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）

列出本实验各练习中对应的OS原理的知识点，并说明本实验中的实现部分如何对应和体现了原理中的基本概念和关键知识

点。

在此练习中，大家需要通过静态分析代码来了解：

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？

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。

修改lab1/tools/gdbinit，内容为

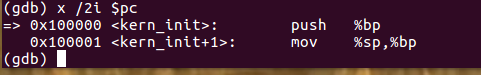


然后在lab1中执行

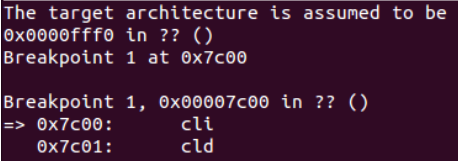


在调试界面输入命令：si来进行单步跟踪

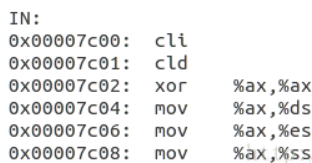
执行命令来查看BIOS代码



1. 在初始化位置0x7c00设置实地址断点,测试断点正常。

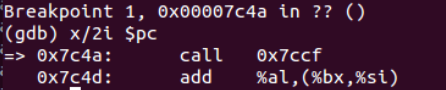


3. 从0x7c00开始跟踪代码运行,将单步跟踪反汇编得到的代码与bootasm.S和 bootblock.asm进行比较。



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。

修改gdbinit文件，在0x7c4a 处设置断点，然后输入make debug,得到结果：



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

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。

如何初始化GDT表

1载入GDT表

Lgdt gdtdesc

如何使能和进入保护模式

movl %cr0, %eax

orl $CR0\_PE\_ON, %eax

movl %eax, %cr0

练习**4**：分析**bootloader**加载**ELF**格式的**OS**的过程。（要求在报告中写出分析）

通过阅读bootmain.c，了解bootloader如何加载ELF文件。通过分析源代码和通过qemu来运行并调试bootloader&OS，

bootloader如何读取硬盘扇区的？

bootloader是如何加载ELF格式的OS？