

lab1 实验报告

陈德丹 221220159 邮箱: 221220159@smail.nju.edu.cn

一、实验进度

我完成了所有内容

二、实验过程

1.1

```
# TODO:通过中断输出Hello World
pushw $13 # pushing the size to print into stack
pushw $message # pushing the address of message into stack
callw displayStr # calling the display function
```

先将准备打印的字符串的长度和地址压入栈中, 然后调用 displayStr

```
displayStr:
    pushw %bp
    movw 4(%esp), %ax
    movw %ax, %bp
    movw 6(%esp), %cx
    movw $0x1301, %axS
    movw $0x000c, %bx
    movw $0x0000, %dx
    int $0x10
    popw %bp
    ret
```

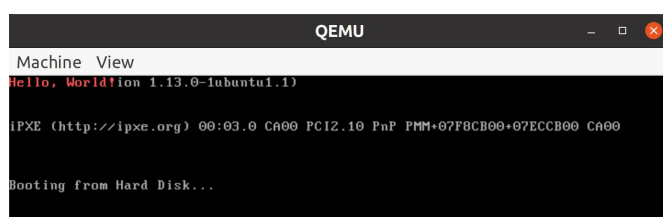
将字符串的地址, 长度以及其他显示参数放入寄存器中, 然后通过 int \$0x10 陷入屏幕中断,

调用 BIOS 打印字符串 Hello, World!

然后将 start.s 中 1.2 和 1.3 的内容注释掉, 此处遇到了一个错误, /*/* */ /*/, 最后结果是

13 配对, 4 在外面, 解决办法是把里面的注释用#代替 (感谢高心尧同学提供的帮助)

执行指令 make os.img 和 make play, 成功输出结果



1.2

```
# TODO:关闭中断
cli
```

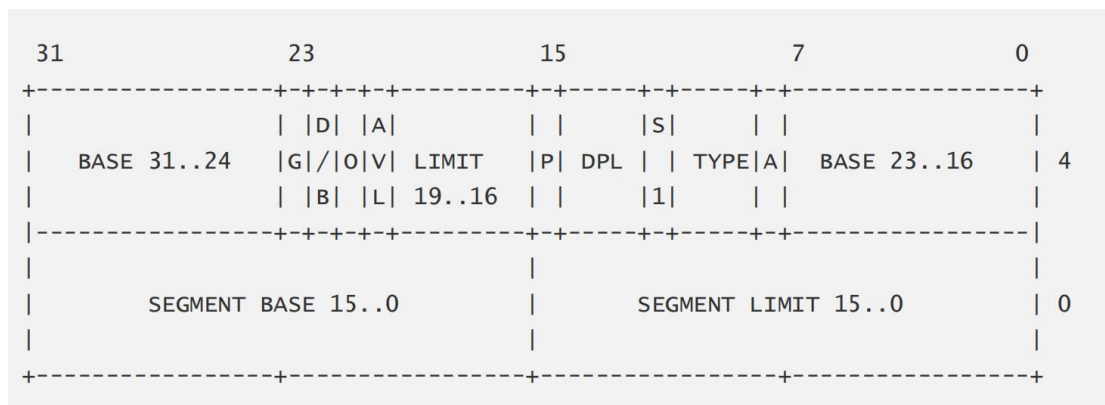
```
# TODO: 设置CR0的PE位（第0位）为1
movl %cr0, %eax
or $0x1, %eax
movl %eax, %cr0
```

注意不能直接在 cr0 上 or, 会报错

```
# TODO:输出Hello World
pushl $13
pushl $message
calll displayStr
displayStr:
    movl 4(%esp), %ebx
    movl 8(%esp), %ecx
    movl $((80*5+0)*2), %edi
    movb $0x0c, %ah
nextChar:
    movb (%ebx), %al
    movw %ax, %gs:(%edi)
    addl $2, %edi
    incl %ebx
    loopnz nextChar # loopnz decrease ecx by 1
    ret
```

首先将字符串在屏幕上显示的初始位置放入 edi 中, 将打印的属性放入 ah 中, 然后循环 13

次 (message 的长度), 把字符放入 al 中, 然后把整个 ax 放入屏幕缓冲区中



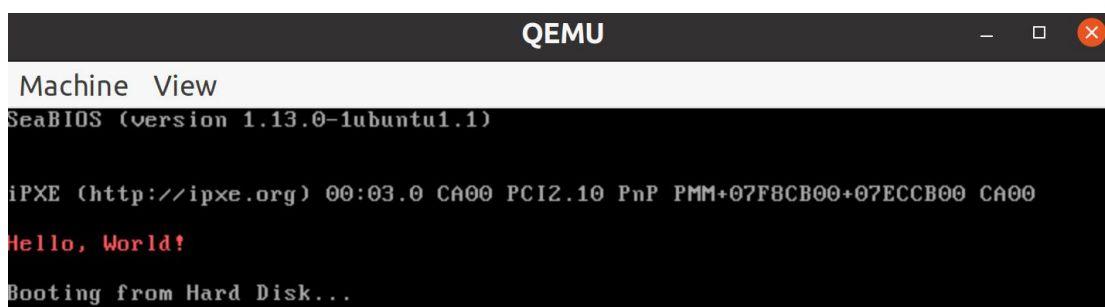
根据上表设置 gdt 表项，注意是小端方式，其中代码段与数据段的基地址都为 0x0，视频段的基地址为 0xb8000

```
# TODO: code segment entry
.word 0xffff,0
.byte 0,0x9a,0xcf,0

# TODO: data segment entry
.word 0xffff,0
.byte 0,0x92,0xcf,0

# TODO: graphics segment entry
.word 0xffff,0xb8000
.byte 0,0x92,0xcf,0
```

成功输出



1.3

start.s 中同样关中断，开启保护模式，设置 GDT 表项

跳转到 boot.c 中 bootMain 函数：

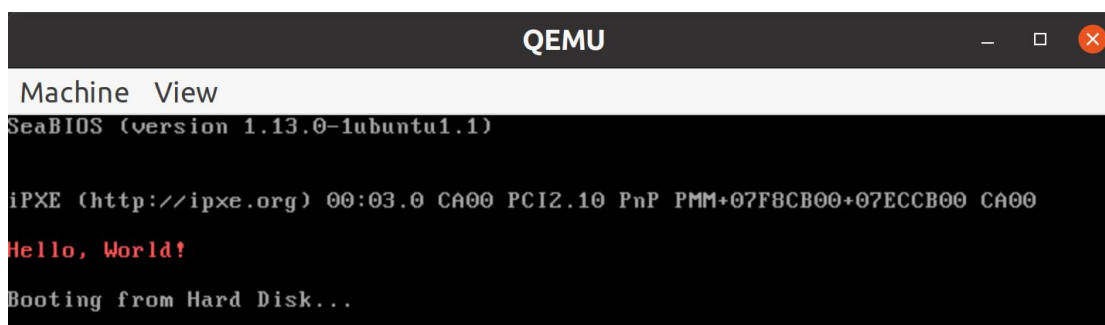
从 app/Makefile 中可知，设置的该 Hello World 程序入口地址为 0x8c00

```
ld -m elf_i386 -e start -Ttext 0x8c00 app.o -o app.elf

void bootMain(void) {
    //TODO
    void (*a)(void);
    a = 0x8c00;
    readSect(a, 0x1);
    a();
}
```

通过函数指针指向内存地址 0x8c00，调用 readSect 函数读取 1 号扇区的内容到该地址，

再通过 app 调用了加载到内存中的可执行程序的入口点，完成输出



三、思考题

1、你弄清楚本小结标题中各种名词的含义和他们间的关系了吗？请在实验报告中阐述。

CPU：CPU（中央处理器）是计算机的核心部件，负责执行程序指令和处理数据。

内存：内存是计算机用于临时存储数据和程序的地方。CPU 可以快速访问内存中的数据，以便执行指令。

BIOS：BIOS 在计算机内存上，它包含了用于初始化硬件和引导计算机的基本程序。

磁盘：磁盘是用于永久存储数据的存储设备，可以分为硬盘和固态硬盘两种类型。磁盘通常被划分为多个分区，每个分区可以包含一个文件系统。

主引导扇区 (MBR)：0 号柱面，0 号磁头，0 号扇区对应的扇区，512 字节，末尾两字节为魔数 0x55 和 0xaa，包含用于引导计算机操作系统的程序代码和分区表信息。

加载程序：加载程序是位于主引导扇区的一段特殊的程序代码，它负责引导计算机并加载操作系统的核心文件。

操作系统：操作系统是控制计算机硬件和软件资源的系统软件。它负责管理计算机的资源、提供用户界面、运行应用程序等。

关系：CPU 通过执行内存中 BIOS 中的引导程序，从磁盘的主引导扇区中加载操作系统的引导加载程序。引导加载程序负责进一步加载操作系统的核心文件到内存中，然后由 CPU 执行。操作系统一旦加载到内存中，就能够控制计算机的硬件资源，并提供用户界面和运行应用程序的环境。