

## 汇编第二次上机实验

### 1、 循环程序设计

数据段中已定义了一个长度为  $n$  ( $n \geq 16$ ) 的字数组  $M$ ，试编写一程序求出  $M$  中绝对值最大的数，把它放在数据段的  $M+2n$  单元中，并将该数的偏移地址存放在  $M+2(n+1)$  单元中， $M$  是数组首地址。要求：

(1) 把自己学号每 2 位一组，分别定义前 5 个字的初始值。例如，某同学的学号为 2195012345，则数组  $M$  的前 5 个字的定义为：21H, 95H, 01H, 23H, 45H，剩余的字根据程序调试需要进行初始化。

### 2、 分支程序设计

设计一个程序，在数据段用自己的学号定义一个字符串，字符串长度大于 20，对该字符串中字符 0~9 字符出现的次数进行计数，计数结果存放在以 COUNT 为首地址的数组中，并将出现次数最多的字符和出现次数显示出来。要求：

(1) 该字符串的前 10 个字符必须为自己的学号，例如某同学的学号为 2195012345，则该字符串的定义为 '2195012345xxxxxxxxxxxxxxxxxx'，字符串中的 x 根据程序调试需要进行初始化；

(2) 如果字符 9 出现的次数为 5 次，且是出现次数最多的字符，则在程序结束时输出： 9，5。若有多个次数相同的字符，则输出字符最大的哪个。

### 1、 循环程序设计

### (1) 反汇编的截图

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
0779:0048 8B87CC17      MOV     AX,[BX+17CC]
0779:004C 8B97CE17      MOV     DX,[BX+17CE]
0779:0050 8946FC        MOV     [BP-04],AX
0779:0053 8956FE        MOV     [BP-02],DX
-g
Program terminated normally
-q
C:\LEARN\EXP2>debug exp2.exe
-u
0779:0000 B87607      MOV     AX,0776
0779:0003 8ED8         MOV     DS,AX
0779:0005 2BC0         SUB     AX,AX
0779:0007 B91000      MOV     CX,0010
0779:000A 8B07         MOV     AX,[BX]
0779:000C A31E00      MOV     [001E],AX
0779:000F 891E2000    MOV     [0020],BX
0779:0013 83C302      ADD     BX,+02
0779:0016 3B07         CMP     AX,[BX]
0779:0018 7C08         JL      0022
0779:001A 83C302      ADD     BX,+02
0779:001D E2F7         LOOP   0016
0779:001F EB15         JMP     0036
-S
```

(2) 在进行计算前，显示数组 M 开始的 n+2 个字的内存值的截图（只能显示这 n+2 个字的内存值，多显示、少显示均扣分）

```
-d ds:0000122
0776:0000 21 00 94 00 21 00 43 00-36 00 11 00 22 00 33 00  !...!.C.6...".3.
0776:0010 44 00 55 00 66 00 77 00-88 00 99 00 00 01 00 00  D.U.f.w.....
0776:0020 00 00                                     ..
```

(3) 执行完计算后，显示数组 M 开始的 n+2 个字的内存值的截图（只能显示这 n+2 个字的内存值，多显示、少显示均扣分）

```
-d ds:0000 122
0776:0000 21 00 94 00 21 00 43 00-36 00 11 00 22 00 33 00  !...!.C.6...".3.
0776:0010 44 00 55 00 66 00 77 00-88 00 99 00 00 01 00 01  D.U.f.w.....
0776:0020 1C 00                                     ..
```

### (4) 源代码

```
datasg segment
```

```
M dw 0021h,0094h,0021h,0043h,0036h,0011h,0022h,0033h,0044h,0055h,0066h,0077h,0088h,0099h,0100h
```

```
ans dw ?
```

```
addr dw ?
```

```
;注意用逗号分隔
```

```
datasg ends
```

```
codesg segment
```

```
assume ds:datasg,cs:codesg
```

```
start:;要加 start 和 end start 要不然数据段不正常
```

```
mov ax,datasg
```

```
mov ds,ax;将数据段地址送 ds
```

```
sub ax,ax;将 ax 清零
```

```
mov cx,16;设置循环次数
```

```
mov ax,ds:[bx];以第一个数字为最大值
```

```
;注意 ds 后面的括号,使用寄存器间接寻址
```

```
;时只能用 bx,bp,si,di,不能用 ax
```

```
mov ans,ax
```

```
mov addr,bx
```

```
add bx,2
```

```
j :cmp ax,ds:[bx];与后一个数比较
```

```
jl c;小于则将后一个数置为最大
```

```
add bx,2;不小于则向后比较
```

```
loop j

jmp e;结束

c :mov ax,ds:[bx];更换最大数

mov ans,ax

mov addr,bx

dec cx;判断循环次数

cmp cx,0

je e;结束

add bx,2;向后比较

jmp j;回到比较环节

e :mov ax,4c00h

int 21h

codesg ends

end start
```

## 2、分支程序设计

### (1) 反汇编的截图

0779:0000	B87607	MOV	AX,0776
0779:0003	8ED8	MOV	DS,AX
0779:0005	BA0000	MOV	DX,0000
0779:0008	BE0000	MOV	SI,0000
0779:000B	8D3E1C00	LEA	DI,[001C]
0779:000F	B91C00	MOV	CX,001C
0779:0012	8A1C	MOV	BL,[SI]
0779:0014	B700	MOV	BH,00
0779:0016	FE41D0	INC	BYTE PTR [BX+DI-30]
0779:0019	46	INC	SI
0779:001A	E2F6	LOOP	0012
0779:001C	8D361C00	LEA	SI,[001C]

0779:001C	8D361C00	LEA	SI,[001C]
0779:0020	BB0000	MOV	BX,0000
0779:0023	B90900	MOV	CX,0009
0779:0026	2BC0	SUB	AX,AX
0779:0028	8A00	MOV	AL,[BX+SI]
0779:002A	881E2600	MOV	[0026],BL
0779:002E	A22900	MOV	[0029],AL
0779:0031	43	INC	BX
0779:0032	3A00	CMP	AL,[BX+SI]
0779:0034	7C08	JL	003E
0779:0036	7418	JZ	0050
0779:0038	43	INC	BX
0779:0039	E2F7	LOOP	0032
0779:003B	EB22	JMP	005F

0779:003B	EB22	JMP	005F
0779:003D	90	NOP	
0779:003E	8A00	MOV	AL,[BX+SI]
0779:0040	881E2600	MOV	[0026],BL
0779:0044	A22900	MOV	[0029],AL
0779:0047	49	DEC	CX
0779:0048	83F900	CMP	CX,+00
0779:004B	7412	JZ	005F
0779:004D	43	INC	BX
0779:004E	EBE2	JMP	0032
0779:0050	3A1E2600	CMP	BL,[0026]
0779:0054	7FE8	JG	003E
0779:0056	43	INC	BX
0779:0057	49	DEC	CX
0779:0058	83F900	CMP	CX,+00



```

-d ds:1c 25
0776:0010                                00 04 04 04      ....
0776:0020  04 02 03 02 02 03              .....

```

(6) 程序在 DOSBox 下直接运行的截图

```

C:\LEARN\EXP202>exp202
4,4

```

(7) 源代码

```

datasg segment

    mystr db '2194214336123456789123456789'
           ;共 28 个

    COUNT db 10 dup(0);存放计数

    ans db '0,$';注意要输出逗号,美元是字符串结尾

    time db '0$'

datasg ends

codesg segment

    assume ds:datasg,cs:codesg

start:

    mov ax,datasg

    mov ds,ax

    mov dx,0

    mov si,0

    lea di,COUNT

    mov cx,28;设置循环次数

;计数模块

```

```
j1: mov bl,ds:[si]
```

```
    mov bh,0
```

```
    inc byte ptr ds:[bx+di-30h]
```

```
    inc si
```

```
    loop j1
```

```
;找出出现最多的字符
```

```
    lea si,COUNT
```

```
    mov bx,0
```

```
    mov cx,9
```

```
    sub ax,ax;将 ax 清零
```

```
    mov al,ds:[bx+si];
```

```
    mov ans,bl
```

```
    mov time,al
```

```
    inc bx
```

```
j2 :cmp al,ds:[bx+si];与后一个数比较
```

```
    jl c;小于则将后一个数置为最大
```

```
    je equal;相等判断
```

```
    inc bx;不小于则向后比较
```

```
    loop j2
```

```
    jmp en;结束
```

```
c :mov al,ds:[bx+si];更换最大数
```



```
mov ans,bl

mov time,al

dec cx;判断循环次数

cmp cx,0

je en;结束

inc bx;向后比较

jmp j2;回到比较环节

equal;;相等情况

    cmp bl,ans

    jg c

    inc bx

    dec cx

    cmp cx,0

    je en

    jmp j2

;输出!!!!

en: add byte ptr ans,30h

    add byte ptr time,30h;转 ascii 码

    mov dx,offset ans;注意输出方式

    mov ah,09h

    int 21h
```

```
mov dx,offset time
```

```
mov ah,09h
```

```
int 21h
```

```
mov ax,4c00h
```

```
int 21h
```

```
codesg ends
```

```
end start
```