

# 关于“段”的总结

贺利坚 主讲



汇编语言程序设计  
Assembly Language

# 各种段——

10000H	23
10001H	11
10002H	22
10003H	66

## 基础

物理地址 = 段地址 × 16 + 偏移地址

## 做法

- 编程时，可以根据需要将一组内存单元定义为一个段。
- 可以将起始地址为16的倍数，长度为N (  $N \leq 64K$  ) 的一组地址连续的内存单元，定义为一个段。
- 将一段内存定义为一个段，用一个段地址指示段，用偏移地址访问段内的单元——在程序中完全可以由程序员安排。

## 三种段

### 数据段

- 将段地址放在 DS 中
- 用 mov、add、sub 等访问内存单元的指令时，CPU 将我们定义的数据段中的内容当作数据段来访问；

### 代码段

- 将段地址放在 CS 中，将段中第一条指令的偏移地址放在 IP 中
- CPU 将执行我们定义的代码段中的指令；

### 栈段

- 将段地址放在 SS 中，将栈顶单元的偏移地址放在 SP 中
- CPU 在需要进行栈操作(push、pop)时，就将我们定义的栈段当作栈空间来用。

# 综合示例：按要求设置段并执行代码

10000H	23	数据段	mov bx, 1000H
10001H	11		mov ds, bx
10002H	22		mov bx, 1001H
10003H	66		mov ss, bx
...			mov sp, 10H
1000FH		栈段	mov ax, [0]
10010H			mov bx, [2]
...			push ax
1001FH			push bx
10020H			pop ax
...		代码段	pop bx
20000H			mov [0], ax
...			mov [2], bx

```
-rds
DS 1000
:1000
-rss
SS 1001
:1001
-rsp
SP 0010
:0010
-rcs
CS 2000
:2000
-rip
IP 0000
:0000
-e ds:0 23 11 22 66
-r
AX=0001 BX=0000 CX=0000 DX=0000 SP=0010 BP=0000 SI=0000 DI=0000
DS=1000 ES=073F SS=1001 CS=2000 IP=0000  NU UP EI PL NZ NA PO NC
2000:0000 0000      ADD      [BX+SI],AL      DS:0000=23
-d ds:0 f
1000:0000  23 11 22 66 00 00 00 00-00 00 00 00 00 00 00 00  #."f.....

-a CS:0000
2000:0000 mov bx, 1000
2000:0003 mov ds, bx
2000:0005 mov bx, 1001
2000:0008 mov ss, bx
2000:000A mov sp, 10
2000:000D mov ax, [0]
-t
AX=0001 BX=1000 CX=0000 DX=0000 SP=0010 BP=00
DS=1000 ES=073F SS=1001 CS=2000 IP=0003  NU U
2000:0003 8EDB      MOV      DS,BX
-t
-d ds:0 f
1000:0000  22 66 23 11 00 00 00 00-00 00 00 00 00 00 00 00  "f#.....
```

# 综合示例：三个段地址可以一样滴！

10000H	23	数据段	mov bx, 1000H
10001H	11		mov ds, bx
10002H	22		mov ss, bx
10003H	66		mov sp, 20H
...			
1000FH			mov ax, [0]
10010H		栈段	mov bx, [2]
...			push ax
1001FH			push bx
10020H		代码段	pop ax
...			pop bx
...			mov [0], ax
...			mov [2], bx

```
-rds
DS 1000
:1000
-rss
SS 1000
:1000
-rcs
CS 1000
:1000
-rsp
SP 0020
:0020
-rip
IP 001F
:0020
-e ds:0 23 11 22 66
-r
AX=6622 BX=1123 CX=0000 DX=0000 SP=0020 BP=0000 SI=0000 DI=0000
DS=1000 ES=073F SS=1000 CS=1000 IP=0020  NU UP EI PL NZ NA PO NC
1000:0020 0000          ADD     [BX+SI],AL          DS:1123=00
```