操作系统实验报告

计科 (大数据、人工智能方向) 17341155 王永康

(实验四: 具有中断处理的内核)

一、实验目的

学习中断机制知识,掌握中断处理程序设计的要求,掌握异步事件编程技术

二、实验内容

- 操作系统工作期间,利用时钟中断,在屏幕24行79列位置轮流显示'|'、'/'和'\',适当控制显示速度,以方便观察效果。
- 编写键盘中断响应程序,原有的你设计的用户程序运行时,键盘事件会做出有事反应:当键盘有按键时,屏幕 适当位置显示"OUCH! OUCH!"。
- 在内核中,对33号、34号、35号和36号中断编写中断服务程序,分别在屏幕1/4区域内显示一些个性化信息。 再编写一个汇编语言的程序,作为用户程序,利用int 33、int 34、int 35和int 36产生中断调用这4个服务程序。

三、实验方案

操作系统实验工具与环境

• 实验支撑环境

。 硬件: 个人计算机

主机操作系统: Windows10虚拟机软件: VMware, Dosbox

• 实验开发工具:

。 汇编语言工具: x86汇编语言

○ 汇编编译工具: TASM+TCC+NASM

。 磁盘写入工具: Winhex

四、实验过程

编写时钟中断程序: 风火轮程序

利用时钟中断号8号,修改中断向量表,8号中断向量对应8*4和8*4+2两个地址,把9号中断需要执行的部分的偏移入口地址放置在8*4中,段地址放置在8*4+2中,具体代码如下:

```
1 org 1000h ; 程序加载到100h, 可用于生成COM
2 ; 设置时钟中断向量 (08h) , 初始化段寄存器
3 _start:
4 mov ax,cs
```

```
; DS = CS
6
       mov ds,ax
7
       mov es,ax
8
9
                         ; AX = 0
       xor ax,ax
                         ; ES = 0
10
       mov es,ax
       mov ax,0B800h ; 文本窗口显存起始地址
11
12
       mov gs, ax ; GS = B800h
13
14
15
       cli
       mov word [es:20h],Timer1 ; 设置时钟中断向量的偏移地址
16
17
       mov word [es:22h],cs
18
       sti
19
20
       mov ax,cs
21
       mov ds,ax
                      ; DS = CS
22
       mov es,ax
23
24
25
       ret
26
27
28
       jmp $ ; 死循环
29
30
    ;时钟中断处理程序
31
   Timer1:
32
       jmp short Timer
33
   Timer:
34
35
       push ax
36
       push bx
37
       push cx
38
       push dx
39
40
       push bp
41
       push es
42
       dec byte [count] ; 递减计数变量 jnz _end ; >0: 跳转
43
44
45
       mov al, byte[cnt]
46
47
48
       cmp al,0
49
       jne here1
50
       mov byte[gs:((80*15+70)*2)],'\'
51
       jmp here0
52
53
       here1:
54
       cmp al,1
55
       jne here2
       mov byte[gs:((80*15+70)*2)],'|'
56
57
       jmp here0
58
```

```
59
        here2:
60
        cmp al,2
61
        jne here3
        mov byte[gs:((80*15+70)*2)],'/'
62
63
        jmp here0
64
65
        here3:
66
        cmp al,3
        jne here0
67
68
        mov byte[gs:((80*15+70)*2)],'-'
69
70
71
        here0:
        inc byte[cnt]
72
73
        cmp byte[cnt],4
74
        jne here5
75
        mov byte[cnt],0
76
        here5:
        mov byte[count],2 ; 重置计数变量=初值delay
77
78
        ;sti
79
    _end:
80
        mov al,20h ; AL = EOI out 20h,al ; 发送EOI到:
81
82
                          ; 发送EOI到主8529A
        out OAOh,al ; 发送EOI到从8529A
83
84
85
86
        pop es
87
        pop bp
88
        pop dx
89
        pop cx
90
91
        pop bx
92
        pop ax
93
                  ; 从中断返回
94
        iret
95
    _data:
        delay equ 4 ; 计时器延迟计数
96
97
        count db delay ; 计时器计数变量, 初值=delay
98
        cnt db 8
99 times 510-($-$$) db 0
100 dw 0xaa55
```

之后,把该程序放在内存偏移量为1000h的位置处,在进入后内核后**默认执行**该段程序,把它加载到1000h处,所以可以做到在屏幕右下角显示一个一直转的风火轮的程序。

加载程序如下:

```
1
_Load_Timer proc

2
push ds

3
push es

4
mov ax,cs ;段地址 ; 存放数据的内存基地址

5
mov es,ax ;设置段地址 (不能直接mov es,段地址)
```

```
mov bx, Time_OffSet ;偏移地址; 存放数据的内存偏移地址
6
7
          mov ah, 2
                              ;功能号
 8
          mov al,1
                              ;扇区数
9
          mov d1,0
                              ;驱动器号; 软盘为0, 硬盘和U盘为80H
                               ;磁头号 ; 起始编号为0
10
          mov dh,0
          mov ch,0
                               ;柱面号 ; 起始编号为0
11
12
          mov cl, byte ptr _Timer_sector
                                       ;起始扇区号 ; 起始编号为1
                               调用读磁盘BIOS的13h功能
13
          int 13H ;
14
          ; 用户程序a.com已加载到指定内存区域中
15
         mov bx, Time_OffSet
16
         call bx
17
18
       pop es
19
       pop ds
20
       ret
21 _Load_Timer endp
```

修改9号中断编程

同理,对9号中断的向量表的位置9*4和9*4+2处存入新的9号中断**执行程序的开始偏移地址**和**段地址**;新9号中断执行的功能是,只能按下键盘,就会在屏幕上显示"OUCH!OUCH!"的字样,并隔一段时间会消失。具体代码如下:(具体说明见注释)

```
1 NewINT9 proc
 2
            ;保护寄存器
3
          push ax
4
          push bx
 5
           push cx
 6
           push bp
 7
           push es
 8
       in al,60h
9
10
       mov ax,cs
11
        mov ds,ax
12
13
            ;;;;;;;;;以下是显示OUCH
14
        ;mov es,St
15
16
           mov si, offset Stt
17
18
        mov ax,0B800h
19
          mov es,ax
20
21
22
        mov es,ax
23
        mov di, (12*80 + 5)*2
24
25
        mov cx,10
26
27
           s:mov al,[si]
28
        mov es:[di],al
```

```
inc si
29
30
       add di,2
31
       loop s
32
       ;;;;;;;;;;
33
34
           ;;;延迟一段时间
35
           call LOOPSs
36
37
           ;;;;;清除OUCH
38
           mov si, offset Smm
39
       mov ax,0B800h
40
      mov es,ax
41
       mov di, (12*80 + 5)*2
42
        mov cx,10
43
    s3:mov al,[si]
44
      mov es:[di],al
45
       inc si
46
       add di,2
       loop s3
47
48
49
           ;;;;;;清除结束
50
51
52
           ;;;;貌似下面是说发生到芯片端口,告诉中断结束???
53
54
      mov al,20h
                         ; AL = EOI
                         ; 发送EOI到主8529A
55
       out 20h,al
                         ; 发送EOI到从8529A
      out OAOh,al
56
57
58
59
           pop es
60
           pop bp
61
           pop cx
62
           pop bx
63
           pop ax
64
       iret
65
66
       ret
67
   NewINT9 endp
68
```

上述代码来自**内核**,修改9号中断的位置是在**内核的用户加载程序**处,并且在调用完用户程序后**恢复**9号中断的原始功能——赋回原值。与此相关的内核代码:

```
1 _Load proc
2
       push ds
3
       push es
4
          ;;;;;;
5
           ;;.....
6
           ;;;;;;相关部分代码:
7
           xor ax,ax
8
           mov es,ax
9
           mov ax,es:[24h]
```

```
10
11
            ;;保存9号中断的原始功能
12
            mov word ptr cs:[stack1],ax
            mov ax, es:[26h]
13
14
            mov word ptr cs:[stack1+2],ax
15
16
            ;;修改9号中断
17
            cli
18
            mov word ptr es:[24h], offset NewINT9
19
            mov ax,cs
20
            mov word ptr es:[26h],ax
21
            sti
22
23
            mov ax,cs
24
            mov ds,ax
25
            mov es,ax
26
27
            ;;;执行用户程序
28
29
            call bx
30
31
32
            ;;;;执行结束,恢复中断
33
            xor ax,ax
34
            mov es,ax
35
            cli
36
            mov ax,cs:[stack1]
37
38
            mov word ptr es:[24h],ax
39
40
            mov ax, cs:[stack1+2]
            mov word ptr es:[26h],ax
41
42
            sti
43
44
        mov ax,cs
45
        mov es,ax
46
        mov ds,ax
47
48
        here4:
49
        pop es
50
        pop ds
51
        ret
52
    _load endp
```

这样做,可以使调用用户程序的时候,按键盘会显示"OUCH! OUCH!"的信息,执行结束自动恢复9号中断。

修改33、34、35、36号程序编程

原理同上,新的33、34、35、36中断指向的程序作用是:去加载第4、5、6、7扇区的程序到内存中,并且运行; 相关代码:

```
1
   NewINT33 proc
2
       push ds
3
       push es
4
 5
           mov byte ptr cs:[Sectors],4;;;指定加载第4扇区的用户程序
 6
           call LOAD_for_INT ;;;调用加载函数
7
8
       pop es
9
       pop ds
10
       iret
11
12
       ret
13 NewINT33 endp
```

加载函数代码:

```
1
   LOAD_for_INT proc
2
       push ds
3
       push es
          mov ax,cs
                                ;段地址 ; 存放数据的内存基地址
4
5
          mov es,ax
                                ;设置段地址 (不能直接mov es,段地址)
 6
          mov bx, OffSetOfUserPrg4 ;偏移地址; 存放数据的内存偏移地址
 7
          mov ah,2
                                ; 功能号
8
          mov al,1
                                ;扇区数
9
          mov d1,0
                                ;驱动器号; 软盘为0, 硬盘和U盘为80H
          mov dh,0
                                ;磁头号; 起始编号为0
10
11
          mov ch,0
                                ;柱面号; 起始编号为0
                                              ;起始扇区号; 起始编号为1
12
          mov cl,byte ptr es:[Sectors]
13
          int 13H ;
                                调用读磁盘BIOS的13h功能
14
          ; 用户程序a.com已加载到指定内存区域中
          mov bx, OffSetOfUserPrg4
15
16
          call bx
17
       pop es
18
       pop ds
19
20
       ret
21 LOAD_for_INT endp
```

再次,编写用户程序:

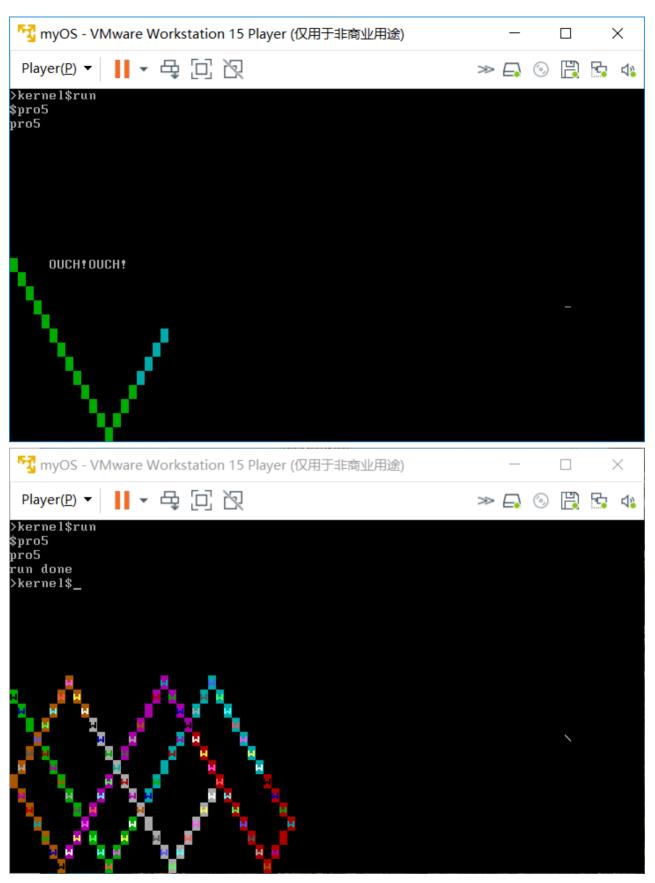
```
1 ;; wang yongkang, haha
2 org 09000h
3
   start:
4
       mov ax,cs
 5
      mov ds,ax
 6
       mov es,ax
 7
       int 33
 8
9
10
        int 34
11
12
        int 35
```

```
13
14 int 36
15
16 ret
17 jmp $
18
19 times 510-($-$$) db 0
20 dw 0xaa55
```

五、实验结果

• 时钟中断演示:

• 9号中断演示



执行33、34、35、36中断其中通过运行程序6, pro5, 将会接连调用pro1,pro2,pro3,pro4,演示效果如下:



• 最终软盘a.img的各项说明:

a.img的扇区	储存内容
1	遇到程序myOS.bin
2	储存软盘信息的文件
3	时钟中断程序timer.bin
4	左上角1/4程序 (pro1)
5	左下角1/4程序 (pro2)
6	右上角1/4程序(pro3)
7	右下角1/4程序(pro4)
8	弹跳字符程序(pro5)
9	int33,34,35,36测试程序 (pro6)
12-16	KER.bin,内核程序

六、实验总结

这一次中断编程的实验, 主要有以下体会:

• 遇到很多问题与**寄存器的保存**有关;

在实验实现过程中,以9号中断的改写为例,开始时没有注意在9号中断里面保护相关寄存器,导致原用户程序出错;

• 段地址保存,偏移地址的保存

更加深入地认识到x86实模式下的寻址模式,段值加偏移,所以中断实现的关键也是需要保存段值和偏移。

- 还遇到一个未解决的问题:
 - 时钟中断的风火轮程序在进入**内核后14秒**才出现,感觉比较奇怪,按照正常的逻辑,应该一来就应该正常运行,这里出现的时间延迟比较奇怪,个人猜测是虚拟机本身虚拟的配置问题引起的。

参考文献

- [1] 凌应标. "04实验课.ppt",中山大学计算机科学系,2015-3
- [2] 于渊.《一个操作系统的实现》[M].电子工业出版社, 2009-6-1
- [3] 王爽.《汇编语言(第3版)》[M].清华大学出版社, 2013-9