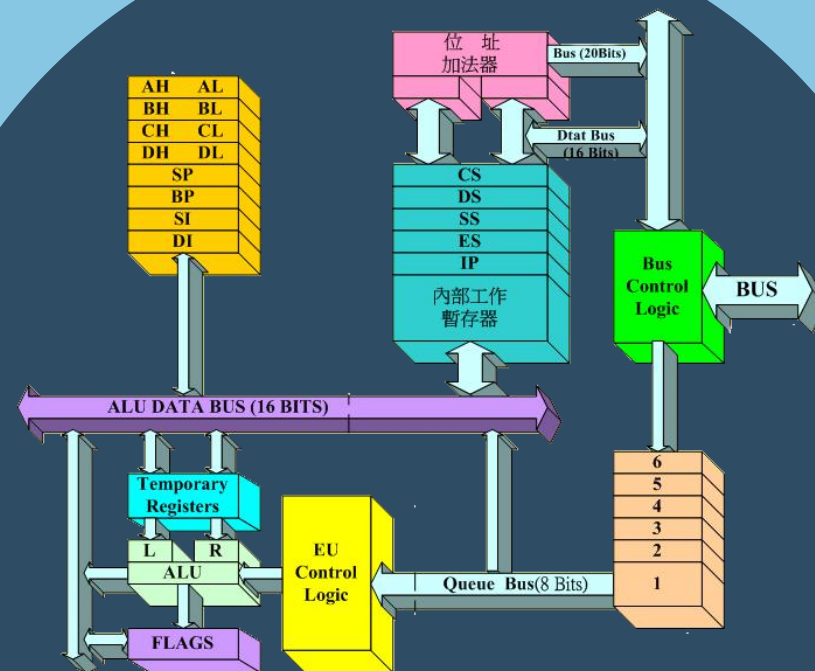


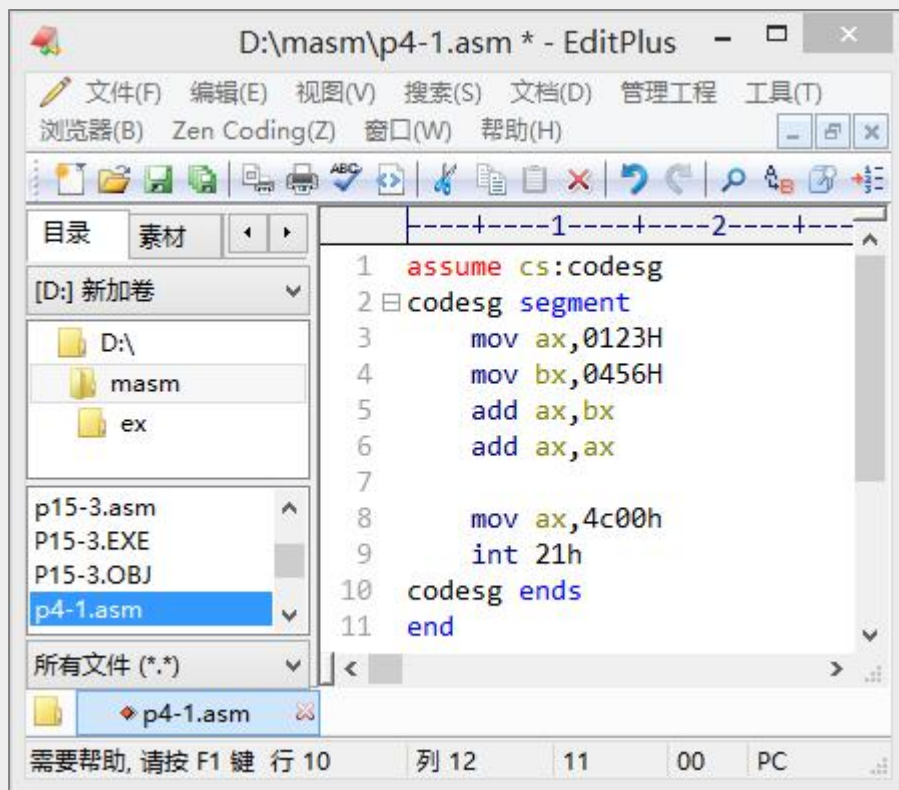
运行及跟踪

贺利坚 主讲



汇编语言程序设计
Assembly Language

回顾



```
C:\>masm p4-1.asm
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.

Object filename [p4-1.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:

51798 + 464746 Bytes symbol space free

0 Warning Errors
0 Severe Errors
```

```
C:\>link p4-1

Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.

Run File [P4-1.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment
```

C:\>p4-1

看看这个环节的门道。

源文件
.asm



目标文件
.obj

目标文件
.obj



可执行文件
.exe

p4-1.asm	2017-2-5 22:20	ASM 文件	1 KB
P4-1.EXE	2017-2-6 15:54	应用程序	1 KB
P4-1.OBJ	2017-2-5 22:20	OBJ 文件	1 KB

... B82301 BB5604 01DB ...



用Debug装载程序

有效代码共
15(0FH)字节

DS=075A

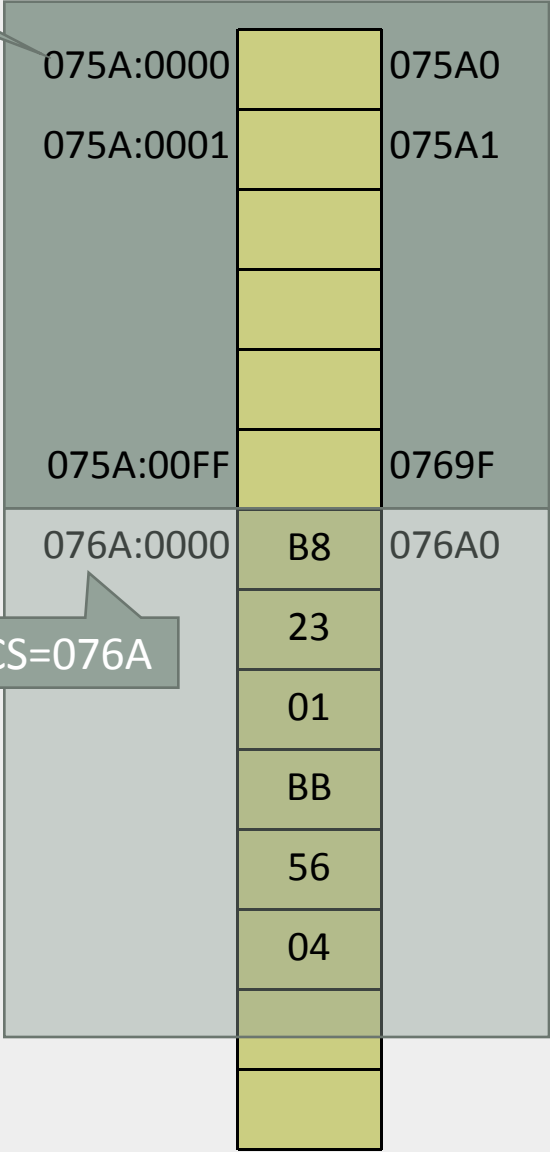
一共256(100H)字节的程序段
前缀(PSP)，作为数据区

```
1  assume cs:codesg
2  codesg segment
3      mov ax,0123H
4      mov bx,0456H
5      add ax,bx
6      add ax,ax
7
8      mov ax,4c00h
9      int 21h
10 codesg ends
11 end
```

```
C:\>debug p4-1.exe
-r
AX=FFFF BX=0000 CX=000F DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=0000  NV UP EI PL NZ NA PO NC
076A:0000 B82301      MOV     AX,0123
CS=076A
```

程序被装入内存的什么地方？

```
-u
076A:0000 B82301      MOV     AX,0123
076A:0003 BB5604      MOV     BX,0456
076A:0006 03C3       ADD     AX,BX
076A:0008 03C0       ADD     AX,AX
076A:000A B8004C      MOV     AX,4C00
076A:000D CD21       INT     21
076A:000F 7CF2       JL      0003
076A:0011 3D7400      CMP     AX,0074
076A:0014 7ED4       JLE     FFEA
076A:0016 FBEF       JMP     0003
```



小结

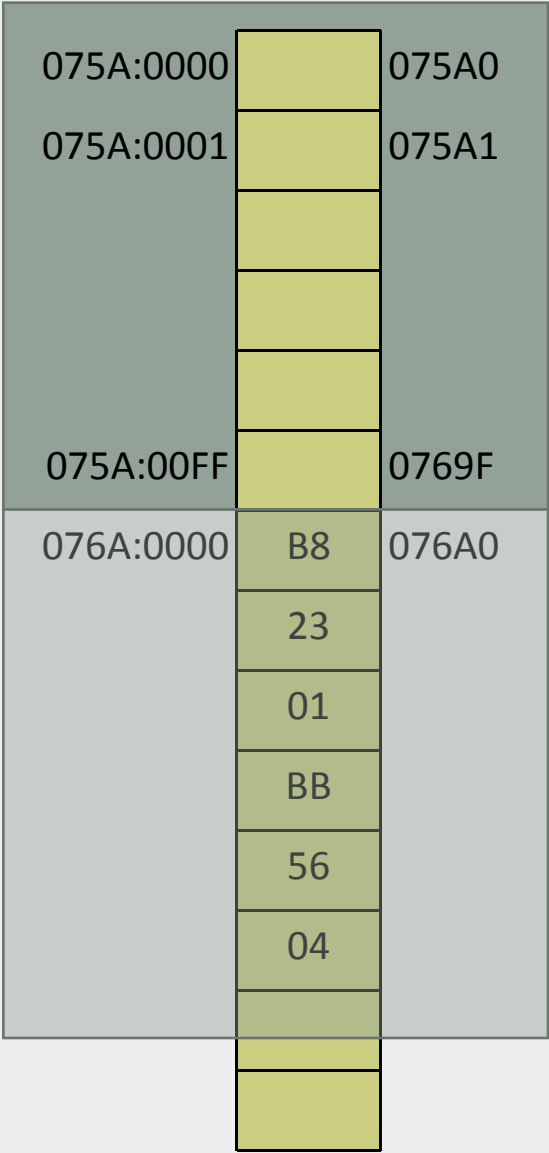
- ☞ 程序加载后，DS中存放着程序所在内存区的段地址，这个内存区的偏移地址为0，则程序所在的内存区的地址为：DS:0。
- ☞ 这个内存区的前256个字节存PSP，DOS用来和程序进行通信。
- ☞ 从256字节处向后的空间存放的是程序，CS的值为DS+10H。
- ☞ 程序加载后，CX中存放代码的长度（字节）。

用Debug单步执行程序

```
1  assume cs:codesg
2  codesg segment
3      mov ax,0123H
4      mov bx,0456H
5      add ax,bx
6      add ax,ax
7
8      mov ax,4c00h
9      int 21h
10 codesg ends
11 end
```

```
C:\>debug p4-1.exe
-r
AX=FFFF BX=0000 CX=000F DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=0000  NU UP EI PL NZ NA PO NC
076A:0000 B82301      MOV     AX,0123
-t
AX=0123 BX=0000 CX=000F DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=0003  NU UP EI PL NZ AC PO NC
076A:0003 BB5604      MOV     BX,0456
-t
AX=0123 BX=0456 CX=000F DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=0006  NU UP EI PL NZ AC PO NC
076A:0006 03C3      ADD     AX,BX
-t
AX=0579 BX=0456 CX=000F DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=0008  NU UP EI PL NZ NA PO NC
076A:0008 03C0      ADD     AX,AX
-t
AX=0AF2 BX=0456 CX=000F DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=000A  NU UP EI PL NZ AC PO NC
076A:000A B8004C      MOV     AX,4C00
-t
AX=4C00 BX=0456 CX=000F DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=000D  NU UP EI PL NZ AC PO NC
076A:000D CD21      INT     21
-t
AX=4C00 BX=0456 CX=000F DX=0000 SP=FFFA BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=F000 IP=14A0  NU UP DI PL NZ AC PO NC
F000:14A0 FB      STI
```

一共256(100H)字节的程序段前缀(PSP)，作为数据区



其他方式执行

```
1  assume cs:codesg
2  codesg segment
3      mov ax,0123H
4      mov bx,0456H
5      add ax,bx
6      add ax,ax
7
8      mov ax,4c00h
9      int 21h
10 codesg ends
11 end
```

继续命令P(Proceed)：类似T命令，逐条执行指令、显示结果。但遇子程序、中断等时，直接执行，然后显示结果。

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。

```
C:\>debug p4-1.exe
-r
AX=FFFF BX=0000 CX=000F DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=0000  NU UP EI PL NZ NA PO NC
076A:0000 B82301      MOV     AX,0123
-p
AX=0123 BX=0000 CX=000F DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=0003  NU UP EI PL NZ NA PO NC
076A:0003 B85604      MOV     BX,0456
-p
AX=0123 BX=0456 CX=000F DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=0006  NU UP EI PL NZ NA PO NC
076A:0006 03C3      ADD     AX,BX
-p
AX=0579 BX=0456 CX=000F DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=0008  NU UP EI PL NZ NA PO NC
076A:0008 03C0      ADD     AX,AX
-p
AX=0AF2 BX=0456 CX=000F DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=000A  NU UP EI PL NZ AC PO NC
076A:000A B8004C      MOV     AX,4C00
-p
AX=4C00 BX=0456 CX=000F DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=000D  NU UP EI PL NZ AC PO NC
076A:000D CD21      INT     21
-p
Program terminated normally
C:\>debug p4-1.exe
-r
AX=FFFF BX=0000 CX=000F DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=0000  NU UP EI PL NZ NA PO NC
076A:0000 B82301      MOV     AX,0123
-g
Program terminated normally
```

一共256(100H)字节的程序段前缀(PSP)，作为数据区

075A:0000		075A0
075A:0001		075A1
075A:00FF		0769F
076A:0000	B8	076A0
	23	
	01	
	BB	
	56	
	04	

程序执行的不同方式


在DOS中执行


```
C:\>p4-1
C:\>
```

在Debug中执行

```
-t
AX=0123 BX=0000 CX=000F DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=0003  NU UP EI PL NZ AC PO NC
076A:0003 BB5604          MOV     BX,0456
-t
AX=0123 BX=0456 CX=000F DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=0006  NU UP EI PL NZ AC PO NC
076A:0006 03C3          ADD     AX,BX
-p
AX=4C00 BX=0456 CX=000F DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=000D  NU UP EI PL NZ AC PO NC
076A:000D CD21          INT     21
-p
Program terminated normally
-
C:\>debug p4-1.exe
-r
AX=FFFF BX=0000 CX=000F DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=0000  NU UP EI PL NZ NA PO NC
076A:0000 B82301          MOV     AX,0123
-g
Program terminated normally
```


程序执行的“常态”

 DOS启动后，计算机由“命令解释器”（程序 command.com ）控制

 运行可执行程序时，command将程序加载入内存，设置CPU的CS:IP指向程序的第一条指令（即程序的入口），使程序得以运行。

 程序运行结束后，返回到“命令解释器”，CPU继续运行command。

 程序执行处于开发周期的运行方式；

 运行Debug时，command程序加载Debug.exe，debug将程序加载入内存，程序运行结束后要返回到Debug中，使用Q命令退出Debug，将返回到command中。