LAB1 实验报告

张运吉（211300063、211300063@ smail.nju.edu.cn）

(南京大学人工智能学院, 南京 210093)

1 实验进度

我已经完成 **LAB1 全部内容。**

2 实验过程

2.1 Lab1.2

2.1.1 修改的代码

根据讲义，在 bootloader/start.s 中填写 gdt，开启 A20 地址线，把 cr0 最低位置为 1。

2.1.2 遇到的问题

第一个是 gdt 如何填写，讲义让我们参考 linux 的实现，因此我去查找了 linux 的手册，找到了很关键的下图：

段	Base	G	Limit	S	Type	DPL	D/B	P
用户代码段	0x00000000	1	0xfffff	1	10	3	1	1
用户数据段	0x00000000	1	0xfffff	1	2	3	1	1
内核代码段	0x00000000	1	0xfffff	1	10	0	1	1
内核数据段	0x00000000	1	0xfffff	1	2	0	1	1

然后我就按照 gdt 表项的结构把相应表项的值算了出来并且填到代码部分，这里还要注意的是大小端的问题。

通过上网查找资料，得知 A20 地址线通过系统端口 0x92 开启，启动 A20 地址线的原因是因为如果 A20 地址线未开，那么地址的第 20 位永远为 0，导致地址空间不连续。

把 cr0 最低位置为 1 比较简单，直接或上 0x1 就可以了。

然后就是关于 hello world 的颜色，一开始是红色的，但讲义上的是绿色，于是我查看源代码，发现只要把传给 ah 寄存器的值改成 0x0a 就可以了。

最后一个是初始化 DS ES FS GS SS 和初始化栈顶指针 ESP，根据段选择子的结构以及对应的段在 gdt 中的索引值计算出各个段选择子的值，然后填写。

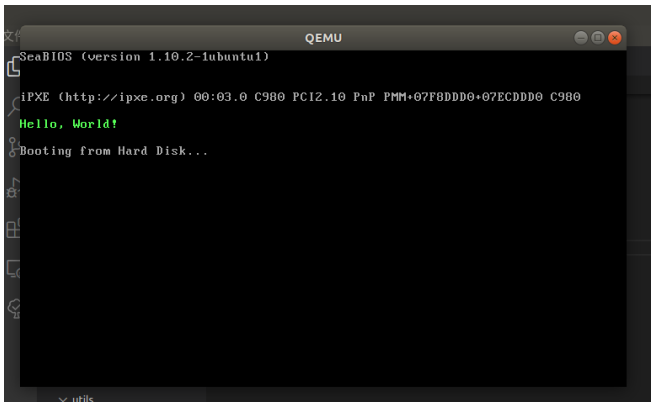
2.2 Lab1.3

2.2.1 修改的代码

Lab1.2 部分的保护模式可以适配到 lab1.3，唯一的修改就是填写 bootmain 函数，具体地，根据讲义内容我做了如下修改：

```
void bootMain(void) {
  // FIXME
  readSect((void*)0x8c00, 1);
  asm volatile("jmp 0x8c00");
}
```

2.3 实验结果



3 思考题

3.1 Ex1：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？

CPU 是计算机的核心部分，负责执行每一条指令；内存是计算机的存储单元，计算机所有的输入输出，都是要从内存来实现的，内存包括只读内存 ROM 和读写内存 RAM；BIOS 是一组固化到计算机内主板上一个 ROM 芯片上的程序；主引导扇区是磁盘上特定扇区的名称，又称主引导记录，主引导扇区包含一个加载程序，用于加载操作系统的代码和数据；操作系统是管理计算机硬件和软件资源的一段程序。

计算机启动后，第一条指令位于 BIOS 中，BIOS 中的程序对计算机硬件检查，检查没有问题后就将磁盘上的主引导扇区加载到内存，然后运行主引导扇区中的加载程序，把操作系统的代码和数据加载到内存，加载完成后，跳转到操作系统的第一条指令执行。

3.2 Ex2：中断向量表是什么？

中断向量表是存储中断向量的列表。中断向量储存了中断类型码和这种类型中断对应的处理程序的地址， CPU 遇到中断时，通过查询中断向量表可以跳转到中断处理程序位置对中断进行处理。

3.3 Ex3：为什么段的大小最大为 64KB？

因为 8086 是 16 为 CPU，段偏移地址只有 16 位，2^16 = 64KB.

4 实验感想

通过 lab1，我对计算机的启动过程以及操作系统的启动过程有了更加深入的了解，明白了实模式和保护模式的差别以及各自的特点，了解了段寄存器、GDTR、段表的格式，同时我也对.s 文件有了更好的了解。