

中山大学数据科学与计算机学院本科生实验报告

(2020 学年春季学期)

课程名称：操作系统实验

任课教师：凌应标

助教：

年级&班级	2018 级 1 班	专业(方向)	计算机科学与技术(大数据方向)
学号	18308045	姓名	谷正阳
电话	13355426001	Email	Guzy0324@163.com
开始日期	2020.4.27	完成日期	2020.4.27

一、实验题目

操作系统 实验 1

二、实验目的

理解操作系统启动过程，了解 nasm 语法

三、实验内容

1. 实验步骤

- (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)程序用x86汇编语言编写，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

2. 实验原理

1. 基本组成：

DISK:

OS

MBR: 512B

启动代码: 446B, 检查分区表正确性, 加载并跳转到磁盘上的引导扇区

硬盘分区表: 64B, 描述分区状态和位置

分区 1: 16B

分区 2: 16B

分区 3: 16B

分区 4: 16B

结束标志 55AA: 2B

分区引导扇区:

跳转指令: 跳转到启动代码, 与 CPU 平台相关

文件卷头: 文件系统描述信息

启动代码: 跳转到 Bootloader

结束标志 55AA

Bootloader

ROM:

BIOS:

1. 基本的输入输出

2. 系统的配置信息

如从硬盘启动，网络启动，光盘启动

3. 开机自检程序

4. 系统自启动程序

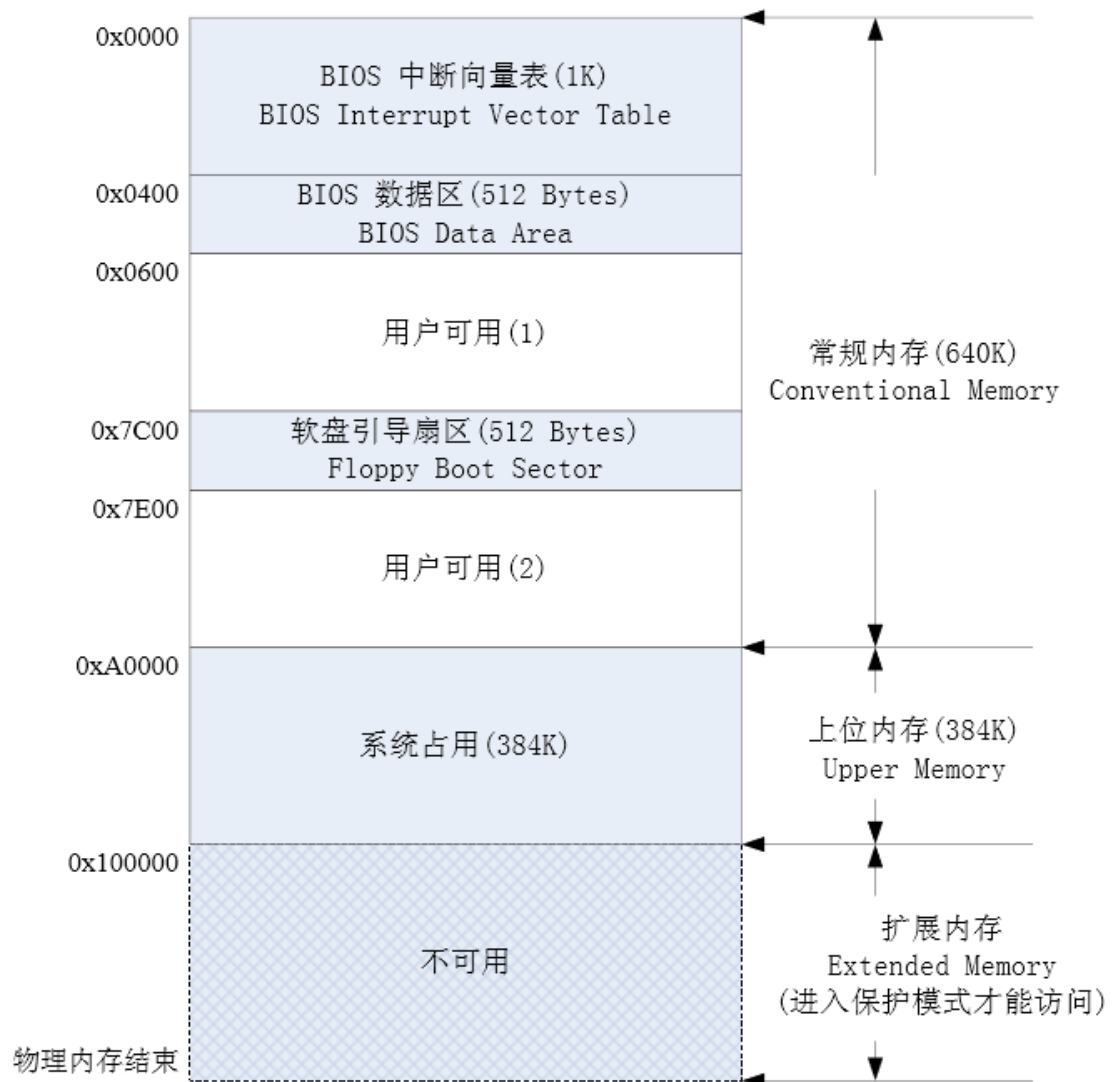


图 3-2：软盘引导扇区被加载后的物理内存的布局

2. 实模式：体现在程序中用到的地址都是真实的物理地址

1) 20 位物理地址 = 16 位段地址 : 4 位偏移地址

- 2) 最大寻址空间 $2^{20}B = 1MB$
- 3) 最大分段 $2^{16}B = 64KB$
- 4) 在 32 位 CPU 下，系统复位或加电时都是以实模式启动
- 5) CPU 单任务运行

3. 基本流程:

a. CPU 初始化

- 1) 寄存器值设为 0xFFFF，其他寄存器值设为 0
- 2) 在 0xFFFF0 执行一条无条件跳转指令，跳转到 BIOS 的真正启动代码处

b. BIOS:

1) 硬件自检 POST:

- a) 检查内存和显卡等关键部件的存在和工作状态
- b) 查找并执行显卡等接口卡 BIOS，进行设备初始化

2) 执行系统 BIOS:

- a) 检测和配置系统中安装的即插即用设备
- b) 更新 CMOS 中的扩展系统配置数据 ESCD
- c) 按指定启动顺序从软盘、硬盘或光驱启动
- d) 引导扇区读入物理内存 0x7C00 处

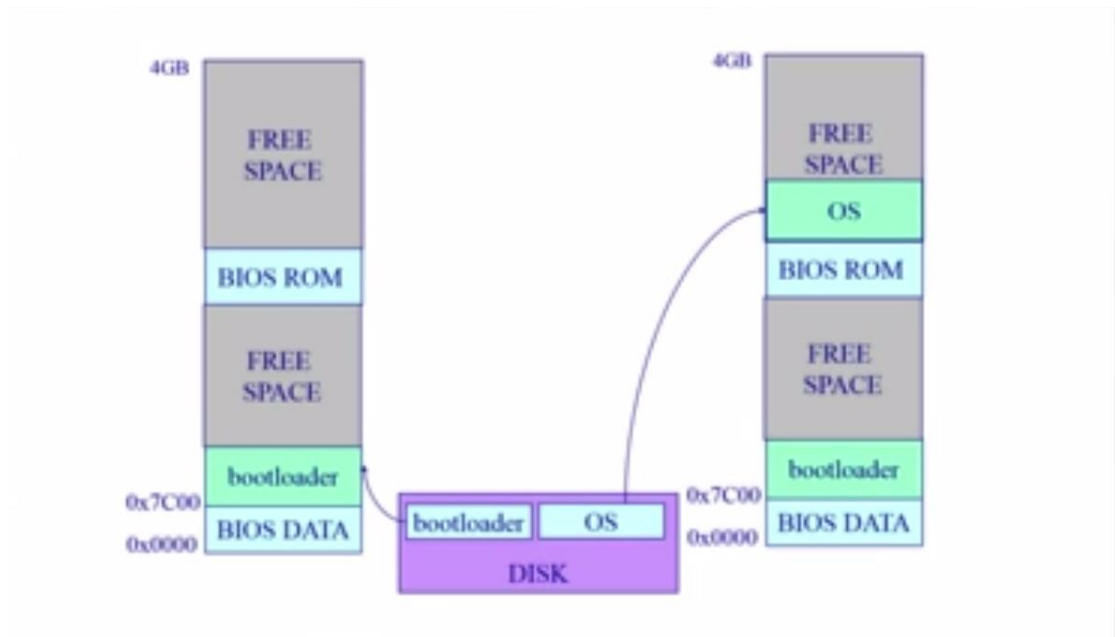
c. MBR

- 1) 检查分区表正确性，加载并跳转到磁盘上的引导扇区
- 2) 加载 Bootloader

d. Bootloader:

- 1) 加载程序：从文件系统中读取启动配置信息

2) 启动菜单：可选的操作系统内核列表和加载参数



3) 操作系统内核：依据配置加载指定操作内核并跳转到内核执行

四、实验结果

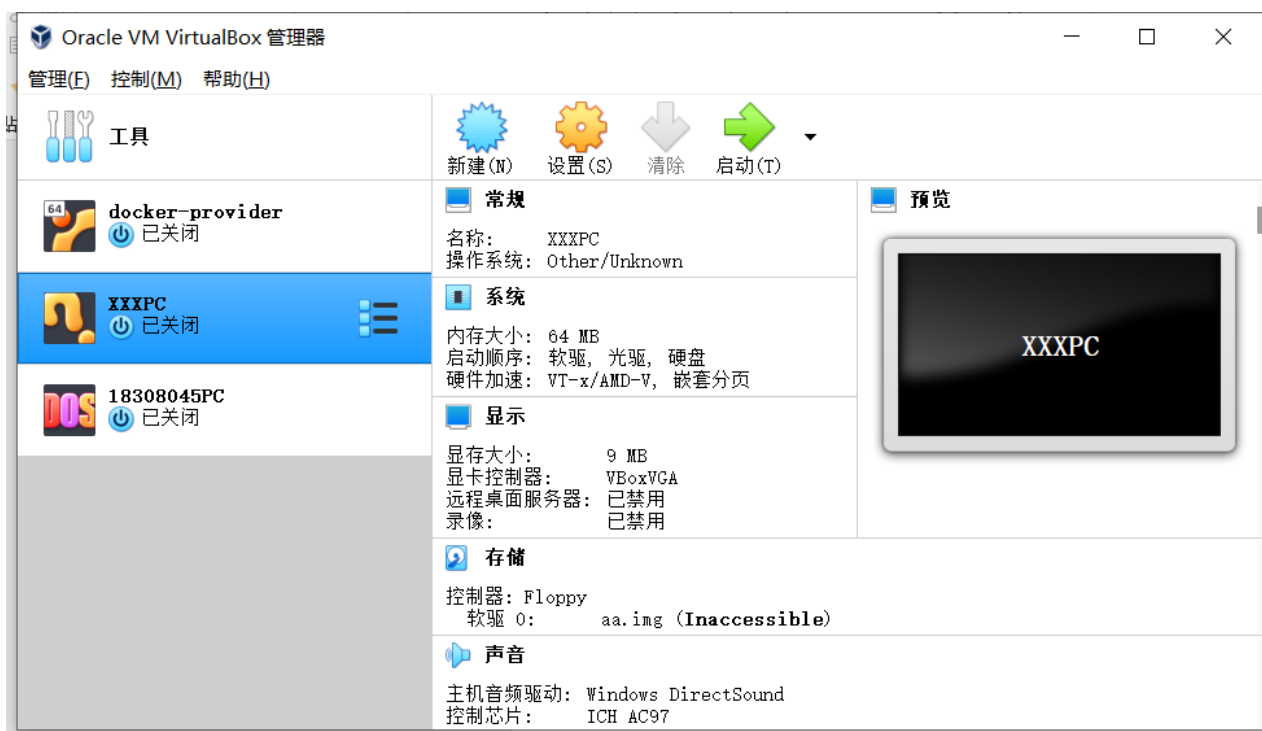
(1) 进入官网下载页下载最新版 VirtualBox

<https://download.virtualbox.org/virtualbox/6.1.6/VirtualBox-6.1.6-137129-Win.exe>

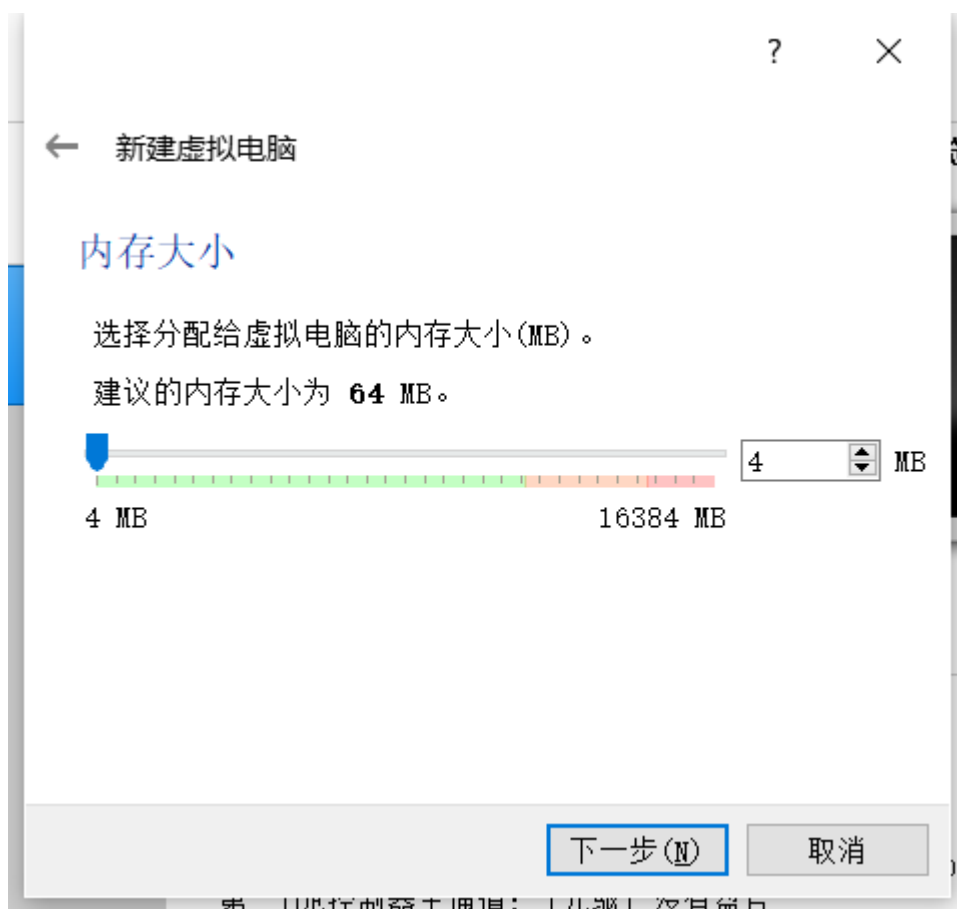
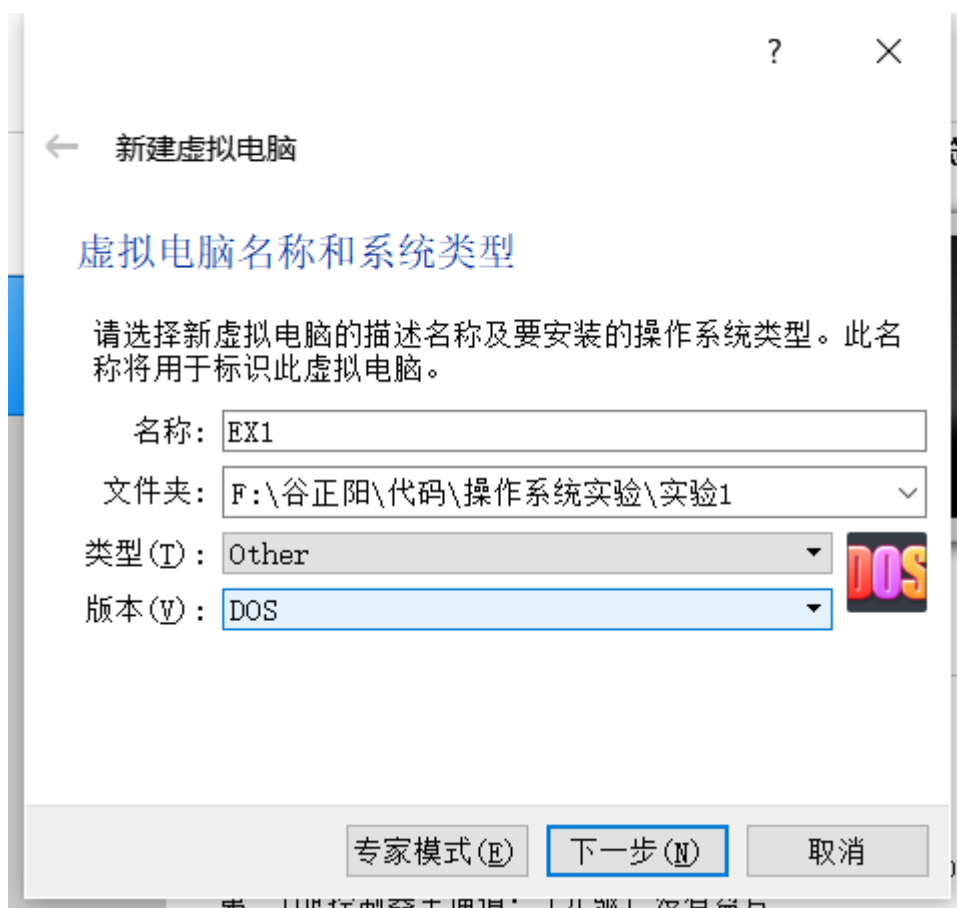
进入安装软件，使用默认安装配置

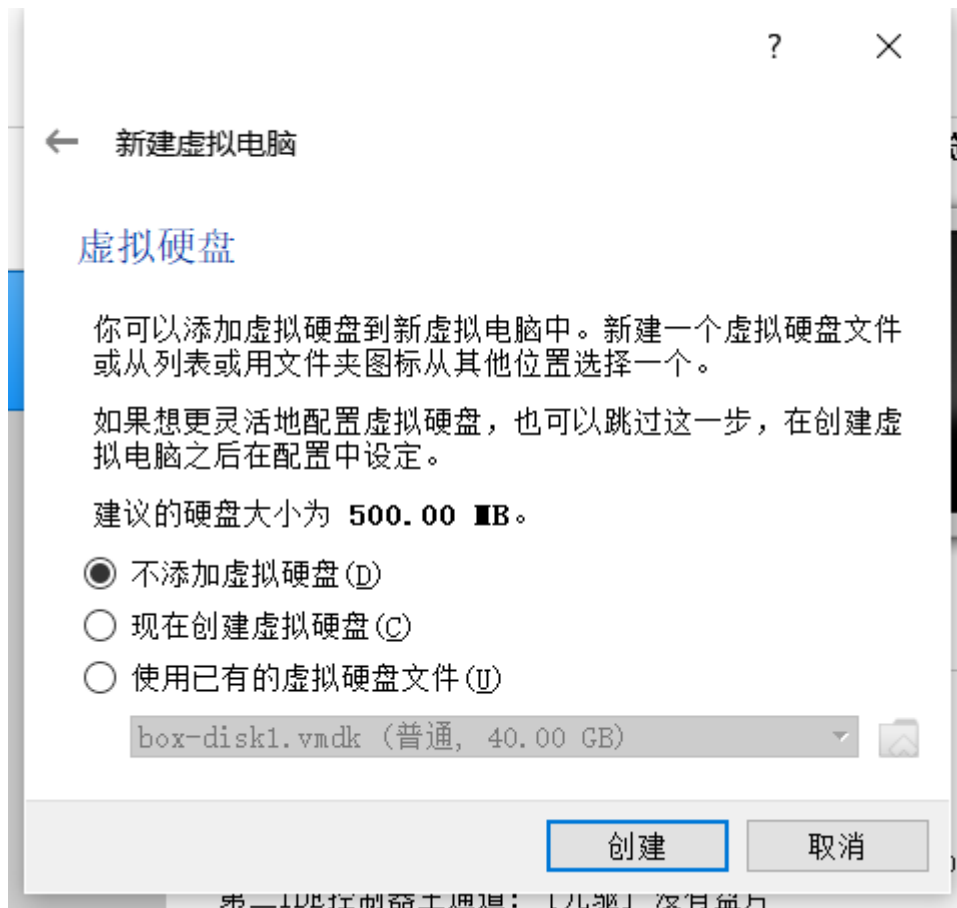


安装成功



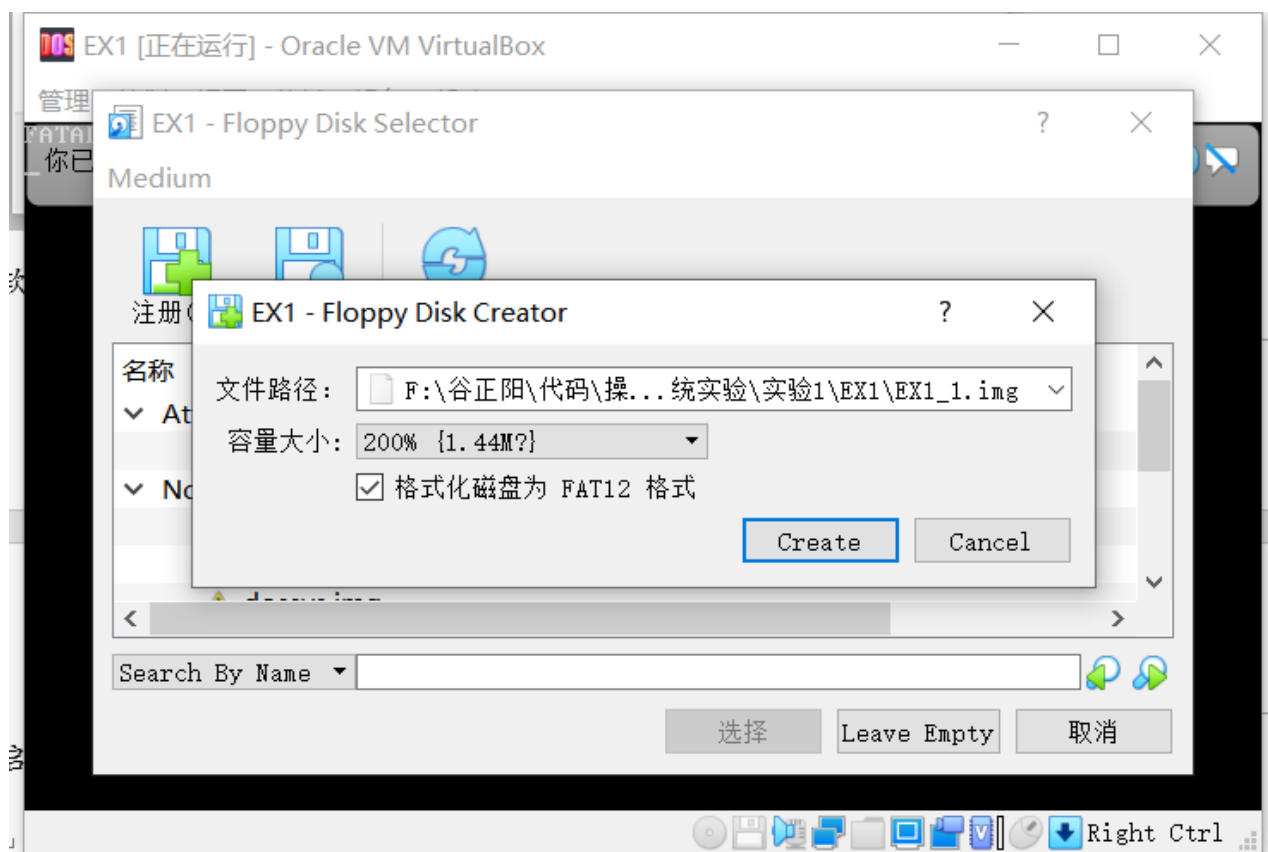
(2) PC创建配置:





软盘创建:

启动→设备→创建软驱→创建



如上步骤共创建3个

名称	修改日期	类型
Logs	2020/04/27 13:44	文件夹
EX1.vbox	2020/04/27 14:28	VirtualB
EX1.vbox-prev	2020/04/27 14:28	VBOX-F
EX1_1.img	2020/04/27 14:11	光盘映像
EX1_2.img	2020/04/27 13:47	光盘映像
EX1_3.img	2020/04/27 13:47	光盘映像

(3)在notepad++中安装HEX-Editor插件

https://github.com/chcg/NPP_HexEdit/releases/download/0.9.6/HexEditor_0.9.6_x64.zip

编写代码string.asm生成512B学号姓名

Address	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	10	11	12	13	14	15	16	17	18	19	1a	1b	1c	1d	1e	1f	Dump
00000000	31	38	33	30	38	30	34	35	67	75	7a	68	65	6e	67	79	61	6e	67	31	38	33	30	38	30	34	35	67	75	7a	68	65	18308045guzhengyang18308045guzhe
00000020	6e	67	79	61	6e	67	31	38	33	30	38	30	34	35	67	75	7a	68	65	6e	67	79	61	6e	67	31	38	33	30	38	30	34	ngyang18308045guzhengyang1830804
00000040	35	67	75	7a	68	65	6e	67	79	61	6e	67	31	38	33	30	38	30	34	35	67	75	7a	68	65	6e	67	79	61	6e	67	31	5guzhengyang18308045guzhengyang1
00000060	38	33	30	38	30	34	35	67	75	7a	68	65	6e	67	79	61	6e	67	31	38	33	30	38	30	34	35	67	75	7a	68	65	e	8308045guzhengyang18308045guzhen
00000080	67	79	61	6e	67	31	38	33	30	38	30	34	35	67	75	7a	68	65	6e	67	79	61	6e	67	31	38	33	30	38	30	34	35	gyang18308045guzhengyang18308045
000000a0	67	75	7a	68	65	6e	67	79	61	6e	67	31	38	33	30	38	30	34	35	67	75	7a	68	65	6e	67	79	61	6e	67	31	38	guzhengyang18308045guzhengyang18
000000c0	33	30	38	30	34	35	67	75	7a	68	65	6e	67	79	61	6e	67	31	38	33	30	38	30	34	35	67	75	7a	68	65	6e	67	308045guzhengyang18308045guzheng
000000e0	79	61	6e	67	31	38	33	30	38	30	34	35	67	75	7a	68	65	6e	67	79	61	6e	67	31	38	33	30	38	30	34	35	67	yang18308045guzhengyang18308045g
00000100	75	7a	68	65	6e	67	79	61	6e	67	31	38	33	30	38	30	34	35	67	75	7a	68	65	6e	67	79	61	6e	67	31	38	33	uzhengyang18308045guzhengyang183
00000120	30	38	30	34	35	67	75	7a	68	65	6e	67	79	61	6e	67	31	38	33	30	38	30	34	35	67	75	7a	68	65	6e	67	79	08045guzhengyang18308045guzhengy
00000140	61	6e	67	31	38	33	30	38	30	34	35	67	75	7a	68	65	6e	67	79	61	6e	67	31	38	33	30	38	30	34	35	67	75	yang18308045guzhengyang18308045gu
00000160	7a	68	65	6e	67	79	61	6e	67	31	38	33	30	38	30	34	35	67	75	7a	68	65	6e	67	79	61	6e	67	31	38	30	34	zhengyang18308045guzhengyang1830
00000180	38	30	34	35	67	75	7a	68	65	6e	67	79	61	6e	67	31	38	33	30	38	30	34	35	67	75	7a	68	65	6e	67	79	61	8045guzhengyang18308045guzhengya
000001a0	6e	67	31	38	33	30	38	30	34	35	67	75	7a	68	65	6e	67	79	61	6e	67	31	38	33	30	38	30	34	35	67	75	7a	ng18308045guzhengyang18308045guz
000001c0	68	65	6e	67	79	61	6e	67	31	38	33	30	38	30	34	35	67	75	7a	68	65	6e	67	79	61	6e	67	31	38	30	34	35	hengyang18308045guzhengyang18308
000001e0	30	34	35	67	75	7a	68	65	6e	67	79	61	6e	67	31	38	33	30	38	30	34	35	67	75	7a	68	65	6e	67	79	61	6e	045guzhengyang18308045guzhengyan

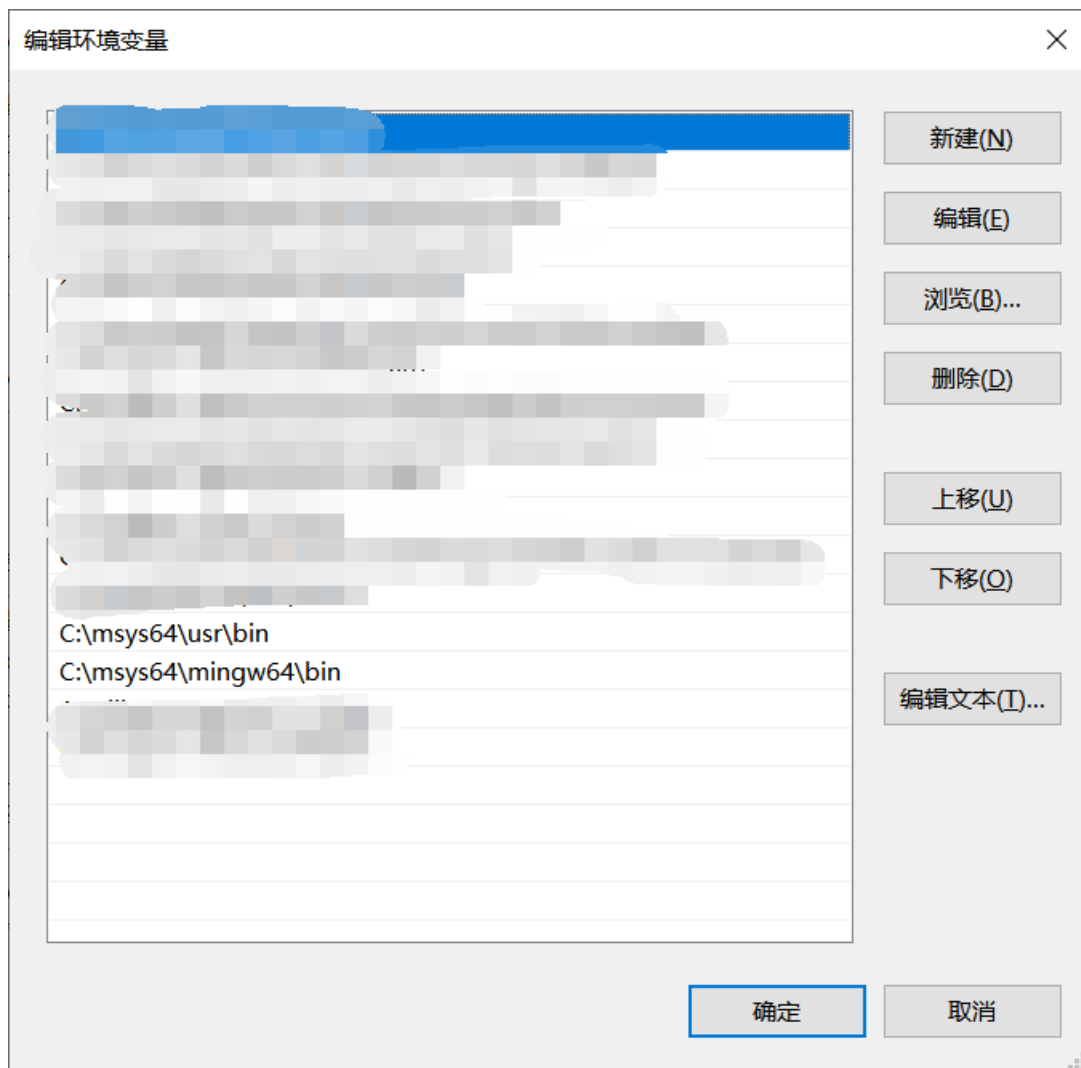
用fillout.cpp填写其至第一个软盘映像文件的首扇区

```
PS F:\谷正阳\代码\操作系统实验\实验1\codes> ./fillout
string.bin
../EX1/EX1_1.img
```

使用notepad++查看

Address	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	10	11	12	13	14	15	16	17	18	19	1a	1b	1c	1d	1e	1f	Dump	
00000000	31	38	33	30	38	30	34	35	67	75	7a	68	65	6e	67	79	61	6e	67	31	38	33	30	38	30	34	35	67	75	7a	68	65	18308045guzhengyang18308045guzhe	
00000020	6e	67	79	61	6e	67	31	38	33	30	38	30	34	35	67	75	7a	68	65	6e	67	79	61	6e	67	31	38	33	30	38	30	34	ngyang18308045guzhengyang1830804	
00000040	35	67	75	7a	68	65	6e	67	79	61	6e	67	31	38	33	30	38	30	34	35	67	75	7a	68	65	6e	67	79	61	6e	67	31	5guzhengyang18308045guzhengyang1	
00000060	38	33	30	38	30	34	35	67	75	7a	68	65	6e	67	79	61	6e	67	31	38	33	30	38	30	34	35	67	75	7a	68	65	6e	8308045guzhengyang18308045guzhen	
00000080	67	79	61	6e	67	31	38	33	30	38	30	34	35	67	75	7a	68	65	6e	67	79	61	6e	67	31	38	33	30	38	30	34	35	yang18308045guzhengyang18308045	
000000a0	67	75	7a	68	65	6e	67	79	61	6e	67	31	38	33	30	38	30	34	35	67	75	7a	68	65	6e	67	79	61	6e	67	31	38	guzhengyang18308045guzhengyang18	
000000c0	33	30	38	30	34	35	67	75	7a	68	65	6e	67	79	61	6e	67	31	38	33	30	38	30	34	35	67	79	61	6e	67	31	38	308045guzhengyang18308045guzheng	
000000e0	79	61	6e	67	31	38	33	30	38	30	34	35	67	75	7a	68	65	6e	67	79	61	6e	67	31	38	33	30	38	30	34	35	67	yang18308045guzhengyang18308045g	
00000100	75	7a	68	65	6e	67	79	61	6e	67	31	38	33	30	38	30	34	35	67	75	7a	68	65	6e	67	31	38	30	34	35	67	31	38	zuzhengyang18308045guzhengyang183
00000120	30	38	30	34	35	67	75	7a	68	65	6e	67	79	61	6e	67	31	38	33	30	38	30	34	35	67	75	7a	68	65	6e	67	79	08045guzhengyang18308045guzhengy	
00000140	61	6e	67	31	38	33	30	38	30	34	35	67	75	7a	68	65	6e	67	79	61	6e	67	31	38	33	30	38	30	34	35	67	75	7a	ang18308045guzhengyang18308045gu
00000160	7a	68	65	6e	67	79	61	6e	67	31	38	33	30	38	30	34	35	67	75	7a	68	65	6e	67	79	61	6e	67	31	38	33	30	38	zhengyang18308045guzhengyang1830
00000180	38	30	34	35	67	75	7a	68	65	6e	67	79	61	6e	67	31	38	33	30	38	30	34	35	67	75	7a	68	65	6e	67	79	61	8045guzhengyang18308045guzhengya	
000001a0	6e	67	31	38	33	30	38	30	34	35	67	75	7a	68	65	6e	67	79	61	6e	67	31	38	33	30	38	30	34	35	67	75	7a	ng18308045guzhengyang18308045guz	
000001c0	68	65	6e	67	79	61	6e	67	31	38	33	30	38	30	34	35	67	75	7a	68	65	6e	67	79	61	6e	67	31	38	30	34	35	38	hengyang18308045guzhengyang18308
000001e0	30	34	35	67	75	7a	68	65	6e	67	79	61	6e	67	31	38	33	30	38	30	34	35	67	75	7a	68	65	6e	67	79	61	6e	045guzhengyang18308045guzhengyan	
00000200	ff	ff	ff	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	?	
00000220	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	?	
00000240	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	?	
00000260	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	?	
00000280	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	?	
000002a0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	?	
000002c0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	?	
000002e0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	?	
00000300	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	?	
00000320	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	?	
00000340	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	?	
00000360	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	?	
00000380	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	?	
000003a0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	?	

(4) 安装msys2, 将其加到环境变量



使用pacman -S命令安装mingw-w64-x86_64-toolchain和mingw64/mingw-w64-x86_64-nasm

```
Windows PowerShell
PS F:\谷正阳\代码\操作系统实验\实验1\EX1> pacman -S mingw64/mingw-w64-x86_64-nasm
warning: mingw-w64-x86_64-nasm-2.14.02-1 is up to date -- reinstalling
resolving dependencies...
looking for conflicting packages...

Packages (1) mingw-w64-x86_64-nasm-2.14.02-1

Total Installed Size: 3.24 MiB
Net Upgrade Size: 0.00 MiB

:: Proceed with installation? [Y/n] _
```

(5)修改stoneN.asm为反弹彩色学号+姓名缩写，并为代码增加必要注释

发现问题:

(1) 此代码是masm语法，nasm语法没有assume伪指令

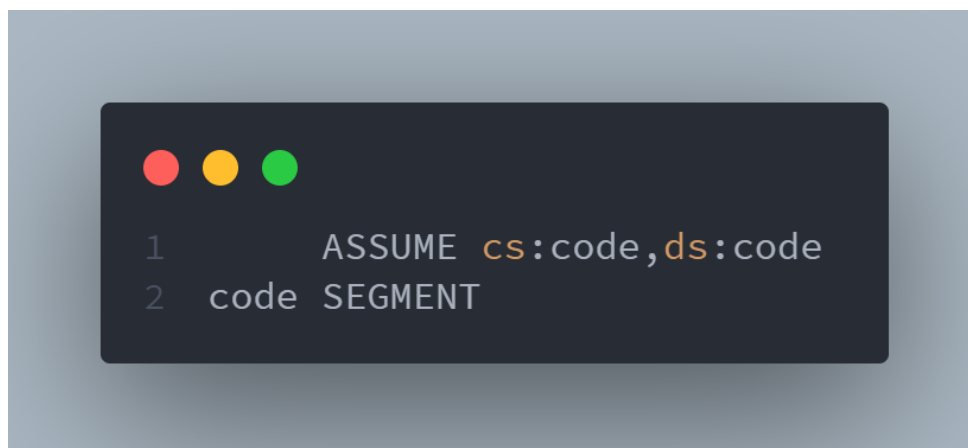


查询assume的用途:

ASSUME伪指令是配合SEGMENT伪指令使用的，其作用是通知MASM寻址某一特定段时要使用哪一个段寄存器。MASM在生成实际指定寻址方式操作

数的机器代码时要用到这些信息。要认识到ASSUME伪指令的存在只是为了告诉MASM如何将各种段寄存器同符号表中的符号关联起来，MASM实际上并不将任何值装入任何段寄存器，也就是说用户必须自己通过CPU指令完成。

结合代码：



得知，此处的assume是把cs和ds绑定到code段上，相当于nasm不加这个语法，因而可以注释掉。

- (2) .386: 告诉汇编器应该生成386处理的伪代码，是masm的伪代码，在nasm中被视作label所以注释掉。

```
stone.asm:11: warning: label alone on a line without a colon might be in error [-w+orphan-labels]
```

- (3) nasm使用segment 段名称和section 段名称来定义段，不同于masm

段名 SEGMENT [定位类型] [组合类型] ['类别']

XXXXXXXXXX

段名 ENDS

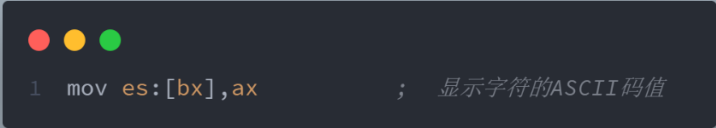
格式

```
stone.asm:162: error: parser: instruction expected
stone.asm:163: error: parser: instruction expected
```

因此，可以注释掉。

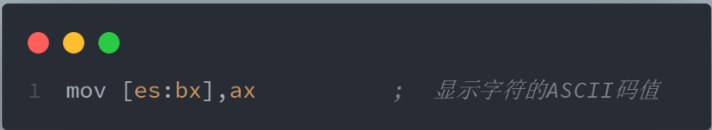
(4) nasm访问内存的语法是：粒度[内存地址]

因而应将以下代码



```
1  mov es:[bx],ax          ; 显示字符的ASCII码值
```

改为

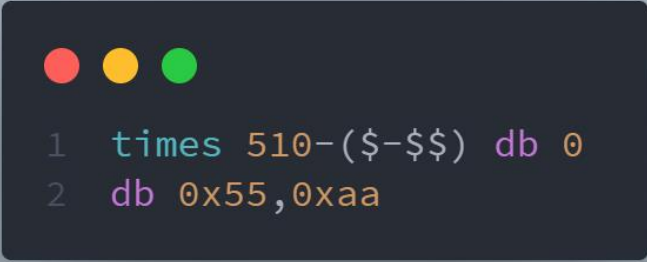


```
1  mov [es:bx],ax          ; 显示字符的ASCII码值
```

编译成功！

(5) 直接修改后缀名为img首扇区无法运行，检查原因发现510和511位0x55和0xaa

使用如下语法填充至512字节。



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

\$表示当前指令地址，\$\$表示段首地址，\$-\$\$表示前面指令共占的字节数，此语法表示填充0至510字节，然后填充0x55，0xaa。

使用notepad++查看结果：

Address	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	10	11	12	13	14	15	16	17	18	19	1a	1b	1c	1d	1e	1f
00000000	b1	00	ba	81	7d	8c	c8	8e	c0	8e	d8	8e	c0	b8	00	b8	8e	c0	ff	0e	78	7d	75	fa	c7	06	78	7d	50	c3	ff	0e
00000020	7a	7d	75	ee	c7	06	78	7d	50	c3	c7	06	7a	7d	44	02	b0	01	3a	06	7c	7d	74	1e	b0	02	3a	06	7c	7d	74	53
00000040	b0	03	3a	06	7c	7d	0f	84	85	00	b0	04	3a	06	7c	7d	0f	84	b5	00	eb	fe	ff	06	7d	7d	ff	06	7f	7d	8b	1e
00000060	7d	7d	b8	19	00	29	d8	74	0e	8b	1e	7f	7d	b8	50	00	29	d8	74	11	e9	cc	00	c7	06	7d	7d	17	00	c6	06	7c
00000080	7d	02	e9	be	00	c7	06	7f	7d	4e	00	c6	06	7c	7d	04	e9	b0	00	ff	0e	7d	7d	ff	06	7f	7d	8b	1e	7f	7d	b8
000000a0	50	00	29	d8	74	0e	8b	1e	7d	7d	b8	ff	ff	29	d8	74	11	e9	8f	00	c7	06	7f	7d	4e	00	c6	06	7c	7d	03	e9
000000c0	81	00	c7	06	7d	7d	01	00	c6	06	7c	7d	01	eb	74	ff	0e	7d	7d	ff	0e	7f	7d	8b	1e	7d	7d	b8	ff	ff	29	d8
000000e0	74	0d	8b	1e	7f	7d	b8	ff	ff	29	d8	74	0f	eb	54	c7	06	7d	7d	01	00	c6	06	7c	7d	04	eb	47	c7	06	7f	7d
00000100	01	00	c6	06	7c	7d	02	eb	3a	ff	06	7d	7d	ff	0e	7f	7d	8b	1e	7f	7d	b8	ff	ff	29	d8	74	0d	8b	1e	7d	7d
00000120	b8	19	00	29	d8	74	0f	eb	1a	c7	06	7f	7d	01	00	c6	06	7c	7d	01	eb	0d	c7	06	7d	7d	17	00	c6	06	7c	7d
00000140	03	eb	00	31	c0	a1	7d	7d	bb	50	00	f7	e3	03	06	7f	7d	bb	02	00	f7	e3	89	c3	88	cc	89	da	89	cb	8a	87
00000160	81	7d	89	d3	80	c1	01	80	f9	0d	74	06	26	89	07	e9	a0	fe	b1	00	eb	f6	eb	fe	50	c3	44	02	01	07	00	00
00000180	00	20	31	38	33	30	38	30	34	35	5f	67	7a	79	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001a0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001c0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001e0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	55	aa

(6) 仍然无法运行，使用bochsdbg断点调试检查问题

```
Bochs for Windows - Console
00000000fffffd: (
): add ah, bh ; 00fc
00000000fffff: (
): xchg dx, ax ; 92
000000010000000: (
): add byte ptr ds:[bx+si], al ; 0000
000000010000002: (
): add byte ptr ds:[bx+si], al ; 0000
<bochs:2> n
Next at t=1
(0) [0x000000fe05b] f000:e05b (unk. ctxt): xor ax, ax ; 31c0
<bochs:3> n
Next at t=2
(0) [0x000000fe05d] f000:e05d (unk. ctxt): out 0x0d, al ; e60d
<bochs:4> n
Next at t=3
(0) [0x000000fe05f] f000:e05f (unk. ctxt): out 0xda, al ; e6da
<bochs:5> n
Next at t=4
(0) [0x000000fe061] f000:e061 (unk. ctxt): mov al, 0xc0 ; b0c0
<bochs:6> n
Next at t=5
(0) [0x000000fe063] f000:e063 (unk. ctxt): out 0xd6, al ; e6d6
<bochs:7> n
Next at t=6
(0) [0x000000fe065] f000:e065 (unk. ctxt): mov al, 0x00 ; b000
<bochs:8>
```

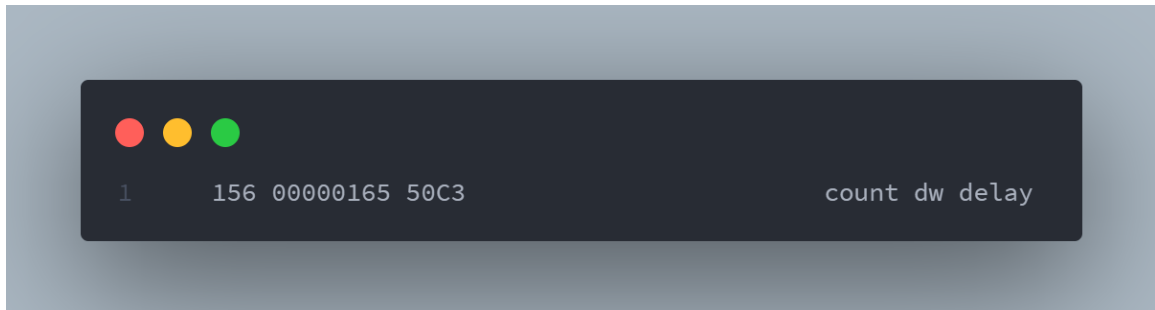
发现前几句都不是代码中的指令，经百度发现，是把mbr加载到物理内存0x7c00处，

因此在此打断点，跳过去查看问题。

```
Bochs for Windows - Console
(0) [0x000000007c02] 0000:7c02 (unk. ctxt): mov es, ax ; 8ec0
<bochs:4> n
Next at t=2094336
(0) [0x000000007c04] 0000:7c04 (unk. ctxt): mov ds, ax ; 8ed8
<bochs:5> n
Next at t=2094337
(0) [0x000000007c06] 0000:7c06 (unk. ctxt): mov es, ax ; 8ec0
<bochs:6> n
Next at t=2094338
(0) [0x000000007c08] 0000:7c08 (unk. ctxt): mov ax, 0xb800 ; b800b8
<bochs:7> n
Next at t=2094339
(0) [0x000000007c0b] 0000:7c0b (unk. ctxt): mov gs, ax ; 8ee8
<bochs:8> n
Next at t=2094340
(0) [0x000000007c0d] 0000:7c0d (unk. ctxt): mov byte ptr ds:0x026e, 0x41 ; c6066e0241
<bochs:9> n
Next at t=2094341
(0) [0x000000007c12] 0000:7c12 (unk. ctxt): dec word ptr ds:0x0265 ; ff0e6502
<bochs:10> x /1dd 0x265
[bochs]:
0x0000000000000265 <bogus+ 0>: 0
<bochs:11>
```

经检查发现如上位置本应该是50000，然而结果却是0。

观察lst文件



发现其正确位置应在 $0x7c00 + 0x156 = 0x7d56$

```
<bochs:6> x /2bx 0x7d65
[bochs]:
0x0000000000007d65 <bogus+ 0>: 0x50 0xc3
```

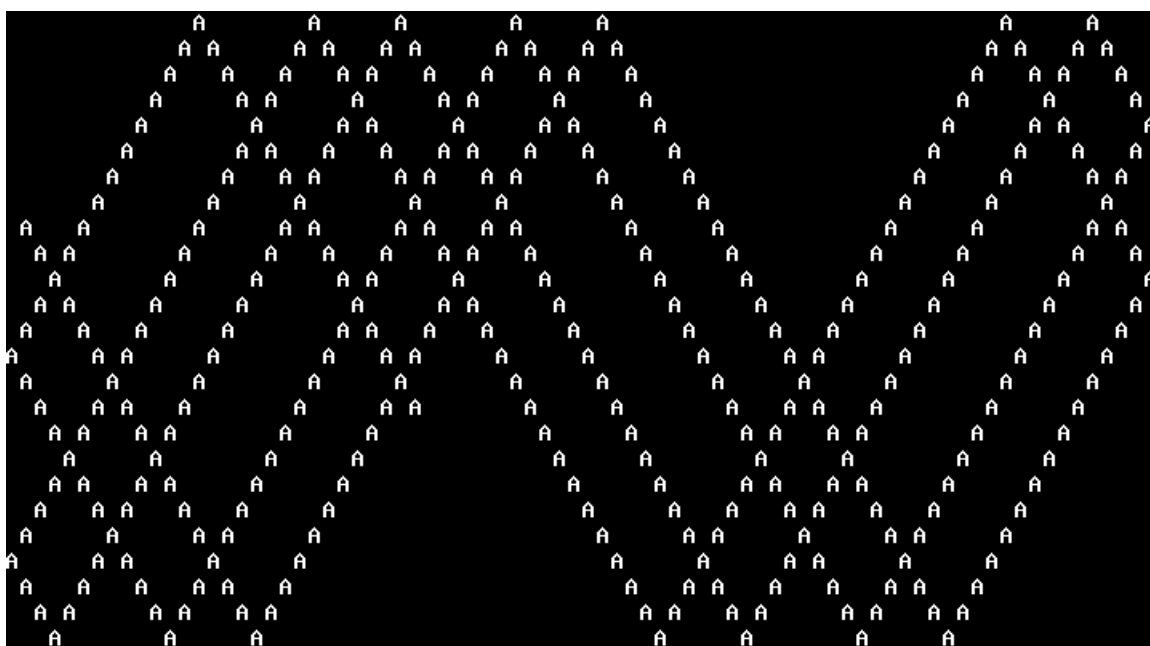
观察得出，物理内存 $0x7d65$ 的值 $0xc350$ 即为50000

经百度,发现org 100h使虚拟内存偏移了100h,应该改为org 0x7c00,这样就对应上了。

(7) 仍然无法显示，断点调试发现可以运行到show，但是[es:bx]的时候ex始终是0，es作为显存的段地址，应该是0xb800，检索代码，发现代码将gs初始化为0xb800而未将es初始化为0xb800，因而将es初始化为0xb800



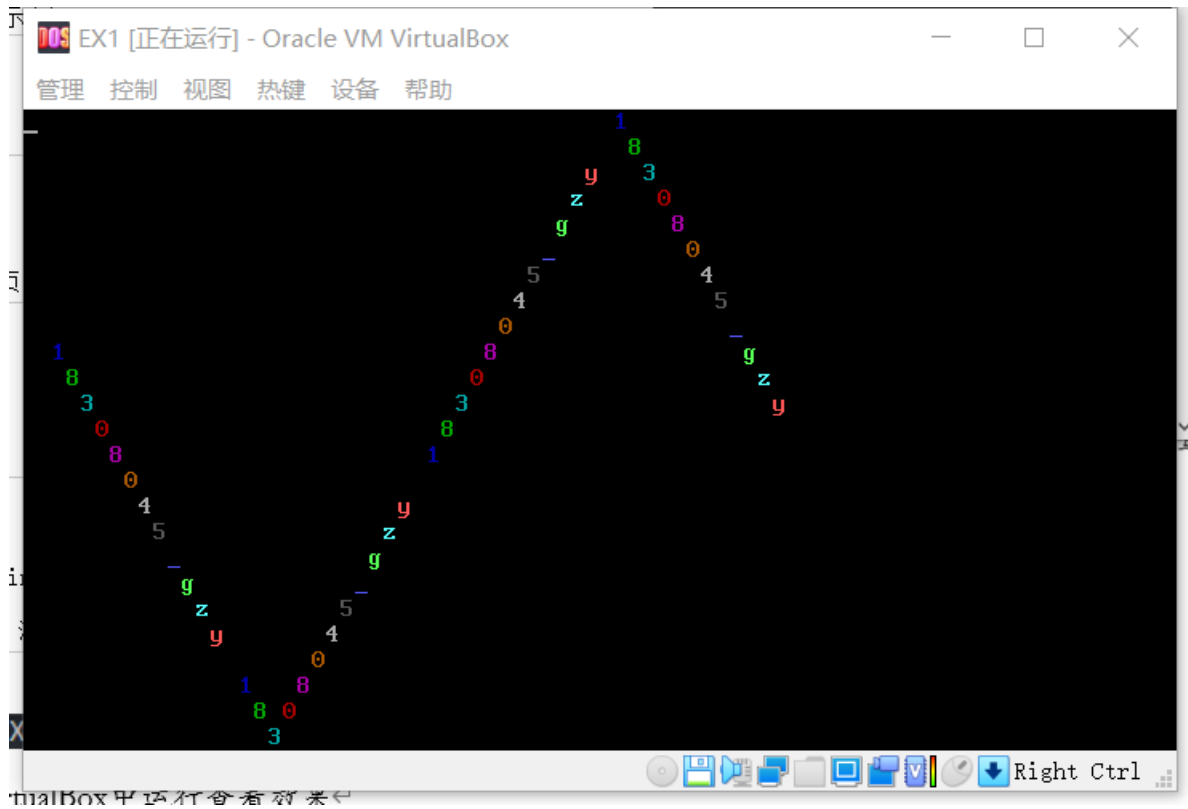
运行成功：



(6)使用fillout.cpp填写生成的mbr到第二个软盘映像中

```
PS F:\谷正阳\代码\操作系统实验\实验1\codes> ./fillout  
mbr.bin  
../EX1/EX1_2.img
```

在VirtualBox中运行查看效果



(7)在github上创建、初始化仓库，提交代码

https://github.com/guzhegyang/operating_system_experiment.git

五、实验感想

深刻理解了操作系统启动过程，且了解了 nasm 的部分语法

附录（流程图，注释过的代码）：

代码见

https://github.com/guzhegyang/operating_system_experiment.git