操作系统实验报告

实验一：接管裸机的控制权

学院：　　数据科学与计算机学院

班级：　　16级计算机科学与技术 教务2班

姓名：　　郑映雪

学号：　　16337327

实验时间：　　2018.3.9~2018.3.11

一、实验目的

学会搭建实验环境并掌握裸机的控制权。

二、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，将其中一个虚拟软盘用DOS格式化为DOS引导盘，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、接管裸机的控制权

设计IBM_PC的一个引导扇区程序，程序功能是：用字符‘A’从屏幕左边某行位置45度角下斜射出，保持一个可观察的适当速度直线运动，碰到屏幕的边后产生反射，改变方向运动，如此类推，不断运动；在此基础上，增加你的个性扩展，如同时控制两个运动的轨迹，或炫酷动态变色，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放进第三张虚拟软盘的首扇区，并用此软盘引导你的XXXPC，直到成功。

三、实验方案

1、硬件或虚拟机配置方法

使用VMware软件构建一个虚拟机，并将这个虚拟机平台的配置调整为4MB的内存，0.1GB硬盘。

2、软件工具与作用

①使用notepad++编写汇编代码，并在控制台中用nasm汇编工具将代码编译成对应的二进制代码，从而使机器可以执行。

②WinHex是一个十六进制编辑器，将生成的.com格式的文件打开可以转化为二进制的形式，从而可以复制进空软盘以进行模拟。同时它也可以通过修改文件中二进制值的方式修改文件。

3、相关原理

本实验中最后一项实验利用了字符显示的原理。字符显示是通过特定字符的代码，利用字符发生器和控制电路来使它们或明或暗以构成字符的轮廓。首先使ES寄存器指向显存段地址B800，然后在后续地址中输入字符的ASCII码和颜色即可显示。

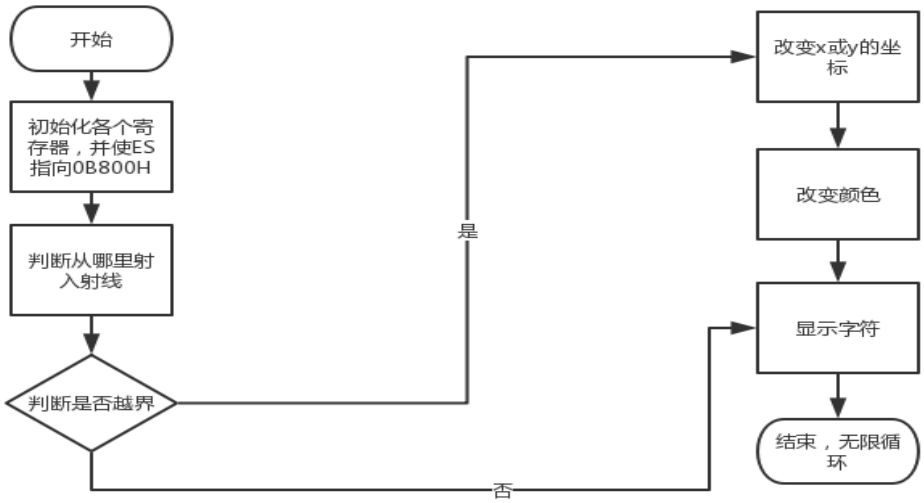
字符变色原理来自于修改显存的奇数单元，以KRGBIRGB 8位二进制数通过不同组合可以显示不同的颜色，具体如下：

R	G	B	背景色	前景色	
			K=0 时不闪烁，K=1 时闪烁	I=0	I=1
0	0	0	黑	黑	灰
0	0	1	蓝	蓝	浅蓝
0	1	0	绿	绿	浅绿
0	1	1	青	青	浅青
1	0	0	红	红	浅红
1	0	1	品（洋）红	品（洋）红	浅品（洋）红
1	1	0	棕	棕	黄
1	1	1	白	白	亮白

在特定位置显示字符的原理是将字符的ascii码送到特定的显存中。具体操作是 mov byte [es:(行数*80+列数)X2]，字符。

4、程序流程

下面是“A”射线在屏幕上持续反射的代码流程图：



5、程序关键模块

在此程序中我在老师的基础上将A改成了*号，并加了碰到边界反射时改变颜

色的功能。主要代码及其说明如下：

```
;xor ax,ax                    ; AX = 0     程序加载到 0000：100h 才能正确执行
mov ax,cs
mov es,ax                     ; ES = 0
mov ds,ax                     ; DS = CS
mov es,ax                     ; ES = CS
mov ax,0B800h                 ; 文本窗口显存起始地址
mov gs,ax                     ; GS = B800h
mov byte[char],'*'
mov byte[co],0FAh             ;颜色初始化

loop1:
    dec word[count]           ; 递减计数变量
    jnz loop1                 ; >0：跳转;
    mov word[count],delay
    dec word[dcount]          ; 递减计数变量
    jnz loop1
    mov word[count],delay
    mov word[dcount],ddelay
;实际上是把 delay 赋值给 count 里的地址，ddelay 赋值给 dcount 的地址，两层循环，起到延时的作用。

;下段是为了判断字符初始时应该从哪个方向射入
mov al,1
cmp al,byte[rdul]
jz  DnRt
mov al,2
cmp al,byte[rdul]
jz  UpRt
mov al,3
cmp al,byte[rdul]
jz  UpLt
mov al,4
cmp al,byte[rdul]
jz  DnLt
jmp $

DnRt:;从左下往右上运动，没有碰到两边就输出，否则反射
    inc word[x]
    inc word[y]
    mov bx,word[x]
    mov ax,25
    sub ax,bx
```

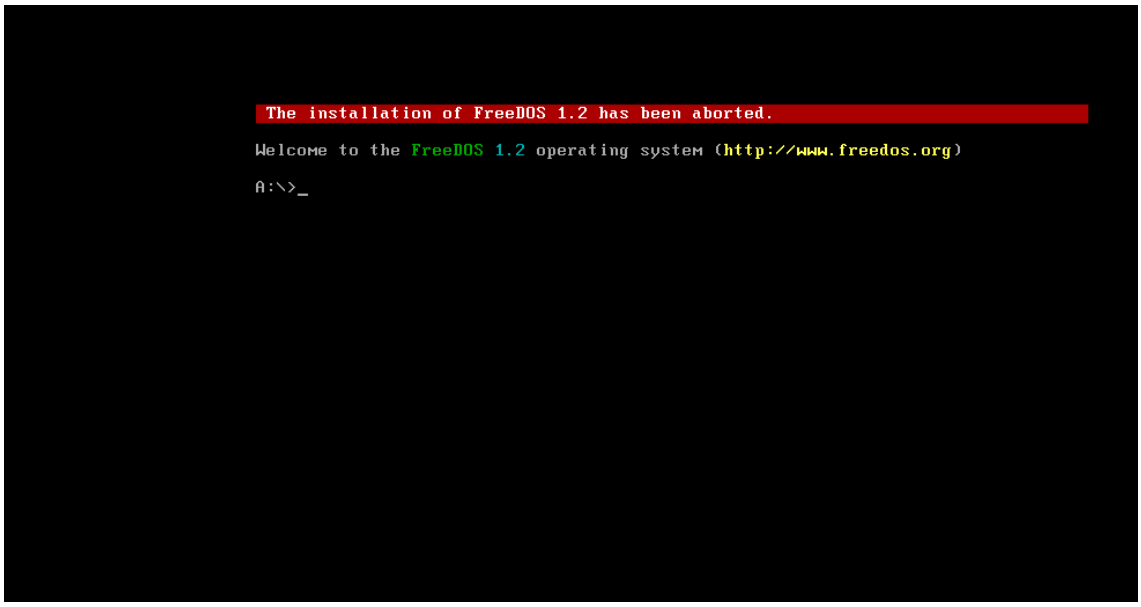
```
        jz    dr2ur
        mov bx,word[y]
        mov ax,80
        sub ax,bx
        jz    dr2dl
        jmp show
dr2ur:;碰到右边界反射
            mov byte[co],09Ah;碰到边界后改变颜色
            mov word[x],23
            mov byte[rdul],Up_Rt
            jmp show
dr2dl:;碰到上边界反射
            mov byte[co],0FAh ;碰到边界后改变颜色
            mov word[y],78
            mov byte[rdul],Dn_Lt
            jmp show

……
后面的程序类似上面的部分，故不粘贴上来了。
```

四、实验过程

1、DOS盘

下载DOS镜像并可以在虚拟机上运行。（见附件）

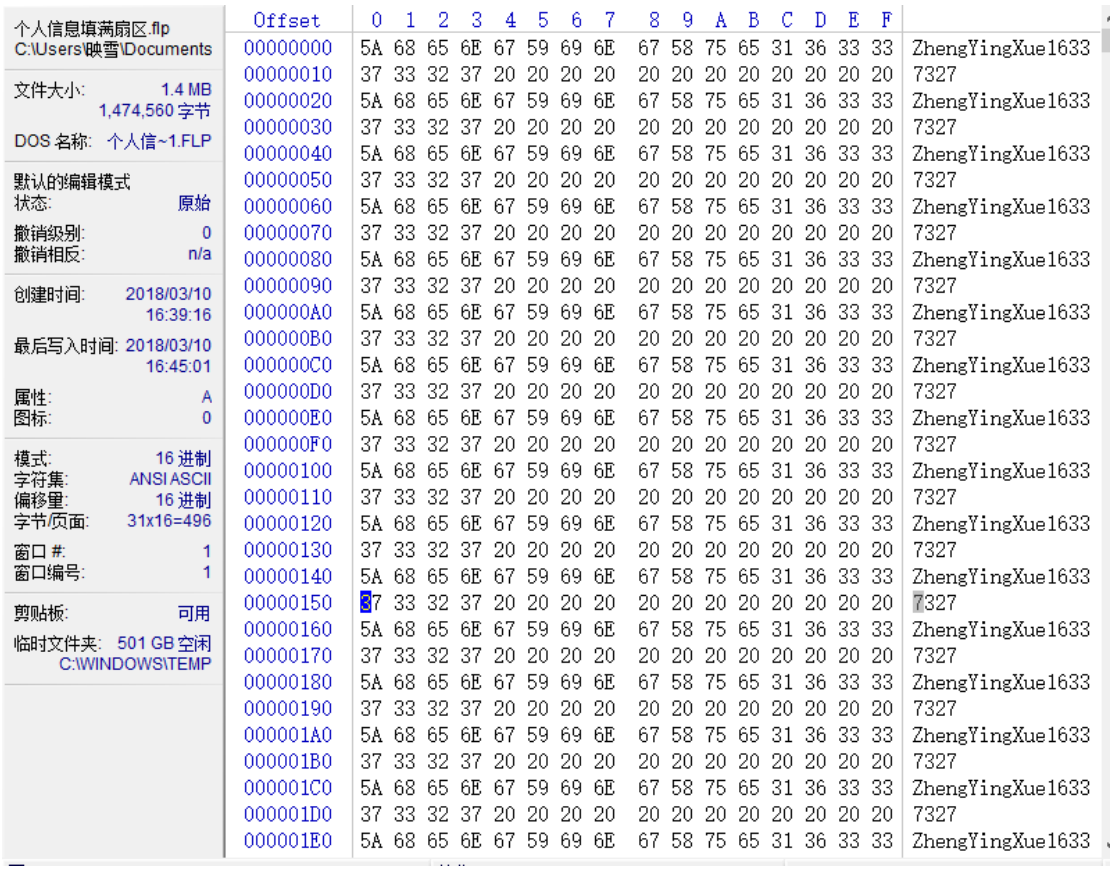


2、将个人信息填满首扇区

新建一个空软盘映像，使用WinHex将自己的信息对应的ASCII码输入到首

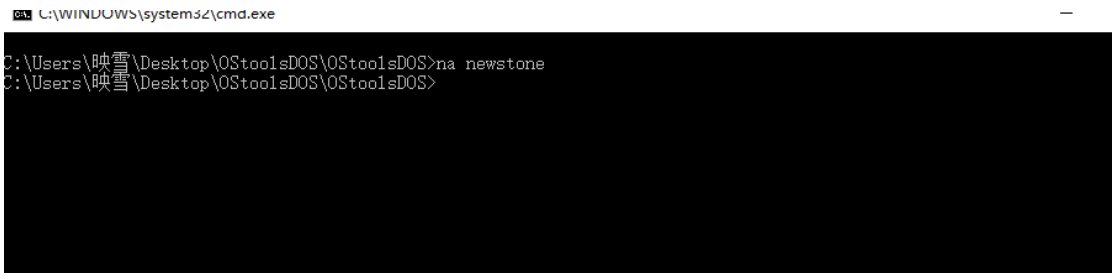
扇区，并将其填满。

填满效果：

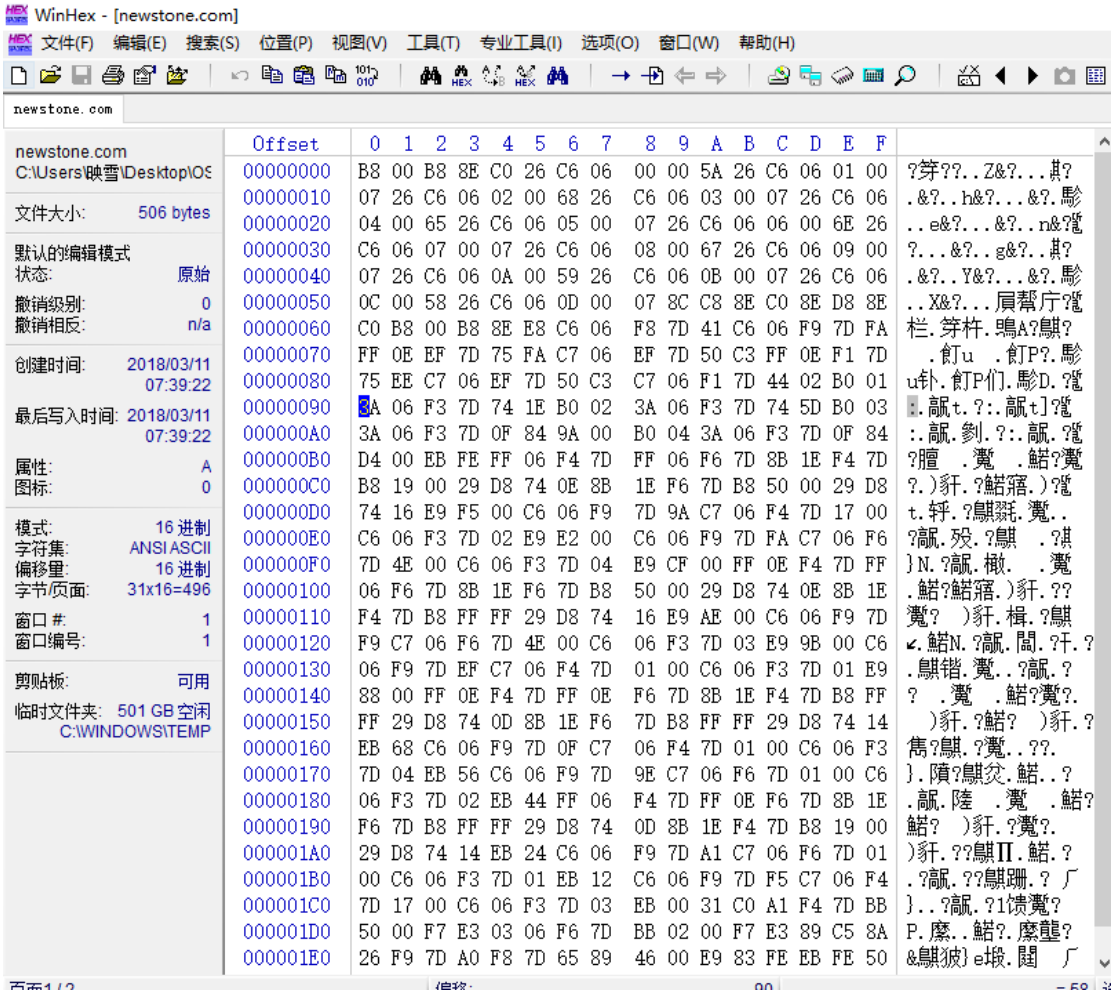


3、理解并个性化修改代码stone.asm

①修改stone代码。原代码使用了masm的伪指令，并不能用nasm编译，所以我将原代码修改成nasm的伪指令，编译通过。然后我加上了个性化的设置，在开头显示我的名字缩写“ZhengYX”，并添上了每次反射时改变颜色的设置。（代码文件见附件）在cmd里使用nasm编译，编译成功。



②用WinHex打开刚才生成的.com文件，可以看到程序已经被译成了十六进制代码。

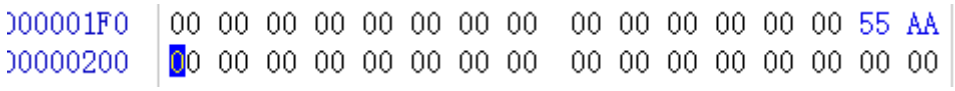


③在建立好的裸机虚拟机上添加一个空软盘映像，如图所示。

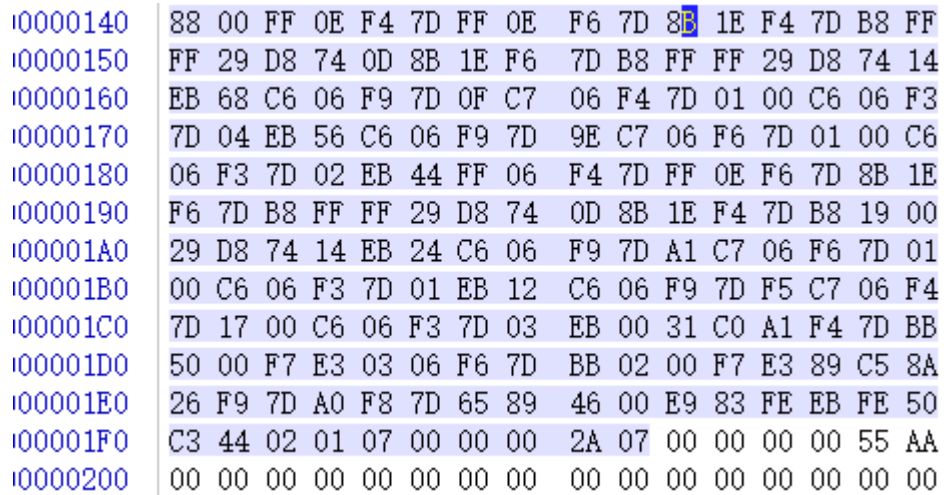
④用WinHex打开这个空软盘映像，可以看到空软盘转化为二进制文件是

全0的。

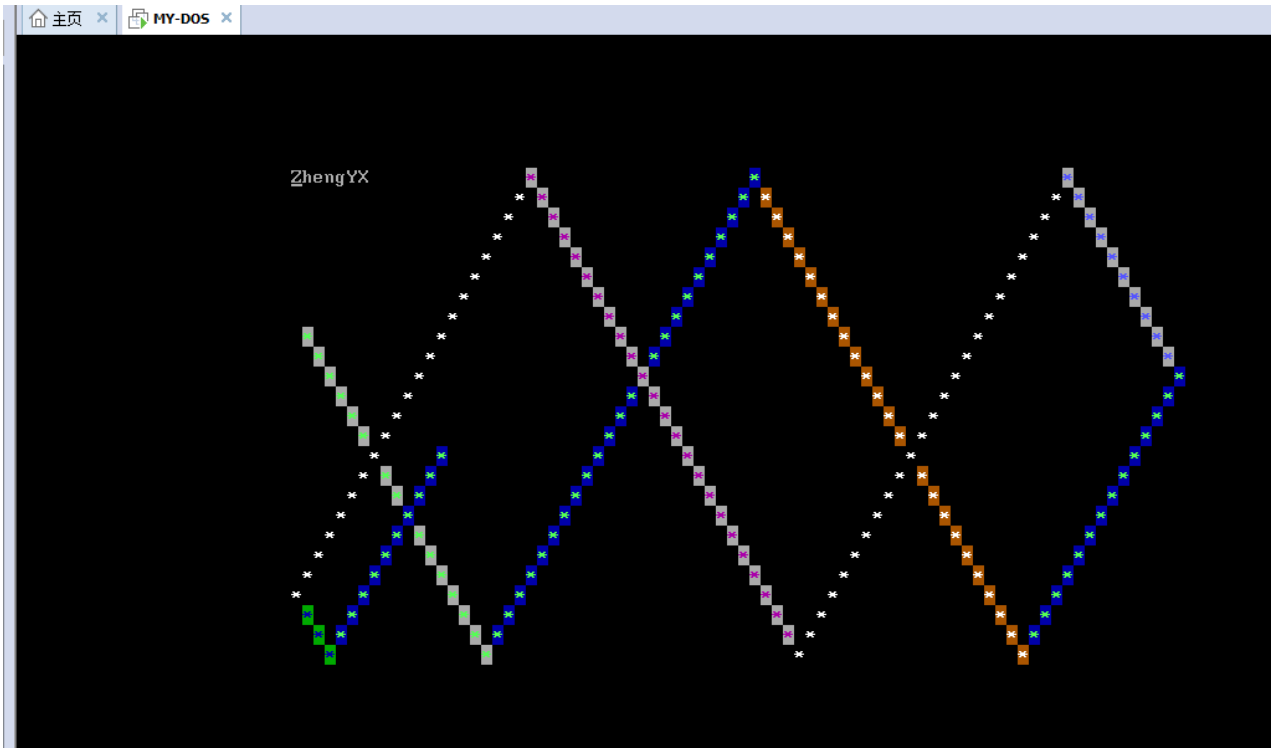
⑤在这个空映像的1F0尾处填上 55 AA。



⑥在空软盘前面预留与代码的二进制文件相同的空间，并将其复制上去。



⑦保存此映像文件，同时打开虚拟机，连接此软盘，显示如下界面。



五、实验总结

这是我的第一个操作系统实验，虽然这只是一个简单的入门级实验，但它是花了我很多很多时间和精力的。老师上课讲的很多东西我都是第一次见，当时就“一脸懵逼”，回去疯狂查资料，但由于没有系统地了解相关的东西，我觉得我现在还有很多需要学习。不过，让我高兴的是，通过这个实验，我学到了不少东西。

首先，我看到代码时，我就觉得跟我们上学期学的汇编语言不太一样，很多地方我都看不大懂。原来代码里使用了大量的nasm的语法，我们上学期在masm里使用的PTR在nasm里就不需要打……如此种种有许多。我上网查找了masm和nasm语法的不同，把代码里的assume等指令删除了，终于可以用na成功编译。

在编辑二进制文件时，我不理解老师上课输入的 55 AA有什么作用，上网查阅后得知，处理器会将主引导扇区的数据加载到逻辑地址 0x0000:0x7c00 中，然后检测最后两字节是否为 0x55 和 0xAA，若存在则主引导扇区有效。所以我们在复制二进制文件进空盘时，一定要最后标上55 AA。但是我不知道如果写的

代码编译后超过512字节了怎么办。（这个问题我到现在也没弄明白，也许以后课堂上有机会可以让我弄懂）

我认真理解了一下老师给的代码，由显示字符的原理开始理解。我明白了显示器如何去显示一个字符，为什么这个字符可以在特定的地方显示。我对照老师的PPT在显示器上输出了一行字符。

在成功运行老师的代码后，我想加上变颜色的功能。通过学习老师的PPT，我明白了只要往字符地址之后的后八位二进制数进行修改即可改变背景色或者前景色。于是我增加了一个控制颜色的变量，在每一次碰到边界的跳转处理段代码都加上了改变颜色变量的设置。这时候出现了一个问题——我的射线开始从左下角跑出来了！我再三检查了初始射线方向的代码，发现rdul这个变量我并没有改变，那是为什么呢？我跑去询问同样使用颜色改变特效的同学，发现他们也有这个问题，但是也不知道为什么。我只好一行行仔细分析，把自己的脑子看成显示器，终于发现——原来控制颜色的变量所控制的地址单元没有初始化！正因为没有初始化，所以它一开始从左上方射入并没有显示出来，直到碰到下边界反弹时地址单元才会有赋值，从而显示出来。我在初始化的模块加上了对颜色控制变量的初始化，果然恢复了从左上角射入的图像。我把这个原因告诉了同样出现这个问题的同学，他们的问题也得到了解决，嘻嘻。

可是，用眼睛去debug实在不是一个程序猿解决问题的高效方法。我看见老师推荐的参考书中有说汇编程序的调试可以用Boch进行，下次程序出现问题我想试着学习用Boch去调试，这样也许可以节省我大量的时间。

虽然自己的水平有限，并不能像许多大佬一样做出非常炫酷的改变，但是能做到现在这样我已经很开心了！希望在以后的学习中能学到更多的东西，做出更多的东西。最后想给老师提一个小建议，希望老师可以讲得再详细一些，同时在说明实验要求的时候也进行一下解释。希望能在老师的教导下得到更大的进步啦！

六、参考文献

1、nasm与masm语法区别 - jiu~ - 博客园

https://www.cnblogs.com/jiu0821/p/4422464.html

2、《操作系统原理实验课件实验 1》 - 凌应标

3、学习 nasm 语言 - 咕咕gu - 博客园

http://www.cnblogs.com/gugugu/archive/2012/12/30/6318074.html