



# hello汇编语言运行文档

## 传统方式

### 1.创建hello.asm文件，代码如下：

```
STKSEG SEGMENT STACK
DW 32 DUP(0)
STKSEG ENDS

DATASEG SEGMENT
    MSG DB "Hello$"
DATASEG ENDS

CODESEG SEGMENT
    ASSUME CS:CODESEG,DS:DATASEG
MAIN PROC FAR
    MOV AX,DATASEG
    MOV DS,AX
    MOV AH,9
    LEA DX,MSG
    INT 21H
    MOV AX,4C00H
    INT 21H
MAIN ENDP
CODESEG ENDS
    END MAIN
```

## 2.汇编及链接

1. 打开DOSBox.exe，输入masm hello.asm，并设置生成的相关文件名

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pro...
HAVE FUN!
The DOSBox Team http://www.dosbox.com

Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>MOUNT C E:\program\DOSBox\masm
Drive C is mounted as local directory E:\program\DOSBox\masm\

Z:\>C:

C:\>masm hello.asm
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Object filename [hello.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:

50122 + 463283 Bytes symbol space free

0 Warning Errors
0 Severe Errors

C:\>S
```

2. 输入hello, 即可运行hello.exe

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pro...
The DOSBox Team http://www.dosbox.com

Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>MOUNT C E:\program\DOSBox\masm
Drive C is mounted as local directory E:\program\DOSBox\masm\

Z:\>C:

C:\>masm hello.asm
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Object filename [hello.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:

50122 + 463283 Bytes symbol space free

0 Warning Errors
0 Severe Errors

C:\>hello
Hello World
C:\>S
```

3. 输入debug hello.exe, 输入-u, 可查看反汇编详情

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pro...
Cross-reference [NUL.CRF]:

50122 + 463283 Bytes symbol space free

0 Warning Errors
0 Severe Errors

C:\>hello
Hello World
C:\>debug hello.exe
-u
076F:0000 B86E07      MOV     AX,076E
076F:0003 8ED8          MOV     DS,AX
076F:0005 B409          MOV     AH,09
076F:0007 BA0000      MOV     DX,0000
076F:000A CD21          INT     21
076F:000C B8004C      MOV     AX,4C00
076F:000F CD21          INT     21
076F:0011 02263947      ADD     AH,[4739]
076F:0015 087506      OR      [DI+06],DH
076F:0018 26            ES:
076F:0019 39570A      CMP     [BX+0A],DX
076F:001C 7403          JZ      0021
076F:001E E8BF8E      CALL   8EE0
-S_
```

## 另类执行方式

### 1.在上述汇编阶段设置.lst文件名

可以查看地址、内容、源码等的对应关系（汇编时.lst文件名默认为null，即不生成.lst文件）

### 2.查看寄存器地址

在debug时输入-r，可看到所有寄存器的地址

且输入-r\*(寄存器名)，回车后可修改寄存器地址

```
-r
AX=FFFF BX=0000 CX=0061 DX=0000 SP=0040 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=076A CS=076F IP=0000  NU UP EI PL NZ NA PO NC
076F:0000 B86E07      MOV     AX,076E
-r CS
CS 076F
:076b
-r
AX=FFFF BX=0000 CX=0061 DX=0000 SP=0040 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=076A CS=076B IP=0000  NU UP EI PL NZ NA PO NC
076B:0000 0000      ADD     [BX+SI],AL      DS:0000=CD
```

### 3.直接写内存方式执行代码

#### 1. 写代码的机器码

将 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（相当于写入 CS : 076B

#### 2. 写数据

将"Hello\$"对应的 ASCII 码 48 65 6c 6c 6f 24 写入内存  
与上一步相同的方法写入debug -e 076a: 0

```
-r
AX=FFFF BX=0000 CX=0061 DX=0000 SP=0040 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=076A CS=076F IP=0000  NU UP EI PL NZ NA PO NC
076F:0000 B86E07          MOV     AX,076E
-r CS
CS 076F
:076b
-r
AX=FFFF BX=0000 CX=0061 DX=0000 SP=0040 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=076A CS=076B IP=0000  NU UP EI PL NZ NA PO NC
076B:0000 0000          ADD     [BX+SI],AL          DS:0000=CD
-r ds
DS 075A
:076a
-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

-e 076a:0
076A:0000 00.48 00.65 00.6c 00.6c 00.6f 00.24
```

//该图为修改CS、DS寄存器后，写入机器码及数据的过程

#### 3. 执行

输入-g

```
-g
Hello
Program terminated normally
```