
题目名称 实验一 汇编语言程序结构

题目关键字 源程序结构 debug

题目录入时间 2013-3-13 22:45:26

一、实验目的

1. 掌握汇编语言程序上机过程。
2. 掌握汇编语言结构。
3. 了解汇编程序、连接程序、DOS系统装入和返回功能。
4. 掌握用DEBUG调试汇编语言程序的方法

二、实验要求

1. 编写完整的汇编语言程序：
 - (1) 逆序输出字符串“BASED ADDRESSING”。
 - (2) 从键盘上输入2个一位的正数，求出它们的和（假设和不超过1位）。
 - (3) 试编写一段程序，要求在长度为100H字节的数组中，找出大于42H的无符号数的个数并存入字单元UP中；找出小于42H的无符号数的个数并存入字单元DOWN中。
2. 在DEBUG下查看程序执行的过程，以及变量值的变化。
3. 演示程序运行结果，请指导教师检查。

三、思考

1. 汇编语言程序中语句END后的标号作用是什么？
2. 采用标准序结构编制源程序，用ret结束程序运行返回操作系统时，可否将语句main proc far中的far属性去掉，为什么？
3. 一个段的大小为多少？一定是64KB吗？如何在DEBUG下查看程序在内存的存放情况？

四、提交内容

将各个源程序代码、exe文件压缩为RAR后提交到作业系统。

题目创建人

题目注释

1.1.asm

```
DATAS SEGMENT
    String db "BASED ADDRESSING"
    count dw $-String
DATAS ENDS

STACKS SEGMENT

STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
    MOV AX,DATAS
    MOV DS,AX
    mov cx,count    ;循环次数
    lea si,count-1  ;将最后一个字符 'G' 的地址给 si
L:
    mov ah,2
    mov dl,[si]
    dec si
    int 21h
    loop L
    MOV AH,4CH
    INT 21H
CODES ENDS
    END START
```

1.2.asm

```
DATAS SEGMENT
    x db "x=$"
    y db "y=$"
    z db "x+y=$"
DATAS ENDS

STACKS SEGMENT

STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
    MOV AX,DATAS
    MOV DS,AX
```

```

mov ah,9      ;显示"x="
lea dx,x
int 21h

mov ah,1      ;输入 X
int 21h
sub al,30h
mov cl,al     ;将 X 放在 CL

mov ah,9      ;显示"y="
lea dx,y
int 21h

mov ah,1      ;输入 Y
int 21h
sub al,30h
add cl,al     ;两数字相加

mov ah,9      ;显示"z="
lea dx,z
int 21h

mov ah,2      ;显示结果
mov di,cl
add di,30h
int 21h

MOV AH,4CH
INT 21H
CODES ENDS
END START

```

1.3.asm

```

DATAS SEGMENT
    array db 128 dup(41h,43h)
    up     dw 0
    down   dw 0
DATAS ENDS

STACKS SEGMENT
STACKS ENDS

```

```

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
    MOV AX,DATAS
    MOV DS,AX
    lea di,array
    mov cx,256
L:
    mov al,[di]
    cmp al,42h
    jl  L1
    ja  L2
L1:
    inc down
    jmp L3
L2:
    inc up
    jmp L3
L3:
    inc di
    loop L

    MOV AH,4CH
    INT 21H
CODES ENDS
END START

```

题目名称 实验二 程序结构与简单程序设计

题目关键字 简单程序设计

题目录入时间 2013-4-1 22:45:55

一、实验目的

1. 掌握汇编语言源程序结构。
2. 掌握汇编语言调试方法。

题目内容

二、实验要求

1. 编写完整的汇编语言程序：
 - (1) 试编写一段程序，要求对键盘输入的小写字母用大写字母显

示出来。

- (2) 编写程序，从键盘接收一个小写字母，然后找出它的前导字母和后续字母，再按顺序显示这三个字符。
 - (3) 试编写一段程序，要求比较两个字符串 `string1` 和 `string2` 所含字符是否相等，如相等则显示 “MATCH”，若不相同则显示 “NO MATCH”。
 - (4) 编写程序，将一个包含有 20 个数据的数组 `M` 分成 2 个数组：正数数组 `P` 和负数数组 `N`，并分别把这两个数组中数据的个数显示出来。
2. 在 DEBUG 下查看程序执行的过程，以及变量值的变化。
3. 演示程序运行结果，请指导教师检查。

三、思考

- 1 对输入的非字母字符如何处理？
2. 字母 “a” 和 “z” 边界字符的处理方法是什么？
3. 显示结果为多位数的时候如何输出？

四、提交内容

将各个源程序代码、.exe 文件，思考题的 WORD 文档压缩为 RAR 后提交到作业系统。

题目创建人

题目注释

```

2.1.asm
DATAS SEGMENT
    crlf db 0ah,0dh,'$'
DATAS ENDS

STACKS SEGMENT
STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
    MOV AX,DATAS
    MOV DS,AX

    mov ah,1    ;输入数据会 AL
    int 21h

```

```

push ax    ;将 AX 存进栈里

lea dx,crif;换行
mov ah,9
int 21h

pop dx     ;栈里数据放进 DX，即将原先 AX 放进 DX
sub di,20h ;小写变大写要减去 20h

mov ah,2   ;输出
int 21h

MOV AH,4CH
INT 21H
CODES ENDS
END START

```

2.2.asm

```

DATAS SEGMENT
    String db 'abcdefghijklmnopqrstuvwxyz'
DATAS ENDS

STACKS SEGMENT
STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
    MOV AX,DATAS
    MOV DS,AX

    lea di,String

    mov ah,1   ;输入字母
    int 21h

L:
    inc di     ;先 DI 自加 1，第一次就是 'a' 的地址，不是 'z'
    cmp al,[di] ;直到找到字母和输入的相匹配
    jnz L

    dec di     ;推前成为前端字母的地址

```

```

    mov cx,3   ;循环次数 3

L2:
    mov di,[di]
    inc di
    mov ah,2
    int 21h
    loop L2

```

```

MOV AH,4CH
INT 21H
CODES ENDS
END START

```

2.3.asm

```

DATAS SEGMENT
    String1 db "this is a program"
    count1 dw $-String1
    String2 db "this is a program"
    count2 dw $-String2
    answer1 db 'match','$'
    answer2 db 'no match','$'
DATAS ENDS

STACKS SEGMENT
STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
    MOV AX,DATAS
    MOV DS,AX

    mov ax,count1
    mov bx,count2
    cmp ax,bx    ;比较两个字符串的长度，如果不等就是不匹配
    jnz next

    mov cx,count1 ;字符串长度
    lea si,string1
    lea di,string2

L:
    mov al,[si]
    cmp al,[di]  ;比较两个字符串的字符，如果不等就跳

```

```

        jnz next
        inc si
        inc di
        loop L

        lea dx,answer1 ;能到这里说明都匹配
        mov ah,9
        int 21h
        jmp finally ;跳去 finally,避免顺序执行了 next

next:
        lea dx,answer2
        mov ah,9
        int 21h

finally:
        MOV AH,4CH
        INT 21H
CODES ENDS
END START

```

2.4.asm

```

DATAS SEGMENT
    M db 5 dup(-1,1,2,3)
    P dw 0
    N dw 0
DATAS ENDS

STACKS SEGMENT

STACKS ENDS

CODES SEGMENT
    ASSUME CS,CODES,DS,DATAS,SS,STACKS
START:
    MOV AX,DATAS
    MOV DS,AX

    lea si,M
    mov cx,20 ;循环次数
L:
    mov al,[si]
    cmp al,0 ;和 0 比较判断正负
    jl L1

```

```

        jg L2
L1:
    inc N
    jmp L3
L2:
    inc P
    jmp L3
L3:
    inc si
    loop L

    mov ax,P ;输出 P
    mov bl,10 ;设置除数 10
    div bl
    mov bh,ah ;将 ah(余数放在 bh 保留)
    mov dl,al ;将商显示
    add dl,30h
    mov ah,2
    int 21h
    mov di,bh ;将余数显示
    add di,30h
    mov ah,2
    int 21h

    mov ax,N ;输出 N
    mov bl,10
    div bl
    mov bh,ah
    mov dl,al
    add dl,30h
    mov ah,2
    int 21h
    mov di,bh
    add di,30h
    mov ah,2
    int 21h

    MOV AH,4CH
    INT 21H
CODES ENDS
END START

```

题目名称	实验三 分支程序设计
题目关键字	分支结构
题目录入时间	2013-4-14 20:14:57
题目内容	<p>一、实验目的</p> <ol style="list-style-type: none">1. 掌握汇编语言源程序结构。2. 掌握汇编语言分支结构程序设计方法。 <p>二、实验要求</p> <p>1. 编写完整的汇编语言程序：</p> <ol style="list-style-type: none">(1) 试编写一段程序，求出首地址为 DATA 的 100D 字节数组中的最小偶数，并把它存放在 AX 中。(2) 编写一段程序，从键盘接收一个四位的十六进制数，并在终端上显示与它等值的二进制数。(3) 键盘上输入一系列以 \$ 为结束符的字符串，然后对其中的非数字字符计数，并显示计数结果。(4) 有一个首地址为 non 的 100 个字的数组，试编制程序删除数组中所有为零的项，并将后续项向前压缩，最后将数组的剩余部分补上零。 <p>2. 在 DEBUG 下查看程序执行的过程，以及变量值的变化。</p> <p>3. 演示程序运行结果，请指导教师检查。</p> <p>三、思考</p> <ol style="list-style-type: none">1. 求极值时程序初始化应该注意什么？2. 如何从键盘上输入多位数？3. 第 (4) 小题有多少种做法？ <p>四、提交内容</p> <p>将各个源程序代码、.exe 文件以及思考题压缩为 RAR 后提交到作业系统。提交时请填写自评内容。</p>
题目创建人	
题目注释	

```
3.1.asm
DATAS SEGMENT
    DATA dw 1 dup(1,2,-1,4,-2,-6)
    count dw ($-DATA)/2
DATAS ENDS

STACKS SEGMENT

STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
    MOV AX,DATAS
    MOV DS,AX

    mov cx,count
    lea si,DATA-2
L1:      ;将第一个偶数存在 AX
    add si,2
    mov ax,[si]
    dec cx
    test al,01h
    jnz L1

    L2:      ;找到另外一个偶数
    add si,2
    mov bx,[si]
    cmp cx,0
    jr  L4
    dec cx
    test bl,01h
    jnz L2

    cmp ax,bx      ;比较两个偶数，如果 ax>bx 那么跳到 L3
    jg  L3
    jmp L2

L3:      mov ax,bx
    jmp L2

L4:      MOV AH,4CH
    INT 21H
CODES ENDS
```

```

END START

3.2.asm
DATAS SEGMENT
    NUM    dw 0
    crlf db 0dh,0ah,'$'
DATAS ENDS

STACKS SEGMENT
STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
    MOV AX,DATAS
    MOV DS,AX
    mov dx,0
    mov cx,4

L:
    mov ah,1
    int 21h
    cmp al,40h
    jg L1
    jl L2

L1:
    sub al,07h      ;大写字母多减 7

L2:
    sub al,30h      ;变成实际数字

    push ax          ;原来数字*16+新接受数字
    mov ax,NUM
    mov bx,16
    mul bx
    pop bx
    adc al,bl
    mov NUM,ax
    loop L           ;循环 4 次

    lea dx,crlf
    mov ah,9
    int 21h

    mov cx,16

```

```

    mov bx,NUM
L3:
    mov dl,31h
    sal bx,1        ;左移 1 位, 改变 CF
    jc L4
    sub dl,1        ;CF 为 0, 所以 DL 减去 1, 变成 0

L4:
    mov ah,2
    int 21h
    loop L3
    MOV AH,4CH
    INT 21H
CODES ENDS
END START

```

```

3.3.asm
DATAS SEGMENT
    String db 128?,128 dup('$')
    crlf db 0dh,0ah,24h
    count dw 0
    num dw 0
DATAS ENDS

STACKS SEGMENT
STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
    MOV AX,DATAS
    MOV DS,AX

    lea dx,String    ;接收字符串
    mov ah,0ah
    int 21h

    lea dx,crlf      ;换行
    mov ah,9
    int 21h

    lea si,String-1  ;cx 存放着字符串个数
    mov cl,[si]

```



```

mov ch,0

lea di,String+2  ;从实际字符串开始
L1:
mov al,[di]
inc di
cmp al,30h
jl L2
cmp al,39h
jg L2
dec num          ;如果是数字，先减 num 后加 num,将其抵消
L2:
inc num
loop L1

mov cx,0
mov ax,num
L3:
mov bl,10
div bl
push ax
inc cx           ;记录余数个数
mov ah,0
cmp al,0
jnz L3

L4:
;将余数输出
pop dx
mov dl,dh
add dl,30h
mov ah,2
int 21h
loop L4
MOV AH,4CH
INT 21H
CODES ENDS
END START

```

```

3.4.asm
DATAS SEGMENT
    mem dw 0,1,2,0,0,3,2,0,5,0,0,1
    count dw ($-mem)/2
DATAS ENDS

STACKS SEGMENT
STACKS ENDS

CODES SEGMENT
ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
MOV AX,DATAS
MOV DS,AX

lea di,mem-2 ;先减 2 再恢复
mov cx,count
L:
cmp cx,0
jz next
add di,2
mov ax,[di]
dec cx
cmp ax,0
jnz L

mov si,di
push cx
L2:
cmp cx,0
jz next
add si,2
mov ax,[si]
dec cx
cmp ax,0
jz L2

mov [di],ax ;复制过去
mov word ptr [si],0 ;复制后自己变 0
pop cx
jmp L
next:
MOV AH,4CH

```

```
INT 21H
CODES ENDS
END START
```

题目名称 实验四 循环程序设计1

题目关键字 循环结构程序

题目录入时间 2013-4-22 17:28:16

一、实验目的
1. 掌握汇编语言源程序结构。
2. 掌握汇编语言循环结构程序设计方法。
二、实验要求
1. 编写完整的汇编语言程序：
(1) 从键盘上输入一串字符（用回车键结束，使用10号功能调用。）放在STRING中，试编制一个程序测试字符串中是否存在数字。如有，则把CL的第6位置1，否则将该位置置0。
(2) 在首地址为DATA的字数组中，存放了100H个16位补码数，试编写一个程序，求出它们的平均值放在AX寄存器中；并求出数组中有多少个数小于此平均值，将结果放在BX寄存器中。
(3) 已知数组A包含15个互不相等的整数，数组B包含20个互不相等的整数。试编制一个程序，把既在A中又在B中出现的整数存放于数组C1中。
(4) 简化的歌德巴赫猜想：任何一个大于6的偶数均可以表示为两个素数之和。现从键盘输入一个0~65535之间的无符号整数，试验证歌德巴赫猜想。
如输入数12, 输出：
12=5+7
如输入数20, 输出：
20=3+17
20=7+13
如输入数7, 输出：
Must be even
如输入数3, 输出：
Must be greater than or equal to 6

2. 演示程序运行结果，请指导教师检查。
三、思考
从键盘上输入一串字符时，数据缓冲区如何定义？

题目创建人

题目注释

```
4.1.asm
DATAS SEGMENT
    String db 128,7,128 dup('$')
DATAS ENDS

STACKS SEGMENT

STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
    MOV AX,DATAS
    MOV DS,AX

    lea dx,String
    mov ah,0ah
    int 21h

    lea si,String+1 ;字符串个数地址
    mov cl,[si]
    mov ch,0
    inc si           ;第一个字符地址
L:
    mov al,[si]
    cmp al,30h
    jl L1
    cmp al,39h
    jg L1
    or cl,00001000b ;CL 第五位置 1
    jmp next
L1:
    inc si
```

```

        loop L
    next
    MOV AH,4CH
    INT 21H
CODES ENDS
    END START

4.2.asm
DATAS SEGMENT
    DATA    dw 1,2,3,4,5,6,7,8,9,10
    count dw ($-DATA)/2
DATAS ENDS

STACKS SEGMENT

STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
    MOV AX,DATAS
    MOV DS,AX
    mov dx,0
    mov ax,0
    mov cx,count
    lea si,DATA
L:
    mov bx,[si]
    add ax,bx
    add si,2
    loop L

    mov bx,count
    div bx
    lea si,DATA
    mov cx,count
    mov bx,0

L2:
    mov dx,[si]
    cmp dx,ax
    jnl L3
    inc bx

L3:

```

```

    add si,2
    loop L2

    MOV AH,4CH
    INT 21H
CODES ENDS
    END START

4.3.asm
DATAS SEGMENT
    A        db 1,3,5,7,9,11,13,15,17,19,21,23,25,27,29
    count1 dw $-A
    B        db 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
    count2 dw $-B
    C1       db 15 dup(?)
DATAS ENDS

STACKS SEGMENT

STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
    MOV AX,DATAS
    MOV DS,AX
    lea di,C1
    lea si,A
    mov cx,count1
    mov bx,0
L:
    mov al,[si]
    push cx
    mov al,A[bx]
    inc bx
    mov cx,count2
    lea si,B
L2:
    mov dl,[si]
    inc si
    cmp al,dl
    jz L3
    loop L2
    jmp L4

```

```

L3:                ;找到就赋给 C1
    mov [di],dl
    inc di
L4:
    pop cx
    loop L

    MOV AH,4CH
    INT 21H
CODES ENDS
END START

```

4.4.asm

这道题目不难，关键是很烦
分几步走。1.取多位十进制数 2.判断质数 3.显示十进制数 4.将每个结果循环输出
有几点要注意
:1.因为数可以用 16 位，所以判断大小记得 JA,JB，而不是 JG, JL
:2.用 BX 作乘法，除法，要留意住 DX
:3.看清楚哪个变量会受到影响，比如 AX 经常被使用，容易数据丢失
;

```

DATAS SEGMENT
    sentence1 db "Must be even$"
    sentence2 db "Must be greater than or equal to 6$"
    crlf      db 0dh,0ah,"$"
    num       dw 0
    temp      dw 0
    temp2     dw 0
    key       dw 0
DATAS ENDS

STACKS SEGMENT

STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
    MOV AX,DATAS
    MOV DS,AX
L:    ;从键盘取得数字，存进 num
    mov ah,1
    int 21h
    cmp al,0dh
    jz  L2

```

```

    sub al,30h
    mov cl,al
    mov ch,0
    mov ax,num
    mov bx,10
    mul bx
    add ax,cx
    mov num,ax
    jmp L
L2:   ;换行
    lea dx,crlf
    mov ah,9
    int 21h

    mov ax,num
    cmp ax,6
    jb  error1
    sar ax,1
    jc  error2
    jmp next
error1: ;必须是偶数
    lea dx,sentence2
    mov ah,9
    int 21h
    mov ah,4ch
    int 21h
error2: ;大于等于 6 才行
    lea dx,sentence1
    mov ah,9
    int 21h
    mov ah,4ch
    int 21h
next:
    mov cx,num
    sub cx,3
    mov bx,1
ba:
    inc bx
    mov ax,num
    sub ax,bx
    mov temp,ax
    call qiuhezhu ;验证是否质数
    cmp key,1     ;key=1 表示是合数
    jz  finally

```

```

mov temp2,ax
mov ax,num
sub ax,temp
xchg ax,temp
call qiuzhishu ;验证另一个是否质数
cmp key,1
jz finally

mov ax,num ;下面这一段代码
call xianshi ;是为了显示
mov ah,2 ;sum = x + y
mov dl,'=' ;这种格式
int 21h ;将前面求到的总数
mov ax,temp2 ;和两个质数分别显示到屏幕
push temp
call xianshi
mov dl,'+'
mov ah,2
int 21h
pop ax
call xianshi

lea dx,crif ;换行
mov ah,9
int 21h

finally:
loop ba
mov ah,4ch
int 21h

qiuzhishu: ;这个子程序都是用来求质数的
push ax
push bx
push cx
push dx
mov bx,2
mov cx,temp
sub cx,2
next0:
mov ax,temp
cmp ax,3

```

```

jb next2 ;2 是质数
mov dx,0
div bx
inc bx
cmp dx,0
jz next1
loop next0
jmp next2
next1:
mov key,1
jmp next3
next2:
mov key,0
next3:
pop dx
pop cx
pop bx
pop ax
ret

xianshi: ;显示多位十进制数字
push ax
push bx
push cx
push dx
mov dx,0
mov cx,0
mov bx,10
L3:
div bx
push dx
mov dx,0
inc cx
cmp ax,0
jnz L3
L4:
pop dx
add dl,30h
mov ah,2
int 21h
loop L4
pop dx
pop cx
pop bx

```

```
pop ax
ret
CODES ENDS
END START
```

题目名称 实验五 循环程序设计2

题目关键字 循环结构

题目录入时间 2013-4-26 23:17:07

一、实验目的

1. 掌握汇编语言源程序结构。
2. 掌握汇编语言循环结构程序设计方法。

二、实验要求

1. 编写完整的汇编语言程序：

(1) 设在 A、B 和 D 字单元中分别存放着三个数。若三个数都不是 0，则求出三个数的和并存放在 S 单元中；若其中有一个数为 0，则把其他两个单元也清零。请编写此程序。

题目内容 (2) 试编写一个程序，要求比较数组 ARRAY 中的三个 16 位补码数，并根据比较结果在终端上显示如下信息：

- 如果三个数都不相等则显示 0；
- 如果三个数有两个相等则显示 1；
- 如果三个数都相等则显示 2。

(3) 根据欧几里德辗转相除法求两个正整数 M、N 的最大公约数，步骤为：

- ① 求 M/N 的余数 R；
- ② 如果 R=0，则 N 为最大公约数，退出循环；
- ③ 用 N 更新 M 的值，用 R 更新 N 的值；
- ④ 重复 ①~③ 步。

(4) Link 指向一个字线性表，其中的首单元保存线性表的长度，编写一个程序，将该线性表中内容为 0FFFFH 的项删除。

2. 演示程序运行结果，请指导教师检查。

四、提交内容

将各个源程序代码、.exe 文件压缩为 RAR 后提交到作业系统。在“自评内容”栏中写出完成作业的情况以及在做作业过程中出现的问题和解决的方法。

题目创建人

题目注释

```
5.1.asm
DATAS SEGMENT
    A dw 2
    B dw 1
    D dw 2
    S dw ?
DATAS ENDS

STACKS SEGMENT
STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
    MOV AX,DATAS
    MOV DS,AX

    lea si,A
    mov bx,0
    mov cx,3

L:
    mov ax,[si]
    cmp ax,0
    jz aoo
    add bx,ax
    add si,2
    loop L
    jmp aoo2
aoo:
    ;三个单元置 0
```

```

mov ax,0
push ax
push ax
push ax
pop A
pop B
pop D
a002:      ;三个数之和
mov dl,bl
add dl,30h
mov ah,2
int 21h

MOV AH,4CH
INT 21H
CODES ENDS
END START

```

5.2.asm

```

DATAS SEGMENT
array dw 1,-1,-1
DATAS ENDS

STACKS SEGMENT

STACKS ENDS

CODES SEGMENT
ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
MOV AX,DATAS
MOV DS,AX

mov cx,0      ;从 0 开始计数，遇到相等就加 1
lea si,array
add si,2
mov ax,[si]   ;记住第二个数，分别和第一个和第三个比较
mov bx,[si+2]
cmp ax,bx
jnz L1
inc cx
L1:

```

```

mov bx,[si+2]
cmp ax,bx
jnz L2
inc cx

L2:
mov dl,cl
add dl,30h
mov ah,2
int 21h

MOV AH,4CH
INT 21H
CODES ENDS
END START

```

5.3.asm

```

DATAS SEGMENT
temp1 dw 0
temp2 dw 0
crif db 0ah,0dh,'$'
DATAS ENDS

STACKS SEGMENT

STACKS ENDS

CODES SEGMENT
ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
MOV AX,DATAS
MOV DS,AX
L:      ;从键盘取得数字，存进 temp1
mov ah,1
int 21h
cmp al,0dh
jz L2
sub al,30h
mov cl,al
mov ch,0
mov ax,temp1
mov bx,10
mul bx
add ax,cx

```

```

mov temp1,ax
jmp L

L2:
lea dx,crif
mov ah,9
int 21h
L3:      ;从键盘取得数字，存进 temp2
mov ah,1
int 21h
cmp al,0dh
jz  L4
sub al,30h
mov cl,al
mov ch,0
mov ax,temp2
mov bx,10
mul bx
add ax,cx
mov temp2,ax
jmp L3
L4:
mov dx,0
mov ax,temp1
mov bx,temp2
continue:
div bx
cmp dx,0
jz  next
mov ax,bx
mov bx,dx
mov dx,0
jmp continue
next:
lea dx,crif      ;换行
mov ah,9
int 21h

mov ax,bx
call xianshi
MOV AH,4CH
INT 21H
xianshi:      ;显示多位十进制数字
push ax

```

```

push bx
push cx
push dx
mov dx,0
mov cx,0
mov bx,10
A1:
div bx
push dx
mov dx,0
inc cx
cmp ax,0
jnz A1
A2:
pop dx
add dl,30h
mov ah,2
int 21h
loop L2
pop dx
pop cx
pop bx
pop ax
ret
CODES ENDS
END START

```

```

5.4.asm
DATAS SEGMENT
Link    dw 9,3,0fffh,2,1,0fffh,1,2,0fffh
finial dw $
DATAS ENDS

STACKS SEGMENT
STACKS ENDS

CODES SEGMENT
ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
MOV AX,DATAS

```



```
MOV DS,AX
lea si,Link ;首地址
add si,2
L2: mov ax,[si]
    cmp ax,0ffffh
    jnz L3
    jz L41
L3: add si,2
    jmp L2
L41: mov di,si
L42: ;找到 0FFFFH 项，和不是这个项数交换
    cmp di,finial
    jz next
    mov bx,[di]
    cmp bx,0ffffh
    jz L5
    mov [si],bx
    mov bx,0ffffh
    mov [di],bx
    jmp L3
L5: add di,2
    jmp L42

next:
MOV AH,4CH
INT 21H
CODES ENDS
END START
```

题目名称 实验六 结构化程序设计

题目关键字 结构化程序 简化段结构

题目录入时间 2013-5-3 21:42:56

一、实验目的

1. 掌握汇编语言简化段结构。
2. 掌握汇编语言综合程序设计方法。

二、实验要求

1. 编写完整的汇编语言程序：

- (1) 从键盘输入一系列字符（以回车符结束），并按字母、数字及其他字符分类计数，最后显示这三类的计数结果（用简化段源程序结构）。
- (2) 已定义了两个整数变量 A 和 B，试编写程序完成下列功能（使用简化段源程序格式）：
 - 若两个数中有一个是奇数，则将奇数存入 A 中，偶数存入 B 中；
 - 若两个数均为奇数，则将两数均加 1 后存回原变量；
 - 若两个数均为偶数，则两个变量均不改变。
- (3) 用减奇数次数的方法，求一个数的近似平方根，这个平方根是一个整数。如求 17 的平方根，可以用 17 相继减去奇数 1、3、5、7、…，当结果为负数时停止，即：
 $17-1-3-5-7-9<0$

题目内容

可以看出，17 在减去 6 次奇数后结果变为负数，可以近似认为 17 的平方根在 4 与 5 之间，计算 NUM 的平方根，如果 NUM=17，则 AMS 中保存结果 4。

- (4) 以下面形式定义一个长整数，其所占用字节数由 N 得到，比如 128 位的数 5746352413DE89674523BC9A78563412H 定义成：

```
NUM    DB    12H,34H,56H,78H,9AH,0BCH,23H,45H
        DB    67H,89H,0DEH,13H,24H,35H,46H,57H
N       DB    $-NUM
```

试编写一个程序，实现对 N 字节的 NUM 的值求补（类似 NEG 的操作），并把求补后的值仍保存在 NUM 中。

2. 演示程序运行结果，请指导教师检查。

题目创建人

题目注释

6.1.asm

```
.model small
.data
letter dw 0 ;字母
digital dw 0 ;数字
other dw 0 ;其他字符
clr db 0dh,0ah,'$'
.code
```

```

start:
    mov ax,@data
    mov ds,ax
    jmp L

upl:
    inc letter
    jmp L

upd:
    inc digital
    jmp L

upo:
    inc other
L:
    mov ah,1
    int 21h
    cmp al,0dh
    jr next
    cmp al,30h
    jl upo
    cmp al,3ah
    jl upd
    cmp al,41h
    jl upo
    cmp al,5bh
    jl upl
    cmp al,61h
    jl upo
    cmp al,7bh
    jl upl
    cmp al,7bh
    ja upo
    jmp L

next:
    lea si,letter ;用 si 控制地址
    mov bx,0
    mov cx,3 ;循环三次, 分别输出字母, 数字, 和其他字符的个数

next2:
    lea dx,cir
    mov ah,9
    int 21h
    mov dx,[si+bx]
    add si,2
    add di,30h
    mov ah,2

```

```

int 21h
loop next2

mov ah,4ch
int 21h
end start

```

6.2.asm

```

.model small
.data
    A dw 2
    B dw 3
.code
start:
    mov ax,@data
    mov ds,ax

    mov ax,A
    rcr ax,1
    jnc L2 ;判断 A 为偶数
    mov bx,B
    rcr bx,1
    jnc next ;两者为偶数, 均没变化
    inc A ;两者为奇数, 分别加 1
    inc B

L2:
    mov bx,B
    push bx
    rcr bx,1
    jnc next ;B 也是偶数, 两者不变
    push A ;A 为偶数, B 为奇数, 两者交换
    push B
    pop A
    pop B

next:
    mov ah,4ch
    int 21h
    end start

```

```
6.3.asm
DATAS SEGMENT
    digital dw 18
DATAS ENDS

STACKS SEGMENT

STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
    MOV AX,DATAS
    MOV DS,AX

    mov ax,digital
    mov bx,1
    mov dx,0
L:
    sub ax,bx
    cmp ax,0
    jl next
    inc dx
    add bx,2
    jmp L
next:
    add di,30h
    mov ah,2
    int 21h
    MOV AH,4CH
    INT 21H
CODES ENDS
END START
```

```
6.4.asm
DATAS SEGMENT
    NUM      DB  12H,34H,56H,78H,9AH,0BCH,23H,45H
              DB  67H,89H,0DEH,13H,24H,35H,46H,57H
    N        DW  $-NUM
DATAS ENDS

STACKS SEGMENT
```

```
STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
    MOV AX,DATAS
    MOV DS,AX

    lea si,NUM
    mov cx,N
    stc                                ;1 在这里
L:
    mov al,[si]
    not al                            求补即是取反+1
    adc al,0
    mov [si],al
    inc si
    loop L

    MOV AH,4CH
    INT 21H
CODES ENDS
END START
```

题目名称	实验七 子程序设计1
题目关键字	子程序
题目录入时间	2013-5-7 10:16:29
一、实验目的	
题目内容	1. 掌握汇编语言完整段和简化段源程序结构。
	2. 掌握汇编语言综合程序设计方法。
二、实验要求	
1. 编写完整的汇编语言程序：	
(1). 写一段子程序 SKIPLINES，完成输出空行的功能。空行的行数由	

用户在主程序中通过键盘输入，并将行数放在 AX 寄存器中。

(2). 设有 10 个学生的成绩分别是 76, 69, 84, 90, 73, 88, 99, 63, 100 和 80 分。试编制一个子程序统计 60-69 分, 70-79 分, 80-89 分, 90-99 分和 100 分的人数, 并分别放到 S6, S7, S8, S9 和 S10 单元中。

(3) 编写一个程序, 计算 $1 + 2 + 3 + 4 + \dots + N$ 的累加和, 并把累加和以十进制形式显示出来。(不能使用公式计算“累加和= $N \times (N+1) / 2$ ”, 必须使用程序实现循环累加来计算)。

(4) 根据《中华人民共和国国家标准 GB 11643-1999》中有关公民身份号码的规定, 公民身份号码是特征组合码, 由十七位数字本体码和一位数字校验码组成。排列顺序从左至右依次为: 六位数字地址码, 八位数字出生日期码, 三位数字顺序码和一位数字校验码。顺序码的奇数分给男性, 偶数分给女性。校验码是根据前面十七位数字码, 按照 ISO 7064:1983.MOD 11-2 校验码计算出来的检验码。

下面介绍计算校验码的算法:

先引入公式: (右边最低位为第 1 位, 左边最高位为第 18 位)

$$S = \sum (A_i * W_i) \quad (i = 2, 3, \dots, 18)$$

A_i : 表示号码字符从右至左包括校验字符在内的位置序号;

W_i : 表示第 i 位置上的号码字符值;

W_i : 表示第 i 位置上的加权因子, 其数值依据公式 $W_i = 2^{(i-1) \bmod 11}$ 计算得出。下表列出身份号码中各个位置上的加权因子 W_i 数值。

i	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2
W_i	7	9	10	5	8	4	2	1	6	3	7	9	10	5	8	4	2

有了 W_i 值表后, 可以简化 S 的计算过程, 得出 S 后计算余数 Y , Y 的计算公式为: $Y = \text{mod}(S, 11)$, 再根据下表找出 Y 对应的校验码即为要求身份证号码的校验码 C 。

Y	0	1	2	3	4	5	6	7	8	9	10
校验码 C	1	0	X	9	8	7	6	5	4	3	2

编写一个程序, 完成从键盘上输入一个 17 位的身份证号 (身份证号的前 17 位, 即上面描述中的第 18 位至第 2 位), 通过计算补充第 1 位后, 将完整的身份证号显示出来。

(5) 设有一段英文, 其字符变量名为 ENG, 并以 \$ 字符结束。试编写一段程序, 查单词 SUN 在该文中出现的次数, 并输出显示出现的次数。

2. 演示程序运行结果, 请指导教师检查。

三、提交内容

将各个源程序代码、.exe 文件压缩为 RAR 后提交到作业系统。在“自评内容”栏中写出完成作业的情况以及在做作业过程中出现的问题和解决的方法。

题目创建人

题目注释

```
7.1.asm
DATAS SEGMENT
    c1rf db 0ah,0dh,'$'
    num dw 0
DATAS ENDS

STACKS SEGMENT
STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
    MOV AX,DATAS
    MOV DS,AX

    call SKIPLINES
    MOV AH,4CH
    INT 21H

SKIPLINES:
L:      ;从键盘取得数字,存进 AX
    mov ah,1
    int 21h
    cmp al,0dh
    jz  L2
    sub al,30h
    mov cl,al
    mov ch,0
    mov ax,num
    mov bx,10
```

```

        mul bx
        add ax,cx
        mov num,ax
        jmp L
L2:      mov ax,num      ;行數放 AX
        mov cx,ax      ;次數
L3:      lea dx,clrf
        mov ah,9
        int 21h
        loop L3
        ret
CODES ENDS
        END START

```

7.2.asm

```

DATAS SEGMENT
        student db 76d,69d,84d,90d,73d,88d,99d,63d,100d,80d
        count    dw 5-student
        s5       dw 0
        s6       dw 0
        s7       dw 0
        s8       dw 0
        s9       dw 0
        s10      dw 0
DATAS ENDS

STACKS SEGMENT

STACKS ENDS

CODES SEGMENT
        ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
        MOV AX,DATAS
        MOV DS,AX

        lea di,student
        mov cx,count
        call statistic

```

```

        MOV AH,4CH
        INT 21H
statistic:
        mov ax,[di]
        inc di
        cmp cx,0
        jz  next
        dec cx
        cmp al,60d
        jl  up5
        cmp al,70d
        jl  up6
        cmp al,80d
        jl  up7
        cmp al,90d
        jl  up8
        cmp al,100d
        jl  up9
        jz  up10
next:
        ret
up5:
        inc s5
        jmp statistic
up6:
        inc s6
        jmp statistic
up7:
        inc s7
        jmp statistic
up8:
        inc s8
        jmp statistic
up9:
        inc s9
        jmp statistic
up10:
        inc s10
        jmp statistic
CODES ENDS
        END START

```

7.3.asm

```
DATAS SEGMENT
    crlf db 0ah,0dh,'$'
    num   dw 0
DATAS ENDS

STACKS SEGMENT
STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
    MOV AX,DATAS
    MOV DS,AX

    call total      ;计算总数
    push ax         ;避免影响先入栈
    lea dx,crlf
    mov ah,9
    int 21h
    pop ax          ;出栈
    call xianshi
    mov ah,4ch
    int 21h

total:
L:      ;从键盘取得数字,存进 num
    mov ah,1
    int 21h
    cmp al,0dh
    jz  L2
    sub al,30h
    mov cl,al
    mov ch,0
    mov ax,num
    mov bx,10
    mul bx
    add ax,cx
    mov num,ax
    jmp L
L2:
    mov cx,num
    mov ax,0
```

```
    mov bx,1
LL:
    add ax,bx
    inc bx
    loop LL
ret

xianshi:      ;显示多位十进制数字
    push ax
    push bx
    push cx
    push dx
    mov dx,0
    mov cx,0
    mov bx,10
L3:
    div bx
    push dx
    mov dx,0
    inc cx
    cmp ax,0
    jnz L3
L4:
    pop dx
    add dl,30h
    mov ah,2
    int 21h
    loop L4
    pop dx
    pop cx
    pop bx
    pop ax
ret
CODES ENDS
END START
```

7.4.asm

```
DATAS SEGMENT
    string db 30 dup('$')
    crlf    db 0ah,0dh,'$'
    quan    db 7d,9d,10d,5d,8d,4d,4d,2d,1d,6d,3d,7d,9d,10d,5d,8d,4d,2d
    tabl    db "10x98765432"
DATAS ENDS

STACKS SEGMENT

STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
    MOV AX,DATAS
    MOV DS,AX

    lea dx,string    ;接收输入字符串
    mov ah,0ah
    int 21h

    lea dx,crlf       ;换行
    mov ah,9
    int 21h

    mov dx,0
    mov cx,17d
    lea si,string+2    ;首字符地址
    lea di,quan        ;权地址
L:
    ;计算总积
    mov ax,[si]
    sub al,30h
    mov bx,[di]
    mul bl
    add dx,ax
    inc si
    inc di
    loop L

    mov ax,dx
    mov bl,11
    div bl             ;余数 Y
```

```
    mov cx,0
    mov cl,ah
    lea si,tabl
    cmp cx,0
    jz L22

L2:
    inc si             ;去到余数 Y 对应的校验码
    loop L2
L22:
    mov bx,0
    lea di,string+2

L3:
    mov di,[di]
    cmp di,0dh
    jz finally
    inc di
    jmp L3
finally:
    ;将第 18 位号码插入原来 17 位的字符串后面
    mov di,[si]
    mov [di],di

    lea dx,string+2    ;十八位一起输出
    mov ah,9
    int 21h

    MOV AH,4CH
    INT 21H
CODES ENDS
END START
```

7.5.asm

```
DATAS SEGMENT
    ENG db "zzsusumxssxzsuxmsumxum$"
    count dw 0
DATAS ENDS

STACKS SEGMENT

STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
```

```

MOV AX,DATAS
MOV DS,AX

mov cx,0
dec cx
L:
inc cx
lea di,ENG
add di,cx
mov al,[di]
inc di
cmp al,24h
jz finally
cmp al,'s'
jnz L

mov al,[di]
inc di
cmp al,24h
jz finally
cmp al,'u'
jnz L

mov al,[di]
inc di
cmp al,24h
jz finally
cmp al,'m'
jnz L

inc count      ;三个判断都通过那么你是 sum 了
jmp L

finally:      ;显示个数
mov ax,count
mov dx,0
mov cx,0
mov bx,10
L3:
div bx
push dx
mov dx,0
inc cx
cmp ax,0

```

```

jnz L3
L4:
pop dx
add di,30h
mov ah,2
int 21h
loop L4

MOV AH,4CH
INT 21H
CODES ENDS
END START

```

题目名称 实验八 子程序设计2

题目关键字 子程序设计

题目录入时间 2013-5-13 18:19:26

一、实验目的

题目内容

1. 掌握汇编语言完整段和简化段源程序结构。
2. 掌握汇编语言综合程序设计方法。
3. 掌握文件的操作。

二、实验要求

1. 编写完整的汇编语言程序：

(1) 编写子程序嵌套结构的程序，把整数分别用二进制和八进制形式显示出来。

主程序 BANDO:把整数变量 VAL1存入堆栈，并调用子程序 PAIRS;

子程序 PAIRS:从堆栈中取出 VAL1,调用二进制显示程序 OUTBIN 显示出与其等效的二进制数，输出八个空格，调用八进制显示程序 OUTOCT 显示出与其等效的八进制数，调用输出回车及换行符的子程序。

(2) 主程序 MAIN 和过程 PROADD 在不同源文件中，要求分别使用堆栈传送参数的方法，用过程 PROADD 累加数组中的所有元素，并把和（不考虑溢出的可能性）送到指定的存储单元中去。

(3) 先从键盘上输入一串数字。将数字按照下表的密码数字进行加密，将加密后的数字在第2行显示出末。然后用解密数字对加密后的数字进行解密，并在第3行显示解密后的数字。其中显示功能由子程序 OUTPUT 完成。

数字0-9与加密和解密数据的对应关系如下：

原 数 字：	0	1	2	3	4	5	6	7	8	9
密码数字：	7	5	9	1	3	6	8	0	2	4
解密数字：	7	3	8	4	9	1	5	0	6	2

(4) 编写一个程序，新建一个文件：d:\abc.txt,从键盘输入文件的内容（不超过100个字符）。然后新建一个文件：d:\def.txt,将d:\abc.txt 文件的内容复制到 d:\def.txt。

题目创建人

题目注释

```
8.1.asm
DATAS SEGMENT
    VAL1    dw    22d
DATAS ENDS

STACKS SEGMENT

STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
main proc far
START:
    MOV AX,DATAS
    MOV DS,AX
```

```

        call bando
        mov ah,4ch
        int 21h
main endp
bando proc near
    push VAL1
    call PAIRS
    ret
bando endp

PAIRS proc near
    pop bx
    pop ax
    push bx
    call OUTBIN    ;二进制输出
    call kongge    ;8个空格键
    call OUTOCT    ;八进制输出
    ret
PAIRS endp

OUTBIN proc near
    mov cx,16
    mov bx,ax
L:
    rcl bx,1
    mov dx,0
    adc dx,30h
    mov ah,2
    int 21h
    loop L
    ret
OUTBIN endp
kongge proc near
    mov cx,8
    mov di,20h
L1:
    mov ah,2
    int 21h
    loop L1
    ret
kongge endp
OUTOCT proc near
    mov dx,0
```

```

        mov cx,0
        mov ax,VAL1
        mov bx,8
L4:
        mov dx,0
        div bx
        push dx
        inc cx
        cmp ax,0
        jnz L4
L5:
        pop dx
        add di,30h
        mov ah,2
        int 21h
        loop L5
ret
OUTOCT endp
CODES ENDS
END START

8-2-1.asm (用“-”)
extrn proadd:far
DATAS SEGMENT common
    array    dw    1,2,3
    count dw 3
    sum      dw    ?
DATAS ENDS

STACKS SEGMENT

STACKS ENDS

CODES1 SEGMENT
ASSUME CS:CODES1,DS:DATAS,SS:STACKS
start:
    push ds
    xor  ax,ax
    push ax
    MOV AX,DATAS
    MOV DS,AX
    call far ptr proadd
    mov ah,4ch
    int 21h

```

```

CODES1 ENDS
END start

```

```

8-2-2.asm
public proadd
DATAS SEGMENT common
    array dw 1,2,3
    count dw 3
    sum    dw ?
DATAS ENDS

STACKS SEGMENT

STACKS ENDS

CODES2 SEGMENT
proadd proc far
ASSUME CS:CODES2,DS:DATAS,SS:STACKS
    MOV AX,DATAS
    MOV DS,AX
    push ax
    push cx
    push si
    lea si,array
    mov cx,count
    xor ax,ax
next:
    add ax,[si]
    add si,2
    loop next
    mov sum,ax
    mov dx,ax
    add di,30h

```

```

mov ah,2
int 21h
pop si
pop cx
pop ax
ret
proadd endp
CODES2 ENDS
end

```

8.3.asm

```

DATAS SEGMENT
    array db 50 dup(?)
    count dw 0
    c1rf db 0dh,0ah,'$'
    cc1 db 7,5,9,1,3,6,8,0,2,4
    cc2 db 0,4,6,9,2,3,1,7,5,8
DATAS ENDS

STACKS SEGMENT

STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
    MOV AX,DATAS
    MOV DS,AX

    lea si,array
    call shuru
shuru:    ;原数字串吸收
    mov ah,1
    int 21h

```

```

    cmp al,0dh
    jz exit
    sub al,30h
    mov [si],al
    inc si
    inc count
    loop shuru
exit:    ;换行
    lea dx,c1rf
    mov ah,9
    int 21h

    mov cx,count
    lea si,array
check:    ;加密替换原数字串
    mov bl,[si]
    lea di,cc1[0x]
    mov al,[di]
    mov [si],al
    inc si
    loop check

    mov cx,count
    lea si,array
next:    ;显示加密后的数字串
    mov dx,[si]
    add di,30h
    mov ah,2
    int 21h
    inc si
    loop next

    lea dx,c1rf    ;换行
    mov ah,9
    int 21h

    call OUTPUT    ;解密并显示
    MOV AH,4CH
    INT 21H

OUTPUT proc near
    mov cx,count
    lea si,array

```

```

check2:
    mov bl,[si]
    lea di,cc2[bx]
    mov al,[di]
    mov [si],al
    inc si
    loop check2

    mov cx,count
    lea si,array
next2:
    mov dx,[si]
    add di,30h
    mov ah,2
    int 21h
    inc si
    loop next2
    ret
OUTPUT endp
CODES ENDS
    END START

```

8.4.asm

```

DATAS SEGMENT
    wenjianming db 'd:\abc.txt',00h
    handle1     dw  ?
    neirong     db  101,?,100 dup(?)
    handle2     dw  ?
    neirong2    db  100 dup(?)
    crlf        db  0ah,0dh,'$'
    wenjianming2 db 'd:\def.txt',00h
DATAS ENDS

STACKS SEGMENT

STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
    MOV AX,DATAS
    MOV DS,AX

```

```

mov ah,3ch      ;建立文件
mov cx,0
lea dx,wenjianming
int 21h
mov handle1,ax  ;装入文件代号

lea dx,neirong  ;输入文件内容
mov ah,0ah
int 21h

mov ah,40h      ;写入到文件中
mov bx,handle1
lea di,neirong+1
mov cl,[di]
lea dx,neirong+2
int 21h

mov ah,3dh      ;为读而打开文件
mov al,00
lea dx,wenjianming
int 21h
mov handle2,ax

mov ah,3fh      ;读文件
mov bx,handle2
lea di,neirong+1
mov cl,[di]
lea dx,neirong2
int 21h

mov ah,3ch      ;建立文件
mov cx,00
lea dx,wenjianming2
int 21h
mov handle1,ax

mov ah,40h      ;写入到文件中
mov bx,handle1
lea di,neirong+1
mov cl,[di]
lea dx,neirong2
int 21h

```

```
MOV AH,4CH  
INT 21H  
CODES ENDS  
END START
```