

王爽汇编第15章,直接定址表

1. 直接定址表
2. 描述了单位长度的标号
3. seg操作符

1. 直接定址表

程序中, code、a、b、start、s都是标号。这些标号仅仅表示了内存单元的地址

```
1  assume cs:code
2  code segment
3      a : db 1,2,3,4,5,6,7,8 ;在后面加有“:”的地址标号,只能在代码段中使用,不能在其他段中使用。
4      b : dw 0
5  start :mov si,offset a
6          mov bx,offset b
7          mov cx,8
8      s : mov al,cs:[si]
9          mov ah,0
10         add cs:[bx],ax
11         inc si
12         loop s
13         mov ax,4c00h
14         int 21h
15 code ends
16 end start
```

2. 描述了单位长度的标号

```
1  assume cs:code
2  code segment
3      a db 1,2,3,4,5,6,7,8 ;标号a、b后面没有“:”,因此它们是可以同时描述内存地址和单元长度的标号。
4                                     ;标号a,描述了地址code:0,和从这个地址开始,以后的内存单元都是字节单元
5      b dw 0 ;标号b描述了地址code:8,和从这个地址开始,以后的内存单元都是字单元。
6  start : mov si,0
7          mov cx,8
8      s :  mov al,a[si]
9          mov ah,0
10         add b,ax
11         inc si
12         loop s
13         mov ax,4c00h
14         int 21h
15 code ends
16 end start
```

使用**数据标号**来描述存储数据的单元的地址和长度。

```
1  assume cs:code,ds:data ;用伪指令assume将标号所在的段和一个段寄存器联系起来(编译器需要)
2  data segment
```

```

3      a db 1,2,3,4,5,6,7,8
4      b dw 0
5  data ends
6  code segment
7  start: mov ax,data
8          mov ds,ax ;真正确定ds寄存器
9          mov si,0
10         mov cx,8
11 s:      mov al,a[si] ;编译为: mov al,[si+0] 默认所访问单元的段地址在ds
12         mov ah,0
13         add b,ax ;编译为: add [8],ax
14         inc si
15         loop s
16         mov ax,4c00h
17         int 21h
18 code ends
19 end start

```

3. seg操作符

seg操作符，功能为取得某一标号的段地址

```

1  data segment
2      a db 1,2,3,4,5,6,7,8
3      b dw 0
4      c dw a, b ;等价于c dw offset a, offset b
5      ;数据标号c处存储的两个字型数据为标号a、b 的偏移地址
6  data ends
7
8  data segment
9      a db 1,2,3,4,5,6,7,8
10     b dw 0
11     c dd a,b ;等价于c dw offset a, seg a, offset b, seg b
12     ;数据标号c处存储的两个双字型数据为标号a的偏移地址和段地址、标号b 的偏移地址和段地址
13 data ends

```

建立一张表，表中依次存储字符“0”~“F”，我们可以通过数值0 ~ 15直接查找到对应的字符

```

1  assume cs:code
2
3  code segment
4  start:
5      mov al,0eh
6
7      call showbyte
8
9      mov ax,4c00h
10     int 21h
11
12 ;子程序:
13 ;用al传送要显示的数据
14
15 showbyte:
16     jmp short show
17
18     table db '0123456789ABCDEF' ;字符表
19
20 show: push bx

```

```

21      push es
22
23      mov ah,al
24      shr ah,1
25      shr ah,1
26      shr ah,1
27      shr ah,1      ;右移4位, ah中得到高4位的值
28      and al,00001111b      ;al中为低4位的值
29
30      mov bl,ah
31      mov bh,0
32      mov ah,table[bx]      ;用高4位的值作为相对于table的偏移, 取得对应的字符
33
34      mov bx,0b800h
35      mov es,bx
36      mov es:[160*12+40*2],ah
37
38      mov bl,al
39      mov bh,0
40      mov al,table[bx]      ;用低4位的值作为相对于table的偏移, 取得对应的字符
41
42      mov es:[160*12+40*2+2],al
43
44      pop es
45      pop bx
46      ret
47
48 code ends
49 end start

```

参考文章:

https://blog.csdn.net/qq_39654127/article/details/88698911 王爽《汇编语言》笔记 (详细)