

操作系统 实验报告

院系：计算机学院

班级：计科 2 班

学号：19335174

姓名：施天予

指导老师：凌应标

2021. 3. 26

一、实验题目

加载执行 COM 格式用户程序的监控程序

二、实验目的

- 1、了解监控程序执行用户程序的主要工作
- 2、了解一种用户程序的格式与运行要求
- 3、加深对监控程序概念的理解
- 4、掌握加载用户程序方法
- 5、掌握几个 BIOS 调用和简单的磁盘空间管理

三、实验要求

- 1、知道引导扇区程序实现用户程序加载的意义
- 2、掌握 COM/BIN 等一种可执行的用户程序格式与运行要求
- 3、将自己实验一的引导扇区程序修改为 3-4 个不同版本的 COM 格式程序，每个程序缩小显示区域，在屏幕特定区域显示，用以测试监控程序，在 1.44MB 软驱映像中存储这些程序。
- 4、重写 1.44MB 软驱引导程序，利用 BIOS 调用，实现一个能执行 COM 格式用户程序的监控程序。
- 5、设计一种简单命令，实现用命令交互执行在 1.44MB 软驱映像中存储几个用户程序。
- 6、编写实验报告，描述实验工作的过程和必要的细节，如截屏或录屏，以证实实验工作的真实性。

四、实验方案

【实验环境】

- 1、实验运行环境：Windows10
- 2、虚拟机软件：VirtualBox

【实验工具】

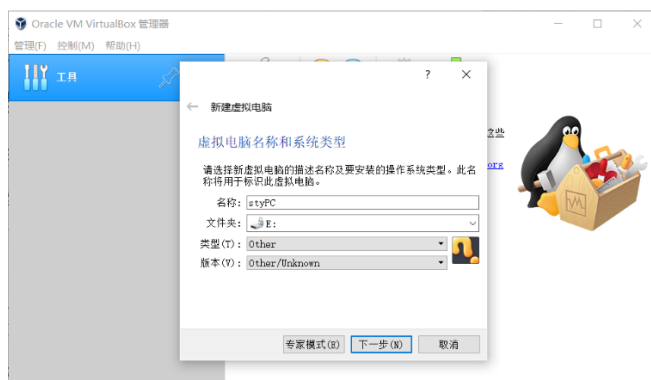
- 1、汇编语言：NASM
- 2、文本编辑器：Notepad++
- 3、相关软件：WinHex

五、实验过程

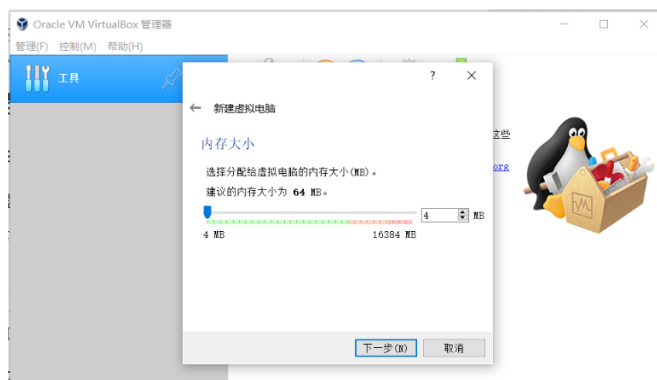
1、搭建和应用实验环境

【安装虚拟机】

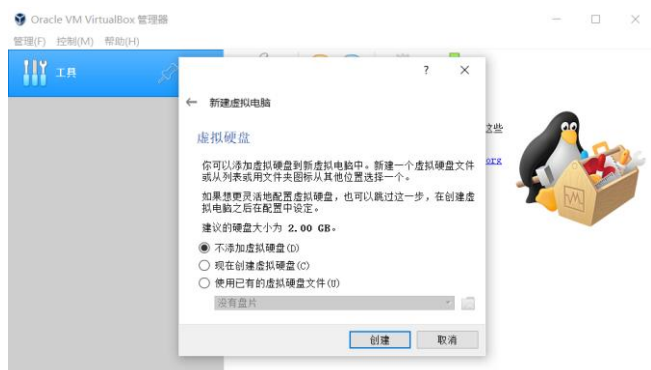
打开 VirtualBox，新建无操作系统的虚拟机



内存设置为 4MB

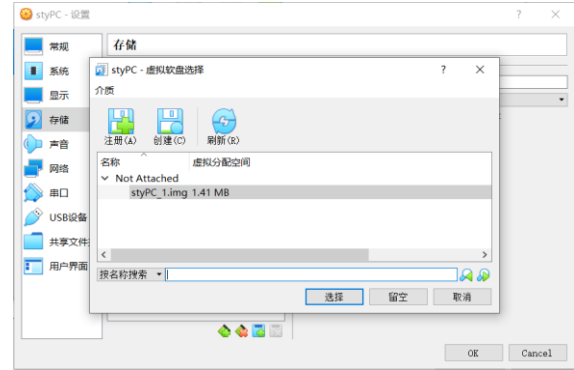
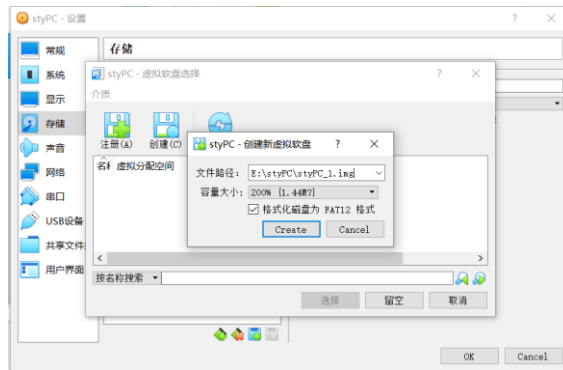


选择不添加虚拟硬盘



【虚拟软盘】

创建虚拟软盘后，选择注册虚拟软盘



2、设计监控程序

【设计代码】

```
1 org 7c00h ; 监控程序地址
2 OffsetOfUserPrgr equ 8100h ; 用户程序地址
3
4 Start:
5     mov ax, cs ; 置其他段寄存器值与CS相同
6     mov ds, ax ; 数据段
7     mov bp, Message ; BP=当前串的偏移地址
8     mov ax, ds ; ES:BP = 串地址
9     mov es, ax ; 置ES=DS
10 cls: ; 清屏
11     mov ah, 6
12     mov al, 0
13     mov ch, 0
14     mov cl, 0
15     mov dh, 24
16     mov dl, 79
17     mov bh, 7
18     int 10h
19 show:
20     mov cx, MessageLength ; CX = 串长 (=9)
21     mov ax, 1301h ; AH = 13h (功能号)、AL = 01h (光标置于串尾)
22     mov bx, 0007h ; 页号为0 (BH = 0) 黑底白字 (BL = 07h)
23     mov dh, 0 ; 行号=0
24     mov dl, 0 ; 列号=0
25     int 10h ; BIOS的10h功能: 显示一行字符
26
27 input:
28     mov ah, 0
29     int 16h
30     cmp al, '1'
31     jz switch
32     cmp al, '2'
33     jz switch
34     cmp al, '3'
35     jz switch
36     cmp al, '4'
37     jz switch
38     jmp input
```

```

39
40 switch:
41     mov bl,'1'
42     cmp al,bl
43     jz program1
44     mov bl,'2'
45     cmp al,bl
46     jz program2
47     mov bl,'3'
48     cmp al,bl
49     jz program3
50     mov bl,'4'
51     cmp al,bl
52     jz program4
53
54 program1:
55     mov ax,cs                ;段地址 ; 存放数据的内存基地址
56     mov es,ax                ;设置段地址 (不能直接mov es,段地址)
57     mov bx, OffsetOfUserPrg    ;偏移地址; 存放数据的内存偏移地址
58     mov ah,2                  ;功能号, 读磁盘
59     mov al,1                  ;读入扇区数
60     mov dl,0                  ;驱动器号 ; 软盘为0, 硬盘和U盘为80H
61     mov dh,0                  ;磁头号 ; 起始编号为0
62     mov ch,0                  ;柱面号 ; 起始编号为0
63     mov cl,2                  ;扇区号2
64     int 13H ;                调用读磁盘BIOS的13h功能
65     jmp OffsetOfUserPrg
66 program2:
67     mov ax,cs                ;段地址 ; 存放数据的内存基地址
68     mov es,ax                ;设置段地址 (不能直接mov es,段地址)
69     mov bx, OffsetOfUserPrg    ;偏移地址; 存放数据的内存偏移地址
70     mov ah,2                  ;功能号, 读磁盘
71     mov al,1                  ;读入扇区数
72     mov dl,0                  ;驱动器号 ; 软盘为0, 硬盘和U盘为80H
73     mov dh,0                  ;磁头号 ; 起始编号为0
74     mov ch,0                  ;柱面号 ; 起始编号为0
75     mov cl,3                  ;扇区号3
76     int 13H ;                调用读磁盘BIOS的13h功能
77     jmp OffsetOfUserPrg
78 program3:
79     mov ax,cs                ;段地址 ; 存放数据的内存基地址
80     mov es,ax                ;设置段地址 (不能直接mov es,段地址)
81     mov bx, OffsetOfUserPrg    ;偏移地址; 存放数据的内存偏移地址
82     mov ah,2                  ;功能号, 读磁盘
83     mov al,1                  ;读入扇区数
84     mov dl,0                  ;驱动器号 ; 软盘为0, 硬盘和U盘为80H
85     mov dh,0                  ;磁头号 ; 起始编号为0
86     mov ch,0                  ;柱面号 ; 起始编号为0
87     mov cl,4                  ;扇区号4
88     int 13H ;                调用读磁盘BIOS的13h功能
89     jmp OffsetOfUserPrg
90 program4:
91     mov ax,cs                ;段地址 ; 存放数据的内存基地址
92     mov es,ax                ;设置段地址 (不能直接mov es,段地址)
93     mov bx, OffsetOfUserPrg    ;偏移地址; 存放数据的内存偏移地址
94     mov ah,2                  ;功能号, 读磁盘
95     mov al,1                  ;读入扇区数
96     mov dl,0                  ;驱动器号 ; 软盘为0, 硬盘和U盘为80H
97     mov dh,0                  ;磁头号 ; 起始编号为0
98     mov ch,0                  ;柱面号 ; 起始编号为0
99     mov cl,5                  ;扇区号5
100    int 13H ;                调用读磁盘BIOS的13h功能
101    jmp OffsetOfUserPrg
102
103 Message:
104     db "Welcome!",0AH,0DH
105     db "Please Enter the number to choose the different program: ",0AH,0D
106     db "1.Number",0AH,0DH
107     db "2.Name",0AH,0DH
108     db "3.Rectangle",0AH,0DH
109     db "4.Stone",0AH,0DH
110
111     MessageLength equ ($-Message)
112     times 510-($-$$) db 0
113     db 0x55,0xaa

```

【代码分析】

按照老师 PPT 中的代码改善，首先将用户程序地址设置为 8100h。因为在监控程序和用户程序切换时会残留之前显示的内容，所以我增加了一个清屏功能。用 int 16h 的 BIOS 功能调用，读取键盘输入，设置 1,2,3,4 跳到不同的用户程序。用户程序 1 在第 2 个扇区，用户程序 2 在第 3 个扇区，用户程序 3 在第 4 个扇区，用户程序 4 在第 5 个扇区。用 times 510-(\$-\$\$) db 0 和 db 0x55,0xaa 最后完成一个引导扇区程序。

3、设计用户程序

【用户程序 1：彩色移动显示学号】

```
1      Dn_Rt equ 1                ;D-Down,U-Up,R-right,L-Left
2      Up_Rt equ 2                ;
3      Up_Lt equ 3                ;
4      Dn_Lt equ 4                ;
5      delay equ 50000            ; 计时器延迟计数,用于控制画框的速度
6      ddelay equ 580            ; 计时器延迟计数,用于控制画框的速度
7      org 8100h
8
9      cls: ;清屏
10     mov ah,6
11     mov al,0
12     mov ch,0
13     mov cl,0
14     mov dh,24
15     mov dl,79
16     mov bh,7
17     int 10h
18
19     start:
20     mov ax,cs                  ;获得程序运行时，代码段在内存的位置
21     mov ds,ax                  ; DS = CS
22     mov ss,ax                  ; SS = CS
23     mov ax,0B800h              ; 文本窗口显存起始地址
24     mov es,ax                  ; ES = B800h
25
26     loop1:
27     dec word[count]            ; 递减计数变量
28     jnz loop1                  ; >0: 跳转;
29     mov word[count],delay
30     dec word[dcount]           ; 递减计数变量
31     jnz loop1
32     mov word[count],delay
33     mov word[dcount],ddelay
34
35     mov al,1
36     cmp al,byte[rdul]
37     jz DnRt
38     mov al,2
```

```

39      cmp al,byte[rdu1]
40      jz  UpRt
41      mov al,3
42      cmp al,byte[rdu1]
43      jz  UpLt
44      mov al,4
45      cmp al,byte[rdu1]
46      jz  DnLt
47      jmp $
48
49 DnRt:
50      inc word[x]
51      inc word[y]
52      mov bx,word[x]
53      mov ax,14
54      sub ax,bx
55      jz  dr2ur
56      mov bx,word[y]
57      mov ax,40
58      sub ax,bx
59      jz  dr2dl
60      jmp show
61 dr2ur:
62      mov word[x],12
63      mov byte[rdu1],Up_Rt
64      jmp show
65 dr2dl:
66      mov word[y],38
67      mov byte[rdu1],Dn_Lt
68      jmp show
69 UpRt:
70      dec word[x]
71      inc word[y]
72      mov bx,word[y]
73      mov ax,40
74      sub ax,bx
75      jz  ur2ul
76      mov bx,word[x]

```

```

115      jz  dl2dr
116      mov bx,word[x]
117      mov ax,14
118      sub ax,bx
119      jz  dl2ul
120      jmp show
121 dl2dr:
122      mov word[y],1
123      mov byte[rdu1],Dn_Rt
124      jmp show
125 dl2ul:
126      mov word[x],12
127      mov byte[rdu1],Up_Lt
128      jmp show
129
130 show:
131      xor ax,ax
132      mov word ax,[x]
133      mov bx,80
134      mul bx
135      add word ax,[y]
136      mov bx,2
137      mul bx
138      mov bx,ax
139      mov si,NUMBER
140      mov cx,8
141      color:
142      mov byte al,[si]

```

```

77      mov ax,-1
78      sub ax,bx
79      jz  ur2dr
80      jmp show
81 ur2ul:
82      mov word[y],38
83      mov byte[rdu1],Up_Lt
84      jmp show
85 ur2dr:
86      mov word[x],1
87      mov byte[rdu1],Dn_Rt
88      jmp show
89 UpLt:
90      dec word[x]
91      dec word[y]
92      mov bx,word[x]
93      mov ax,-1
94      sub ax,bx
95      jz  ul2dl
96      mov bx,word[y]
97      mov ax,-1
98      sub ax,bx
99      jz  ul2ur
100     jmp show
101 ul2dl:
102     mov word[x],1
103     mov byte[rdu1],Dn_Lt
104     jmp show
105 ul2ur:
106     mov word[y],1
107     mov byte[rdu1],Up_Rt
108     jmp show
109 DnLt:
110     inc word[x]
111     dec word[y]
112     mov bx,word[y]
113     mov ax,-1
114     sub ax,bx

```

```

143      mov [es:bx],ax
144      inc si
145      inc bx
146      inc bx
147      loop color
148
149      ; 输入空格后弹出程序
150      mov ah,1
151      int 16h
152      mov bl,20h
153      cmp al,bl
154      jz  Quit
155      jmp loop1
156
157 Quit:
158      jmp 7c00h
159 end:
160      jmp $ ; 停止画框，无限循环
161
162 datadef:
163 count dw delay
164 dcount dw ddelay
165 rdu1 db Dn_Rt ; 向右下运动
166 x dw -1
167 y dw 0
168 NUMBER db "19335174"
169
170 times 512-($-$$) db 0

```

【代码分析】

每个用户程序都要在开头 org 8100h，并且进行一次清屏操作。第一个用户程序与实验 1 的代码类似，只是变色移动的是学号，并且改了一下边界参数使其出现在屏幕左上方区域。在程序末尾使用 int 16h 的 1 号功能，实现捕捉键盘输入，在输入空格时，用 jmp 7c00h 返回监控程序。times 512-(\$-\$\$) db 0 在最后使生成的 COM 文件 512 字节。

【用户程序 2：彩色移动显示姓名】

```

1      Dn_Rt equ 1           ;D-Down,U-Up,R-right,L-Left
2      Up_Rt equ 2
3      Up_Lt equ 3
4      Dn_Lt equ 4
5      delay equ 50000      ; 计时器延迟计数,用于控制画框的速度
6      ddelay equ 580      ; 计时器延迟计数,用于控制画框的速度
7      org 8100h
8
9      cls: ;清屏
10     mov ah,6
11     mov al,0
12     mov ch,0
13     mov cl,0
14     mov dh,24
15     mov dl,79
16     mov bh,7
17     int 10h
18
19     start:
20     mov ax,cs             ;获得程序运行时,代码段在内存的位置
21     mov ds,ax             ; DS = CS
22     mov ss,ax             ; SS = CS
23     mov ax,0B800h        ; 文本窗口显存起始地址
24     mov es,ax             ; ES = B800h
25
26     loop1:
27     dec word[count]       ; 递减计数变量
28     jnz loop1             ; >0: 跳转;
29     mov word[count],delay
30     dec word[dcount]      ; 递减计数变量
31     jnz loop1
32     mov word[count],delay
33     mov word[dcount],ddelay
34
35     mov al,1
36     cmp al,byte[rdu1]
37     jz DnRt
38     mov al,2
39
40     cmp al,byte[rdu1]
41     jz UpRt
42     mov al,3
43     cmp al,byte[rdu1]
44     jz UpLt
45     mov al,4
46     cmp al,byte[rdu1]
47     jz DnLt
48     jmp $
49
50     DnRt:
51     inc word[x]
52     inc word[y]
53     mov bx,word[x]
54     sub ax,bx
55     jz dr2ur
56     mov bx,word[y]
57     mov ax,40
58     sub ax,bx
59     jz dr2dl
60     jmp show
61
62     dr2ur:
63     mov word[x],23
64     mov byte[rdu1],Up_Rt
65     jmp show
66
67     dr2dl:
68     mov word[y],38
69     mov byte[rdu1],Dn_Lt
70     jmp show
71
72     UpRt:
73     dec word[x]
74     inc word[y]
75     mov bx,word[y]
76     mov ax,40
77     sub ax,bx
78     jz ur2ul
79     mov bx,word[x]
80     sub ax,bx
81     jz ur2dr
82     jmp show
83
84     ur2ul:
85     mov word[y],38
86     mov byte[rdu1],Up_Lt
87     jmp show
88
89     ur2dr:
90     mov word[x],14
91     mov byte[rdu1],Dn_Rt
92     jmp show
93
94     UpLt:
95     dec word[x]
96     dec word[y]
97     mov bx,word[x]
98     mov ax,12
99     sub ax,bx
100    jz ul2dl
101    mov bx,word[y]
102    mov ax,-1
103    sub ax,bx
104    jz ul2ur
105    jmp show
106
107    ul2dl:
108    mov word[x],14
109    mov byte[rdu1],Dn_Lt
110    jmp show
111
112    ul2ur:
113    mov word[y],1
114    mov byte[rdu1],Up_Rt
115    jmp show
116
117    DnLt:
118    inc word[x]
119    dec word[y]
120    mov bx,word[y]
121    mov ax,-1
122    sub ax,bx
123    jz dl2ul
124    mov bx,word[x]
125    sub ax,bx
126    jz dl2dr
127    jmp show
128
129    dl2ul:
130    mov word[x],23
131    mov byte[rdu1],Up_Lt
132    jmp show
133
134    dl2dr:
135    mov word[y],1
136    mov byte[rdu1],Dn_Rt
137    jmp show
138
139    show:
140    xor ax,ax
141    mov word ax,[x]
142    mov bx,80
143    mul bx
144    add word ax,[y]
145    mov bx,2
146    mul bx
147    mov bx,ax
148    mov si,NAME
149    mov cx,3 ; 字符串长度
150    color:
151    mov byte al,[si]
152
153    mov [es:bx],ax
154    inc si
155    inc bx
156    inc bx
157    loop color
158
159    ; 输入空格后弹出程序
160    mov ah,1
161    int 16h
162    mov bl,20h
163    cmp al,bl
164    jz Quit
165    jmp loop1
166
167    Quit:
168    jmp 7c00h
169
170    end:
171    jmp $ ; 停止画框,无限循环
172
173    datadef:
174    count dw delay
175    dcount dw ddelay
176    rdu1 db Dn_Rt ; 向右下运动
177    x dw 12
178    y dw 0
179    NAME db "STY"
180    times 512-($-$$) db 0

```


【代码分析】

用户程序 2 与用户程序 1 基本类似，只是彩色移动的是姓名大写缩写。再改了一下边界的参数，实现在屏幕左下角显示。

【用户程序 3：“A” 字矩形和学号姓名显示】

```
1  Dn_Rt equ 1 ;D-Down,U-Up,R-right,L-Left
2  Up_Rt equ 2 ;
3  Up_Lt equ 3 ;
4  Dn_Lt equ 4 ;
5  delay equ 50000 ; 计时器延迟计数,用于控制画框的速度
6  ddelay equ 580 ; 计时器延迟计数,用于控制画框的速度
7  org 8100h
8
9  cls: ;清屏
10 mov ah,6
11 mov al,0
12 mov ch,0
13 mov cl,0
14 mov dh,24
15 mov dl,79
16 mov bh,7
17 int 10h
18
19 start:
20 mov ax,cs ;获得程序运行时，代码段在内存的位置
21 mov ds,ax ; DS = CS
22 mov ss,ax ; SS = CS
23 mov ax,0B800h ; 文本窗口显存起始地址
24 mov es,ax ; ES = B800h
25
26 loop1:
27 dec word[count] ; 递减计数变量
28 jnz loop1 ; >0: 跳转;
29 mov word[count],delay
30 dec word[dcount] ; 递减计数变量
31 jnz loop1
32 mov word[count],delay
33 mov word[dcount],ddelay
34
35 mov al,1
36 cmp al,byte[rdul]
37 jz DnRt
38 mov al,2
39
40 cmp al,byte[rdul]
41 jz UpRt
42 mov al,3
43 cmp al,byte[rdul]
44 jz UpLt
45 mov al,4
46 cmp al,byte[rdul]
47 jz DnLt
48 jmp $
49
50 DnRt:
51 inc word[x]
52 inc word[y]
53 mov bx,word[x]
54 mov ax,14
55 sub ax,bx
56 jz dr2ur
57 mov bx,word[y]
58 mov ax,40
59 sub ax,bx
60 jz dr2dl
61 jmp show
62
63 dr2ur:
64 mov word[x],12
65 mov byte[rdul],Up_Rt
66 jmp show
67
68 dr2dl:
69 mov word[y],38
70 mov byte[rdul],Dn_Lt
71 jmp show
72
73 UpRt:
74 dec word[x]
75 inc word[y]
76 mov bx,word[y]
77 mov ax,40
78 sub ax,bx
79 jz ur2ul
80 mov bx,word[x]
81
82 ur2ul:
83 mov word[y],38
84 mov byte[rdul],Up_Lt
85 jmp show
86
87 ur2dr:
88 mov word[x],1
89 mov byte[rdul],Dn_Rt
90 jmp show
91
92 UpLt:
93 dec word[x]
94 dec word[y]
95 mov bx,word[x]
96 mov ax,-1
97 sub ax,bx
98 jz ul2dl
99 mov bx,word[y]
100 mov ax,-1
101 sub ax,bx
102 jz ul2ur
103 jmp show
104
105 ul2dl:
106 mov word[x],1
107 mov byte[rdul],Dn_Lt
108 jmp show
109
110 ul2ur:
111 mov word[y],1
112 mov byte[rdul],Up_Rt
113 jmp show
114
115 DnLt:
116 inc word[x]
117 dec word[y]
118 mov bx,word[y]
119 mov ax,-1
120 sub ax,bx
```

```

115     jz dl2dr
116     mov bx,word[x]
117     mov ax,14
118     sub ax,bx
119     jz dl2ul
120     jmp show
121 dl2dr:
122     mov word[y],1
123     mov byte[rdul],Dn_Rt
124     jmp show
125 dl2ul:
126     mov word[x],12
127     mov byte[rdul],Up_Lt
128     jmp show
129
130 show:
131     xor ax,ax
132     mov word ax,[x]
133     mov bx,80
134     mul bx
135     add word ax,[y]
136     mov bx,2
137     mul bx
138     mov bx,ax
139     mov si, NUMBER
140     mov cx, 8
141     color:
142     mov byte al,[si]
143
144     mov [es:bx],ax
145     inc si
146     inc bx
147     inc bx
148     loop color
149
150     ; 输入空格后弹出程序
151     mov ah,1
152     int 16h
153     mov bl,20h
154     cmp al,bl
155     jz Quit
156     jmp loop1
157
158 Quit:
159     jmp 7c00h
160 end:
161     jmp $ ; 停止画框，无限循环
162
163 datadef:
164     count dw delay
165     dcount dw ddelay
166     rdul db Dn_Rt ; 向右下运动
167     x dw -1
168     y dw 0
169     NUMBER db "19335174"
170     times 512-($-$$) db 0

```

【代码分析】

用户程序 3 沿用了之前实验 0 的代码，在屏幕右上角显示一个“A”字形，并在中间显示学号姓名，黑底白字。

【用户程序 4: Stone】

```

1     Dn_Rt equ 1 ;D-Down,U-Up,R-right,L-Left
2     Up_Rt equ 2 ;
3     Up_Lt equ 3 ;
4     Dn_Lt equ 4 ;
5     delay equ 50000 ; 计时器延迟计数,用于控制画框的速度
6     ddelay equ 580 ; 计时器延迟计数,用于控制画框的速度
7     org 8100h
8
9     cls: ;清屏
10     mov ah,6
11     mov al,0
12     mov ch,0
13     mov cl,0
14     mov dh,24
15     mov dl,79
16     mov bh,7
17     int 10h
18
19     start:
20     mov ax,cs ;获得程序运行时，代码段在内存的位置
21     mov ds,ax ; DS = CS
22     mov ss,ax ; SS = CS
23     mov ax,0B800h ; 文本窗口显存起始地址
24     mov es,ax ; ES = B800h
25     mov byte[char],'A'
26
27     loop1:
28     dec word[count] ; 递减计数变量
29     jnz loop1 ; >0: 跳转;
30     mov word[count],delay
31     dec word[dcount] ; 递减计数变量
32     jnz loop1
33     mov word[count],delay
34     mov word[dcount],ddelay
35
36     mov al,1
37     cmp al,byte[rdul]
38     jz DnRt

```

```

39      mov al,2
40      cmp al,byte[rdul]
41      jz UpRt
42      mov al,3
43      cmp al,byte[rdul]
44      jz UpLt
45      mov al,4
46      cmp al,byte[rdul]
47      jz DnLt
48      jmp $
49
50 DnRt:
51      inc word[x]
52      inc word[y]
53      mov bx,word[x]
54      mov ax,25
55      sub ax,bx
56      jz dr2ur
57      mov bx,word[y]
58      mov ax,80
59      sub ax,bx
60      jz dr2dl
61      jmp show
62 dr2ur:
63      mov word[x],23
64      mov byte[rdul],Up_Rt
65      jmp show
66 dr2dl:
67      mov word[y],78
68      mov byte[rdul],Dn_Lt
69      jmp show
70 UpRt:
71      dec word[x]
72      inc word[y]
73      mov bx,word[y]
74      mov ax,80
75      sub ax,bx
76      jz ur2ul

```

```

115     dec word[y]
116     mov bx,word[y]
117     mov ax,39
118     sub ax,bx
119     jz dl2dr
120     mov bx,word[x]
121     mov ax,25
122     sub ax,bx
123     jz dl2ul
124     jmp show
125
126 dl2dr:
127     mov word[y],41
128     mov byte[rdul],Dn_Rt
129     jmp show
130
131 dl2ul:
132     mov word[x],23
133     mov byte[rdul],Up_Lt
134     jmp show
135
136 show:
137     xor ax,ax                ; 计算显存地址
138     mov ax,word[x]
139     mov bx,80
140     mul bx
141     add ax,word[y]
142     mov bx,2

```

```

77     mov bx,word[x]
78     mov ax,12
79     sub ax,bx
80     jz ur2dr
81     jmp show
82 ur2ul:
83     mov word[y],78
84     mov byte[rdul],Up_Lt
85     jmp show
86 ur2dr:
87     mov word[x],14
88     mov byte[rdul],Dn_Rt
89     jmp show
90
91 UpLt:
92     dec word[x]
93     dec word[y]
94     mov bx,word[x]
95     mov ax,12
96     sub ax,bx
97     jz ul2dl
98     mov bx,word[y]
99     mov ax,39
100    sub ax,bx
101    jz ul2ur
102    jmp show
103
104 ul2dl:
105     mov word[x],14
106     mov byte[rdul],Dn_Lt
107     jmp show
108 ul2ur:
109     mov word[y],41
110     mov byte[rdul],Up_Rt
111     jmp show
112
113 DnLt:
114     inc word[x]

```

```

143     mul bx
144     mov bp,ax
145     mov ah,0Fh
146     mov al,byte[char]
147     mov word[es:bp],ax
148     ; 输入空格后弹出程序
149     mov ah,1
150     int 16h
151     mov bl,20h
152     cmp al,bl
153     jz Quit
154     jmp loop1
155
156 Quit:
157     jmp 7c00h
158 end:
159     jmp $                ; 停止画框，无限循环
160
161 datadef:
162     count dw delay
163     dcount dw ddelay
164     rdul db Dn_Rt        ; 向右下运动
165
166     x dw 12;            小球起始位置
167     y dw 52
168     char db 'A'
169
170     times 512-($-$$) db 0

```

【代码分析】

用户程序 4 使用了老师的 stone 程序，改进代码使之显示在屏幕右下角。

4、NASM 编译

```

nasm
Microsoft Windows [版本 10.0.18363.1441]
(c) 2019 Microsoft Corporation。保留所有权利。

E:\nasm>nasm -f bin os.asm -o os.com

E:\nasm>nasm -f bin 1.asm -o 1.com

E:\nasm>nasm -f bin 2.asm -o 2.com

E:\nasm>nasm -f bin 3.asm -o 3.com

E:\nasm>nasm -f bin 4.asm -o 4.com

```

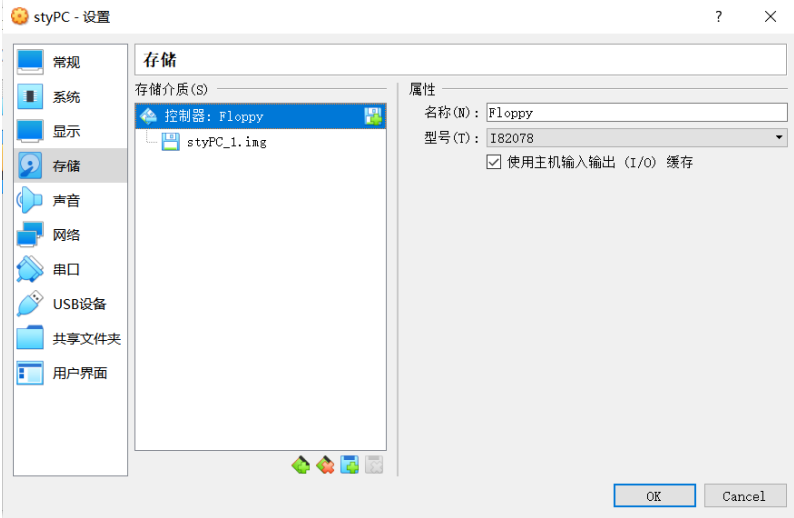
5、使用 WinHex 修改软盘

使用 WinHex 打开所有的 COM 文件和软盘文件。将监控程序 os.com 替换到第一个扇区（0-511 字节），将 1.com 替换到第二个扇区（512-1023 字节），将 2.com 替换到第三个扇区（1024-1535 字节），将 3.com 替换到第四个扇区（1536-2047 字节），将 4.com 替换到第五个扇区（2048-2559 字节）。

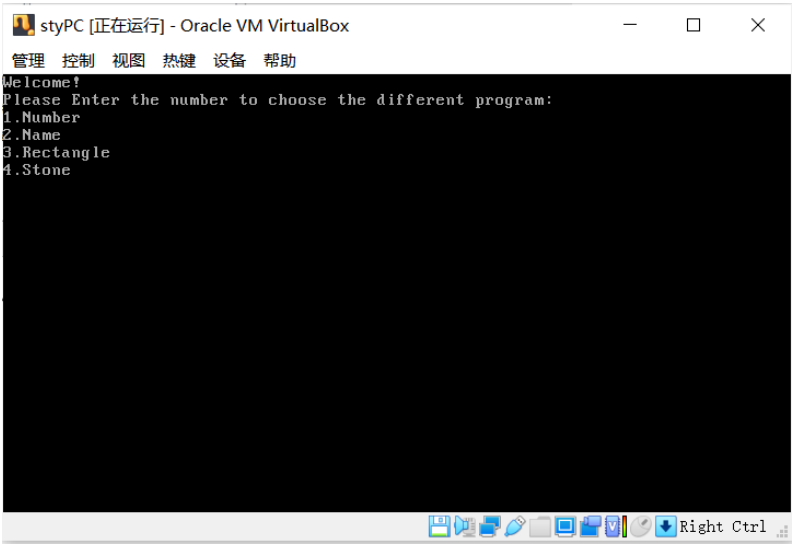
Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	ASCII
00000000	8C	C8	8E	D8	BD	B8	7C	8C	D8	8E	C0	B4	06	B0	00	05	gēzōw, (gēzā' * u
00000016	00	B1	00	B6	18	B2	4F	B7	07	CD	10	B9	6D	00	B8	01	± u ± 'O' i m
00000032	13	BB	07	00	B6	00	B2	00	CD	10	B4	00	CD	16	3C	31	» ± ' i ' i < i
00000048	74	0E	3C	32	74	0A	3C	33	74	06	3C	34	74	02	EB	2A	t < t < c3t < 4t < 6t
00000064	B3	31	39	D8	74	12	D3	32	38	D8	74	24	B3	33	D8	D8	*18t *28t *38t
00000080	74	36	B3	34	38	D8	74	48	8C	C8	8E	C0	BB	00	B1	B4	t * 48t t gēzā' *
00000096	02	B0	01	B2	00	B6	00	B5	00	B1	02	CD	13	E9	90	04	* ± u ± i é
00000112	8C	C9	8E	C0	BB	00	B1	B4	02	B0	01	B2	00	B6	00	B5	gēzā' * ± u
00000128	00	B1	03	CD	13	E9	78	04	8C	C8	8E	C0	BB	00	B1	B4	± i é gēzā' *
00000144	00	B0	01	B2	00	B6	00	B5	00	B1	04	CD	13	E9	60	04	* ± u ± i é
00000160	8C	C8	8E	C0	BB	00	B1	B4	02	B0	01	B2	00	B6	00	B5	gēzā' * ± u
00000176	00	B1	05	CD	13	E9	48	04	57	65	6C	63	6F	6D	65	21	± i é Welcomel
00000192	0A	0D	50	6C	65	61	73	65	20	45	6E	74	65	72	20	74	Please Enter t
00000208	68	65	20	6E	75	6D	62	65	72	20	74	6F	20	63	68	6D	he number to cho
00000224	6F	73	65	20	74	68	65	20	64	69	66	66	65	72	65	6E	ose the differen
00000240	74	70	70	72	6F	67	72	61	6D	3A	20	0A	0D	31	2E	4E	t program: 1.N
00000256	75	6D	62	65	72	0A	0D	33	2E	4E	61	6D	65	0A	0D	33	umber 2.Name 3
00000272	2E	52	65	63	74	61	6E	67	6C	65	0A	0D	34	2E	53	74	.Rectangle 4.St
00000288	6F	6E	65	0A	0D	00	00	00	00	00	00	00	00	00	00	00	one
00000304	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000320	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000336	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000352	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000368	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000384	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000400	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000416	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000432	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000448	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000464	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000480	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000496	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000512	B4	06	B0	00	B5	00	B1	00	B6	18	B2	4F	B7	07	CD	10	* ± u ± 'O' i
00000528	8C	C8	8E	D8	BD	B8	7C	8C	D8	8E	C0	FF	0E	B3	82	75	gēzōw, zāy f, u
00000544	FA	C7	06	B3	82	50	C3	FF	0E	B3	82	75	EE	C7	06	83	ūq f, pāy -, uīq f
00000560	B2	50	C3	C7	06	B3	82	44	02	B0	01	3A	06	B7	B2	74	, Pāq -, D * : t, t
00000576	1E	B0	02	3A	06	B7	B2	74	53	B0	03	3A	06	B7	B2	0F	* : t, tS * : t,
00000592	84	B5	00	B0	04	3A	06	B7	B2	0F	B4	B5	00	EB	FE	FF	- : t, t, - u ēpy
00000608	06	B8	82	FF	06	B8	82	8B	1E	B8	82	B8	0E	00	29	D8	, y S, (,) Ø
00000624	74	0E	8B	1E	8A	82	B8	28	00	29	D8	74	11	E9	CC	00	t < S, () t < 6
00000640	C7	06	B8	82	OC	00	C6	06	B7	B2	04	EB	82	0E	00	C6	Ĉ, ± t, 6h Ĉ
00000656	8A	82	26	00	C6	06	B7	82	04	E9	B0	00	FF	0E	B8	82	S, ± t, 6h y
00000672	06	B8	82	8B	1E	8A	82	B8	28	00	29	D8	74	0E	8B	y S, () t < c	
00000688	1E	B8	82	B8	FF	FF	29	D8	74	11	E9	8F	00	C7	06	8A	, yy) t < 6 Ĉ
00000704	B2	26	00	C6	06	B7	82	03	E9	B1	00	C7	06	B8	82	01	, ± t, 6 Ĉ
00000720	FF	C6	06	B7	82	01	EB	74	FF	0E	B8	82	FF	0E	B8	82	± t, ety, y S,

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	ASCII	
00000736	8B	1E	88	82	B8	FF	FF	29	D8	74	0D	8B	1E	8A	82	B8	< ', yy) t < S,	
00000752	FF	FF	29	D8	74	0F	EB	54	C7	06	B8	82	01	00	C6	06	yy) t < 6 Ĉ, ± t	
00000768	B7	B2	04	EB	47	C7	06	8A	B2	01	00	C6	06	B7	B2	02	t, 6q Ĉ, ± t,	
00000784	EB	3A	FF	06	B8	B2	FF	0E	8A	B2	8B	1E	8A	82	B8	FF	ety, y S, () S, y	
00000800	FF	29	D8	74	0D	8B	1E	88	B2	B8	0E	00	29	D8	74	0F	y) t < ',) t <	
00000816	EB	1A	C7	06	8A	B2	01	00	C6	06	B7	82	01	EB	00	C7	ē Ĉ S, ± t, ē Ĉ	
00000832	06	B8	82	OC	00	C6	06	B7	B2	03	EB	08	00	31	C0	A1	B8	, ± t, ē ĬĬ,
00000848	B2	B8	50	00	FF	E3	03	06	8A	B2	BB	02	00	FF	E3	B9	, » ± S, » -āh	
00000864	C3	BE	8C	B2	B9	08	00	8A	04	26	B9	07	46	43	E2	E2	ĬĬĬĬ: S āh FCCĬ	
00000880	F6	B4	01	CD	16	B3	20	38	D8	74	03	E9	9D	FE	E9	7F	o' i' S t < 6 pē	
00000896	F9	EB	FE	50	C3	44	02	01	FF	FF	00	00	00	31	39	33	33	ēp pĬĬ yy 1933
00000912	35	31	37	34	00	00	00	00	00	00	00	00	00	00	00	00	00	5174
00000928	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000944	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000960	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000976	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000992	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00001008	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00001024	B4	06	B0	00	B5	00	B1	00	B6	18	B2	4F	B7	07	CD	10	* ± u ± 'O' i	
00001040	8C	C8	8E	D8	BD	B8	7C	8C	D8	8E	C0	FF	0E	B3	82	75	gēzōw, zāy f, u	
00001056	FA	C7	06	B3	82	50	C3	FF	0E	B3	82	75	EE	C7	06	83	ūq f, pāy -, uīq f	
00001072	B2	50	C3	C7	06	B3	82	44	02	B0	01	3A	06	B7	B2	74	, Pāq -, D * : t, t	
00001088	1E	B0	02	3A	06	B7	B2	74	53	B0	03	3A	06	B7	B2	0F	* : t, tS * : t,	
00001104	84	B5	00	B0	04	3A	06	B7	B2	0F	B4	B5	00	EB	FE	FF	- : t, t, - u ēpy	
00001120	06	B8	82	FF	06	B8	82	8B	1E	B8	82	B8	0E	00	29	D8	, y S, (,) Ø	
00001136	74	0E	8B	1E	8A	82	B8	28	00	29	D8	74	11	E9	CC	00	t < S, () t < 6	
00001152	C7	06	B8	82	17	00	C6	06	B7	B2	02	E9	BE	00	C7	06	Ĉ, ± t, 6h Ĉ	
00001168	8A	B2	26	00	C6	06	B7	82	04	E9	B0	00	FF	0E	B8	82	S, ± t, 6h y	
00001184	FF	06	8A	82	8B	1E	8A	82	B8	28	00	29	D8	74	0E	8B	y S, () t < c	
00001200	1E	B8	82	B8	OC	00	29	D8	74	11	E9	8F	00	C7	06	8A	,) t < 6 Ĉ	
00001216	B2	26	00	C6	06	B7	B2	03	E9	B1	00	C7	06	B8	82	0E	, ± t, 6 Ĉ	
00001232	00	C6	06	B7	B2	01	EB	74	FF	0E	B8	82	FF	0E	B8	82	± t, ety, y S,	
00001248	8B	1E	88	82	B8	OC	00	29	D8	74	0D	8B	1E	8A	82	B8	< ',) t < S,	
00001264	FF	FF	29	D8	74	0F	EB	54	C7	06	B8	82	0E	00	C6	06	yy) t < 6 Ĉ, ± t	
00001280	B7	B2	04	EB	47	C7	06	8A	B2	01	00	C6	06	B7	B2	02	t, 6q Ĉ, ± t,	
00001296	EB	3A	FF	06	B8	B2	FF	0E	8A	B2	8B	1E	8A	82	B8	FF	ety, y S, () S, y	
00001312	FF	29	D8	74	0D	8B	1E	88	B2	B8	0E	00	29	D8	74	0F	y) t < ',) t <	
00001328	EB	1A	C7	06	8A	B2	01	00	C6	06	B7	82	01	EB	00	C7	ē Ĉ S, ± t, ē Ĉ	
00001344	06	B8	82	17	00	C6	06	B7	B2	03	EB	00	00	31	C0	A1	B8	, ± t, ē ĬĬ,
00001360	B2	B8	50	00	FF	E3	03	06	8A	B2	BB	02	00	FF	E3	B9	, » ± S, » -āh	
00001376	C3	BE	8C	B2	B9	03	00	8A	04	26	B9	07	46	43	E2	E2	ĬĬĬĬ: S āh FCCĬ	

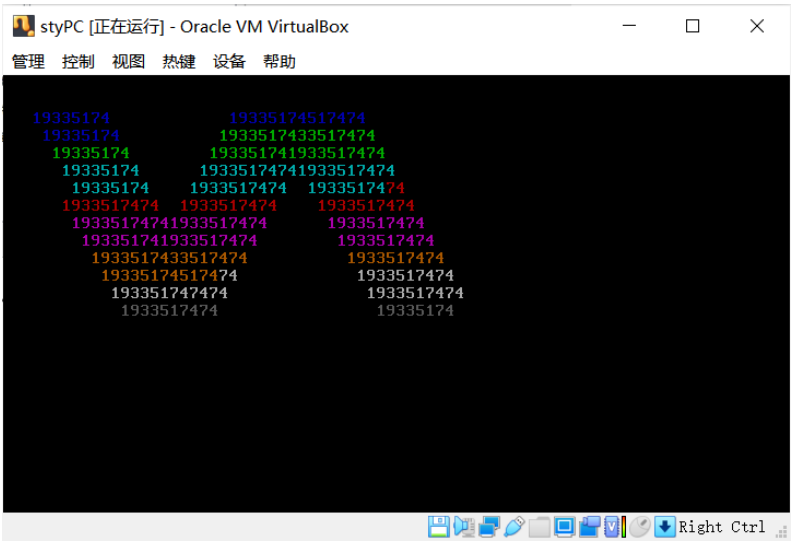
6、用软盘启动裸机



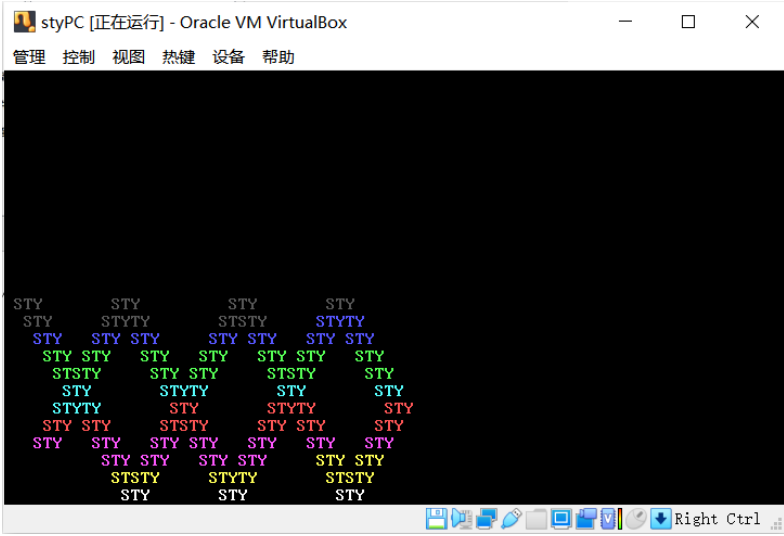
监控程序



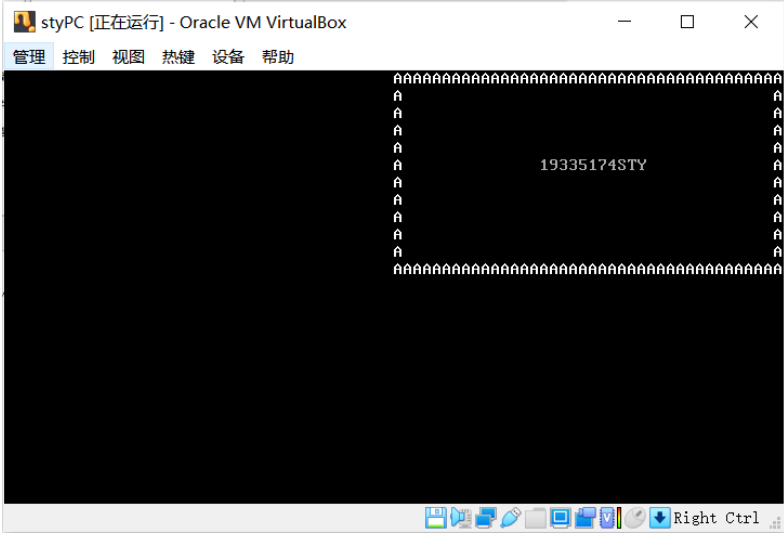
用户程序 1



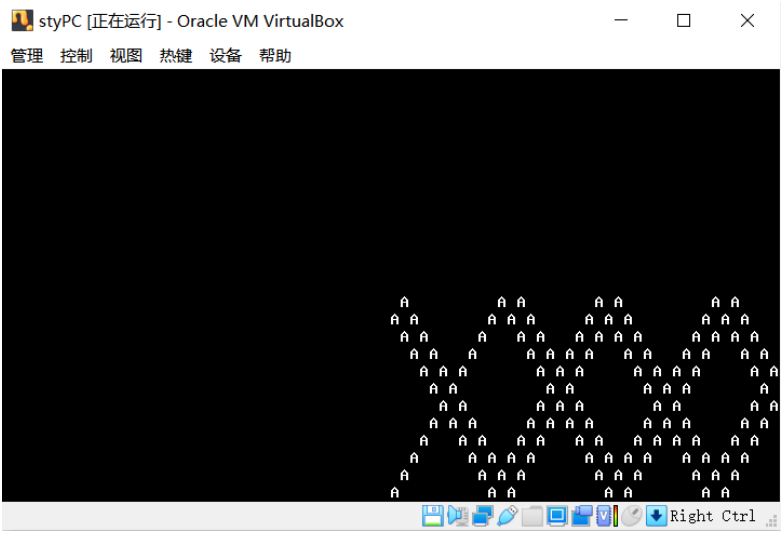
用户程序 2



用户程序 3



用户程序 4



六、实验总结

在这次实验中，我对软盘驱动裸机有了更深入的了解，自己写一个监控程序和几个用户程序，并且如何调用用户程序和调用完成时返回是我一开始困惑的地方。我也明白了如何用 int 16 的功能，int 10 输出字符串，也明白了 jmp 更多的用处。

在实验过程中，我也遇到了几个问题。比如我一开始以为 0x55 和 0xaa 是每个扇区末尾都要加的，导致我的程序出错，后来我才知道这只是引导扇区所要求的。在设置字符串的位置时，我一开始把 mov dh, 05h 和 mov dl, 37h 合并成了 mov dx 0537h，却不能正常显示字符串，也不知道是什么原因。在设计用户程序如何返回监控程序时，一开始不知道如何操作，后来发现老师 ppt 有 int 16 的 1 号功能可以读键盘才设计出来，但在返回时，我一开始想用 ret 指令却不能正常返回，后来想起可以用 jmp 7c00h 才完成了这个功能。希望这些我现在还有疑问的问题在日后我都能顺利解决！

虽然整个实验让我精疲力竭，在一次次编译和调试失败时我都痛苦万分，但俗话说“功夫不负有心人”，经过长时间的努力，收获成功时的我也是快乐无比的。希望在后面的实验中，我也能再接再厉，继续加油，勇往直前！

七、参考文献

《汇编语言 （第 3 版）》

《X86 汇编语言：从实模式到保护模式》

<https://blog.csdn.net/miragel993/article/details/29908929>

<https://blog.csdn.net/iostream992/article/details/83077838>