

## DEBUG 程序的使用

一、在 DOS 的提示符下，可如下键入 Debug 启动调试程序：

**DEBUG [路径\文件名] [参数 1] [参数 2]**

Debug 后可以不带文件名，仅运行 Debug 程序；需要时，再用 N 和 L 命令调入被调试程序。命令中可以带有被调试程序的文件名，则运行 Debug 的同时，还将指定的程序调入主存；参数 1/2 是被调试程序所需要的参数。

在 Debug 程序调入后，根据有无被调试程序及其类型相应设置寄存器组的内容，发出 Debug 的提示符“—”，此时就可用 Debug 命令来调试程序。

- 运行 Debug 程序时，如果不带被调试程序，则所有段寄存器值相等，都指向当前可用的主存段；除 SP 之外的通用寄存器都设置为 0，而 SP 指示当前堆栈项在这个段的尾部；IP=0100h；状态标志都是清 0 状态。
- 运行 Debug 程序时，如果带入的被调试程序扩展名不是.EXE，则 BX.CX 包含被调试文件大小的字节数（BX 为高 16 位），其他同不带被调试程序的情况。
- 运行 Debug 程序时，如果带入的被调试程序扩展名是.EXE，则需要重新定位。此时，CS:IP 和 SS:SP 根据被调试程序确定，分别指向代码段和堆栈段。DS=ES 指向当前可用的主存段，BX.CX 包含被调试文件大小的字节数（BX 为高 16 位），其他通用寄存器为 0，状态标志都是清 0 状态。

### 二、DEBUG 命令的格式

Debug 的命令都是一个字母，后跟一个或多个参数：**字母 [参数]**

命令的使用中注意：

- ① 字母不分大小写；
- ② 只使用 16 进制数，没有后缀字母；
- ③ 分隔符（空格或逗号）只在两个数值之间是必须的，命令和参数间可无分隔符；
- ④ 每个命令只有按了回车键后才有效，可以用 Ctrl+Break 中止命令的执行；
- ⑤ 命令如果不符合 Debug 的规则，则将以“error”提示，并用“^”指示错误位置。

许多命令的参数是主存逻辑地址，形式是“段基地址：偏移地址”。其中，段基地址可以是段寄存器或数值；偏移地址是数值。如果不输入段地址，则采用默认值，可以是缺省段寄存器值。如果没有提供偏移地址，则通常就是当前偏移地址。

对主存操作的命令还支持地址范围这种参数，它的形式是：“开始地址 结束地址”（结束地址不能具有段地址），或者是：“开始地址 L 字节长度”。

### 三、DEBUG 子命令

#### 1、显示命令 D

D (Dump) 命令显示主存单元的内容，它的格式如下（注意分号后的部分用于解释命令功能，不是命令本身）：

**D [地址]** ；显示当前或指定开始地址的主存内容

**D [范围]** ；显示指定范围的主存内容

例如，显示当前（接着上一个 D 命令显示的最后一个地址）主存内容：

左边部分是主存逻辑地址，中间是连续 16 个字节的主存内容（16 进制数，以字节为单位），右边部分是这 16 个字节内容的 ASCII 字符显示，不可显示字符用点“.”表示。一个 D 命令仅显示“8 行×16 个字节”（80 列显示模式）内容。

再如：

**-d 100** ；显示数据段 100h 开始的主存单元

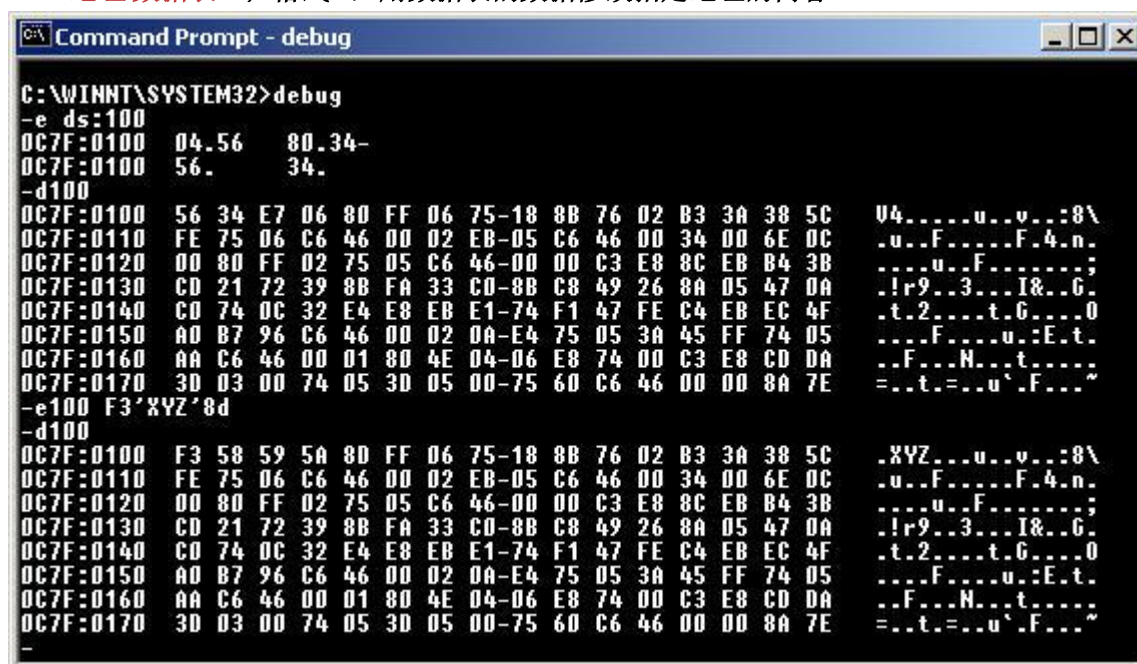
**-d 2f0 L 20** : 显示 ds:2f0h 开始的 20h 个主存数据

**E (Enter)** 命令用于修改主存内容，它有两种格式：

**E 地址**：格式 1，修改指定地址的内容

**E 地址数据表**：格式 2，用数据表的数据修正

Command Prompt - debug



格式 1 是逐个单元相继修改的方法。例如，键入“e ds:100”，Debug 显示原来内容，用户可以直接输入新数据，然后按空格键显示下一个单元的内容，或者按“—”键显示上一个单元的内容；不需要修改可以直接按空格或“—”键；这样，用户可以不断修改相继单元的内容，直到用回车键结束该命令为止。

格式 2 可以一次修改多个单元，例如：

-e ds:100 F3`XYZ8D ；用 F3`X`Y`Z`8D 这 5 个数据替代 DS:0100 ~ 0104 的原来内容

**F (Fill)** 命令用于对一个主存区域填写内容, 同时改写原来的内容, 其格式为:

## F 范围数据表

该命令用数据表的数据写入指定范围的主存。如果数据个数超过指定的范围，则忽略多出的项；如果数据个数小于指定的范围，则重复使用这些数据，直到填满指定范围。

```

C:\WINNT\SYSTEM32>debug
-d200
0C7F:0200  82 93 B4 60 CD 21 BE 2B-93 BF 82 93 E8 93 E3 C3  ...'.!+.-----
0C7F:0210  33 C0 89 3E E6 99 A2 E9-99 A2 EA 99 8A F8 9C 57  3...>-----W
0C7F:0220  33 C9 88 0E 15 98 AC E8-3B E3 75 1D 3C 20 74 F6  3.....;.u.< t.
0C7F:0230  3C 09 74 F2 86 06 EA 99-0A C0 74 EA F6 C7 80 74  <.t.....t...t
0C7F:0240  05 C6 06 15 98 01 E9 2D-01 3A C3 75 05 80 CF 80  .....-.:..u...
0C7F:0250  EB 04 3C 0D 75 03 E9 18-01 3A 06 B6 96 75 03 E9  ..<.u....:..u..
0C7F:0260  17 01 B2 3A 38 14 75 1D-80 3E A4 98 01 75 03 E8  ...:8.u...>...u..
0C7F:0270  EB 0D E8 5C 01 AC E8 58-01 89 3E E6 99 C6 06 E8  ...\....X...>....
-f200 210 11 22 33 44 55 66 77 88
-d200
0C7F:0200  11 22 33 44 55 66 77 88-11 22 33 44 55 66 77 88  .."3DUfw.."3DUfw.
0C7F:0210  11 C0 89 3E E6 99 A2 E9-99 A2 EA 99 8A F8 9C 57  ...>-----W
0C7F:0220  33 C9 88 0E 15 98 AC E8-3B E3 75 1D 3C 20 74 F6  3.....;.u.< t.
0C7F:0230  3C 09 74 F2 86 06 EA 99-0A C0 74 EA F6 C7 80 74  <.t.....t...t
0C7F:0240  05 C6 06 15 98 01 E9 2D-01 3A C3 75 05 80 CF 80  .....-.:..u...
0C7F:0250  EB 04 3C 0D 75 03 E9 18-01 3A 06 B6 96 75 03 E9  ..<.u....:..u..
0C7F:0260  17 01 B2 3A 38 14 75 1D-80 3E A4 98 01 75 03 E8  ...:8.u...>...u..
0C7F:0270  EB 0D E8 5C 01 AC E8 58-01 89 3E E6 99 C6 06 E8  ...\....X...>....
-

```

#### 4、寄存器命令 R

R (Register) 命令用于显示和修改处理器的寄存器，它有三种格式。

**R**；格式 1，显示所有寄存器内容和标志位状态

例如，当我们刚进入 Debug 时，就可以执行该命令，显示示例如下：

```

C:\WINNT\SYSTEM32>debug
-r
AX=0000  BX=0000  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=0C7F  ES=0C7F  SS=0C7F  CS=0C7F  IP=0100  NU UP EI PL NZ NA PO NC
0C7F:0100  A10402      MOV     AX,[0204]      DS:0204=21CD
-rax
AX 0000
:89ab
-rf
NU UP EI PL NZ NA PO NC  -ov cy
-rbx
BX 0000
:15
-r
AX=89AB  BX=0015  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=0C7F  ES=0C7F  SS=0C7F  CS=0C7F  IP=0100  OV UP EI PL NZ NA PO CY
0C7F:0100  A10402      MOV     AX,[0204]      DS:0204=21CD
-

```

其中，前两行给出所有寄存器的值，包括各个标志状态。最后一行给出了当前 CS:IP 处的指令；由于这是一个涉及数据的指令，这一行的最后还给出相应单元的内容。

**R 寄存器名**；格式 2，显示和修改指定寄存器

例如，键入“r ax”，Debug 给出当前 AX 内容，冒号后用于输入新数据，如不修改则按 Enter 键。

**RF**；格式 3，显示和修改标志位

Debug 将显示当前各个标志位的状态。显示的符号及其状态如表 F1.1 所示，用户只要输入这些符号就可以修改对应的标志状态，键入的顺序可以任意。

标志	置位符号	复位符号
溢出 OF	OV	NV
方向 DF	DN	UP
中断 IF	EI	DI
符号 SF	NG	PL
零位 ZF	ZR	NZ
辅助 AF	AC	NA
奇偶 PF	PE	PO
进位 CF	CY	NC

## 5、汇编命令 A

汇编命令 A (Assemble) 用于将输入的汇编指令汇编成为机器代码保存于主存。

**A [地址]** ; 从指定地址开始汇编指令

A 命令中如果没有指定地址, 则接着上一个 A 命令的最后一个单元开始; 若还没有使用过 A 命令, 则从当前 CS:IP 开始。

输入 A 命令后, 就可以输入 8086 指令, Debug 将它们汇编成机器代码, 相继地存放在指定地址开始的存储区中, 记住最后要输入一个回车结束 A 命令。进行汇编的步骤如下:

- ① 输入汇编命令 A [地址], 按回车。Debug 提示地址, 等待你输入新指令;
- ② 输入汇编指令, 按回车;
- ③ 如上继续输入汇编指令, 直到输入所有指令;
- ④ 不输入内容就按回车, 结束汇编, 返回 Debug 的提示符状态。

```

C:\WINNT\SYSTEM32>debug
-a200
0C7F:0200 mov al,05
0C7F:0202 mov ax,500
0C7F:0205 mov ax,bx
0C7F:0207 mov ax,[2000h]
0C7F:0207 mov ax,[2000]
0C7F:020A es:
0C7F:020B mov ax,[2000]
0C7F:020E db 'Hello,Assembly!',d,a
0C7F:0220 dw 2345
0C7F:0222
-u200 20b
0C7F:0200 B005      MOV     AL,05
0C7F:0202 B80005    MOV     AX,0500
0C7F:0205 89D8     MOV     AX,BX
0C7F:0207 A10020    MOV     AX,[2000]
0C7F:020A 26       ES:
0C7F:020B A10020    MOV     AX,[2000]
-d 20e 221
0C7F:0200                                48 65      He
0C7F:0210 6C 6C 6F 2C 41 73 73 65-6D 62 6C 79 20 21 0D 0A llo,Assembly !..
0C7F:0220 45 23                                E#

```

A 命令支持标准的 8086 (和 8087 浮点) 指令系统以及汇编语言语句格式, 但要注意以下一些规则:



- 所有输入的数值都是 16 进制数；
- 段超越指令需要在相应指令前，单独一行输入；
- 段间（远）返回的助记符要使用 RETF；
- A 命令也支持最常用的两个伪指令 DB 和 DW。

## 6、反汇编命令 U

反汇编命令 U（Unassemble）将主存内容按照机器代码形成汇编指令显示：

**U [地址]** ；从指定地址开始，反汇编 32 个字节（80 列显示模式）

**U 范围** ；对指定范围的主存内容进行反汇编

U 命令中如果没有指定地址，则接着上一个 U 命令的最后一个单元开始；若还没有使用过 U 命令，则从当前 CS:IP 开始。

```

C:\WINNT\SYSTEM32>debug d:\masm611\lt301a.exe
-u
0CF2:0000 BAF40C      MOV     DX,0CF4
0CF2:0003 8EDA          MOV     DS,DX
0CF2:0005 8CD3          MOV     BX,SS
0CF2:0007 2BDA          SUB     BX,DX
0CF2:0009 D1E3          SHL     BX,1
0CF2:000B D1E3          SHL     BX,1
0CF2:000D D1E3          SHL     BX,1
0CF2:000F D1E3          SHL     BX,1
0CF2:0011 FA           CLI
0CF2:0012 8ED2          MOV     SS,DX
0CF2:0014 03E3          ADD     SP,BX
0CF2:0016 FB           STI
0CF2:0017 BA0400      MOV     DX,0004
0CF2:001A B409          MOV     AH,09
0CF2:001C CD21          INT     21
0CF2:001E B8004C      MOV     AX,4C00
-u 17 21
0CF2:0017 BA0400      MOV     DX,0004
0CF2:001A B409          MOV     AH,09
0CF2:001C CD21          INT     21
0CF2:001E B8004C      MOV     AX,4C00
0CF2:0021 CD21          INT     21

```

屏幕显示的左边是主存逻辑地址，中间是该指令的机器代码，而右边则是对应的指令汇编格式。

## 7、运行命令 G

运行命令 G（Go）从指定地址处开始运行程序，直到遇到断点或者程序正常结束。

**G [=地址] [断点地址 1,断点地址 2,...,断点地址 10]**

G 命令等号后的地址指定程序段运行的起始地址，如不指定则从当前的 CS:IP 开始运行。断点地址如果只有偏移地址，则默认是代码段 CS；断点可以没有，但最多只能有 10 个。

程序遇到断点（实际上就是断点中断指令 INT 3），停止执行，并显示当前所有寄存器和标志位的内容、以及下一条将要执行的指令（显示内容同 R 命令），以便观察程序运行到此的情况。程序正常结束，将显示“Program terminated normally”。

注意，**G 命令以及后面的 T 和 P 命令要指向正确的指令代码**，否则会出现不可预测的结果，例如“死机”。

## 8、跟踪命令 T

跟踪命令 T（Trace），也称为单步命令，每执行一条指令就显示运行结果，使程序员可以细致地观察程序的执行情况。

**T [=地址]** ；逐条指令跟踪

**T [=地址] [数值]** ；多条指令跟踪

从指定地址起执行一条或数值参数指定条数的指令后停下来，每条指令执行后都要显示所有寄存器和标志位的值以及下一条指令。如未指定地址则从当前的 CS:IP 开始执行。注意给出的执行地址前有一个等号，否则会被认为是被跟踪指令的条数（数值）。

```

C:\>Command Prompt - debug d:\masm611\lt301a.exe
0CF2:001E B8004C      MOV     AX,4C00
-t 0

AX=0000 BX=0020 CX=0038 DX=0CF4 SP=0420 BP=0000 SI=0000 DI=0000
DS=0CF4 ES=0CE2 SS=0CF4 CS=0CF2 IP=0003  NV UP EI PL NZ NA PO NC
0CF2:0003 8EDA      MOV     DS,DX
-t 3

AX=0000 BX=0020 CX=0038 DX=0CF4 SP=0420 BP=0000 SI=0000 DI=0000
DS=0CF4 ES=0CE2 SS=0CF4 CS=0CF2 IP=0005  NV UP EI PL NZ NA PO NC
0CF2:0005 8C03      MOV     BX,SS

AX=0000 BX=0CF4 CX=0038 DX=0CF4 SP=0420 BP=0000 SI=0000 DI=0000
DS=0CF4 ES=0CE2 SS=0CF4 CS=0CF2 IP=0007  NV UP EI PL NZ NA PO NC
0CF2:0007 2BD4      SUB     BX,DX

AX=0000 BX=0000 CX=0038 DX=0CF4 SP=0420 BP=0000 SI=0000 DI=0000
DS=0CF4 ES=0CE2 SS=0CF4 CS=0CF2 IP=0009  NV UP EI PL ZR NA PE NC
0CF2:0009 D1E3      SHL     BX,1
-t

AX=0000 BX=0000 CX=0038 DX=0CF4 SP=0420 BP=0000 SI=0000 DI=0000
DS=0CF4 ES=0CE2 SS=0CF4 CS=0CF2 IP=000B  NV UP EI PL ZR NA PE NC
0CF2:000B D1E3      SHL     BX,1
-

```

T 命令逐条指令执行程序，遇到子程序（CALL）或中断调用（INT n）指令也不例外，也会进入到子程序或中断服务程序当中执行。

## 9、继续命令 P

继续命令 P（Proceed）类似 T 命令，逐条执行指令、显示结果。但是当遇到子程序调用、中断功能调用和循环指令等时，不在子程序、中断服务程序或循环体中单步执行，而是直接执行完成子程序、中断服务程序或循环体，然后显示结果。

当不需要调试子程序、中断服务程序或循环程序段时，要应用 P 命令，而不是 T 命令。

P [=地址] [数值]

```

C:\>Command Prompt - debug d:\masm611\lt301a.exe
0CF2:0017 BA0400      MOV     DX,0004
-p

AX=0000 BX=0020 CX=0038 DX=0004 SP=0420 BP=0000 SI=0000 DI=0000
DS=0CF4 ES=0CE2 SS=0CF4 CS=0CF2 IP=001A  NV UP EI PL NZ NA PO NC
0CF2:001A B409      MOV     AH,09
-p

AX=0900 BX=0020 CX=0038 DX=0004 SP=0420 BP=0000 SI=0000 DI=0000
DS=0CF4 ES=0CE2 SS=0CF4 CS=0CF2 IP=001C  NV UP EI PL NZ NA PO NC
0CF2:001C CD21      INT     21
-p
Hello,Everybody !

AX=0924 BX=0020 CX=0038 DX=0004 SP=0420 BP=0000 SI=0000 DI=0000
DS=0CF4 ES=0CE2 SS=0CF4 CS=0CF2 IP=001E  NV UP EI PL NZ NA PO NC
0CF2:001E B8004C      MOV     AX,4C00
-p 2

AX=4C00 BX=0020 CX=0038 DX=0004 SP=0420 BP=0000 SI=0000 DI=0000
DS=0CF4 ES=0CE2 SS=0CF4 CS=0CF2 IP=0021  NV UP EI PL NZ NA PO NC
0CF2:0021 CD21      INT     21

Program terminated normally
-

```

## 10、退出命令 Q

退出命令 Q (Quit) 使 Debug 程序退出, 返回 DOS。Q 命令并无存盘功能, 可使用 W 命令存盘。

## 11、命名命令 N

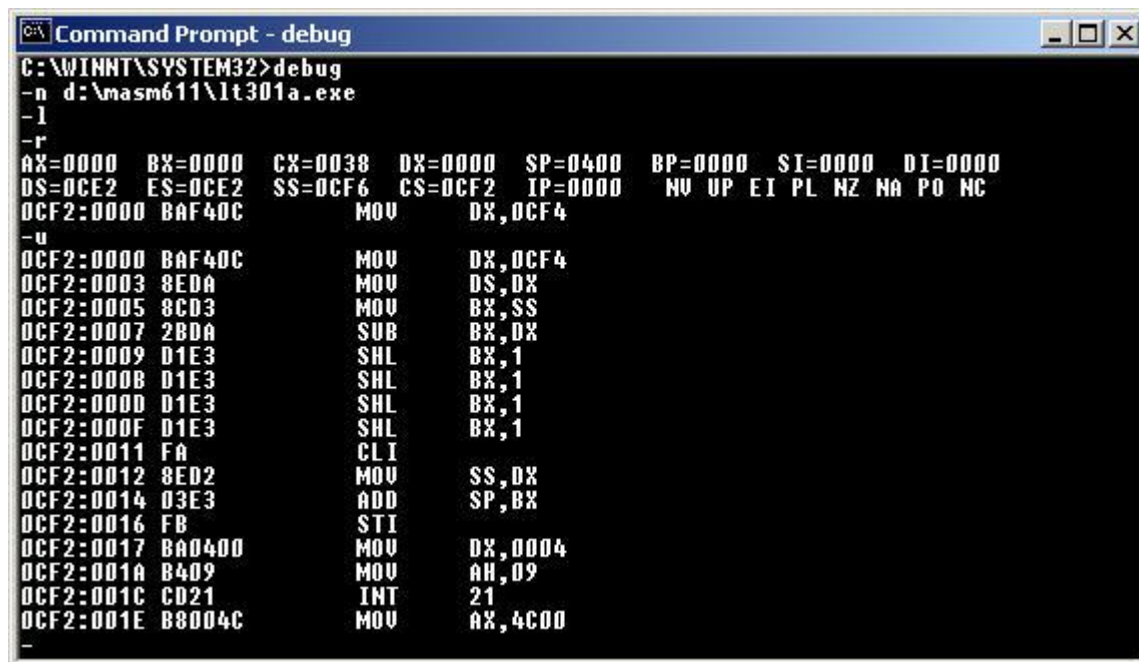
命名命令 N (Name) 把一个或两个可以包含路径的文件全名存入 Debug 中, 以便在其后用 L 或 W 命令把文件装入或存盘。

N 文件名 1[, 文件名 2]

## 12、装入命令 L

装入命令 L (Load) 将磁盘中的文件或扇区内容装载到主存中。

L [地址] ; 格式 1: 装入由 N 命令指定的文件



```
C:\WINNT\SYSTEM32>debug
-n d:\masm611\lt301a.exe
-l
-r
AX=0000 BX=0000 CX=0038 DX=0000 SP=0400 BP=0000 SI=0000 DI=0000
DS=0CE2 ES=0CE2 SS=0CF6 CS=0CF2 IP=0000  NU UP EI PL NZ NA PO NC
OCF2:0000 BAF40C      MOV     DX,OCF4
-u
OCF2:0000 BAF40C      MOV     DX,OCF4
OCF2:0003 8ED0      MOV     DS,DX
OCF2:0005 8CD3      MOV     BX,SS
OCF2:0007 2BD0      SUB     BX,DX
OCF2:0009 D1E3      SHL     BX,1
OCF2:000B D1E3      SHL     BX,1
OCF2:000D D1E3      SHL     BX,1
OCF2:000F D1E3      SHL     BX,1
OCF2:0011 FA      CLI
OCF2:0012 8ED2      MOV     SS,DX
OCF2:0014 03E3      ADD     SP,BX
OCF2:0016 FB      STI
OCF2:0017 BA0400     MOV     DX,0004
OCF2:001A B409      MOV     AH,09
OCF2:001C CD21      INT     21
OCF2:001E B8004C     MOV     AX,4C00
-
```

格式 1 的 L 命令装载一个文件到给定的主存地址处。

L 地址 驱动器 扇区号 扇区数 ; 格式 2: 装入指定磁盘扇区范围的内容

格式 2 的 L 命令装载磁盘的若干扇区 (最多 80h) 到给定的主存地址处; 缺省段地址是 CS。其中, 0 表示 A 盘, 1 表示 B 盘, 2 表示 C 盘, ……。

## 13、写盘命令 W

写盘命令 W (Write) 主存内容写入磁盘的文件或扇区中, 与 L 命令相反。

W [地址] ; 格式 1: 将由 N 命令指定的文件写入磁盘

```

C:\WINNT\SYSTEM32>debug
-a
0C7F:0100 mov ah,9
0C7F:0102 mov dx,110
0C7F:0105 int 21
0C7F:0107 mov ax,4c00
0C7F:010A int 21
0C7F:010C
-a110
0C7F:0110 db 'Hello'
0C7F:0115
-rcx
CX 0000
:15
-n d:\masm611\lt301.com
-w
Writing 00015 bytes
-q
C:\WINNT\SYSTEM32>_

```

格式 1 的 W 命令将指定开始地址的数据写入一个文件（这个文件应该已经用 N 命令命名）；如未指定地址则从 CS : 100 开始。要写入文件的字节数应先放入 BX（高字）和 CX（低字）中。如果采用这个 W 命令保存你的可执行程序，它的扩展名应是 COM；它不能写入具有 EXE 和 HEX 扩展名的文件。

W 地址 驱动器 扇区号 扇区数 ; 格式 2: 把数据写入指定磁盘扇区范围

格式 2 的 W 命令将指定地址的数据写入磁盘的若干扇区（最多 80H）；如果没有给出段地址，则缺省是 CS。其他说明同 L 命令。由于格式 2 的 W 命令直接对磁盘写入，没有经过 DOS 文件系统管理，所以一定要小心，否则可能无法利用 DOS 文件系统读写

#### 14、其它命令

- ① 比较命令 C (Compare)  
C 范围 地址 ; 将指定范围的内容与指定地址内容比较
- ② 16 进制数计算命令 H (Hex)  
H 数字 1, 数字 2 ; 同时计算两个 16 进制数字的和与差
- ③ 输入命令 I (Input)  
I 端口地址 ; 从指定 I/O 端口输入一个字节，并显示
- ④ 输出命令 O (Output)  
O 端口地址 字节数据 ; 将数据输出到指定的 I/O 端口
- ⑤ 传送命令 M (Move)  
M 范围 地址 ; 将指定范围的内容传送到指定地址处
- ⑥ 查找命令 S (Search)  
S 范围 数据 ; 在指定范围内查找指定的数据
- ⑦ 帮助命令?  
? ; 显示各命令的简要说明