

本科生实验报告

实验课程:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_操作系统\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

实验名称:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_保护模式\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

专业名称:\_\_\_\_\_\_\_\_\_\_\_计算机科学与技术\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

学生姓名:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_凌国明\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

学生学号:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_21307077\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

实验地点:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_教室\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

实验成绩:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

报告时间:\_\_\_\_\_\_\_\_\_\_\_\_\_\_2023.03.22\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **实验要求**

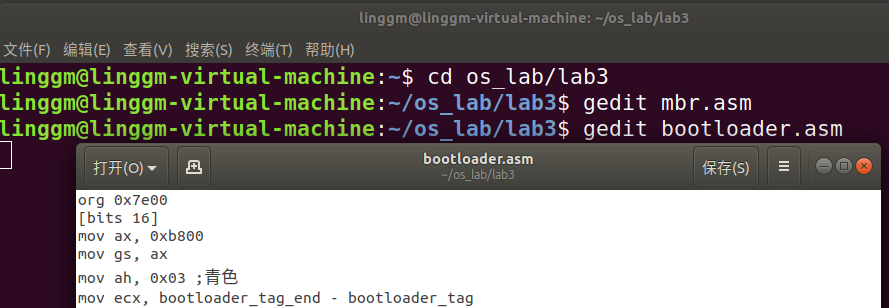
* 学习到如何从16位的实模式跳转到32位的保护模式
* 在平坦模式下运行32位程序
* 学习如何使用I/O端口和硬件交互。

**实验任务**

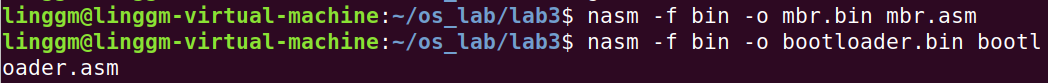
* 复现Example 1，说说你是怎么做的并提供结果截图
* 在Example 1中将LBA28读取硬盘的方式换成CHS读取，同时给出逻辑扇区号向CHS的转换公式。（利用int13h中断）
* 复现Example 2，使用gdb或其他debug工具在进入保护模式的4个重要步骤上设置断点，并结合代码、寄存器的内容等来分析这4个步骤，最后附上结果截图。gdb的使用可以参考appendix的“debug with gdb and qemu”部份。
* 改造“Lab2-Assignment 4”为32位代码，即在保护模式后执行自定义的汇编程序

1. **实验过程**
2. **复现example1**

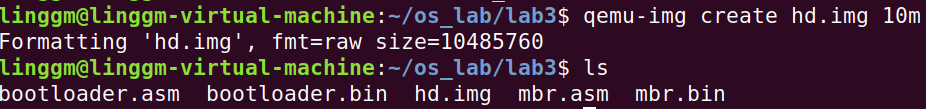
* 第一步：编写mbr和bootloader，mbr的任务是加载bootloader，bootloader的任务是打印字符（代码详见代码展示部分）



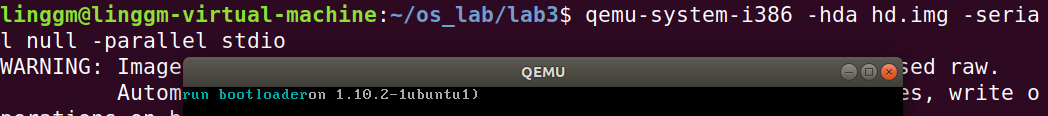
* 第二步，编成bin格式文件



* 第三步：创建磁盘



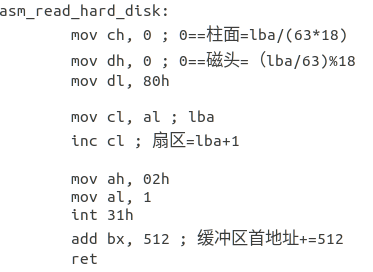
* 第四步：启动qemu测试



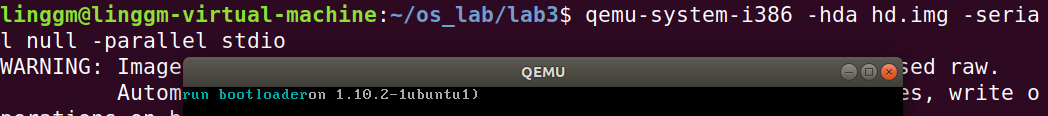
成功输出，复现完成

1. **CHS读磁盘**

* 第一步：改写mbr中的asm\_read\_hard\_disk函数



* 第二步：与复现example1相同



**C = LBA // （每柱面磁道数 \* 每磁道扇区数）**

**H = （LBA // 63）% 每柱面磁道数**

**S = （LBA % 每磁道扇区数）+ 1**

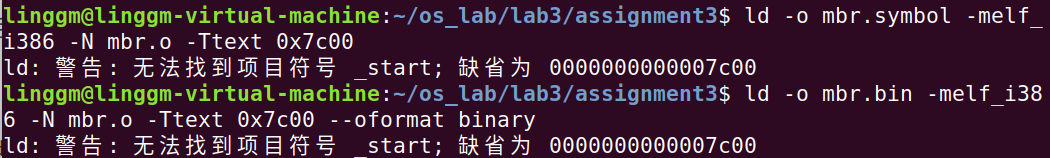


1. **利用gdb进行example2的debug分析**

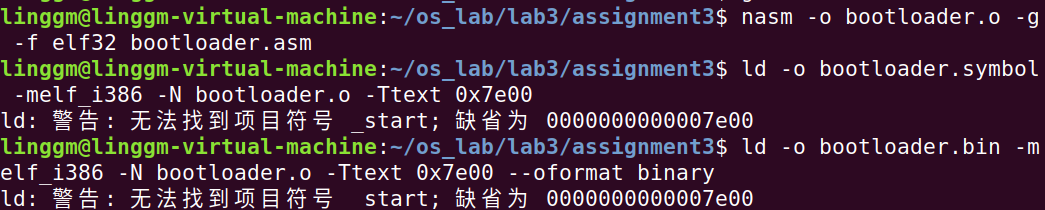
* 第一步：编译nasm，生成可重定位文件mbr.o，-g是添加debug信息



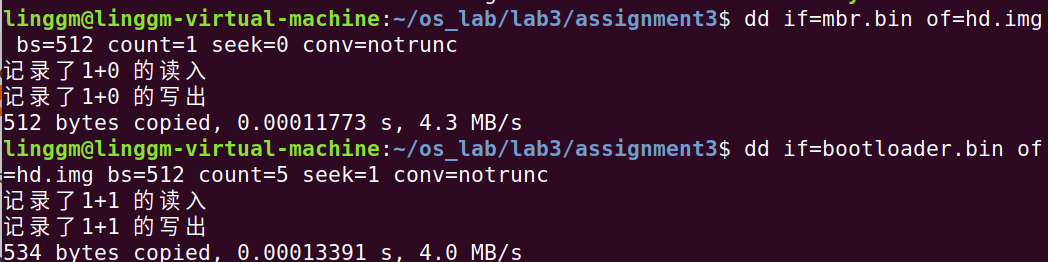
* 第二步：为可重定位文件mbr.o指定起始地址0x7c00，链接生成可执行文件mbr.symbol和mbr.bin



* 第三步：对bootloader做前两步操作

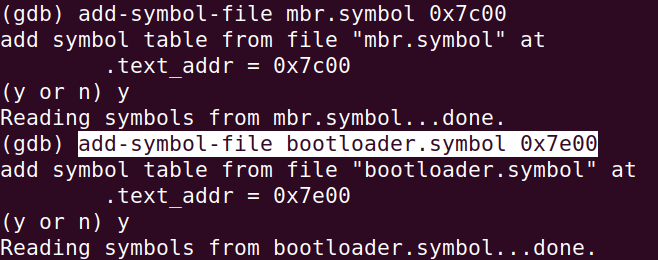


* 第四步：分别将mbr.bin和bootloader.bin写入磁盘



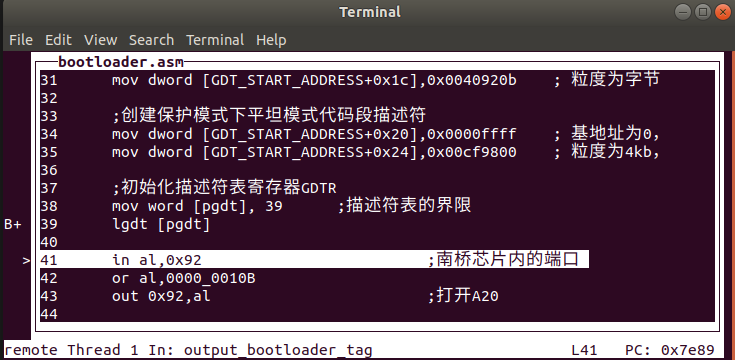
* 第五步：启动qemu，连接gdb，加载符号表





* 第五步：查看加载GDTR时

Layout src



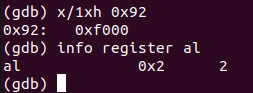
设置断点

pgdt地址

确实为39

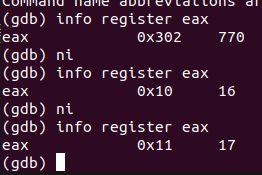
pgdt加载39

* 第六步：查看开第二十一根地址线

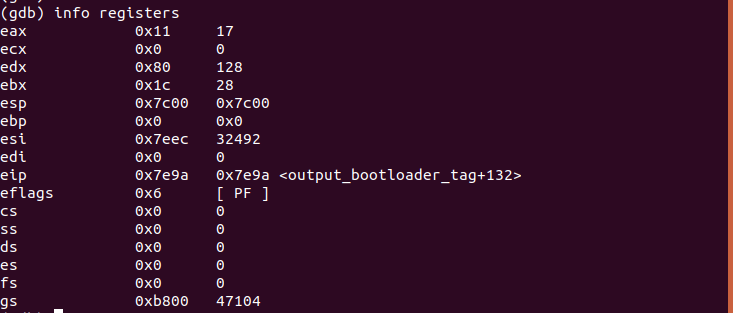


* 第七步：查看打开cr0的保护模式标志位

修改前cr0是16，修改后是17

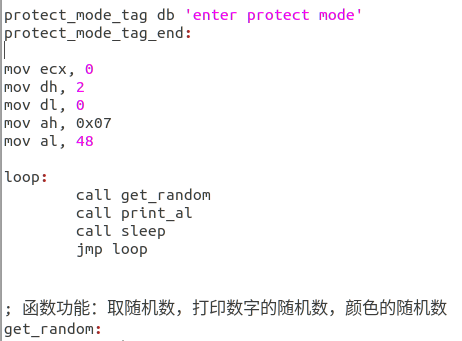
****

* 第八步：远跳转，进入保护模式



1. **将字符弹射程序改成32位，并运行**

步骤与复现example2步骤相同，mbr代码也相同，bootloader前面部分代码也相同，只是在bootloader后面的enter\_protect\_mode后加入代码

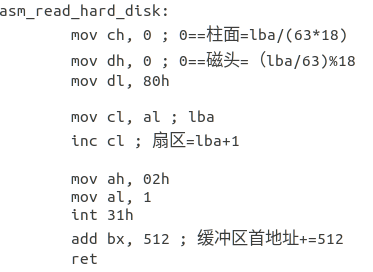


**注意：GDT中未引入中断向量表，故不可以使用intxx系列软中断**

加入代码后的步骤与复现example1的步骤一致，不过多叙述



1. **关键代码**
2. **复现example1的代码没有改动**
3. **利用int31h通过CHS方式读取磁盘的代码如下**

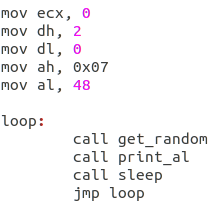


**C = LBA // （每柱面磁道数 \* 每磁道扇区数）**

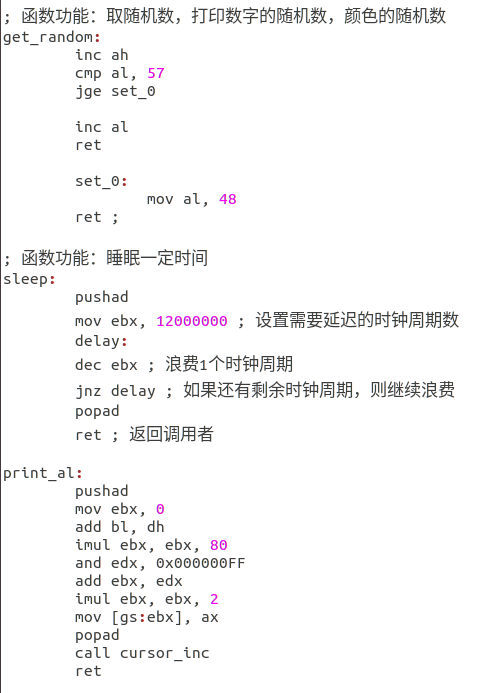
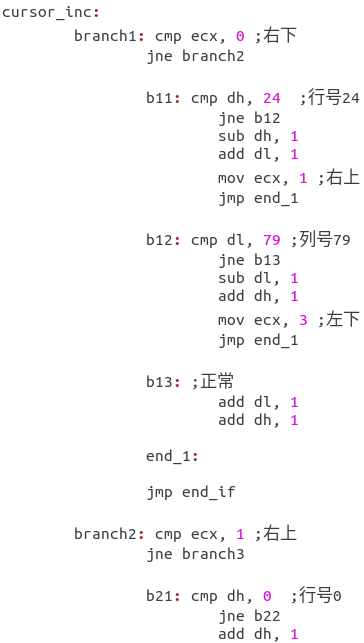
**H = （LBA // 63）% 每柱面磁道数**

**S = （LBA % 每磁道扇区数）+ 1**

1. **复现example2代码没有改动**
2. **字符弹射代码如下**



主函数：开始行号dh为2，列号dl为0，颜色ah，字符al

Get\_random函数设置字符及其颜色

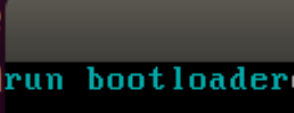
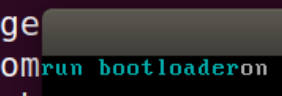
Sleep函数睡眠一定时间

Print\_al函数将字符打印在屏幕上

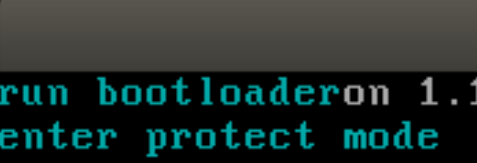
Crusor\_inc函数设置行列号，进行弹射

**注意：GDT中未引入中断向量表，故不可以使用intxx系列软中断**

1. **实验结果**
2. **Assignment1**

1. **Assignment2（debug信息见过程部分）**



1. **Assignment3**

****