

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. 19

一、实验题目

实验环境与引导程序设计

二、实验目的

- 1、掌握 NASM 的语法
- 2、掌握用汇编器的用法
- 3、掌握创建空白软盘镜像的方法
- 4、掌握虚拟机的设置及使用方法

三、实验要求

- 1、搭建和应用实验环境

虚拟机安装，生成一个基本配置的虚拟机 XXXPC 和多个 1.44MB 容量的虚拟软盘，用 WinHex 工具将其中一个虚拟软盘的首扇区填满你的个人信息。

- 2、设计引导扇区程序

设计 IBM_PC 的一个引导扇区程序，程序是前两次实验设计的程序基础上，增加更多个性扩展，如炫酷动态变色，个性画面，如此等等，自由不限。还要在屏幕某个区域特别的方式显示你的学号姓名等个人信息。将这个程序的机器码放进第二张虚拟软盘的首扇区，并用此软盘引导你的 XXXPC，直到成功。

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】

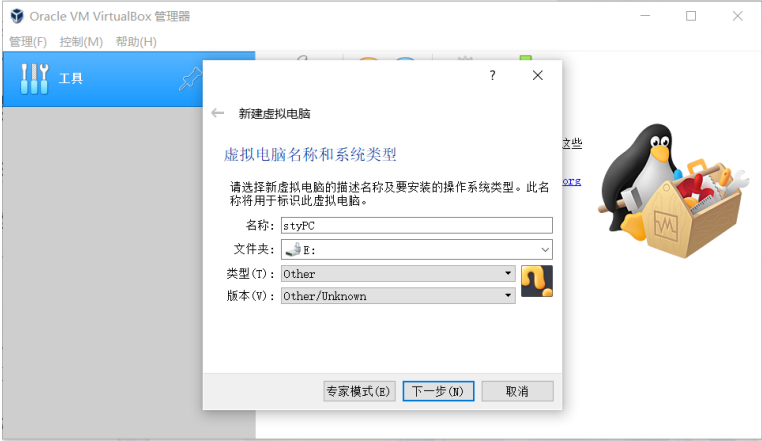
- 1、汇编语言：NASM
- 2、文本编辑器：Notepad++
- 3、相关软件：DOSBox, WinHex

五、实验过程

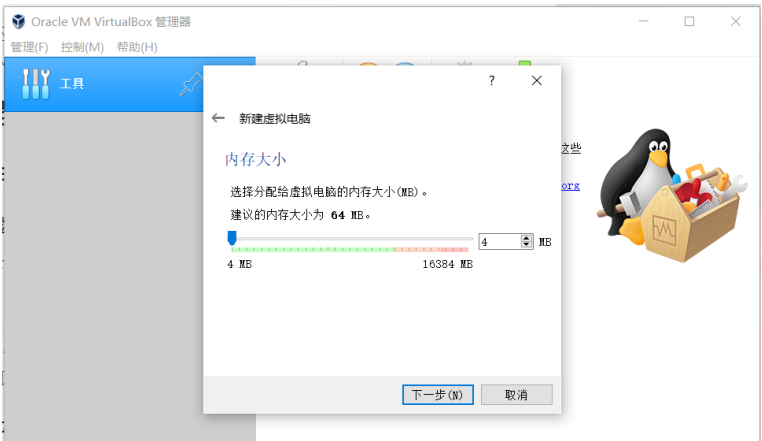
1、搭建和应用实验环境

【安装虚拟机】

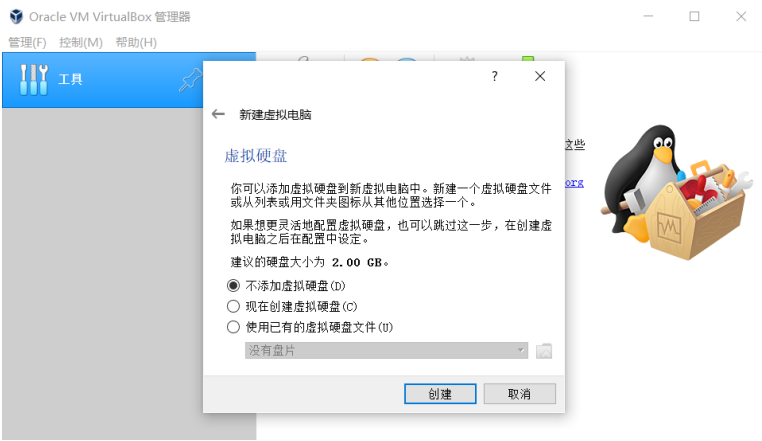
打开 VirtualBox，新建无操作系统的虚拟机



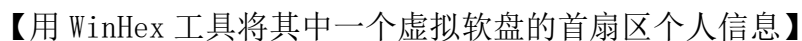
内存设置为 4MB



选择不添加虚拟硬盘



创建虚拟软盘后，选择注册虚拟软盘



```
1 #include <stdio.h>
2 char s[] = "19335174STY", c[1440 << 10];
3 int main() {
4     FILE* fp;
5     fread(c, sizeof(*c), sizeof(c), fopen("E:\\styPC\\styPC_1.img", "rb+"));
6     for (int i = 91; i < 510; ++i) {
7         c[i] = s[i % (sizeof(s) - 1)];
8     }
9     fwrite(c, sizeof(*c), sizeof(c), fopen("E:\\styPC\\a.img", "wb"));
10    return 0;
11 }
```

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	EB	3C	90	49	50	52	54	20	36	2E	32	00	02	01	01	00	ë< IPRT 6.2
00000010	02	E0	00	40	0B	F0	09	00	12	00	02	00	00	00	00	00	à @ ð
00000020	00	00	00	00	00	00	00	29	E2	7D	C1	98	20	20	20	20)a)ã~
00000030	20	20	20	20	20	20	46	41	54	31	32	20	20	20	CD	19	FAT12 í
00000040	EC	CC	00	00	00	00	00	00	00	00	00	00	00	00	00	00	ïî
00000050	00	00	00	00	00	00	00	00	00	00	00	33	35	31	37	34	35174
00000060	53	54	59	31	39	33	33	35	31	37	34	53	54	59	31	39	STY19335174STY19
00000070	33	33	35	31	37	34	53	54	59	31	39	33	33	35	31	37	335174STY1933517
00000080	34	53	54	59	31	39	33	33	35	31	37	34	53	54	59	31	4STY19335174STY1
00000090	39	33	33	35	31	37	34	53	54	59	31	39	33	33	35	31	9335174STY193351
000000A0	37	34	53	54	59	31	39	33	33	35	31	37	34	53	54	59	74STY19335174STY
000000B0	31	39	33	33	35	31	37	34	53	54	59	31	39	33	33	35	19335174STY19335
000000C0	31	37	34	53	54	59	31	39	33	33	35	31	37	34	53	54	174STY19335174ST
000000D0	59	31	39	33	33	35	31	37	34	53	54	59	31	39	33	33	Y19335174STY1933
000000E0	35	31	37	34	53	54	59	31	39	33	33	35	31	37	34	53	5174STY19335174S
000000F0	54	59	31	39	33	33	35	31	37	34	53	54	59	31	39	33	TY19335174STY193
00000100	33	35	31	37	34	53	54	59	31	39	33	33	35	31	37	34	35174STY19335174
00000110	53	54	59	31	39	33	33	35	31	37	34	53	54	59	31	39	STY19335174STY19
00000120	33	33	35	31	37	34	53	54	59	31	39	33	33	35	31	37	335174STY1933517
00000130	34	53	54	59	31	39	33	33	35	31	37	34	53	54	59	31	4STY19335174STY1
00000140	39	33	33	35	31	37	34	53	54	59	31	39	33	33	35	31	9335174STY193351
00000150	37	34	53	54	59	31	39	33	33	35	31	37	34	53	54	59	74STY19335174STY
00000160	31	39	33	33	35	31	37	34	53	54	59	31	39	33	33	35	19335174STY19335
00000170	31	37	34	53	54	59	31	39	33	33	35	31	37	34	53	54	174STY19335174ST
00000180	59	31	39	33	33	35	31	37	34	53	54	59	31	39	33	33	Y19335174STY1933
00000190	35	31	37	34	53	54	59	31	39	33	33	35	31	37	34	53	5174STY19335174S
000001A0	54	59	31	39	33	33	35	31	37	34	53	54	59	31	39	33	

2、设计引导扇区程序

【设计代码】

```
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 7c00h
8   start:
9       mov ax,cs          ;获得程序运行时,代码段在内存的位置
10      mov ds,ax          ; DS = CS
11      mov ss,ax          ; SS = CS
12      mov ax,0B800h       ; 文本窗口显存起始地址
13      mov es,ax          ; ES = B800h
14  loop1:
15      dec word[count]     ; 递减计数变量
16      jnz loop1          ; >0: 跳转;
17      mov word[count],delay
18      dec word[dcount]    ; 递减计数变量
19      jnz loop1
20      mov word[count],delay
21      mov word[dcount],ddelay
22
23      mov al,1
24      cmp al,byte[rdul]
25      jz DnRt
26      mov al,2
27      cmp al,byte[rdul]
28      jz UpRt
29      mov al,3
30      cmp al,byte[rdul]
31      jz UpLt
32      mov al,4
33      cmp al,byte[rdul]
34      jz DnLt
35      jmp $
36
37  DnRt:
38      inc word[x]
39
40      inc word[y]
41      mov bx,word[x]
42      mov ax,25
43      sub ax,bx
44      jz dr2ur
45      mov bx,word[y]
46      mov ax,80
47      sub ax,bx
48      jz dr2dl
49      jmp show
50  dr2ur:
51      mov word[x],23
52      mov byte[rdul],Up_Rt
53      jmp show
54  dr2dl:
55      mov word[y],78
56      mov byte[rdul],Dn_Lt
57      jmp show
58  UpRt:
59      dec word[x]
60      inc word[y]
61      mov bx,word[y]
62      mov ax,80
63      sub ax,bx
64      jz ur2ul
65      mov bx,word[x]
66      mov ax,-1
67      sub ax,bx
68      jz ur2dr
69      jmp show
70  ur2ul:
71      mov word[y],78
72      mov byte[rdul],Up_Lt
73      jmp show
74  ur2dr:
75      mov word[x],1
76      mov byte[rdul],Dn_Rt
77      jmp show
```

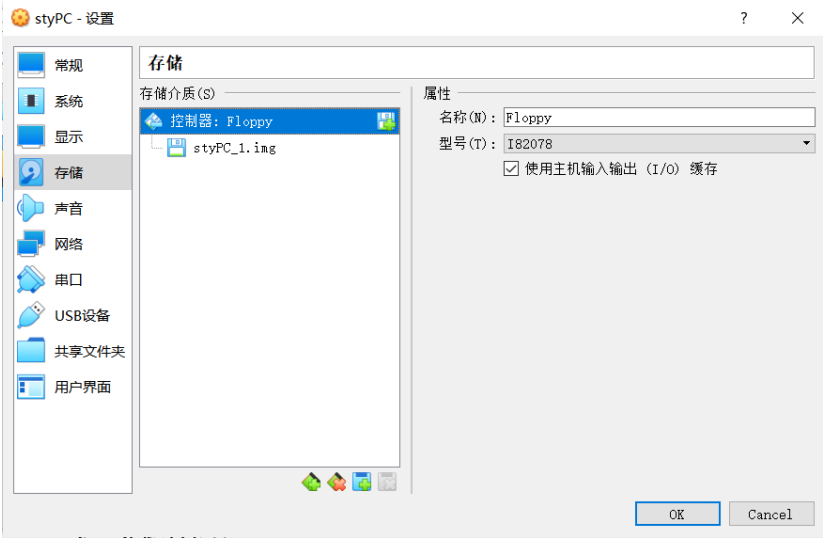
```

77 UpLt:
78     dec word[x]
79     dec word[y]
80     mov bx,word[x]
81     mov ax,-1
82     sub ax,bx
83     jz  ul2dl
84     mov bx,word[y]
85     mov ax,-1
86     sub ax,bx
87     jz  ul2ur
88     jmp show
89 ul2dl:
90     mov word[x],1
91     mov byte[rdul],Dn_Lt
92     jmp show
93 ul2ur:
94     mov word[y],1
95     mov byte[rdul],Up_Rt
96     jmp show
97 DnLt:
98     inc word[x]
99     dec word[y]
100    mov bx,word[y]
101    mov ax,-1
102    sub ax,bx
103    jz  dl2dr
104    mov bx,word[x]
105    mov ax,25
106    sub ax,bx
107    jz  dl2ul
108    jmp show
109 dl2dr:
110    mov word[y],1
111    mov byte[rdul],Dn_Rt
112    jmp show
113 dl2ul:
114    mov word[x],23
115
116    mov byte[rdul],Up_Lt
117    jmp show
118 show:
119    xor ax,ax
120    mov word ax,[x]
121    mov bx,80
122    mul bx
123    add word ax,[y]
124    mov bx,2
125    mul bx
126    mov bx,ax
127    mov si, NAME
128    mov cx, 11
129    color:
130        mov byte al,[si]
131        mov [es:bx],ax
132        inc si
133        inc bx
134        inc bx
135    loop color
136    jmp loop1
137 end:
138     jmp $ ; 停止画框，无限循环
139
140 datadef:
141     count dw delay
142     dcount dw ddelay
143     rdul db Dn_Rt ; 向右下运动
144     x dw -1
145     y dw 0
146     NAME db "19335174STY"
147
148 times 510-($-$$) db 0
149 db 0x55,0xaa

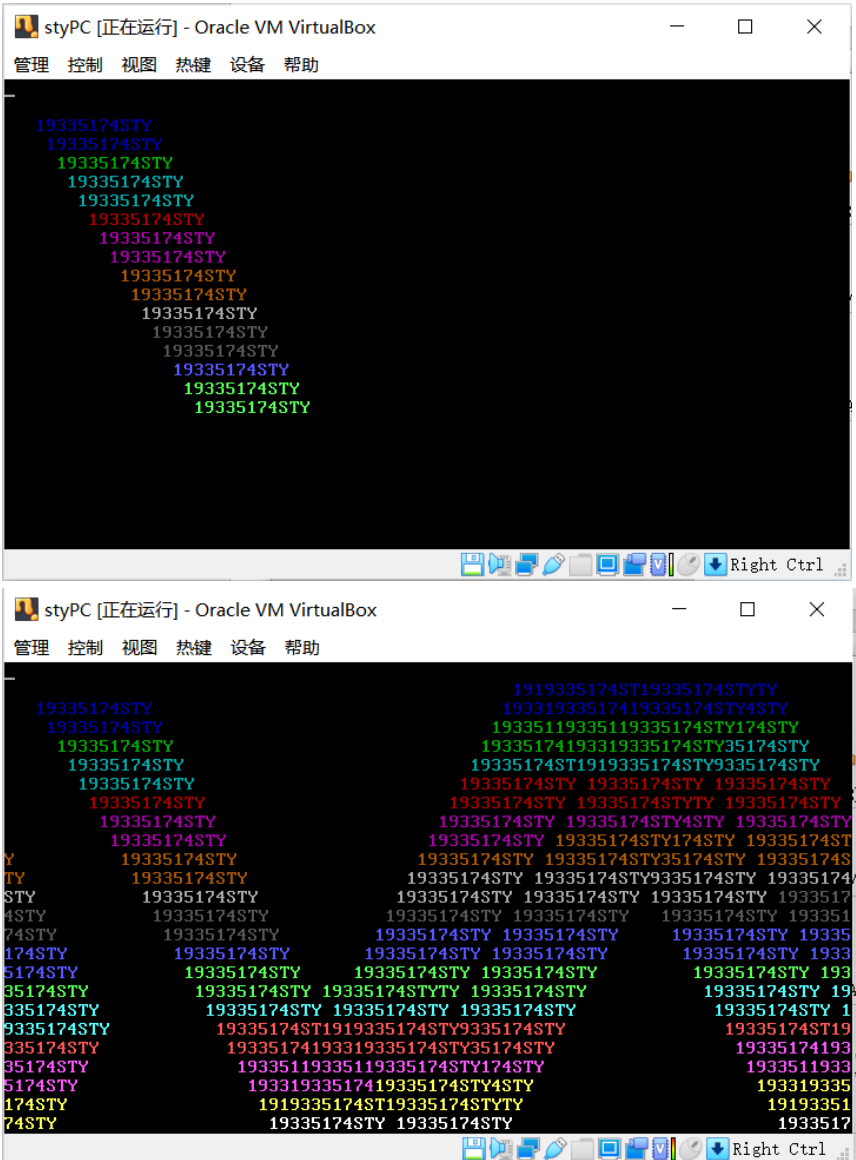
```

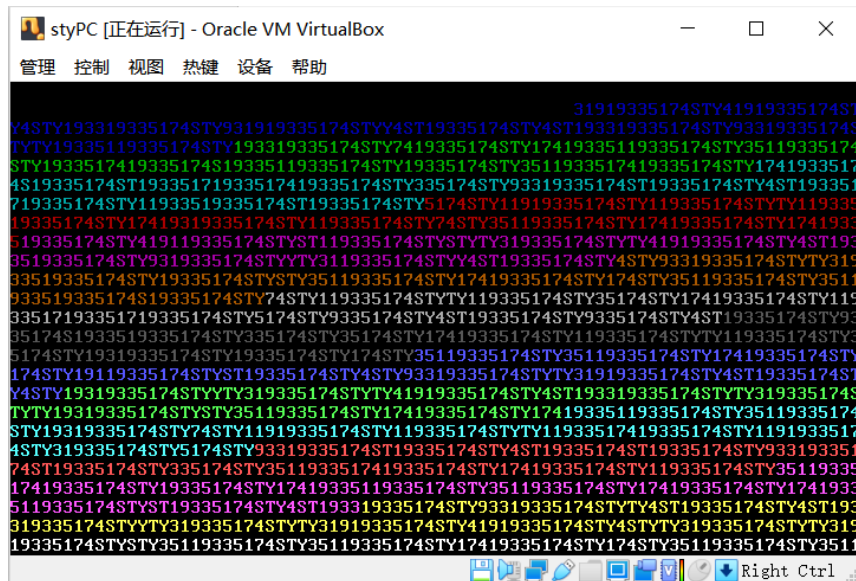
[illegible]

将软盘文件放入虚拟机中，作为引导扇区程序



打开虚拟机，结果正确！





六、实验总结

在这次实验中，我学会了如何安装创建虚拟机，还学会了创建软盘并作为扇区引导程序启动虚拟机，展现我自己写的汇编程序。

虽然这次的代码部分相对比较简单，只是更改了颜色，增加了学号姓名，我也创新地尝试用 C 代码写程序将个人信息填入软盘文件中。但是在整个实验过程中我还是遇到了许多问题，比如我一开始使用的 WinHex 是试用版，导致我在尝试将 bb.bin 复制到 img 软盘文件时反馈说无法保存 200kb 以上文件，浪费了我很多时间。。。还有我在 DOSBox 和虚拟机中都运行了我的汇编代码程序，我发现 delay equ 50000 这句代码，在 DOSBox 中字符串会移动地特别慢，但在虚拟机中移动速度就恰好合适，也不知道是什么原因。

这周终于开始了真的操作系统实验，能在一个裸机上直接展示出我写的程序，让我十分自豪，兴奋不已。我知道后面的实验会越来越难，但我也准备好迎接挑战了。只有不断努力，才能有所收获。

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/lulipeng_cpp/article/details/8161982