

实验一

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	2
2	实验目的	2
3	实验要求	2
4	实验内容	2
5	实验方案	3
6	实验过程	3
6.1	安装虚拟机	3
6.2	获取可视化编辑十六进制文件内容的工具	5
6.3	安装 nasm	6
6.4	编辑引导程序代码	9
6.5	创建虚拟机，生成 3 个 1.44MB 的软盘映像文件	14
6.6	Message.img 写入个人信息	18
7	实验总结	18

1 实验题目

编写一个引导扇区程序

2 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. 学习 PC 字符显示方法、复习加强汇编语言程序设计能力

3 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，如截屏或录屏，以证实实验工作的真实性，按时打包提交实验相关文档

4 实验内容

1. 在自己的电脑上安装一种虚拟机软件，在实验报告中记录主要的安装步骤和截屏
2. 利用虚拟机软件，生成有 1.44MB 软驱的一个 PC 虚拟机，列出 PC 虚拟机的配置，并生成有 1.44MB 软盘映像文件 3 个
3. 安装 winHex 等可视化编辑十六进制文件内容的工具，对第一个软盘映像文件的首扇区填满个人学号姓名拼音

4. 安装一种 x86 汇编程序和一种编辑汇编/C 源程序代码的工具或集成环境
5. 用 x86 汇编语言编写一个程序，参考字符反弹运动示范程，修改或重写程序，直接对文本方式的显存进行操作，以某种运动轨迹或几何图像在屏幕一个区域显示字符或字符串，还可以有各种个性化变化效果，能看到个人学号或姓名拼音
6. 程序汇编后满足引导扇区程序的要求，利用工具将其制作写入 1.44mb 软盘映像的引导扇区中，保证在虚拟机中能引导执行，观察到效果
7. 建立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

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。

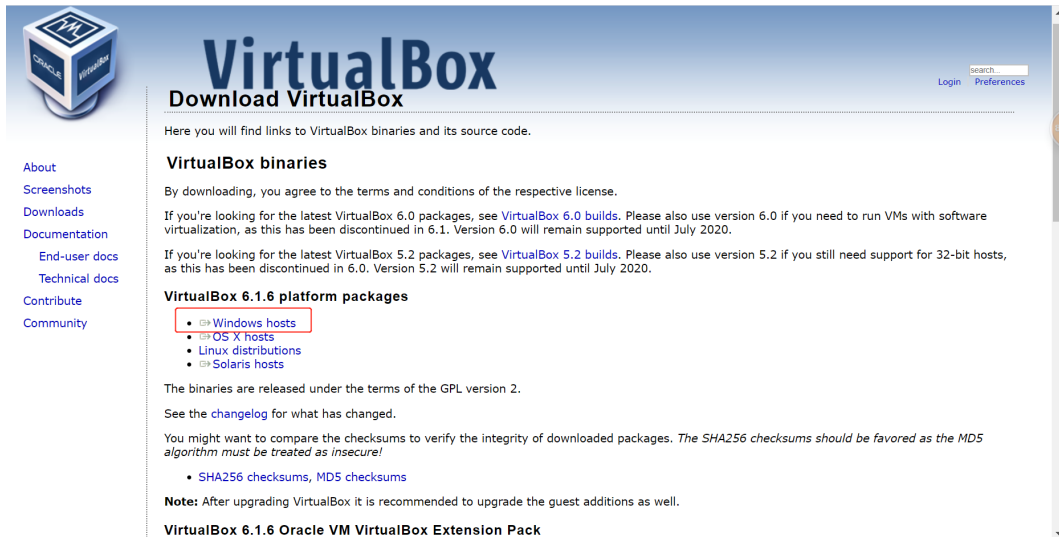


图 1: VirtualBox 下载



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。

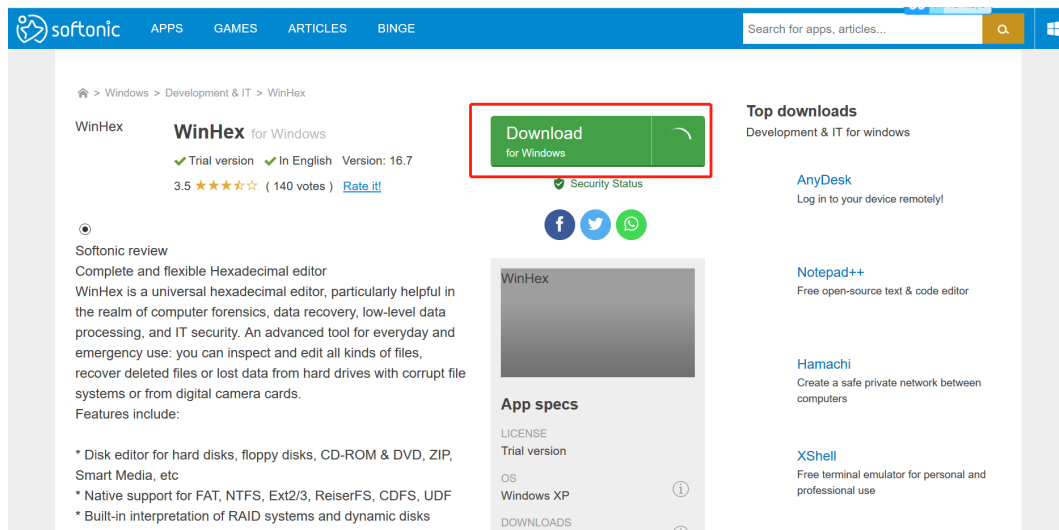


图 4: WinHex 安装

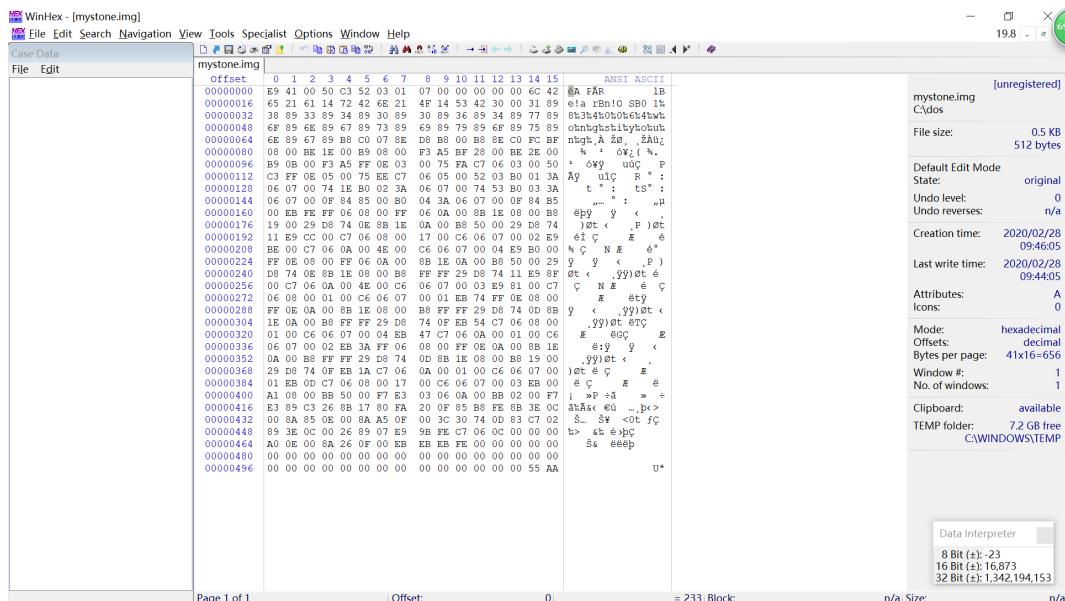


图 5: WinHex 使用示意

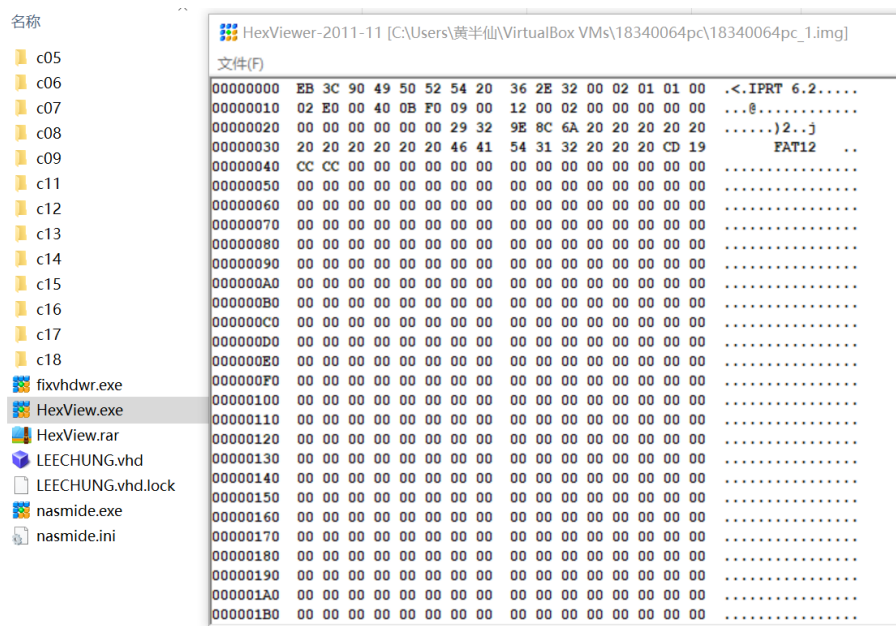


图 6: HexView 使用示意

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

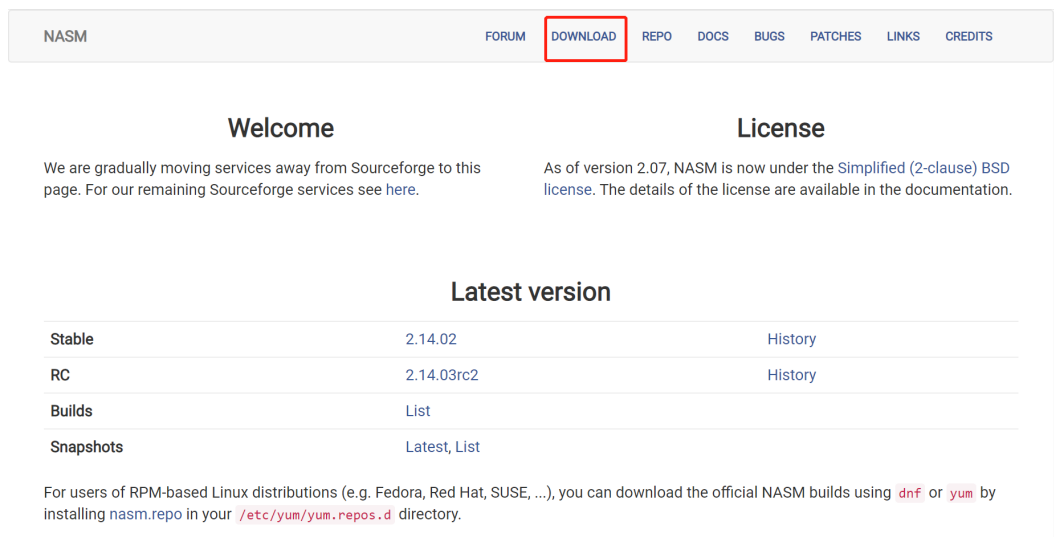


图 7: NASM 下载

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。

安装成功如图:

Index of /pub/nasm/releasebuilds





















	Name	Last modified	Size	Description
	Parent Directory		-	
	2.14.03rc2/	2018-12-30 08:06	-	
	2.14.03rc1/	2018-12-27 11:46	-	
	2.14.02/	2018-12-26 05:46	-	
	2.14.01/	2018-12-22 18:26	-	
	2.14.01rc5/	2018-12-14 13:26	-	
	2.14.01rc4/	2018-12-13 17:06	-	
	2.14.01rc3/	2018-12-12 18:06	-	
	2.14.01rc2/	2018-12-11 13:46	-	
	2.14.01rc1/	2018-11-26 21:46	-	
	2.14/	2018-11-07 13:46	-	
	2.14rc16/	2018-10-14 14:46	-	
	2.14rc15/	2018-07-06 03:26	-	
	2.14rc14/	2018-06-28 02:46	-	
	2.14rc13/	2018-06-27 21:26	-	
	2.14rc12/	2018-06-27 20:30	-	
	2.14rc11/	2018-06-25 23:26	-	
	2.14rc10/	2018-06-25 17:46	-	
	2.14rc9/	2018-06-25 17:26	-	
	2.14rc8/	2018-06-25 15:06	-	

图 8: NASM 下载

Index of /pub/nasm/releasebuilds/2.11.02/win32

	Name	Last modified	Size	Description
	Parent Directory		-	
	nasm-2.11.02-installer.exe	2014-02-19 16:06	762K	
	nasm-2.11.02-win32.zip	2014-02-19 16:06	423K	Executable only

图 9: NASM 下载

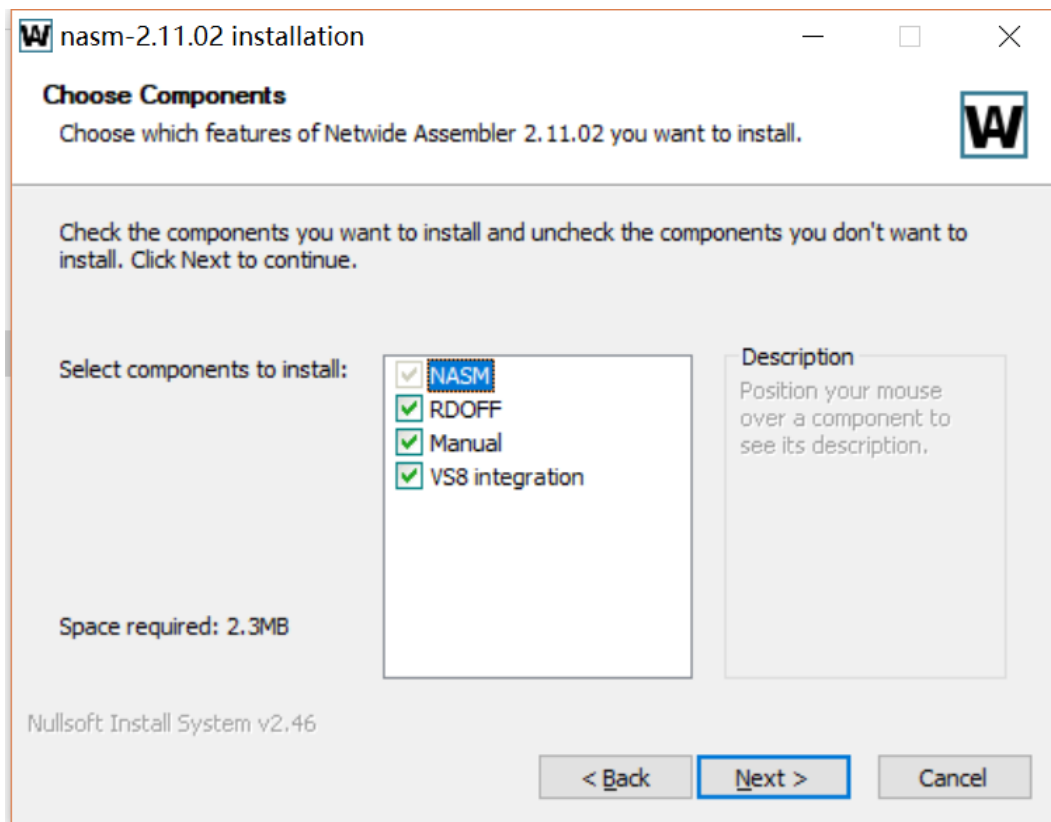


图 10: NASM 安装

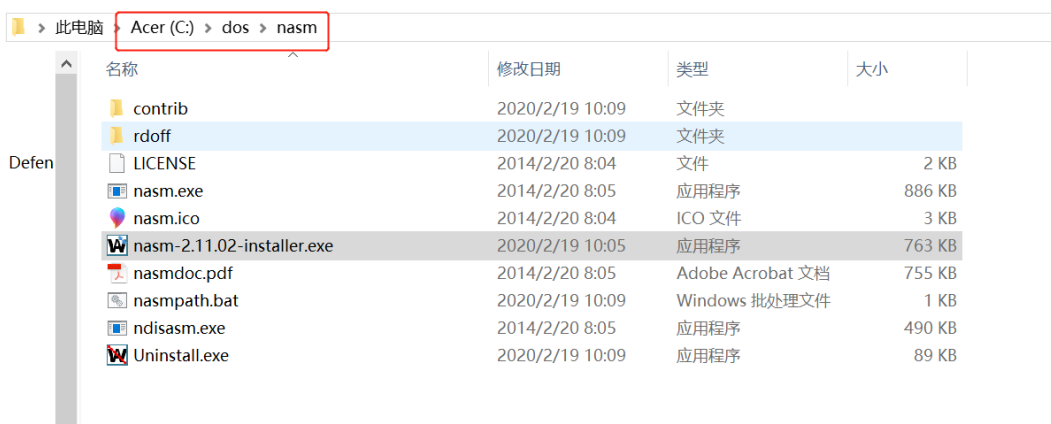


图 11: NASM 安装完毕

为了方面使用，配置环境变量：

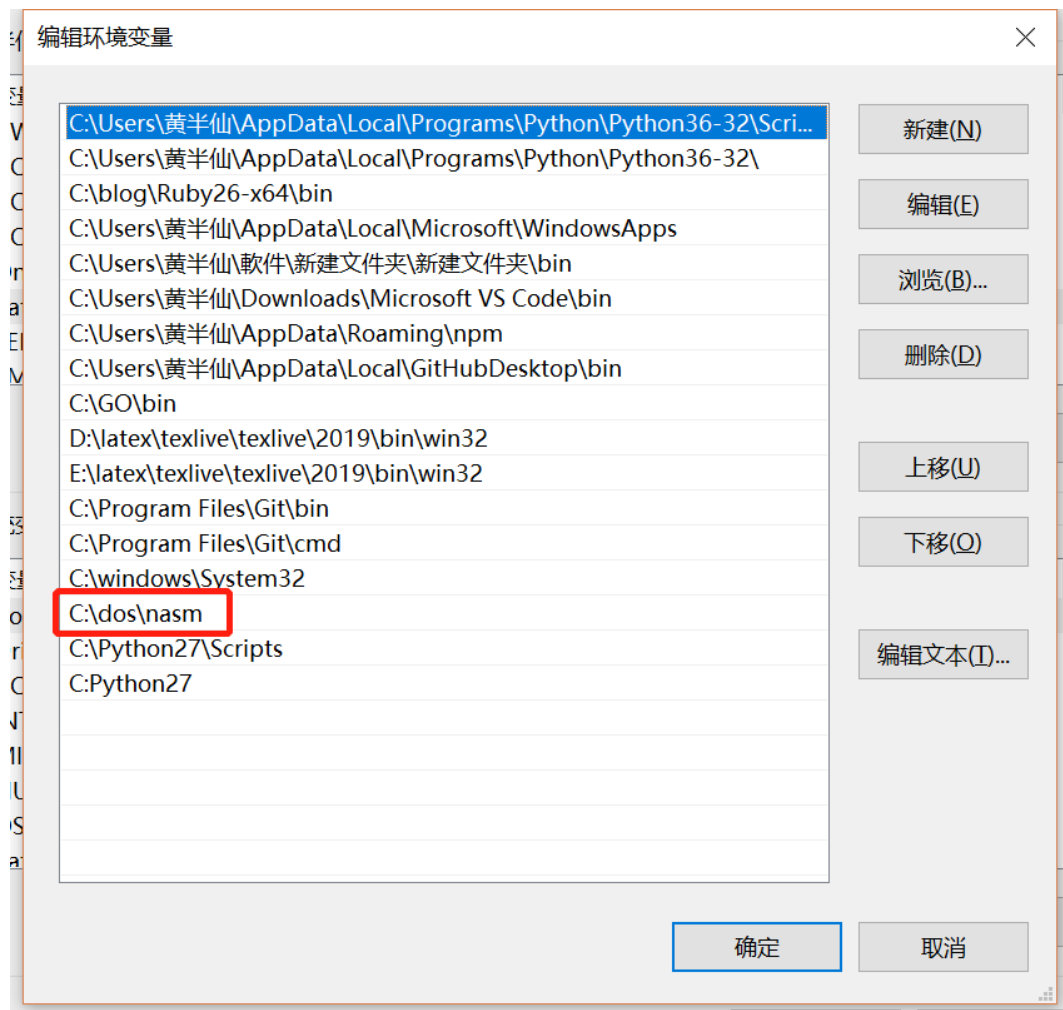


图 12: 配置环境变量

在命令窗口输入 `nasm -v` 命令检验

```
C:\WINDOWS\system32>nasm -v
NASM version 2.11.02 compiled on Feb 19 2014

C:\WINDOWS\system32>_
```

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；e'ses 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
; ASSUME cs:code,ds:code
; code SEGMENT
start:
; xor ax,ax
; mov ax,cs
; mov es,ax
; mov ds,ax
; mov es,ax
; mov ax,0B800h
; mov gs,ax
; mov byte[char], 'A'
```

图 14: 老师提供的部分代码

```
org 7c00h
; ASSUME cs:code,ds:code
; code SEGMENT
start:
; xor ax,ax
; mov ax,cs
; mov ds,ax
; mov ax,0B800h
; mov es,ax
```

图 15: 一种修改方式

```

; .386
; org 7c00h
; ASSUME cs:code,ds:code
; code SEGMENT
start:
    mov ax,0x7c00
    mov ds,ax
    mov ax,0B800h
    mov es,ax
    jmp 1:

```

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：

	7	6	5	4	3	2	1	0
含义	<u>BL</u>	<u>R</u>	<u>G</u>	<u>B</u>	<u>I</u>	<u>R</u>	<u>G</u>	<u>B</u>
	闪烁	背景	高亮	前景				
R:	红色							
G:	绿色							
B:	蓝色							

图 17: 字符属性

```

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

```
1 showmeg:
2     cld
3     mov     di,8                ;指定显示屏的位置
4     mov si,number             ;指向要显示的学号
5     mov cx,8                  ;8位学号，循环8次
6     rep movsw
7     mov di,28h                ;指定显示屏的位置
8     mov si,name               ;指向要显示的姓名拼音
9     mov cx,11                 ;循环11次
10    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：

```
1 show:
2     mov ax,word[x]             ;word[x]当前行
3     mov bx,80
4     mul bx;                     ;word[x]*80
5     add ax,word[y]             ;行+列=当前位置
6     mov bx,2
7     mul bx                     ;2*(行+列),一个字符显示占两个字节
8     mov bx,ax                  ;计算结果ax送给bx
9
10    mov dx,[es:bx]              ;获得当前显示屏的字符ascii码
11    cmp dl,' '                  ;如果为空，就显示，否则直接进入下一次循环
12    jnz loop1                   ;避开显示的信息
13
14    mov di,[cnt]                ;循环显示”learnos”的计数
15    mov al,byte[char+di]        ;要显示的字符
16    mov ah,byte[char+di+1]      ;设置字符属性
17
18    cmp al,'0'                  ;”learnos”末尾标志
19    jz s1
20
21    add di,2                    ;指向下一次要显示的字符，
22    2个字节
23
24    mov [cnt],di                ;存储下一次要显示的字符的偏移量
25
26 con:
27    mov [es:bx],ax              ;送入显示器
28    jmp loop1                   ;继续蛇形运动
```

```

28 | s1:
29 |     mov word[cnt],0           ;重新循环"learnos"
30 |     mov al,byte[char]        ;要显示的字符
31 |     mov ah,byte[char+1]      ;设置字符属性
32 |     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

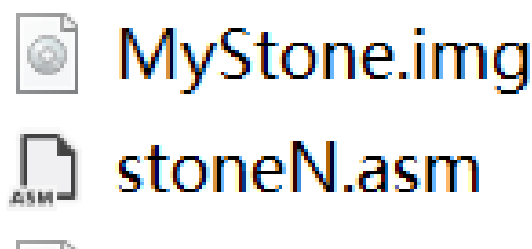


图 19: 编译结果

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，源于代码：

```

1 | times 510-($-$) db 0
2 | 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

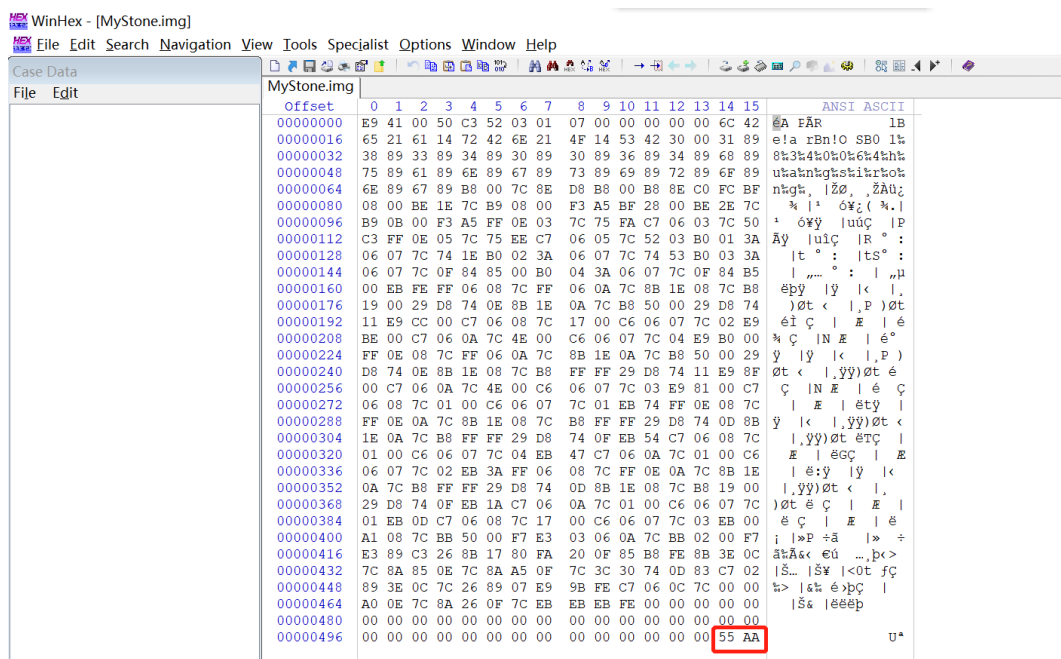


图 20: MyStone

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 设置虚拟机信息：

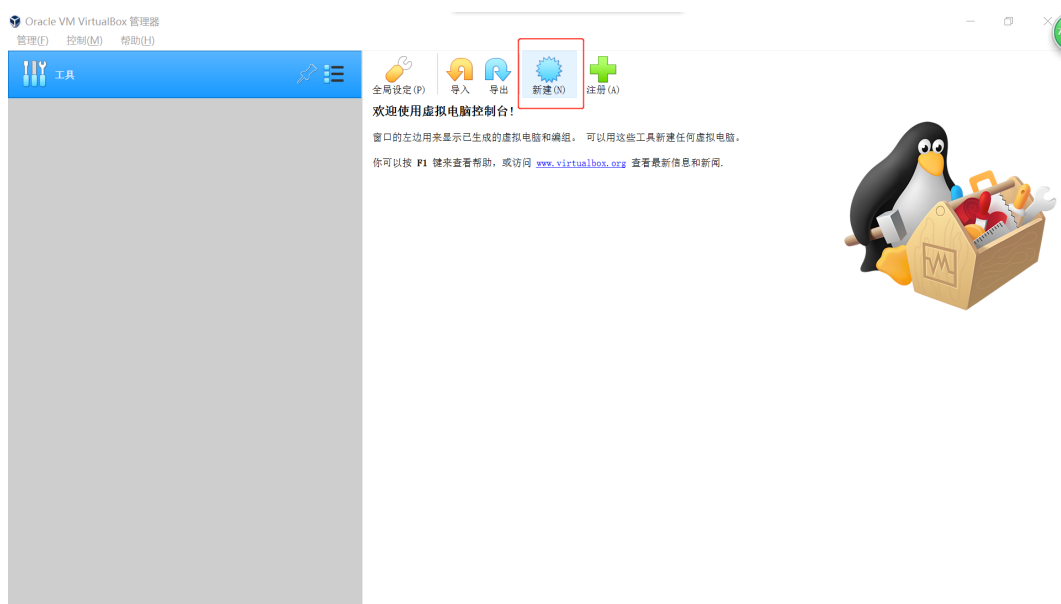


图 21: 新建虚拟机

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 , bootimg

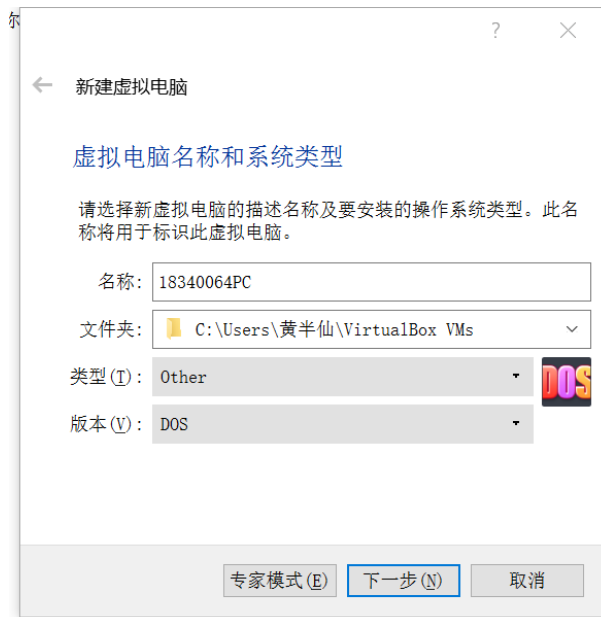


图 22: 设置虚拟机名字，类型

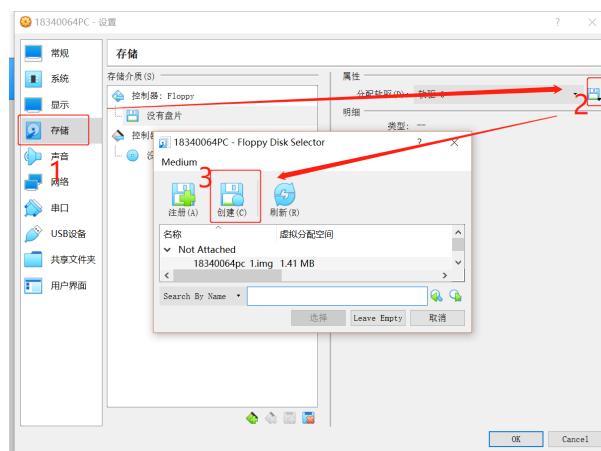


图 23: 创建 dos 映像文件

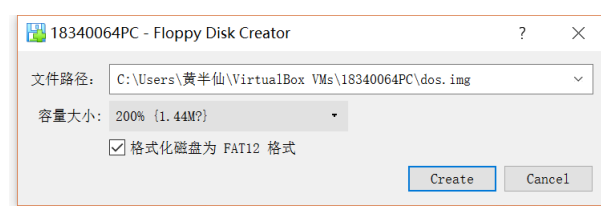


图 24: 创建 dos 映像文件

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

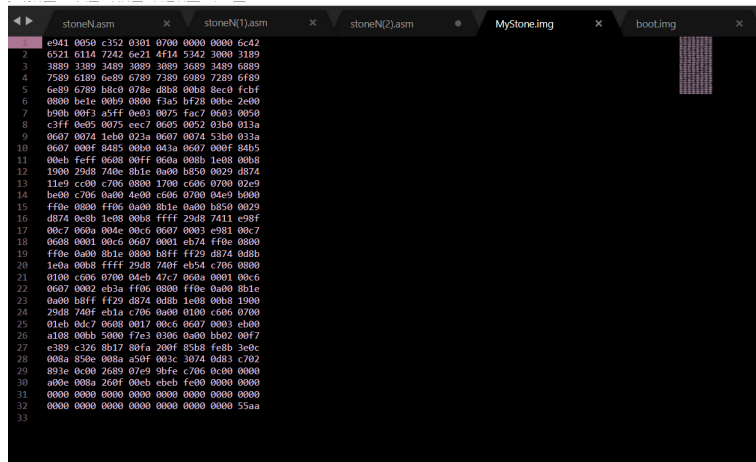


图 25: MyStone 文件

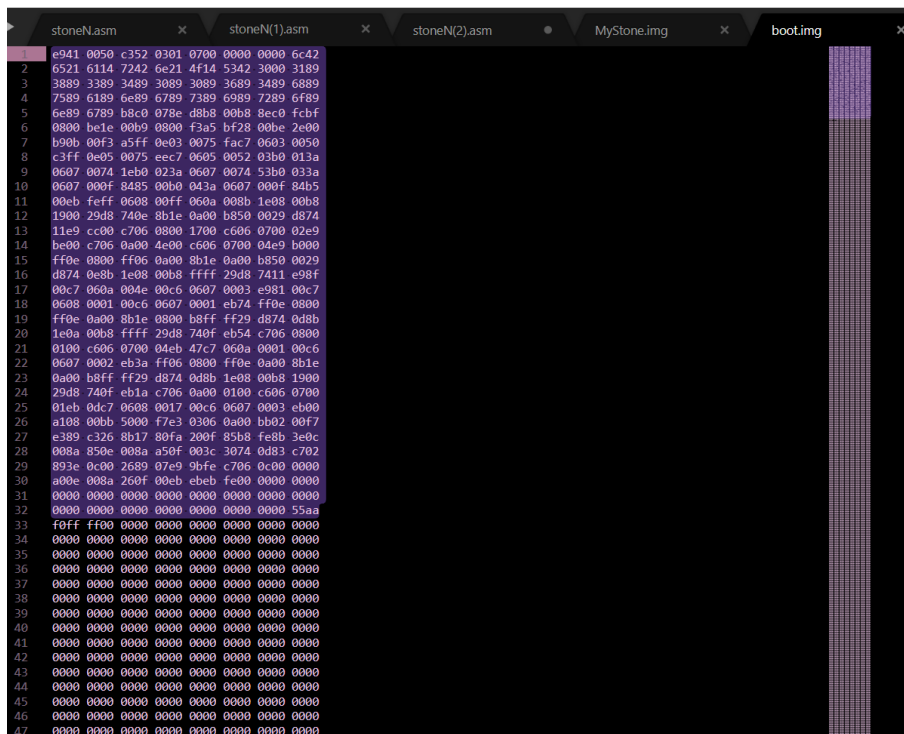


图 26: boot 文件

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

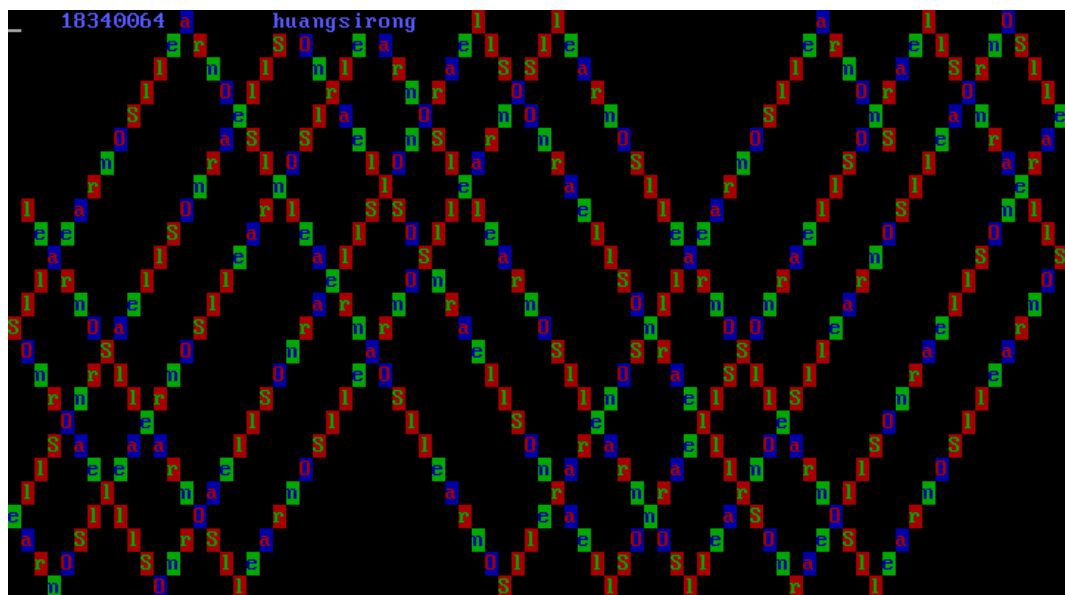


图 28: 启动画面

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

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

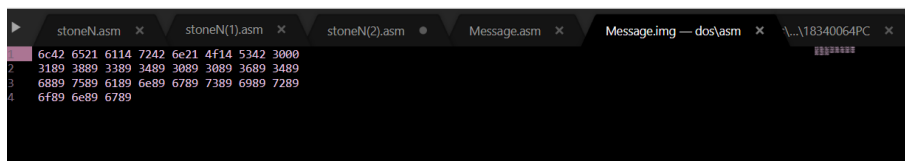


图 29: 汇编生成的机器码

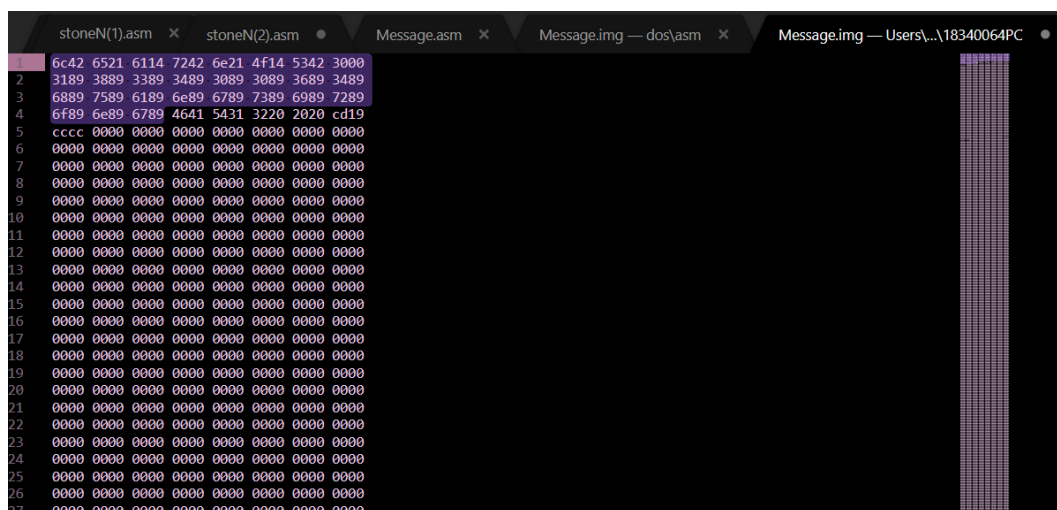


图 30: 写入 Message.img

7 实验总结

实验成功之后，把过程写进实验报告，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！