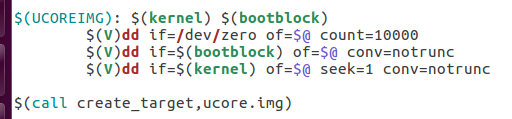
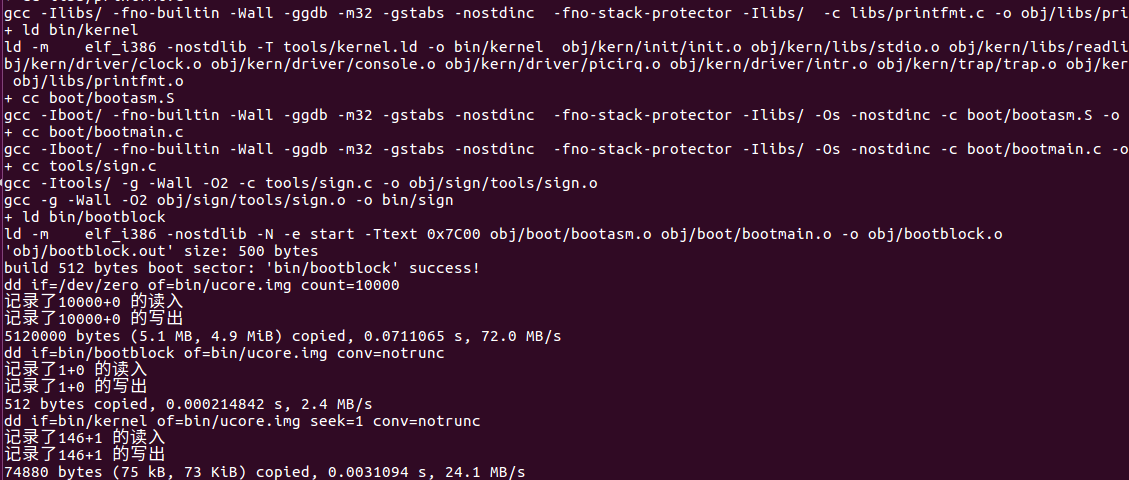
练习1：

打开makefile



生成ucore.img需要kernel和bootblock

输入make V=

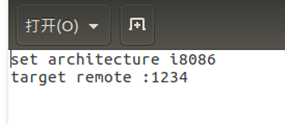


通过GCC编译kernel和bootblock相关的代码生成.o文件。在生成kernel和bootblock后就可以生成ucore.img

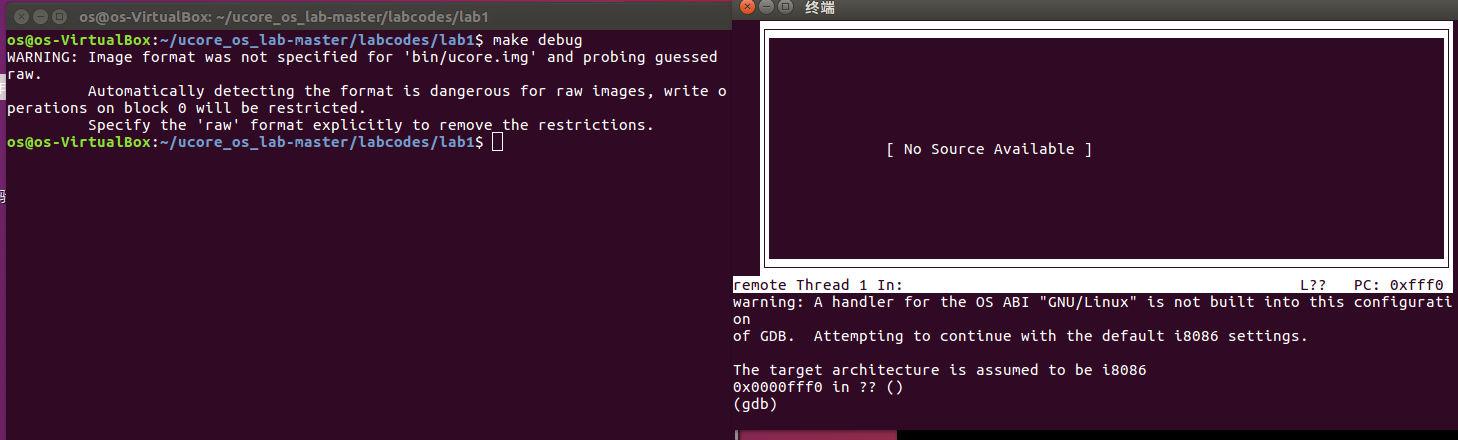
练习2：

1、

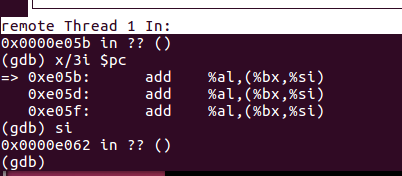
在lab1中修改gdbinit为：



输入make debug



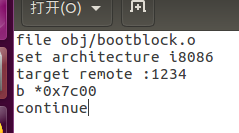
输入si可以查看BIOS的执行过程中cs和pc和内存地址变化，输入x/ni $pc可得到n步的汇编语言



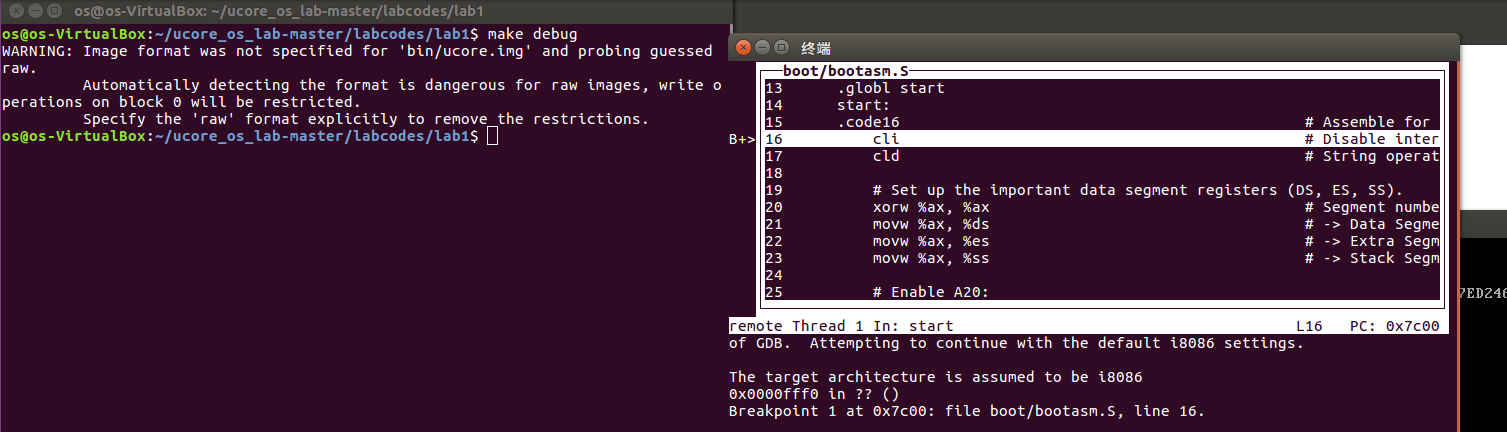
可知bios存储在内存中，cpu执行从0x0000fff0开始。

2、3

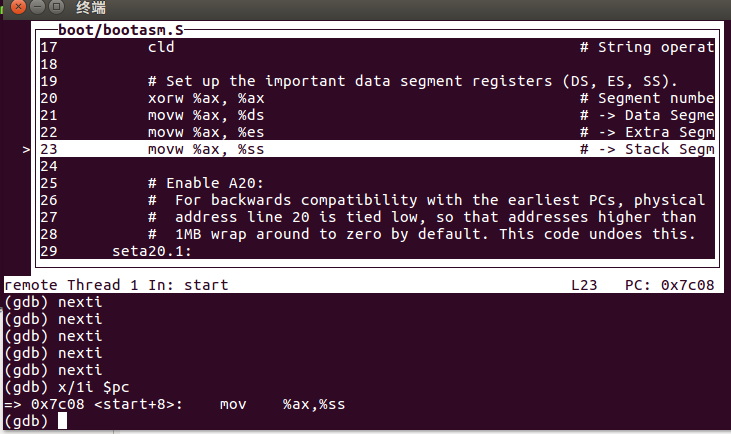
在lab1中修改gdbinit为：



输入make debug



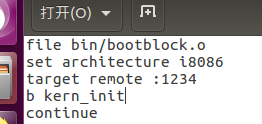
输入si可以查看BIOS的执行过程中cs和pc和内存地址变化，输入x/ni $pc可得到n步的汇编语言



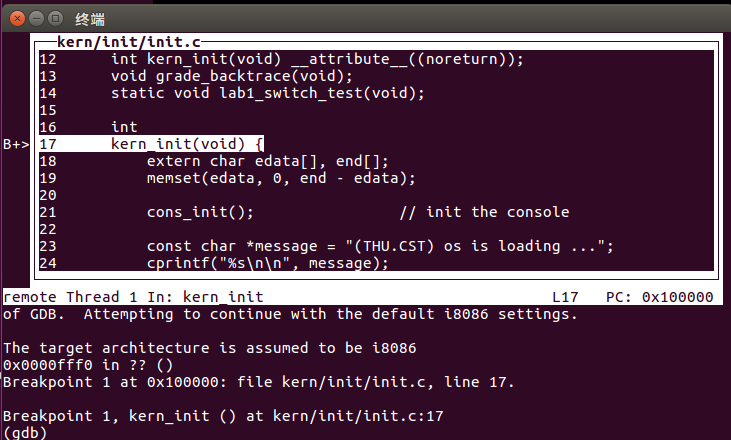
可知0x7C00为主导程序入口，代码与bootasm.S一致。反汇编代码也与bootasm.s一致。

4

在lab1中修改gdbinit为：



输入debug 可知在内核入口处增加了断点，代码执行停在kern\_init函数。



练习3：

1. 为何开启A20,以及如何开启A20?

当A20地址线控制禁止时，则程序就像在8086 中运行，1MB以上的地是不可访问的。在保护模式下 A20 地址线控制是要打开的。为了使能所有地址位的寻址能力,必须向键盘控制器8042发送- - 个命令。键盘控制器8042将会将它的的某个输出引脚的输出置高电平,作为A20地址线控制的输入。一旦设置成功之后,内存将不会再被绕回(memory wrapping)这 样我们就可以寻址intel 80286 CPU支持的16M内存空间，或者是寻址intel 80386以上级别CPU支持的所有4G内存空间了。

2、如何初始化GDT表

查看bootasm.s可知所有GDT表项初始化为全段,除了空段

3、如何使能进入保护模式?

通过长跳转指令ljmp $PROT\_ MODE\_ CSEG, Sprotcseg进入了保护模式。进入保护模式之后还有一个步骤:把所有的数据段寄存器指向上面的GDT描述符表中的数据段( 0x10)。