

# Lab1 实验报告

191300087 左之睿 [1710670843@qq.com](mailto:1710670843@qq.com)

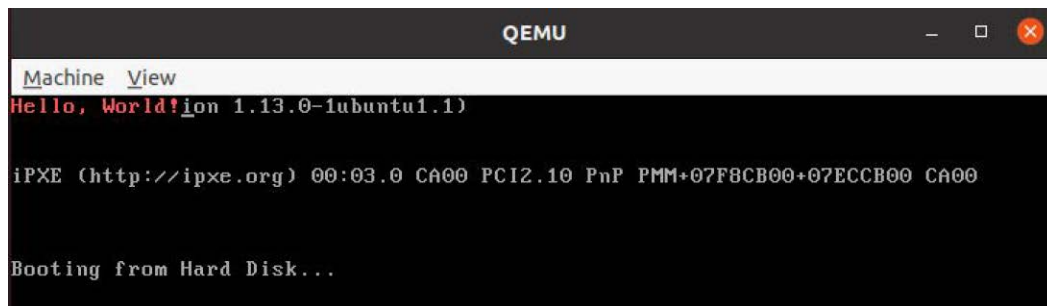
## 一、实验进度

我完成了全部任务

## 二、实验结果


如下图所示

### 1、实模式



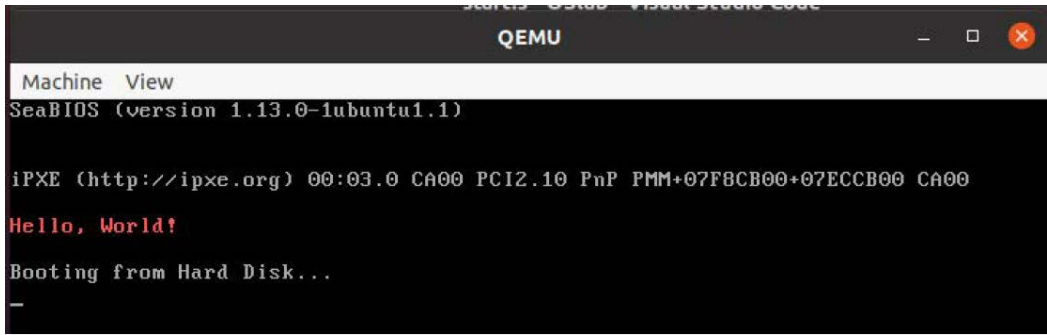
A screenshot of a QEMU window titled "QEMU". The window has a menu bar with "Machine" and "View". The main display area shows the following text: "Hello, World!ion 1.13.0-1ubuntu1.1)", "iPXE (http://ipxe.org) 00:03.0 CA00 PCI2.10 PnP PMM+07F8CB00+07ECCB00 CA00", and "Booting from Hard Disk...".

### 2、保护模式



A screenshot of a QEMU window titled "QEMU". The window has a menu bar with "Machine" and "View". The main display area shows the following text: "SeaBIOS (version 1.13.0-1ubuntu1.1)", "iPXE (http://ipxe.org) 00:03.0 CA00 PCI2.10 PnP PMM+07F8CB00+07ECCB00 CA00", "Hello, World!", and "Booting from Hard Disk...".

### 3、保护模式下加载磁盘



## 三、实验修改的代码位置

实验中修改的代码位于boot.c(任务三)和start.s中，其中，实模式（乃至全部实验）的编程思路来自于index.md中1.4节指引我们自己DIY扇区并输出Hello World这一部分以及app.s中的框架代码。

### 任务一

使用int指令引发中断，查阅中断向量表可以知道0x10号中断可以显示字符串，故通过使用该指令来打印字符串。这一部分代码与index.md中的指引几乎完全相同。

### 任务二

cli指令起到禁止中断的作用，该部分要设置CR0的第0位以及填写GDT表项

```
cli #close
inb $0x92,%al
orb $0x02,%al
outb %al,$0x92
data32 addr32 lgdt gdtDesc # load GDTR
```

GDT表项在框架代码中已经给出大致结构，填写部分就不再展示

### 任务三

这部分代码与任务二极其相似，我只在start.s相关位置添加了jmp bootMain并填写了boot.c中的bootMain函数

```
...
movl $0x8000,%eax
movl %eax,%esp
jmp bootMain
pushl $13
...

void bootMain(void) {
void (*elf)(void);
elf=(void(*) (void))0x8c00;
for(int i=0;i<200;i++){
    readSect((void*)(elf+512*i),1+i);
}
elf();
}
```

## 四、一点问题

实验中遇到的

```
movb $0x0c, %ah
```

向寄存器传入字符串颜色信息的代码，但我无论是否保留这行代码始终能够输出红色字符，估计默认颜色就是红色？