

实验1：接管裸机的控制权

院系	专业	年级	姓名
数据科学与计算机学院	人工智能与大数据	2018级	陈琮昊

一、实验目的：

- 1、了解原型操作系统设计实验教学方法与要求
- 2、了解计算机硬件系统开机引导方法与过程
- 3、掌握操作系统的引导程序设计方法与开发工具
- 4、复习加强汇编语言程序设计能力

二、实验要求：

- 1、知道原型操作系统设计实验的两条线路和前6个实验项目的差别
- 2、掌握PC电脑利用 1.44MB 软驱的开机引导方法与过程的步骤
- 3、在自己的电脑上安装配置引导程序设计的开发工具与环境
- 4、参考样版汇编程序，完成在PC虚拟机上设计一个 1.44MB 软驱的引导程序的完整工作。
- 5、编写实验报告，描述实验工作的过程和必要的细节，以证实实验工作的真实性。

三、实验方案：

1.运行环境：

该实验在Windows下运行。

所需软件：VMware Workstation 15.5 Pro 、 NASM 2.14.02

虚拟机 VMware 用来运行自己设计的操作系统， NASM 用来进行汇编语言的编译。

2.实验流程：

在软硬件配备好以后，修改 `stone.asm` 以实现要求的功能，然后使用 NASM 转为机器代码，得到 `.img` 文件，并放入虚拟机，启动虚拟机，观察结果。

3.程序分析：

主函数是用来引导扇区并将整个过程的大致流程表示出来：

```

mov ax,cs
mov ds,ax          ; DS = CS
mov es,ax          ; ES = CS
mov ax,0b800h
mov gs,ax
call ifm           ;显示学号和姓名
call start         ;显示字符
jmp $

```

可以看到主函数中调用了两个函数，`ifm`和`start`，`ifm`是显示学号、姓名信息的程序，`start`则是实现反弹程序。

`ifm`的实现较为简单，此处不予赘述。

`start`程序开始则是确定显存起始地址，`loop1`程序开始则是进行延时操作。

接下来的代码块则是通过比较来确定运动方向（共4种）：

```

mov al,1
cmp al,byte[rdu1]
jz DnRt
mov al,2
cmp al,byte[rdu1]
jz UpRt
mov al,3
cmp al,byte[rdu1]
jz UpLt
mov al,4
cmp al,byte[rdu1]
jz DnLt
jmp $

```

接下来就是四个方向运动的代码块，在这个代码块需要考虑碰到边界时可能弹的方向，因此还需找出判断的临界条件。此处举一例来做分析，其他的三个运动方向同理。

```

; 右下方向
DnRt:
    inc word[x]
    inc word[y]
    mov ax,word[x]          ;此时位置的x坐标
    cmp ax,81               ;和长方形区域的长比较
    jz dr2d1                ;若到达x方向最大值，则往左下弹（长方形的竖直方向的边为反射面）
    mov ax,word[y]          ;此时位置的y坐标
    cmp ax,26               ;和长方形的宽比较
    jz dr2ur                ;若到达y方向最大值，则往右上弹（长方形的水平方向的边为反射面）
    jmp show

dr2d1:
    mov word[x],79
    mov byte[rdu1],Dn_Lt    ;往左下弹
    jmp show

```

```
dr2ur:
    mov word[y],24
    mov byte[rdu1],Up_Rt    ;往右上弹
    jmp show
```

再往后则是在画面显示的代码，这部分代码通过修改一些参数可以显示特定的字符与特定的颜色，当然还可以实现变色变字符这样更强大的功能，在下一部分有解释，此处就不再解释了。

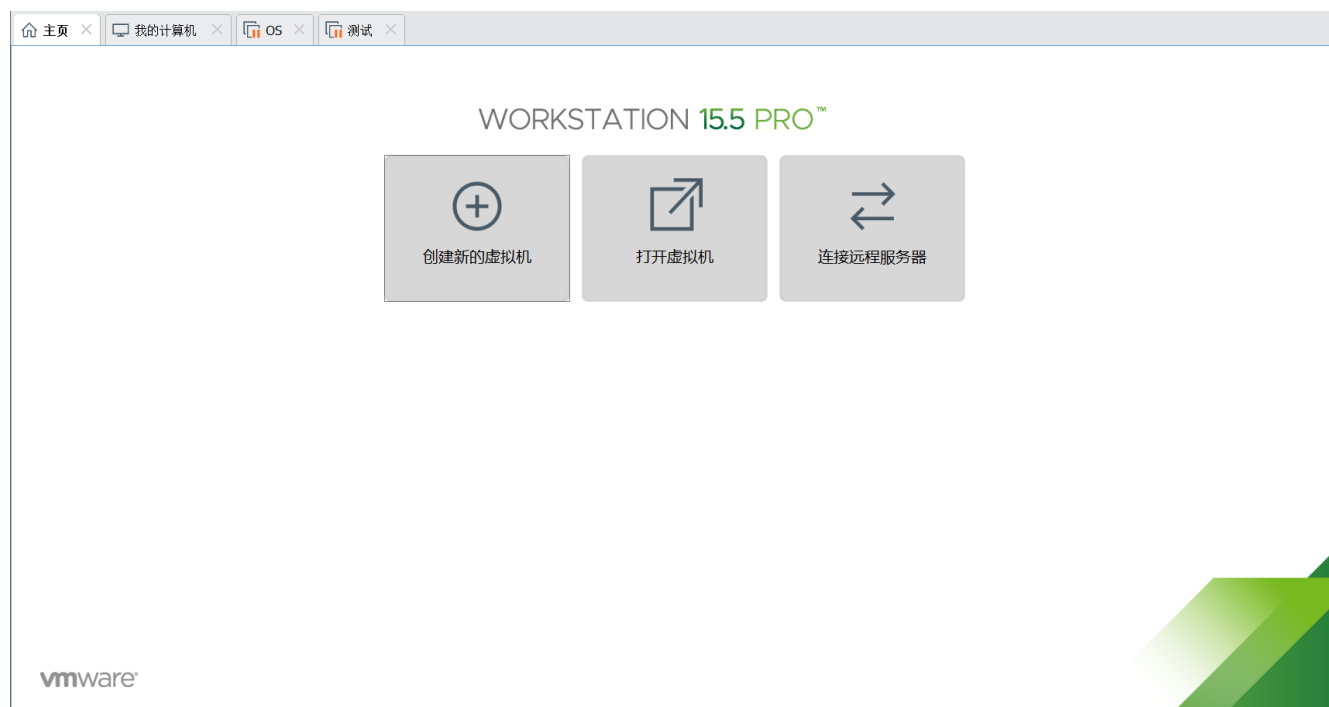
四、实验过程：

1.软件的下载与安装：

在官网下载 VMware Workstation 15.5 Pro：

<https://www.vmware.com/cn/products/workstation-pro/workstation-pro-evaluation.html>

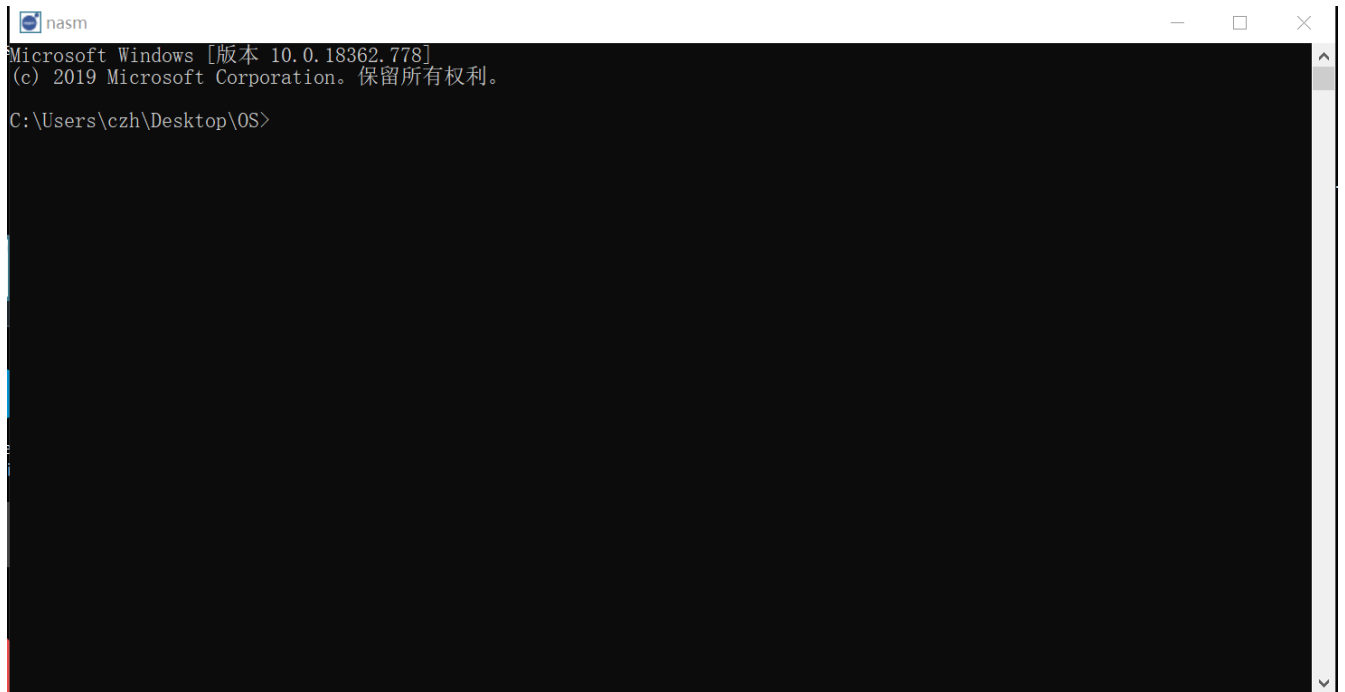
安装该软件后初始界面如下（已经输入密钥并初步设置后）：



在官网下载 NASM 2.14.02：

<https://www.nasm.us/>

安装后打开 NASM 的界面如下：



2.编写汇编代码：

根据老师所给的参考文件 `stone.asm` 进行修改，代码如下：

(1) 基础实验：

```
;程序源代码 (stone.asm)
    org 07c00h                ; 程序加载到100h, 可用于生成COM
    Dn_Rt equ 1                ; D-Down, U-Up, R-right, L-Left
    Up_Rt equ 2
    Up_Lt equ 3
    Dn_Lt equ 4
    delay equ 50000            ; 计时器延迟计数, 用于控制画框的速度
    ddelay equ 580             ; 计时器延迟计数, 用于控制画框的速度
;开始引导扇区
    mov ax, cs
    mov ds, ax                ; DS = CS
    mov es, ax                ; ES = CS
    mov ax, 0b800h
    mov gs, ax
    call ifm                  ; 显示学号和姓名
    call start                ; 显示字符
    jmp $
;显示学号和姓名的代码块
ifm:
    mov bp, 0                 ; BP为当前串的偏移地址
    mov byte[gs:bp+0], '1'
    mov byte[gs:bp+2], '8'
    mov byte[gs:bp+4], '3'
    mov byte[gs:bp+6], '4'
    mov byte[gs:bp+8], '0'
    mov byte[gs:bp+10], '0'
```



```

        jmp show

dr2dl:
    mov word[x],79
    mov byte[rdu1],Dn_Lt    ;往左下弹
    jmp show

dr2ur:
    mov word[y],24
    mov byte[rdu1],Up_Rt    ;往右上弹
    jmp show
;右上:
UpRt:
    inc word[x]
    dec word[y]
    mov ax,word[y]
    cmp ax,1
    jz ur2dr                ;往右下弹 (原理同上)
    mov ax,word[x]
    cmp ax,81
    jz ur2ul                ;往左上弹 (原理同上)
    jmp show

ur2dr:
    mov word[y],3
    mov byte[rdu1],Dn_Rt
    jmp show

ur2ul:
    mov word[x],79
    mov byte[rdu1],Up_Lt
    jmp show
;左上
UpLt:
    dec word[x]
    dec word[y]
    mov ax,word[x]
    cmp ax,0
    jz ul2ur                ;往右上弹
    mov ax,word[y]
    cmp ax,1
    jz ul2dl                ;往左下弹
    jmp show

ul2ur:
    mov word[x],2
    mov byte[rdu1],Up_Rt
    jmp show

ul2dl:
    mov word[y],3
    mov byte[rdu1],Dn_Lt

```

```

    jmp show
;左下
DnLt:
    dec word[x]
    inc word[y]
    mov ax,word[x]
    cmp ax,0
    jz d12dr                ;往右下弹
    mov ax,word[y]
    cmp ax,26
    jz d12ul                ;往左上弹
    jmp show

d12dr:
    mov word[x],2
    mov byte[rdu1],Dn_Rt
    jmp show

d12ul:
    mov word[y],24
    mov byte[rdu1],Up_Lt
    jmp show
; 显示
show:
    mov ax,word[y]
    dec ax
    mov bx,160
    mul bx
    mov cx,ax
    mov ax,word[x]
    dec ax
    add ax,ax
    add ax,cx
    mov bp,ax
    mov ah,0Fh                ; 0000: 黑底、1111: 亮白字 (默认值为07h)
    mov al,byte[char]         ; AL = 显示字符值 (默认值为20h=空格符)
    mov word[gs:bp],ax        ; 显示字符的ASCII码值
    jmp loop1

end:
    jmp $

datadef:
    count dw delay
    dcount dw ddelay
    rdu1 db Dn_Rt
    x     dw 3
    y     dw 1
    char db 'A'

```

(2) 复杂实验:

大部分代码都相同，只是增加了变色和变字符的部分，其中字符是从a变到z（26个英文字母小写），颜色则是10个为一个周期。下面的代码块为复杂实验与基础实验的差异部分：

首先起始字母由A变成a，只需小小的改动：

```
start:
    mov ax,0B800h          ; 文本窗口显存起始地址
    mov gs,ax              ; GS = B800h
    mov byte[char], 'a'    ; 由'A'变成了'a'
```

然后就是显示的过程需要实时变换字符和颜色：

```
;显示
show:
    mov ax,word[y]
    dec ax
    mov bx,160
    mul bx
    mov cx,ax      ;存储在cx
    mov ax,word[x]
    dec ax
    add ax,ax
    add ax,cx
    mov bp,ax
    mov ah,byte[color]      ; 变换颜色
    mov al,byte[char]      ; 变换字符
    mov word[gs:bp],ax      ; 显示字符的ASCII码值
    cmp ah,10              ; 最后为10号颜色
    jnz bianse
    mov byte[color],1      ; 初始为1号颜色
    jmp go
;变色
bianse:
    inc byte[color]        ; 颜色变化+1

go:
    mov ah,byte[char]
    cmp ah,122             ; 最后为字符z（其ASCII码为122）
    jnz bianzi
    mov byte[char],97      ; 初始为字符a（其ASCII码为97）
    jmp goway
;变字符
bianzi:
    inc byte[char]        ; ASCII码+1

goway:
    jmp loop1             ; 变换颜色字符过程结束返回循环

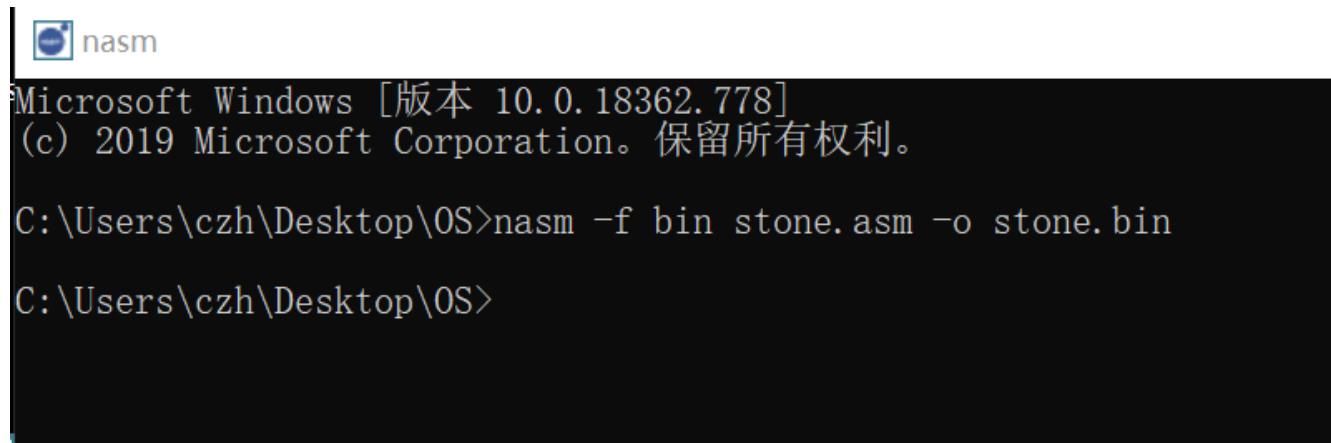
end:
    jmp $
```



```
datadef:
    count dw delay
    dcount dw ddelay
    rdu1 db Dn_Rt
    x     dw 3
    y     dw 1
    char db 97
    color db 6
```

3.配置所需文件:

在这一步，直接用记事本把所编写的汇编语言程序保存起来，命名为了方便仍为 `stone.asm`。将这个记事本文件放入 `NASM` 的工作目录下，然后打开 `NASM`，敲入如下指令：





```
nasm
Microsoft Windows [版本 10.0.18362.778]
(c) 2019 Microsoft Corporation。保留所有权利。

C:\Users\czh\Desktop\OS>nasm -f bin stone.asm -o stone.bin

C:\Users\czh\Desktop\OS>
```

接下来可以在和 `stone.asm` 以及 `NASM` 相同的目录下找到文件 `stone.bin`：

 stone.asm	2020/4/26 14:47	ASM 文件	4 KB
 stone.bin	2020/4/26 15:35	BIN 文件	1 KB

至此，就将编写好的汇编代码编译成了机器代码，接下来的工作就是导入虚拟机并运行。

4.导入虚拟机:

打开虚拟机，选择 `创建新的虚拟机` 一项，然后进行如下配置：

安装客户机操作系统

虚拟机如同物理机，需要操作系统。您将如何安装客户机操作系统？

☒ 稍后安装操作系统(S)。

创建的虚拟机将包含一个空白硬盘。

帮助

< 上一步(B)

下一步(N) >

取消

新建虚拟机向导



选择客户机操作系统

此虚拟机中将安装哪种操作系统？

客户机操作系统

☐ Microsoft Windows(W)

☐ Linux(L)

☐ VMware ESX(X)

☒ 其他(O)

版本(V)

其他



帮助

< 上一步(B)

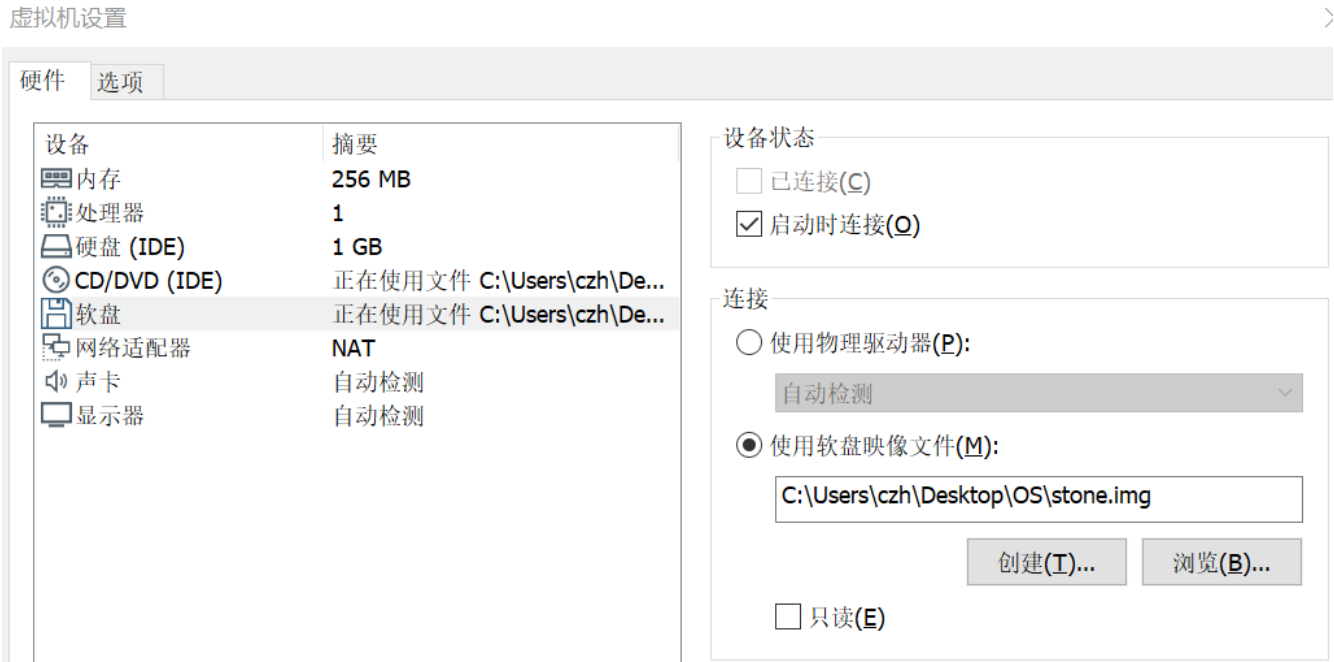
下一步(N) >

取消

于是所使用的虚拟机也已配置完成。（后续步骤中选择磁盘容量可视情况而定，其他步骤则不赘述了。）

前一步生成了 stone.bin 文件，获得 stone.img 文件最简便的方式就是将 stone.bin 文件复制一份并直接进行重命名，命名为 stone.img，接下来就是将 stone.img 文件导入虚拟机，步骤如下：

打开已创建好的虚拟机，然后选择 编辑虚拟机设置，然后添加 软盘，在 连接 一栏选择 使用软盘映像文件，将 stone.img 导入，确定 即可。

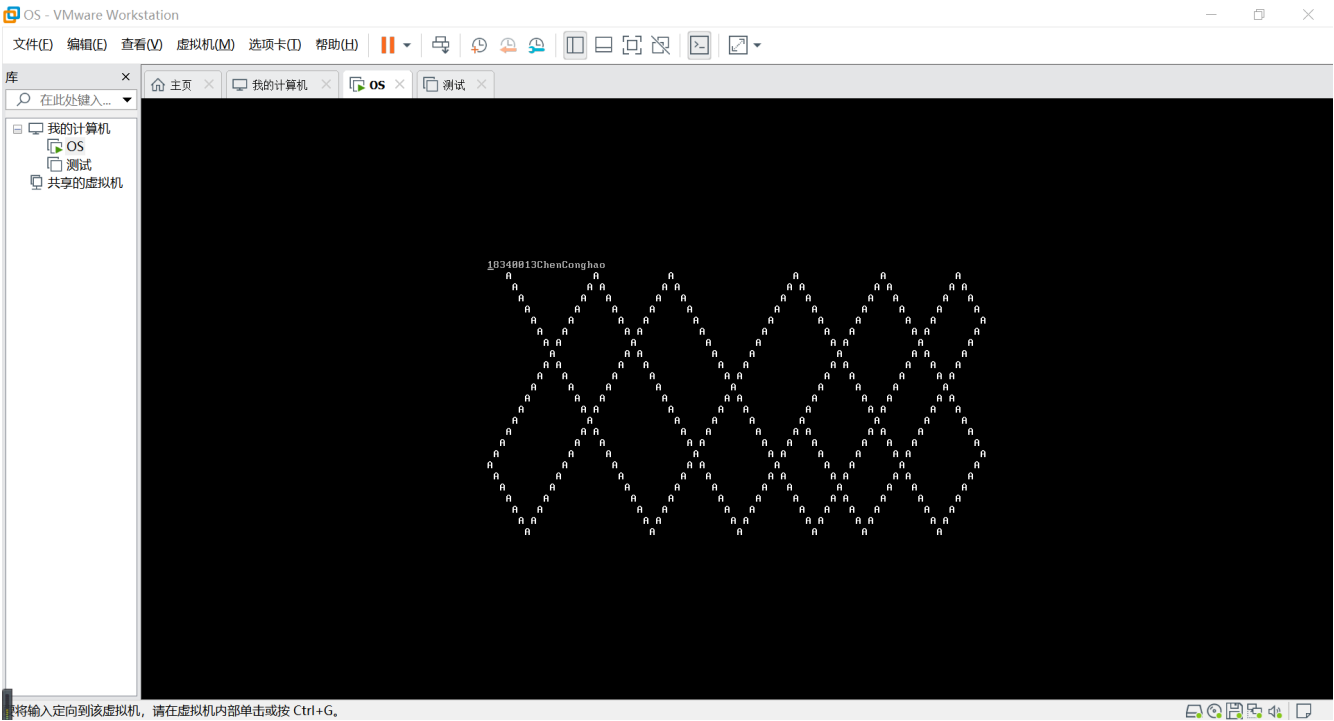


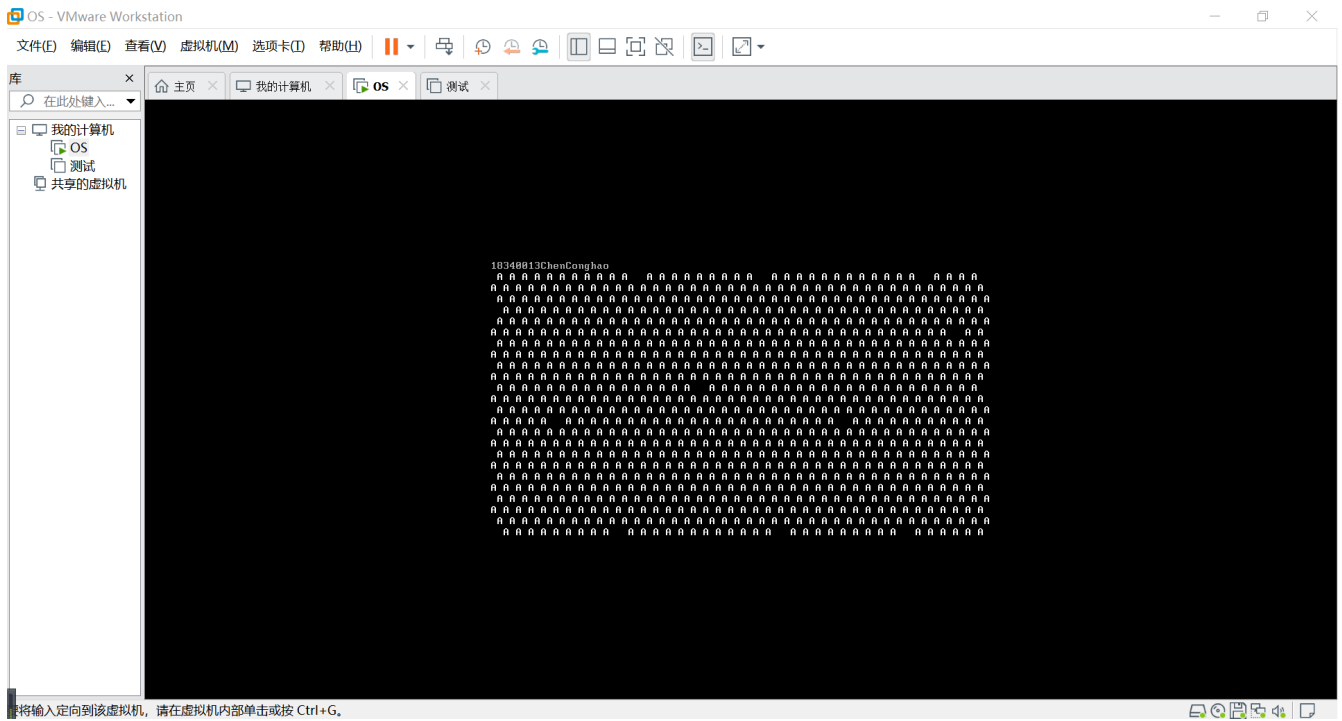
注：代码文件里有两个 .asm 文件和 .img 文件，分别为 stonewhite 和 stonecolor，其中 stonewhite 是基础实验，stonecolor 是复杂实验！将两个 .img 文件按上述进行操作导入虚拟机即可运行。

5.运行显示：

添加软盘完成后，打开虚拟机，从软盘启动后即可看到如下结果：

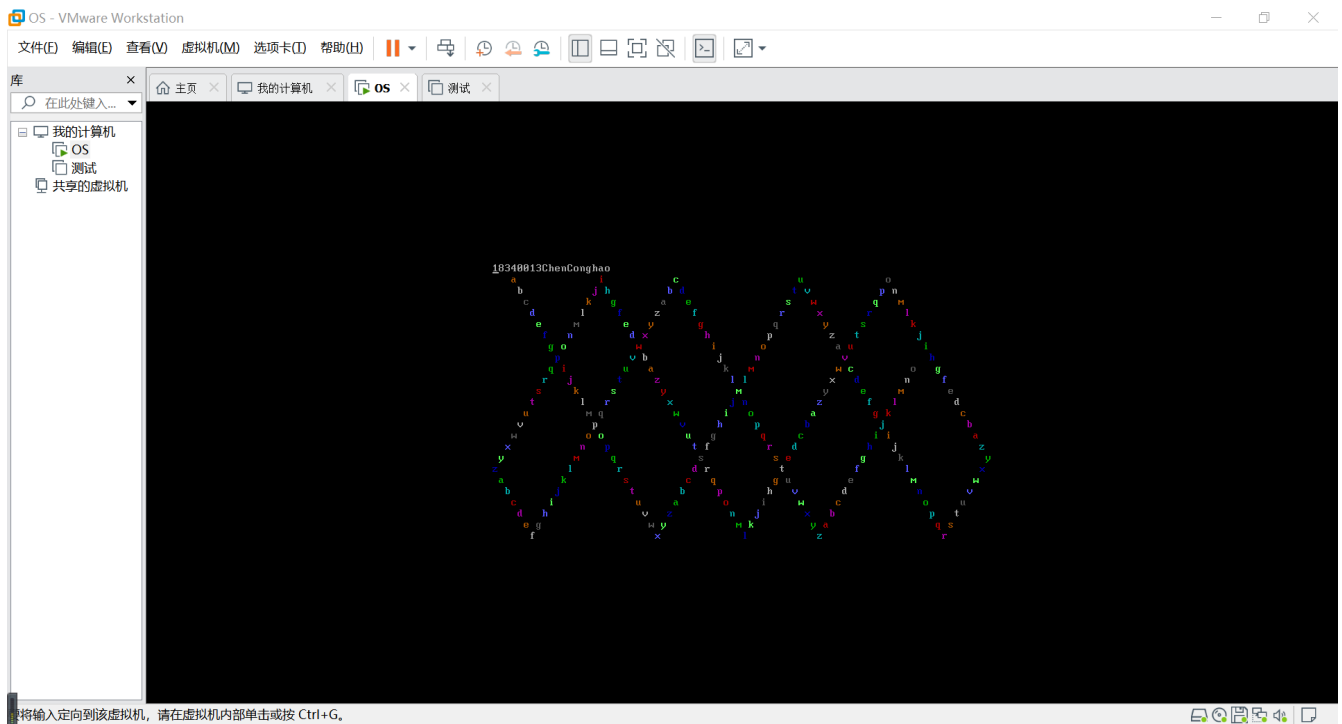
(1) 基础实验：

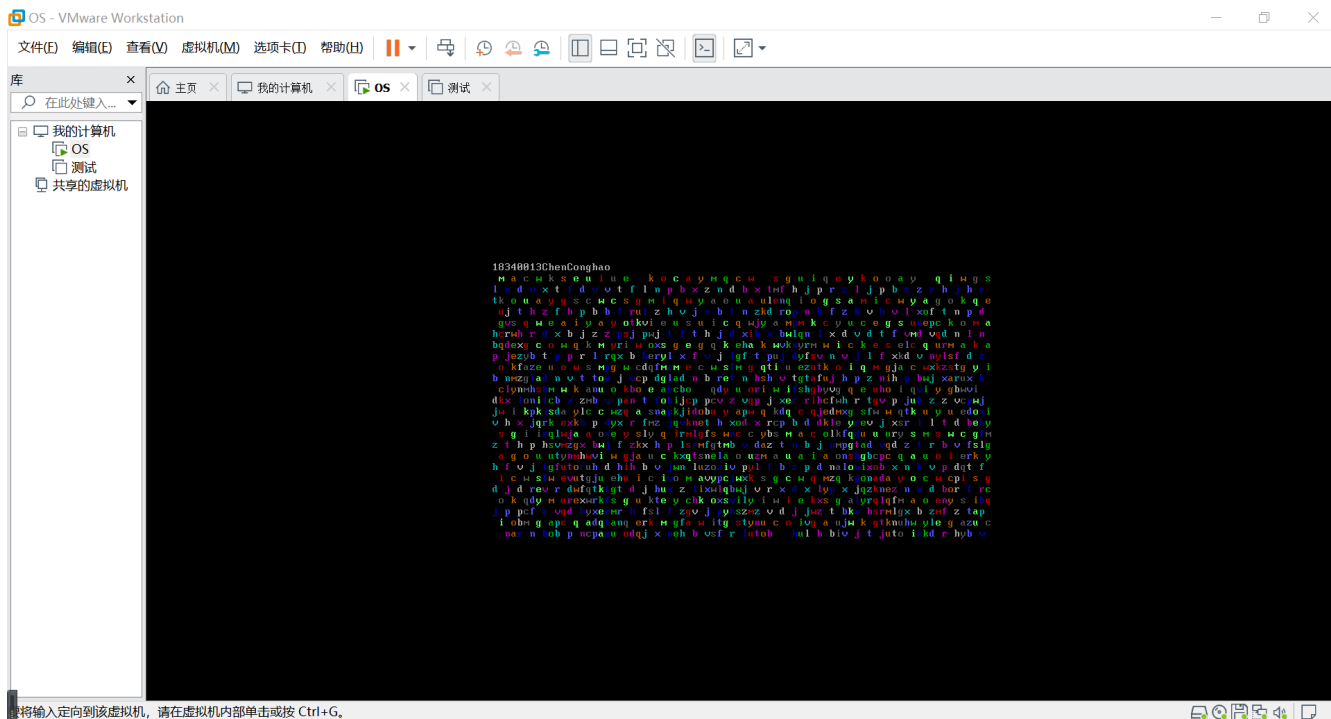




可以看到，成功的显示了学号、姓名以及反弹的黑底白字A。

(2) 复杂实验：





可以看到，成功的显示了学号、姓名，而且还实现了字母a~z变色反弹的功能。

五、实验总结：

在本次实验中，首先就是要配置好所需要的环境，如虚拟机，汇编程序所需的工具。其下载安装过程并不会出现什么问题，而且 VMware、NASM 下载起来也很快，用不了多久就可以安装并使用了。第一步完成后，接下来就是开始进行实验操作了。说实话，在这一瞬间我有一点点的害怕，怕自己把程序跑崩了怎么办。不过看到之前老师在微信群发的其他同学的 demo，想想就觉得能实现出这样的效果一定会很刺激。于是就开始了实验。先是阅读老师给的 stone.asm 汇编程序，大部分还是能看懂的，因为之前学过 MIPS 汇编程序，二者有相近的地方；不懂的指令就要去查一下这条指令是干什么的。这样一来对我后面添加新的功能起了帮助。把代码部分搞定以后，接下来就是在虚拟机上运行。一开始不知道 NASM 怎么用，而且不知道怎么让这个程序在虚拟机上跑，后来参考了老师给我们推荐的这本书：《x86 汇编语言：从实模式到保护模式》，这本书很详细的介绍了 NASM 的使用，让我很快的搞定了 bin、img 文件的生成；随后又上网查阅了相关的资料，明白了如何让虚拟机从软盘启动（根据参考文献中所给网址）。在添加软盘的过程中，我注意到一点：只有虚拟机在关机的情况下才能够添加软盘。我在设置好虚拟机以后就直接添加软盘，发现虚拟机设置一栏里不能进行操作，后来才注意到当时虚拟机处在待机状态，把它关闭以后就可添加软盘了。都调试好以后打开虚拟机，就看到了结果，看到结果成功显示的那一刻内心还是有点小激动的。因为通过自己的不断调试、配置，实现了自己的一个操作系统（尽管它看上去很简单），觉得之前的灰心、害怕等等都是值得的。在实验过程中，和同学相互交流碰到的问题，互相帮助，这个过程也是很开心、很有意义的，让我学到了不少的东西。

六、参考文献：

李忠，王晓波，余洁。《x86 汇编语言：从实模式到保护模式》。电子工业出版社，2012。

<https://blog.csdn.net/m47838704/article/details/46545895>