

9.8

同:

均能中断主程序执行本程序，然后返回断点继续执行子程序。

异:

1. 中断服务程序入口地址是固定的，子程序调用入口地址是用户自己设定的。
2. 中断服务子程序返回指令，除具有子程序返回指令所具有的全部功能之外，还有清除中断相应时被置位的优先级状态，比较较低中断和恢复中断逻辑等功能。
3. 中断服务子程序在满足中断申请的条件下随时发生的，而子程序调用是用户主程序事先安排好的。

9.12

- 1、在写入中断向量之前，应该考虑所采用的中断类型是否已经被系统或其他程序占用。
如果与其有冲突,应将原有类型号的中断向量取出保存起来，等中断调用结束之后再恢复它。
- 2、写好用户自定义的中断程序，然后将该中断程序的中断向量写入中断向量表中。
- 3、可以使用中断驻留，让其他程序也可以调用用户自定义的中断。
- 4、通过调用中断或者触发中断来实现效果。

9.32

DATAS SEGMENT

MSG DB 'HELLO!\$'

DATAS ENDS

CODES SEGMENT

ASSUME CS:CODES,DS:DATAS,SS:STACKS

START:

MOV AX,DATAS

MOV DS,AX

;清屏

MOV AH,6

MOV AL,0

MOV BH,07H ;黑底白字

MOV CH,0 ;左上(0,0)

MOV CL,0

MOV DH,24

MOV DL,79 ;右下(24,79)->25 * 80

INT 10H

;移动光标到开头

MOV AH,2

MOV DH,0

MOV DL,0

MOV BH,0

INT 10H

LEA DX,MSG

MOV AH,9

INT 21H

;移动输入光标到第二行开头

MOV AH,2

MOV DH,1

MOV DL,0

MOV BH,0

INT 10H

;输入

MOV AL,0

MOV AH,10

INT 21H

MOV AH,4CH

INT 21H

CODES ENDS

END START

9.34

DATAS SEGMENT

MSG1 DB 'DATE : \$'

MSG2 DB ' TIME : \$'

Y DW ?

M DB ?

D DB ?

```
H DB ?  
MIN DB ?  
S DB ?  
P DB ?  
DATAS ENDS
```

```
CODES SEGMENT
```

```
    ASSUME CS:CODES,DS:DATAS
```

```
START:
```

```
    MOV AX,DATAS
```

```
    MOV DS,AX
```

```
    ;清屏
```

```
    MOV AH,6
```

```
    MOV AL,0
```

```
    MOV BH,07H    ;黑底白字
```

```
    MOV CH,0      ;左上(0,0)
```

```
    MOV CL,0
```

```
    MOV DH,24
```

```
    MOV DL,79     ;右下(24,79)->25 * 80
```

```
    INT 10H
```

```
    ;打开窗口
```

```
    MOV AH,6
```

```
    MOV AL,0
```

```
    MOV BH,70H
```

```
    MOV CH,5
```

```
    MOV CL,10
```

```
    MOV DH,14
```

```
    MOV DL,50
```

```
    INT 10H
```

```
    ;移动光标到5行10列
```

```
    MOV AH,2
```

```
    MOV DH,5
```

```
    MOV DL,10
```

```
    MOV BH,0
```

```
    INT 10H
```

```
    MOV AH,9
```

```
    LEA DX,MSG1
```

```
    INT 21H
```

```
    ;显示日期
```

```
    MOV AH,2AH    ;CX->year,DH->month,DL->day,AL->week
```

```
INT 21H
MOV Y,CX
MOV M,DH
MOV D,DL
MOV BX,Y
CALL DIS
CALL SPLIT
MOV BL,M
CALL DIS
CALL SPLIT
MOV BL,D
CALL DIS
```

```
MOV AH,9
LEA DX,MSG2
INT 21H
```

```
MOV AH,2CH      ;CH->hour,CL->minute,DH->second,DL->millisecond
INT 21H
MOV H,CH
MOV MIN,CL
MOV S,AH
MOV P,AL
;显示时间
MOV BH,0
MOV BL,H
CALL DIS
CALL DOT
MOV BL,MIN
CALL DIS
CALL DOT
MOV BL,S
CALL DIS
CALL DOT
MOV BL,P
CALL DIS
```

```
MOV AH,2CH
INT 21H
```

```
MOV AH,4CH
INT 21H
```

DIS PROC

PUSH AX

PUSH BX

PUSH CX

PUSH DX

MOV AX,BX

MOV CX,0

MOV BX,10

LET1: ;十进制表示

MOV DX,0

INC CX

IDIV BX ;÷10

PUSH DX ;DX是余数, 入栈

CMP AX,0

JNZ LET1

LET2: ;弹出栈显示余数

POP AX

ADD AX,30H

MOV DL,AL

MOV AH,2

INT 21H

LOOP LET2

POP DX

POP CX

POP BX

POP AX

MOV BX,0

RET

DIS ENDP

SPLIT PROC

PUSH AX

PUSH DX

MOV DL,'/'

MOV AH,2

INT 21H

POP DX

POP AX

RET

SPLIT ENDP

```

DOT PROC
    PUSH AX
    PUSH DX

    MOV DL,','
    MOV AH,2
    INT 21H

    POP DX
    POP AX
    RET
DOT ENDP
CODES ENDS
END START

```

9.37

```

DATAS SEGMENT
    CONTENT DB 15,?,16 DUP(?)
    MSG DB 'INPUT TEXT: $'
    UPPER DB 0
    LOWER DB 0
DATAS ENDS

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

    ;清屏
    MOV AH,6
    MOV AL,0
    MOV BH,07H ;黑底白字
    MOV CH,0 ;左上(0,0)
    MOV CL,0
    MOV DH,24
    MOV DL,79 ;右下(24,79)->25 * 80
    INT 10H

    ;打开窗口
    MOV AH,6

```

```
MOV AL,0
MOV BH,70H
MOV CH,5
MOV CL,10
MOV DH,14
MOV DL,50
INT 10H
```

;移动光标到5行10列

```
MOV AH,2
MOV DH,5
MOV DL,10
MOV BH,0
INT 10H
```

```
LEA DX,MSG
MOV AH,9
INT 21H
```

```
MOV AL,0
MOV AH,10
LEA DX,CONTENT
INT 21H
```

;判断大小写

```
MOV CL,CONTENT[1]
MOV SI,0
```

LET0:

```
CMP CL,0
JE LETE
MOV AL,CONTENT[2+SI]
TEST AL,20H
JE LET1
INC LOWER
INC SI
DEC CL
JMP LET0
```

LET1:

```
INC UPPER
INC SI
DEC CL
JMP LET0
```

LETE:

;移动光标到6行10列

```
MOV AH,2
MOV DH,6
MOV DL,10
MOV BH,0
INT 10H
```

```
MOV DL,UPPER
ADD DL,30H
MOV AH,2
INT 21H
```

;移动光标到7行10列

```
MOV AH,2
MOV DH,7
MOV DL,10
MOV BH,0
INT 10H
```

```
MOV DL,LOWER
ADD DL,30H
INT 21H
```

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

9.40

```
CODES SEGMENT
    ASSUME CS:CODES
START:
```

;将80H号中断放入向量表中

```
LEA DX,OUTPUTSTART
MOV AX,SEG OUTPUTSTART
MOV DS,AX
MOV AL,80H
MOV AH,25H
INT 21H
```


INT 80H

;中断驻留

MOV AH,31H

MOV AL,0

MOV DX,OUTPUTEND-OUTPUTSTART+16

MOV AH,4CH

INT 21H

;中断程序

OUTPUTSTART:

JMP CODE

NUMBERS DB ' ',9,8,7,6,5,4,3,2,1

COLOR DB 3DH

CODE:

;现场保护

PUSH DS

PUSH DX

PUSH AX

PUSH CS

POP DS

;写显存

MOV BX,0B800H

MOV ES,BX

MOV BX,1

MOV CX,9 ;9个数字改变颜色

S:

MOV AH,2

MOV SI,CX

MOV DL,NUMBERS[SI]

ADD DL,30H

INT 21H

MOV AL,COLOR

ADD BYTE PTR ES:[BX],AL

ADD AL,3DH ;下一个字符变色

MOV COLOR,AL

ADD BX,2

LOOP S

;现场恢复

POP AX

```
POP DX
POP DS
IRET
OUTPUTEND:NOP
CODES ENDS
END START
```