实验二

实 验 报 告

学院 数据科学与计算机学院

专业 计算机类

年级 18 级

姓名 黄思蓉

学号 18340064

课程名称 操作系统原理实验

目录

1 实验题目

加载用户程序

2 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

3 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

4 实验内容

1. 将自己实验一的引导扇区程序修改为一个的 COM 格式程序，程序缩小显示区域，在屏幕第一个 1/4 区域显示，显示一些信息后，程序会结束退出，可以在 DOS 中运行。在 1.44MB 软驱映像中制定一个或多个扇区，存储这个用户程序 a。相似地、将自己实验一的引导扇区程序修改为第二、第三、第四个的 COM 格式程序，程序缩小显示区域，在屏幕第二、第三、第四个 1/4 区域显示，在 1.44MB 软驱映像中制定一个或多个扇区，存储用户程序 b、用户程序 c、用户程序 d。

2. 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。程序可以按操作选择，执行一个或几个用户程序。解决加载用户程序和返回监控程序的问题，执行完一个用户程序后，可以执行下一个。

3. 设计一种命令，可以在一个命令中指定某种顺序执行若干个用户程序。可以反复接受命令。

4. 在映像盘上，设计一个表格，记录盘上有几个用户程序，放在那个位置等等信息，如果可以，让监控程序显示出表格信息。

5. 拓展自己的软件项目管理目录，管理实验项目相关文档

5 实验方案

1. 安装虚拟机 VirtualBox

2. 获取可视化编辑十六进制文件内容的工具

3. 安装 nasm 编译器，编译 asm 文件

4. 学习 x86，编辑引导程序代码

6 实验过程

6.1 安装虚拟机

登陆官网 `https://www.virtualbox.org/wiki/Downloads`，如下图 1 点击 Windows hosts 即可

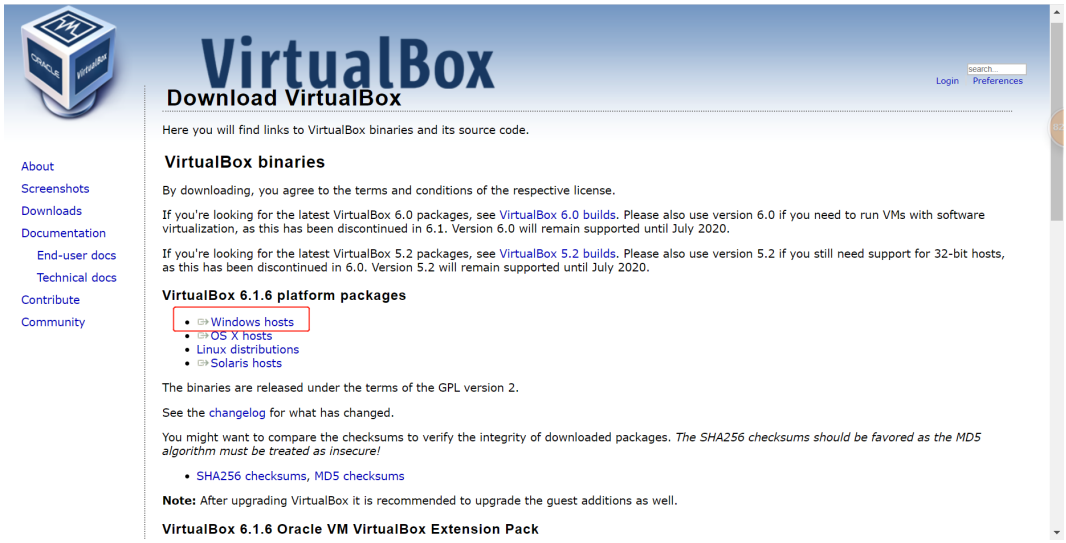

图 1: VirtualBox 下载

接着是安装过程，直接点击下载好的 VirtualBox-6.1.4-136177-Win.exe 文件，如图 2（写此报告时已经安装好虚拟机了）。接着只用一直点击下一步即可。安装过程是参照这篇博客 `https://blog.csdn.net/qq_33690342/article/details/81412167`，安装过程很顺利，在这里不赘述。

图 2: VirtualBox 安装

图 3: VirtualBox 安装过程

6.2 获取可视化编辑十六进制文件内容的工具

登陆 winhex 官网下载即可 `https://winhex.en.softonic.com/`，另外也可以获取李忠老师的《x86 汇编语言-从实模式到保护模式》这本书的配套工具 HexView, 不过该工具只能查看不能编辑保存。

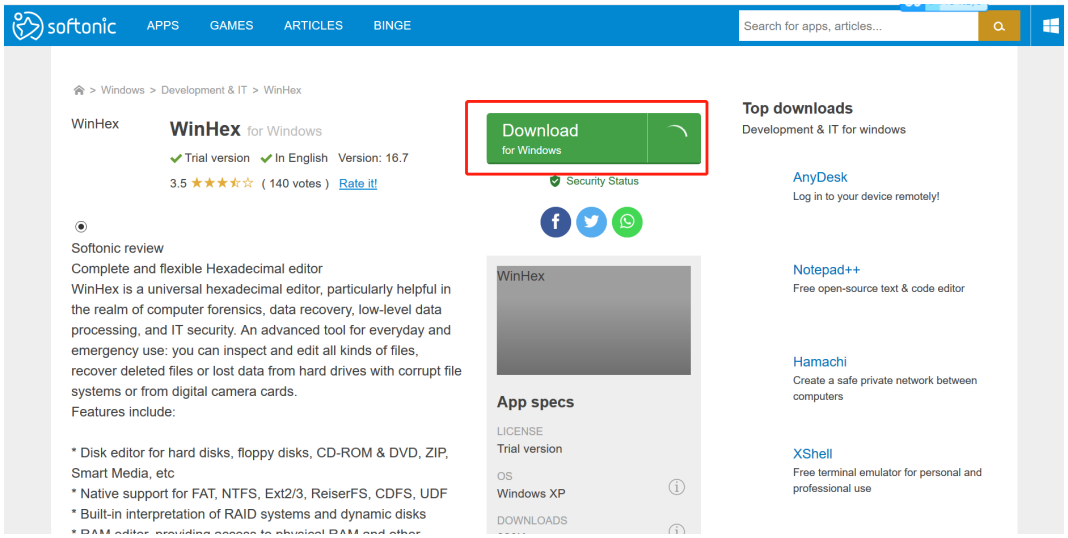

图 4: WinHex 下载

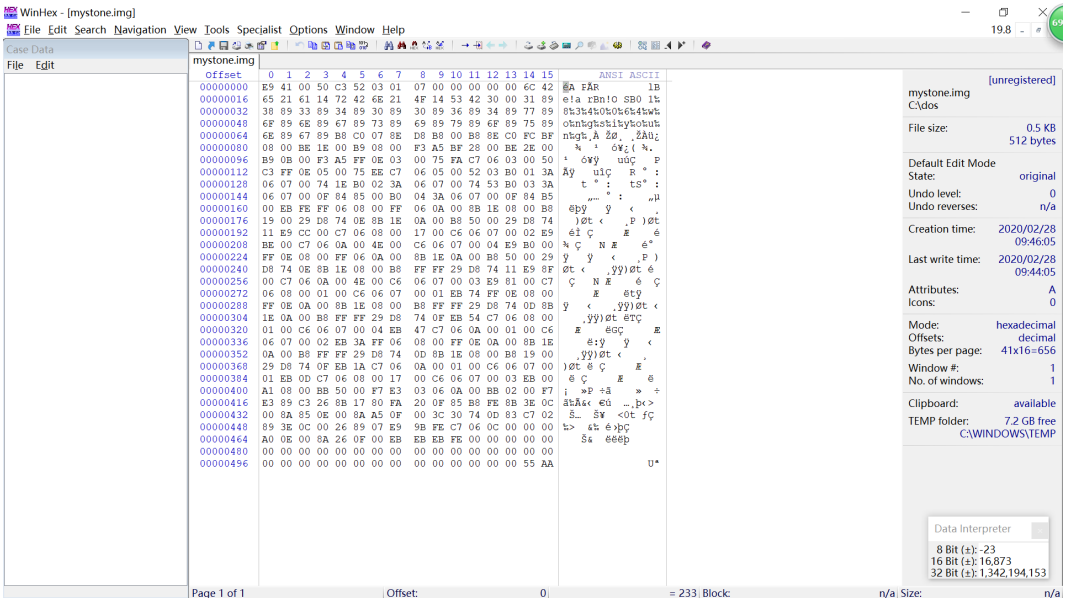

图 5: WinHex 使用示意

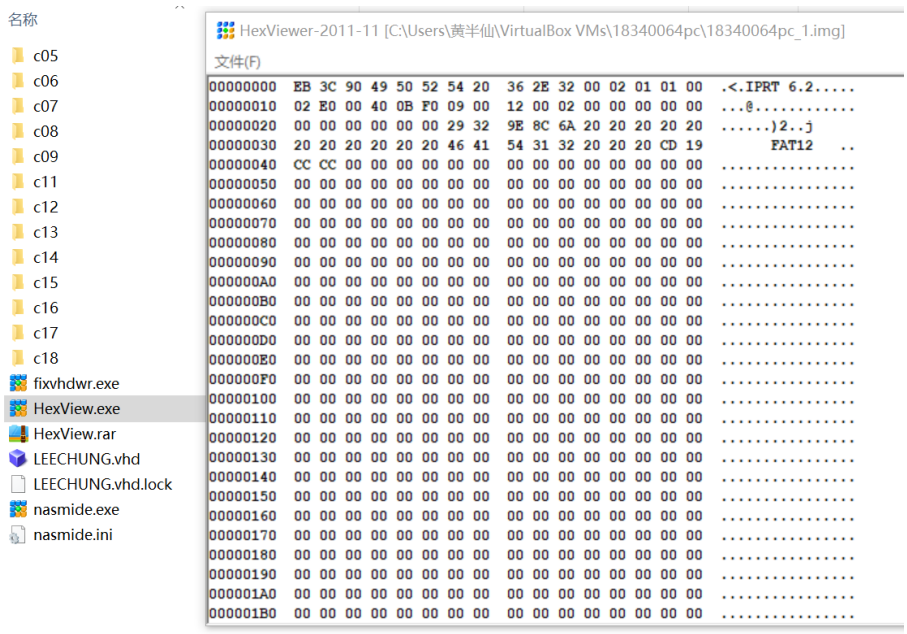

图 6: HexView 使用示意

6.3 安装 nasm

同样地登陆 nasm 官网 `https://www.nasm.us/` , 点击 DOWNLOAD，选择合适的版本下载，我选择的是 nasm-2.11.02-installer

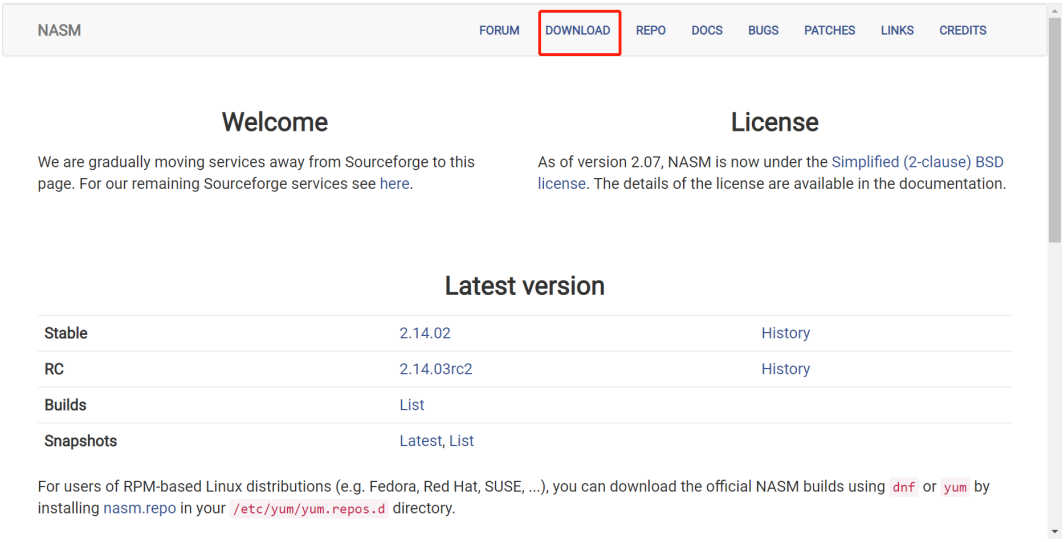

图 7: NASM 下载

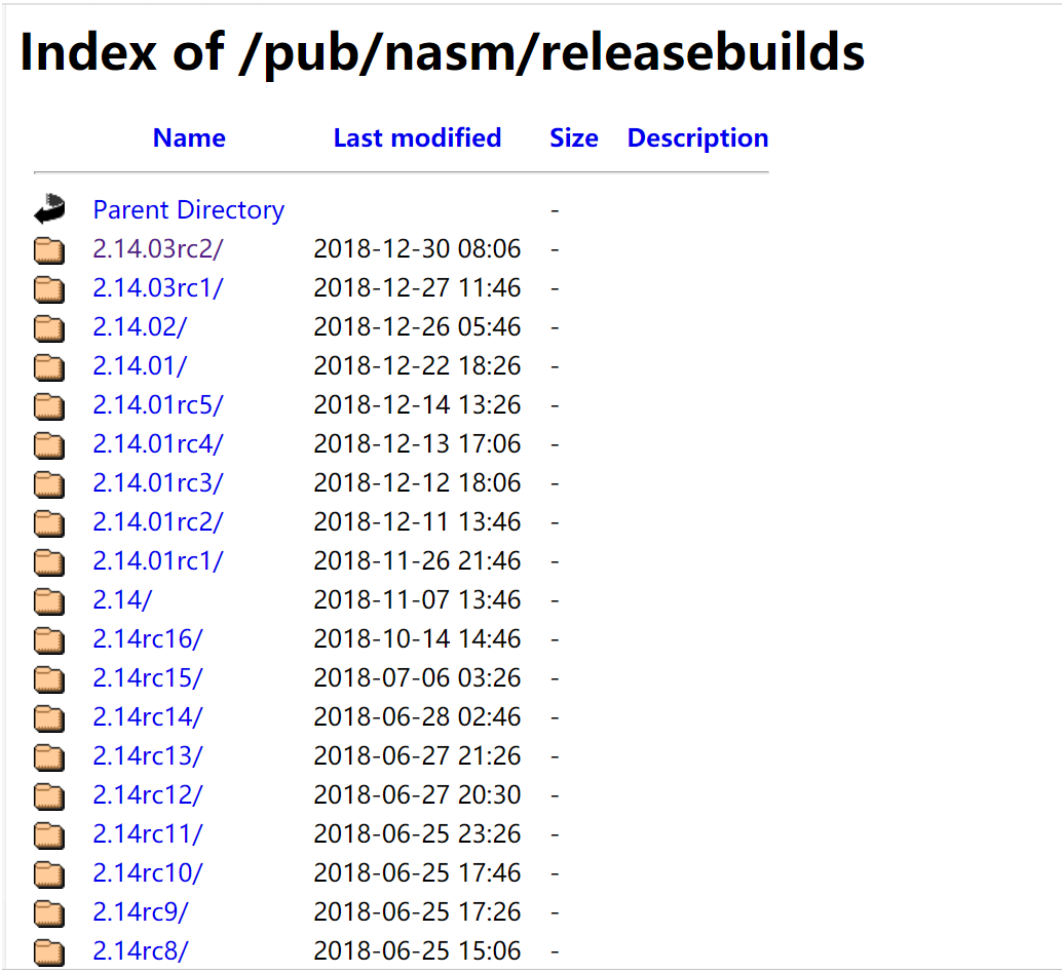

图 8: NASM 下载

图 9: NASM 下载

下面是安装过程，双击从官网下载的 exe 文件，也是只用点击 next，直到出现 install 按钮即可，其中可以改变文件安装路径。

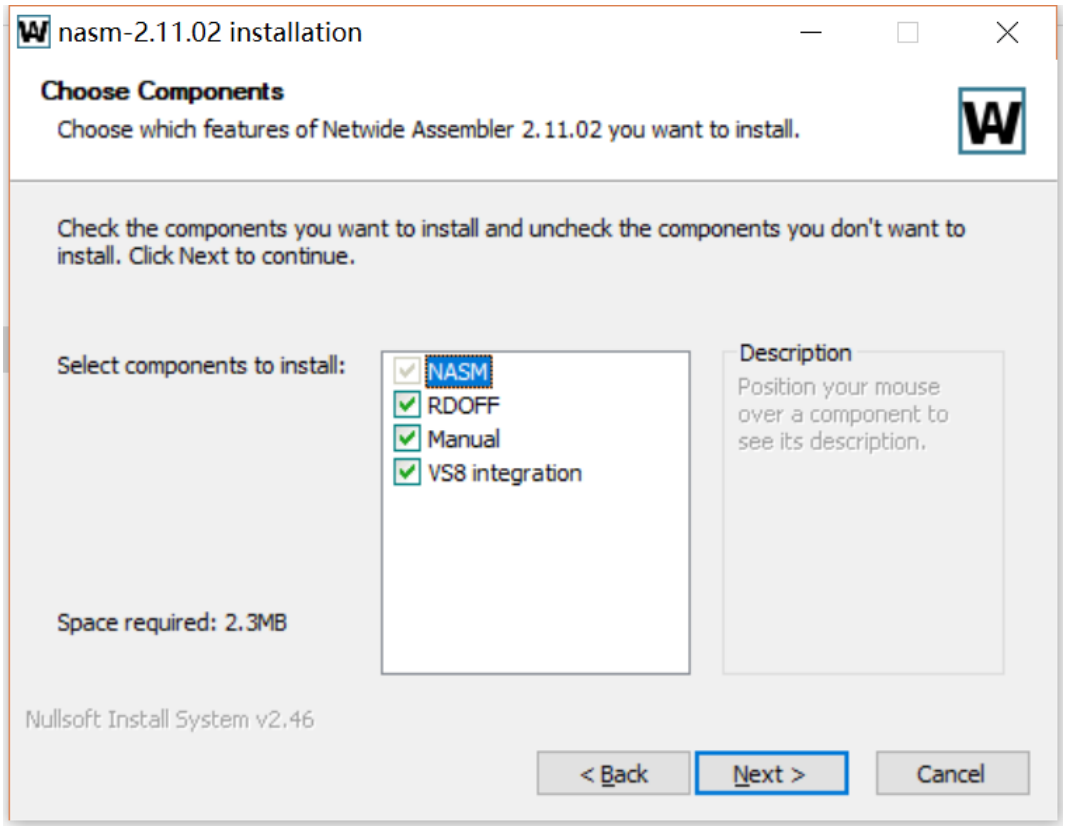

图 10: NASM 安装

安装成功如图：

图 11: NASM 安装完毕

为了方便使用，配置环境变量：

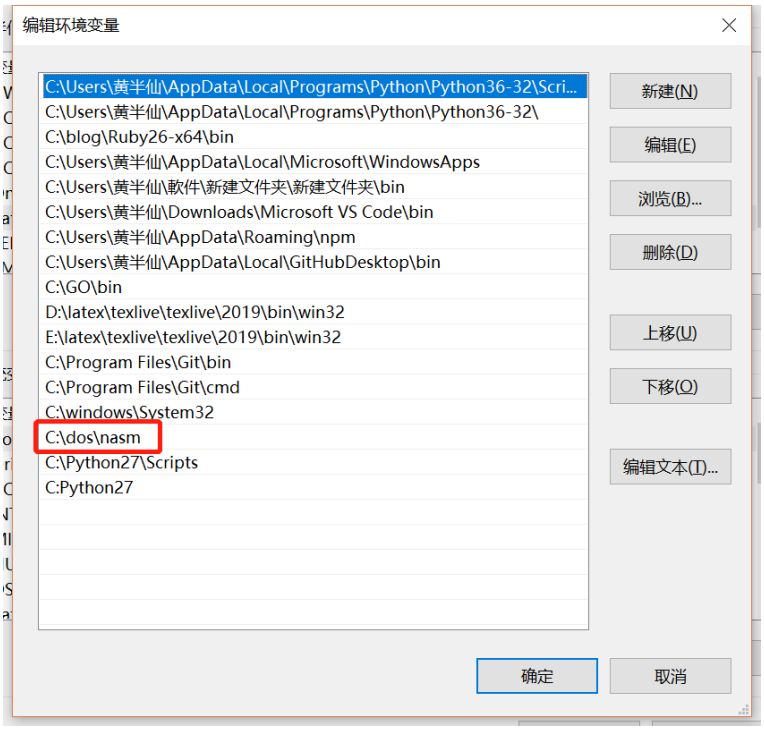

图 12: 配置环境变量

在命令窗口输入 nasm -v 命令检验

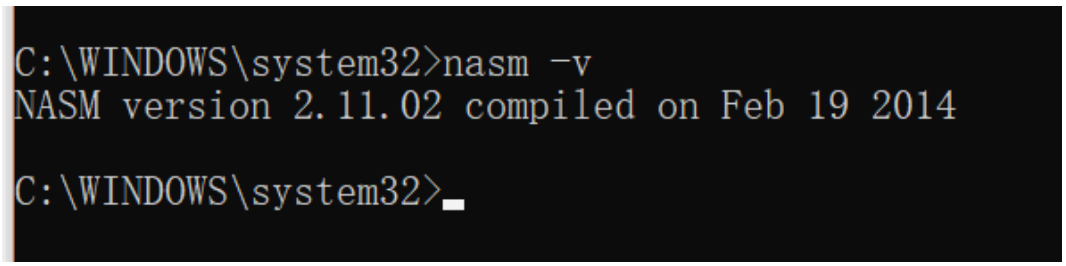

图 13: nasm -v 命令

6.4 编辑引导程序代码

老师提供的代码其实已经实现了以字符‘A’作运动轨迹, 但是编译失败，只用稍微改一下下面的代码就可以了！其中 ds 指向地址 0x7c00，因为 bios 执行完后，cpu 从这里开始执行指令；es 指向 0B800h，因为在内存地址空间中，B800h~BFFFFh 共 32KB 的空间，为 80x25 彩色字符模式的显示缓冲区。

```
;       .386
        org 7c00h                       ; ³ÌÐò¼ÓÔØµ½100h£¬¿ÉÓÃÓÚÉú
;       ASSUME cs:code,ds:code
;     code SEGMENT
start:
    ;xor ax,ax                          ; AX = 0    ³ÌÐò¼ÓÔØµ½0000£
    mov ax,cs
    mov es,ax                           ; ES = 0
    mov ds,ax                           ; DS = CS
    mov es,ax                           ; ES = CS
    mov ax,0B800h                       ; ÎÄ±¾´°¿ÚÔ´æÆðÊ¼µØÖ·
    mov gs,ax                           ; GS = B800h
    mov byte[char],'A'
```

图 14: 老师提供的部分代码

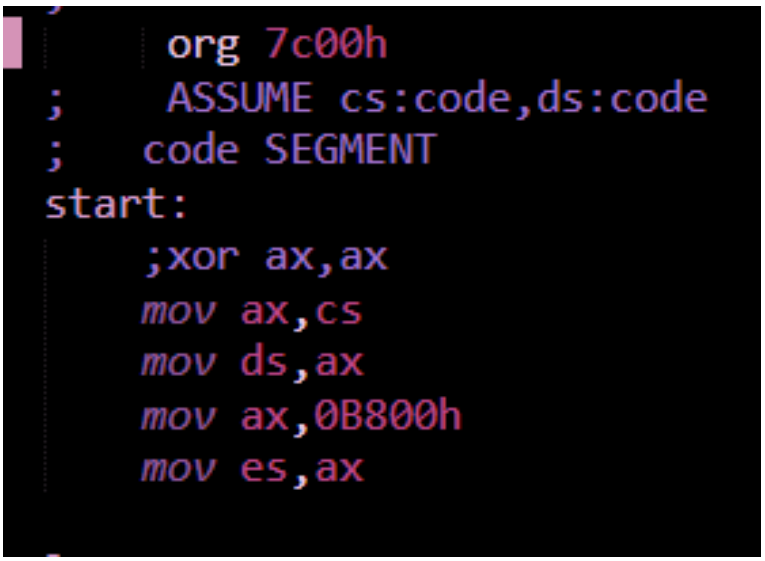

图 15: 一种修改方式

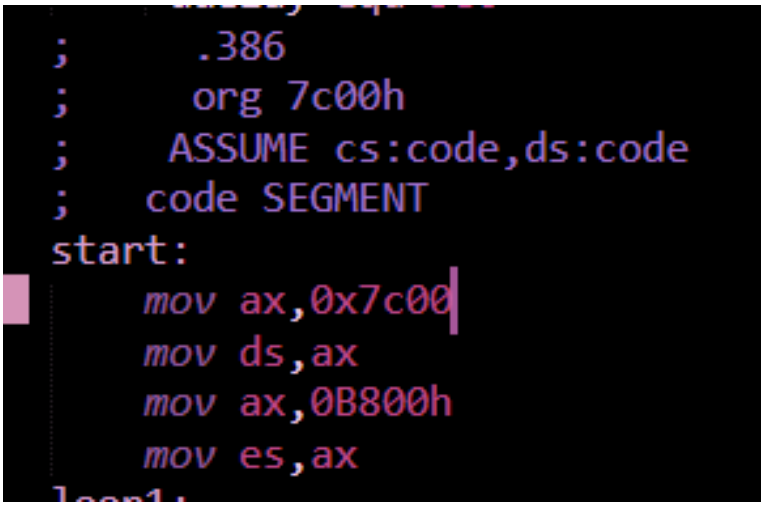

图 16: 另一种修改方式

接着修改数据段，增加个人信息：学号 (18340064), 姓名（huangsirong），根据显示原理：低位字节存储字符的 ASCII 码，高位字节存储字符的属性，属性字节格式：

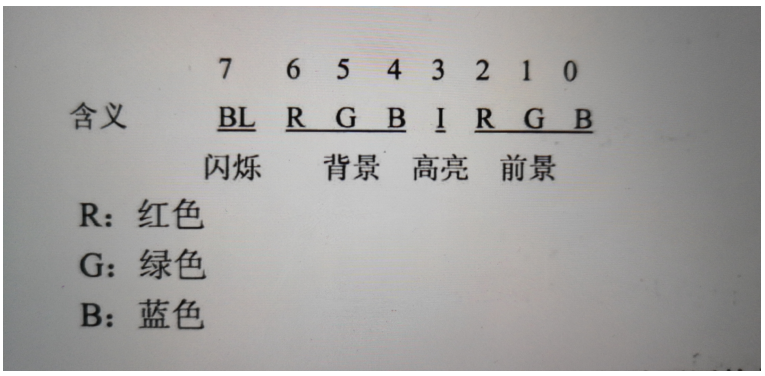

图 17: 字符属性

```
datadef:
    count dw delay                  ;延迟计时
    dcount dw ddelay                ;延迟计时
    rdul db Dn_Rt                           ;运动方向
    x     dw 7              ;行位置
    y     dw 0              ;列位置
    cnt   dw 0              ;计数器
    char db 'l',42h,'e',21h,'a',14h,'r',42h,'n',21h,'O',14h,'S',42h,'0',0
                        ;要显示的字符
    number  db  '1',89h,'8',89h,'3',89h,'4',89h,'0',89h,'0',89h,'6',89h
        ,'4',89h  ;要显示的学号
    name        db 'h',89h,'u',89h,'a',89h,'n',89h,'g',89h,'s',89h,'i',89
        h,'r',89h,'o',89h,'n',89h,'g',89h ;要显示的姓名拼音
```

在开始以蛇形运动轨迹显示"learnos" 之前，显示个人信息：

```
showmeg:
        cld
        mov     di,8                    ;指定显示屏的位置
        mov si,number       ;指向要显示的学号
        mov cx,8            ;8位学号，循环8次
        rep movsw
        mov di,28h                  ;指定显示屏的位置
        mov si,name         ;指向要显示的姓名拼音
        mov cx,11           ;循环11次
        rep movsw
```

为了避免 learnos 在显示过程中覆盖了已显示的姓名学号，修改 show 部分的程序，避开：

```
show:
```

```
    mov ax,word[x]              ;word[x]当前行
        mov bx,80
        mul bx;                             ;word[x]*80
        add ax,word[y]                      ;行+列=当前位置
        mov bx,2
        mul bx                      ;2*(行+列),一个字符显示占两个字节
        mov bx,ax                                   ;计算结果ax送给bx

        mov dx,[es:bx]                      ;获得当前显示屏的字符ascii码
        cmp dl,' '                  ;如果为空，就显示，否则直接进入下一次循
            环，跳过本次显示
        jnz loop1                                   ;避开显示的信息

        mov di,[cnt]                        ;循环显示"learnos"的计数
        mov al,byte[char+di]        ;要显示的字符
        mov ah,byte[char+di+1]      ;设置字符属性

        cmp al,'0'                                  ;"learnos"末尾标志
        jz s1

        add di,2                                    ;指向下一次要显示的字符,
            2个字节
        mov [cnt],di                        ;存储下一次要显示的字符的偏移量

con:
        mov [es:bx],ax                      ;送入显示器
        jmp loop1                                   ;继续蛇形运动

s1:
        mov word[cnt],0                     ;重新循环"learnos"
        mov al,byte[char]                   ;要显示的字符
        mov ah,byte[char+1]                 ;设置字符属性
        jmp con
```

至此，在老师提供的代码上编辑完毕，用 nasm 命令编译该程序，在终端输入命令 nasm xxx.asm -o xxx.img 即可

```
C:\dos\asm>nasm stoneN.asm -o MyStone.img

C:\dos\asm>
```

图 18: 编译

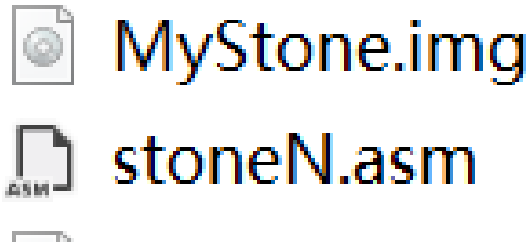

图 19: 编译结果

获得 img 映像文件后，使用 winhex 打开，可以看到有 512 字节且以 0x55aa 结尾，是可引导程序，缘于代码：

```
times 510-($-$$) db 0
 db 0x55,0xaa
```

只要前面的代码量不超多 510 字节就可以。接着用 winhex 写入利用虚拟机生成的 1.44MB 的软盘映像文件中

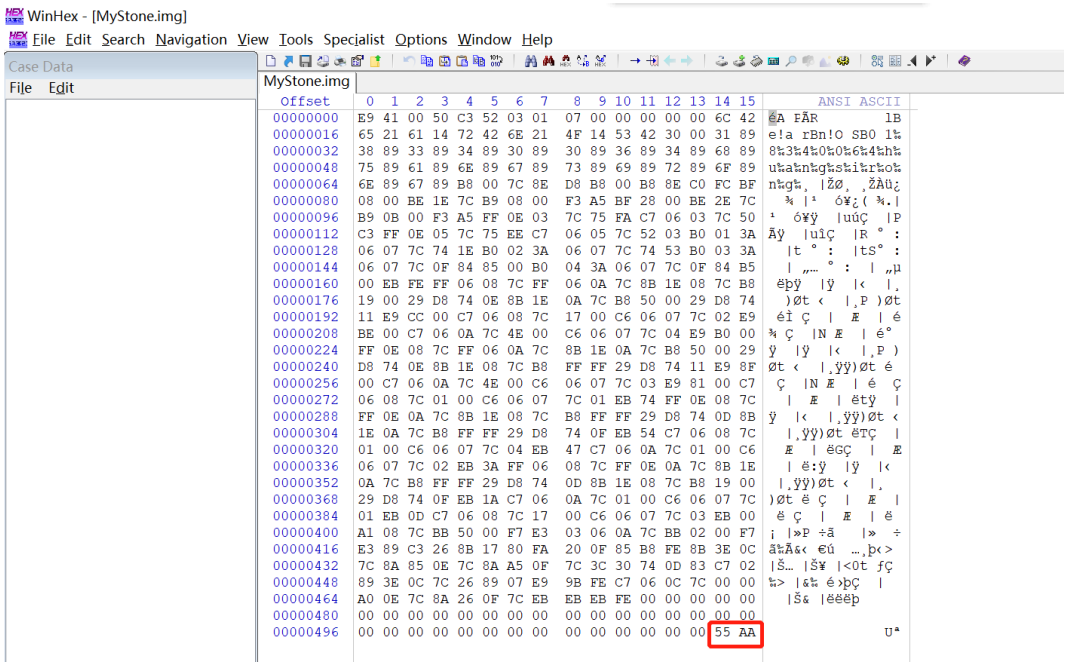

图 20: MyStone

6.5 创建虚拟机，生成 3 个 1.44MB 的软盘映像文件

打开 VirtualBox，点击新建

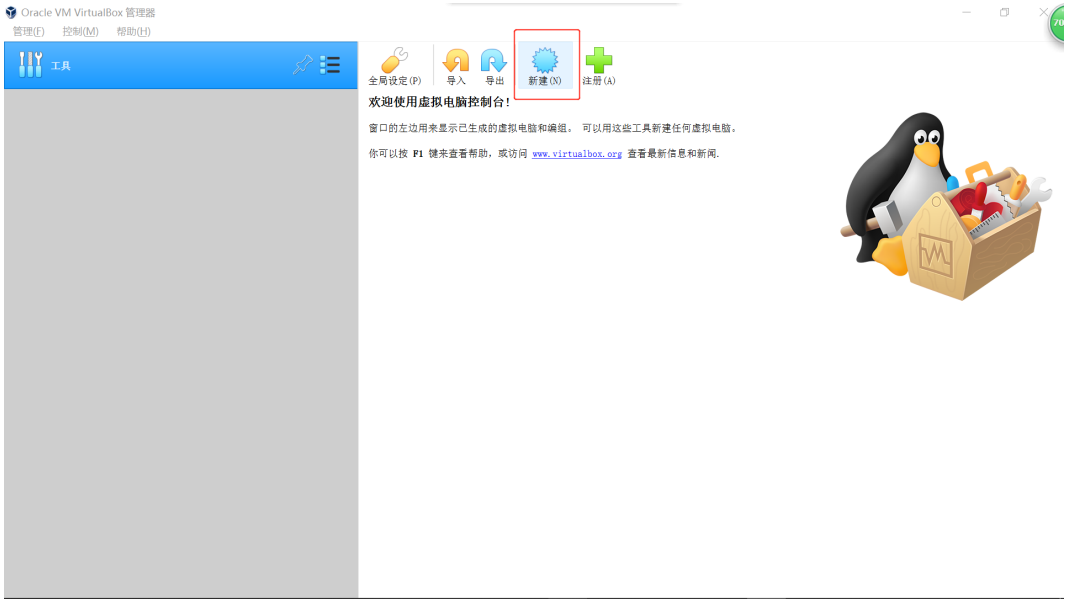

图 21: 新建虚拟机

设置虚拟机信息：

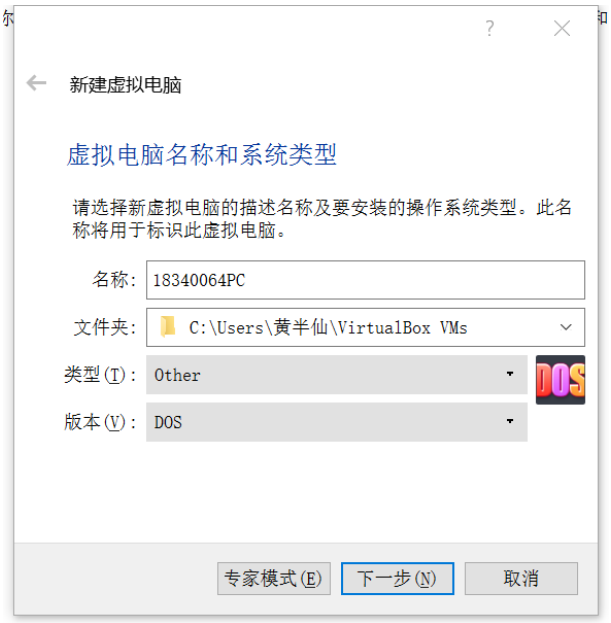

图 22: 设置虚拟机名字，类型

创建软盘映像文件，按照老师的要求，创建三个映像文件分别为 dos 格式化软盘，写自己信息的软盘，写入引导程序的软盘，如下面的图展示 dos 创建过程，其余两个也是一样的步骤，分别命名为：dos.img，Message.img，boot.img

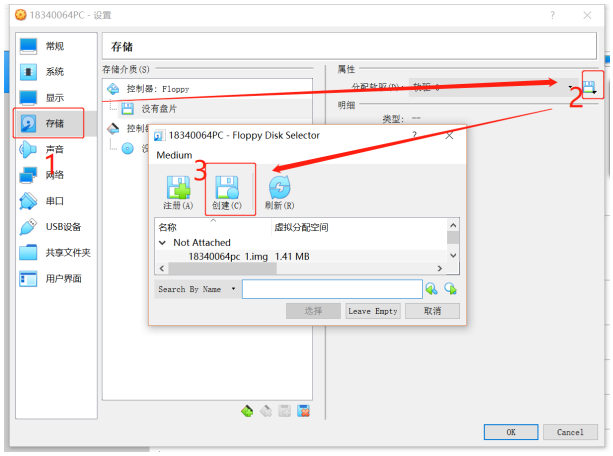

图 23: 创建 dos 映像文件

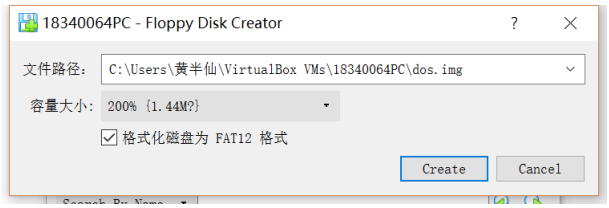

图 24: 创建 dos 映像文件

然后将刚刚生成的 MyStone.img 文件的 512 字节写入 boot.img 的第一个 512 字节中。本来是用前面写的 winhex 来修改的，但是 winhex 显示 boot.img 文件超过 200kb 无法修改，后来直接用 sublime 打开，也可以修改保存

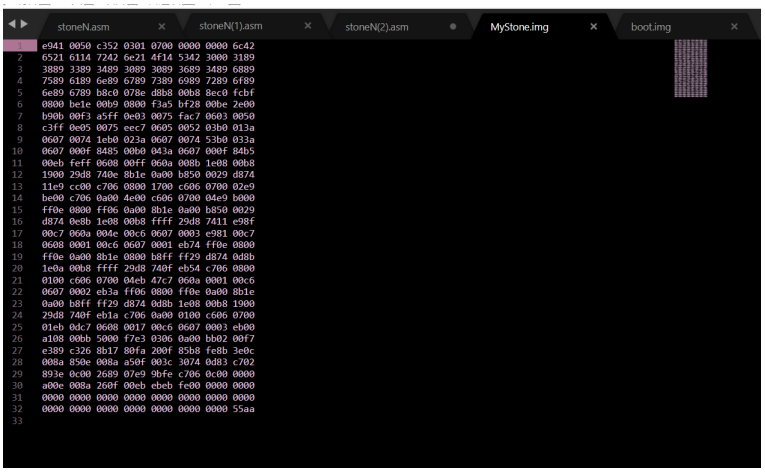

图 25: MyStone 文件

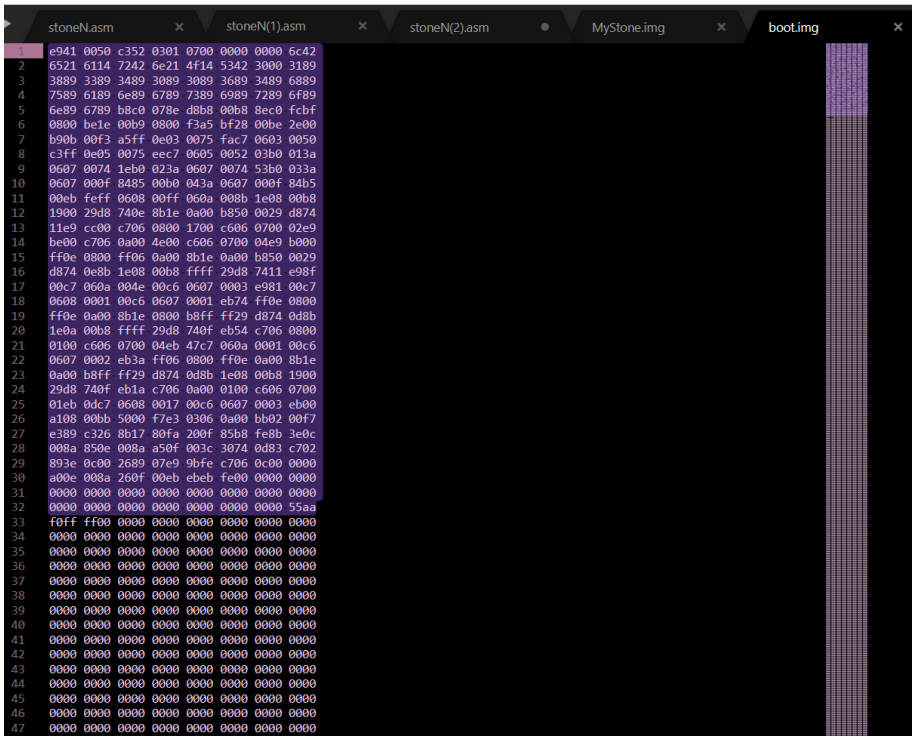

图 26: boot 文件

设置虚拟机软驱为刚刚改动的 boot.img，然后开机启动，结果如下：

图 27: 设置虚拟机软驱

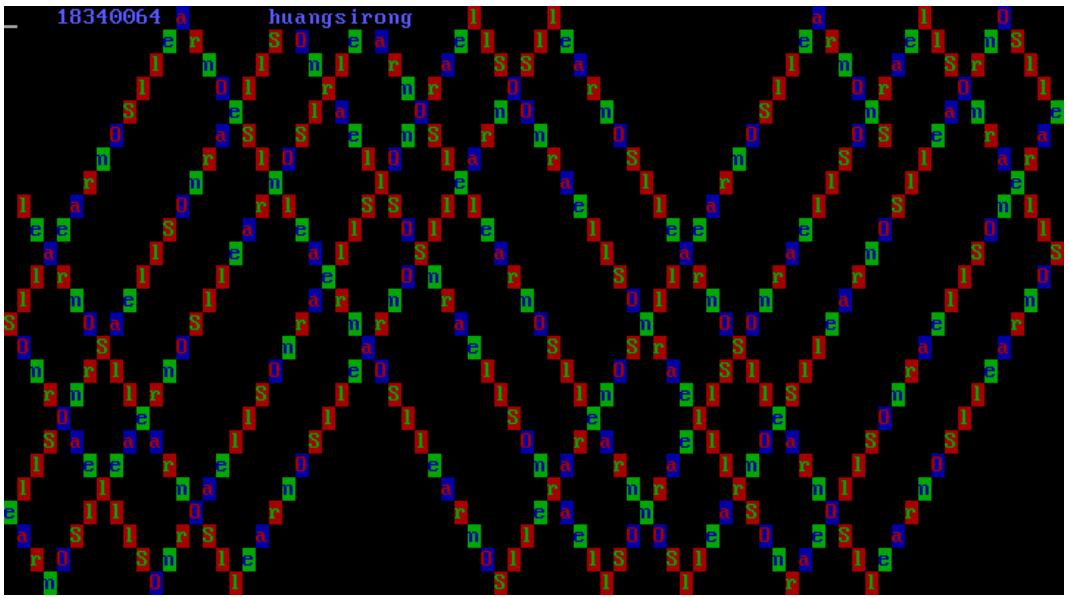

图 28: 启动画面

6.6 Message.img 写入个人信息

使用汇编代码生成个人信息的机器码，然后写入 Message.img

```
datadef:
    number   db   '1',89h,'8',89h,'3',89h,'4',89h,'0',89h,'0',89h,'6',89h
        ,'4',89h    ;学号
    name          db 'h',89h,'u',89h,'a',89h,'n',89h,'g',89h,'s',89h,'i',89
        h,'r',89h,'o',89h,'n',89h,'g',89h ;姓名
```

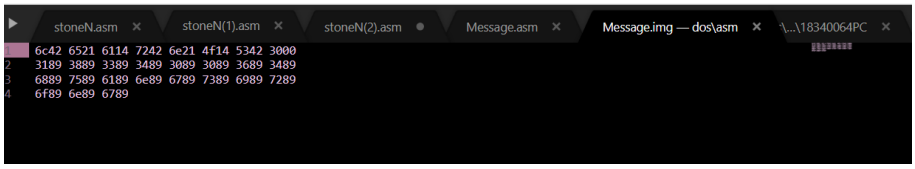

图 29: 汇编生成的机器码

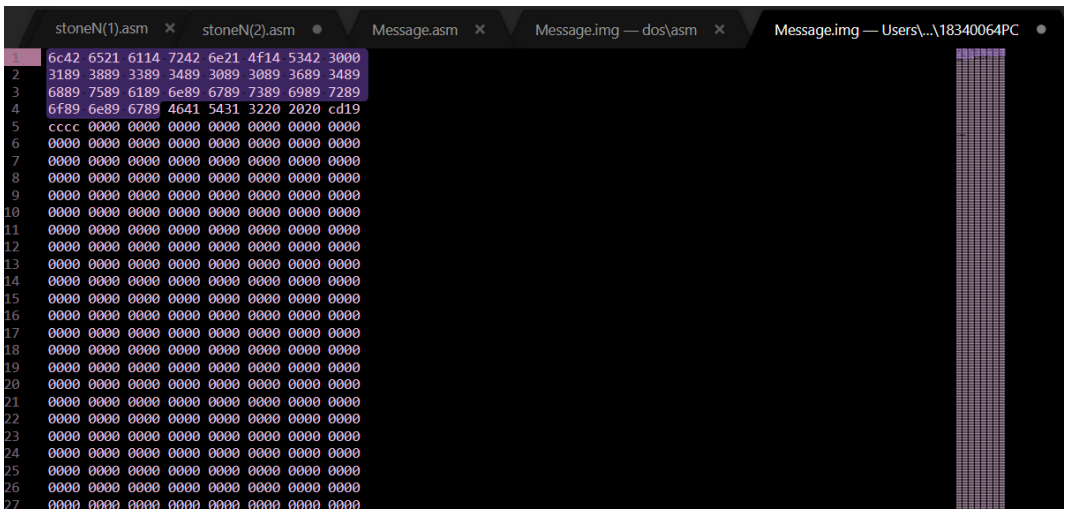

图 30: 写入 Message.img

7 实验总结

实验成功之后，把过程写成实验报告，回看实验报告的流程看似很简单，但其中有很多曲折。

一开始对于实验内容其实是很模糊的，主要也是因为缺乏汇编知识。在学习计算机组成原理的时候接触的是 mips 指令，对于 x86 是不了解的，所以一开始在看到老师的代码的时候，一来看不懂，二来直接对老师提供的代码编译会报错。

所以一开始是先看了王爽的《汇编语言（第 3 版）》才对汇编语言有所了解，接着也看了李忠的《x86 汇编语言-从实模式到保护模式》，再回看老师提供的代码就看懂了。

不过因为王爽的汇编语言书是用 masm 的，而李忠的汇编语言书用的 nasm，一开始并不知道两者的区别，后来实践的时候就知道了……所以因为先看的王爽的汇编书，而用 masm 只能编译成.obj 和.exe 文件，无法直接在虚拟机中配置使用，在这里折腾了挺久的，后来使用 nasm，可以很方便的编译成 img 或 bin 文件就解决问题了。在完成了添加个人信息和字符的一些个性变化后，尝试过写如同时控制两个运动的轨迹，写了之后感觉逻辑上是可以的，但是代码量太长了，超过了 512 字节，还在想着改进办法……

以及在老师修改了实验要求和内容之后，不是很明白为什么要搞三个盘，明明一个盘装引导程序的，后来想想可能是为后面的实验做准备……

最后总的来说，当启动虚拟机后，看到画面的运动轨迹还是很有成就感的！另外也要加紧汇编语言的学习！

8 参考资料

1. 王爽著.《汇编语言（第 3 版）》. 清华大学出版社.2003 年 9 月
2. 李忠著. 《x86 汇编语言-从实模式到保护模式》. 电子工业出版社，2013 年 1 月
3. 虚拟机 VirtualBox 安装教程. https://blog.csdn.net/qq_33690342/article/details/81412167