

作业二

2005010130 侯杰

4.1 对于下列两条数据定义伪操作，给出汇编后内存中的存储情况：

```
(1) BYTE_VAR DB 'BYTE', 12, -12H, 3 DUP(0, ?, 2 DUP(1, 2), ?)
```

```
(2) WORD_VAR DW 5 DUP(0, 1, 2), ?, -5, 'BY', 'TE', 256H
```

答：下面图形表示内存，上方为低地址，下方为高地址
(1)

42
59
54
45
0C
EE
00
--
01
02
01
02
--
00
--
01
02
01
02
--
00
--
01
02
01
02
--

(2)

00
00
01
00
02
00
00
00
01
00
02
00
00
00
01
00
02
00
00
00
01
00
02
00
00
00
01
00
02
00
--
--
FB
FF
59
42
45
54
56
02

4.2 假设数据段中数据定义如下：

```
PARTNO  DW   ?
PNAME   DB   16  DUP (?)
COUNT  DD   ?
PLENTH  EQU  $-PARTNO
```

问 PLENTH 的值为多少？它表示什么意义？

答：PLENTH 的值为 $2*1+1*16+4*1=2+16+4=22d$ ，它表示以上数据定义所占据的内存空间大小（单位为字节）。

4.3 假设数据段中数据定义如下：

```
ORG 100H
A  DB   '34'  100H
B  DD   1      102H
X  LABEL BYTE 106H
Y  DW   'AB'  106H
```

执行下列程序段并填空：

```
LEA    DI, B           ; (DI) = 0102 H
MOV    AX, WORD PTR A  ; (AX) = 3433 H
MOV    BX, OFFSET X    ; (BX) = 0106 H
LEA    BP, Y           ; (BP) = 0106 H
MOV    CL, X           ; (CL) = 42 H
```

4.4 假设数据段中数据定义如下：

```
VAR     DW   '34'
VAR1    DB   100, 'ABCD'
VAR2    DD   1
COUNT  EQU  $-VAR1
X       DW   5  DUP (COUNT DUP (0))
Y       LABEL WORD
Z       DB   '123456'
V       DW   2, $-VAR
```

执行下面程序段并回答问题：

```
MOV    AX, COUNT       ; (AX) = 0009H
MOV    BX, Z-X          ; (BX) = 005AH
MOV    CX, V+2          ; (CX) = 006DH
MOV    DX, VAR          ; (DX) = 3334H
MOV    Y+3, 2
MOV    SI, Y+4          ; (SI) = 3600H
ADD    Z+5, 1
MOV    DI, WORD PTR Z+4 ; (DI) = 3700H
```

4.5 对于下面的数据定义, 各条 MOV 指令单独执行后, 有关寄存器的内容是什么?

```
FLDB      DB      ?  
TABLEA    DW      20 DUP(?)  
  
TABLEB    DB      'ABCD'
```

(1) MOV AX, TYPE FLDB (AX)=0001H

(2) MOV AX, TYPE TABLEA (AX)=0002H

(3) MOV CX, LENGTH TABLEA (CX)=0014H

(4) MOV DX, SIZE TABLEA (DX)=0028H

(5) MOV CX, LENGTH TABLEB (CX)=0001H

4.6 请把程序 example.asm 填写完整, 它的运行结果是什么?

```
; example.asm  
data1 segment  
string1      db      'thanks you'  
string2      db      'thanks you'  
mass1        db      'match!',13,10,'$'  
mass2        db      'no match!',13,10,'$'  
data1 ends
```

```
program segment  
main  proc far  
      assume cs:program, ds:data1  
begin:  
      push  ds  
      sub   ax,ax  
push  ax  
  
      mov   ax,data1  
      mov   ds,ax  
mov   es,ax  
      lea   si,string1  
      lea   di,string2  
      cld  
      mov   cx,10  
repe  cmpsb
```

```

            jz      match
            lea     dx,mass2

            jmp     short disp
match:
            lea     dx,mass1
disp:
            mov     ah,09
            int     21h
            ret
main endp
program ends
end begin

```

4.7 写一个完整的程序放在代码段 C_SEG 中，要求把数据段 D_SEG 中的 AUGEND 和附加段 E_SEG 中的 ADDEND 相加，并把结果存放在 D_SEG 中的 SUM 中。其中 AUGEND、ADDEND 和 SUM 均为双精度数，AUGEND 赋值为 99251，ADDEND 赋值为-15962。

答：程序如下（另附源程序文件 4_7.asm）

```

D_SEG segment
    AUGEND dd 99251
    SUM dd ?
D_SEG ends

E_SEG segment
    ADDEND dd -15962
E_SEG ends

C_SEG segment
    main proc far
        assume cs:C_SEG,ds:D_SEG,es:E_SEG
start:
    push ds
    sub ax,ax
    push ax
    mov ax,D_SEG
    mov ds,ax
    mov ax,E_SEG
    mov es,ax

```

```

        ;start calculation
        mov ax,word ptr AUGEND
        mov dx,word ptr AUGEND+2
        mov cx,word ptr ADDEND
        mov bx,word ptr ADDEND+2
        add ax,cx
        adc dx,bx
        mov word ptr SUM,ax
        mov word ptr SUM+2,dx

        ret
    main endp
C_SEG ends
    end start

```

4.8 编写一个完整的程序，要求把含有 23H,24H,25H,26H 四个字符数据的数
据区复制 20 次。

答：程序如下（另附源程序文件 **4_8.asm**）

```

data_seg segment
    SOURCE db 23h,24h,25h,26h
data_seg ends

extra_seg segment
    DEST db 80 dup (?)
    DEST_END label byte
extra_seg ends

code_seg segment
    main proc far
        assume cs:code_seg,ds:data_seg,es:extra_seg
    start:
        mov ax,data_seg
        mov ds,ax
        mov ax,extra_seg
        mov es,ax
        ;start copying
        cld
        mov si,0
        mov di,0
        mov cx,4
        lea si,SOURCE

```

```

        lea di,DEST
        rep movsb
again:
        mov cx,4
        lea si,SOURCE
        rep movsb
        cmp di,offset DEST+80
        jne again      ;if not finish, do it again
        ;exit
        mov ax,4c00h
        int 21h
        main endp
code_seg ends
        end start

```

5.1 不用串操作指令，把字符串 string1 的内容传送到字符串 string2，字符长度为 count。

答：程序如下（另附源程序文件 [5_1.asm](#)）

```

data_seg segment
        string1 db 'demo string'
        count equ $-string1
        string2 db count dup(?)
data_seg ends

code_seg segment
        main proc far
        assume cs:code_seg,ds:data_seg
start:
        mov ax,data_seg
        mov ds,ax
        ;start copying
        lea si,string1
        lea di,string2
        mov cx,count
copy:
        mov al,ds:[si]
        mov ds:[di],al
        inc si
        inc di
        dec cx
        jnz copy

```

```

        ;exit
        mov ax,4c00h
        int 21h
        main endp
code_seg ends
        end start

```

5.2 从键盘接收一个个位数 N，然后响铃 N 次（响铃的 ASCII 码为 07）。

答：程序如下（另附源程序文件 [5_2.asm](#)）

```

code_seg segment
    main proc far
        assume cs:code_seg
start:
    mov ah,1
    int 21h
    sub al,30h ;from ACSII to binary
    cmp al,10  ;see if the input number is <10
    jnb exit   ;if the input number >=10 then exit
    mov cl,al   ;count bell
    mov dl,07h
    mov ah,2
bell:
    int 21h
    dec cl
    jnz bell
exit:
    mov ax,4c00h
    int 21h
    main endp
code_seg ends
        end start

```

5.3 数组 A 和 B 分别包含 10 个互不相等的整数，把既在 A 中又在 B 中的整数存放于数组 C 中。

答：程序如下（另附源程序文件 [5_3.asm](#)）

```

data_seg segment
    arrayA db 1,3,4,6,9,12,22,40,50,57;demo array
    arrayB db 2,3,4,7,9,15,22,33,40,57;demo array
    arrayC db 10 dup(?)

```



```

data_seg ends

code_seg segment
    main proc far
        assume cs:code_seg,ds:data_seg
start:
    mov ax,data_seg
    mov ds,ax
    mov es,ax;IMPORTANT!!!

    cld
    mov bx,0;bx is used for the index of arrayC
    mov dx,0
    mov si,0
    mov dx,10;dx is used for the count for outer loop

inner:
    mov cx,10;cx is used for the count for inner loop
    lea di,arrayB
    mov al,arrayA[si]
    repne scasb
    jne NotFound
Found:
    mov arrayC[bx],al;append an element to arrayC
    inc bx
NotFound:
    inc si
    dec dx
    jz exit
    jmp inner

exit:
    mov ax,4c00h
    int 21h
    main endp
code_seg ends
    end start

```

5.4 编写如下矩阵相乘的程序：

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix}$$

答：程序如下（另附源程序文件 5_4.asm）

```

data_seg segment
    A_row1 db 3,6,2,4;demo array
    A_row2 db 7,1,4,10;demo array
    A_row3 db 8,2,9,17;demo array
    A_row4 db 9,11,12,7;demo array
    B_col1 db 2,21,41,13;demo array
    C_col1 dw 4 dup(?)

code_seg segment
    main proc far
        assume cs:code_seg,ds:data_seg
start:
    mov ax,data_seg
    mov ds,ax
    mov dx,0;dx is used to store the sum of mutiplication
    mov bx,0;bx is used to locate line number in A and C
    mov si,0;si is used to locate column number in A and line
number in B
    mov cx,4
outer:
    push cx
    mov cl,2
    shl bx,cl;4 times of bx equals to the start location of
certain line in A
    mov cx,4
inner:

    mov al,A_row1[bx][si]
    mul B_col1[si]
    add dx,ax
    inc si
    loop inner

    shr bx,1;2 times of bx equals to the start location of
certain line in C
    mov C_col1[bx],dx
    shr bx,1;restore bx to line number
    inc bx;next line
    mov dx,0
    mov si,0
    pop cx
    loop outer

```

```

exit:
    mov ax,4c00h
    int 21h
    main endp
code_seg ends
end start

```

5.5 计算 Fibonacci 数。计算公式如下：

$FIB(1)=1$

$FIB(2)=1$

$FIB(n)=FIB(n-1)+FIB(n-2) \quad n>2$

假设数据段定义如下：

```

data segment
    n dw ?

    result dw ? ; 存放 FIB(n) 的结果
data ends

```

答：程序如下（另附源程序文件 [5_5.asm](#)）

```

data segment
    n dw 11
    result dw ?
data ends

code segment
    main proc far
        assume cs:code,ds:data
start:
    ;initialize
    mov ax,data
    mov ds,ax
    ;start
    mov cx,n
    sub cx,2;We need to loop for n-2 times in order to calculate
FIB(n)
    mov ax,1
    mov bx,1
calc:
    mov dx,bx;temporarily storing previous bx
    add bx,ax
    mov ax,dx
    loop calc

```

```

mov result,bx;store in to result

;exit
mov ax,4c00h
int 21h
main endp
code ends
end start

```

5.6 改写教材 161 页例 5.1 的程序（将 BX 中的二进制以十六进制形式显示在屏幕上）。

要求：10~15 的出现以小写字母 a~f 表示；十六进制数后加'H'；若十六进制数最高位为字母，在其前面补 0。

例如：(BX)= 1111 0000 0000 0000 输出 0f000H

(BX)= 1000 0000 0001 1111 输出 801fH

答：程序如下（另附源程序文件 5_6.asm）

```

program segment
main proc far
    assume cs:program
start:
    ;initialize
    push ds
    sub ax,ax
    push ax
    ;main part of program

    mov bx,0fc9fh
    mov ch,4
rotate:
    mov cl,4
    rol bx,cl
    mov al,bl
    and al,0fh
    add al,30h
    cmp al,3ah ;compare to '9'
    jl printit ;if <'9', print it directly
    add al,27h ;else, add 7h to be a letter

```

```

printit:
    mov dh,al    ;backup for al, for int 21h will change ax
    cmp ch,4
    jl printit_normal
    ;(now it seems that this is the first char)
    cmp al,3ah
    jl printit_normal
    ;(now it seems that this is the first char and it's letter)
    mov dl,30h ;output '0'
    mov ah,2
    int 21h
printit_normal:
    mov al,dh    ;restore al
    mov dl,al
    mov ah,2
    int 21h
    dec ch
    jnz rotate ;if having not finished outputting 4 chars,
continue
    mov dl,48h ;else
    mov ah,2
    int 21h
    ret
main endp
program ends
    end start

```