**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**](mailto: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

1. **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：

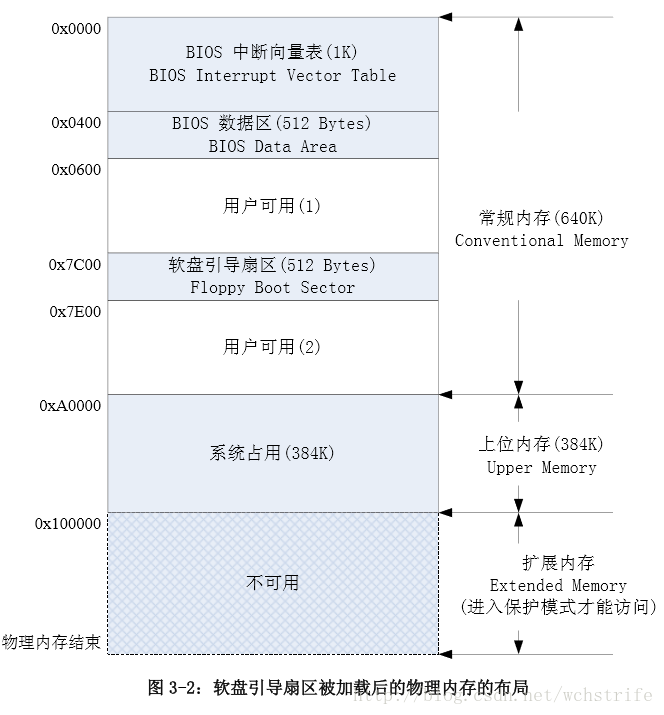
1. 基本的输入输出

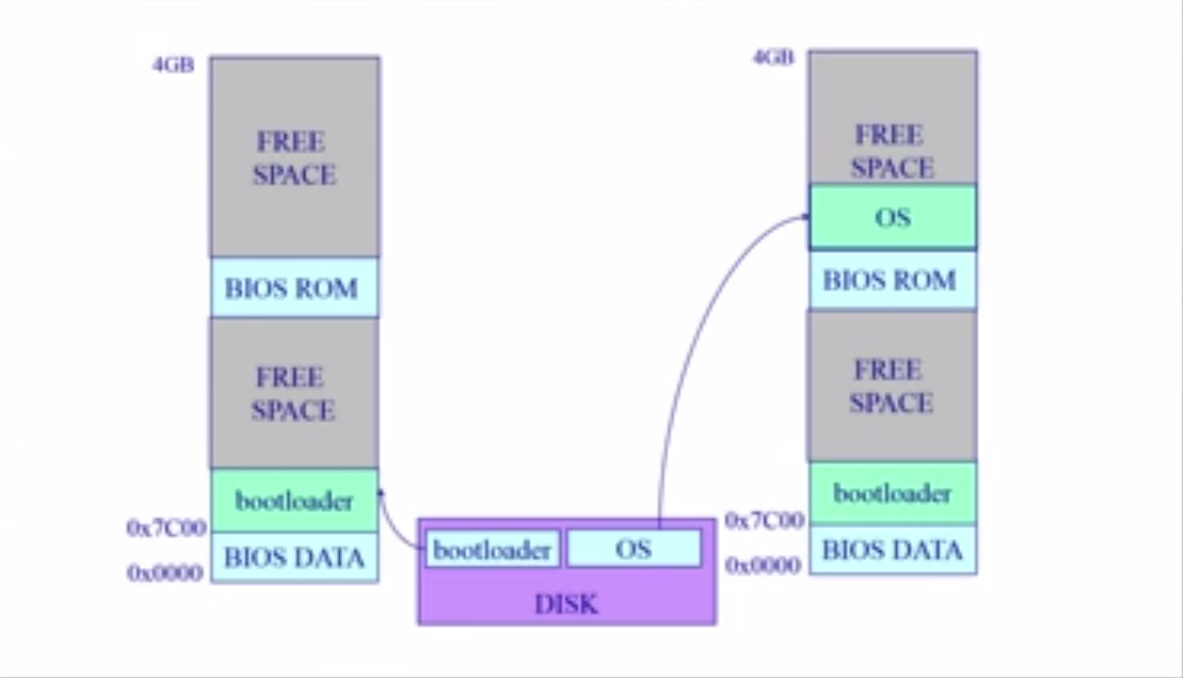
2. 系统的配置信息

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



1. 实模式：体现在程序中用到的地址都是真实的物理地址
2. 20位物理地址 = 16位段地址 : 4位偏移地址
3. 最大寻址空间2^20B = 1MB
4. 最大分段2^16B = 64KB
5. 在32位CPU下，系统复位或加电时都是以实模式启动
6. CPU单任务运行
7. 基本流程：
8. CPU初始化
9. 寄存器值设为0xFFFF，其他寄存器值设为0
10. 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
11. BIOS：
12. 硬件自检POST：
    1. 检查内存和显卡等关键部件的存在和工作状态
    2. 查找并执行显卡等接口卡BIOS，进行设备初始化
13. 执行系统BIOS：
    1. 检测和配置系统中安装的即插即用设备
    2. 更新CMOS中的扩展系统配置数据ESCD
    3. 按指定启动顺序从软盘、硬盘或光驱启动
    4. 引导扇区读入物理内存 0x7C00 处
14. MBR
15. 检查分区表正确性，加载并跳转到磁盘上的引导扇区
16. 加载Bootloader
17. Bootloader：
18. 加载程序：从文件系统中读取启动配置信息
19. 启动菜单：可选的操作系统内核列表和加载参数
20. 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**

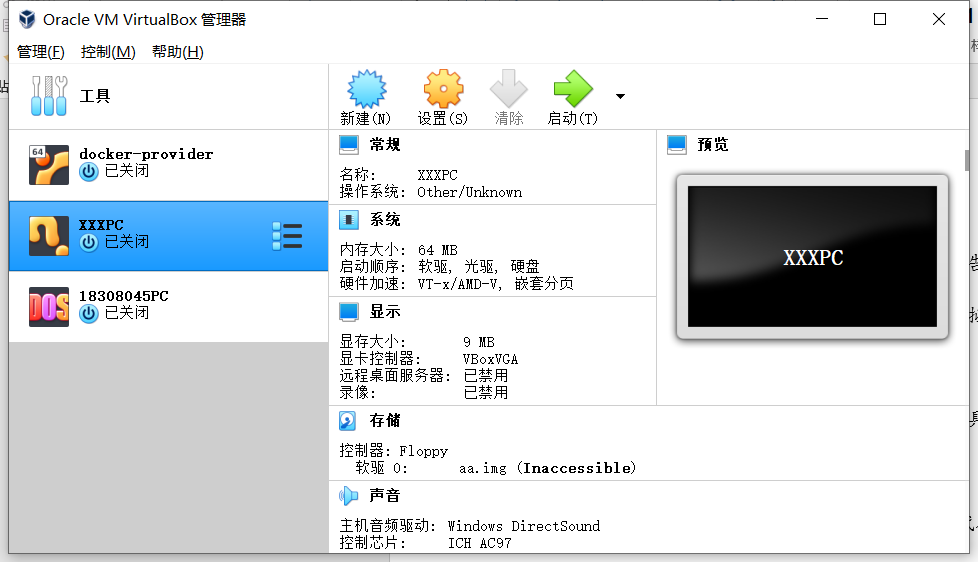
(1) 进入官网下载页下载最新版VirtualBox

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

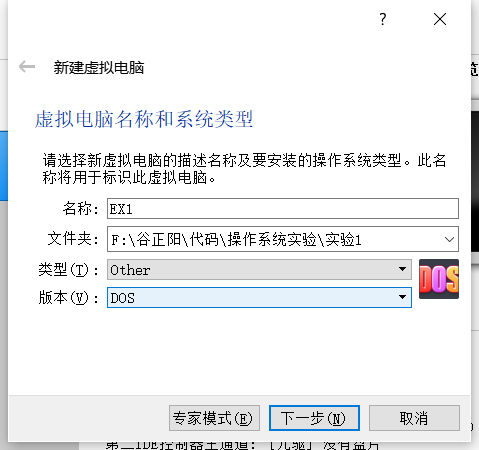
进入安装软件，使用默认安装配置

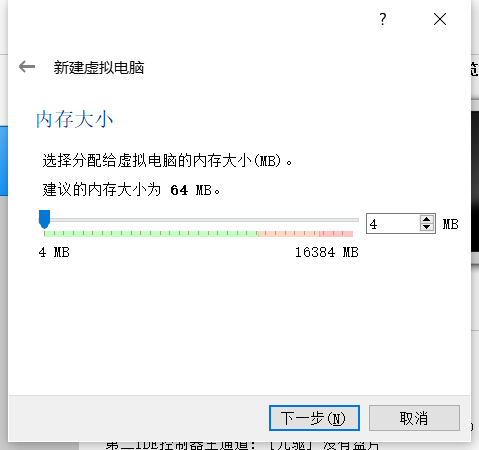


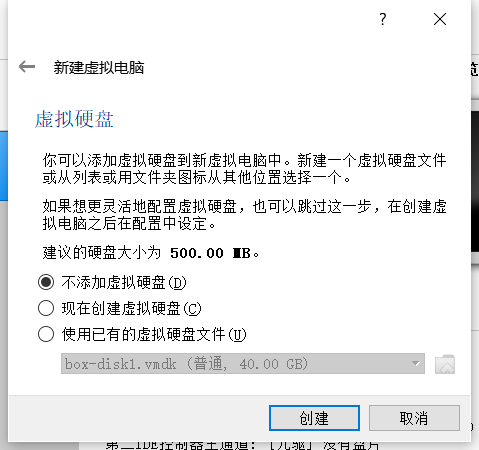
安装成功



(2) PC创建配置：

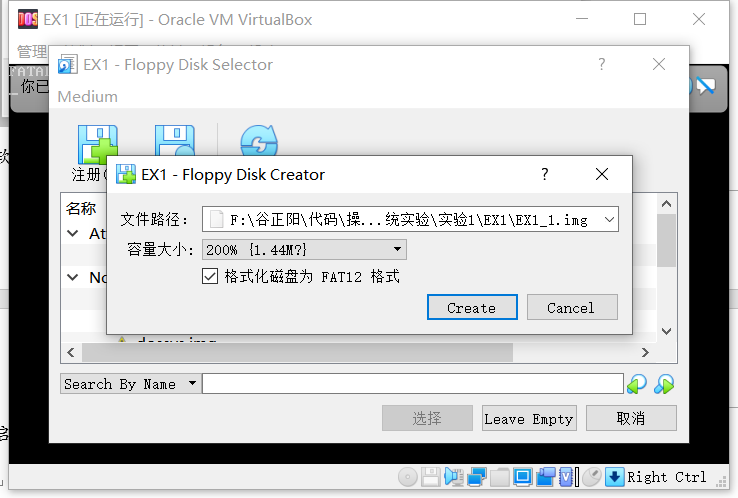




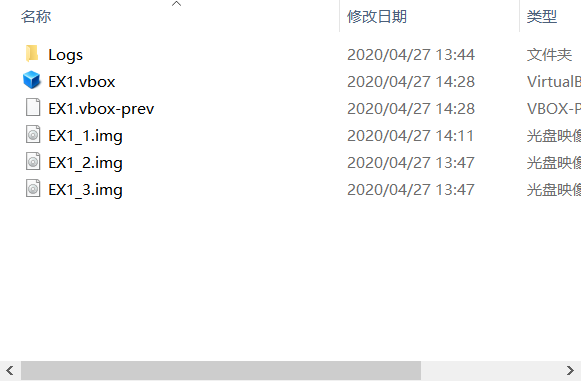


软盘创建：

启动→设备→创建软驱→创建



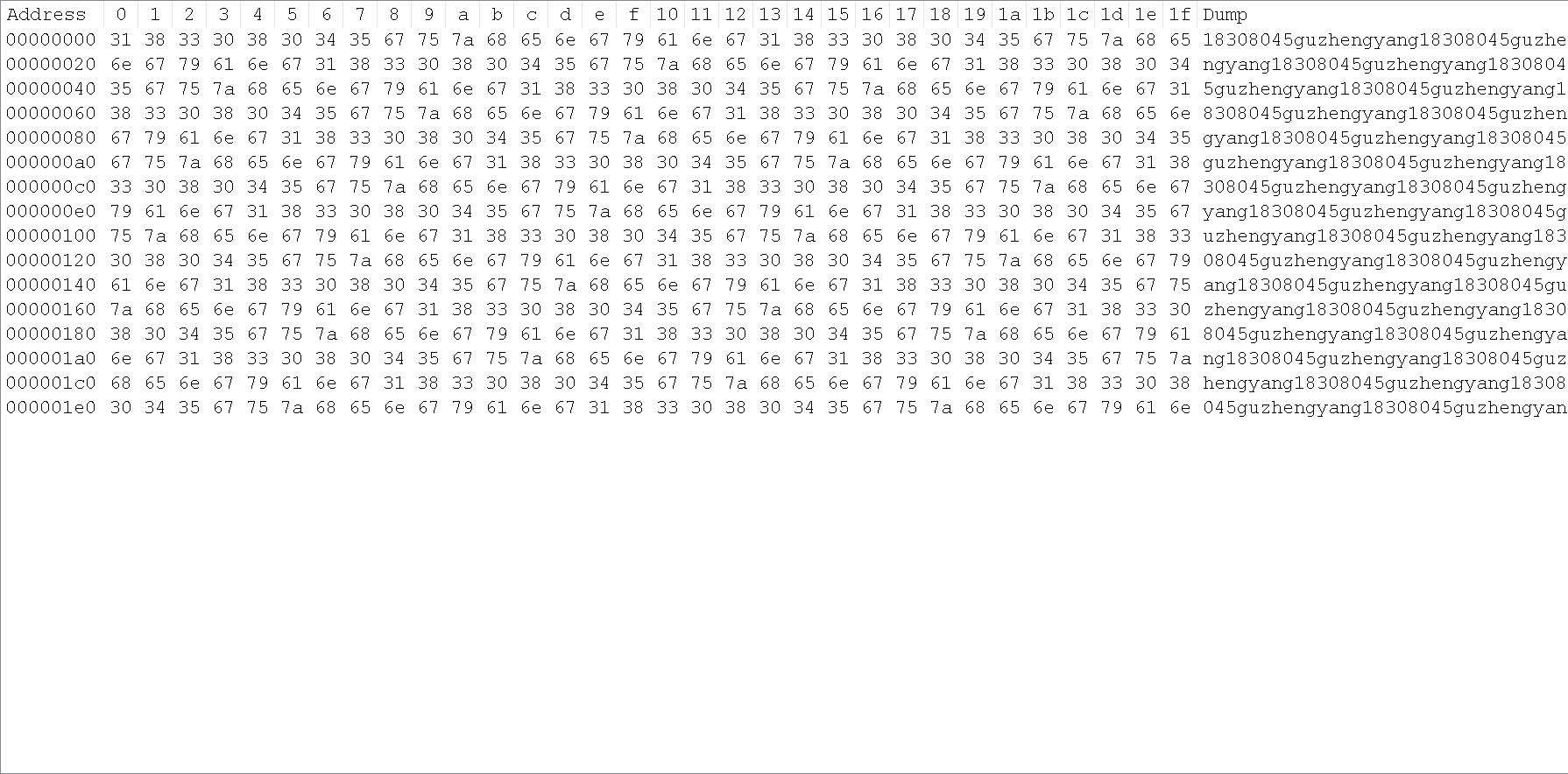
如上步骤共创建3个



(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>

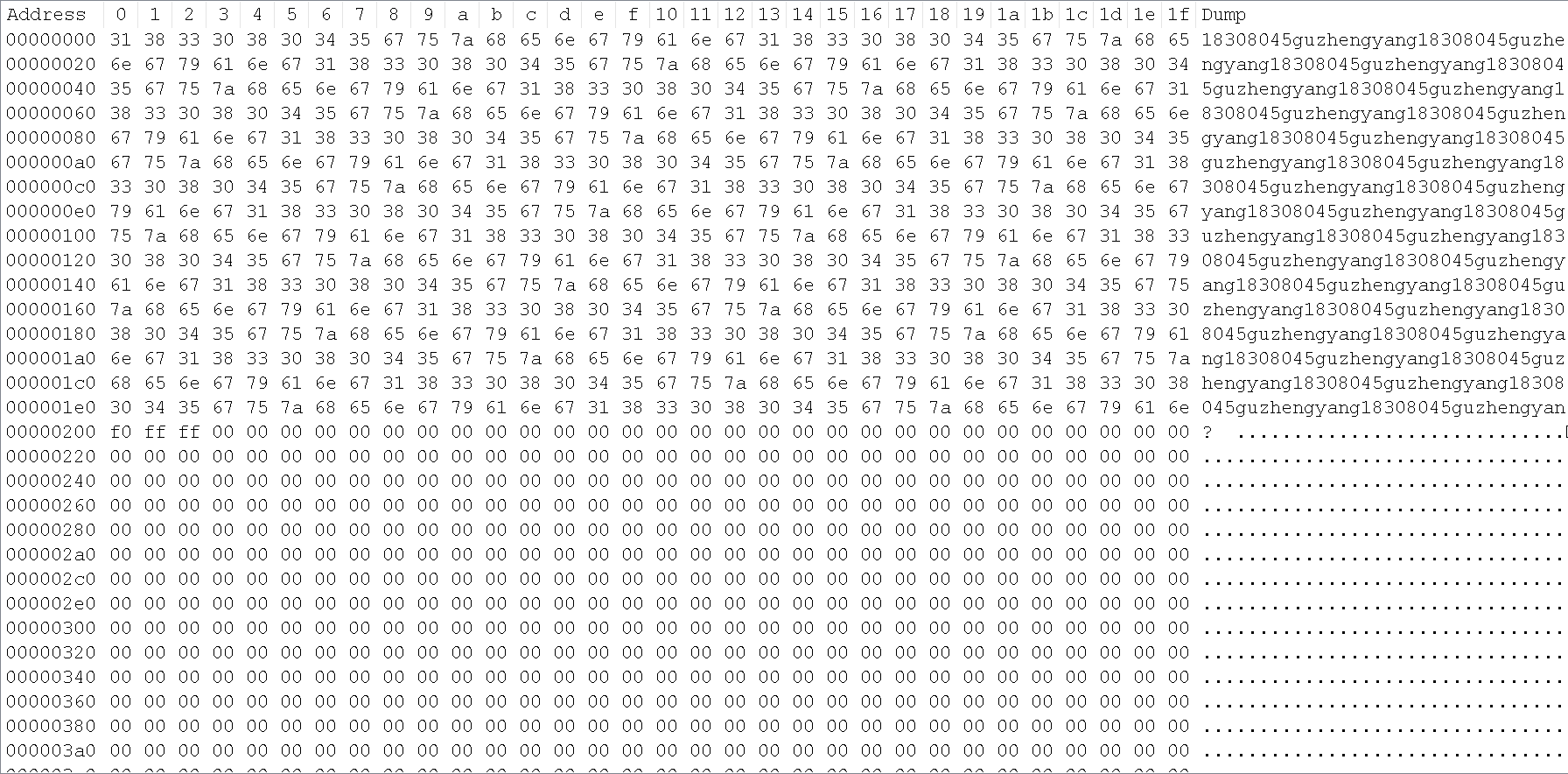
编写代码string.asm生成512B学号姓名



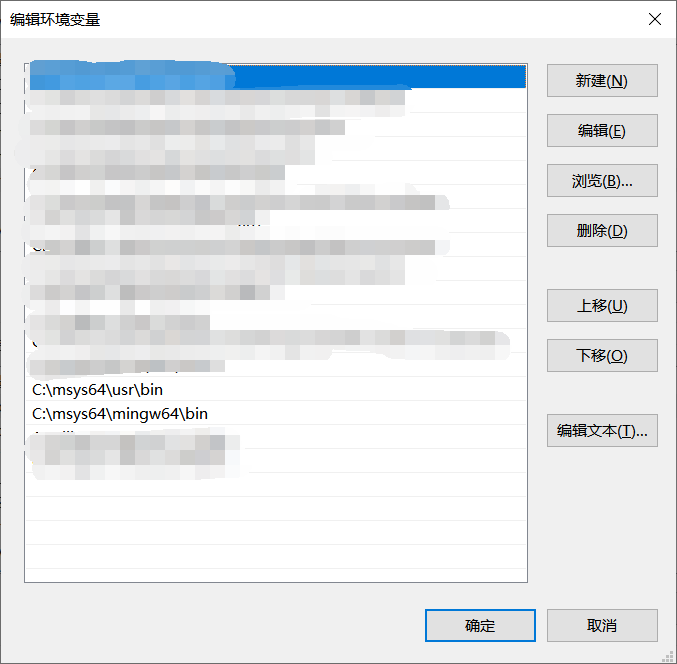
用fillout.cpp填写其至第一个软盘映像文件的首扇区



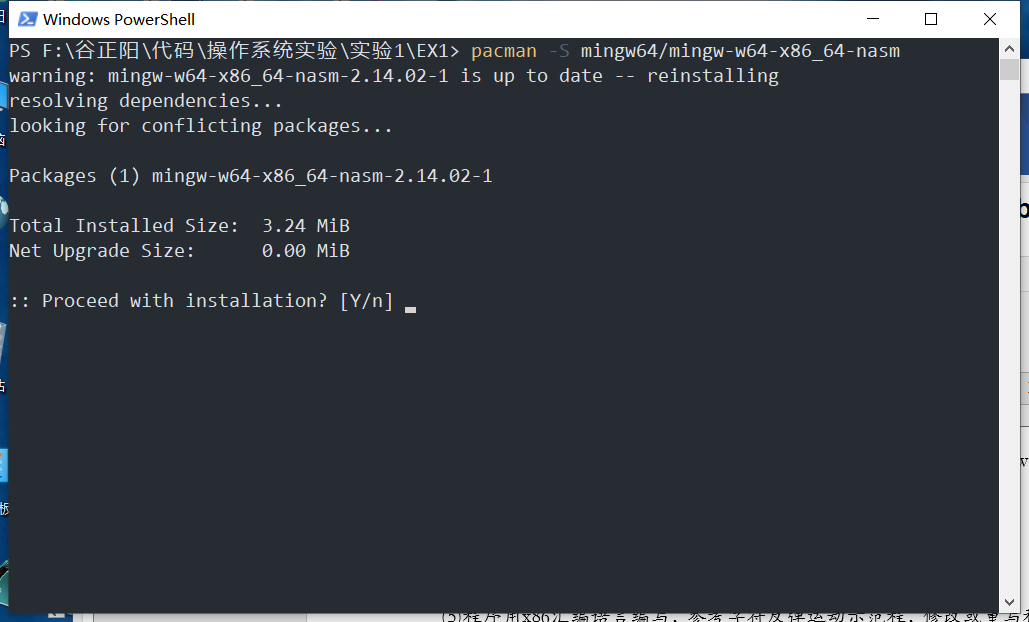
使用notepad++查看



(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



(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

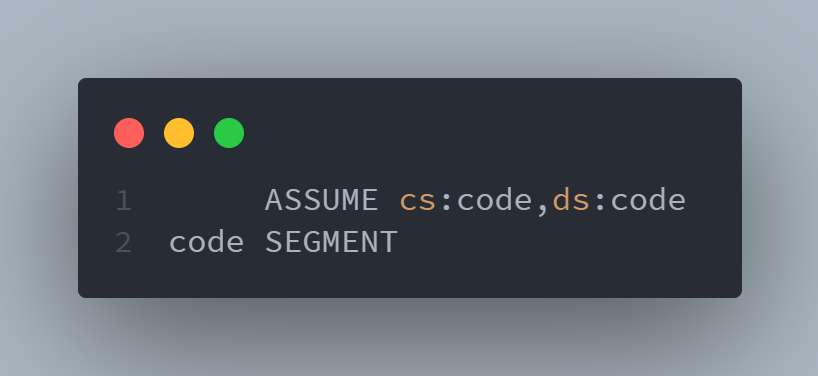




查询assume的用途：

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。



(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']

　　　　　　　　XXXXXXXXX

　　　　　　段名 ENDS

格式



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



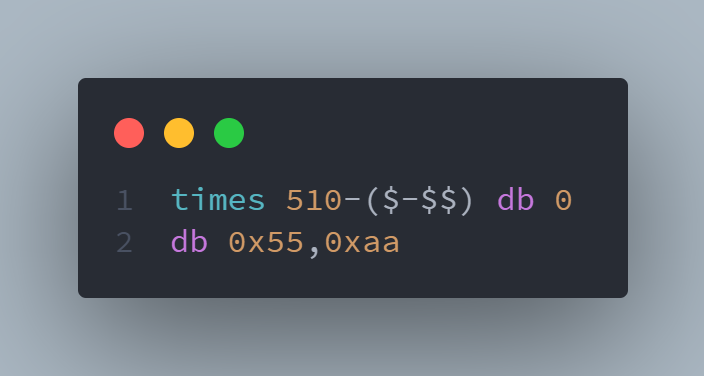
改为



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。

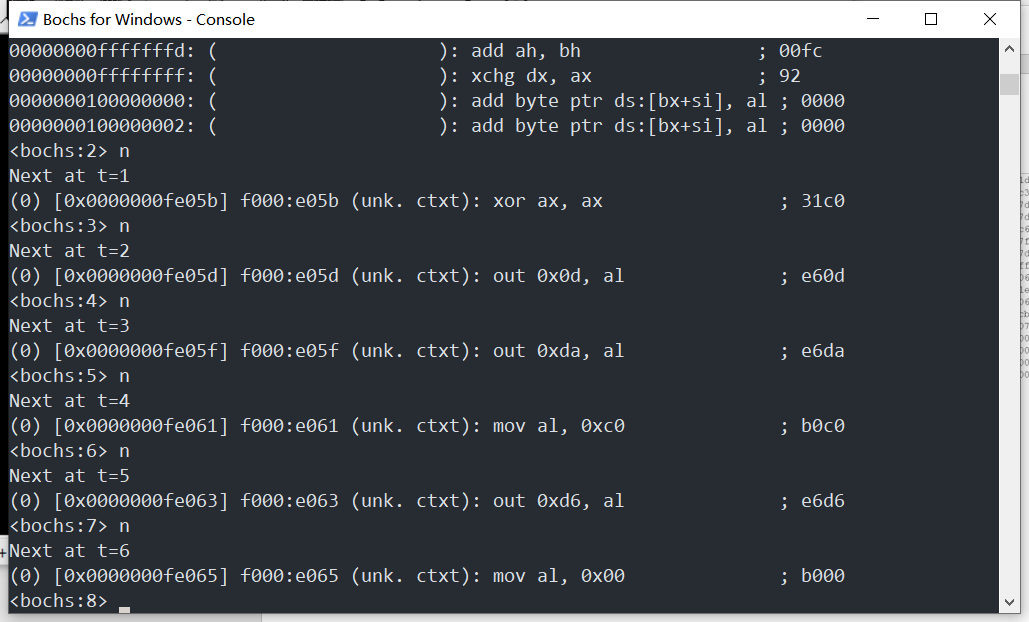


$表示当前指令地址，$$表示段首地址，$-$$表示前面指令共占的字节数，此语法表示填充0至510字节，然后填充0x55，0xaa。

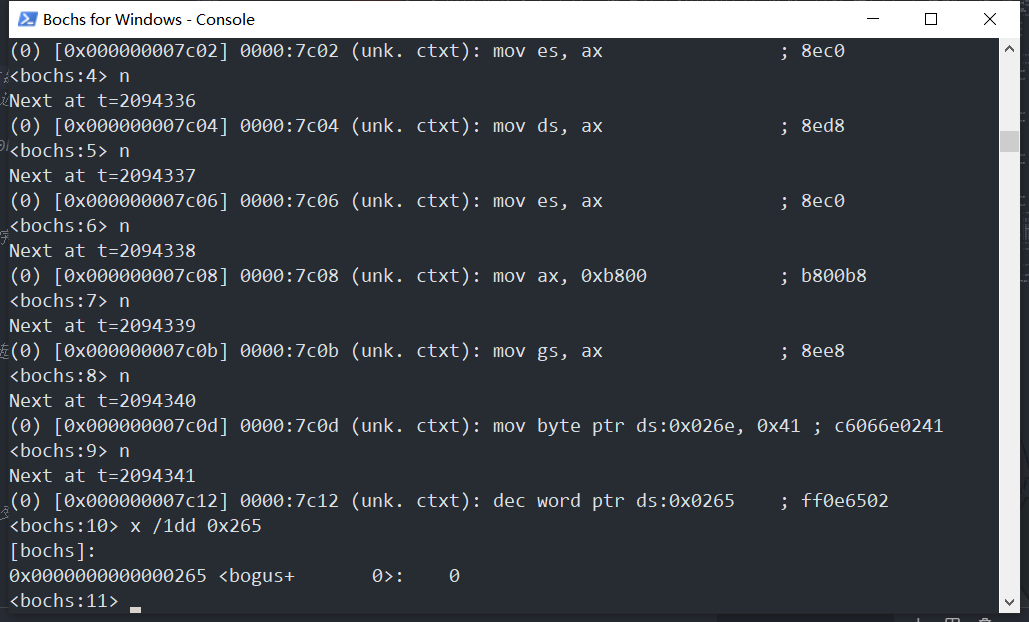
使用notepad++查看结果：



(6) 仍然无法运行，使用bochsdbg断点调试检查问题

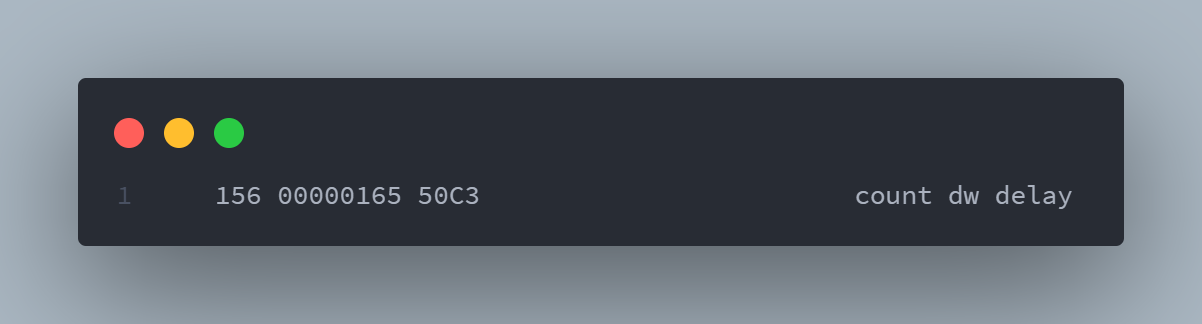


发现前几句都不是代码中的指令，经百度发现，是把mbr加载到物理内存0x7c00处，因此在此打断点，跳过去查看问题。

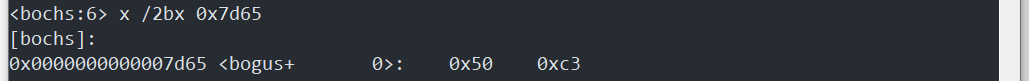


经检查发现如上位置本应该是50000，然而结果却是0。

观察lst文件



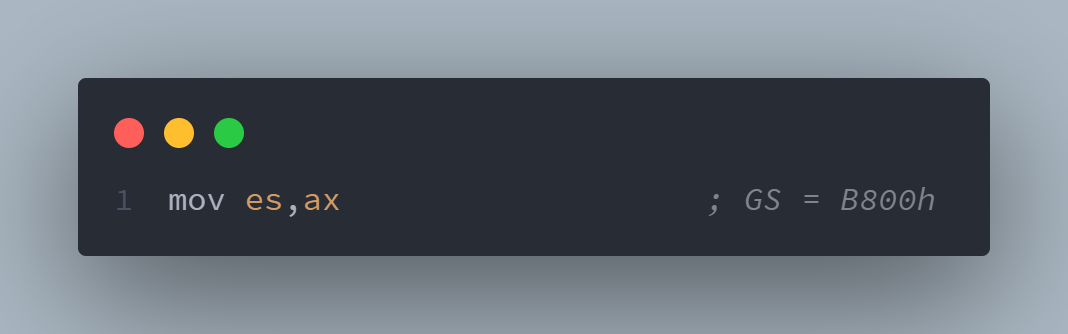
发现其正确位置应在0x7c00+0x156=0x7d56



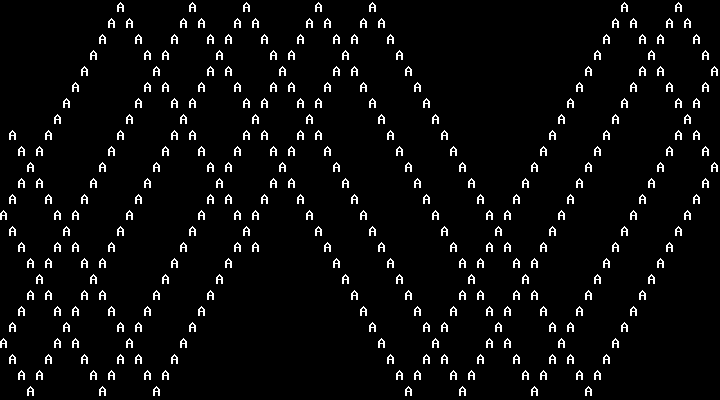
观察得出，物理内存0x7d65的值0xc350即为50000

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初始化未0xb800，因而将es初始化为0xb800



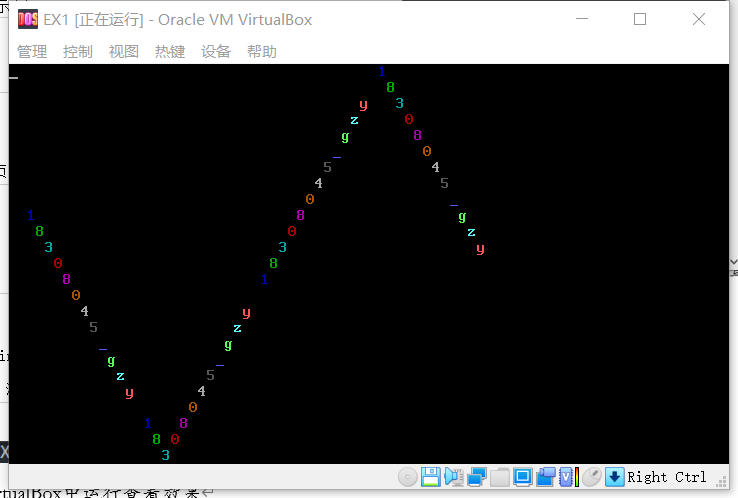
运行成功：



(6)使用fillout.cpp填写生成的mbr到第二个软盘映像中



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/guzhegnyang/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/guzhegnyang/operating_system_experiment.git>