

# 实验十 键盘中断

学号： 04022212

姓名： 钟 源

## 一、实验任务和实验结果

请写出**已通过验收**的各个实验任务的具体内容、调试通过的源程序（**加注释**）和实验结果。

实验结果请截图，并加以必要说明。

### 1.任务：

要求每按下任意一个键就向 CPU 发出中断请求信号，该信号由 8259 的 IRQ1 引入，中断类  
型号为 09，CPU 响应中断后转入执行 KEYINTS 中断服务程序，并在屏幕上显示“OK! ”，按下  
10 次键后返回 DOS。

#### 附加任务：

1) 通过 DOS 系统功能调用的 25H、35H 功能实现中断向量的设置和读取；

2) 改变按键后屏幕显示的字符串内容和返回 DOS 之前的按键次数，

比如：按键后在屏幕上显示“KEYINT”，按下 9 次键后返回 DOS。按键显示的字符串内容和  
返回 DOS 之前的按键次数 各位同学可以自己设置，尽量不要太雷同，要求显示字符串的字符个  
数 > 3，按键次数 > 8。

3) 在按键后显示的字符串前面加上按键次数，在字符串后面加个空格，这样两次按键显示字  
符串之间有个空格间隔区分一下；

4) 按键次数达到后(比如 9 次)，不等 25 行太阳图标显示完，立即返回 DOS；

5) 修改显示字符的属性，如，红底白字，蓝底黄字，.....；

6) 其他：鼓励同学们自己增加其他的附加功能。

2. 源程序及注释：

```
STACK SEGMENT ;堆栈段定义
    STK DW 200H DUP(?)
STACK ENDS

DATA SEGMENT
    KEY DB 0 ;存储按下键盘的次数，初值为0
;2) 改变按键后屏幕显示的字符串内容和返回 DOS 之前的按键次数。
    MSG DB ?,?,"-KUAI DIAN FANG JIA(QAQ)" ;存储按下键盘后的提示信息
    SLENGTH EQU $-MSG ;提示字符串长度
    COUNT EQU 12 ;最大按键次数
    LINE DB 20 ;最大显示行数
    INLINE DB 3;
        DB ?;
        DB 3 DUP(?); ;缓冲区
    TXT1 DB "Please input the number of line(<25):$" ;文本提示 1
    ERROR1 DB 0DH,0AH,'Wrong Input Format! ',0AH,'$' ;错误提示 1
    ERROR2 DB 0DH,0AH,'The Input is too big! ',0AH,'$' ;错误提示 2
DATA ENDS

CODE SEGMENT
    ASSUME CS:CODE, SS:STACK, DS:DATA
START:
    MOV AX, STACK
    MOV SS, AX
    MOV AX, DATA
    MOV DS, AX
    ;输出 TXT1 文本
    LEA SI,TXT1 ;取 TXT1 地址
    MOV DX,SI ;赋值地址给 DX
    MOV AH,09 ;功能号 09H
    INT 21H ;类型号 21H，显示 DS: DX 指向的以$结尾的字符串
    ;键盘输入行数到缓冲区
    LEA SI,INLINE;
    MOV DX,SI ;赋值地址给 DX
    MOV AH, 0AH
    INT 21H ;输入

    MOV BX, OFFSET INLINE+2 ;BX 取到数组的首地址
    CALL INPUTError ;判断输入格式是否正确

RIG:
    CALL ASCtoNUM ;把缓存区 ASC 码的转换成 NUM 码
    CALL VALUEError ;判断时间符合实际
```

```
NEXT1:  ;保存原 09H 中断内容

;1) 通过 DOS 系统功能调用的 25H、35H 功能实现中断向量的设置和读取。

    MOV AH,35H

    MOV AL,09H      ;使用 DOS 系统功能获取中断向量

    INT 21H

    PUSH BX

    PUSH ES

    CLI              ;关中断，设置中断向量

    PUSH DS

    MOV AX,SEG KEYINTS

    MOV DS,AX

    MOV DX,OFFSET KEYINTS

    MOV AH,25H

    MOV AL,09H

    INT 21H

    POP DS

    STI              ;开中断

AGAIN:

    CALL FAR PTR SHOW1 ;显示一轮太阳图形

    CMP KEY, COUNT    ;判断按键是否按下 COUNT 次

    JB  AGAIN         ;键盘未按下十次，则继续下一轮显示太阳图形

    ;若按下了十次，进行处理

    CLI              ;屏蔽外部中断，准备将原有中断向量恢复到中断向量表中

    POP DS

    POP DX

    MOV AH,25H

    MOV AL,09H

    INT 21H

    STI              ;打开外部中断

    ;返回 DOS

    MOV AH, 4CH

    INT 21H

;_____

;判断输入格式是否合规的子程序

INPUTError PROC

    MOV BX, OFFSET INLINE+2;BX 取到数组的首地址

    MOV CL, 00H        ;开始 CL 置零

    MOV AL, [BX]        ;提取 BX 对应的值

    CALL NUMError       ;判断是否是数字

    INC BX              ;判断下一字节

    MOV AL, [BX]        ;提取 BX 对应的值

    CALL NUMError       ;判断是否是数字
```

```

    CMP CL, 00H          ;CL=0 说明格式正确

    JE RIG              ;正确跳转
error:

    MOV DX, OFFSET ERROR1

    MOV AH, 9

    INT 21H            ;在屏幕显示 ERROR1

    JMP START          ;跳回开头

    RET

INPUTerror ENDP

;判断数字错误的子程序
NUMerror PROC

    CMP AL, 39H        ;判断是否大于‘9’

    JA ERR1

    CMP AL, 30H        ;判断是否小于‘0’

    JB ERR1

    RET

ERR1:

    MOV CL, 01H        ;其 ASC 码若不在数字范围，CL 置 1

    RET

NUMerror ENDP

;ASC 变压缩 NUM 的子程序
ASCtoNUM PROC

    PUSH SI

    MOV BX, OFFSET INLINE+2 ;指针指向数据区

    MOV SI,OFFSET LINE

    SUB CX,CX

    ;转换个位

    MOV CL, [BX+1]

    SUB CL, 30H

    ;转换十位

    MOV AL, [BX]

    SUB AL,30H

    MOV DL,10

    MUL DL

    ADD AX,CX          ;整合相加

    MOV [SI], AL        ;储存,不考虑输入行数高于 255 的可能

    SUB CX,CX

    MOV CL, [SI]        ;行数传递给 CL

    POP SI

    RET

ASCtoNUM ENDP

```

```
;判断时间符合实际的子程序

VALUEerror  PROC

    CMP  CL, 25D          ;将行数和 25 比较

    JA   DO              ;>25 则跳转

    JMP  NEXT1

DO: MOV  DX, OFFSET  ERROR2

    MOV  AH, 9

    INT  21H             ;在屏幕显示 ERROR2

    JMP  START           ;跳回开头

    RET

VALUEerror  ENDP

;_____

;键盘中断子程序

KEYINTS  PROC FAR

    PUSH AX

    PUSH SI

    STI                ;在子程序内打开外部中断

    IN   AL, 60H        ;通过 8255A 的 PA 口(PA 口地址为 60H)读取键盘扫描码

    MOV  AH, AL         ;从 8255A 的 PA 口读入键盘扫描码，将其存入 AH

    IN   AL, 61H        ;读入 PB 口的当前值

    ;完成在 PB（7）口的一次正脉冲输出，以通知键盘控制器已经处理了中断。

    OR   AL, 80H        ;PB（7）置 1

    OUT  61H, AL

    AND  AL, 7FH        ;PB（7）置 0，

    OUT  61H, AL

    ;

    TEST AH, 80H        ;检测键盘扫描码的最高位是否为 1，【弹起码-按下码=80H】

    JNZ  GO             ;若最高位为 1，说明此时键盘释放，跳转到子程序出口处

    STI

    INC  KEY            ;若最高位为 0，说明此时键盘按下，给计数器加一

    CALL FAR PTR SHOW2  ;并显示提示信息

GO:

    MOV  AL, 20H

    OUT  20H, AL        ;给 8259A 的 OCW1 发出表示中断结束的 EOI 命令

    POP  SI

    POP  AX

    IRET

KEYINTS  ENDP

;延时 1s 子程序

DELAYS  PROC

    ;压栈

    PUSH DX

    PUSH CX

    PUSH BX
```

```

        PUSH AX

        ;中断调用延时

        MOV AH, 2CH

        INT 21H          ;DOS 调用系统时间

        MOV BH, DH       ;存储当前秒数

BEGIN:

        INT 21H          ;调用系统时间

        CMP BH, DH       ;比较当前时间与上次调用的时间是否变化

        JNE NEXT         ;若有变化，则完成延时，退出程序

        JMP BEGIN        ;若无变化，则继续调用系统时间

NEXT:

        ;弹出

        POP AX

        POP BX

        POP CX

        POP DX

        RET

DELAYS ENDP

```

;显示太阳图形的子程序

```

SHOW1  PROC FAR

        ;压栈

        PUSH AX

        PUSH BX

        PUSH CX

        PUSH DX

        ;每轮显示太阳

        MOV AH, 15

        INT 10H          ;读取当前显示状态

        MOV AH, 0

        INT 10H          ;设置显示方式

        MOV CX, 1         ;要显示的字符数

        MOV DX, 0        ;设置光标起始位置为行号为0，列号为0

```

REPEAT:

;4) 按键次数达到后(比如9次)，不等25行太阳图标显示完，立即返回DOS;

```

        CMP KEY, COUNT

        JAE EXIT          ;若KEY达到COUNT，转至EXIT

        ;KEY小于COUNT的情况

        MOV AH, 2

        INT 10H          ;设置光标位置

        MOV AL, 0FH       ;太阳图形的ASCII码

        MOV CX, 1         ;要显示的字符数

        MOV AH, 10

        INT 10H          ;显示单个太阳图形

        CALL DELAYS       ;延时一秒

```

```

MOV AL, 03          ;设置显示方式(80*25 彩色文本)

MOV AH, 0

INT 10H            ;清屏

INC DH             ;行号+1

ADD DL, 2          ;列号+2

SUB BX,BX

MOV SI, OFFSET LINE;BX 取到数组的首地址

CMP DH, [SI]

JB REPEAT          ;如果没到 LINE 行，则跳转到 REPEAT 更新光标位置；到了 LINE 行则退出子程序
EXIT:

;弹出

POP DX

POP CX

POP BX

POP AX

RET

SHOW1 ENDP

;显示提示信息的子程序

SHOW2 PROC FAR

;压栈

PUSH SI

PUSH DX

PUSH CX

PUSH BX

PUSH AX

;读取当前光标位置

MOV AH, 3

MOV BH, 0

INT 10H

INC DL             ;列号+1（相当于加空格）

MOV AH, 2

INT 10H            ;将光标设在下一个位置，避免覆盖原字符

LEA SI, MSG        ;将字符串 MSG 的首地址装入 SI

MOV BYTE PTR[SI],30H

MOV BL,KEY

;3）在按键后显示的字符串前面加上按键次数，在字符串后面加个空格

CMP KEY,10

JB LESSthan

MOREthan:

MOV BYTE PTR[SI],31H

SUB BL,10

LESSthan:

MOV AL,BL

```

```

    OR    AL,30H

    MOV   BYTE PTR[SI+1],AL

    MOV   CX, SLENGTH    ;将字符串 MSG 的长度装入 CX
NEXTC:
;5) 修改显示字符的属性

    MOV   BL,01111000B    ;白底黑字

    MOV   AH,09H

    INT   10H

    ;循环显示提示信息的字符

    LODSB                ;从内存中的 SI 指向的位置读取一个字节，并将其存入 AL 寄存器，同时 SI 寄存器递增。

    MOV   AH, 0EH        ;设置 AH 寄存器为 0EH，表示显示字符和属性。

    MOV   BX, 1

    INT   10H

    LOOP  NEXTC

    ;弹出

    POP   AX

    POP   BX

    POP   CX

    POP   DX

    POP   SI

    RET

SHOW2   ENDP

CODE    ENDS

        END   START

```

### 3.程序设计思路：

(1) 主程序开始于 `START`：

① 首先输出提示信息 "Please input the number of line(<25):", 然后等待用户输入。

② 指令 `LEA SI,TXT1` 将提示信息的地址加载到 SI 寄存器中，然后通过 `INT 21H` 中断调用功能号 `AH=09H` 将提示信息显示在屏幕上。

(2) 在 `INPUTError` 过程中进行输入格式验证：

① 通过 `NUMError` 过程判断输入的字符是否为数字，若不是数字则将错误标记 `CL` 置为 1。

② 如果输入格式错误，跳转至 `error` 标签，显示错误提示信息并返回程序开始处等待



重新输入。

(3) 在 `ASCtoNUM` 过程中将 ASCII 码转换为数字：

- ① 首先获取缓冲区的地址，并将其存入 `BX` 寄存器。
- ② 将个位和十位的 ASCII 码转换为数字，并将结果存入 `SI` 寄存器。
- ③ 结果传递给 `CL` 寄存器，即行数的值。

(4) 在 `VALUEError` 过程中进行输入值的大小验证：

判断输入的行数是否大于 25，如果大于 25，则显示错误信息并返回程序开始处等待重新输入。

(5) 通过 DOS 系统调用设置键盘中断向量：

- ① 通过 `INT 21H` 调用功能号 `AH=35H` 获取键盘中断的原始中断向量，并保存。
- ② 使用 `INT 21H` 调用功能号 `AH=25H` 设置键盘中断向量为 `KEYINTS` 过程的地址。

(6) 在 `KEYINTS` 过程中处理键盘中断：

- ① 通过输入输出指令读取键盘扫描码，判断是否有键盘按下或弹起。
- ② 若按键按下，则计数器 `KEY` 加一，并调用 `SHOW2` 过程显示提示信息。

(7) 在 `SHOW1` 过程中显示太阳图案：

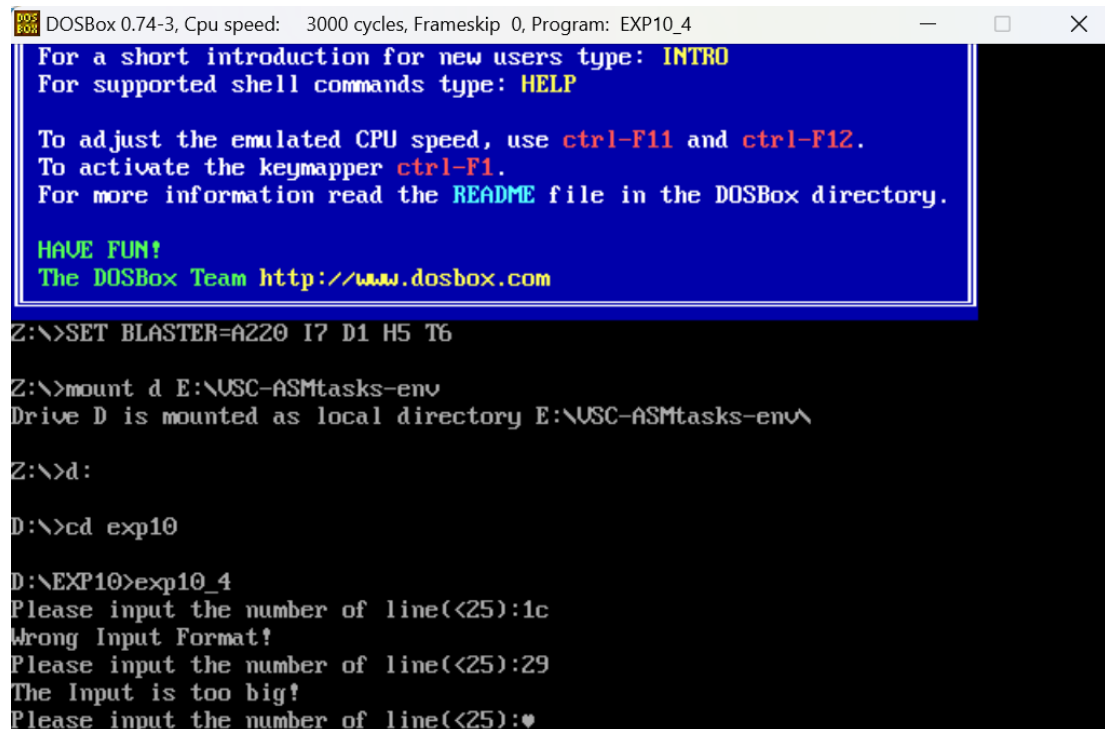
- ① 使用 BIOS 中断 `INT 10H` 设置显示模式，并使用 `INT 10H` 清屏。
- ② 循环显示太阳图案，每次显示一个太阳图案，并进行一秒的延时。
- ③ 如果按键次数达到设定的 `COUNT` 值，则退出循环，返回到主程序继续执行。

(8) 在 `SHOW2` 过程中显示提示信息：

- ① 通过 BIOS 中断 `INT 10H` 获取当前光标位置，并将光标移动到下一个位置。
- ② 根据按键次数更新提示信息中的按键次数，并根据指定的显示方式显示提示信息。

总的来说，该程序除了完成了规定的附加功能外，还实现了用户自己决定并输入循环行数，并检测输入是否有效的功能。

#### 4.实验结果:



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: EXP10_4
For a short introduction for new users type: INTRO
For supported shell commands type: HELP

To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.

HAVE FUN!
The DOSBox Team http://www.dosbox.com

Z:\>SET BLASTER=A220 I7 D1 H5 T6

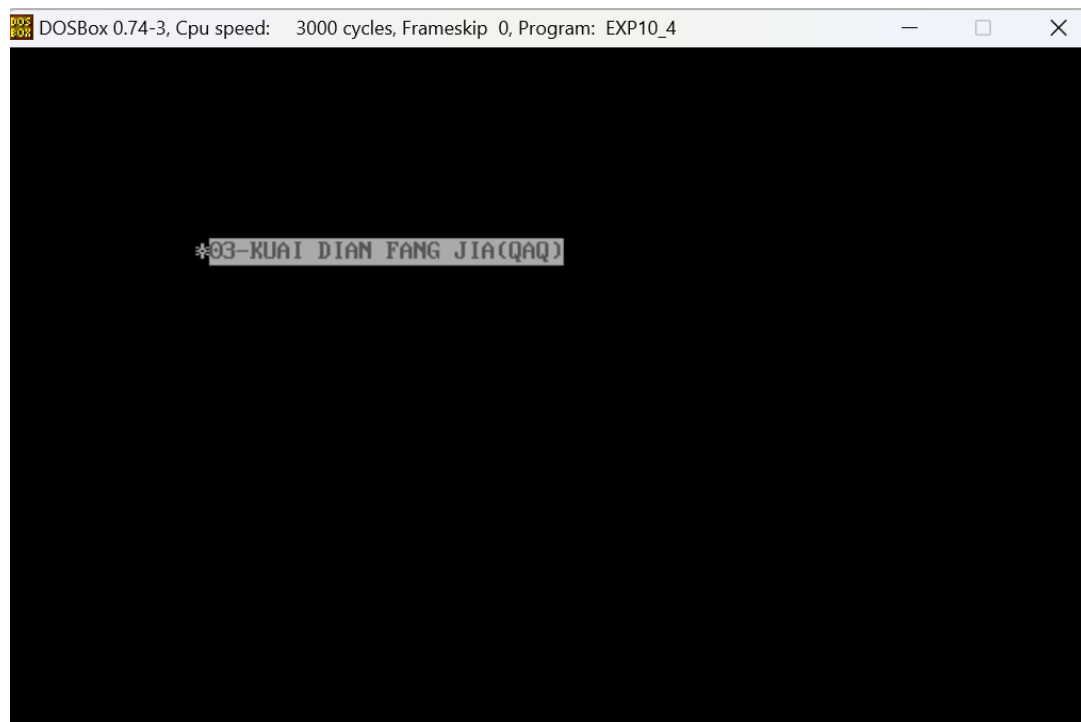
Z:\>mount d E:\USC-ASMTasks-env
Drive D is mounted as local directory E:\USC-ASMTasks-env\

Z:\>d:

D:\>cd exp10

D:\EXP10>exp10_4
Please input the number of line(<25):1c
Wrong Input Format!
Please input the number of line(<25):29
The Input is too big!
Please input the number of line(<25):♥
```

实现了检验输入行数是否出错，并相应报错的功能。



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: EXP10_4

03-KUAI DIAN FANG JIA(QAQ)
```

实现了按下按键弹出提示消息，设置其字体背景颜色，消息之间有间隔，消息不残留在屏幕，提示按键次数等功能。

## 二、实验总结（实验中遇到的问题、解决方法和实验收获）

1. 很多功能重复的片段，可以单独写成子程序，简化代码，增强可读性，可理解成高级语言里的函数调用，比较便捷易懂。

2. 问题：实验中出现 Forward needs override or far,

解决方法：由于子程序是 FAR 类型的，调用时应该用 CALL FAR PTR DISP1

3. 问题：如果按住建则会一直显示 OK，出现 N 次也不会跳出程序

解决方法：只要在中断程序显示 OK! 的程序中，在每次显示 OK! 之前先判断 OK! 是否已经超过 N 次，超过 10 次退出。

## 三、回答思考题：

1. 键盘上某个键按下和释放时都会向 8259 发出中断请求，要求只在键按下时显示‘OK!’，键释放时不显示，在中断服务程序 KEYINTS 中是如何实现的？

答：每当有键按下或释放时，键盘以串行方式向系统板的键盘接口电路传送数据，即扫描码。当键按下时，送向主机的扫描码是键编号，而键释放时，扫描码为键编号加 80H(即第 7 位置 1)。

在程序中，使用 IN 指令读取扫描码，判断第七位的状态，如果 D7=1，说明是释放键，此时使用转移指令，不让程序执行显示 OK! 就可以做到。

2. 完成附加功能 4(按键次数达到后(比如 9 次)，不等 25 行太阳图标显示完，立即返回 DOS)的同学，请简要说明：

1) 是如何实现该附加功能的？

答：在循环显示小太阳的子程序里，判断是否已经响应过 N（N 在数据段设置）次中断，如果已经响应过，则直接出栈，入栈后直接出战栈，就不执行中间的显示过程。此时在主程序中也有判断是否中断满 N 次的程序，已经满了 N 次，就在主程序里恢复原来的中断，返回 DOS。

2) 如果程序运行结束后，键盘不能正常使用（按键没反应），可能是哪些原因？

答：① 压栈和出栈（PUSH 和 POP）没有对应，如果不一一对应，会导致子程序结束时无法正常返回 CS: IP 或 IP 值。

② 在处理中断时没有一层层结束中断，而是直接返回 DOS，同样导致无法正常返回 CS: IP 或 IP 值。