

操作系统 实验一

171860607

白晋斌

一、实验要求

本实验通过实现一个简单的引导程序，介绍系统启动的基本过程

1. 在实模式下实现一个 Hello World 程序

在实模式下在终端中打印 Hello, World!

2. 在保护模式下实现一个 Hello World 程序

从实模式切换至保护模式，并在保护模式下在终端中打印 Hello, World!

3. 在保护模式下加载磁盘中的 Hello World 程序运行

从实模式切换至保护模式，在保护模式下读取磁盘 1 号扇区中的 Hello World 程序至内存中的相应位置，跳转执行该 Hello World 程序，并在终端中打印 Hello, World!

二、实验过程

首先配置环境，执行的命令依次如下：

```
$sudo apt-get update
```

```
$sudo apt-get install qemu-system-x86
```

```
$sudo apt-get install vim
```

```
$sudo apt-get install gcc
```

```
$sudo apt-get install gdb
```

```
$sudo apt-get install binutils
```

```
$sudo apt-get install make
```

```
$sudo apt-get install perl
```

切换到 lab 目录下之后，依次输入（切换目录过程略去）

```
cd bootloader; make bootloader.bin
```

```
cd app; make app.bin
```

```
cat bootloader/bootloader.bin app/app.bin > os.img
```

（1）保护模式下的 helloworld

之后键入：

```
qemu-system-i386 os.img
```

得下图

```
/* Protected Mode Hello World */
```

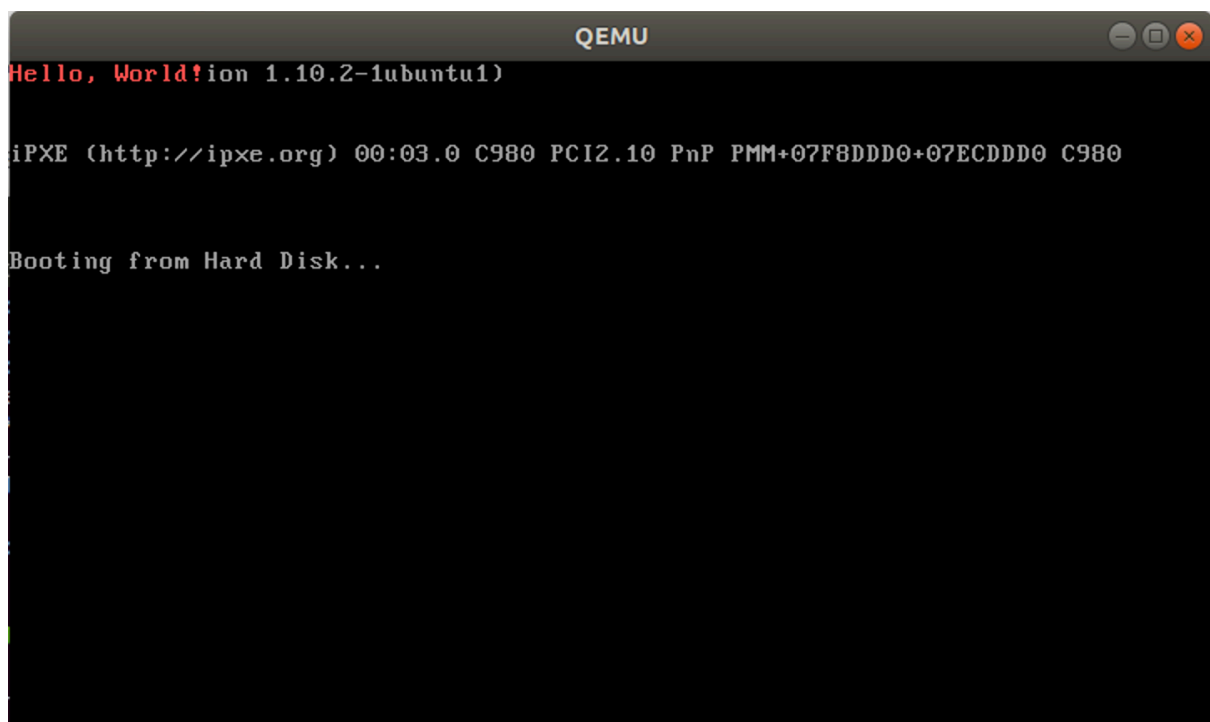
A screenshot of a QEMU terminal window. The title bar says "QEMU". The terminal text is: "SeaBIOS (version 1.10.2-1ubuntu1)", "iPXE (http://ipxe.org) 00:03.0 C980 PCI2.10 PnP PMM+07F8DDDD0+07ECDDDD0 C980", "Hello, World!" (in red), and "Booting from Hard Disk...".

```
SeaBIOS (version 1.10.2-1ubuntu1)

iPXE (http://ipxe.org) 00:03.0 C980 PCI2.10 PnP PMM+07F8DDDD0+07ECDDDD0 C980
Hello, World!
Booting from Hard Disk...
```

(2) 实模式下的 helloworld

将 start.s 中实模式注释的代码取消注释，保护模式中的代码注释掉，再次生成 os.img，运行得

A screenshot of a QEMU terminal window. The title bar says "QEMU". The terminal text is: "Hello, World!" (in red), "SeaBIOS (version 1.10.2-1ubuntu1)", "iPXE (http://ipxe.org) 00:03.0 C980 PCI2.10 PnP PMM+07F8DDDD0+07ECDDDD0 C980", and "Booting from Hard Disk...".

```
Hello, World!
SeaBIOS (version 1.10.2-1ubuntu1)

iPXE (http://ipxe.org) 00:03.0 C980 PCI2.10 PnP PMM+07F8DDDD0+07ECDDDD0 C980

Booting from Hard Disk...
```

(3) 磁盘中的 helloworld

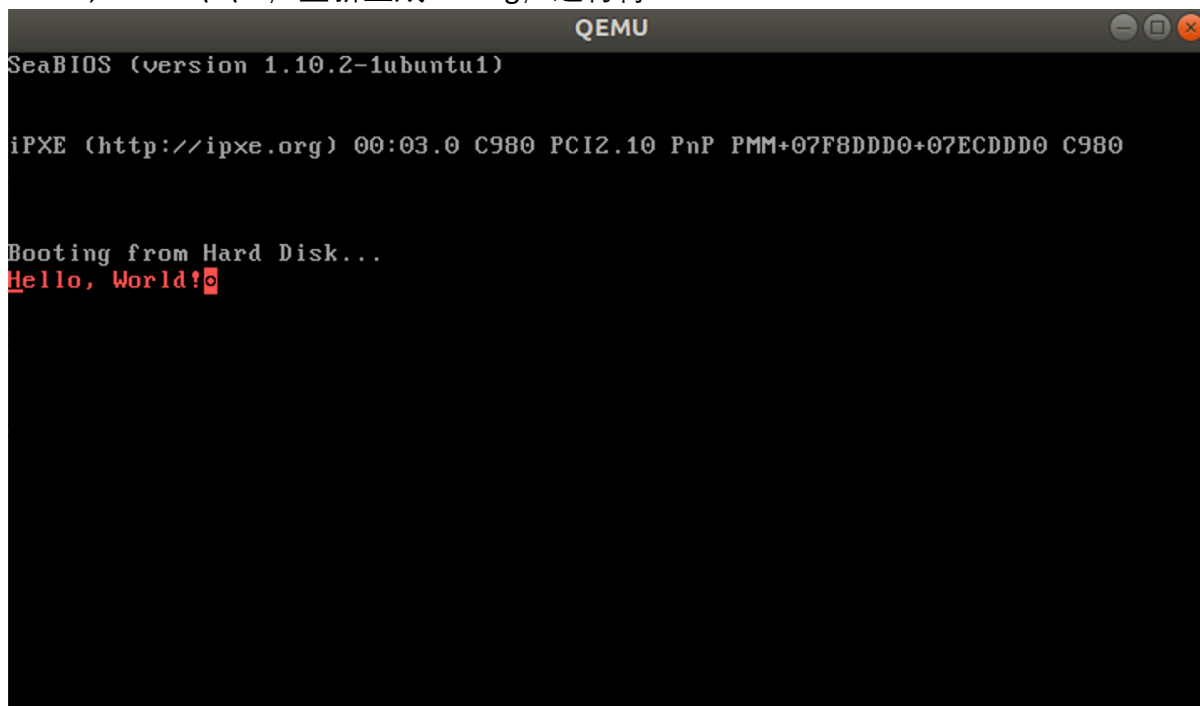
```
jmp bootMain
```

在 start.s 中添加一句跳转指令，在 app.s 中添加教程中所述的

通过写显存打印字符 **H**

```
movl $((80*5+0)*2), %edi    #在第5行第0列打印
movb $0x0c, %ah             #黑底红字
movb $42, %al               #42为H的ASCII码
movw %ax, %gs:(%edi)        #写显存
```

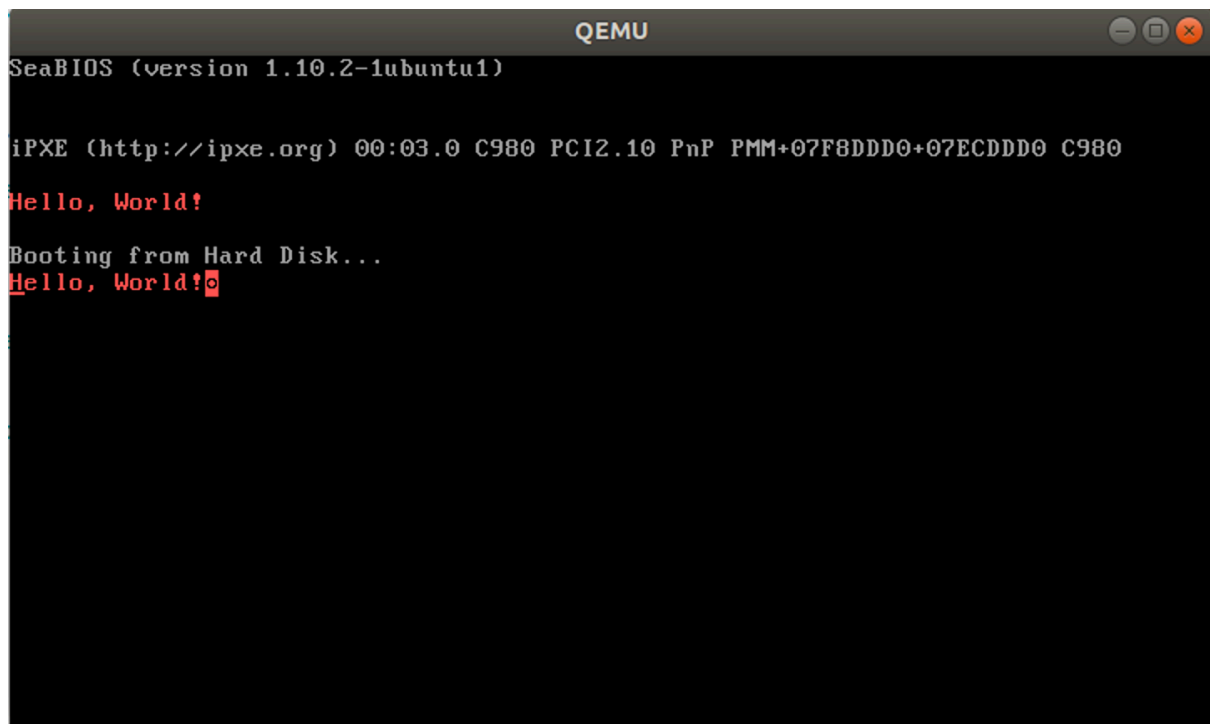
“Hello, World!\n\0”，重新生成 os.img，运行得



(注意：此处显示的 o 型字符实际为换行符)

并未出现同学所述的两个 hello world，原因是 jmp 时未跳回原代码，改为 call，如图。

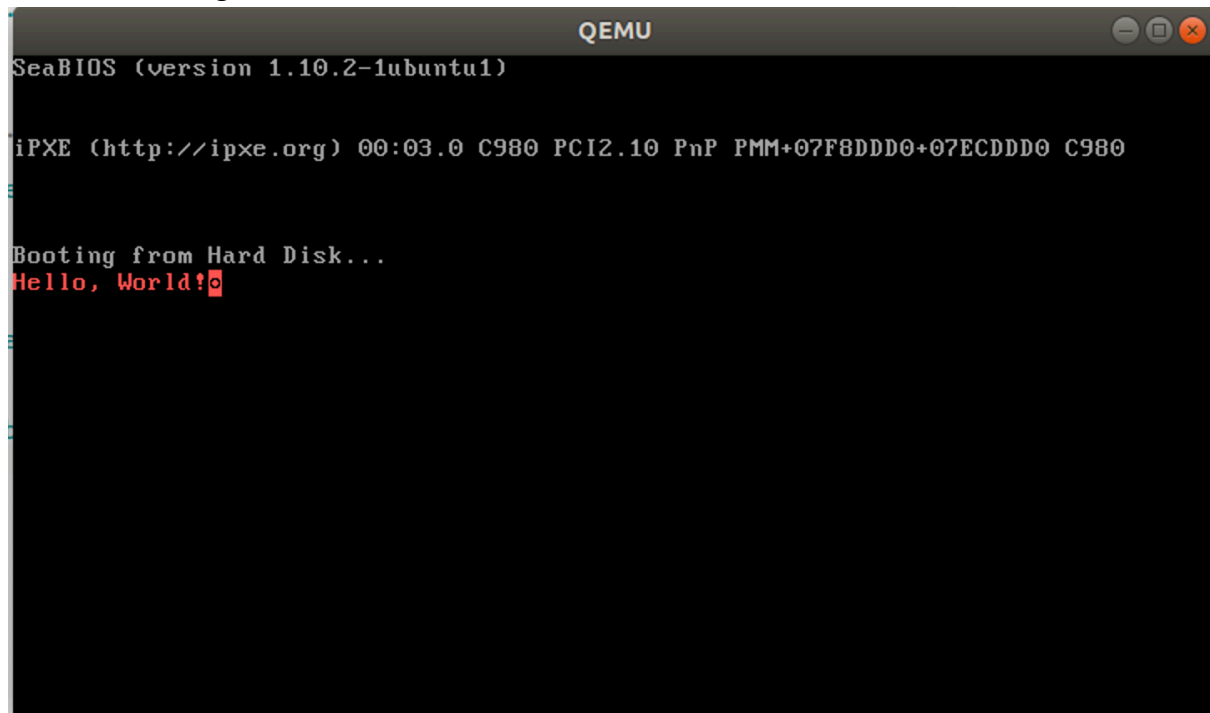
```
call bootMain
```



按照老师要求只需要留一个 hello world。故注释掉实模式代码中输出的那部分。

```
#pushl $13
#pushl $message
#calll displayStr
..
```

重新生成 os.img



完结撒花~

三、拓展功能

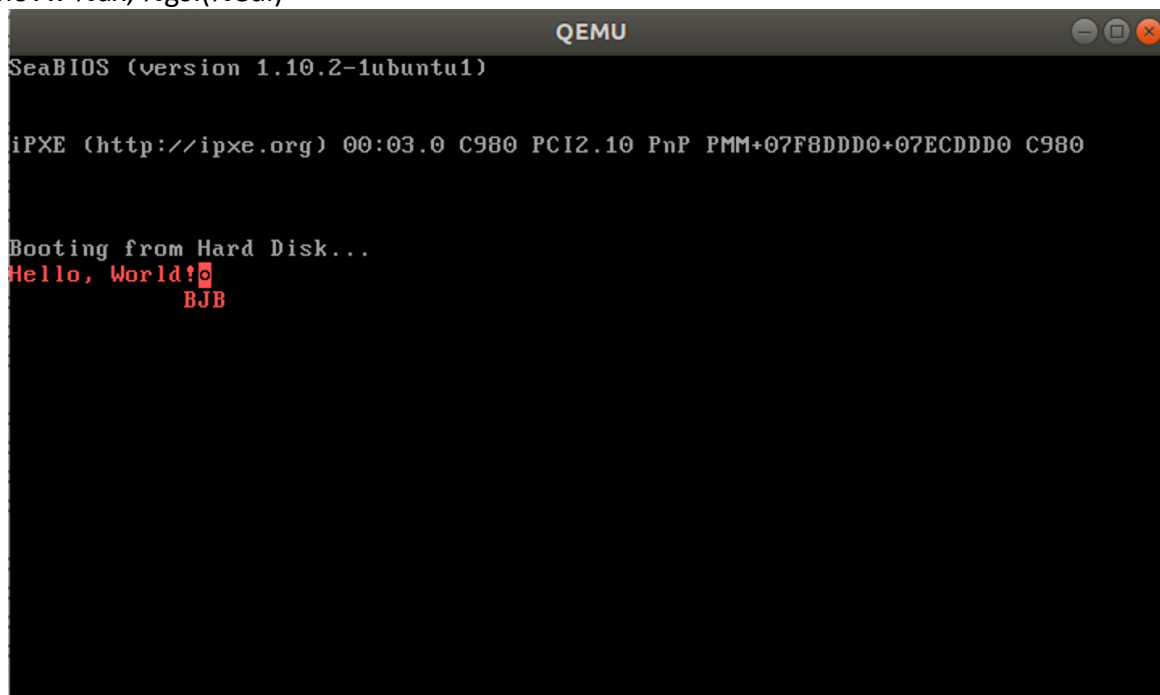
上课听老师一讲，决定对字体做以下尝试。

1.

```
movl $((80*9+12)*2), %edi
movb $0x0c, %ah
movb $66, %al          #B
movw %ax, %gs:(%edi)
```

```
movl $((80*9+13)*2), %edi
movb $0x0c, %ah
movb $74, %al          #J
movw %ax, %gs:(%edi)
```

```
movl $((80*9+14)*2), %edi
movb $0x0c, %ah
movb $66, %al          #B
movw %ax, %gs:(%edi)
```



2.对字体颜色进行修改，并插入空白字符，实现字符闪烁及颜色变化“闪闪发光”。写入多行，实现字符的“跳跃”。（展示效果 pdf 不可见，具体以程序运行结果为准）



```
10 Booting from Hard Disk...
10 Hello, World!
10 BJB
10 $
10
10
10
10
10
```