源码一：使用 LEA 指令来将字符串指令打到寄存器当中。

```
.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
```

用 debug -u 反汇编查看寄存器状态

```
D:\>debug test.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]
```

源码二：使用 offset 伪指令来得到字符串的地址。

```
.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
```

```
D:\>DEBUG TEST.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]
-
```

两种代码在执行的效果都是一样的，都是计算得到 hello world 的地址并将其打入寄存器 DX 当中，LEA 指令会直接计算变量或标签的地址并打到寄存器当中，offset 指令会返回变量的内存地址，然后再赋值给寄存器。相比之下 offset 指令会比 LEA 指令要多进行一次赋值的计算。

直接用 debug 写内存方式实现 Hello world:
在 Debug 下用-e 076a:0
输入 48 65 6c 6c 6f 24 来将 Hello$字符串打入内存中。

```
D:\>debug
-e 076a:0
076A:0000  00.48   00.65   00.6c   00.6c   00.6f   00.24
```

再用-e 076b:0

输入 b8 6b 07 be d8 ba 02 00 b4 09 cd 21 b8 00 4c cd 21

```
-e 076b:0
076B:0000  00.b8   00.6b   00.07   00.be   00.d8   00.ba   00.02   00.00
076B:0008  00.b4   00.09   00.cd   00.21   00.b8   00.00   00.4c   00.cd
076B:0010  00.21
```

修改寄存器的状态来执行命令。

```
DOSBox 0.74-3, Cpu speed:    3000 cycles, Frameskip  0, Program:   DOSBOX

-r
AX=0000  BX=0000  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=0740  ES=0740  SS=0740  CS=0740  IP=0100   NV UP EI PL NZ NA PO NC
0740:0100 0000          ADD     [BX+SI],AL                      DS:0000=CD
-r cs
CS 0740
:076b
-r ds
DS 0740
:076a
-r
AX=0000  BX=0000  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=076A  ES=0740  SS=0740  CS=076B  IP=0100   NV UP EI PL NZ NA PO NC
076B:0100 0000          ADD     [BX+SI],AL                      DS:0000=48
-r ip
IP 0100
:0
-r
AX=0000  BX=0000  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=076A  ES=0740  SS=0740  CS=076B  IP=0000   NV UP EI PL NZ NA PO NC
076B:0000 B86B07        MOV     AX,076B
-g
Hello
D:\>
```