

# 汇编程序实例

## 汇编程序实例

二进制显示

统计区间人数

寻找最大数

数组累加

宏指令编写子程序

### 代码转换

十六进制ASCII ——> 二进制

十进制ASCII ——> 二进制

4位BCD码数→二进制数显示

二进制数→十六进制数显示

二进制数→十进制数

### 字符串处理

数据查找

字符串比较

## 二进制显示

- 程序要求

在显示器上将BX的内容以二进制显示在屏幕上

```
;二进制显示
.486
CODE SEGMENT USE16
ASSUME CS:CODE
BEG:    MOV BX,5678H
        MOV CX,16
LAST:   MOV AL,'0'
        ROL BX,1
        JNC NEXT
        MOV AL,'1'
NEXT:   MOV AH,0EH
        INT 10H
        LOOP LAST
        MOV AH,4CH
        INT 21H
CODE ENDS
```

- 知识点1: ROL 非进位循环左移; SAL 补零非循环左移; RCL 带进位循环左移
- 知识点2: BIOS的OE功能调用, 打印AL中的字符

## 统计区间人数

- 程序要求

统计月收入在2000~4000间的人数, 并用十进制数显示

```
.586
DATA SEGMENT USE16
    NUM DW 1000,23232,2300,4895,2999,1299,8769,4545,9990
DATA ENDS
CODE SEGMENT USE16
ASSUME CS:CODE,DS:DATA
BEG:    MOV AX,DATA
        MOV DS,AX
        MOV BX,OFFSET NUM
        MOV CX,9
        MOV DL,0
LAST:   CMP WORD PTR [BX],2000
        JC NEXT
        CMP WORD PTR [BX],4000
        JA NEXT
        INC DL
NEXT:   INC BX
        INC BX
        LOOP LAST
        ADD DL,30H ;
        MOV AH,2;
        INT 21H
        MOV AH,4CH
        INT 21H
CODE ENDS
END BEG
```

- 知识点1: JC判断A是否小于B, 等效命令JB; JA判断A是否大于B
- 知识点2: 将数字转换为符号格式, 需要加上48或者30H
- 知识点3: DOS功能调用2号功能, 为打印DL中的字符

## 寻找最大数

- 程序要求

假设从BUF单元开始为一个 ASCII码字符串,找出其中的最大数送屏幕显示。

```
.486
DATA SEGMENT USE16
    BUF DB 'AJIIHJNZ89123'
    COT EQU $-BUF
    DIS DB 'MAX= ',?,0DH,0AH,'$'
DATA ENDS

CODE SEGMENT USE16
ASSUME CS:CODE,DS:DATA
BEG:    MOV AX,DATA
        MOV DS,AX
        MOV CX,COT
        MOV BX,OFFSET BUF
        MOV AL,0
LAST:   CMP [BX],AL
        JB NEXT
        MOV AL,[BX]
NEXT:   INC BX
        LOOP LAST
        MOV DIS+4,AL
        MOV DX,OFFSET DIS
        MOV AH,09H
        INT 21H
        MOV AH,4CH
        INT 21H

CODE ENDS
END BEG
```

- 知识点1: `COT EQU $-BUF` 用来计算BUF段的长度
- 知识点2: `0DH 0AH` 表示回车换行的字符, 常用于字符串结束
- 知识点3: 调用DOS的09功能, 打印DS:DX段的字符串, 以\$为结束标志

## 数组累加

- 程序要求

设ARRAY是5个字元素的数组, 用子程序计算数组元素的累加和(不计进位), 并将结果存入变量RESULT中。

```
DATA SEGMENT USE16
    ARRAY DW 1111H,2222H,3333H,4444H,5555H
    RESULT DW ?
DATA ENDS
CODE SEGMENT USE16
```

```

ASSUME CS:CODE,DS:DATA
BEG:    MOV AX,DATA
        MOV DS,AX
        MOV CX,5
        MOV BX,OFFSET ARRAY
        CALL COMPUTE
XYZ:    MOV RESULT,AX
EXIT:   MOV AH,
        MOV AH,4CH
        INT 21H

COMPUTE PROC
        MOV AX,0
AGA:    ADD AX,[BX]
        ADD BX,2
        LOOP AGA
        RET
COMPUTE ENDP
CODE ENDS
END BEG

```

- 知识点1: 利用 `CALL XXX + XXX PROC + XXX ENDP` 语句格式可以编写子程序调用, 子程序内部最后需要用 `RET` 进行返回

## 宏指令编写子程序

- 程序要求

定位显示彩色字符串, 要求置显示器为彩色文本方式, 并在0行5列 显示 黑底绿色 HELLO; 12行36列 显示 黑底红色 WELCOME! ; 24行66列 显示 黑底黄色 BYE\_BYE

```

.486
DISP MACRO Y,X,VAR,LENGTH,COLOR
    MOV AH,13H
    MOV AL,1
    MOV BH,0                ;选择0页显示屏
    MOV BL,COLOR            ;属性字(颜色值) →BL
    MOV CX,LENGTH           ;串长度 →CX
    MOV DH,Y                ;行号 →DH
    MOV DL,X                ;列号 →DL
    MOV BP,OFFSET VAR       ;串有效地址→BP
    INT 10H
ENDM

EDATA SEGMENT USE16
    SS1 DB 'HELLO'
    SS2 DB 'WELCOME !'
    SS3 DB 'BYE_BYE'
EDATA ENDS

```

```

CODE SEGMENT USE16
ASSUME CS:CODE,ES:EDATA
    MOV AX,EDATA
    MOV ES,AX
    MOV AX,3
    INT 10H
    DISP 0,5,SS1,5,2          ;0行5列显示绿色HELLO
    DISP 12,36,SS2,8,4        ;12行36列显示红色WELCOME
    DISP 24,66,SS3,7,0EH      ;24行66列显示黄色BYE_BYE
SCAN:  MOV AH,1
        INT 16H
        JZ SCAN                ;等待用户键入,无键入转
        MOV AX,2
        INT 10H                ;恢复80×25黑白文本方式
        MOV AH,4CH
        INT 21H
CODE ENDS
END BEG

```

- 知识点1: 使用 `宏名 实参列表` 进行调用, 利用 `宏名 MACRO 形参列表` + `ENDM` 对宏指令进行定义
- 知识点2: 调用BIOS的13H的功能用于写入字符串, 具体参数设置参考

<b>Input:</b>	AH	=	13h
	AL	=	Output mode
			00h Attribute in BL, do not update cursor position.
			01h Attribute in BL, update cursor position.
			02h Attribute in string buffer, do not update cursor position.
			03h Attribute in string buffer, update cursor position.
	BH	=	Video page number
	BL	=	Attribute of all characters in character string
	CX	=	Number of characters in buffer
	DH	=	Screen line number
	DL	=	Screen column number
	ES:BP	=	Segment:offset address of string buffer

- 知识点3: 16H的01号功能调用用于返回键盘状态

## 代码转换

十六进制ASCII ——> 二进制

- 程序要求

键盘输入的一位十六进制ASCII码→二进制数显示。

```
DATA SEGMENT USE16
    DIS1 DB 'Please',00H,'Enter',0DH,0AH,'$'
    DIS2 DB '=', '$'
DATA ENDS
CODE SEGMENT USE16
ASSUME CS:CODE,DS:DATA
BEG:    MOV AX,DATA
        MOV DS,AX
        MOV AH,09H
        MOV DX,OFFSET DIS1
        INT 21H
        MOV AH,01H
        INT 21H
        MOV AH,09H
        MOV DX,OFFSET DIS2
        INT 21H
        MOV DL,AL
        SUB DL,30H
        MOV CX,8
LAST:   MOV AL,'0'
        ROL DL,1
        JNC NEXT
        MOV AL,'1'
NEXT:   MOV AH,0EH
        INT 10H
        LOOP LAST
        MOV AH,4CH
        INT 21H
CODE ENDS
END BEG
```

- 知识点1: 00H字符对应空格, 0DH+0AH 回车换行
- 知识点2: DOS功能09H,打印DS: DX的字符串
- 知识点3: DOS功能的01H,读取键盘输入, 传至AL
- 其余知识点参考二进制显示子程序

## 十进制ASCII ——> 二进制

- 程序要求

将BUF缓冲区中保存的四位十进制数ASCII码（假设十进制数的范围为0-65535）转换为对应的二进制数并显示。

```
DATA SEGMENT USE16
    BUF DB '3','4','8','9'
```

```

        COUNT EQU $-BUF
DATA ENDS
CODE SEGMENT USE16
ASSUME CS:CODE,DS:DATA
BEG:    MOV AX,DATA
        MOV DS,AX
        MOV CX,COUNT
        MOV SI,OFFSET BUF
        MOV AX,0
        MOV DH,0
        MOV BX,10
AGA:    MUL BX
        MOV DL,[SI]
        SUB DL,30H
        ADD AX,DX
        INC SI
        LOOP AGA
        MOV BX,AX
        CALL DISP
        MOV AH,2
        MOV DL,'B'
        INT 21H
EXIT:   MOV AH,4CH
        INT 21H

;-----
DISP PROC
        MOV CX,16
LAST:   MOV DL,'0'
        RCL BX,1
        JNC NEXT
        MOV DX,'1'
NEXT:   MOV AH,2
        INT 21H
        LOOP LAST
        RET
DISP ENDP
CODE ENDS
END BEG

```

- 知识点1: DOS功能的02号功能调用, 显示DL的字符

## 4位BCD码数→二进制数显示

```

.486
DATA SEGMENT USE16
        NUM DW 1001H
DATA ENDS

```

```

CODE SEGMENT USE16
ASSUME CS:CODE,DS:DATA
BEG:
    MOV AX,DATA
    MOV DS,AX
    MOV CX,4
    MOV AX,0
    MOV DX,NUM

JAA:
    ROL DX,4
    MOV BX,DX
    AND BX,000FH
    IMUL AX,10
    ADD AX,BX
    LOOP JAA
    CALL DISP
    MOV AH,4CH
    INT 21H

DISP PROC
    MOV CX,16
    MOV BX,AX
LAST1: MOV AL,'0'
    ROL BX,1
    JNC NEXT
    MOV AL,'1'
NEXT:  MOV AH,0EH
    INT 10H
    LOOP LAST1
    RET

DISP ENDP
CODE ENDS
END BEG

```

- 知识点1：IMUL用于有符号的乘法，无位数要求，小于七位二进制的数均可直接使用
- 知识点2：BIOS的0E功能调用的参数如下：

### Function 0Eh Write Character

<b>Input:</b>	AH	=	0Eh
	AL	=	ASCII Code of the character
	BH	=	Active video page number.
	BL	=	Foreground color (graphics modes)
<b>Output:</b>	AL	=	No registers set. 40:50h is updated

二进制数→十六进制数显示



```

.586
DATA SEGMENT USE16
    BUF DW 987AH
DATA ENDS
CODE SEGMENT USE16
ASSUME CS:CODE,DS:DATA
BEG:    MOV AX,DATA
        MOV DS,AX
        MOV DX,BUF
        MOV CX,4
        SAL EDX,16
AGA:    ROL EDX,4
        AND DL,0FH
        CMP DL,10
        JC NEXT
        ADD DL,7
NEXT:   ADD DL,30H
        MOV AH,2
        INT 21H
        LOOP AGA
        MOV AH,4CH
        INT 21H
CODE ENDS
END BEG

```

- 知识点1: DOS的02功能调用显示DL字符

## 二进制数→十进制数

- 程序要求

8位二进制数→十进制数, 比较法,

比较法的关键: 判断某二进制数(假设为BEN单元的内容)包含几个100, 几个10, 几个1

```

.586
CMPDISP MACRO NN
LOCAL LAST,NEXT
    MOV DL,0 ;DL清0
LAST:  CMP BEN,NN ;比较
        JB NEXT ;BEN < NN 转
        INC DL ;DL加1
        SUB BEN,NN ;BEN-NN→BEN
        JMP LAST
NEXT:  ADD DL,30H ;转换为ASCII码
ENDM

DATA SEGMENT USE16
    BEN DW 1287H ;4743
    TAB DW 10000,1000,100,10,1

```

```

        COUNT EQU ($-TAB)/2
        BUF DB COUNT DUP(?),'$' ; 输出缓冲区
DATA ENDS

CODE SEGMENT USE16
ASSUME CS:CODE,DS:DATA
BEG:     MOV AX,DATA
        MOV DS,AX
        MOV CX,COUNT
        MOV BX,OFFSET TAB
        MOV SI,OFFSET BUF ;输出缓冲区地址→SI
AGA:     MOV AX,[BX]
        CMPDISP AX ;调用宏得到ASCII码
        MOV [SI],DL ;将ASCII码保存到输出缓冲区中
        ADD BX,2 ;BX+2→BX
        INC SI
        LOOP AGA
        MOV SI,OFFSET BUF ;输出缓冲区地址重新→SI
NOSP:    CMP BYTE PTR [SI],30H ;判断字符是否为'0'
        JNZ DISP ;不等,跳转到DISP显示
        INC SI
        JMP NOSP
DISP:    MOV AH,9 ; 显示十进制数
        MOV DX,SI
        INT 21H
        MOV AH,4CH
        INT 21H
CODE ENDS
END BEG

```

- 知识点1: 计数定义可以进行计算 `COUNT EQU ($-TAB)/2`
- 知识点2: 重复定义语句 `DB n DUP(?)`

## 字符串处理

### 数据查找

- 程序要求

设从BUF单元开始, 存有一字符串, 找出其中 ASCII码最小和最大的字符,并送屏幕显示。

```

.586
DATA SEGMENT USE16
        BUF DB 'DLSIEFLIEFAWOKFADL'
        COUNT EQU $-BUF

```

```

        MAX DB 'MAX= ',?,0DH,0AH,'$'
        MIN DB 'MIN= ',?, '$'
DATA ENDS
CODE SEGMENT USE16
ASSUME CS:CODE,DS:DATA
BEG:     MOV AX,DATA
        MOV DS,AX
        MOV AL,BUF
        MOV MAX+4,AL      ;假设第一个数是最大数
        MOV MIN+4,AL      ;假设第一个数是最小数
        MOV BX,OFFSET BUF+1
        MOV CX,COUNT-1    ;比较次数
LAST:    MOV AL,[BX]
        CMP AL,MAX+4      ;比较最大数
        JNA LESS
        MOV MAX+4,AL      ;大数→MAX+4
LESS:    CMP AL,MIN+4      ;比较最小数
        JNC NEXT
        MOV MIN+4,AL      ;小数→MIN+4
NEXT:    INC BX
        LOOP LAST
        MOV AH,9
        MOV DX,OFFSET MAX
        INT 21H           ;显示最大字符
        MOV AH,9
        MOV DX,OFFSET MIN
        INT 21H           ;显示最小字符
        MOV AH,4CH
        INT 21H
CODE ENDS
END BEG

```

- 知识点1：可以对变量直接进行加减取得相应元素内容： `BUF+1` , `COUNT-1` , `MAX+4` , `MIN+4`

## 字符串比较

- 程序要求

比较从键盘输入的字符串STR1，和数据段中定义的字符串STR2是否相等。若相等，则置FLAG单元为‘Y’，不相等则置为‘N’

```

.586
DATA SEGMENT USE16
    STR1 DB 30,?,30 DUP(?) ;输入字符串缓冲区
    STR2 DB 'WELCOME'
    COUNT EQU $-STR2 ;统计串长度
    FLAG DB 'N' ;存放比较结果，初始为‘N’
DATA ENDS

```

```

CODE SEGMENT USE16
ASSUME CS:CODE,DS:DATA,ES:DATA
BEG:    MOV AX, DATA
        MOV DS, AX ;DS初始化
        MOV ES, AX ;ES初始化
        MOV CX, COUNT
        MOV AH, 0AH ;从键盘输入字符串STR2
        MOV DX, OFFSET STR1
        INT 21H
        MOV CL, STR1+1
        MOV CH, 0 ;输入字符串长度→CX
        CMP CX, COUNT ;比较STR1和STR2的长度是否相等
        JNZ EXIT ;字符串比较不相等, 跳转
        MOV SI, OFFSET STR1+2 ;原串首址→SI
        MOV DI, OFFSET STR2 ;目标区首址→DI
        CLD ;D标志为0, 增址型
LOAD:   REPE CMPSB ;两字符串比较
        JNZ EXIT ;字符串比较不相等, 跳转
        MOV FLAG, 'Y' ;字符串相等, 置FLAG为'Y'
EXIT:   MOV AH, 4CH
        INT 21H
CODE ENDS
END BEG

```

- 知识点：字符串操作

- 比较两串串字符是否相等, 结果存入入Z标志
- 源串串首首址/末址→DS:SI
- 目目串串首首址/末址→ES:DI
- 0/1 →D

```

CMPSB ;字节串比较
CMPSW ;字型串比较
CMPSD ;双字串比较
      ;重复前缀

REPE CMPSB
REPE CMPSW
REPE CMPSD
REPNE CMPSB
REPNE CMPSW
REPNE CMPSD

```