实验十 键盘中断

学号：61822313 姓名：钟锦程 实验日期：2024 年 4 月 29 日

# 实验任务和实验结果

## 基础任务

* + 1. 实验任务的具体内容

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。

* + 1. 调试通过的源程序

1

**DATA SEGMENT**

**OKSTR DB 'OK!',0DH,0AH,'$' TIME DB 00H**

**NINE EQU 9**

**DATA ENDS*;TIME***是按键次数 **STACK SEGMENT**

**DW 200H DUP(?) STACK ENDS**

**CODE SEGMENT**

**ASSUME CS:CODE,DS:DATA,SS:STACK DELAY PROC *;*** 延时程序

**PUSH CX PUSH DX MOV DX,36H**

**DL500:**

**MOV CX, 08FFFH DL10MS:**

**LOOP DL10MS DEC DX**

**JNZ DL500**

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

1

1. **POP DX**
2. **POP CX**
3. **RET**
4. **DELAY ENDP**

25

1. **DELAY2 PROC *;*** 延时程序***2,***用于显示***OK***时的延时，略短于***1***秒
2. **PUSH CX**
3. **PUSH DX**
4. **MOV DX,16H**

30 **DL5002:**

1. **MOV CX, 01FFFH**
2. **DL10MS2:**
3. **LOOP DL10MS2**
4. **DEC DX**
5. **JNZ DL5002**
6. **POP DX**
7. **POP CX**
8. **RET**
9. **DELAY2 ENDP**
10. **DISP1 PROC FAR *;*** 显示太阳

|  |  |  |
| --- | --- | --- |
| 41 | **PUSH** | **AX** |
| 42 | **PUSH** | **BX** |
| 43 | **PUSH** | **CX** |
| 44 | **PUSH** | **DX** |
| 45 | **MOV** | **AH,15 *;*** 读当前显示状态 |
| 46 | **INT** | **10H** |
| 47 | **MOV** | **AH,0 *;*** 设置显示方式 |
| 48 | **INT** | **10H** |
| 49 | **MOV** | **DX,0 *;*** 行号为***0***，列号为***0*** |
| 50 | **REPT:** |  |
| 51 | **MOV** | **AH,2 *;*** 设置光标位置 |
| 52 | **INT** | **10H** |
| 53 | **MOV** | **AL,0FH *; OFH***是太阳图形的***ASCII***码 |
| 54 | **MOV** | **CX,1 *;*** 重复字符的次数 |
| 55 | **MOV** | **AH,10 *;*** 写字符 |
| 56 | **INT** | **10H** |
| 57 | **CALL** | **DELAY** |
| 58 | **SUB** | **AL,AL** |
| 59 | **MOV** | **AH,0 *;*** 清除原图形 |

|  |  |  |
| --- | --- | --- |
| 60 | **INT** | **10H** |
| 61 | **INC** | **DH *;*** 行号***+1*** |
| 62 | **ADD** | **DL,2 *;*** 列号***+2*** |
| 63 | **CMP** | **DH,20 *;*** 判断是否到***20***行，不等继续显示太阳，相等返回，如果到***25***行再返回，会 |
|  |  | 导致轨迹不是严格对角 |
| 64 | **JNE** | **REPT** |
| 65 | **POP** | **DX** |
| 66 | **POP** | **CX** |
| 67 | **POP** | **BX** |
| 68 | **POP** | **AX** |
| 69 | **RET** |  |

1. **DISP1 ENDP**
2. **DISP2 PROC FAR *;*** 显示***OK***
3. **PUSH CX**
4. **PUSH BX**
5. **PUSH AX**
6. **MOV CX,3*;***待显示的字符数
7. **NEXTC:**
8. **LODSB *;*** 字符串***”OK!”***在数据段中定义，***AL<***—***[SI]***
9. **MOV AH, 0EH *;*** 用***teletype***格式写字符，并移动光标
10. **MOV BX,1**
11. **INT 10H**
12. **CALL DELAY2**
13. **LOOP NEXTC**
14. **POP AX**
15. **POP BX**
16. **POP CX**
17. **RET**
18. **DISP2 ENDP**

88

89 **KEYINT PROC FAR**

|  |  |  |
| --- | --- | --- |
| 90 | **PUSH** | **AX** |
| 91 | **PUSH** | **BX** |
| 92 | **PUSH** | **DX** |
| 93 | **STI** |  |

1. **IN AL,60H *;*** 通过***8255A***的***PA***口***(PA***口地址为***60H)***读取键盘扫描码
2. **MOV AH,AL**
3. **IN AL,61H *;*** 从***8255APB***口***(PB***口地址为***61H)***的***PB7***输出一个正脉冲
4. **OR AL,80H *; PB7***置***1***
5. **OUT 61H,AL**
6. **AND AL,7FH *; PB7***清零
7. **OUT 61H,AL *;***再输出一个负脉冲
8. **TEST AH,80H *;*** 相等时代表键被释放，开中断，显示字符
9. **JNE BACK *;*** 不等，中断结束返回
10. **STI**
11. **INC TIME**
12. **MOV SI,OFFSET OKSTR *;*** 初始化***SI***，在***DISP2***中不再对***SI***进行初始化
13. **CALL DISP2**
14. **BACK:**
15. **MOV AL,20H**
16. **OUT 20H,AL**
17. **POP DX**
18. **POP BX**
19. **POP AX**
20. **IRET**
21. **KEYINT ENDP**
22. **START:**
23. **MOV AX,DATA**
24. **MOV DS,AX**
25. **MOV AX,STACK**
26. **MOV SS,AX**
27. **MOV AX,0**
28. **MOV ES,AX**
29. **MOV SI,OFFSET OKSTR**
30. **MOV AX,ES:[24H] *; 9∗4=24H,***压入中断向量的偏移地址
31. **PUSH AX**
32. **MOV AX,ES:[26H] *;***压入中断向量的段地址
33. **PUSH AX**
34. **CLI*;***关中断，载入自定义中断子程序地址时不允许中断
35. **MOV AX,SEG KEYINT**
36. **MOV ES:[26H],AX**
37. **MOV AX,OFFSET KEYINT**
38. **MOV ES:[24H],AX**
39. **STI**
40. **AGIN:**
41. **CALL DISP1**
42. **CMP TIME,10**
43. **JB AGIN**

|  |  |  |
| --- | --- | --- |
| 137 | **CLI** | ***;*** 禁止下方程序中断发生，保护代码运行 |
| 138 | **POP** | **AX** |
| 139 | **MOV** | **ES:[26H],AX*;***弹出原中断向量的段地址 |
| 140 | **POP** | **AX** |
| 141 | **MOV** | **ES:[24H],AX*;***弹出原中断向量的偏移地址 |
| 142 | **STI** | ***;*** 开中断 |
| 143 | **MOV** | **AH,4CH** |
| 144 | **INT** | **21H** |
| 145  146 | **CODE ENDS END START** | |

* + 1. 实验结果

如图[1](#_bookmark0)是程序正常运行的截图，如果不按键，则当图标走到右下角后会从左上角重新开始，循环往复。图[2](#_bookmark1)是按一次键的运行截图。可以看到在本应显示太阳符号的地方显示”OK!”，如果多次按键（或长按），会不间断显示”OK!”（如图[3](#_bookmark2)），直到所有”OK!” 显示完毕才会继续显示太阳符号。当按键总次数达到 10 次时，程序会在本次太阳显示循环结束后中止程序，如图[4](#_bookmark3)。

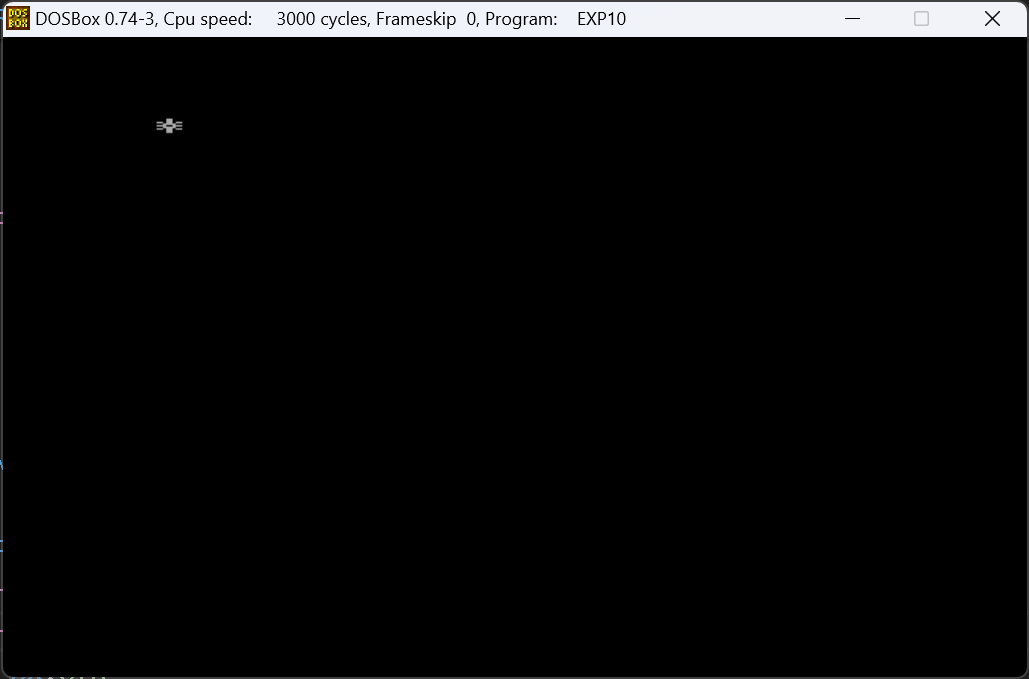
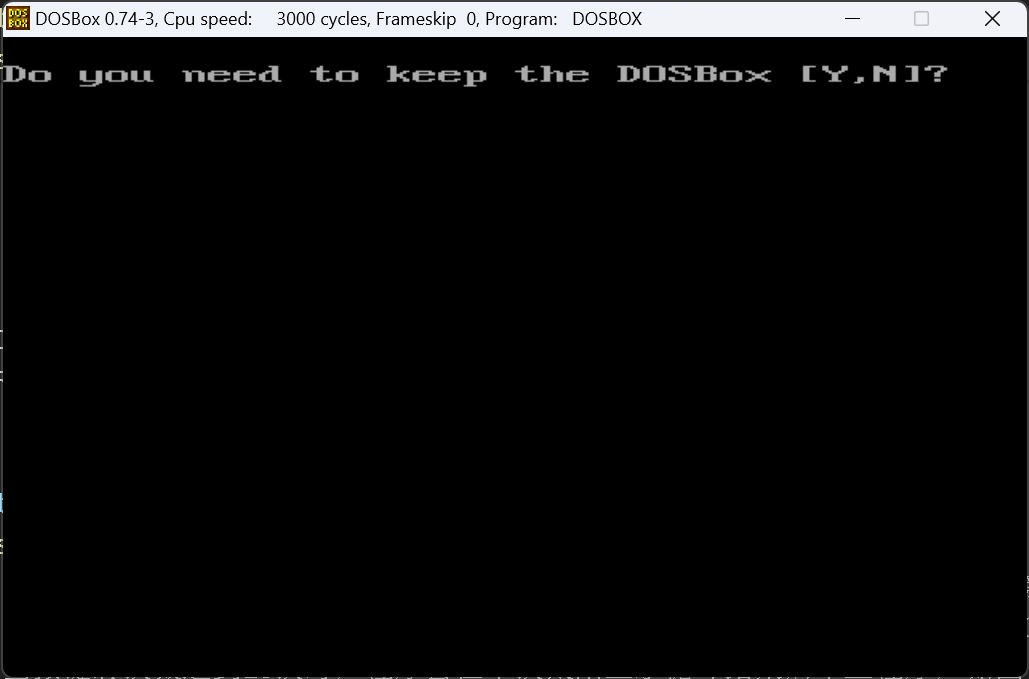
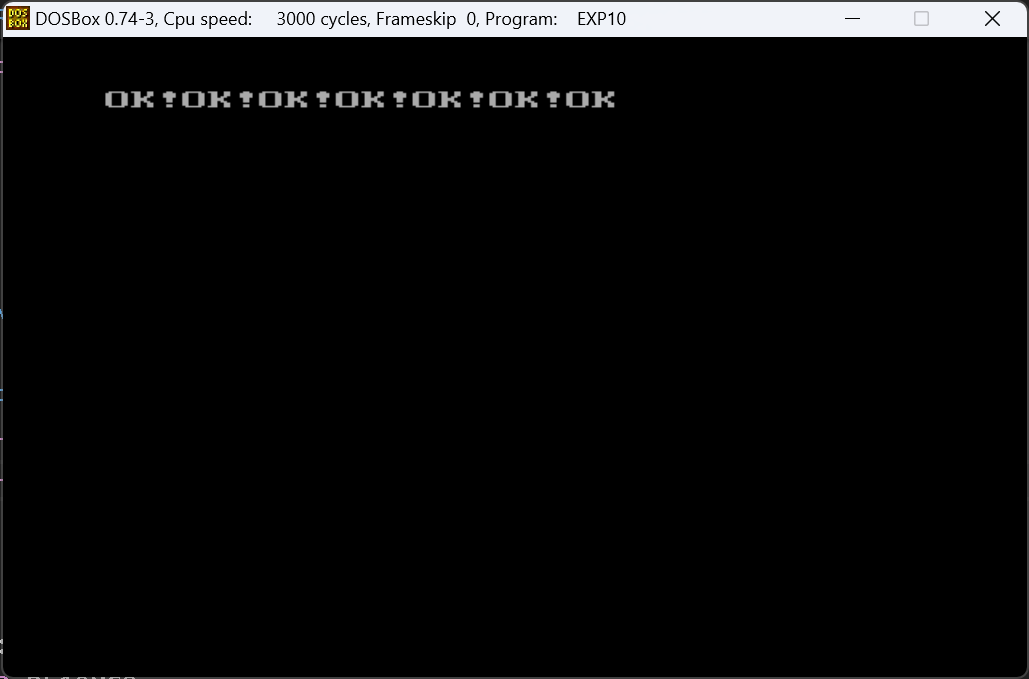


图 1: 基础实验任务结果截图 1 图 2: 基础实验任务结果截图 2

图 3: 基础实验任务结果截图 1 图 4: 基础实验任务结果截图 2



## 附加任务

* + 1. 实验任务的具体内容
       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；
       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
    2. 调试通过的源程序

1

**DATA SEGMENT TIME DB 00H**

**SHOWTIME DB 00H*;SHOWTIME***存放***TIME***对应的***ASCII***码 **OKSTR DB 'HI,ZJC!',20H**

**N EQU $-SHOWTIME**

**COUNT DB N*;COUNT***用于在显示过程中计数递减，便于直接中断而不用等本次太阳循环结束

**NINE EQU 9*;***定义数值常量便于调试 **DATA ENDS*;TIME***是按键次数

**STACK SEGMENT**

**DW 200H DUP(?) STACK ENDS**

2

3

4

5

6

7

8

9

10

11

|  |  |
| --- | --- |
| 12 | **CODE SEGMENT** |
| 13 | **ASSUME CS:CODE,DS:DATA,SS:STACK** |
| 14 | **DELAY PROC *;*** 延时程序 |
| 15 | **PUSH CX** |
| 16 | **PUSH DX** |
| 17 | **MOV DX,36H** |
| 18 | **DL500:** |
| 19 | **MOV CX, 08FFFH** |
| 20 | **DL10MS:** |
| 21 | **LOOP DL10MS** |
| 22 | **DEC DX** |
| 23 | **JNZ DL500** |
| 24 | **POP DX** |
| 25 | **POP CX** |
| 26 | **RET** |
| 27 | **DELAY ENDP** |
| 28 |  |
| 29 | **DELAY2 PROC *;*** 延时程序***2*** |
| 30 | **PUSH CX** |
| 31 | **PUSH DX** |
| 32 | **MOV DX,16H** |
| 33 | **DL5002:** |
| 34 | **MOV CX, 01FFFH** |
| 35 | **DL10MS2:** |
| 36 | **LOOP DL10MS2** |
| 37 | **DEC DX** |
| 38 | **JNZ DL5002** |
| 39 | **POP DX** |
| 40 | **POP CX** |
| 41 | **RET** |
| 42 | **DELAY2 ENDP** |
| 43 |  |
| 44 | **DISP1 PROC FAR *;*** 显示太阳 |
| 45 | **PUSH AX** |
| 46 | **PUSH BX** |
| 47 | **PUSH CX** |
| 48 | **PUSH DX** |
| 49 | **MOV AH,15 *;*** 读当前显示状态 |
| 50 | **INT 10H** |

|  |  |  |
| --- | --- | --- |
| 51 | **MOV** | **AL,1** |
| 52 | **MOV** | **AH,0 *;*** 设置显示方式 |
| 53 | **INT** | **10H** |
| 54 | **MOV** | **DX,0 *;*** 行号为***0***，列号为***0*** |
| 55 | **REPT:** |  |
| 56 | **MOV** | **AH,2 *;*** 设置光标位置 |
| 57 | **INT** | **10H** |
| 58 | **MOV** | **AL,0FH *; OFH***是太阳图形的***ASCII***码 |
| 59 | **MOV** | **CX,1 *;*** 重复字符的次数 |
| 60 | **MOV** | **BL,14H *;***字符颜色信息为白底红字加闪烁（***b7***控制字符是否闪烁，***b6−b4***为背景 |
| 61 | **MOV** | 色，***b3−b0***为前景色）  **AH,9 *;*** 写字符 |
| 62 | **INT** | **10H** |

1. **CALL DELAY**
2. **SUB AL,AL**
3. **MOV AH,0 *;*** 清除原图形
4. **INT 10H**
5. **CMP TIME,NINE*;***当次数达到设定值时，直接结束显示
6. **JAE QUIT1**
7. **INC DH *;*** 行号***+1***
8. **ADD DL,2 *;*** 列号***+2***
9. **CMP DH,20 *;*** 判断是否到***20***行，不等则继续显示太阳，相等返回
10. **JNE REPT**
11. **QUIT1:**
12. **POP DX**
13. **POP CX**
14. **POP BX**
15. **POP AX**
16. **RET**
17. **DISP1 ENDP**

80

1. **DISP2 PROC FAR *;*** 中断显示信息
2. ***; INT 10H*** 中***AH=03H***读取光标信息。入口参数：***BH=***显示页码
3. ***;*** 出口参数：***CH=***光标的起始行 ***CL=***光标的终止行 ***DH=***行***(Y***坐标***) DL=***列 ***(X***坐标***)***

|  |  |  |
| --- | --- | --- |
| 84 | **PUSH** | **DX** |
| 85 | **PUSH** | **CX** |
| 86 | **PUSH** | **BX** |
| 87 | **PUSH** | **AX** |

1. **MOV COUNT,N*;***用待显示的字符串的长度更新***COUNT***
2. **NEXTC:**
3. **LODSB *;*** 字符串在数据段中定义并已经由***SI***指向首址
4. **MOV BH,00H**
5. **MOV AH,03H**
6. **INT 10H**
7. **MOV AH,9 *;***写字符
8. **MOV CX,1 *;***重复一次
9. **MOV BL,24H *;***字符颜色信息为绿底红字
10. **INT 10H**
11. **INC DL *;***每打印***1***个字符，光标的列号***+1***
12. **MOV AH,2 *;*** 更新光标位置
13. **INT 10H**
14. **CALL DELAY2**
15. **DEC COUNT**
16. **CMP COUNT,0 *;***当整个中断信息串都显示完，才退出循环
17. **JA NEXTC**
18. **POP AX**
19. **POP BX**
20. **POP CX**
21. **POP DX**
22. **RET**
23. **DISP2 ENDP**

111

1. **KEYINT PROC FAR**
2. **PUSH AX**
3. **PUSH BX**
4. **PUSH DX**
5. **STI**
6. **IN AL,60H *;*** 通过***8255A***的***PA***口***(PA***口地址为***60H)***读取键盘扫描码
7. **MOV AH,AL**
8. **IN AL,61H *;*** 从***8255APB***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）
9. **OR AL,80H *; PB7***置***1***
10. **OUT 61H,AL**
11. **AND AL,7FH*; PB7***清零
12. **OUT 61H,AL**
13. **TEST AH,80H *;*** 相等时代表键被释放，开中断，显示字符
14. **JNE BACK**
15. **STI**

|  |  |  |
| --- | --- | --- |
| 127 | **INC** | **TIME** |
| 128 | **MOV** | **AL,0** |
| 129 | **ADD** | **AL,TIME** |

1. **OR AL,30H**
2. **MOV SI,OFFSET SHOWTIME *;*** 初始化***SI,***使其指向中断信息字符串首址
3. **MOV [SI],AL**
4. **CALL DISP2**
5. **BACK:**

|  |  |  |
| --- | --- | --- |
| 135 | **MOV** | **AL,20H** |
| 136 | **OUT** | **20H,AL*;***结束中断 |
| 137 | **POP** | **DX** |
| 138 | **POP** | **BX** |
| 139 | **POP** | **AX** |
| 140 | **IRET** |  |
| 141 | **KEYINT ENDP** | |
| 142 |  | |
| 143 **ST** | **ART:** | |
| 144 | **MOV AX,DATA** | |
| 145 | **MOV DS,AX** | |
| 146 | **MOV AX,STACK** | |
| 147 | **MOV SS,AX** | |
| 148 | **MOV AX,0** | |
| 149 | **MOV ES,AX** | |
| 150 | **MOV SI,OFFSET TIME** | |
| 151 | **MOV AH,35H *;INT 21H 35H***号：取中断向量，入口***AL=***中断类型，出口***ES:BX=***中 | |
|  | 断向量 | |
| 152 | **MOV AL,9** | |
| 153 | **INT 21H** | |
| 154 | **PUSH BX *;***将原中断处理程序的***CS:IP***压栈 | |
| 155 | **PUSH ES** | |
| 156 | **MOV AX,0 *;***恢复***ES!***重要***!*** | |
| 157 | **MOV ES,AX** | |
| 158 | **PUSH DS *;***保护***DS***与***DX!***重要***!*** | |
| 159 | **PUSH DX** | |
| 160 | **CLI** | |
| 161 | **MOV AX,SEG KEYINT *;***将自定义中断服务程序的地址放入原***09***号中断向量处 | |
| 162 | **MOV DS,AX** | |
| 163 | **MOV DX,OFFSET KEYINT** | |
| 164 | **MOV AL,9** | |

|  |  |  |
| --- | --- | --- |
| 165 | **MOV** | **AH,25H**  类型号 |
| 166 | **INT** | **21H** |
| 167 | **STI** |  |
| 168 | **POP** | **DX** |
| 169 | **POP** | **DS** |

|  |  |
| --- | --- |
| 170 **AGIN:** | |
| 171 | **CALL DISP1** |
| 172 | **CMP TIME,NINE** |
| 173 | **JB AGIN** |
| 174 | **CLI *;*** 禁止下方程序中断发生，保护代码运行 |
| 175 | **POP DS *;***恢复原***09***号中断向量 |
| 176 | **POP DX** |
| 177 | **MOV AL,9** |
| 178 | **MOV AH,25H** |
| 179 | **INT 21H** |
| 180 | **STI *;*** 开中断 |
| 181 | **MOV AH,4CH** |
| 182 | **INT 21H** |
| 183  184 | **CODE ENDS END START** |

* + 1. 实验结果

***;INT 21H 25H***号：设置中断向量，入口***DS:DX=***中断向量，***AL=***中断

如图[5](#_bookmark4)是程序正常运行的截图，太阳符号被修改为白底红字加闪烁（附加功能 5、6）。图[6](#_bookmark5)是按一次键的运行截图。显示了绿底红字的中断信息”HI!ZJC” 并在该字符串前面显示了按键次数（附加功能 2）。图[7](#_bookmark6)是长时间按键的截图，程序不断打印中断信息，信息之间有空格（附加功能 3）。图[8](#_bookmark7)是按键次数达到 9 次时的截图，可以看到程序直接退出返回 DOS，而非等到本次太阳显示完才结束（附加功能 4）。（附加功能 1 可从程序源代码中看到）

图 5: 附加实验任务结果截图 1 图 6: 附加实验任务结果截图 2

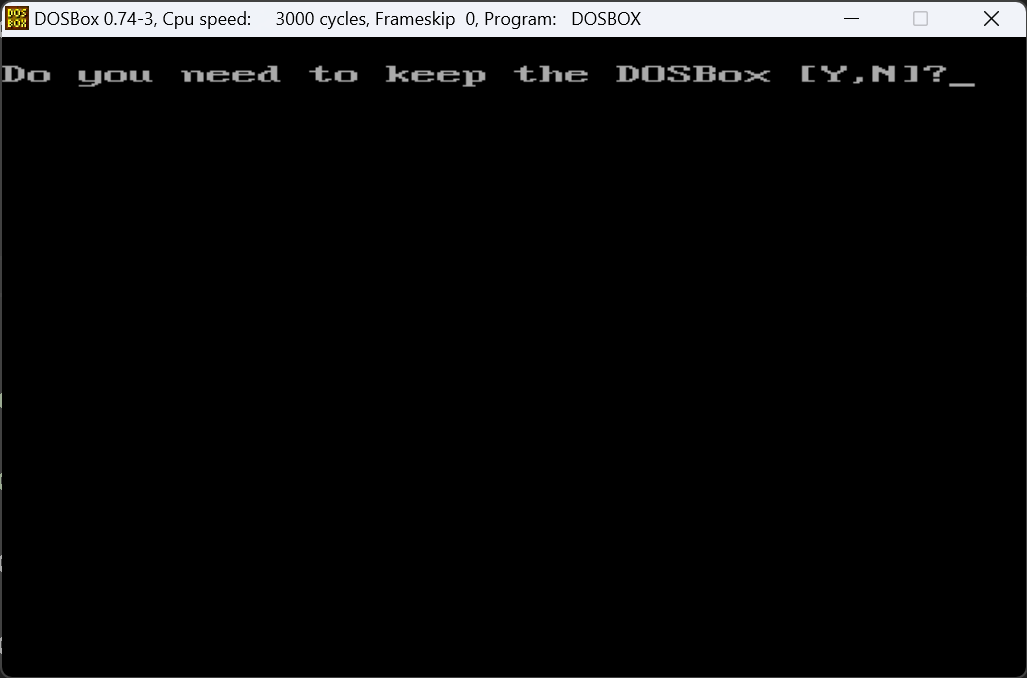
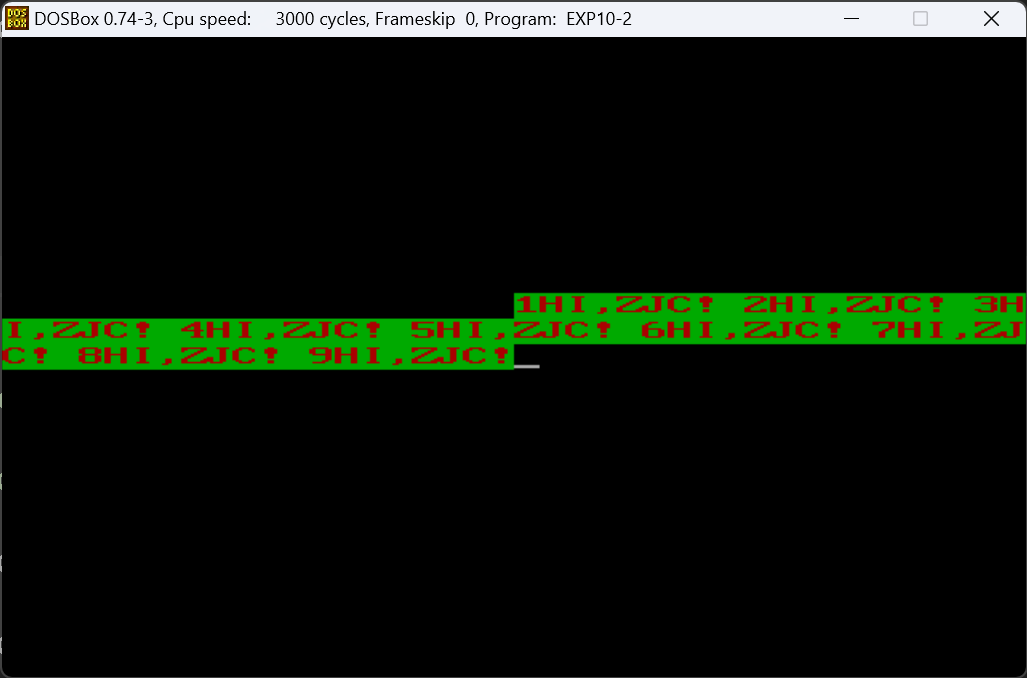
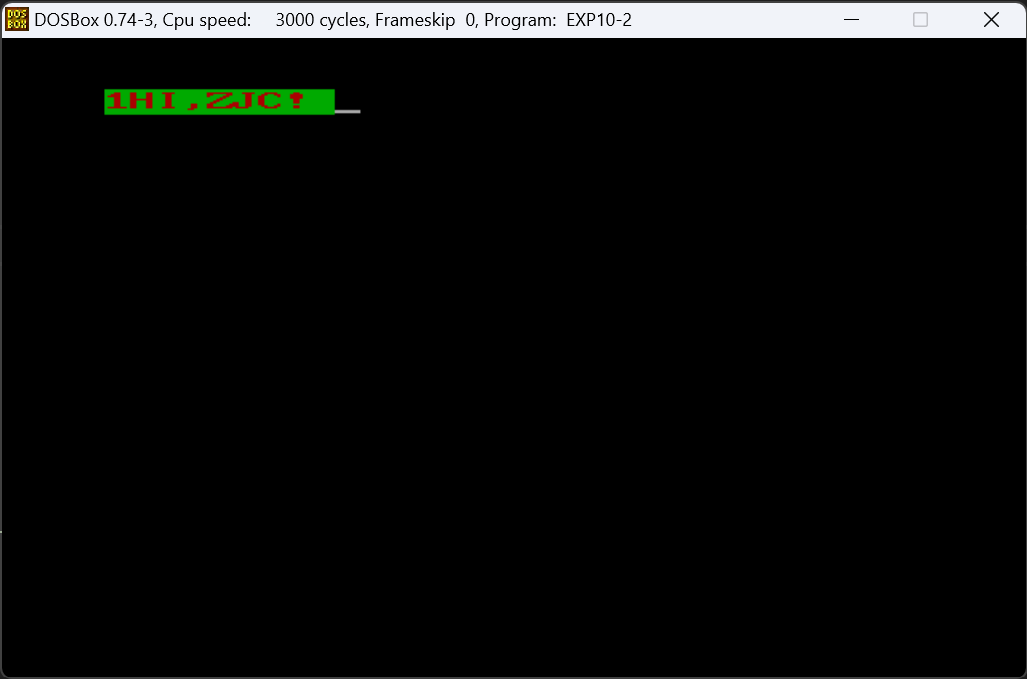
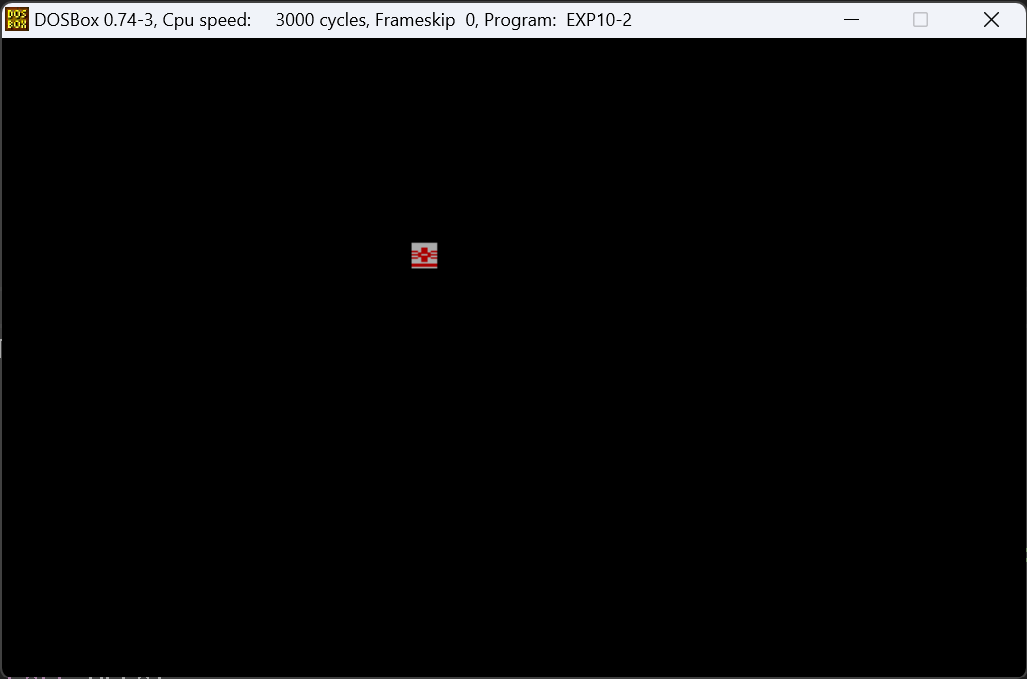


图 7: 附加实验任务结果截图 3 图 8: 附加实验任务结果截图 4

# 实验总结

本次实验首次接触键盘中断，对自定义中断服务程序的流程不熟悉，并且还没有形成保护现场的习惯。在用 DOS 中断服务 (AH=25H INT 21H) 编写中断服务程序 KEYINT 时，由于没有保护 ES 的值，导致读取中断向量后 ES 被改变，中断程序跳转错误。在输入键扫描码，键状态触发器复位的设计过程中也遇到了不小的困难。

另外，在本次实验中较为细致的学习了屏幕显示的相关知识，比如设置显示参数，控制光标等等。编写程序时比较麻烦的一点就是对光标的控制。如果要自定义字符属性就不能使用 Teletype 方法打印，而必须自己手动控制光标位置。

# 思考题

1. 当键盘上的一个按键按下时，键盘会发送一个中断信号给 CPU, 与此同时，键盘会在指定端口 (0x60) 输出一个数值 a，当按键弹起时，键盘又给端口输出一个数值 b. 通过查

表得，a 的最高位都是 0，b 的最高位都是 1，因此若按键释放，指令 TEST AH,80H 使

ZF=1，后续 JNE 指令跳转到 BACK，直接退出 KEYINT，不显示字符。

1. 附加功能如何实现：修改 DISP1 子程序，在 REPT 部分加入指令 CMP TIME,NINE 和

JAE QUIT1，当按键达到指定次数时直接跳转到退出部分，而不需等待本次显示结束。

1. 键盘不能正常使用的可能原因：在程序末尾没有恢复原 09 号中断向量的内容，导致键盘中断处理异常。