4 汇编语言程序设计

- 汇编语言程序格式和伪指令
- 常用 DOS 功能调用
- 汇编语言程序设计

4.1 汇编语言程序格式和伪指令

```
DSEG    SEGMENT
D1      DB   12H
D2      DB   —2          ;Define variable
D3      DB   ？
D4      DB  'Example',0DH,0AH,'$'
DSEG    ENDS

SSEG    SEGMENT   STACK
        DW   100 DUP(?)
SSEG    ENDS
```

```
CSEG    SEGMENT                 ;define segment
        ASSUME  CS：CSEG，DS：DSEG
        ASSUME  SS：SSEG
MAIN    PROC      FAR           ;define Procedure
START： PUSH      DS            ;return to DOS
        MOV       AX，0
        PUSH      AX
        MOV       AX，DSEG
        MOV       DS，AX
        MOV       AH，9
        MOV       DX，OFFSET D4
        INT       21H
        MOV       AL，D1
        ADD       AL，D2
        MOV       D3，AL
        RET
MAIN    ENDP
CSEG    ENDS
        END START               ;End program
```

Structure EQU syntactical MACRO Example Assembler Debugger

SEGMENT / ENDS

格式:

```
Name      SEGMENT  [options]      ;Begin segment
          ……
Name      ENDS                    ;End segment
```

options: [定位类型] [组合类型] [分类名]

功能: 段定义.

ASSUME

格式:

ASSUME 段寄存器名:段名符 [, 段寄存器名:段名符, …]

功能: 段分配, 将段寄存器与段名关联.

ORG

格式: ORG Expression

功能 : 设置段内偏移地址.

PROC / ENDP

格式:

```
        Name        PROC  Type
                    ……                  ;  Instructions
                    RET
        Name        ENDP
```

Type: NEAR , FAR

功能: 过程定义.

END

格式:　　　END Expression

Expression: 第一条指令标号.

功能:　　　源程序结束.

汇编语言程序结构

```
SSEG      SEGMENT  STACK
          ……                          ；  Define stack segment
SSEG      ENDS

DSEG    SEGMENT
          ……                          ；  define data segment
DSEG    ENDS

ESEG      SEGMENT
          ……                          ；  define extra segment
ESEG      ENDS

CSEG      SEGMENT
          ASSUME  CS： CSEG， DS： DSEG，  ES： ESEG， SS： SSEG
MAIN      PROC      FAR
          ……                          ；  define code segment
          RET
MAIN      ENDP
CSEG      ENDS
          END       MAIN              ；   end of the program
```

DB/DW/DD/DQ/DT

格式：	Name Data_Type Expression
Data_Type：	DB ——定义字节变量,分配存储单元并赋初值.
	DW ——Define Word .
	DD ——Define Doubleword
	DQ ——Define Quadword
	DT ——Define Ten Bytes
Expression：	constant, character string, ?, DUP directive, label
功能：	变量定义.

Example：

```
DSEG   SEGMENT
XB     DB     10，—4，’AB’，？
XW     DW     XB+2，100H，-5，’AB’，$+2
XD     DD     3*20，0FFFDH
XDUP   DB     2(1，2 DUP(2))
DSEG   ENDS
```

EQU

格式: **Name EQU Expression**

功能: 赋值.

宏指令

宏定义　Name　MACRO　<parameters>

　　　　　……　　　;; Instructions

　　　　　ENDM

宏调用　Name　real_variable

宏展开

宏定义

```
MOVE    MACRO   A，B
        PUSH    AX
        MOV     AX，B
        MOV     A，AX
        POP     AX
        ENDM
```

※代码段外

宏调用

```
MOVE  VAR1，VAR2
```

※ 代码段中

宏展开

```
1       PUSH    AX
1       MOV     AX    ，
VAR2
1       MOV     VAR1  ，
AX
1       POP     AX
```

※ .OBJ和.EXE中已宏展开

汇编语言语句格式

- 伪指令格式

  [名字] 助记符 [参数] [[;注释]

- 指令格式

  [标号:] [前缀指令] 助记符 操作数 [;注释]

- 参数（操作数）

  常数、寄存器、标号、变量、表达式

运算符和操作符

类型	符号	功能
算术运算符	＋, －, ×, /, MOD	加, 减, 乘, 除, 模除
逻辑运算符	AND, OR, XOR, NOT	与, 或, 异或, 非运算
关系运算符	EQ, NE, LT, LE, GT, GE	=,≠,<,≤,>,≥ (结果为真输出全1, 为假输出全0)
分析运算符	OFFSET	返回偏移地址.
	SEG	返回段基址.
	TYPE	变量类型, 返回元素字节数.
	LENGTH	返回元素个数. (DUP前重复次数）
	SIZE	返回变量总字节数 (SIZE=TYPE * LENGTH)
合成运算符	PTR	修改类型属性.
	SHORT	短转移说明.
其它运算符	（）	改变运算符优先级.
	[ ]	下标或间接寻址.

Example

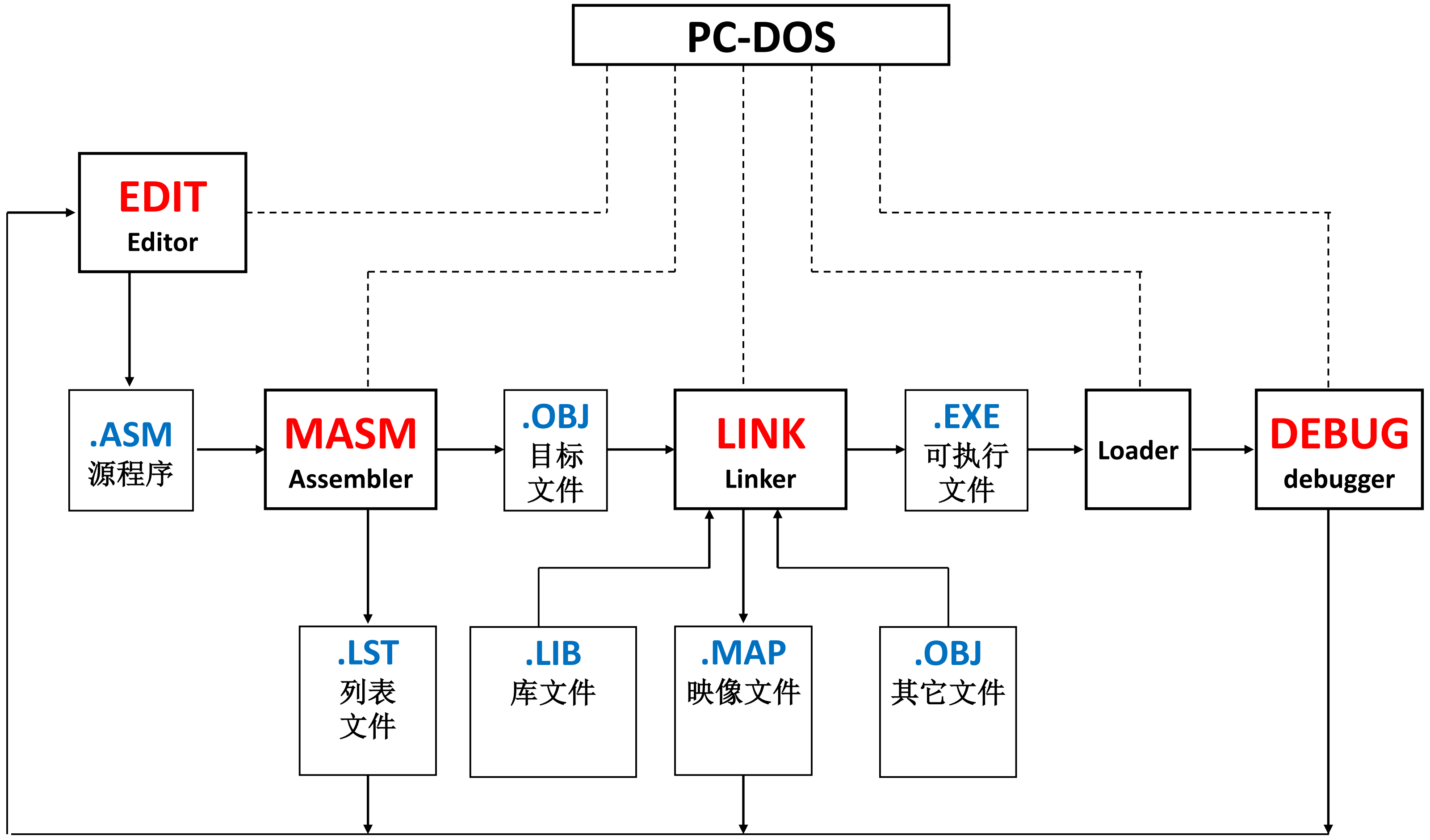
数据段定义如下：

```
DATA    SEGMENT
        ORG     100H
DA1     DB      ‘ABC’，42H
DA2     DW      03H，’BC’，$+2
DA3     DW      DA2
AA1     EQU     $－DA1
        ORG     $＋4
BB1     DB      10 DUP(2，2 DUP(?))
DATA    ENDS
```

目的操作数=?

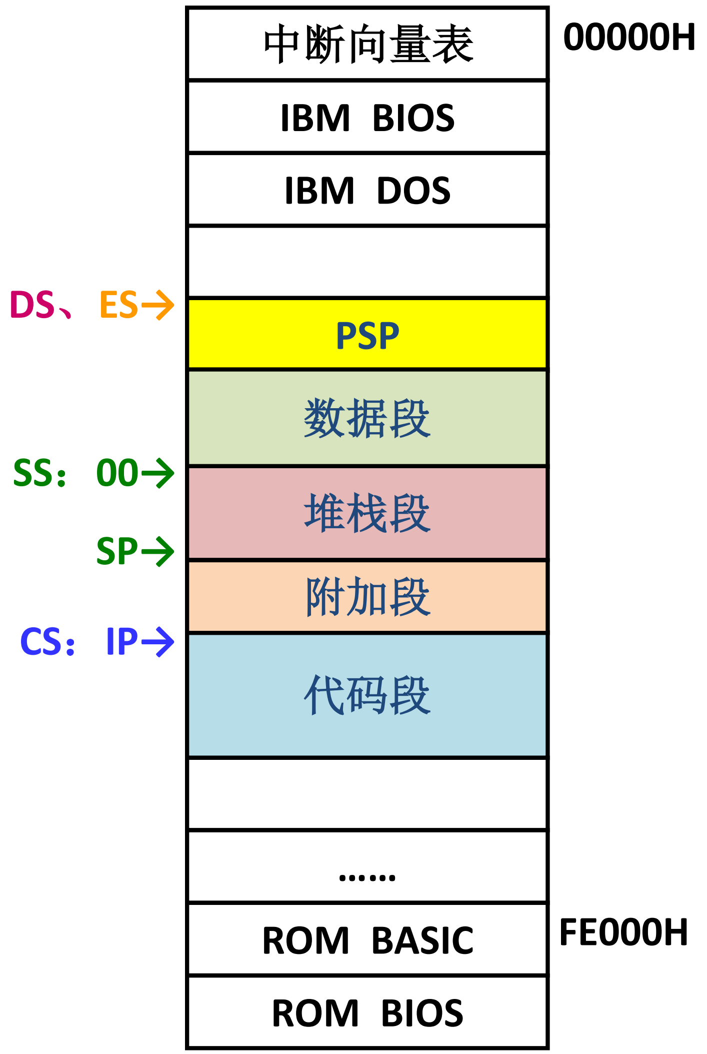
```
LEA     BX，DA2
MOV     DI，OFFSET BB1
MOV     AL，TYPE DA1
MOV     AX，AA1
MOV     AL，LENGTH BB1
MOV     AL，BYTE PTR DA2
MOV     AX，DA2+2
MOV     AX，DA3
MOV     AL，SIZE BB1
```

4.2 汇编语言程序的运行过程



程序装入与运行（汇编语言和DOS接口）

- ✓ 建立程序前缀区PSP
- ✓ 程序装入到内存中并定位
- ✓ 设置段寄存器值
- ✓ 执行程序



汇编语言主程序格式

```
MAIN        PROC   FAR
START:      PUSH   DS
            MOV  AX， 0
            PUSH AX
            MOV  AX， DSEG
            MOV  DS， AX
            ……                    ； instructions

            RET
MAIN        ENDP
```

Debug 调试命令

Command	Command Syntax	Description
Dump	-D [range]	Displays a portion of memory in hex and ASCII.
Enter	-E address [list]	Places individual bytes in memory.
Register	-R [register]	Displays the register contents on the screen.
Go	-G [=address] [addresses]	Executes the program in memory.
Trace	-T [=address] [value]	Executes one or more instructions from the current CS:IP location or optional address, if specified.
Proceed	-P [=address] [number]	Traces the program without entering the subroutine or interrupt.
Unassemble	-U [range]	Translates memory into assembly language mnemonics.
Quit	-Q	Quit from debug.

FR标志位含义

标志位	置位	显示	含义	复位	显示	含义
溢出标志	OF=1	OV	***Overflow***	OF=0	NV	***Not Overflow***
方向标志	DF=1	DN	***Down***	DF=0	UP	***Up***
中断标志	IF=1	EI	***Enable Interrupt***	IF=0	DI	***DI—Disable Interrupt***
符号标志	SF=1	NG	***Negative***	SF=0	PL	***Plus***
零标志	ZF=1	ZR	***Zero***	ZF=0	NZ	***Not Zero***
辅助进位	AF=1	AC	***Auxiliary Carry***	AF=0	NA	***Not Auxiliary Carry***
奇偶标志	PF=1	PE	***Parity Even***	PF=0	PO	***Parity Odd***
进位标志	CF=1	CY	***Carry***	CF=0	NC	***Not Carry***

4.3 DOS系统功能调用

- 系统功能调用方法

✓ 功能号→ **AH**

✓ 入口参数

✓ **INT 21H**

✓ 出口参数

常用DOS功能调用

功能号	功能	入口参数	出口参数
01H	等待从键盘输入一字符,并在屏幕上显示,检查Ctrl+Break		AL=输入字符
02H	显示单个字符	DL=字符ASCII码	

06H	键盘输入一字符（不等待,不判断,不回显）或屏幕显示一字符	DL=0FFH（输入）
		DL=字符（输出）
			ZF (=0时
			AL=输入字符)

08H	等待从键盘输入一字符,无回显,检查Ctrl+Break		AL=输入字符
09H	显示以’$’结尾的字符串	DS:DX=字符串首地址	
0AH	输入字符串到内存缓冲区	DS:DX=缓冲区首地址	
4CH	程序终止		

4CH号调用

```
MAIN      PROC    FAR
START：   PUSH    DS
          MOV     AX，0
          PUSH    AX
          MOV     AX，DSEG
          MOV     DS，AX
          ……      ;instructions

          RET
MAIN      ENDP
```

```
MAIN      PROC
START：   MOV     AX，DSEG
          MOV     DS，AX

          ……  ;instructions

          MOV     AX，4C00H
          INT     21H
MAIN      ENDP
```

4.4 汇编语言程序设计

- 程序结构

✓ 顺序结构

✓ 分支结构

✓ 循环结构

✓ 子程序结构

- Examples

✓ 查找并统计负数个数

✓ 字符串传送

✓ ASCII→Binary

✓ 查找字符

✓ 查找最大值

✓ 排序

✓ 统计数字,字母,其它字符

✓ 回文判断

Example: 字数据的二进制显示.

```
CSEG    SEGMENT
        ASSUME  CS：  CSEG
MAIN    PROC    FAR
        PUSH    DS
        XOR     AX，  AX
        PUSH    AX
        MOV     CX，  16
L1：    ROL     BX，  1
        MOV     DX，  BX
        AND     DL，  1
        ADD     DL，  30H
        MOV     AH，  02H
        INT     21H
        LOOP    L1
        RET
MAIN    ENDP
CSEG    ENDS
        END     MAIN
```

```
DISPBX2     PROC
            PUSH CX
            PUSH DX
            PUSH AX
            MOV  CX，  16
L1：        ROL  BX，  1
            MOV  DX，  BX
            AND  DL，  1
            ADD  DL，  30H
            MOV  AH，  02H
            INT  21H
            LOOP L1
            POP  AX
            POP  DX
            POP  CX
            RET
DISPBX2     ENDP
```

Example: 字数据的十六进制显示.

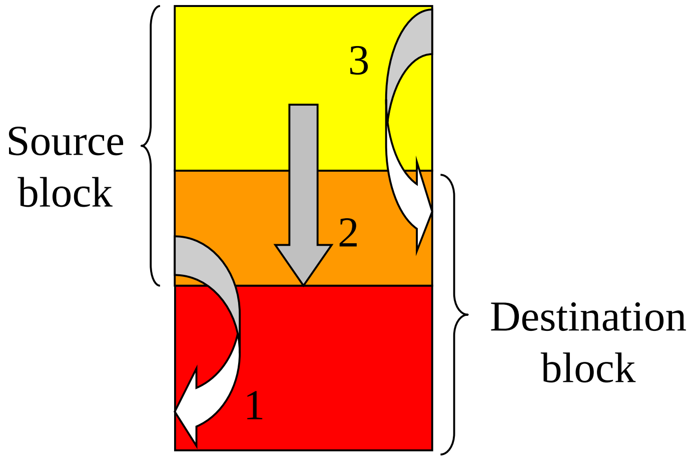
```
DSEG      SEGMENT
DATA      DW        0a12H
DSEG      ENDS
SSEG      SEGMEMT   STACK
          DW        100 DUP(？ )
SSEG      ENDS
CSEG      SEGMENT
          ASSUME    CS:CSEG,DS:DSEG,SS:SSEG
MAIN      PROC      FAR
          PUSH      DS
          XOR       AX， AX
          PUSH      AX
          MOV       AX， DSEG
          MOV       DS， AX
          MOV       BX， 26
          CALL      DISPBX16
          MOV       BX， DATA
          CALL      DISPBX16
          RET
MAIN      ENDP
```

```
DISPBX16      PROC
              PUSH  CX
              PUSH  DX
              MOV   CH， 04H
L1:           MOV   CL， 04H
              ROL   BX， CL
              MOV   DX， BX
              AND   DL， 0FH
              CMP   DL， 9H
              JBE   L2
              ADD   DL， 07H
L2:           ADD   DL， 30H
              MOV   AH， 02H
              INT   21H
              DEC   CH
         JNZ  L1
              POP   DX
              POP   CX
              RET
DISPBX16      ENDP
CSEG          ENDS
              END   MAIN
```

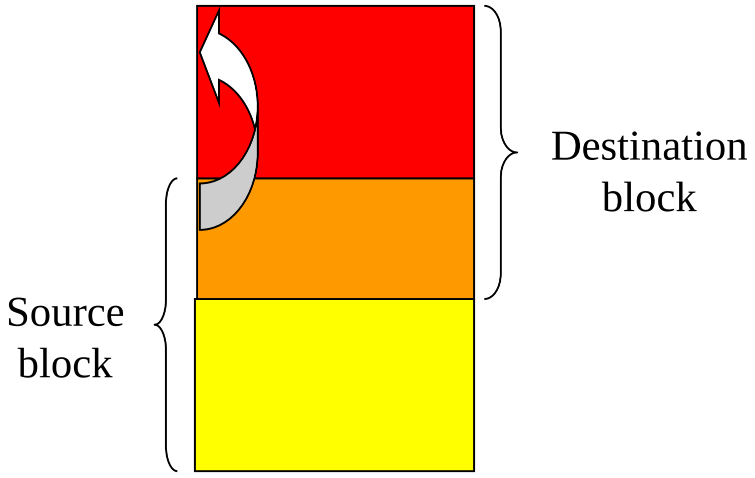
Example: 从ARRAY中查找负数,并统计负数的个数.

```
DSEG    SEGMENT
ARRAY   DB  92H,23H,96H,0A3H,25H,……
COUNT EQU $-ARRAY
MES1    DB  ‘Negative numbers are:$’
MES2    DB  0DH， 0AH， ‘Negative number count is:$’
DSEG    ENDS
SSEG    SEGMENT    STACK
        DW  100 DUP(?)
SSEG    ENDS
CSEG    SEGMENT
        ASSUME CS:CSEG， DS:DSEG
        ASSUME SS:SSEG
MAIN    PROC        FAR
        MOV         AX， DSEG
        MOV         DS， AX
        MOV         DX， OFFSET  MES1
        MOV         AH， 9
        INT         21H
        MOV         CX， COUNT
        MOV         BH， 0     ;负数个数
        LEA         SI， ARRAY
BACK：  MOV     AL， [SI]
        TEST    AL， 80H
        JZ      NEXT
        MOV     BL， AL
        CALL    DISPBL16
        INC     BH
NEXT：  INC     SI
        LOOP    BACK
        MOV     DX， OFFSET  MES2
        MOV     AH， 9
        INT     21H
        MOV     BL， BH
        CALL    DISPBL16
EXIT：  MOV     AX， 4C00H
        INT     21H
MAIN    ENDP
CSEG    ENDS
        END     MAIN
```

Example：字符串传送.



（a） Decrement mode



（b）Increment mode

```
DSEG     SEGMENT
STRG     DB  100 DUP（？ ）
DSEG     ENDS
MIAIN    PROC   FAR
         MOV    AX， DSEG
         MOV    DS， AX
         MOV    ES， AX
         MOV    CX， 8
         LEA    SI， STRG
         LEA    DI， STRG+4
         CMP    SI， DI
         JA     DOWN
         JB     UP
         JMP    EXIT
UP：     STD
         MOV    AX， CX
         DEC    AX
         ADD    SI， AX
         ADD    DI， AX
         JMP    TRANS
DOWN：   CLD
TRANS：  REP    MOVSB
EXIT：   MOV    AX， 4C00H
         INT    21H
MAIN     ENDP
```

Example： ASCII→Binary

```
DSEG    SEGMENT
ASCD    DB    ‘3756’
BIND    DW     0
MULD    DW     1
DSEG    ENDS
```

```
MAIN    PROC    FAR
START： PUSH    DS
        MOV     AX， 0
        PUSH    AX
        MOV     AX， DSEG
        MOV     DS， AX
        MOV     CX， 10
        LEA     SI， ASCD
        MOV     BX， 4     ;计数+指针
NEXT：  MOV     AL， [SI+BX-1]
        AND     AX， 000FH
        MUL     MULD
        ADD     BIND， AX
        MOV     AX， MULD
        MUL     CX
        MOV     MULD， AX
        DEC     BX
        JNZ     NEXT
        RET
MAIN    ENDP
```

Example：查找字符.

```
DSEG        SEGMENT
X1          DB  100 DUP（？）
X2          DB  ‘FOUND$’
X3          DB  ‘NOT  FOUND$’
DSEG        ENDS
```

```
MAIN      PROC      FAR
START：   MOV       AX，DSEG
          MOV       DS，AX
          MOV       AH，1
          INT       21H
          MOV       DI，OFFSET  X1
          MOV       CX，100
AGAIN：   CMP       AL，[DI]
          JE        FOUND
          INC       DI
          LOOP      AGAIN
          MOV       DX，OFFSET  X3
          MOV       AH，9
          INT       21H
          JMP       EXIT
FOUND：   MOV       DX，OFFSET  X2
          MOV       AH，9
          INT       21H
EXIT：    MOV       AX，4C00H
          INT       21H
MAIN      ENDP
```

Example：查找最大值.

```
DSEG  SEGMENT
MAX   DB  ?
NUM   DB  -1,2,10,-128,6
COUNT EQU  $-NUM
DSEG  ENDS
```

```
MAIN     PROC    FAR
         MOV     AX，DSEG
         MOV     DS，AX
         MOV     CX，COUNT-1
         MOV     SI，OFFSET NUM
         MOV     AL，[SI]
NEXT1:   INC     SI
         CMP     AL，[SI]
         JG      NEXT
         MOV     AL，[SI]
NEXT:    LOOP    NEXT1
         MOV     MAX，AL
         MOV   AX，4C00H
         INT   21H
MAIN     ENDP
```

Example：升序排序.

```
DSEG    SEGMENT
BUF     DB   65，12，13，－39
DSEG    ENDS
```

```
MAIN    PROC    FAR
        MOV     AX，DSEG
        MOV     DS，AX
        MOV     CX，4
        DEC     CX
        MOV     DX，CX
L1:     MOV     CX，DX
        MOV     BX，0
L2:     MOV     AL，[BUF+BX]
        CMP     AL，[BUF+BX+1]
        JLE     CONT
        XCHG    AL，[BUF+BX+1]
        MOV     [BUF+BX]，AL
CONT:   INC     BX
        LOOP    L2
        DEC     DX
        JNZ     L1
        MOV     AX，4C00H
        INT     21H
MAIN    ENDP
```

Example：

统计数字、字母、其它字符个数.

```
DSEG    SEGMENT
BUF     DB  80
        DB  ?
        DB  80 DUP(?)
STR     DB  0AH，0DH
        DB 'Enter the ',
        DB 'string:$'
STR1    DB  0AH，0DH
        DB 'Total number:$'
STR2    DB 0AH，0DH
        DB 'Total alphabet:$'
STR3    DB 0AH，0DH
        DB 'Special '
        DB ' character:$'
NUM     DB  ?
ALPHA   DB  ?
SPC     DB  ?
DSEG    ENDS
```

```
MAIN    PROC   FAR
        MOV    AX，DSEG
        MOV    DS，AX
        MOV    AH，9
        LEA    DX，STR
        INT    21H
        LEA    DX，BUF
        MOV    AH，10
        INT    21H
        MOV    CL，BUF+1
        MOV    CH，0
        MOV    BX，2
        MOV    DX，0
LP:     MOV    AH，BUF[BX]
        CMP    AH，30H
        JB     NEXT
        CMP    AH，39H
        JA     ABCS
        INC    DH ;numbers
        JMP    NEXT
ABCS:   CMP    AH，41H
        JB     NEXT
        CMP    AH，5AH
        JA     ABCB
        INC    DL ;alphabets
        JMP    NEXT
```

```
ABCB:   CMP    AH，61H
        JB     NEXT
        CMP    AH，7AH
        JA     NEXT
        INC    DL;alphabets
NEXT:   INC    BX
        LOOP   LP
        MOV    NUM，DH
        MOV    ALPHA，DL
        MOV    AH，BUF+1
        SUB    AH，DH
        SUB    AH，DL
        MOV    SPC，AH
        MOV    AH，9
        LEA    DX，STR1
        INT    21H
        MOV    BL，NUM
        CALL   DISPBL16
        MOV    AH，9
        LEA    DX，STR2
        INT    21H
        MOV    BL，ALPHA
        CALL   DISPBL16
        MOV    AX，4C00H
        INT    21H
MAIN    ENDP
```

Example：回文判断.

```
DSEG   SEGMENT
M1     DB  10,13, ‘ENTER THE STRING: $’
M2     DB  10,13,’STRING IS PALINDROME$’
M3     DB  10,13,’ STRING IS NOT PALINDROME:$’
BUFF   DB  80
       DB  ？
       DB  80 DUP(0)
DSEG   ENDS

MAIN       PROC   FAR
           MOV    AX，DSEG
           MOV    DS，AX
           MOV    AH，9
           MOV    DX，OFFSET M1
           INT    21H
           MOV    AH，0AH
           MOV    DX，OFFSET BUFF
           INT    21H
           MOV    BX，OFFSET BUFF+2
           MOV    CH，0
           MOV    CL，BUFF+1
           MOV    DI，CX
           DEC    DI
           SAR    CL，1
           MOV    SI，0
BACK:      MOV    AL，[BX+DI]
           MOV    AH，[BX+SI]
           CMP    AL，AH
           JNZ    LAST
           DEC    DI
           INC    SI
           DEC    CL
           JNZ    BACK
           LEA    DX，M2
           MOV    AH，9
           INT    21H
           JMP    EXIT
LAST:      LEA    DX，M3
           MOV    AH，9
           INT    21H
EXIT:      MOV    AX，4C00H
           INT    21H
MAIN       ENDP
```