## 1. Overview

Every operating system is responsible for enabling other programs to be run. However, who enables the operating system? This function is performed by a small program known as the bootloader. Although bootloaders take a different form with each computer architecture, the core ideas remain the same. In this project, you will create a bootloader for the x86 architecture.

In the x86 architecture, the bootup procedure looks for a bootloader on the first sector of the booting device (a USB flash disk, hard drive, image file, etc.) and is automatically loaded and executed during the system booting process. It is responsible for loading the rest of the operating system from the booting device into memory. The bootloader cannot rely upon any functionality from the operating system, such as OS calls for disk I/O, or linking an object file against static libraries. However, the x86 architecture requires the presence of the Basic Input Output System (BIOS), which provides minimal terminal, keyboard and disk support. The bootloader will use these BIOS calls to load the kernel from disk into memory, set up a stack space, and then switch control to the kernel. At this point, the OS begins running.

The first 512 bytes of a disk are known as the boot sector or Master Boot Record. The boot sector is an area of the disk reserved for booting purposes. If the boot sector of a disk contains a valid boot sector (the last word of the sector must contain the signature 0xAA55), then the disk is treated by the BIOS as bootable.

In the lab, we'll write a bootloader that displays "Hello, World!" in assembly language. Just for fun.

The bootloader:

- Stored in the Master Boot Record (MBR), on the first sector of the disk
- Size is exactly 1 sector, or 512 bytes
- Ends with a signature of `0x55` and `0xaa` at bytes 511 and 512
- Loaded by the BIOS POST interrupt `0x19` at address `0x7c00`
- Runs in 16 bits mode

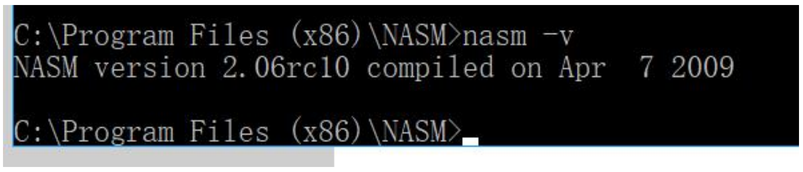
Keep in mind that we have no file system and no operating system. No DOS, no Windows, no Linux and none of the associated interrupts, libraries and routines.

## 2. The Lab Process

### 2.1 Install NASM

https://www.nasm.us/

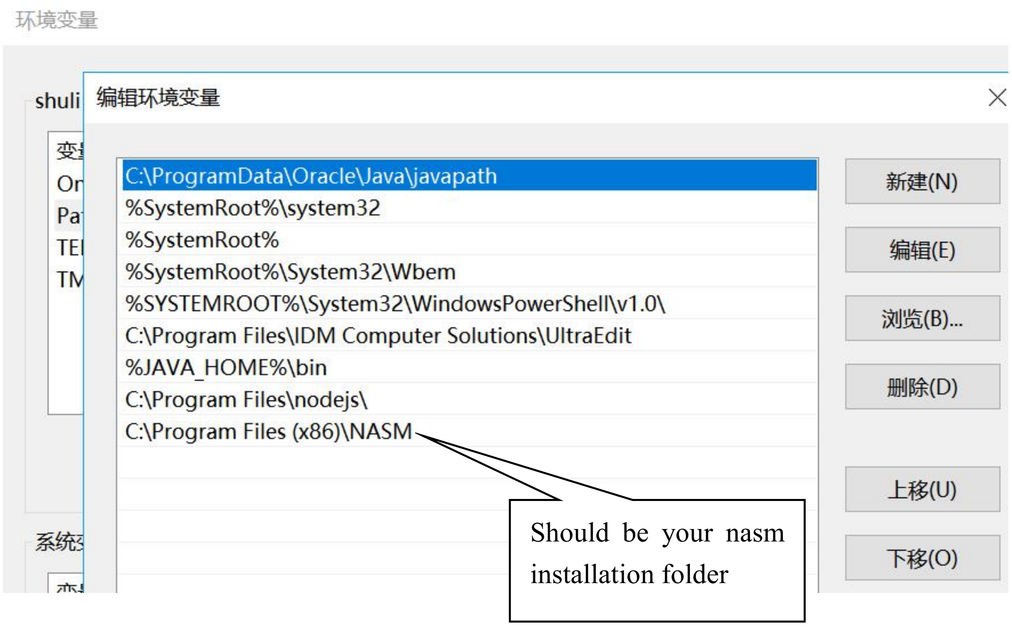
After installation, type “cmd” in the Windows search bar, then enter the following command:
**nasm -v**
This command will show nasm version you just have installed in your computer.



```
C:\Program Files (x86)\NASM>nasm -v
NASM version 2.06rc10 compiled on Apr  7 2009

C:\Program Files (x86)\NASM>
```

You may need to edit an **Environment Variable** in Windows to make this work.

Tip: In the System Properties window, click on the Advanced tab, then click the Environment Variables button near the bottom of that tab. In the Environment Variables window, highlight the Path variable in the "System variables" section and click the Edit button. Then add the nasm directory you just installed.

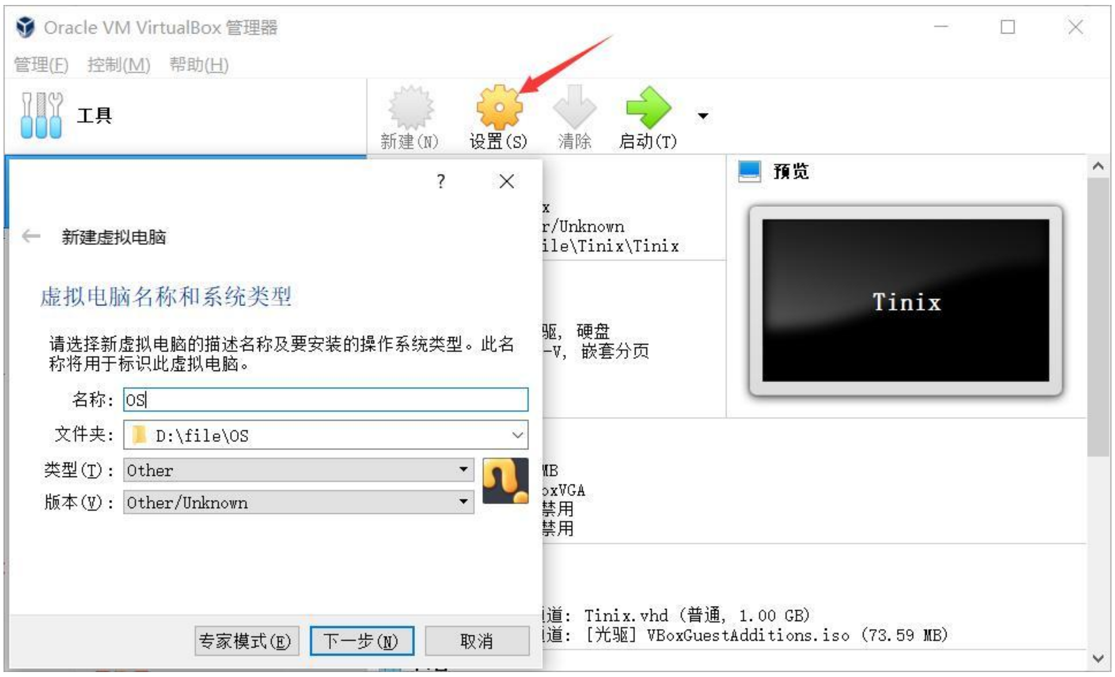


```
C:\ProgramData\Oracle\Java\javapath
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
C:\Program Files\IDM Computer Solutions\UltraEdit
%JAVA_HOME%\bin
C:\Program Files\nodejs\
C:\Program Files (x86)\NASM
```

### 2.2 Install VirtualBox

1) Download VirtualBox from official website https://www.virtualbox.org/wiki/Downloads and install.

2) Then, creating a New Virtual Machine in VirtualBox



新建虚拟电脑

虚拟电脑名称和系统类型

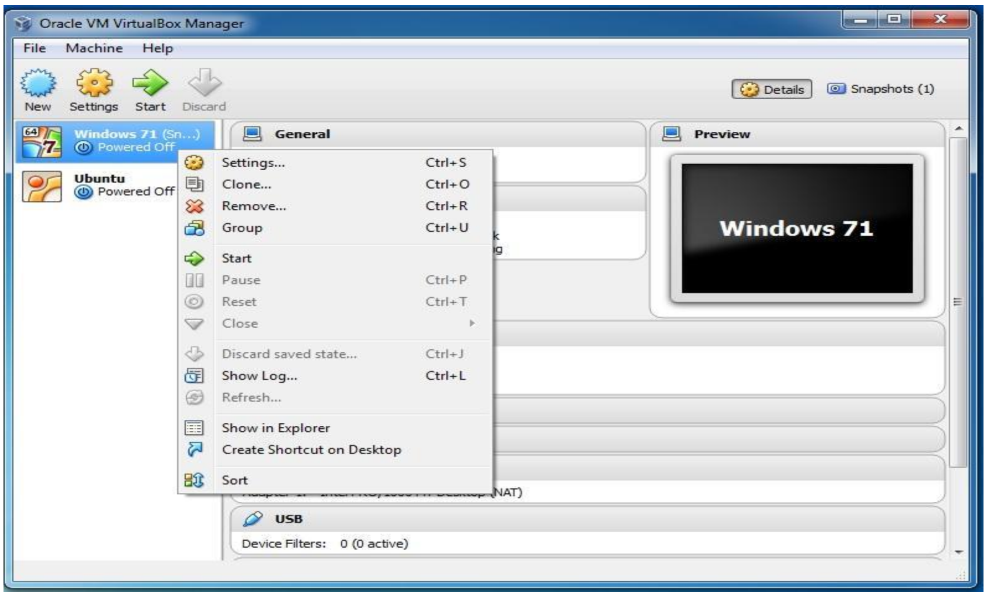
请选择新虚拟电脑的描述名称及要安装的操作系统类型。此名称将用于标识此虚拟电脑。

名称: OS
文件夹: D:\file\OS
类型(T): Other
版本(V): Other/Unknown

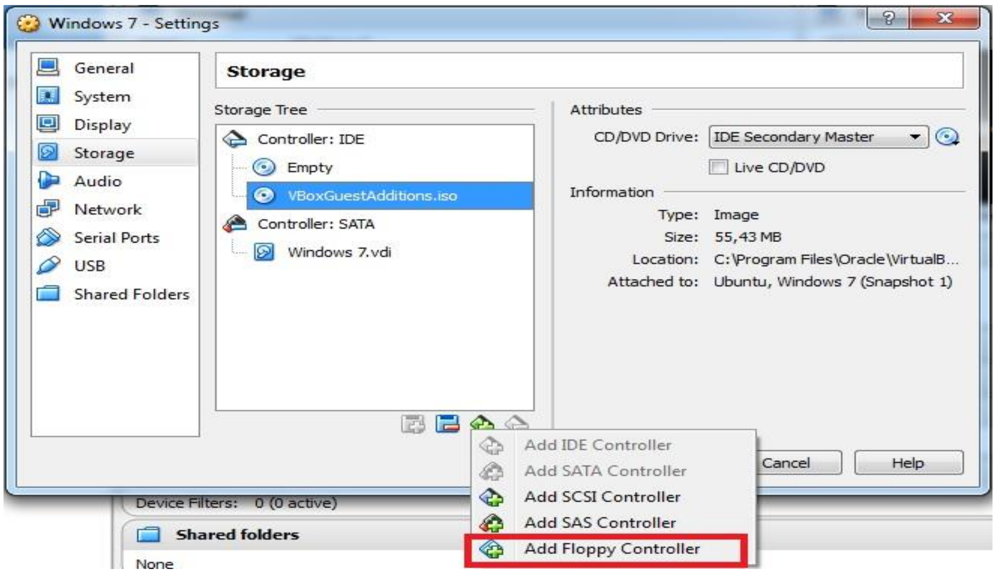
https://docs.oracle.com/cd/E26217_01/E26796/html/qs-create-vm.html
Check the above link to get help on how to create a new virtual machine in VirtulBox.

3) Create a virtual floppy disk by VirtualBox
In Oracle VirtualBox, it is possible to add a floppy drive to a virtual machine. Virtual floppy drives, like virtual CD/DVD drives, can be connected to either a host floppy drive (if you have one) or a disk image, which must be in RAW format.

Here are the steps to add a floppy drive to a virtual machine in Oracle VirtualBox:
Select the virtual machine from the Oracle VM **VirtualBox Manager** and choose **Settings**:



Click **Storage > Add CD/DVD Device**:



Attributes

CD/DVD Drive: IDE Secondary Master

Live CD/DVD

Information

Type: Image

Size: 55,43 MB

Location: C:\Program Files\Oracle\VirtualB...

Attached to: Ubuntu, Windows 7 (Snapshot 1)

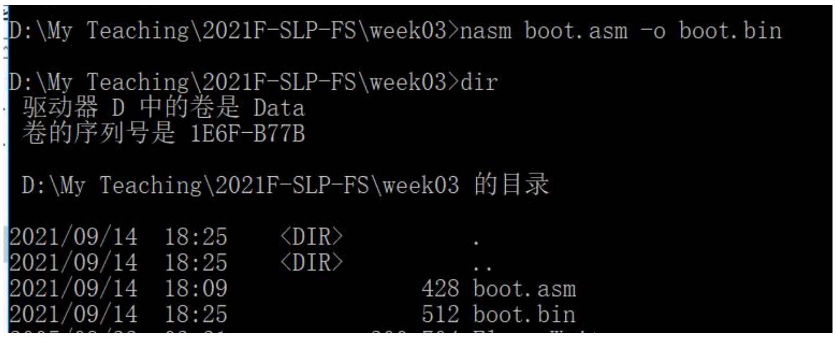
Click **Add Floppy Controller** and press **OK** to save the changes.

## 2.3 Write code for BootLoader

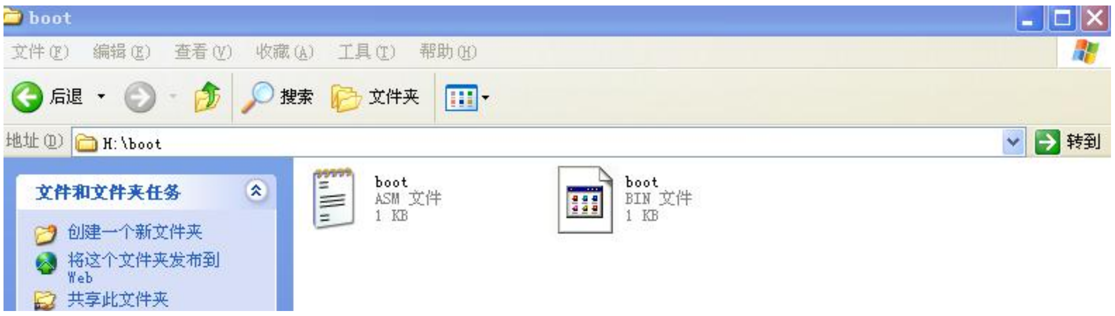
```
-------------------------------------boot.asm-----------------------------------------------------

      org 07c00h              ;
      mov ax, cs
      mov ds, ax
      mov es, ax
      call DispStr            ;
      jmp $                   ;

DispStr:
      mov ax, BootMessage
      mov bp, ax              ;
      mov cx, 16              ;
      mov ax, 01301h          ;
      mov bx, 000ch           ;
      mov dl, 0
      int 10h                 ;
      ret
BootMessage:      db "Hello, OS world!"
times 510-($-$$) db 0     ;
                              ;
dw 0xaa55

---------------------------------boot.asm--------------------------------------------------------
```

1) Compile your “boot.asm” by NASM assembler. You need to install the NASM first. Open the nasm, compile the boot.asm file to binary file boot.bin which size of 512 bits by the following command:

**nasm boot.asm –o boot.bin**



```
D:\My Teaching\2021F-SLP-FS\week03>nasm boot.asm -o boot.bin

D:\My Teaching\2021F-SLP-FS\week03>dir
 驱动器 D 中的卷是 Data
 卷的序列号是 1E6F-B77B

 D:\My Teaching\2021F-SLP-FS\week03 的目录

2021/09/14  18:25    <DIR>          .
2021/09/14  18:25    <DIR>          ..
2021/09/14  18:09               428 boot.asm
2021/09/14  18:25               512 boot.bin
```



boot

文件(F) 编辑(E) 查看(V) 收藏(A) 工具(T) 帮助(H)

后退 搜索 文件夹

地址(D) H:\boot 转到

文件和文件夹任务

创建一个新文件夹

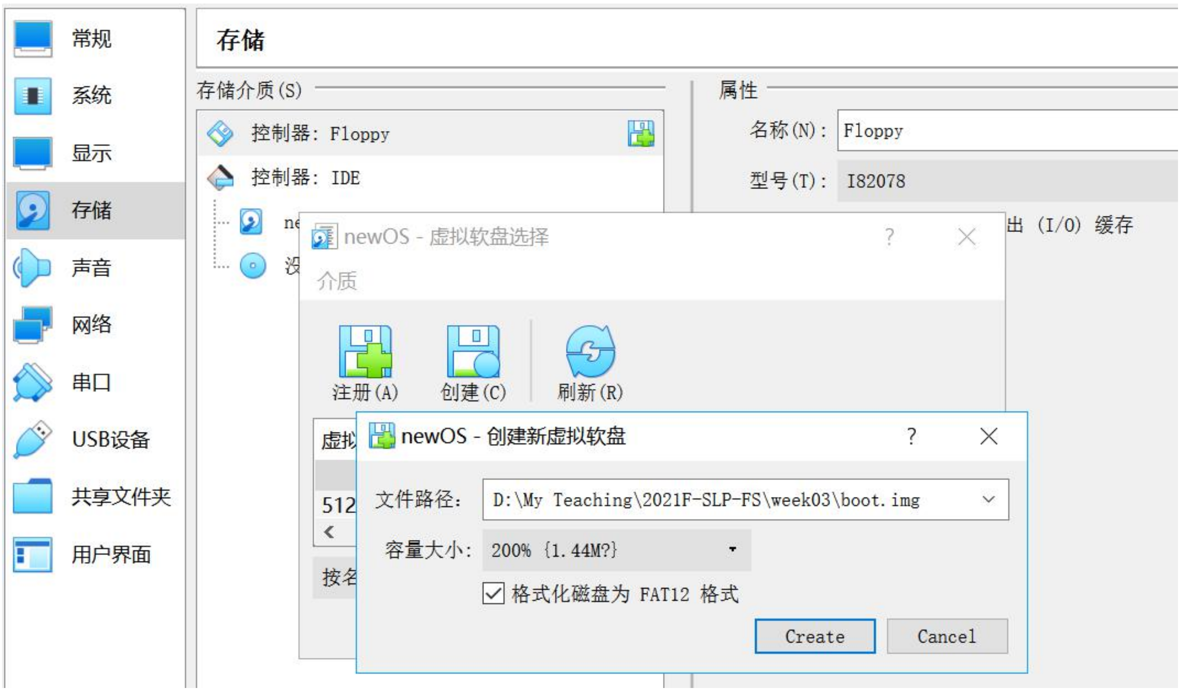
将这个文件夹发布到 Web

共享此文件夹

boot ASM 文件 1 KB

boot BIN 文件 1 KB

2) Create a boot.img file under the virtual floppy disk



常规
系统
显示
存储
声音
网络
串口
USB设备
共享文件夹
用户界面

存储

存储介质(S)

控制器: Floppy

控制器: IDE

属性

名称(N): Floppy

型号(T): I82078

newOS - 虚拟软盘选择

介质

注册(A) 创建(C) 刷新(R)

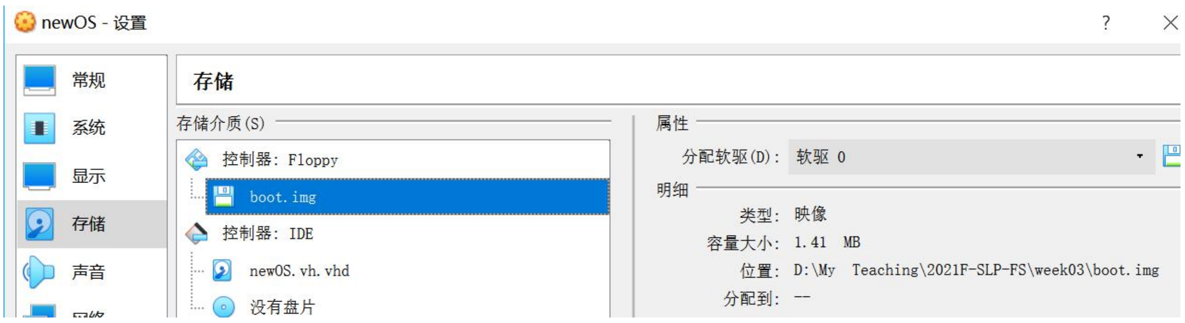
newOS - 创建新虚拟软盘

文件路径: D:\My Teaching\2021F-SLP-FS\week03\boot.img

容量大小: 200% {1.44M?}

☑ 格式化磁盘为 FAT12 格式

Create Cancel


newOS - 设置

常规
系统
显示
存储
声音

存储

存储介质(S)

控制器: Floppy
boot.img
控制器: IDE
newOS.vh.vhd
没有盘片

属性

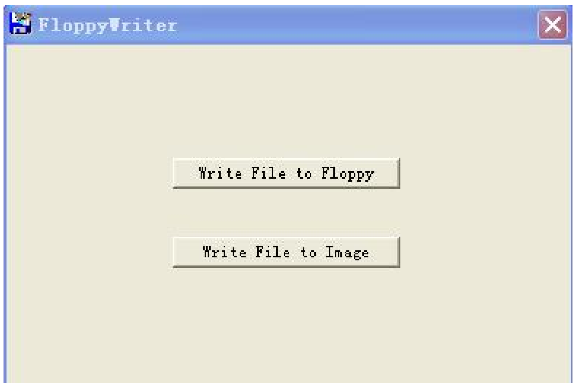
分配软驱(D): 软驱 0

明细

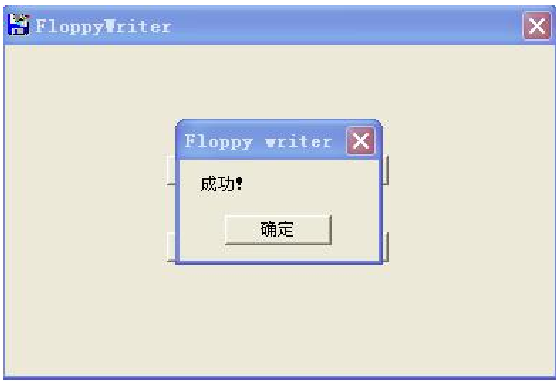
类型: 映像
容量大小: 1.41 MB
位置: D:\My Teaching\2021F-SLP-FS\week03\boot.img
分配到: --

3) Write the BootLoader boot.bin to virtual floppy disk of virtual machine use tool **FloppyWriter**.    Remember it should be done in the first sector of floppy disk.

Open FloppyWriter,



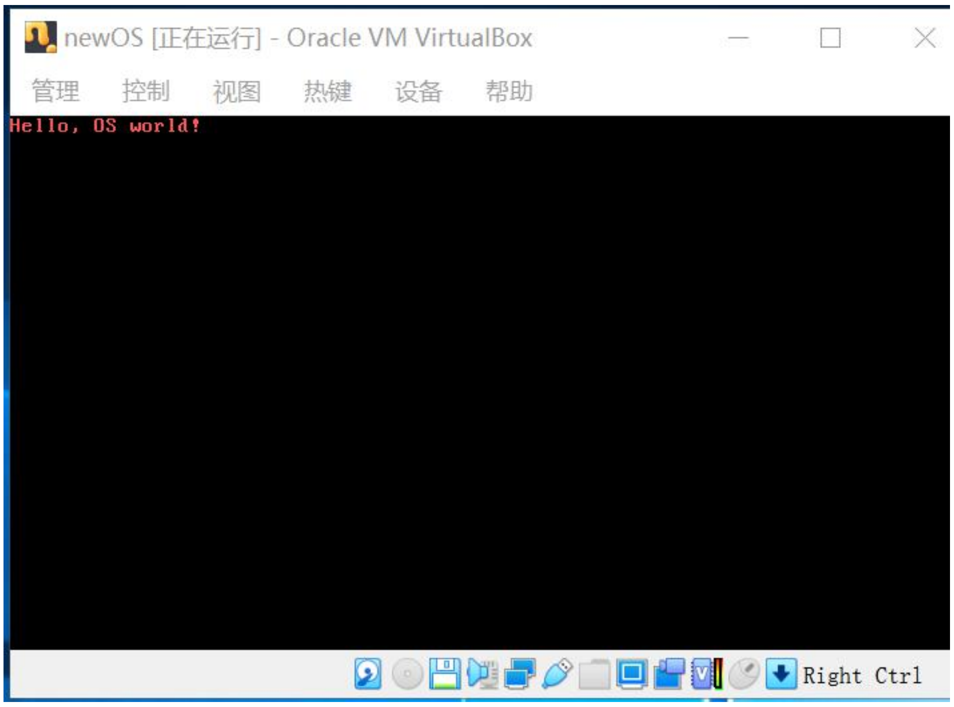
choose“Write File to Image”, select the “boot.bin”, click“Open”, then select boot.img we just create. Then the boot.bin is written to first sector of virtual floppy boot.img. If success, the the following information box will rise up.



成功!

## 2.4 Start up virtual machine with homemade BootLoader

In VirtualBox Workstation, click “Power on your virtual machine”, you should see the “hello” message print out to screen.


newOS [正在运行] - Oracle VM VirtualBox
管理 控制 视图 热键 设备 帮助
Right Ctrl

```
Hello, OS world!
```

## 3. Your task

3.1 Study boot.asm source code, write comments for each line of code.
3.2 Change the source code of this assembly program. The new program should show your student id and name on the screen. Show me the source code and the screen shot of testing.