

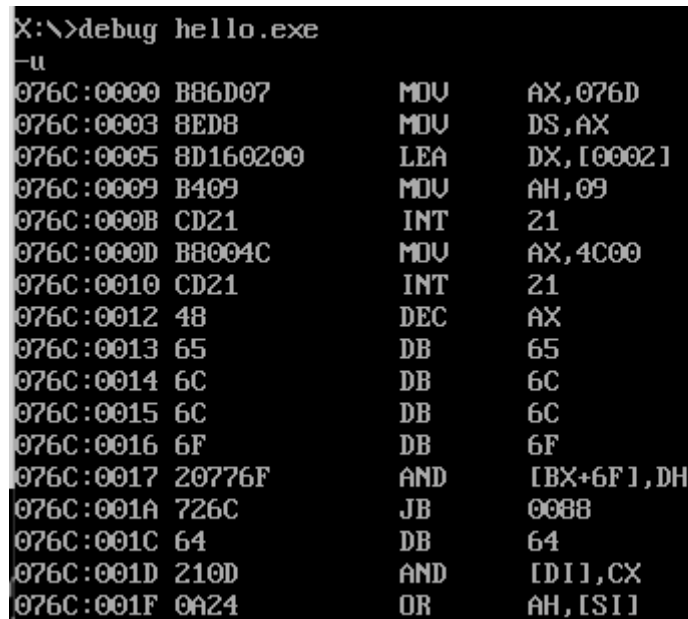
helloworld

一、正常输出

1. 源码一

```
.model small
.data
Hello      DB 'Hello world!',0dh,0ah,'$'
.code
START:     MOV AX,@DATA
           MOV DS,AX
           LEA DX,Hello
           MOV AH,9
           INT 21H

           MOV AX,4C00H
           INT 21h
END        START
```



```
X:\>debug hello.exe
-u
076C:0000 B86D07      MOV     AX,076D
076C:0003 8ED8             MOV     DS,AX
076C:0005 8D160200         LEA     DX,[0002]
076C:0009 B409             MOV     AH,09
076C:000B CD21             INT     21
076C:000D B8004C      MOV     AX,4C00
076C:0010 CD21             INT     21
076C:0012 48             DEC     AX
076C:0013 65             DB      65
076C:0014 6C             DB      6C
076C:0015 6C             DB      6C
076C:0016 6F             DB      6F
076C:0017 20776F      AND     [BX+6F],DH
076C:001A 726C             JB      0088
076C:001C 64             DB      64
076C:001D 210D      AND     [DI],CX
076C:001F 0A24             OR      AH,[SI]
```

可以看到，第三条指令，也就是LEA（装载有效地址）向DX中装载的是0002中的内容，也就是hello。

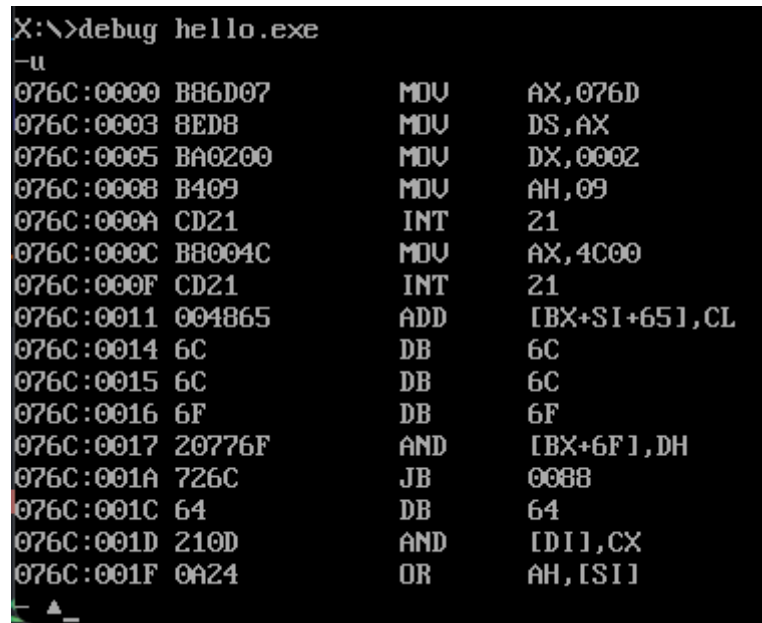
2. 源码二

```

.model small
.data
Hello      DB 'Hello world!',0dh,0ah,'$'
.code
START:      MOV AX,@DATA
            MOV DS,AX
            MOV DX,offset Hello
            MOV AH,9
            INT 21H

            MOV AX,4C00H
            INT 21h
END         START

```



```

X:\>debug hello.exe
-u
076C:0000 B86D07      MOV     AX,076D
076C:0003 8ED8          MOV     DS,AX
076C:0005 BA0200      MOV     DX,0002
076C:0008 B409          MOV     AH,09
076C:000A CD21          INT     21
076C:000C B8004C      MOV     AX,4C00
076C:000F CD21          INT     21
076C:0011 004865      ADD     [BX+SI+65],CL
076C:0014 6C          DB     6C
076C:0015 6C          DB     6C
076C:0016 6F          DB     6F
076C:0017 20776F      AND     [BX+6F],DH
076C:001A 726C          JB     0088
076C:001C 64          DB     64
076C:001D 210D          AND     [DI],CX
076C:001F 0A24          OR     AH,[SI]

```

可以看到，第三条地址MOV是将hello的偏移量0002移动到DX中。

二、另类执行

1. 在汇编阶段选择生成列表文件 (*.lst) --可直接用写字板打开（显示地址、内容、源码等 对应关系）

```
TEST.asm U  HELLO.asm U  HELLO.LST U x
HELLO.LST
1  Microsoft (R) Macro Assembler Version 5.00 9/22/24 15:57:54
2  Page 1-1
3
4
5      1      .model small
6      2 0000      .data
7      3 0000 48 65 6C 6C 6F 20 77 Hello      DB 'Hello world!','0dh,0ah','$'
8
9      4      6F 72 6C 64 21 0D 0A
10     5      24
11     6 0000      .code
12     7 0000 B8 ---- R      START:      MOV AX,@DATA
13     8 0003 8E D8      MOV DS,AX
14     9 0005 BA 0000 R      MOV DX,offset Hello
15    10 0008 B4 09      MOV AH,9
16    11 000A CD 21      INT 21H
17    12
18    13 000C B8 4C00      MOV AX,4C00H
19    14 000F CD 21      INT 21h
20    15 0011      END      START
21  Microsoft (R) Macro Assembler Version 5.00 9/22/24 15:57:54
22  Symbols-1
23
24
25 Segments and Groups:
26
27      Name      Length  Align  Combine Class
28
29  DGROUP . . . . . GROUP
```

2. 在汇编阶段选择生成交叉引用文件 (*.crf) --不能直接用写字板打开，用 CREF 工具转成 *.ref 文件后可用写字板浏览。（显示符号的定义及引用位置）

```
TEST.asm U  HELLO.asm U  HELLO.REF U x
HELLO.REF
1  Microsoft Cross-Reference Version 5.00 Sun Sep 22 16:01:21 2024
2
3
4
5  Symbol Cross-Reference      (# definition, + modification)  Cref-1
6
7  CODE . . . . . 6
8
9  DATA . . . . . 2
10 DGROUP . . . . . 7
11
12 HELLO. . . . . 3# 9
13
14 START. . . . . 7# 15
15
16 _DATA. . . . . 2#
17 _TEXT. . . . . 6#
18
19
20 7 Symbols
21
```

3. 直接写内存方式执行代码：

1. A) 写数据“Hello\$”对应的 ASCII 码 48 65 6c 6c 6f 24 写入内存 Debug 下用-e 076a: 0 回车一次写入（用空格自动分开了）-----相当于 DS: 076A
2. B) 写代码的机器码 b8 6b 07 be d8 ba 02 00 b4 09 cd 21 b8 00 4c cd 21（17 个字节）写入内存 Debug 下用-e 076b: 0 回车 一次写入（用空格自动分开了）-----相当于 CS: 076B

3. C) 修改寄存器及执行

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
DS=075C ES=075C SS=076B CS=076C IP=0000  NU UP EI PL NZ NA PO NC
076C:0000 216D07      AND     [DI+07],BP      DS:0007=ADFF
-r cs
CS 076C
:076b
-r ds
DS 075C
:076a
-r
AX=FFFF BX=0000 CX=0021 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075C SS=076B CS=076B IP=0000  NU UP EI PL NZ NA PO NC
076B:0000 B86B07      MOV     AX,076B
-r ip
IP 0000
:0
-r
AX=FFFF BX=0000 CX=0021 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075C SS=076B CS=076B IP=0000  NU UP EI PL NZ NA PO NC
076B:0000 B86B07      MOV     AX,076B
-g
Hello
Program terminated normally
-q
X:\>_
```