

# 汇编语言程序设计

## 作业选讲

P21 13, 18, 20, 22

13.将下列16位十六进制有符号数转换为十进制数

a.6BF9

解析： $6BF9 = 6 \times 16^3 + 11 \times 16^2 + 15 \times 16^1 + 9 = 27641$

b.C123

解析：对于有符号16位整数，有：  
负数的补码等于正数的反码+1，则负数的补码减1后取反码即可得  
这个负数所对应的正数。 $C123\text{ h} - 0001\text{ h} = C122\text{ h}$  取反  $\rightarrow 3EDD\text{ h}$   
 $= 16093\text{ d}$

所以，这个负数为-16093。

18. 下列有符号十进制整数的8位二进制（补码）表示分别是什么？

a. -72      答案 10111000

解析：72（十进制）=48（十六进制）=01001000（二进制）

其负数=原数取反+1；

01001000 取反 = 10110111，对其+1 = 10111000=B8

b. -98      答案 10011110

解析同上

c. -26      答案 11100110

解析同上

20. 下列每组16进制整数的和分别是多少？

a.  $7C4 + 3BE$

答案：2946

b.  $B69 + 7AD$

答案：4886

22. ASCII字符大写 “G” 的十六进制和十进制表示分别是什么？

答案：47h和71d

查ASCII表即可。

# P26 1, 2, 4, 5

1.中央处理单元CPU包含寄存器和哪些其他构件？

答：中央处理单元CPU是执行算数和逻辑操作的部件，由寄存器，高频时钟，控制单元/CU和算数逻辑单元/ALU构成。

2.中央处理单元通过哪三种总线与计算机系统的其他部分相连？

答：数据总线，地址总线和控制总线。

数据总线（Data Bus）：在CPU与RAM之间来回传送需要处理或是需要储存的数据。

地址总线（Address Bus）：用来指定在RAM（Random Access Memory）之中储存的数据的地址。

控制总线（Control Bus）：将微处理器控制单元（Control Unit）的信号，传送到周边设备。

4.指令执行周期包含哪三个基本步骤？

答：取指，译码和执行。

5.指令执行周期中，如果用到存储器操作数，则还需要哪两个步骤？

答：取操作数  
存放结果

# P30 1

1.x86处理器的3个基本操作模式是什么？

答: 保护模式， 实地址模式和系统管理模式。



P38 1, 3, 4, 6

1. 32位模式下，除了堆栈指针ESP寄存器，还有哪些寄存器指向堆栈的参数？

答：EBP

3.当无符号数算术运算结果超过目标标志大小时，应设置哪个标志位？

答：CF 进位标志位。

4. 当有符号数算术运算结果对目标位置而言太大或太小时，应设置哪个标志位？

答:OF 溢出标志位

6. 算术或逻辑运算产生负数结果时，应设置哪个标志位？

答：SF 符号标志位

P71 6, 7, 10, 11, 14, 15

6. 假设有数值456789ABh ,按小端序列列出其字节内容。

答：ABH , 89H , 67H , 45H

7. 声明一个数组， 其中包含120个未初始化无符号双字数值。

答： myArray DWORD 120 DUP(?)

10.声明一个16位无符号整数变量wArray, 使其具有3个初值。

答： wArray WORD 1000h, 2000h, 3000h

11. 声明一个字符串变量, 包含你喜欢的颜色名字, 并将其初始化为空字节结束的字符串。

答： favColor BYTE "blue",0

14. 声明一个数组bArray, 包含20个无符号字节, 将其所有的元素都初始化为0.

答: bArray BYTE 20 DUP(0)

15. 写出下述双字变量在内存中的字节序列 (低->高)

val dword 87654321h

答: 21h 43h 65h 87h

# 编程练习 1 , 2

1. 整数表达式计算：利用寄存器计算表达式： $A = (A + B) - (C - D)$ . 整数值分配给eax,ebx,ecx,edx。
2. 编写程序，为一周7天定义符号常量，创建一个数组变量，用这些符号常量作为初始值。

P104 – 107 4.9.1 4,5,6,9,10,17

4.执行下列指令后，EAX的值是多少？

```
Mov eax, 1002FFFFh
```

```
Neg ax
```

解析：ax是eax的低16位，即FFFFh，对其求补，结果为0001h  
因此eax的值为 10020001h。

5. 执行下列指令后，奇偶标志位的值是多少？

Mov al,1

Add al,3

解析：执行完之后， $al = 1 + 3 = 4$ ，转化为8位2进制为：

0000 0100 低8位中1的个数是1，为奇数，故PF=0



6. 执行下列指令后，EAX和符号标志位的值分别为多少？

Mov eax,5

Sub eax,6

解析：指令执行完之后  $\text{eax} = 5 - 6 = -1$  即FFFFFFFFh，故SF=1.

10. 执行下列指令后， EAX的值是多少？

.data

dVal DWORD 12345678h

.code

Mov ax,3

Mov word ptr dval+2,ax

Mov eax,dVal

答:     eax = 00035678h

17.顺序执行下列指令，则每条指令目标操作数的十六进制值是多少？

Mov al,var1 ; a.

Mov ah,[var1+3] ; b.

答:FCH 01H

## 4.9.2 1, 2, 7

1. 有一组变量名为three的双字变量，编写一组mov指令来交换该变量的高位字和低位字。

解析：

```
mov ax,word ptr three
```

```
mov bx,word ptr three+2
```

```
mov word ptr three,bx
```

```
mov word ptr three+2,ax
```

2.用不超过3条的XCHG指令对4个8位寄存器的值进行重排序，将其顺序从A,B,C,D调整为B,C,D,A

解析：

```
xchg al,bl
```

```
xchg al,cl
```

```
xchg al,dl
```

7.用汇编语言实现算术表达式： $EAX = -val2 + 7 - val3 + val1$ .假设val1,val2,val3都是32位整型变量。

# P107 4.10编程练习 1, 2, 8

1.将大端顺序转化为小端顺序。

使用下面的变量和mov指令编写程序，将数值从大端顺序复制为小端顺序，颠倒字节的顺序。32位数的十六进制值为12345678.

```
.data
```

```
bigEndian byte 12h,34h,56h,78h
```

```
littleEndian dword ?
```

# 答案

```
.386
.model flat,stdcall
.stack 4096
ExitProcess proto,dwExitCode:dword

.data
bigEndian byte 12h,34h,56h,78h
littleEndian dword ?

.code
main proc
    mov ah,byte ptr bigEndian           ; high byte
    mov al,byte ptr bigEndian+1         ; 2nd byte
    mov word ptr littleEndian+2,ax      ; high word
    mov ah,byte ptr bigEndian+2         ; 3rd byte
    mov al,byte ptr bigEndian+3         ; 4th byte
    mov word ptr littleEndian,ax        ; low word

    invoke ExitProcess,0
main endp
end main
```



## 2.交换数组元素对。

编写循环程序，用变址寻址交换数组中的数值对，每对中包含偶数个元素。即，元素 $i$ 与元素 $i+1$ 交换，元素 $i+2$ 与元素 $i+3$ 交换，类推。

```

.386
.model flat,stdcall
.stack 4096
ExitProcess proto,dwExitCode:dword

.data
array DWORD 1,2,3,4,5,6,7,8,9,10,11,12

.code
main PROC
    ; point to the first two array elements
    mov esi,0                                ; beginning of array
    mov ecx,LENGTHOF array / 2              ; loop (N / 2) times

    ; The loop swaps adjacent array elements, gradually
    ; moving towards the last pair.
L1:
    ; exchange array[esi] with array[edi], using indexed addressing.

    mov eax,array[esi]
    xchg eax,array[esi+4]
    mov array[esi],eax

    add esi,TYPE array*2                    ; pointer moves forward
    loop L1

    invoke ExitProcess,0
main endp
end main

```

## 8.数组元素移位

编写循环程序，用变址寻址把一个32位整数数组中的元素向前（向右）循环移动一个位置，数组最后一个元素的值移动到第一个位置上。如 数组[10, 20, 30, 40] 移位后转换为[40, 10, 20, 30].

```
.data
```

```
array dword 10h,20h,30h,40h
```

```
arraySize = 4
```

```
.code
```

```
nain proc
```

```
    mov rdi,3
```

```
    mov rsi,2
```

```
    mov eax,array[rdi*4]           ; save last value
```

```
    mov ecx,3
```

```
L1:
```

```
    mov edx,array[rsi*4]
```

```
    mov array[rdi*4],edx
```

```
    dec rsi
```

```
    dec rdi
```

```
    loop L1
```

```
    mov array[rdi*4],eax           ; store saved value in first position
```

```
    mov ecx,0                     ; assign a process return code
```

```
    call ExitProcess              ; terminate the program
```

```
nain endp
```

```
end
```

# 第五章 过程

P144 15, 16, 17, 18

15.执行下列指令后，EAX的最后数值是多少。

Push 5

Push 6

Pop eax

Pop eax

答：5

16.运行如下示例代码时，下面哪个对执行情况的陈述是正确的？

Main PROC

    push 10

    push 20

    call Ex2Sub

    pop eax

    INVOKE ExitProcess,0

Main endp

Ex2Sub PROC

    pop eax

    ret

Ex2Sub ENDP

- A.到第6行代码时, eax将等于10
- B.到第十行代码时, 程序将因运行时错误而暂停
- C.到第6行代码时, eax将等于20
- D.到第11行代码时, 程序将因运行时错误而暂停

答案：D

17.运行如下示例代码时，下面哪个对执行情况的描述是正确的？

Main PROC

mov eax,30

push eax

push 40

call Ex3Sub

INVOKE ExitProcess,0

Main ENDP

Ex3Sub PROC

pusha

mov eax,80

popa

ret

Ex3Sub ENDP



- a. 到第6行代码, eax将等于40
- b. 到第6行代码, 程序将因运行时错误而暂停
- c. 到第6行代码, eax将等于30
- d. 到第13行代码, 程序将因运行时错误而暂停

答案：C

18.运行如下示例代码时，下面哪个对执行情况的陈述是正确的？

Main PROC

    mov eax,40

    push offset Here

    jmp Ex4Sub

Here :

    mov eax,30

    INVOKE ExitProcess,0

Main ENDP

Ex4Sub PROC

    ret

Ex4Sub ENDP

- a.到第7行代码, eax将等于30
- b.到第4行代码, 程序将因运行时错误而暂停
- c.到第6行代码, eax将等于30
- d.到第11行代码, 程序将因运行时错误而暂停

答案 : b

P186 4, 5, 6, 7, 9

4.编写指令，若DX中的无符号数小于等于CX中的数，则跳转到标号L1。

解析：

```
cmp dx,cx
```

```
jbe L1
```

L1:

5.编写指令， 若AX中的有符号数大于CX中的数， 则跳转到标号L2

解析：

```
cmp ax,cx
```

```
jg L2
```

L2:

6. 编写指令，清除AL的位0和位1，若目的操作数等于零，则代码跳转到标号L3，否则跳转到标号L4。

解析：

```
and al,11111100b
```

```
jz L3
```

```
jmp L4
```

L3:

L4:

7.汇编语言实现短路求值。假设vall和X是32位变量

If (vall > ecx ) and ( ecx >edx)

    X=1;

Else

    X=2;

# 代码

```
cmp val1,ecx
```

```
jna L1
```

```
cmp ecx,edx
```

```
jna L1
```

```
mov X,1
```

```
jmp next
```

```
L1: mov X,2
```

```
next:
```



9.汇编语言实现下面伪代码。使用短路求值，并假设X是32位变量。

If (ebx > ecx and ebx >edx ) or (edx > eax)

    X = 1

Else

    X = 2

```
cmp ebx,ecx          ; ebx > ecx?
jna L1               ; no: try condition after OR
cmp ebx,edx          ; yes: is ebx > edx?
jna L1               ; no: try condition after OR
jmp L2               ; yes: set X to 1
```

```
;-----OR(edx > eax)-----
```

```
L1: cmp edx,eax      ; edx > eax?
jna L3               ; no: set X to 2
```

```
L2: mov X,1          ; yes:set X to 1
    jmp next         ; and quit
```

```
L3: mov X,2          ; set X to 2
next:
```

P224 3, 5, 6, 8, 10, 12

3.编写一条逻辑移位指令，实现eax乘以16

答：shl      eax,4

5.编写一条循环移位指令，交换DL寄存器的高4位和低4位。

答：**ror dl,4**

6.编写一条SHLD指令，把AX寄存器的最高位移入DX的最低位，并把DX左移一位。

答：shld dx,ax,1

8.编写指令序列，把三个内存字节左移一位，使用数据如下：

wordArray word 810Dh,0c064h,93ABh

答：这个扩展精度整数为：93AB C064 810Dh

.code

shl wordArray,1

rcl wordArray+2,1

rcl wordArray+4,1

10.编写指令，实现-276除以10，并把结果保存入一个16位变量val1.

答：

```
mov ax,-276
```

```
cwd ; sign-extend AX into DX
```

```
mov bx,10
```

```
idiv bx
```

```
mov val1,ax ; quotient
```

12.使用32位有符号操作数，用汇编语言实现下述C++表达式：  
 $Val1 = (val2 / val3) * (val1 + val2)$

答：

```
mov  eax,val2
```

```
cdq                                ; extend EAX into EDX
```

```
idiv val3                        ; EAX = quotient
```

```
mov  ebx,val1
```

```
add  ebx,val2
```

```
imul ebx
```

```
mov  val1,eax                    ; lower 32 bits of product
```

P276 1, 3, 4, 6

1.若过程有堆栈参数和局部变量，那么在结尾部分应包含哪些语句。

3.使用SRDCALL调用规范的程序在过程调用之后如何清除堆栈？

4.为什么LEA指令比OFFSET运算符功能强？

答：

LEA可以返回间接操作数的偏移量；它对于获取堆栈参数的偏移量特别有用。

lea指令在执行时计算出偏移地址，offset操作符在汇编阶段取得变量的偏移地址。offset无需在执行时计算，指令的执行速度更快，相对的，lea指令则可以获取到汇编阶段无法确定的偏移地址

6.与STDCALL调用规范相比，C调用规范会有哪些优势？

答：C调用约定允许可变长度参数列表。

# P276 算法基础 1

1.下面是过程AddThree的调用指令序列，该过程实现3个双字的加法。

Push 10h

Push 20h


Push 30h

Call AddThree

请画出EBP被压入运行时堆栈后过程栈帧的示意图



答：

10h	[EBP + 16]	
20h	[EBP + 12]	
30h	[EBP + 8]	
(return addr)	[EBP + 4]	
EBP		EBP ESP

P305 4, 5

4.编写指令, 用CMPSW比较两个16位的数组sourcew和targetw。

答：

```
mov ecx,count  
mov esi,offset sourcew  
mov edi,offset targetw  
cld  
repe cmpsw
```

5.编写指令，用SCASW在数组wordArray中扫描16位的数值0100h，并将匹配元素的偏移量复制到EAX寄存器。

```
cld
```

```
mov ax,0100h
```

```
mov edi,OFFSET wordArray
```

```
mov ecx,LENGTHOF wordArray
```

```
repne scasw
```

```
jnz quit ;没有找到，退出
```

```
sub edi,2 ;找到，调整偏移地址
```

```
mov eax,edi
```

```
quit:...
```

# 期末考试题型及分值分布

- 一、填空题（每空1分，共15分）
- 二、单选题（每小题2分，共20分）
- 三、简答题（每小题5分，共25分）
- 四、写程序段题（每小题6分，共24分）
- 五、综合编程题（每小题16分，共16分）