实验 1

1.实现一个程序输出 Hello world!

```
        DATA SEGMENT
        TARS DW ’ Hello world! ’ ’ $’ ；输入字符串
        DATA ENDS
         CODE SEGMENT
         ASSUME CS:CODE,DS:DATA,ES:DATA
MAINS: MOV AX,DATA
          MOV DS,AX
          MOV DX,OFFSET TARS
          MOV AH,09H
          INT 21H
         MOV AX,4CH
         INT 21H
         CODE ENDS
         END MAINS
```

2.编程实现 Z=5X+3Y+10。掌握表达式计算程序设计，观察段寄存器、通用寄存器的变化。

```
DATA SEGMENT
X DB 1
Y DB 2
U DB 5
V DB 3
Z DW ?
DATA ENDS
STACKS SEGMENT
STACKS ENDS
CODE SEGMENT
ASSUME CS:CODE,DS:DATA,SS:STACKS
MAINS: MOV AX,DATA
          MOV DS,AX
        MOV AL,X
        MUL U
        MOV BL,Y
        MOV DX,AX
        MOV AX,BX
        MUL V
        ADD AX,DX
        ADC AX,10
       MOV Z,AX
  ;-----------------------显示 SUM
          MOV BL,100
             DIV BL
             MOV DL,AL
```

```
            ADD DL,30H
            MOV CL,AH;保护余数
            MOV AH,2
            INT 21H
            MOV AL,CL
            CBW
            MOV BL,10
            DIV BL
            MOV DL,AL
            ADD DL,30H
            MOV CL,AH;保护余数
            MOV AH,2
            INT 21H
            MOV DL,CL
            ADD DL,30H
            MOV AH,2
            INT 21H
            MOV AH,4CH
            INT 21H
CODE ENDS
END MAINS
```

3. 编程实现三分支的符号函数

```
  DATA SEGMENT
X DB 0
Y DB ?
DATA ENDS
CODE SEGMENT
ASSUME CS:CODE,DS:DATA
MAINS: PUSH DS
SBB AX,AX
PUSH AX
MOV AX,DATA
MOV DS,AX
;-----------------
MOV AL,X
CMP AL,0
JGE BIGER
MOV AL,0FFH
MOV Y,AL
MOV DL,'-'
MOV AH,2
INT 21H
MOV DL,31H
MOV AH,2
```

```
INT 21H
MOV AH,4CH
    INT 21H
JMP FINS
;-------------
BIGER: JE EQUAL
     MOV AL,1
     MOV Y,AL
     MOV DL,31H
     MOV AH,2
     INT 21H
     MOV AH,4CH
    INT 21H
     JMP FINS
EQUAL: MOV Y,AL
    MOV DL,30H
    MOV AH,2
    INT 21H
    MOV AH,4CH
    INT 21H
      JMP FINS
FINS: RET
CODE ENDS
     END MAINS
```

4. 编程实现对 1～50 累加求和。

```
DATA SEGMENT
     SUM DW ?
DATA ENDS
CODE SEGMENT
ASSUME CS:CODE,DS:DATA
;----------------------------------
MAINS: MOV AX,DATA
       MOV DS,AX
       XOR AX,AX
       MOV CL,50
SUMING: ADD AX,CX
        DEC CX
        JNZ SUMING
        MOV SUM,AX
;-------------------------
        PUSH AX
        MOV AX,SUM
        CWD
        MOV BX,1000    ;取商的低位，此处为0001
        DIV BX
        MOV CL,AL
        ADD AL,30H      ;ASCII 码存入 DL 用于显示输出
        MOV DL,AL
        MOV AH,02H
        INT 21H         ;第一位输出
;-------------------------
        MOV CH,0
        MOV AX,CX
        MOV BX,1000
        MUL BX ;千位的数×1000，存入 AX、DX  位的数×1000，存入 AX、DX
        MOV CX,AX    ;结果的低位存入 CX
        POP AX
        SUB AX,CX
        PUSH AX
        MOV BX,100    ;对百位的显示与千位显示思路相同
        DIV BX
        MOV CL,AL
        ADD AL,30H      ;ASCII 码存入 DL 用于显示输出
        MOV DL,AL
        MOV AH,02H
        INT 21H         ;第二位输出
;---------------------------
      MOV AX,CX
      MOV BL,100
      MUL BL
      MOV CX,AX
      POP AX
      SUB AX,CX
      PUSH AX
      MOV    CL,10                    ;对十位的显示与千位显示思路相同
      DIV    CL
      MOV CL,AL
      ADD    AL,30H
      MOV    DL,AL
      MOV    AH,02H
      INT    21H                    ;第三位输出
;--------------------------
       MOV AX,CX
       MOV BL,10
       MUL BL
       MOV CX,AX
       POP AX
       SUB AX,CX
```

```
        ADD    AL,30H
        MOV    DL,AL
        MOV    AH,02H
        INT    21H                   ;第四位输出
;--------------------------
        MOV AH,4CH
        INT 21H
        CODE ENDS
        END MAINS
```

5.编写统计X数据（AL）中的“1”的个数子程序，并编写主程序调用

```
DATA SEGMENT
X       DB 0FEH
TARS DB 80H,40H,20H,10H,08H,04H,02H,01H    ;屏蔽字
RESULT DW ?
DATA ENDS
;---------------------------
CODE SEGMENT
ASSUME CS:CODE,DS:DATA
MAIN PROC FAR
   PUSH DS
          MOV AX,0;
          PUSH AX
          MOV AX,DATA
          MOV DS,AX
          CALL FAR PTR COUNTS;调用
          MOV DL,CL
          ADD DL,30H
          MOV AH,02H
          INT 21H      ;显示
          RET
    MAIN ENDP

;--------------------------------
    COUNTS PROC FAR
          MOV AL,X
          MOV CX,0         ;计数
          MOV SI,0       ;指针指第一个
BLOOPS: TEST AL,TARS[SI]
          JZ NEXT
          INC CX
NEXT:    INC SI          ;指下一个元素
         CMP SI,8       ;对比完毕？
         JNE BLOOPS    ;未完成，继续
         RET    ;完成返回
```

```
COUNTS    ENDP
CODE ENDS
END MAIN
```

(7)

6.编写在屏幕上显示小人像的程序；编程实现动画图像，实现小人像的移动。

掌握 DOS 中断调用、BIOS 中断调用方法，学习字符的显示、动作间的延时设计，光标的移动。

```
DATAS SEGMENT
CHRTAB DW 5
            DB 01H,0,0,0DBH,1,0,13H,1,0
            DB 2FH,-1,-1,5CH,0,2
DATAS ENDS
;----------------------------------------------;
STACKS SEGMENT PARA STACK 'STACK'
            DB 100 DUP(?)
STACKS ENDS
;----------------------------------------------;
CODES SEGMENT
       ASSUME CS:CODES,DS:DATAS
START:
      MOVING      PROC FAR
      PUSH DS
      MOV AX,0
      PUSH AX
      MOV AX,DATAS
      MOV DS,AX
;----------------------------------------------;显示方式 80*25 黑白
       STI
       MOV AL,02
       MOV AH,0
       INT 10H

       MOV DH,20
      MOV DL,5                                      ;定位初始
;----------------------------------------------;跳跃程序 start
MOVE:

      MOV DI,OFFSET CHRTAB
       MOV CX,[DI]
       DEC DH
       DEC DL
      SUB DH,2

       ADD DI,2
;----------------------------------------------;显示函数
```

```
      PUSH DX                                        ;保护位置坐标
NEXT :
      ADD DH,[DI+1]
      ADD DL,[DI+2]
      MOV AH,2
      INT 10H
      MOV AL,[DI]
      PUSH CX
      MOV CX,1
      MOV AH,10
      INT 10H
      POP CX
      ADD DI,3
      LOOP NEXT
;-----------------------------------------------;延时函数
DELAY:
      PUSH BX
      PUSH CX
       MOV BX,1000
CYCOUT:
      MOV CX,1000
CYCIN:
      LOOP CYCIN
      DEC BX
      JNZ CYCOUT
       POP CX
       POP BX

;-----------------------------------------------;清除函数
      MOV DI,OFFSET CHRTAB
       MOV CX,[DI]
      POP DX
       ADD DI,2
CLEAR:
      ADD DH,[DI+1]
      ADD DL,[DI+2]
      MOV AH,2
      INT 10H
      MOV AL,00H
      PUSH CX
      MOV CX,1
      MOV AH,10
      INT 10H
      POP CX
```

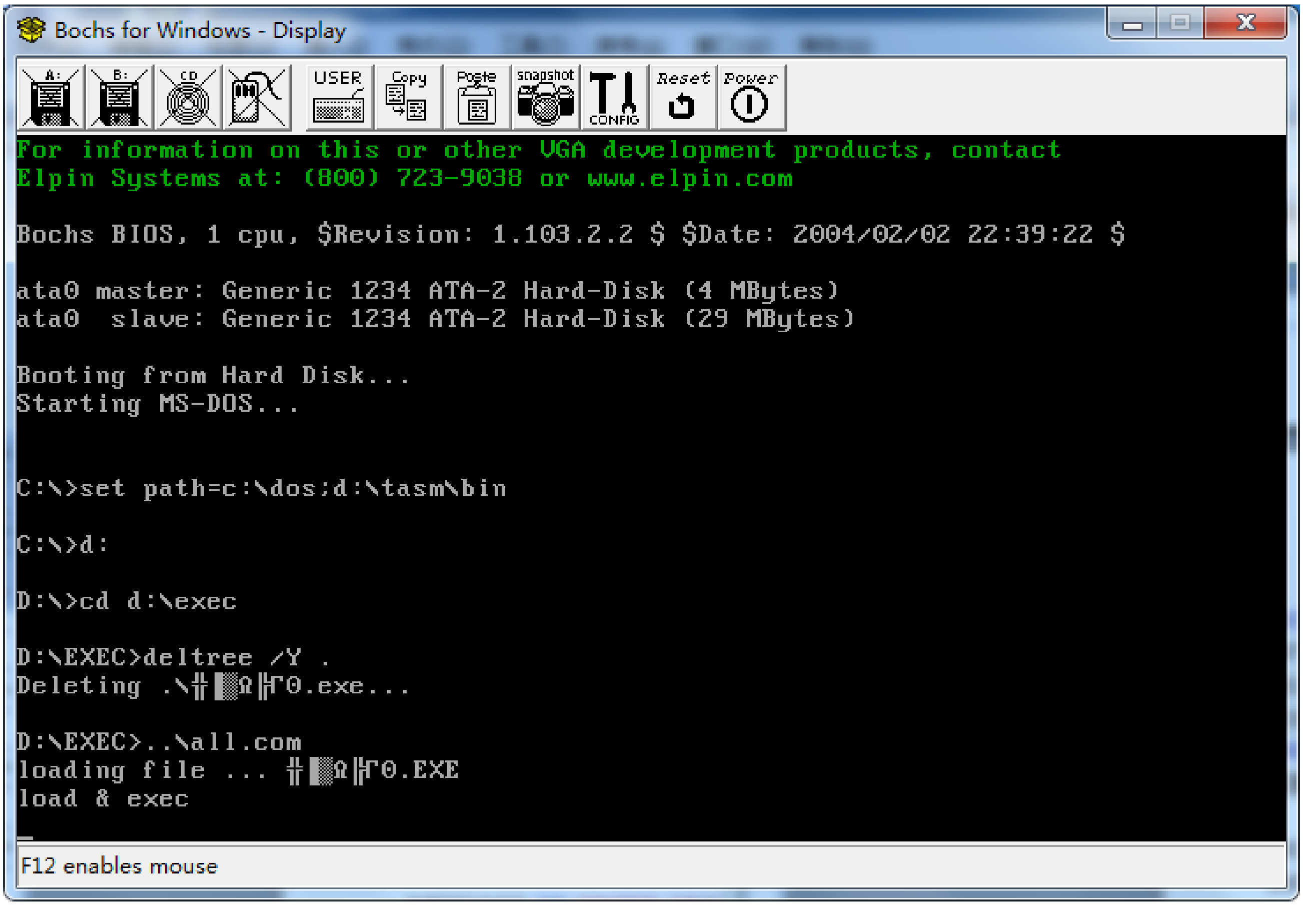
```
        ADD DI,3
        LOOP CLEAR
        CMP DH,10
        JLE MOVE2
JMP MOVE
;----------------------------------------------;跳跃程序 over
MOVE2:
        MOV DI,OFFSET CHRTAB
         MOV CX,[DI]
        DEC DH
         DEC DL
        ADD DL,2
         ADD DI,2
;----------------------------------------------;显示函数
        PUSH DX                                          ;保护位置坐标
NEXT2 :
        ADD DH,[DI+1]
        ADD DL,[DI+2]
        MOV AH,2
        INT 10H
        MOV AL,[DI]
        PUSH CX
        MOV CX,1
        MOV AH,10
        INT 10H
        POP CX
        ADD DI,3
        LOOP NEXT2
;----------------------------------------------;延时函数
DELAY2:
        PUSH BX
        PUSH CX
         MOV BX,100
CYCOUT2:
        MOV CX,30
CYCIN2:
        LOOP CYCIN
        DEC BX
        JNZ CYCOUT
         POP CX
         POP BX

;----------------------------------------------;清除函数
        MOV DI,OFFSET CHRTAB
```

```
     MOV CX,[DI]
    POP DX
     ADD DI,2
CLEAR2:
    ADD DH,[DI+1]
    ADD DL,[DI+2]
    MOV AH,2
    INT 10H
    MOV AL,00H
    PUSH CX
    MOV CX,1
    MOV AH,10
    INT 10H
    POP CX
    ADD DI,3
    LOOP CLEAR2

JMP MOVE2
;----------------------------------------------------;跳跃程序 over
RET
MOVING      ENDP
CODES ENDS
END START
```

实验二 I/O 地址译码

```
DATA SEGMENT
DATA ENDS
CODE SEGMENT
ASSUME DS:DATA,ES:DATA,CS:CODE
FLASHS:
    MOV  DX,2A0H
        OUT  DX,AL
       CALL  DELAYS
    MOV  DX,2A8H
        OUT  DX,AL
       CALL  DELAYS
        JMP  FLASHS
          RET
     DELAYS PROC
       MOV  BL,64H
 DELAY: MOV  CX,0FFFFH
 CYCLE: LOOP  CYCLE
        DEC  BL
        JNZ  DELAY
```

```
MOV AH,4CH
INT 21H
DELAYS ENDP
CODE ENDS
END FLASHS
```

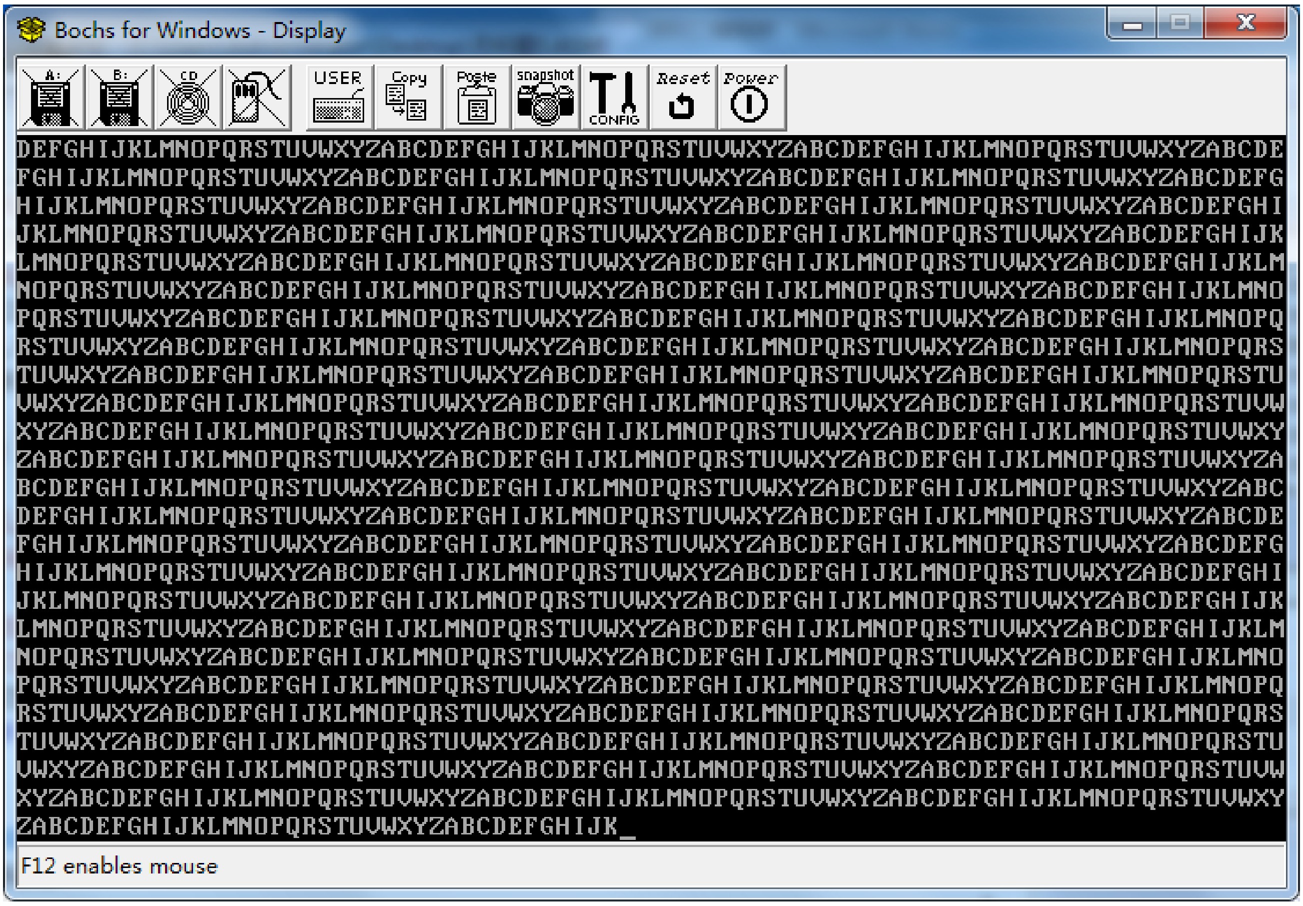


Bochs for Windows - Display

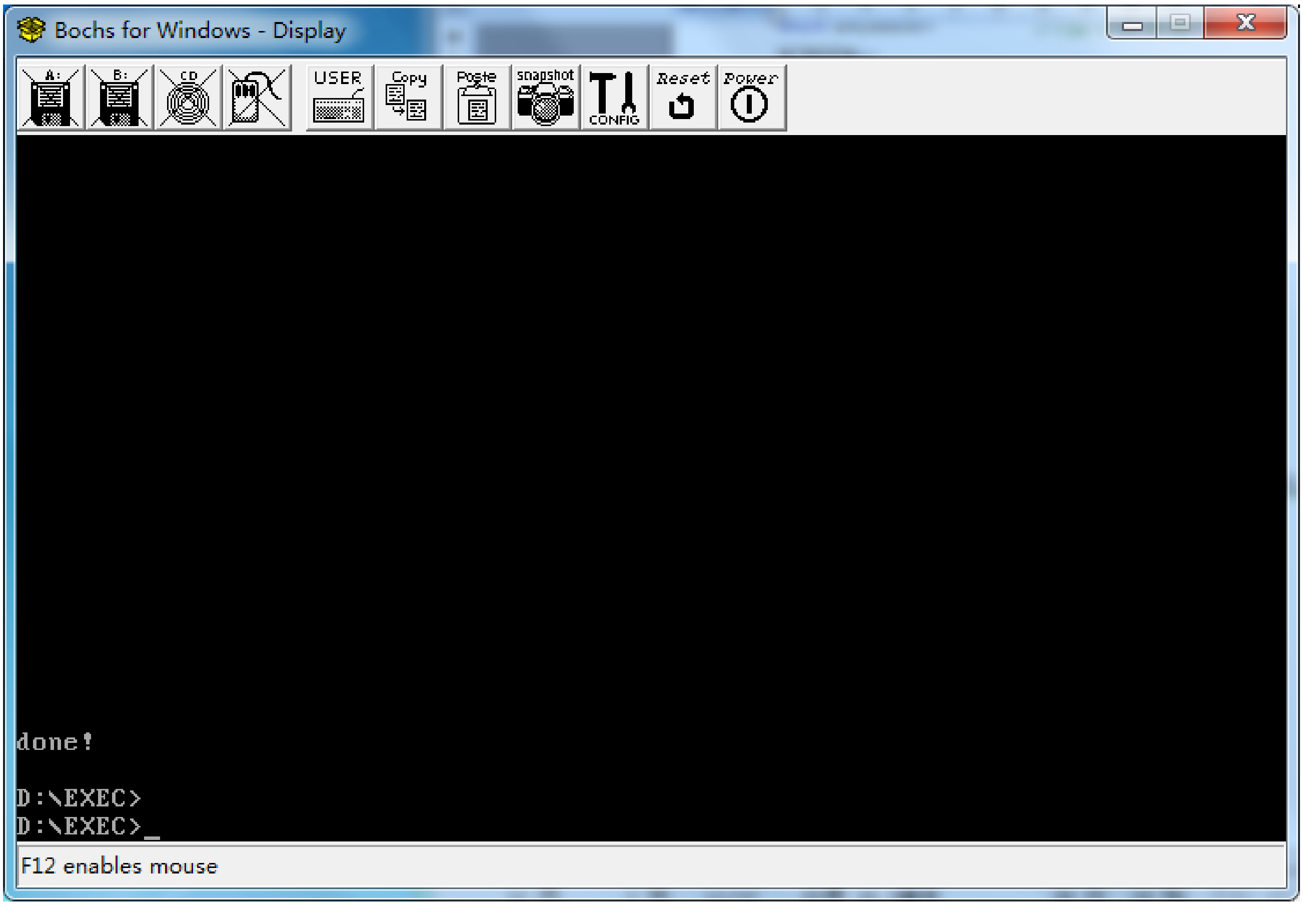
```
For information on this or other VGA development products, contact
Elpin Systems at: (800) 723-9038 or www.elpin.com

Bochs BIOS, 1 cpu, $Revision: 1.103.2.2 $ $Date: 2004/02/02 22:39:22 $

ata0 master: Generic 1234 ATA-2 Hard-Disk (4 MBytes)
ata0  slave: Generic 1234 ATA-2 Hard-Disk (29 MBytes)

Booting from Hard Disk...
Starting MS-DOS...


C:\>set path=c:\dos;d:\tasm\bin

C:\>d:

D:\>cd d:\exec

D:\EXEC>deltree /Y .
Deleting .\╫▌▒Ω╟ΓΘ.exe...

D:\EXEC>..\all.com
loading file ... ╫▌▒Ω╟ΓΘ.EXE
load & exec

_
```

实验三 存储器读写实验

```
     DATAS SEGMENT
MESSAGE DB 'ENTER ANY KEY TO EXIT TO DOS!',13,10,'$'
DATAS ENDS
   CODES SEGMENT
ASSUME CS:CODES,DS:DATAS
  MAINS:
       PUSH AX
       XOR AX,AX     ;清 0
       MOV AX,DATAS
       MOV DS,AX    ;初始化
       MOV CX,256     ;移入计数初值
       MOV DX,0D000H  ;起始地址
       MOV ES,DX     ;附加段地址寄存器
       MOV BX,6000H    ;存储地址
CYCLE: MOV AX,40H      ;A-Z ASCII 码
OUTPUT:
       INC AX
       MOV ES:[BX],AX     ;段超越前缀
       INC BX           ;指下一个单元
       CMP AX,5AH        ;到Z停
       JNZ OUTPUT         ;反复存入
       DEC CX             ;CX-1
       JNZ CYCLE          ;再存一轮
      MOV AH ,09H         ;中断提示
       LEA DX,MESSAGE     ;指提示信息
        INT 21H
       MOV AH,01H         ;键盘输入
        INT 21H
        MOV CX,256*26
MOV DX,0D000H
MOV ES,DX
MOV BX,6000H         ;为显示做准备
SCREEN:
       MOV DL,ES:[BX]
        MOV AH,02H
         INT 21H
        INC BX
        DEC CX
       JNZ SCREEN       ;循环显示
         RET
     CODES ENDS
      END  MAINS
```



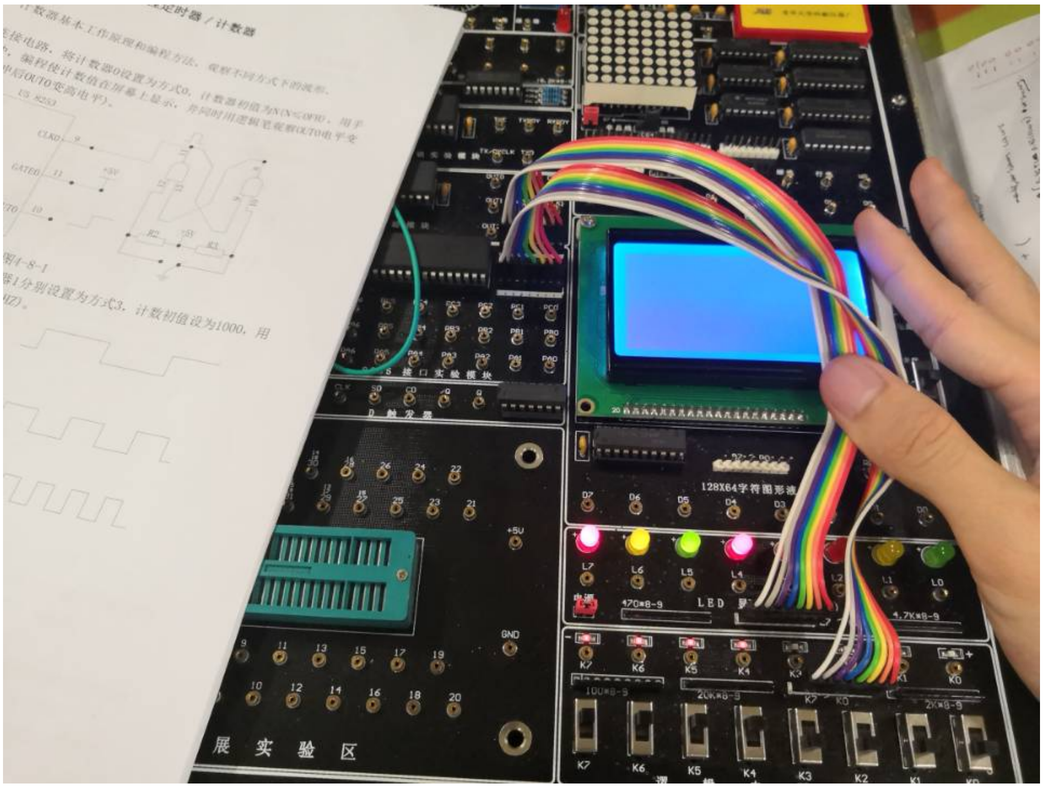
```
DEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDE
FGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFG
HIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHI
JKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJK
LMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLM
NOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNO
PQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQ
RSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRS
TUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTU
VWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVW
XYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXY
ZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZA
BCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABC
DEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDE
FGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFG
HIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHI
JKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJK
LMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLM
NOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNO
PQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQ
RSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRS
TUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTU
VWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVW
XYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXY
ZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJK_
```

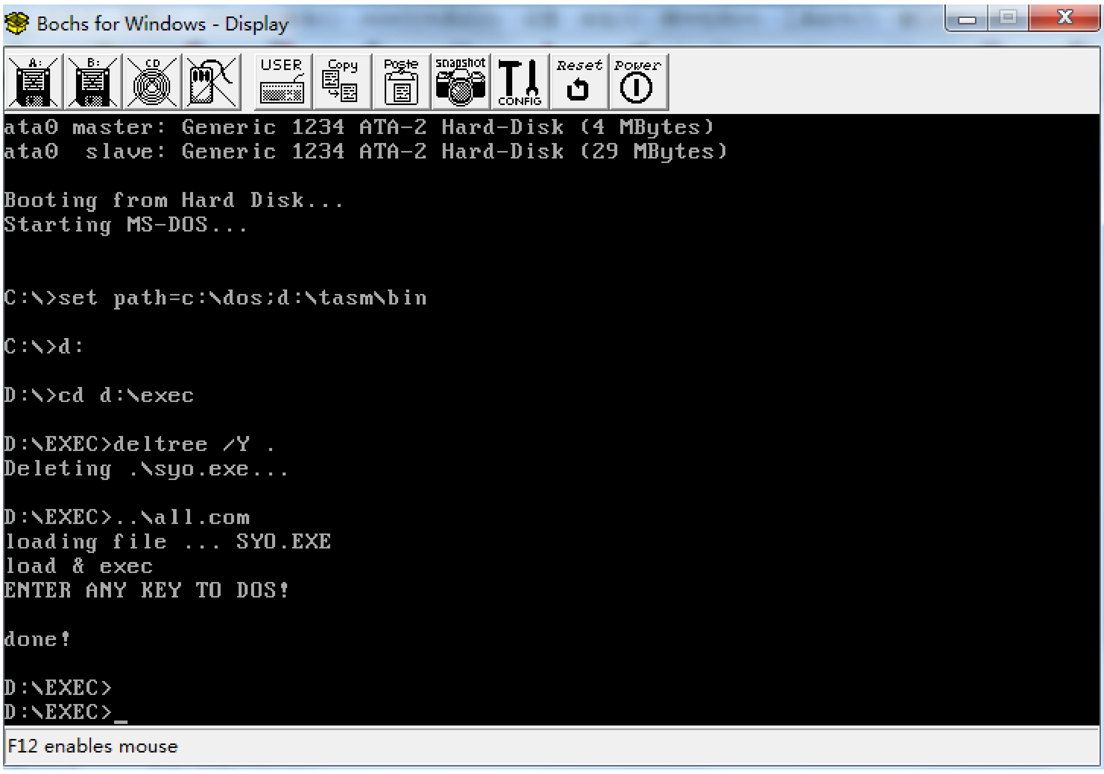


```
done!

D:\EXEC>
D:\EXEC>_
```

实验四 可编程并行接口

```
            DATAS SEGMENT
MESSAGE DB ’ENTER ANY KEY TO EXIT TO DOS!’,13,10,’$’
DATAS ENDS
   CODES SEGMENT
ASSUME CS:CODES,DS:DATAS
  MAINS:
       PUSH AX
       XOR AX,AX     ;清0
       MOV AX,DATAS
       MOV DS,AX   ;初始化
       LEA DX,MESSAGE   ;指提示信息
       MOV AH,09H        ;中断提示
        INT 21H
       MOV DX,28BH
       MOV AL,89H   ;控制字设置工作方式
       OUT DX,AL      ;控制字送到控制端口
   READING:
       MOV DX,28AH
       IN AL,DX      ;C口输入
  WRITING:
       MOV DX,288H
       OUT DX,AL     ;A口输出
       MOV DL,0FFH     ;检查是否有按键
        MOV AH,06H
        INT 21H
       JNZ EXITS  ;无按键按下。跳到EXITS
JMP READING       ;有按键按下，跳到READING
     EXITS:
     MOV AH,4CH
        INT 21H
        CODES ENDS
END MAINS
```





```
ata0 master: Generic 1234 ATA-2 Hard-Disk (4 MBytes)
ata0  slave: Generic 1234 ATA-2 Hard-Disk (29 MBytes)

Booting from Hard Disk...
Starting MS-DOS...


C:\>set path=c:\dos;d:\tasm\bin

C:\>d:

D:\>cd d:\exec

D:\EXEC>deltree /Y .
Deleting .\syo.exe...

D:\EXEC>..\all.com
loading file ... SYO.EXE
load & exec
ENTER ANY KEY TO DOS!

done!

D:\EXEC>
D:\EXEC>_
```

F12 enables mouse

实验五 可编程定时器/计数器

（1）

```
CODE    SEGMENT
        ASSUME        CS:CODE
START:
        MOV AL,10H    ;控制字 00010000
MOV DX,283H
OUT DX,AL    ;控制字送到控制端口
MOV DX,280H
MOV AL,0FH    ;设计数初值为 0FH
OUT DX,AL    ;计数初值送到计数器 0 端口
CYCLE:
        IN AL,DX
        CALL DISP
        PUSH DX
        MOV    AH,06H
        MOV    DL,0FFH           ;直接控制台，输入到 AL
        INT    21H
        MOV    AH,01H
        POP    DX
        JZ      CYCLE
DISP PROC NEAR              ;显示子程序
        PUSH DX
        AND    AL,0FH                ;取低四位
        MOV DL,AL
        CMP DL,9                     ;判断是否<=9
        JLE NUM                      ;若是则为'0'-'9',ASCII 码加 30H
        ADD DL,37H                  ;否则为'A'-'F',ASCII 码加 37H
        JMP EXIT
NUM:    ADD DL,30H
EXIT:
        MOV    AH,02H                ;显示
        INT    21H
        MOV AH,02H
        MOV    DL,0DH
        INT    21H
        POP    DX
        RET                          ;子程序返回
DISP    ENDP
CODE ENDS
        END         START
```

```
（2）DATA SEGMENT
DATA ENDS
CODE SEGMENT
ASSUME CS:CODE,DS:DATA
START:
MOV AX,DATA
MOV DS,AX    ;给DS赋初值
SET0:
MOV DX,283H
MOV AL,36H    ;计数器0控制字为00110110
OUT DX,AL    ;控制字送到控制端口
MOV DX,280H
MOV AX,1000    ;设计数初值1000
OUT DX,AL        ;送计数初值低八位
MOV AL,AH
OUT DX,AL        ;送计数初值高八位
SET1:
MOV DX,283H
MOV AL,76H ;计数器1控制字为01110110
OUT DX,AL ;控制字送到控制端口
MOV DX,281H
MOV AX,1000 ;设计数初值1000
OUT DX,AL ;送计数初值低八位
MOV AL,AH
OUT DX,AL ;送计数初值高八位
CODE ENDS
END START
```

实验六 中断

(1)

```
DATA SEGMENT
            MESS DB 'TPCA interrupt1',0DH,0AH ,'$'
            DATA ENDS
CODE SEGMENT
            ASSUME CS:CODE, DS :DATA
START :
            MOV AX,CS
            MOV DS,AX
            MOV DX ,OFFSET INT3
            MOV AX,250BH
            INT 21H             ;设置中断向量类型号
```

```
        IN AL,21H
        AND AL,0F7H    ;设置屏蔽字
        OUT 21H,AL

        MOV CX,10          ;设置总次数 10 次
        STI
         HERE:JMP HERE
INT3:   MOV AX,DATA
        MOV DS,AX
        MOV DX,OFFSET MESS
        MOV AH ,09
        INT 21H

        MOV AL ,20H
        OUT 20H,AL ;发出中断结束命令 EOI，IRR 复位
        LOOP NEXT

        IN AL,21H        ;关中断
        OR AL,08H
        OUT 21H,AL

        STI                ;打开中断
        MOV AH,4CH
        INT 21H
NEXT:   IRET
        CODE ENDS
        END START
```

(2)

```
DATA SEGMENT
INTA00 EQU 20H
INTA01 EQU 21H
INTXA00 EQU 0A0H
INTXA01 EQU 0A1H
DATA ENDS
CODE SEGMENT
ASSUME DS:DATA, CS:CODE
START:
MOV AX,CS
MOV DS,AX
MOV DX,OFFSET INTPROC
MOV AX,2572H    ;给中断向量
INT 21H
CLI    ;关中断
MOV DX, INTA01
```

```
IN AL, DX
AND AL, 0FBH
OUT DX, AL ;设置屏蔽字 MOV DX, INTXA01
;设置从机屏蔽字
MOV AL, 0F2H
OUT DX, AL
MOV BX, 10; INTERUPTS TIMES
STI          ;开中断
LL: JMP LL
INTPROC:
MOV AX, DATA
MOV DS, AX
MOV DL, 0FH
MOV AH, 02H
INT 21H
MOV DX, INTA00
MOV AL, 20H
OUT DX, AL       ;主机  IRR  复位
MOV DX, INTXA00
MOV AL, 20H
OUT DX, AL         ;从机  IRR  复位
SUB BX, 1
JNZ NEXT
MOV DX,INTA01      ;主机中断结束
IN AL,DX
OR AL,04H
OUT DX,AL
MOV DX, INTXA01      ;主机中断结束
IN AL, DX
OR AL, 04H
OUT DX, AL
STI
MOV AH,4CH
INT 21H
NEXT: IRET
CODE ENDS
END START
```

实验七 拓展中断

```
DATA    SEGMENT
MES1    DB'YOU CAN PLAY A KEY ON THE KEYBOARD!',0DH,0AH,'$'
MES2    DD MES1
MESS1 DB 'HELLO! THIS IS INTERRUPT * 0 *!',0DH,0AH,'$'
MESS2 DB 'HELLO! THIS IS INTERRUPT * 1 *!',0DH,0AH,'$'
DATA    ENDS
CODE    SEGMENT
    ASSUME CS:CODE,DS:DATA
START: MOV      DX,2B0H         ;I8259_1
        MOV     AL,13H
        OUT     DX,AL
        MOV     DX,2B1H         ;I8259_2
        MOV     AL,08H
        OUT     DX,AL
        MOV     AL,0FH
        OUT     DX,AL

        CLI
        MOV     DX,2B1H;O8259_1
        MOV     AL,0
        OUT     DX,AL
        STI
        MOV     AX,DATA
        MOV     DS,AX
        MOV     DX,OFFSET MES1
        MOV     AH,09H
        INT 21H

TT:  MOV    AH,0BH
       INT   21H
       CMP AL,0
       JNZ   NEXT
       MOV      DX,2B0H        ;O8259_3
       MOV      AL,0FH
       OUT  DX,AL
       NOP
       IN     AL,DX
       MOV      BL,AL
       AND  AL,80H
       JNZ   JUMP
       JMP  TT
JUMP:
    MOV     AL,BL
       AND  AL,07H
       CMP  AL,0
       JZ      INNT0
       CMP  AL,1
       JZ      INNT1
       JMP   TT
INNT0:
    MOV     DX,OFFSET MESS1
       MOV        AH,09H
       INT    21H
       JMP   TT
INNT1:
    MOV     DX,OFFSET MESS2
       MOV        AH,09H
       INT    21H
       JMP   TT
NEXT:   MOV      AH,4CH
        INT  21H
CODE     ENDS
    END START
```