

# 汇编语言DEBUG命令详解||汇编命令||DEBUG的常用命令：A,U,R,T,D,E,Q 等等

汇编命令

[汇编语言](#)Debug命令详解

DEBUG的常用命令：A,U,R,T,D,E,Q 等等

## 汇编语言DEBUG命令详解

### R命令的使用

作用：观看和修改[寄存器](#)的值。

在提示符“-”下输入以下命令：R。DEBUG将会显示出当前所有寄存器和标志位的状态。接下来再输入命令rax。在提示符“:”后输入100。该命令的作用是将寄存器ax的值设置为100（注意：DEBUG使用的是十六进制，这里的100相当于十进制的256。）

最后再执行r命令，观看修改后的寄存器值。

```
-r
AX=0000  BX=0000  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=0AE7  ES=0AE7  SS=0AE7  CS=0AE7  IP=0100  NU UP EI PL NZ NA PO NC
0AE7:0100 4D          DEC      BP
-rax
AX 0000
:0100
-r
AX=0100  BX=0000  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=0AE7  ES=0AE7  SS=0AE7  CS=0AE7  IP=0100  NU UP EI PL NZ NA PO NC
0AE7:0100 4D          DEC      BP
-
```

### H命令的使用

H命令作用：计算两个十六进制数的和与差。

在提示符“-”下输入以下命令：h 10 1。观看命令执行结果。

```
-h 10 1
0011  000F
-
```

运行结果的前一个数是计算出来的和，后一个数是计算出来的差。计算结果均用十六进制形式表示。

### D命令的使用

D命令作用：显示内存区域的内容。

在提示符“-”下连续执行命令R、D、D。观看命令执行结果。

```
-r
AX=0100 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0AE7 ES=0AE7 SS=0AE7 CS=0AE7 IP=0100  NU UP EI PL NZ NA PO NC
0AE7:0100 4D          DEC     BP
-d
0AE7:0100 4D 00 00 37 00 65 20 63-73 3A 63 6F 64 65 73 67 M..7.e cs:codesg
0AE7:0110 43 4F 4D 53 50 45 43 3D-43 3A 5C 57 34 00 D6 0A COMSPEC=C:\W4...
0AE7:0120 57 53 5C 53 59 53 54 45-4D 33 32 5C 43 4F 4D 4D WS\SYSTEM32\COMM
0AE7:0130 41 4E 44 2E 43 4F 4D 00-41 4C 4C 55 53 45 52 53 AND.COM.ALLUSERS
0AE7:0140 50 52 4F 46 49 4C 45 3D-43 3A 5C 44 4F 43 55 4D PROFILE=C:\DOCUM
0AE7:0150 45 7E 31 5C 41 4C 4C 55-53 45 7E 31 00 41 50 50 E~1\ALLUSE~1.APP
0AE7:0160 44 41 54 41 3D 43 3A 5C-44 4F 43 55 4D 45 7E 31 DATA=C:\DOCUME~1
0AE7:0170 5C 41 44 4D 49 4E 49 7E-31 5C 41 50 50 4C 49 43 \ADMINI~1\APPLIC
-d
0AE7:0180 7E 31 00 43 4C 49 45 4E-54 4E 41 4D 45 3D 43 6F ~1.CLIENTNAME=Co
0AE7:0190 6E 73 6F 6C 65 00 43 4F-4D 4D 4F 4E 50 52 4F 47 nsole.COMMONPROG
0AE7:01A0 52 41 4D 46 49 4C 45 53-3D 43 3A 5C 50 52 4F 47 RAMFILES=C:\PROG
0AE7:01B0 52 41 7E 31 5C 43 4F 4D-4D 4F 4E 7E 31 00 43 4F RA~1\COMMON~1.CO
0AE7:01C0 4D 50 55 54 45 52 4E 41-4D 45 3D 44 48 2D 43 41 MPUTERNAME=DH-CA
0AE7:01D0 38 38 32 32 41 42 39 35-38 39 00 46 50 5F 4E 4F 8822AB9589.FP_NO
0AE7:01E0 5F 48 4F 53 54 5F 43 48-45 43 4B 3D 4E 4F 00 48 HOST_CHECK_NO_H
0AE7:01F0 4F 4D 45 44 52 49 56 45-3D 43 3A 00 48 4F 4D 45 ONEDRIUE=C:\HOME
```

前面已经介绍过了，命令R的作用是显示当前寄存器的值。而命令D的作用是显示内存区域的内容，最左边是内存的起始地址，中间以十六进制形式显示内存值，最右边是以ASCII码的形式显示内存值。每行最多显示16个字节的内容。

命令D可以带参数也可省略参数。设DEBUG启动时DS的值为X，当省略参数时，命令D显示内容以X:100为起始，每次显示128个字节的内容。以后再执行不带参数的命令D时，DEBUG将按上次的位置接着显示下去。

带参数时DEBUG能够显示指定地址范围的内容。带参数的方式有三种：

方式一：d [起始位置]。DEBUG从起始位置开始显示128个字节的内容。在提示符“-”下执行命令d 1000:100。观看命令执行结果。

```
-d 1000:100
1000:0100 05 0C 00 52 50 E8 48 F9-83 C4 04 89 86 7E FF 8B ...RP.H.....~..
1000:0110 D8 8A 07 2A E4 40 50 53-B8 1E 1D 50 E8 93 23 83 ...*.@PS...P..#.
1000:0120 C4 06 8D 46 80 50 B8 1E-1D 50 E8 EF BF 83 C4 04 ...F.P...P.....
1000:0130 B8 FF FF 50 FF B6 7E FF-B8 1E 1D 50 E8 73 AD 83 ...P...~....P.s..
1000:0140 C4 06 0A C0 75 37 8A 1E-1E 1D 2A FF 88 BF 1F 1D ...u?.....*....
1000:0150 B8 1F 1D 50 B8 CD 0F 50-E8 51 F7 83 C4 04 B8 FF ...P...P.Q.....
1000:0160 FF 50 2B C0 50 B8 1E 1D-50 E8 74 C3 83 C4 06 A1 .P+.P...P.t.....
1000:0170 3A 21 8B 16 3C 21 A3 9C-07 89 16 9E 07 8B E5 5D :!...<!.....l
```

方式二：d [起始位置] [结束位置]。DEBUG从起始位置开始一直显示到结束位置。在提示符“-”下执行命令d 1000:100 200。观看命令执行结果。

```
-d 1000:100 200
1000:0100 05 0C 00 52 50 E8 48 F9-83 C4 04 89 86 7E FF 8B ...RP.H.....~..
1000:0110 D8 8A 07 2A E4 40 50 53-B8 1E 1D 50 E8 93 23 83 ...*.@PS...P..#.
1000:0120 C4 06 8D 46 80 50 B8 1E-1D 50 E8 EF BF 83 C4 04 ...F.P...P.....
1000:0130 B8 FF FF 50 FF B6 7E FF-B8 1E 1D 50 E8 73 AD 83 ...P...~....P.s..
1000:0140 C4 06 0A C0 75 37 8A 1E-1E 1D 2A FF 88 BF 1F 1D ...u?.....*....
1000:0150 B8 1F 1D 50 B8 CD 0F 50-E8 51 F7 83 C4 04 B8 FF ...P...P.Q.....
1000:0160 FF 50 2B C0 50 B8 1E 1D-50 E8 74 C3 83 C4 06 A1 .P+.P...P.t.....
1000:0170 3A 21 8B 16 3C 21 A3 9C-07 89 16 9E 07 8B E5 5D :!...<!.....l
1000:0180 C3 90 56 FF 36 9A 07 8B-1E A6 07 D1 E3 8B 36 9A ..U.6.....6..
1000:0190 16 FF 30 B8 48 0C 50 FF-36 24 21 E8 F2 17 83 C4 ..0.H.P.6$!....
1000:01A0 08 5E C3 90 55 8B EC 56-C4 5E 04 26 80 7F 0A 00 .^..U..U.^.&....
1000:01B0 74 3A 26 83 7F 06 00 74-19 26 8B 5F 06 D1 E3 8B t:&....t.&....
1000:01C0 36 96 16 8B 18 8B 36 A8-25 F6 00 01 74 04 FF 06 6.....6.%...t...
1000:01D0 42 0C BE 46 0C 8B 1C FF-04 D1 E3 D1 E3 8B 36 42 B..F.....6B
1000:01E0 23 8B 46 0C 8B 56 0E 89-00 89 50 02 5E 5D C3 90 #.F..U....P.^l..
1000:01F0 55 8B EC FF 36 98 16 FF-36 D0 25 FF 76 06 FF 76 U..6...6.%..u..u
1000:0200 04
```

方式三：d [起始位置] [L长度]，长度以L参数为标识。DEBUG从起始位置开始显示指定长度的内容。在提示符“-”下执行命令d ds:100 L10。观看命令执行结果。

```
-d ds:100 L10
0AE7:0100  4D 00 00 37 00 65 20 63-73 3A 63 6F 64 65 73 67  M..7.e cs:codesg
```

## E命令的使用

E命令作用：改变内存单位的内容。

E命令的使用方式为：E [起始位置]。

在提示符 “-” 下输入以下命令：e 1000:100。

```
-d 1000:100 L1
1000:0100  01
-e 1000:100
1000:0100  01.05
-d 1000:100 L1
1000:0100  05
```

DEBUG首先显示[1000:100]的内容01.，这时可以修改该字节的值。如果还要修改后续的内容，可以按空格键继续。当要跳过某个字节时，可以按连续的两个空格跳到后一个字节去。

## F命令的使用

F命令作用：使用指定的值填充指定内存区域中的地址。

F命令的使用方式为：F [范围] [填充列表]。

在提示符 “-” 下输入以下命令：F 1AF5:100 L20 1 2 3 4 5。执行命令D 1AF5:100观看命令执行结果。

```
-F 1AF5:100 L20 1 2 3 4 5
-D 1AF5:100
1AF5:0100  01 02 03 04 05 01 02 03-04 05 01 02 03 04 05 01  .....
1AF5:0110  02 03 04 05 01 02 03 04-05 01 02 03 04 05 01 02  .....
1AF5:0120  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....
```

说明：该命令是用字节序列01、02、03、04、05轮流填充从1AF5:100开始长度为20H的内存区域。

在提示符 “-” 下输入以下命令：F 1AF5:100 13F 41 42 43 44。

```
-f 1af5:100 13f 41 42 43 44
-d 1af5:100
1AF5:0100  41 42 43 44 41 42 43 44-41 42 43 44 41 42 43 44  ABCDABCDABCDABCD
1AF5:0110  41 42 43 44 41 42 43 44-41 42 43 44 41 42 43 44  ABCDABCDABCDABCD
1AF5:0120  41 42 43 44 41 42 43 44-41 42 43 44 41 42 43 44  ABCDABCDABCDABCD
1AF5:0130  41 42 43 44 41 42 43 44-41 42 43 44 41 42 43 44  ABCDABCDABCDABCD
1AF5:0140  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....
1AF5:0150  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....
1AF5:0160  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....
1AF5:0170  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....
```

说明：该命令是用字节序列41、42、43、44轮流填充从1AF5:100开始一直到1AF5:13F的内存区域。

## M命令的使用

M命令作用：将指定内存区域的数据复制到指定的地址去。

M命令的使用方式为：M [范围] [指定地址]。

在提示符 “-” 下输入以下命令：M 1AF5:100 13F 1AF5:140。执行命令D 1AF5:100观看命令执行结果。

```
-m 1af5:100 13f 1af5:140
-d 1af5:100
1AF5:0100  41 42 43 44 41 42 43 44-41 42 43 44 41 42 43 44  ABCDABCDABCDABCD
1AF5:0110  41 42 43 44 41 42 43 44-41 42 43 44 41 42 43 44  ABCDABCDABCDABCD
1AF5:0120  41 42 43 44 41 42 43 44-41 42 43 44 41 42 43 44  ABCDABCDABCDABCD
1AF5:0130  41 42 43 44 41 42 43 44-41 42 43 44 41 42 43 44  ABCDABCDABCDABCD
1AF5:0140  41 42 43 44 41 42 43 44-41 42 43 44 41 42 43 44  ABCDABCDABCDABCD
1AF5:0150  41 42 43 44 41 42 43 44-41 42 43 44 41 42 43 44  ABCDABCDABCDABCD
1AF5:0160  41 42 43 44 41 42 43 44-41 42 43 44 41 42 43 44  ABCDABCDABCDABCD
1AF5:0170  41 42 43 44 41 42 43 44-41 42 43 44 41 42 43 44  ABCDABCDABCDABCD
-
```

## C命令的使用

C命令作用：将两块内存的内容进行比较。

C命令的使用方式为：C [范围] [指定地址]，意思就是将指定范围的内存区域与从指定地址开始的相同长度的内存区域逐个字节进行比较，列出不同的内容。

在提示符 “-” 下输入以下命令：C 1AF5:100 13F 1AF5:140。由于两块内容完全相同，所以命令执行后没有任何显示。

在提示符 “-” 下输入以下命令：C 1AF5:100 107 1AF5:180，比较的区域长度为8个字节。命令执行后列出比较结果不同的各个字节。

```
-C 1AF5:100 13F 1AF5:140
-C 1AF5:100 107 1AF5:180
1AF5:0100  41 00 1AF5:0180
1AF5:0101  42 00 1AF5:0181
1AF5:0102  43 00 1AF5:0182
1AF5:0103  44 00 1AF5:0183
1AF5:0104  41 00 1AF5:0184
1AF5:0105  42 00 1AF5:0185
1AF5:0106  43 00 1AF5:0186
1AF5:0107  44 00 1AF5:0187
-
```

## S命令的使用

S命令作用：在指定的内存区域中搜索指定的串。

S命令的使用方式为：S [范围] [指定串]。

在提示符 “-” 下输入以下命令：D 1AF5:100 11F。显示该区域的内存值。

在提示符 “-” 下输入以下命令：S 1AF5:100 11F 41 42 43 44。搜索该区域是否存在字节串41 42 43 44，并将搜索结果——列出。

```
-d 1af5:100 11f
1AF5:0100  41 42 43 44 41 42 43 44-41 42 43 44 41 42 43 44  ABCDABCDABCDABCD
1AF5:0110  41 42 43 44 41 42 43 44-41 42 43 44 41 42 43 44  ABCDABCDABCDABCD
-s 1af5:100 11f 41 42 43 44
1AF5:0100
1AF5:0104
1AF5:0108
1AF5:010C
1AF5:0110
1AF5:0114
1AF5:0118
1AF5:011C
-
```

从执行结果可以看出，总共搜索到八处。

## A命令的使用

A命令作用：输入汇编指令。

以下的程序要在屏幕上显示“ABCD”四个字符。

首先用E命令将“ABCD”四个字符预先放在内存CS:200处，然后执行A100命令输入汇编程序代码：MOV AX,CS MOV DS,AX MOV DX,200 MOV AH,9 INT 21 INT 20（说明：前两行汇编指令用于将段寄存器CS的值赋给段寄存器DS。第三到第五行汇编代码的作用是显示以“ ”四个字符预先放在内存CS:200处，然后执行A100命令输入汇编程序代码：MOV AX,CS MOV DS,AX MOV DX,200 MOV AH,9 INT 21 INT 20（说明：前两行汇编指令用于将段寄存器CS的值赋给段寄存器DS。第三到第五行汇编代码的作用是显示以“ ”四个字符预先放在内存CS:200处，然后执行A100命令输入汇编程序代码：MOV AX,CS MOV DS,AX MOV DX,200 MOV AH,9 INT 21 INT 20（说明：前两行汇编指令用于将段寄存器CS的值赋给段寄存器DS。第三到第五行汇编代码的作用是显示以“ ”为结尾的字符串。最后一行用于结束程序。）

```
-e cs:200
0AF5:0200  41.41  B7.42  96.43  C6.44  46.24
-a100
0AF5:0100 mov ax,cs
0AF5:0102 mov ds,ax
0AF5:0104 mov dx,200
0AF5:0107 mov ah,9
0AF5:0109 int 21
0AF5:010B int 20
0AF5:010D
-
```

## G命令的使用

G命令作用：执行汇编指令。

G命令的使用方法是：G [=起始地址] [断点地址]，意思是从起始地址开始执行到断点地址。如果不设置断点，则程序一直运行到中止指令才停止。

在设置完示例九的内存数据并且输入完示例九的程序后运行这些汇编代码。在DEBUG中执行命令G=100，观看运行结果。

```
-a100
0AF5:0100 mov ax,cs
0AF5:0102 mov ds,ax
0AF5:0104 mov dx,200
0AF5:0107 mov ah,9
0AF5:0109 int 21
0AF5:010B int 20
0AF5:010D
-g=100
ABCD
Program terminated normally
-
```

汇编程序运行后在屏幕上显示出“ABCD”四个字符。

接下来在DEBUG中执行G=100 10B，意思是从地址CS:100开始，一直运行到CS:10B停止。观看运行结果。

命令执行后，不但显示出字符串“ABCD”，而且列出当前寄存器和标志位的值。



```

-a100
0AF5:0100 mov ax,cs
0AF5:0102 mov ds,ax
0AF5:0104 mov ah,9
0AF5:0106 mov dx,200
0AF5:0109 int 21
0AF5:010B int 20
0AF5:010D
-g=100
ABCD
Program terminated normally
-g=100 10b
ABCD
AX=0924 BX=0000 CX=0000 DX=0200 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0AF5 ES=0AF5 SS=0AF5 CS=0AF5 IP=010B  NU UP EI PL NZ NA PO NC
0AF5:010B C020          INT     https://blog.csdn.net/weixin_43809545
-

```

```

-a100
0AF5:0100 mov ax,cs
0AF5:0102 mov ds,ax
0AF5:0104 mov ah,9
0AF5:0106 mov dx,200
0AF5:0109 int 21
0AF5:010B int 20
0AF5:010D
-g=100
ABCD
Program terminated normally
-g=100 10b
ABCD
AX=0924 BX=0000 CX=0000 DX=0200 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0AF5 ES=0AF5 SS=0AF5 CS=0AF5 IP=010B  NU UP EI PL NZ NA PO NC
0AF5:010B C020          INT     https://blog.csdn.net/weixin_43809545
-

```

g命令在loop中的使用：

使用loop循环时，如果要跳出循环执行后面的代码，可以先用u命令查看代码的偏移地址，在用g命令跳转到此偏移到此代码处，如下：

```

AX=007B BX=0000 CX=00EC DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=0B30 ES=0B30 SS=0B40 CS=0B40 IP=0009  NU UP EI PL NZ NA PE NC
0B40:0009 E2FB          LOOP    0006
-u
0B40:0009 E2FB          LOOP    0006
0B40:000B B8004C        MOV     AX,4C00
0B40:000E CD21        INT     21
0B40:0010 E38B        JCXZ    FF9D
0B40:0012 87BE228B     XCHG    DI,[BP+8B22]
0B40:0016 97          XCHG    DI,AX
0B40:0017 C0          DB      C0
0B40:0018 228986FA        AND     CL,[BX+DI+FA86]
0B40:001C FE8996FC        DEC     BYTE PTR [BX+DI+FC96]
0B40:0020 FEC4        INC     AH
0B40:0022 9E          SAHF
0B40:0023 FA          CLI
0B40:0024 FE268A47     JMP     [478A]
0B40:0028 0C2A        OR      AL,2A
-g 000b
AX=7164 BX=0000 CX=0000 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=0B30 ES=0B30 SS=0B40 CS=0B40 IP=000B  NU UP EI PL NZ NA PO NC
0B40:000B B8004C        MOV     AX,4C00
https://blog.csdn.net/weixin_43809545

```

用u命令可以看到循环的命令位于0b40:000b，所以要跳出循环就用 g 000b 即可。

## U命令的使用

U命令作用：对机器代码反汇编显示。

U命令的使用方法是：U [范围]。如果范围参数只输入了起始地址，则只对20H个字节的机器代码反汇编。执行命令U100，观看反汇编结果。

```

-g=100
ABCD
Program terminated normally
-g=100 10b
ABCD
AX=0924 BX=0000 CX=0000 DX=0200 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0AF5 ES=0AF5 SS=0AF5 CS=0AF5 IP=010B  NU UP EI PL NZ NA PO NC
0AF5:010B CD20      INT     20
-u100
0AF5:0100 8CC8      MOV     AX,CS
0AF5:0102 8ED8      MOV     DS,AX
0AF5:0104 B409      MOV     AH,09
0AF5:0106 BA0002    MOV     DX,0200
0AF5:0109 CD21      INT     21
0AF5:010B CD20      INT     20
0AF5:010D 3F        AAS
0AF5:010E 7527      JNZ     0137
0AF5:0110 80FC00    CMP     AH,00
0AF5:0113 7420      JZ      0135
0AF5:0115 80FC01    CMP     AH,01
0AF5:0118 7522      JNZ     013C
0AF5:011A 3ACE      CMP     CL,DH
0AF5:011C 3400      XOR     AL,00
0AF5:011E E40A      IN      http://blog.csdn.net/weixin_43809545

```

执行命令U100 10B，观看反汇编结果。该命令的作用是对从100到10B的机器代码进行反汇编。

```

-u100 10c
0AF5:0100 8CC8      MOV     AX,CS
0AF5:0102 8ED8      MOV     DS,AX
0AF5:0104 B409      MOV     AH,09
0AF5:0106 BA0002    MOV     DX,0200
0AF5:0109 CD21      INT     21
0AF5:010B CD20      INT     20

```

## N命令的使用

N命令作用：设置文件名，为将刚才编写的汇编程序存盘做准备。

以下的DEBUG命令序列作用将刚才的汇编程序存为磁盘的COM可执行程序。

D200 20F

U100 10C

N E:\FIRST.COM

RCX

:110

W

第一和第二条命令的作用是检查一下刚才编写的汇编指令。第三条命令的作用是设置存盘文件名为E:\FIRST.COM，第四条命令的作用是设置存盘文件大小为110H个字节。最后一条命令是将文件存盘。

```

-d200 20f
0AF5:0200 41 42 43 44 24 00 02 0A-E4 75 05 3A 45 FF 74 05  ABCD$.u.:E.t.
-u100 10c
0AF5:0100 8CC8      MOV     AX,CS
0AF5:0102 8ED8      MOV     DS,AX
0AF5:0104 B409      MOV     AH,09
0AF5:0106 BA0002    MOV     DX,0200
0AF5:0109 CD21      INT     21
0AF5:010B CD20      INT     20
-n e:\first.com
-r cx
CX 0000
:110
-w
Writing 00110 bytes

```

文件存盘后执行E:\FIRST.COM，观看存盘的可执行文件的运行效果。

```
E:\>first
ABCD
E:\>
```

## W命令的使用

W命令作用：将文件或者特定扇区写入磁盘。

在示例“N命令的使用”中已经实验了如何使用W命令将文件存盘。

在没有很好地掌握汇编语言和磁盘文件系统前，暂时不要使用W命令写磁盘扇区，否则很容易损坏磁盘文件，甚至破坏整个磁盘的文件系统。

## L命令的使用

L命令作用：从磁盘中将文件或扇区内容读入内存。

将文件调入内存必须先用DEBUG的N命令设定文件名。以下例子是将E:\FIRST.COM读入内存。

N FIRST.COM

L

观看调入程序的汇编代码可以使用DEBUG的U命令，用U100观看调入的COM文件。

```
-n e:\first.com
-l
-u100
0B47:0100 8CC8      MOV     AX,CS
0B47:0102 8ED8      MOV     DS,AX
0B47:0104 B409      MOV     AH,09
0B47:0106 BA0002 MOV     DX,0200
0B47:0109 CD21      INT     21
0B47:010B CD20      INT     20
```

读取磁盘扇区的方式是：L [内存地址] [磁盘驱动器号] [起始扇区] [扇区数]。“内存地址”指定要在其中加载文件或扇区内容的内存位置，如果不指定“内存地址”的话，DEBUG将使用CS寄存器中的当前地址。“磁盘驱动器号”指定包含读取指定扇区的磁盘的驱动器，该值是数值型：0=A，1=B，2=C等。“起始扇区”指定要加载其内容的第一个扇区的十六进制数。“扇区数”指定要加载其内容的连续扇区的十六进制数。

只有要加载特定扇区的内容而不是加载文件时，才能使用[磁盘驱动器号] [起始扇区] [扇区数]参数。

例如：要将C盘第一扇区读取到内存DS:300的位置，相应的DEBUG命令为L DS:300 2 1。但是由于Windows操作系统对文件系统的保护，这条命令可能会被操作系统禁止运行。



## T命令的使用

T命令作用：执行汇编程序，单步跟踪。

T命令的使用方式是T [=地址] [指令数]。如果忽略“地址”的话，T命令从CS:IP处开始运行。“指令数”是要单步执行的指令的数量。



以下示例对E:\FIRST.COM进行单步跟踪。

N E:\FIRST.COM

L

U100 10B

R

T=100

T

```
-n e:\first.com
-l
-u 100 10b
0B47:0100 8CC8      MOV     AX,CS
0B47:0102 8ED8      MOV     DS,AX
0B47:0104 B409      MOV     AH,09
0B47:0106 BA0002    MOV     DX,0200
0B47:0109 CD21      INT     21
0B47:010B CD20      INT     20
-r
AX=0000 BX=0000 CX=0110 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=0B47 ES=0B47 SS=0B47 CS=0B47 IP=0100  NU UP EI PL NZ NA PO NC
0B47:0100 8CC8      MOV     AX,CS
-t=100
AX=0B47 BX=0000 CX=0110 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=0B47 ES=0B47 SS=0B47 CS=0B47 IP=0102  NU UP EI PL NZ NA PO NC
0B47:0102 8ED8      MOV     DS,AX
-t
AX=0B47 BX=0000 CX=0110 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=0B47 ES=0B47 SS=0B47 CS=0B47 IP=0104  NU UP EI PL NZ NA PO NC
0B47:0104 B409      MOV     AH,09
-
https://blog.csdn.net/weixin_43809545
```

第一、二条命令是装入文件，第三条命令是列出程序反汇编代码，第四条命令是显示当前寄存器值，第五条命令是从CS:100处开始单步跟踪，第六条命令是继续跟踪后续的指令。

## P命令的使用

P命令作用：执行汇编程序，单步跟踪。与T命令不同的是：P命令不会跟踪进入子程序或软中断。

P命令的使用方式与T命令的使用方式完全相同。

```
AX=0B47 BX=0000 CX=0110 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=0B47 ES=0B47 SS=0B47 CS=0B47 IP=0102  NU UP EI PL NZ NA PO NC
0B47:0102 8ED8      MOV     DS,AX
-t
AX=0B47 BX=0000 CX=0110 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=0B47 ES=0B47 SS=0B47 CS=0B47 IP=0104  NU UP EI PL NZ NA PO NC
0B47:0104 B409      MOV     AH,09
-p
AX=0B47 BX=0000 CX=0110 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=0B47 ES=0B47 SS=0B47 CS=0B47 IP=0106  NU UP EI PL NZ NA PO NC
0B47:0106 BA0002    MOV     DX,0200
-p
AX=0B47 BX=0000 CX=0110 DX=0200 SP=FFFE BP=0000 SI=0000 DI=0000
DS=0B47 ES=0B47 SS=0B47 CS=0B47 IP=0109  NU UP EI PL NZ NA PO NC
0B47:0109 CD21      INT     21
-p
ABCD
AX=0B24 BX=0000 CX=0110 DX=0200 SP=FFFE BP=0000 SI=0000 DI=0000
DS=0B47 ES=0B47 SS=0B47 CS=0B47 IP=010B  NU UP EI PL NZ NA PO NC
0B47:010B CD20      INT     20
-
https://blog.csdn.net/weixin_43809545
```

p命令还可以用于结束本次循环，进入下一次循环。

## I命令的使用

I命令作用：从计算机输入端口读取数据并显示。

I命令的用法是I [端口地址]。例如从3F8号端口读取数据并显示的命令为：I 3F8。这里不对该命令做解释。

## O命令的使用

O命令作用：向计算机输出端口送出数据。

O命令的用法是O [端口地址] [字节值]。例如向278号端口发出数据20H的命令为：O 278 20。这里不对该命令做解释。

## Q命令的使用

Q命令的作用是退出DEBUG，回到DOS状态。

来源：<https://www.cnblogs.com/tiger2soft/p/5094917.html>