子CLI进程 [即：下推入 (Push) CLI]。
\$H	求助命令：列出DEBUG能提供有关信息的各种题目 (topic)。
Keyword \$H	求助命令：提供 keyword 所标识的题目的特有信息。
\$Y	关闭当前的注记文件。
logfile \$Y	关闭当前的注记文件，如果有的话，同时打开一新的注记文件。
; Comment	将注解字符行 Comment 送入当前的注记文件。

本章的其余部分将详细描述每个一般用途命令。

2—5.1 求助命令

调试实用程序提供一个显示关于使用各个DEBUG命令的信息的求助 (help) 命令。发