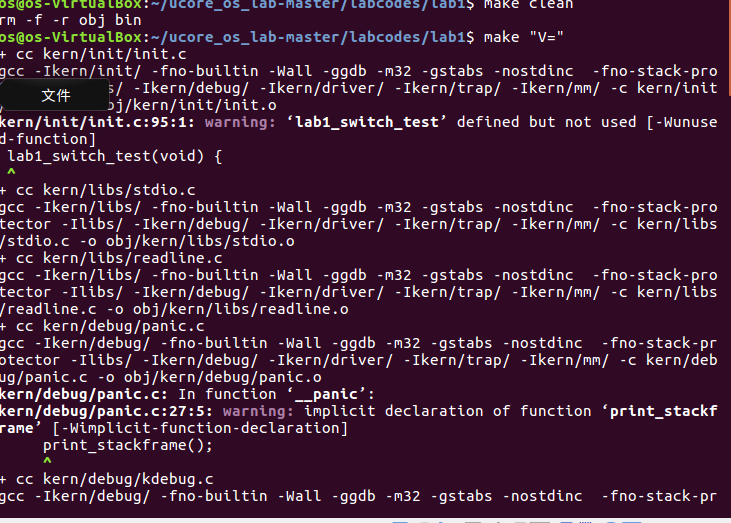
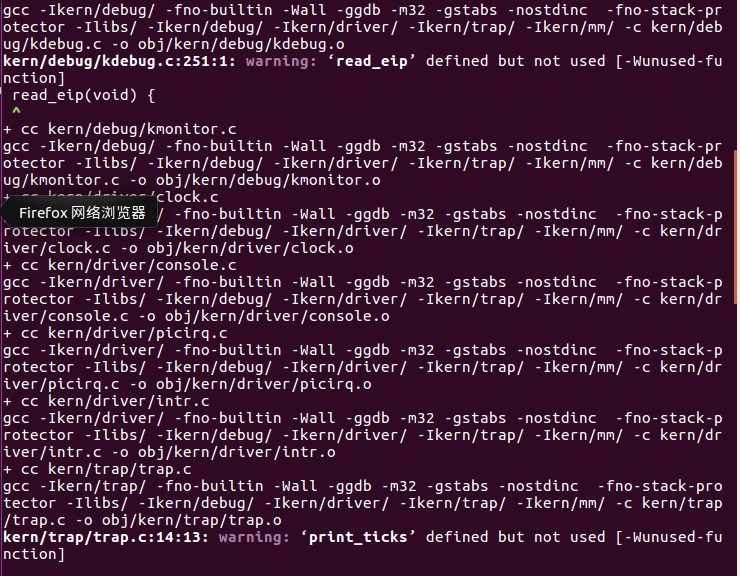
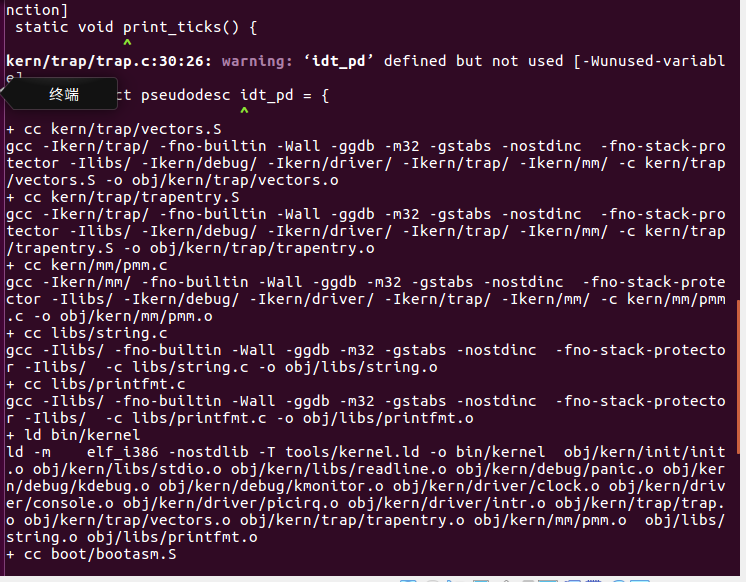
练习**1**：理解通过**make**生成执行文件的过程。（要求在报告中写出对下述问题的回答）

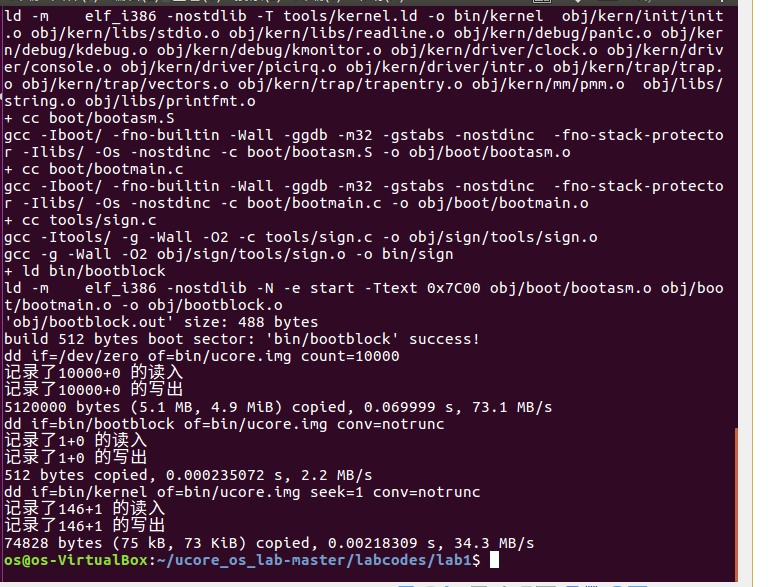
1. 操作系统镜像文件ucore.img是如何一步一步生成的？(需要比较详细地解释Makefile中每一条相关命令和命令参数的含

义，以及说明命令导致的结果)









1～138行：定义各种变量/函数，设置参数，进行准备工作。

140~153行：生成bin/kernel

155~170行：生成bin/bootblock

172~176行：生成bin/sign

178~188行：生成bin/ucore.img

189~201行：收尾工作/定义变量

203~269行：定义各种make目标

练习**2**：使用**qemu**执行并调试**lab1**中的软件。（要求在报告中简要写出练习过程）

为了熟悉使用qemu和gdb进行的调试工作，我们进行如下的小练习：

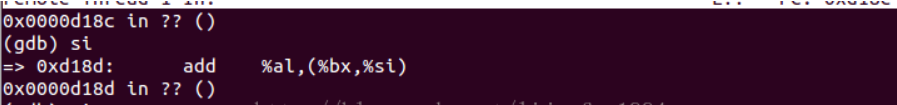
1. 从CPU加电后执行的第一条指令开始，单步跟踪BIOS的执行。

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





在初始化位置 0x7c00 设置实地址断点,测试断点正常。



在 bootloader 或内核代码中任意找一处代码,设置断点并进行测试。

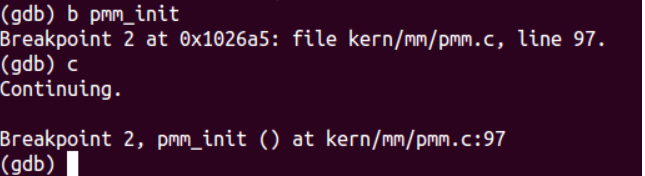
程序进入init.c中的kern\_init()

b pmm\_init设置断点

这是初始化gdt的函数。

c执行

遇到断点，在pmm.c中的pmm\_init处停止



练习**3**：分析**bootloader**进入保护模式的过程。（要求在报告中写出分析）

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。

提示：需要阅读小节**“**保护模式和分段机制**”**和lab1/boot/bootasm.S源码，了解如何从实模式切换到保护模式，需要了解：

为何开启A20，以及如何开启A20

如何初始化GDT表

如何使能和进入保护模式

bootbloader首先屏蔽所有中断,

之后将段寄存器清零,

打开A20地址线,

加载GDT的基地址,

切换到保护模式,

跳转到32位代码. 在32位代码中, bootloader重新设置保护模式下的段寄存器, 然后设置栈顶指针, 之后跳转到C代码。