**东南大学信息科学与工程学院**

**实 验 报 告**

**课程名称： 微机实验**

实验名称： 实验十 键盘中断

院 （系）： 信息学院 专 业： 信息工程

姓 名： 李宗尧 学 号： 04019715

实 验 室: 416 实验组别：

同组人员： 实验时间： 2021. 5. 19

评定成绩： 审阅教师：

**一、实验目的**

1．了解Intel 8086CPU的中断处理功能以及IBM-PC的中断结构。

2．了解8259中断控制器的使用。

3．掌握键盘中断的编程，观察中断的执行情况。

**二、实验任务**

1．基本：

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。

2．附加功能：

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）在显示”OK!”的前面增加显示按键次数；

3）按键10次后，不等25行太阳图标显示完，立即返回DOS；√

4）修改显示字符的属性，如，红底白字，蓝底黄字……

**三、实验原理**

键盘与主机是通过5芯螺旋形的电缆相连的，其中包括数据线、时钟线、复位线、+5v电源线和地线。(电缆插入系统板后部的插座)

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。一个扫描码移位传送完，键盘接口电路便向主机发出中断请求信号IRQ1(中断类型码为09H)，此信号送到8259A产生中断请求。

CPU响应中断请求时，查中断向量表，从09H×4开始的连续四个单元中取出中断向量（IRQ1中断服程序KEYINTS的入口地址指针），转去执行中断服务程序KEYINTS。

主程序和键盘中断服务程序的流程图如图l-10-1和图1-10-2所示。请根据流程图编写主程序和键盘中断服务程序。

在主程序中应先读取并保存中断类型号09H的原中断向量，然后再设置新的中断向量，即将中断服务程序KEYINTS入口地址的偏移量和段基址存入以09×4为起始地址的四个单元内。

在键盘中断程序KEYINTS中，保护现场、开中断之后，就通过8255A的PA口(PA口地址为60H)读取键盘扫描码，接着从8255A PB口 (PB口地址为61H)的 PB7输出一个正脉冲（即PB7先输出高电平，再输出低电平），先输出的高电平信号反相之后控制键盘状态触发器的清零端，使IRQ1清零，撤消中断请求信号。再输出的低电平信号允许位寄存器输出数据，这样就为传递下一个键盘扫描码作好了准备。

8255A PB口的PB7 输出正脉冲来发键盘状态复位命令的具体指令如下；

IN AL，61H ；输入PB口的当前值

OR AL，80H ；PB7置1

OUT 61H，AL

AND AL，7FH ；PB7清零

OUT 61H，AL

在键盘中断程序结束前，应发出“中断结束”(EOI)命令(控制字为20H)，给8259A的操作命令字OCW1(端口地址为20H)。具体指令如下；

MOV AL，20H

OUT 20H，AL

然后,实现中断返回。

另外，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)。例如,按下和释放“A”键将向主机发送两个扫描码：1EH和9EH。



以下是主程序中显示“太阳”图形的子程序DISPl，仅供参考。

DISPl PROC FAR

PUSH AX

PUSH BX

PUSH CX

PUSH DX

MOV AH，15 ; 读当前显示状态

INT 10H

MOV AH，0 ; 设置显示方式

INT 10H

MOV DX，0 ; 行号为0，列号为0

REPT: MOV AH，2 ; 设置光标位置

INT 10H

MOV AL，0FH ; 0FH--太阳图形的ASCII码

MOV CX，1 ; 显示字符个数

MOV AH，10 ; 写字符

INT 10H

CALL DELAY

SUB AL，AL

MOV AH，10 ; 清除原图形

INT 10H

INC DH ; 行号+1

ADD DL，2 ; 列号+2

CMP DH，25 ; 是否到25行？

JB REPT

POP DX

POP CX

POP BX

POP AX

RET

DISPl ENDP

以下是中断服务程序中显示"OK！"字符子程序DISP2，仅供参考。

DISP2 PROC FAR

PUSH CX

PUSH BX

PUSH AX

MOV CX, 3

NEXTC：LODSB ; 字符串"OK！"在数据段中定义，AL←[SI]

MOV AH，0EH ; 写字符，并移动光标

MOV BX，01

INT 10H

CALL DELAY

LOOP NEXTC

POP AX

POP BX

POP CX

RET

DISP2 ENDP

以下是延时1秒子程序

DELAY PROC

PUSH CX

PUSH DX

MOV DX，20

DL500： MOV CX，2801

DL10ms：LOOP DL10ms

DEC DX

JNZ DL500

POP DX

POP CX

RET。

DELAY ENDP

**四、程序源代码**（橙色部分为附加任务1，绿色部分为附加任务2，蓝色部分为附加任务3，紫色部分为附加任务4；红色部分为思考题内容）

DATA SEGMENT

KEY\_COUNTS DB ?

OK\_PRINT DB " OK! " ;字符串中的空格为按键次数预留

DATA ENDS

STACK SEGMENT ;为保护现场操作建立栈段空间

DW 100 DUP(?)

STACK ENDS

CODE SEGMENT

ASSUME CS:CODE,DS:DATA,ES:DATA,SS:STACK

START:MOV AX,STACK

MOV SS,AX

MOV AX,DATA

MOV DS,AX

MOV ES,AX

MOV AL,09H ;用中断类型21H的35H功能取中断向量保存

MOV AH,35H

INT 21H

PUSH ES

PUSH BX

MOV KEY\_COUNTS,00H ;按键次数置零

MOV DX,OFFSET KEYINTS ;用中断类型21H的25H功能设置中断向量

MOV AX,SEG KEYINTS

MOV DS,AX

MOV AL,09H

MOV AH,25H

INT 21H

MOV AX,DATA ;恢复数据段地址

MOV DS,AX

STI ;开中断

SUN:CALL DISPl ;输出太阳符号

CMP KEY\_COUNTS,10 ;比较按键次数

JB SUN ;未到10，继续输出太阳

CLI ;关中断

POP DX ;用中断类型21H的25H功能恢复中断向量

POP DS

MOV AL,09H

MOV AH,25H

INT 21H

MOV AH,4CH ;返回DOS

INT 21H

KEYINTS PROC NEAR

PUSH AX ;保护现场

PUSH BX

PUSH DX

PUSH SI

STI ;开中断

IN AL,60H ;读取键盘扫描码

MOV AH,AL ;保护键盘扫描码

IN AL,61H ;输入PB口的当前值

OR AL,80H ;PB7置1，产生中断请求信号

OUT 61H,AL

AND AL,7FH ;PB7置0，为下一次读取扫描码做准备

OUT 61H,AL

TEST AH,80H ;根据扫描码判断按键为按下还是松开

JNZ PASS ;若为松开状态，则不计按键数，不输出OK

INC KEY\_COUNTS ;按键数+1

CMP KEY\_COUNTS,10 ;保证按键达到十次后结束程序

JA PASS

MOV AL, KEY\_COUNTS

ADD AL, 30H ;将按键次数转换为ASCII码，以便输出

CMP AL, 3AH

JNE PRINT

MOV OK\_PRINT, 31H ;按键次数为10的ASCII码

MOV OK\_PRINT+1, 30H

JMP CHGSI

PRINT: MOV OK\_PRINT, AL

CHGSI: LEA BP,OK\_PRINT

MOV AX, SEG OK\_PRINT ;为DISP2中进行INT 10H中断作预处理

MOV ES, AX

CALL DISP2

PASS:MOV AL,20H ;发出中断结束命令

OUT 20H,AL

POP SI ;恢复现场

POP DX

POP BX

POP AX

STI ;开中断

IRET ;中断返回

KEYINTS ENDP

DISPl PROC NEAR

PUSH AX ;保护现场

PUSH BX

PUSH CX

PUSH DX

MOV AH,15 ;读当前显示状态

INT 10H

MOV AH,0 ;设置显示方式

INT 10H

MOV CX,1 ;设置显示字符的个数

MOV DX,0 ;设置行列为0

REPT:MOV AH,2 ;设置光标位

INT 10H

CMP KEY\_COUNTS,10 ;比较按键次数

JNB PASS1 ;次数达到要求则不再输出太阳

MOV AL,0FH ;读出太阳图形

MOV AH,10 ;设置功能号，写字符

INT 10H

CALL DELAY ;调用延时子程序

SUB AL,AL

MOV AH,10 ;清除原图形

INT 10H

INC DH ;设置下一次输出图形的位置

ADD DL,2

CMP DH,25 ;未输出完1页以前，持续输出

JNE REPT

PASS1:POP DX ;恢复现场

POP CX

POP BX

POP AX

RET

DISPl ENDP

DISP2 PROC NEAR

PUSH CX ;保护现场

PUSH BX

PUSH AX

PUSH DX

MOV BH, 00H

MOV AH, 03H ;获取光标位置

INT 10H

MOV CX,7 ;包含了按键次数和空格，故需要七个

MOV AH,13H ;用AH=13H的INT 10H中断改变字体颜色

MOV BL,4FH

MOV BH, 00H

MOV AL, 01H

INT 10H

CALL DELAY

POP DX

POP AX ;恢复现场

POP BX

POP CX

RET

DISP2 ENDP

DELAY PROC NEAR ;延时子程序

PUSH CX

PUSH DX

MOV DX,120

DL500:MOV CX,2801

DL10ms:LOOP DL10ms

DEC DX

JNZ DL500

POP DX

POP CX

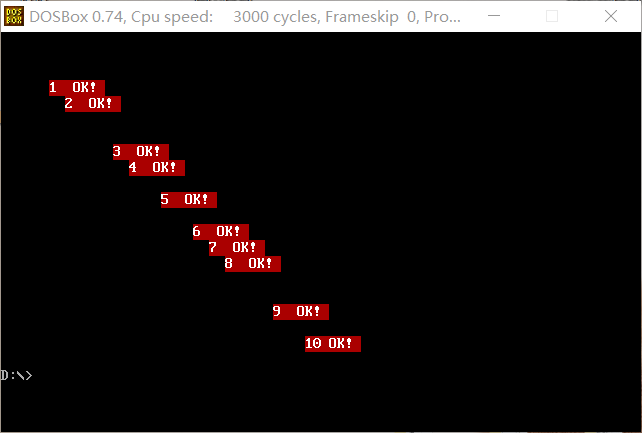
RET

DELAY ENDP

CODE ENDS

END START

**调试结果：**



**四、思考题与实验总结**

**1、思考题：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，则中断服务程序KEYINTS应如何修改?**

答：可以根据键盘上按键按下和释放时的键盘扫描码第七位不同，进而判断按键的状态，具体操作如下：读取键盘扫描码（IN AL,60H）后保留（MOV AH,AL），之后根据扫描码的第七位（TEST AH,80H）：是0则为按下状态，计数加1并输出“OK！”；是1则为松开，不再计数及输出“OK！”（JNZ PASS）。这样即可实现要求功能。

**2、实验中遇到的问题、分析原因及解决方法**

①在实现附加功能二的过程中，起初观察到输出的“OK!”字符不全，发现原因是字符串长度改变，但进行INT 10H中断时未调整CH。调整后解决了问题。

②在实现附加功能四，即修改字符的属性时，起初想要利用功能号为09H的INT 10H中断，但是按照INT 10H中断操作设置不同颜色后，输出的字符始终是黑白色。经过不断的调研，发现功能号为09H的INT 10H中断中，若要设置字符颜色和背景颜色，需要在图形模式下进行，但是本实验其他功能的实现需要在文本模式下实现，于是无法使用功能号为09H的INT 10H中断。调研后发现可以用功能号为13H的INT 10H中断（输出字符串）代替功能号为09H的INT 10H中断，且该中断设置颜色时没有工作模式限制，于是，使用功能号为13H的INT 10H中断终于实现了设置字符颜色和背景颜色的功能。