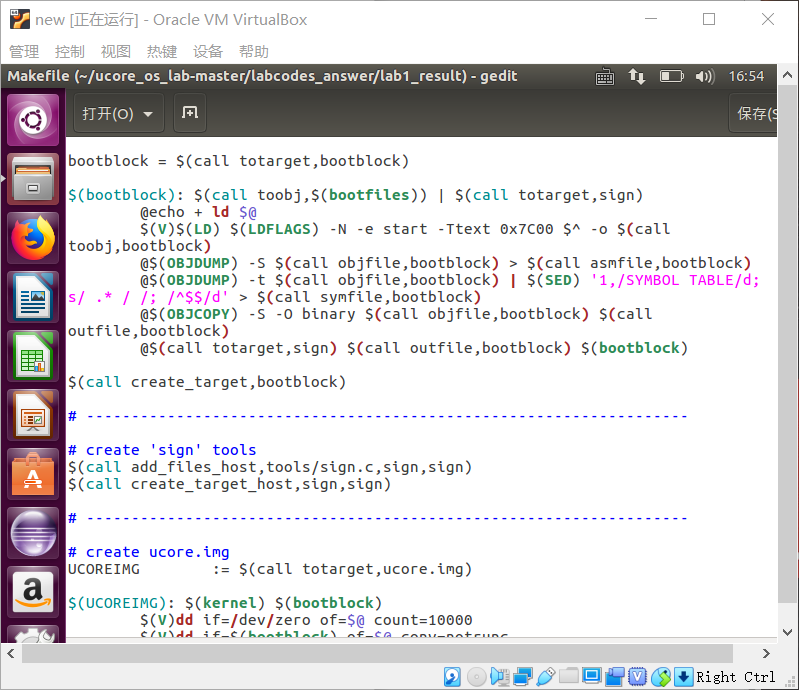
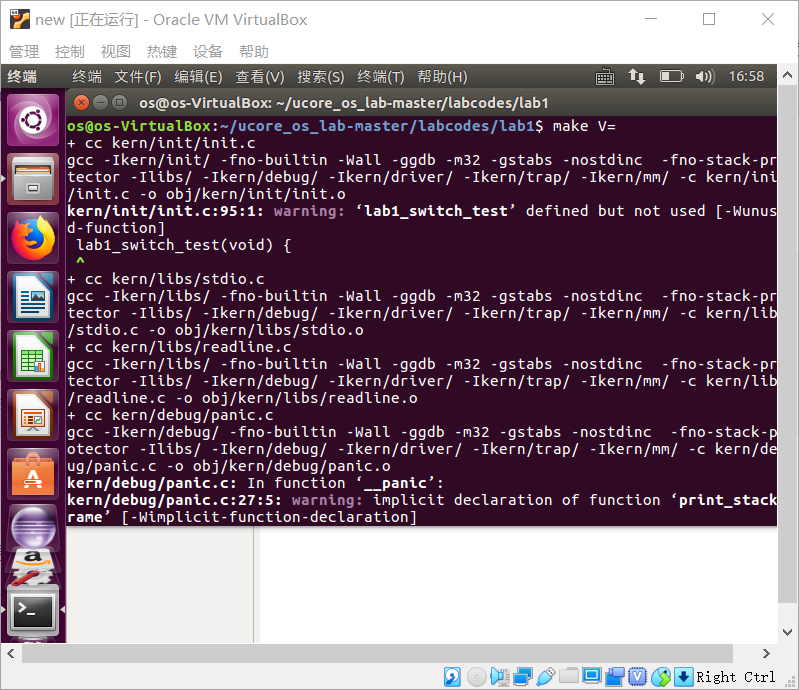
练习1：理解通过make生成执行文件的过程。

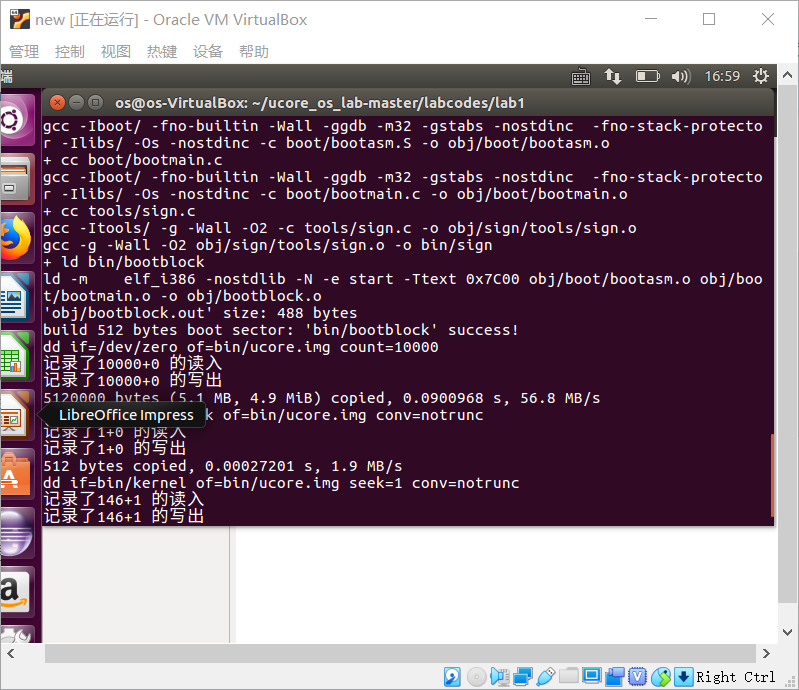
1. 操作系统镜像文件ucore.img是如何一步一步生成的？(需要比较详细地解释Makefile中每一条相关命令和命令参数的含 义，以及说明命令导致的结果)

2. 一个被系统认为是符合规范的硬盘主引导扇区的特征是什么？

（1）：







+ cc kern/init/init.c //编译init.c

+ cc kern/libs/readline.c //编译readline.c

+ cc kern/libs/stdio.c //编译stdlio.c

+ cc kern/debug/kdebug.c //编译kdebug.c

+ cc kern/debug/kmonitor.c //编译komnitor.c

+ cc kern/debug/panic.c //编译panic.c

+ cc kern/driver/clock.c //编译clock.c

+ cc kern/driver/console.c //编译console.c

+ cc kern/driver/intr.c //编译intr.c

+ cc kern/driver/picirq.c //编译prcirq.c

+ cc kern/trap/trap.c //编译trap.c

+ cc kern/trap/trapentry.S //编译trapentry.S

+ cc kern/trap/vectors.S //编译vectors.S

+ cc kern/mm/pmm.c //编译pmm.c

+ cc libs/printfmt.c //编译printfmt.c

+ cc libs/string.c //编译string.c

+ ld bin/kernel //链接成kernel

+ cc boot/bootasm.S //编译bootasm.c

+ cc boot/bootmain.c //编译bootmain.c

+ cc tools/sign.c //编译sign.c

+ ld bin/bootblock //根据sign规范生成bootblock

dd if=/dev/zero of=bin/ucore.img count=10000

//创建大小为10000个块的ucore.img，初始化为0，每个块为512字节

dd if=bin/bootblock of=bin/ucore.img conv=notrunc

//把bootblock中的内容写到第一个块

dd if=bin/kernel of=bin/ucore.img seek=1 conv=notrun

//从第二个块开始写kernel中的内容

（2）：一个被系统认为是符合规范的硬盘主引导扇区的特征是什么？

1 大小为512字节

2 多余的空间填0

3 第510个（倒数第二个）字节是0x55，

4 第511个（倒数第一个）字节是0xAA。

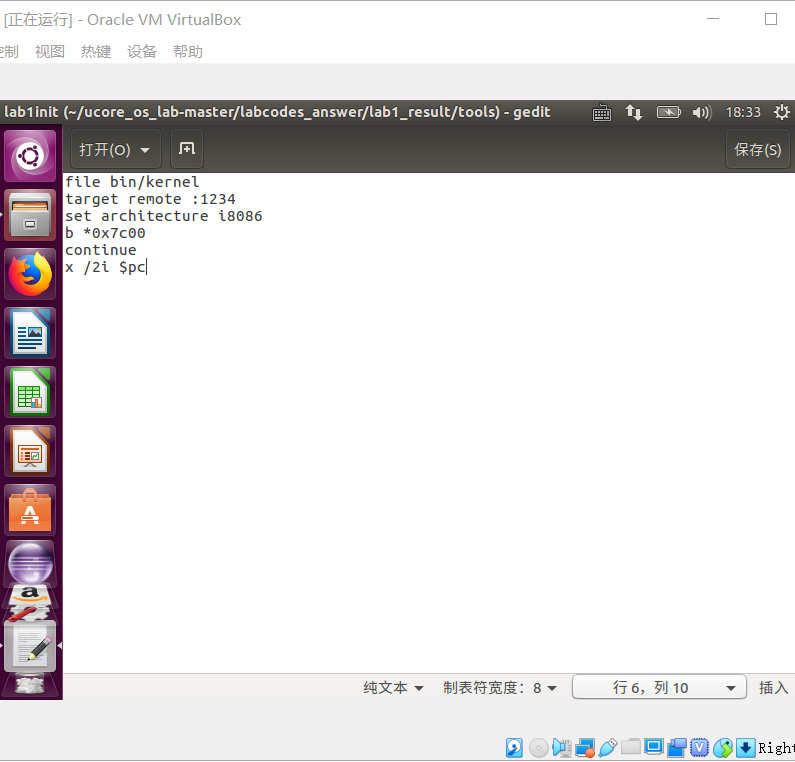
练习2：使用qemu执行并调试lab1中的软件。

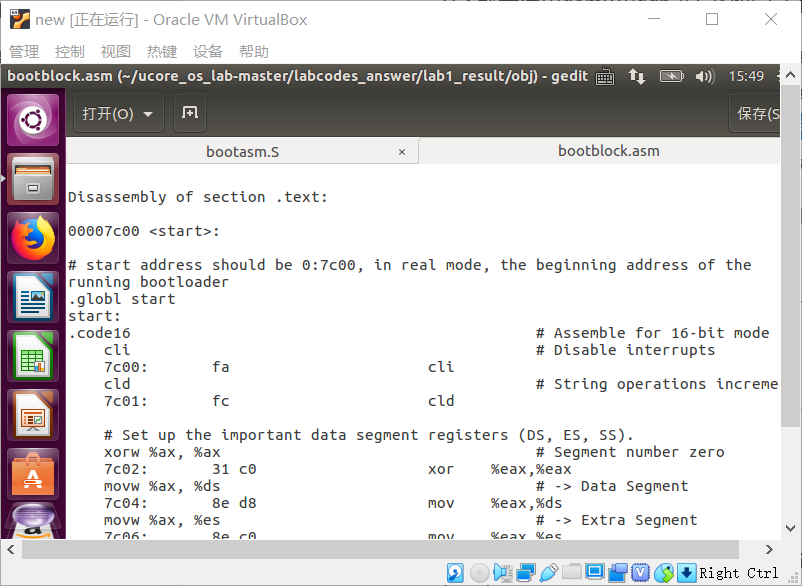
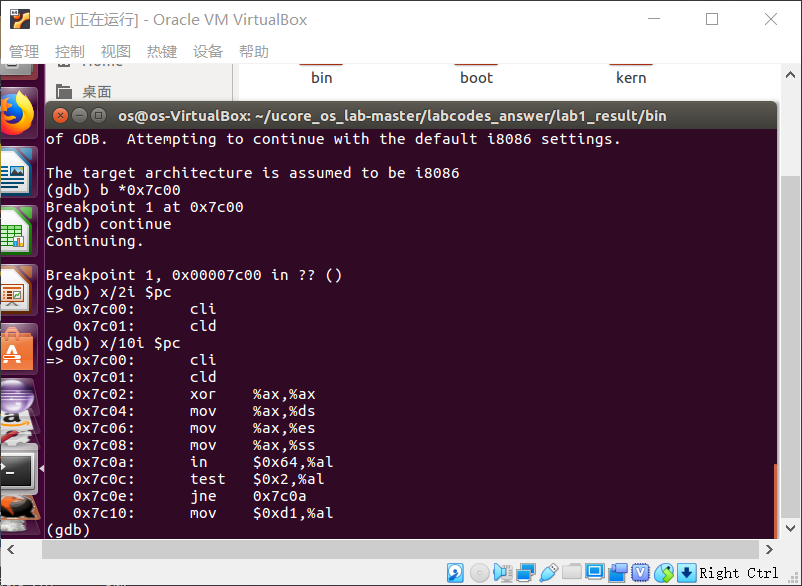
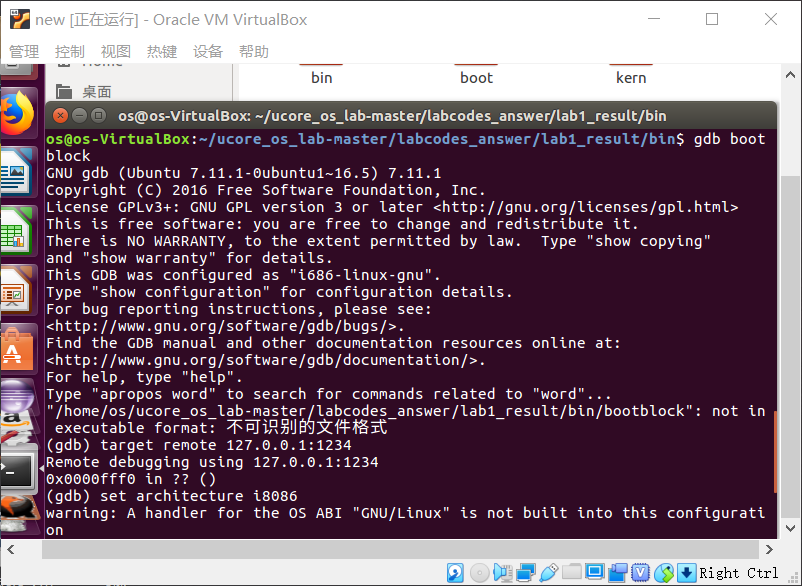
1. 从CPU加电后执行的第一条指令开始，单步跟踪BIOS的执行。

2. 在初始化位置0x7c00设置实地址断点,测试断点正常。

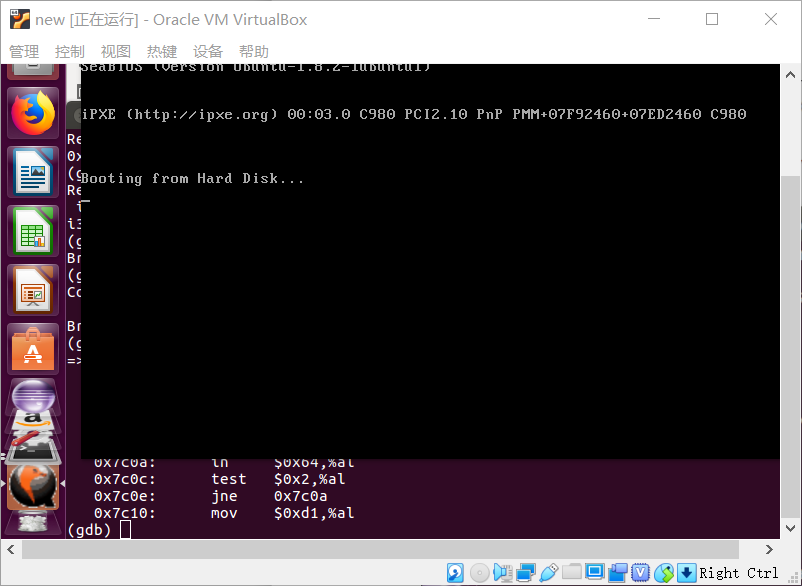
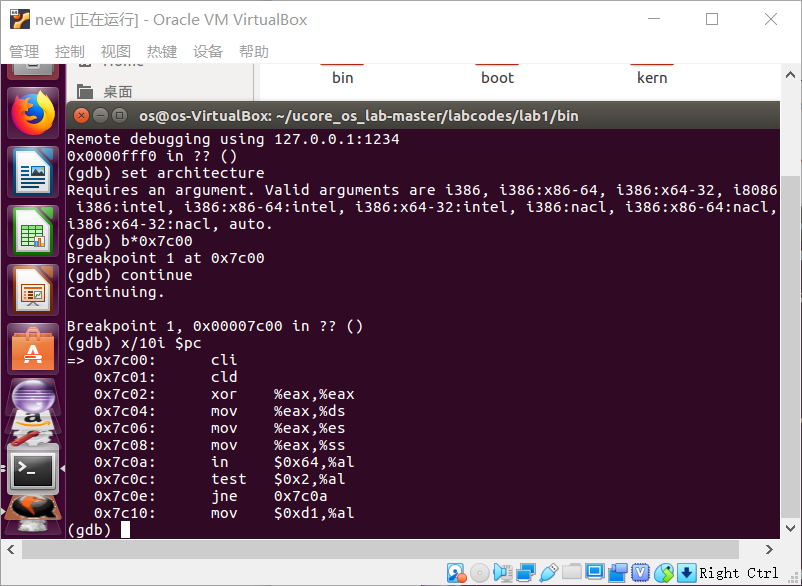
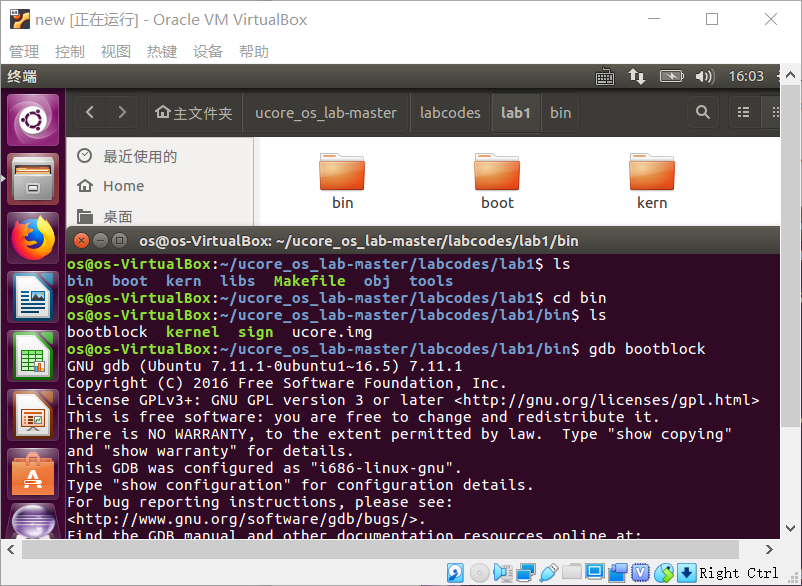
3. 从0x7c00开始跟踪代码运行,将单步跟踪反汇编得到的代码与bootasm.S和 bootblock.asm进行比较。

4. 自己找一个bootloader或内核中的代码位置，设置断点并进行测试。





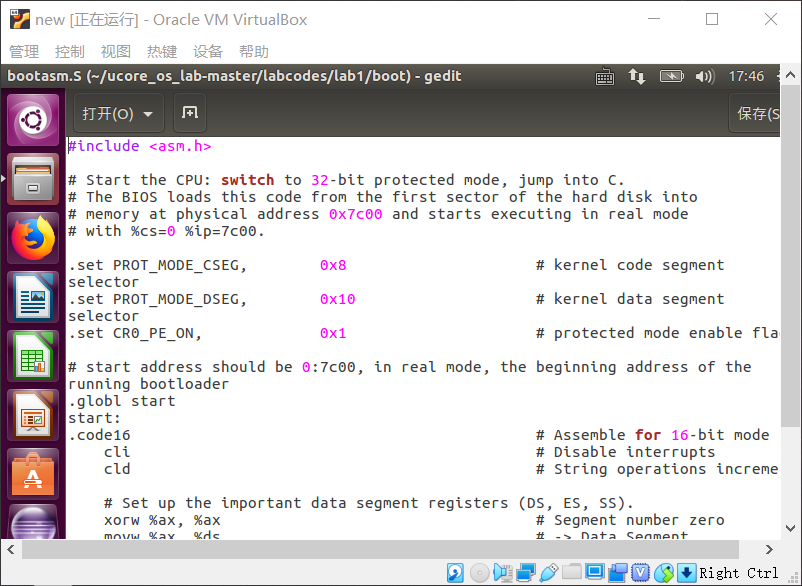
bootasm.S和 bootblock.asm基本一致



练习3：分析bootloader进入保护模式的过程。

1、BIOS将通过读取硬盘主引导扇区到内存，并转跳到对应内存中的位置执行bootloader。请分析bootloader是如何完成从实模 式进入保护模式的。

（1）：关中断和清除数据段寄存器



2、为何开启A20，以及如何开启A20

（2）：

初始时A20为0，访问超过1MB的地址时，就会从0循环计数，将A20地址线置为1之后，才可以访问4G内存。A20地址位由8042控制，8042有2个有两个I/O端口：0x60和0x64。

打开流程：

等待8042 Input buffer为空；

发送Write 8042 Output Port （P2）命令到8042 Input buffer；

等待8042 Input buffer为空；

将8042 Output Port（P2）得到字节的第2位置1，然后写入8042 Input buffer；

3、如何初始化GDT表

（3）：

1）：载入GDT表；

2）：进入保护模式；

3）：通过长转跳更新cs的基地址

4）：设置段寄存器，并建立堆栈

5）：转跳到保护模式，通过call bootmain进入BOOT主方法

4.、如何使能和进入保护模式

（4）：将cr0寄存器置1