

微型计算机应用专辑 ⑪

IBM PC 实用调试软件

集 锦

许利民 郭晋渝 李 选

编 著



软件报编辑部

968709

TP31

TP31
3827

3827

软件报专辑第八期

IBM PC 实用调试软件集锦

微型计算机应用专辑(十一)

许利民 李 选 郭晋渝 编 译

郭晋渝 审 校

软件报编辑部

1989.10 成都

105000

责任编辑：袁晓义

封面设计：袁晓梅

* * * *

需要与本书配套软件的读者请与成都237信箱
科技公司开发部联系，邮政编码610015。

* * * *

微型计算机应用专辑如有错装漏页承印厂负责调换

出版：《软件报》社

印刷：成都华新联营印刷二厂

地址：成都金河街75号

地址：成都石羊石桥

广告经营许可证：292号

国内统一刊号：CN51—0090

定价：13.00元

编者的话

近几年来,我国的微型计算机应用已遍及各行各业,微型机应用技术发展很快,很多用户都渴望掌握新的微机应用知识,以不断充实自己迎接新的挑战。为了使更多的人能够较快掌握微机应用新技术把微机用得更好,在生产和管理中发挥更大作用,创造出更好的效益,我们拟定组织有关方面的同志编辑一套《微型计算机应用专辑》以满足从事微机开发,应用专业人员和广大读者的需要。

本书专为高中级软件编制,调试、分析人员而编写,着重介绍世界上精华调试软件的使用方法,使您迅速掌握这些软件,在生产和科研中达到事半功倍的效果。全书的主要内容有:

PC—KIT的功能及特点

PC—KIT是一个智能化的开发工具软件包,应用范围有:

- 1、IBM—PC/XT/AT机软、硬件的维护及管理;
- 2、IBM—PC/XT/AT机宏汇编语言的程序设计调试及学习;
- 3、IBM—PC/XT/AT机机器指令程序(.com和.EXE文件)的测试,功能分析、改造和移植,也是软件汉化的有力工具。

PC—KIT由于采用了多级菜单人机对话方式,简单、宜学。它不仅是IBM—PC机硬、软件系统人员的有力工具包,而且对于一般的计算机人员,也是一种不可多得的智能工具软件。

高级智能反汇编软件ASMGEN和DISASM

ASMGEN和DISASM能够将IBM—PC机的各种可运行文件(.com和.EXE)的指令程序自动转换成用符号和标号表示的宏汇编ASCII源程序和数据区。DISASM还能将内存块,特别是ROM里的指令程序反汇编成宏汇编源程序。它们是软件分析、移植和汉化的必备工具。若能将它们和PC—KIT结合起来使用,更是如虎添翼。

调试工具软件

所有版本的DOS都配备有DEBUG调试程序。对于经验丰富的程序员来说,它是一个高速有效的调试工具,但其有限的功能和低级的键盘命令的方式及屏幕方式限制了它的用途。

SYMDEB符号调试程序,是对“十年一贯制”的DEBUG调试程序的一个升级换代产品。它的命令和参数格式与DEBUG完全兼容,同时还增加了许多新的功能。例如符号调试、断点及其通过次数的设置、堆栈反向跟踪,灵活实用的命令参数(可用二进制、八进制、十进制、十六进制、ASCII、整型数、浮点、实型数等)等新增功能,支持对Microsoft C、FORTRAN、PASCAL等多种语言的源程序显示和符号调试,使人觉得耳目一新,有“更上一层楼”之感。

SID—86是著名的DIGITAL RESEARCH公司的产品,它也是一个符号调试程

序。它是8位微机上的著名调试软件ZSID在IBM—PC机上的实现。它的命令和参数格式不但同ZSID完全兼容，同时增加了许多新的命令。该程序的功能大致和上面介绍的SYMDEB相当。

FSD(又名Windows debug)在此之前曾经是IBM公司内部秘而不宣的全屏幕、多窗口、交互式调试工具软件。用户可以通过各个窗口同时监测程序、数据、堆栈、和系统及寄存器状态。并将调试屏幕和用户软件运行显示屏幕分开，随时都可以自由切换。命令简单、使用方便灵活，大部分调试工作只需按动功能键F1~F10就可完成。并带有联机帮助文件，供用户随时查阅。

PFIX是著名软件公司PHONIX的产品，它除了具有FSD的绝大部分功能外，用户还可根据自己的需要，自由定义各窗口的位置和尺寸。更奇妙的是，它不象别的符号调试程序，调试时所用的符号表必须在编译时就已生成好；而可由用户随时自由定义符号和标号。这对于调试没有源程序的.com和.EXE文件非常有用，使用户完全摆脱了对内存实际地址进行记忆的烦恼，只须按自己临时定义的符号和标号，就可非常方便地调试和运行程序了。同时，它还支持对协处理器8087的调试。

codeview是Microsoft公司近两年推出的全屏幕调试软件，用户可根据自己的情况，选用鼠标、屏幕下拉式菜单或键盘命令等不同的方式进行软件调试。作为一种符号调试软件，它不但可以汇编指令程序，更重要的是面向高级语言(BASIC、FORTRAN、C、PASCAL等)，能对高级语言源程序进行逐行调试，设置、观察、修改断点的各种变量和结果；在高级语言的调试方式和手段方面，是一个新的飞跃和里程碑。

本书是《微型计算机应用专辑》中的第十一册。

由于时间匆忙，难免有不足之处，请读者给予斧正。

编者

一九八九年十月于成都

目 录

DEBUG调试程序.....	(1)
一、怎样启动DEBUG程序.....	(1)
二、DEBUG命令参数.....	(2)
三、DEBUG命令.....	(4)
四、A (汇编)命令.....	(5)
五、C (比较)命令.....	(7)
六、D (转储)命令.....	(7)
七、E (输入)命令.....	(8)
八、F (填写)命令.....	(9)
九、G (执行)命令.....	(9)
十、H (十六进制算术运算)命令.....	(10)
十一、I (输入)命令.....	(11)
十二、L (装入)命令.....	(11)
十三、M (传送)命令.....	(12)
十四、N (命名)命令.....	(12)
十五、O (输出)命令.....	(13)
十六、P (继续)命令.....	(13)
十七、Q (退出)命令.....	(13)
十八、R (寄存器)命令.....	(14)
十九、S (检索)命令.....	(16)
二十、T (跟踪)命令.....	(16)
二十一、U (反汇编)命令.....	(17)
二十二、W (写)命令.....	(18)
符号调试实用程序SYMDEB	(20)
一、概述.....	(20)
二、符号调试准备.....	(20)
2.1使用MASM的符号调试准备工作.....	(20)
2.2使用语言编译器时的符号调试准备工作.....	(21)
2.3使用MAPSYM程序建立符号文件.....	(23)
三、SYMDEB的启动.....	(24)
3.1 仅用一个可执行文件启动SPMDEB.....	(24)
3.2 SYMDEB进行启动符号调试.....	(25)

3.3	将参数传给已装入的程序.....	(26)
3.4	不使用文件启动SYMDEB	(26)
四、	使用SYMDEB的选择项.....	(27)
4.1	指定IBM兼容方式	(27)
4.2	置交互式断点键.....	(27)
4.3	为非IBM硬件设置非屏蔽中断.....	(28)
4.4	允许屏幕翻动.....	(28)
4.5	指定启动命令.....	(28)
五、	命令参数的说明.....	(29)
5.1	符号.....	(29)
5.2	数.....	(30)
5.3	地址.....	(31)
5.4	地址范围.....	(31)
5.5	目标范围.....	(32)
5.6	行号.....	(32)
5.7	串.....	(33)
5.8	表达式.....	(33)
六、	使用SYMDEB命令.....	(34)
6.1	汇编命令 (Assemble)	(35)
6.2	断点命令 (Breakpoint)	(37)
6.3	注释命令 (Comment)	(40)
6.4	比较命令 (Compare)	(40)
6.5	显示命令 (Display)	(41)
6.6	内存显示命令 (Dump)	(42)
6.7	键入命令 (Enter)	(47)
6.8	检查符号映象命令 (Examine Symbol map)	(51)
6.9	填充命令 (Fill)	(53)
6.10	GO命令 (GO)	(53)
6.11	求助命令 (Help)	(54)
6.12	Hex命令 (Hex)	(55)
6.13	输入命令 (Input)	(55)
6.14	装入命令 (Load)	(55)
6.15	移动命令 (Move)	(56)
6.16	命名命令 (Name)	(57)
6.17	打开映象命令 (Open Map)	(58)
6.18	输出命令 (Output)	(59)
6.19	P跟踪命令 (Ptrace)	(59)
6.20	退出命令 (Quit)	(60)

6.21	重建的命令 (Redirection)	(61)
6.22	寄存器命令 (Register)	(62)
6.23	屏蔽翻动命令 (Screen Swap)	(65)
6.24	查寻命令 (Search)	(65)
6.25	设置源方式命令 (Set Source Mode)	(65)
6.26	Shell Escape命令 (Shell Escape)	(67)
6.27	源行命令 (Source Line)	(67)
6.28	栈跟踪命令 (Stack Trace)	(68)
6.29	符号设置命令 (Symbol Set)	(68)
6.30	跟踪命令 (Trace)	(69)
6.31	逆汇编命令 (Unassemble)	(71)
6.32	察看命令 (View)	(73)
6.33	写命令 (Write)	(74)
七、	SYMDEB实例	(75)
7.1	汇编和装入	(79)
7.2	用SYMDEB检查一个程序	(79)
八、	SYMDEB出错信息	(86)

程序维护器 MAKE (88)

一、	概述	(88)
二、	MAKE的使用	(88)
2.1	创建MAKE的描述文件	(88)
2.2	MAKE的启动	(89)
2.3	使用MAKE的选择项	(90)
2.4	宏定义的使用	(90)
2.5	宏定义的嵌套	(91)
2.6	使用特殊的宏	(92)
2.7	推导规则	(92)
三、	程序的维护, 一个例子	(93)
四、	MAKE出错信息	(94)

PC KIT机器指令程序的辅助分析与调试工具 (96)

一、	PC—KIT的开发背景和设计思想	(96)
二、	PC—KIT使用指南	(99)
2.1	DOS+子系统	(99)
2.2	ASM+子系统	(100)
2.3	DEBUG+子系统	(100)
三、	PC—KIT的程序实现	(113)
四、	IBM—PC机器指令程序的汉化技术	
	——PC—KIT软件的应用	(123)
4.1	IBM—PC机器码程序的汉化技术	(123)
4.2	机器码程序的汉化实例	(125)

IBM PC机器指令程序的高级反汇编技术 (129)

一、	概述	(129)
二、	ASMGEN软件的功能与使用	(131)
三、	ASMGEN软件的设计与实现	(135)

符号调试程序SID—86 (227)

第一节	运行SID—86	(227)
第二节	SID—86的表示方法	(229)

第三节	SID—86命令	(233)
第四节	默认的段值	(249)
第五节	A和L命令的汇编语句语法	(251)
源程序调试工具Code View		(254)
第一节	概述	(254)
1.1	Code View的运行环境	(255)
1.2	Code View的启动	(260)
第二节	使用指南	(261)
2.1	Code View的窗口显示方式	(261)
2.2	Code View的顺序显示方式	(263)
2.3	Code View的功能键命令	(264)
2.4	Code View的菜单命令	(266)
2.5	Code View的对话命令	(273)
第三节	表达式	(290)
3.1	C表达式	(290)
3.2	FORTRAN表达式	(293)
3.3	BASIC的表达式	(295)
3.4	Pascal表达式	(296)
3.5	汇编语言表达式	(297)
3.6	行号	(297)
3.7	寄存器	(297)
3.8	地址和地址范围	(298)
3.9	存储器操作符	(298)
全屏幕调试程序FSD		(299)
第一节	FSD的启动	(299)
第二节	FSD的主屏幕	(300)
第三节	主屏幕功能键的功能及使用方法	(304)
第四节	FSD的命令	(309)
第五节	FSD中断点的设置	(315)
第六节	一个用FSD调试程序的例子	(318)
附录A	FSD命令一览表	(322)
附录B	FSD功能键一览表	(323)
附录C	FSD的提示及错误信息一览表	(323)
窗口调试软件.PFIX		(327)
第一节	PFIX的启动	(327)
第二节	PFIX的主屏幕	(327)
第三节	PFIX的初始化设置	(332)
第四节	菜单式命令	(335)
第五节	调试功能键的功能及使用方法	(349)
第六节	调试实例	(351)
附录A	调试功能键一览表	(354)
附录B	菜单式命令一览表	(355)

DEBUG 调试程序

本文说明怎样使用调试程序DEBUG。

DEBUG程序可以被用于：

- 提供一个被控制的测试环境，因而你可以监视和控制被调试程序的执行。你可以直接确定程序中的问题，若此问题已被解决就立即执行程序。你不需要重新汇编程序就可发现修改的是否正确。
- 装入、更改或显示任何文件。
- 执行目标文件。目标文件是机器语言形式的可执行程序。

一、怎样启动DEBUG程序

为启动DEBUG，打入：

```
DEBUG [d, ] [path] [filename [.ext]] [parm1] [parm2]
```

如果你输入文件名(filename)，DEBUG程序把指定的文件装入存储器。你可以用立即命令的方式去更改、显示或执行指定文件的内容。

假如没有输入文件名，你必须用现有的存储器内容工作或者使用Name和Load命令把需要的文件装入存储器。然后，以用户命令去更改、显示或执行存储器的内容。

选择参数parm1和parm2说明了指定文件名的可选参数。例如：

```
DEBUG DISKCOMP - COM A: B,
```

在这个命令中，A:和B:是DEBUG为DISKCOMP程序所准备的参数。

当DEBUG程序启动时，对于被调试的程序，寄存器和标志按下面的值设置：

· 段寄存器(CS, DS, ES和SS)设置在空闲存储器的底部：即DEBUG程序结束之后的第一个段。

· 指令指针(IP)被设置成十六进制的0100。

· 堆栈指针(SP)设置到段的末尾或者装入程序过渡部分较低一个的底部。在位移量是6处的段的大小减去十六进制的100，以便允许建立这么大的堆栈。

· 其余的寄存器(AX, BX, CX, DX, BP, SI, 和DI)设置为零。然而，如果你带着一个指定文件启动DEBUG程序，则CX寄存器包含该文件的字节长度。若文件长度大于64K，则此长度包括在寄存器BX和CX(高位部分在BX)中。

· 标志的最初状态是：

```
NV UP EI PL NZ NA PO NC
```

· 默认磁盘的转移地址设置成代码段中的十六进制数80。

· 所有可用的存储器已被分配。因此，装入程序要去配置存储器的任何尝试都会失败。

如果由DEBUG装入的文件有·EXE的扩展名，DEBUG进行必要的重新定位，并且把段寄存器、堆栈指针和指令指针设置成文件中指定的值。DS和ES寄存器，无论怎样总是指向最低可用段上的程序段级。BX和CX寄存器包含了程序的大小（大小少于此文件）。

如果在连接程序建立该文件时指定了适当的参数，程序被装入存储器的高端。关于装入·EXE文件的更多信息可参阅DOS技术说明书的第9章的“·EXE文件的结构和装入”。

若由DEBUG装入的文件有一个·HEX的扩展名，则文件被认为是INTEL中的十六进制形式，并且在装入期间转换成可执行的形式。

二、DEBUG 命令参数

参 数	定 义
address	<p>以下列形式之一，输入一部分或两部分组成的名称：</p> <ul style="list-style-type: none"> · 一个用字母表示的段寄存器名称，加上一个偏移量，例如： CS:0200 · 一个段地址加上一个偏移量，例如： 4AB:0200 · 只有一个偏移量，例如： 200 <p>（在这种情况下，每个命令使用一个默认段。）</p> <p>注：</p> <ol style="list-style-type: none"> 1. 前两种形式中，要用冒号分隔。 2. 所有数字值是十六进制的，可以输进1—4个字符。 3. 地址中指定的存储器必须是合法的；即它们必须是实际存在的。若试图访问一个不存在的存储器。将产生不可预料的结果。
byte	输入一个或两个十六进制字符值。
drive	<p>输入一个或两个数字（例如，0表示A驱动器，1表示B驱动器）指出将要从哪个驱动器装入数据或者要向哪个驱动器写入。</p> <p>（请参考Load和Write命令）</p>
filespec	<p>输入由一个指定的驱动器、文件名和文件扩展名组成的1至3部分文件的说明。三个方面都是可选的。然而，为了富有意义的命名命令（Name），你至少指定一个驱动器标志符或一个文件名。</p> <p>（参考Name命令。）</p>

参 数	定 义
list	输入一个或多个字节以及/或者字符串值。例如： F3 "XYZ" 8D 4 "abcd" 在表中有五项（即，三个字节输入项和两个字符串输入项。共计十个字节）。
portaddress	输入1—4个字符的十六进制值去指定一个8位或16位的口地址。 （参考Input和Output命令。）
range	输入下列型式之一，去指定低地址和高地址的范围。 • address address 例如： CS:100 110 注：在第二个地址中只允许一个偏移量。地址之间必须用空格或逗号分隔。 • address L value 这里value是由命令处理的十六进制的字节数。例如： CS:100L11
range	注： 1. 范围的极限是十六进制10000，这样，数值的总和以及地址偏移部分不能超出64K字节。用4个十六进制字符去指定一个64K字节的值，输入0000（或者0）。 2. 在range中指定的存储器地址必须是合法的；即，它们必须是实际存在的，若试图访问不存在的存储器地址，将产生不可预料的结果。
registername	参考寄存器命令。
sector sector	输入1—3个字符的十六进制值去指定： 1. 起始相对扇区数目。 2. 被装入或写入的扇区编号的数目。 在DEBUG中，相对扇区是通过计算磁盘表面的扇区得到的。在0道1扇区0磁头（盘上的第一个扇区）上的扇区为6号相对扇区。继续数该道和该磁头上的每个扇区，然后接着数同一道的下一个磁头的第一个扇区。当该道全部磁头上的所有扇区都数完了，继续数下一道上的磁头0上的第一个扇区。 用单个命令可以装入或写入的最大扇区数是十六进制80，一个扇区包括512字节。 （参考Load和Write命令）

参 数	定 义
string	<p>输入用引号括起来的字符串。引号可以用单的 (') 或双的 (")。</p> <p>字符串中字符的ASCII码值被用于字节值的表。</p> <p>在一个字符串内部，成对相向的引号可以当作字符串一样随便使用。然而，若在字符串中必须使用同样的引号（象定界符号）时，引号必须是双重的。这个双重引号不出现在存储器中。</p> <p>例如：</p> <ol style="list-style-type: none"> 1. 'This "literal" is correct' 2. 'This' 'literal' 'is correct' 3. 'This'literal' is not correct" 4. 'This' "'literal'" is not correct" 5. "This 'literal' is correct" 6. "This "'literal'" is correct" 7. "This "literal" is not correct" 8. "This ' 'literal' ' is not correct" <p>上面的第二种和第六种情况中，字literal在存储器中被括在一组引号内。在第四种和第八种情况中，字literal是不正确的，除非你真的想在存储器内把它括在两对括号内。</p>
value	<p>输入1—4个字符的十六进制数值以便确定：</p> <ul style="list-style-type: none"> · 被加上或减去的数（参考Hexarithmic命令），或者 · 用Trace命令去执行的指令数目，或者 · 有效指令的字节数。（参考Trace, Proceed, 和Hexarithmic命令。）

三、DEBUG命令

以下各节详细描述怎样使用DEBUG程序的命令。并按字母表次序排列，分别说明其格式和作用并适当地举例。

所有DEBUG命令的公共信息

- 一个命令是单个字母，通常后跟一个或多个参数。
 - 命令和参数可以输入大写或小写字母，也可以二者合用。
 - 命令和参数可以用定界符隔开，但仅限于两个连续的十六进制数值之间。
- 于是，下面的命令是等价的：

```
dc:100 110
d cs:100 110
d,cs:100,110
```

- 按Ctrl-Break键结束命令。
- 只有按Enter键后，命令才有效。
- 对于产生大量输出的命令，可以按Ctrl-Num Lock键去暂停滚动着的显示，以便你读取。当按任何键时又继续显示。
- 在使用DEBUG程序时，你可以使用DOS用户手册第2章内描述的控制键和DOS编辑键。

· 若遇到一个句法错误，显示该行并以如下方式指出错误：

```
d cs:1φφ Cs:11φ
      ~error
```

在这种情况下，Dump命令期望第二个地址只包含一个十六进制偏移量，它发现S不是一个合法的十六进制字符。

- DEBUG程序的提示符是一个连字符（-）。

四、A（汇编）命令

作用：把IBM PC宏汇编语言的语句直接汇编送入存储器。

格式：A [address]

说明：输入到汇编命令的所有数字都是十六进制的。你输入的汇编语句，被汇编送入在address中指定地址为起点的连续存储器地址中。若没有指定地址，并且前面没有使用汇编命令，该语句被汇编到CS:0100区域；或者汇编到由前面的A assemble命令汇编的最后一条指令的下面。当所有需要的语句输入之后，提示输入下一个语句时，按压Enter键回到DEBUG的提示符。对于不合法的语言，DEBUG将显示：

```
~error
```

并且重新显示当前汇编的地址。

DEBUG支持标准的8086/8088汇编语言句法（以及8087指令），用如下的规则：

- 所有输入的数值是十六进制的，并且可以作为1—4个字符输入。
- 助记符前缀必须在它们的操作码之前输入，也可以单独一行输入。
- 段的代替助记符是，CS，DS，ES，和SS。
- 字符串操作助记符必须表明字符串的大小。例如，MOVSW 必须用于传送字的字符串而MOVSB是传送字节的字符串
- 用于远地址返回的助记符是RETF。
- 汇编程序将自动地汇编短的、近的或远的跳转并且按字节取代到目标地址。这些可以用 NEAR或 FAR前缀取代如下例：

```
0100:0500 JMP 502 ; a 2 byte short jump
```

```
0100:0502 JMP NEAR 505      ; a 3 byte near jump
0100:0505 JMP FAR 50A       ; a 5 byte far jump
```

NEAR前缀可以简写为NE，但FAR前缀不得简化。

· DEBUG不能够告诉某些操作数指的是一个字存储器地址还是字节存储器地址。这种情况下，数据类型必须用前缀“WORD PTR”或“BYTE PTR”来表明。DEBUG也将接收缩写的“WO”和“BY”。例如：

```
NEG BYTE PTR [128]
DEC WO [SI]
```

· DEBUG也不能告诉一个操作数是否指的是一个存储器地址还是一个立即数。DEBUG使用共同约定的方括号内的操作数表示存储器。例如：

```
MOV AX, 21; Load AX with 21H (将21H装入AX中)
MOV AX, [21]; Load AX with the contents of memory location 21H
(将存储器地址21H的内容装入AX中)
```

· 还包括两个通用的伪指令。DB操作码将把字节值直接汇编到存储器。DW操作码将把字值直接汇编到存储器。例如：

```
DB 1, 2, 3, 4, "THIS IS AN EXAMPLE"
DB "THIS IS A QUOTE:"
DB . "THIS IS A QUOTE:"
DW 1000, 2000, 3000: ", BACH."
```

· 支持所有型式的寄存器间址命令。例如：

```
ADD BX, 34 [BP+2] [SI-1]
POP [BP+DI]
PUSH [SI]
```

· 支持所有操作码的同义助记符。例如：

```
LOOPZ 100
LOOPE 100
JA 200
JNBE 200
```

· 对于8087的操作码，WAIT或FWAIT前缀必须被指明。例如：

```
FWAIT FADD ST, ST(3); This line will assemble a FWAIT prefix
本行将汇编FWAIT前缀
FLD TBYTE PTR [BX]; This line will not
(本行不汇编)
```

举例：

```
C> debug
```

```

-a200
08B4:0200 xor ax, ax
08B4:0202 mov [bx], ax
08B4:0204 ret
08B4:0205

```

五、C (比较) 命令

作用：比较两个存储块的内容。

格式：C range address

说明：比较两块存储器的内容；比较的长度由range确定。若发现不相等的字节，就显示它们的地址和内容。其格式如下：

```
addr1 byte1 byte2 addr2
```

其前半 (addr1 byte1) 指的是不匹配的地址和它的内容，后半指的是在 address 中找到的字节。

如果在range的开始地址，你只输入一个偏移量，则C命令认为段包含在DS寄存器中。为range指定一个结束地址，只须输入一个偏移量。

举例：C100 L20 200

表示DS:100开始的32个字节 (十六进制的20) 的存储器与DS:200开始的32个字节的存储器进行比较。比较范围是L20。

六、D (转储) 命令

作用：显示部分存储器的内容。

格式：D [address] 或D [range]

说明：转储被显示为两部分。

1. 十六进制部分，每个字节以十六进制显示。

2. ASCII码部分，每个字节被显示成ASCII字符，不可打印的字符 (ASCII ϕ —31 和127—255) 用句号 (·) 表示。

在40列的系统显示格式中，每一行由8个字节的边界开始，显示8个字节。在80列系统显示格式中，每一行由16个字节的边界开始显示16个字节。第8个和第9个字节之间有一个连字符 (—)

Dump命令有两种任选的格式：

选择1

使用这种选择显示十六进制40个字节 (40列模式) 的内容，或者十六进制80个字节 (80列模式) 的内容。例如：

```
D address
```

或

```
D
```


从指定的地址开始转储其内容。

如果你没有指定地址，D命令则以起始地址是前一个D命令显示的最后一个地址的下面一个地址。因而，可以连续输入没有参数的D命令，转储连续的40个字节或80个字节的范围。

假如前面没有输入D命令，则由DEBUG最初的原始段寄存器指示的段地址开始，指定偏移量为十六进制100。

选择2

使用这个选择可显示确定地址范围的内容。例如：

D range

若起始地址中只输入偏移量，D命令以为段包含在DS寄存器中。若给定结束地址，且用偏移量输入。例如：

D CS:100 10C

40列的显示格式可能看上去如下示：

```
04BA:0100 42 45 52 54 41 20 54 00  
BERTA T,  
04BA:0108 20 42 4F 52 47  
BORG
```

七、E（输入）命令

作用：此命令有两种操作模式：

- 从指定地址开始，用清单中（见选择1）包含的值取代一个或多个字节的内容。
- 在一个指令序列中，显示并且允许更改（看选择2）。

格式：E address [list]

说明：若地址部分只输入偏移量，则E命令认为段包括在DS寄存器中。

Enter命令有两种选择格式：

选择1

使用这种选择把清单（list）内容放入指定起始地址的存储器中。

E address list

例如：

E ds:100:F3 "xyz" 8D

用清单中指定的5个字节装满存储器地址，ds:100到ds:104。

选择2

使用这种选择去显示一个单元的地址和字节，然后系统等待你输入。例如：

E address

输入一个或两个十六进制数的字符去替换这个字节的内容。然后取下面三个动作中任意一个动作：

1. 按压空格键进入下一个地址，显示其内容。若你要改变其内容，请用选择1。要