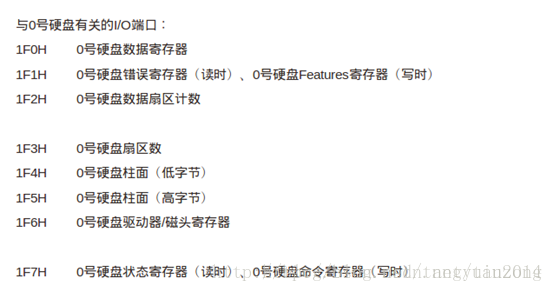
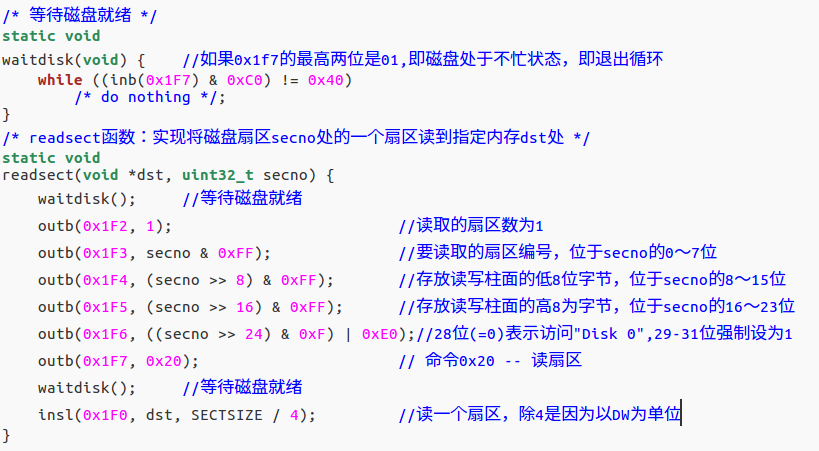
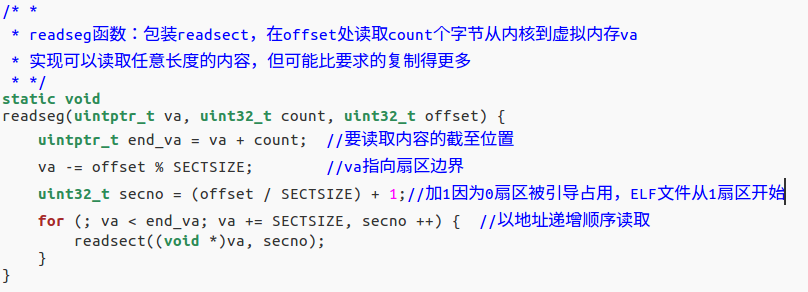
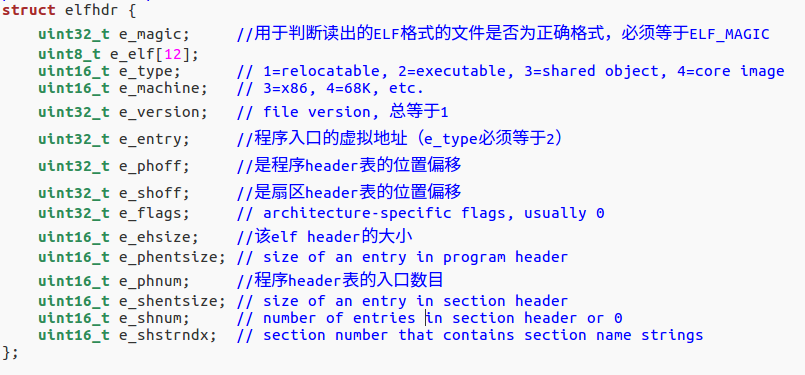
练习四：分析bootloader加载ELF格式的OS的过程   
1.bootloader如何读取硬盘扇区？   
　　(1)在练习3中实现了bootloader让CPU进入保护模式，下一步的工作就是从硬盘上加载并运行OS。参考指导书2.3.2节的硬盘访问概述后发现，bootloader的访问硬盘都是LBA模式的PIO（Program IO）方式，即所有的IO操作是通过CPU访问硬盘的IO地址寄存器完成。   
　　一般主板有2个IDE通道，每个通道可以接2个IDE硬盘。访问第一个硬盘的扇区可设置IO地址寄存器0x1f0-0x1f7实现的，具体参数见下图。一般第一个IDE通道通过访问IO地址0x1f0-0x1f7来实现，第二个IDE通道通过访问0x170-0x17f实现。每个通道的主从盘的选择通过第6个IO偏移地址寄存器来设置。    
  
　　(2)硬盘数据存储在硬盘扇区中，一个扇区的大小为512B，读一个扇区的流程（bootmain.c中的readsect函数）如下：   
    　　1、等待磁盘准备好；    
    　　2、发出读取扇区的命令；    
    　　3、等待磁盘准备好；    
    　　4、把磁盘扇区数据读到指定内存。    
　　在bootmain.c中的waitdisk函数用来实现等待磁盘就绪的功能，readsect函数实现将磁盘扇区secno处的一个扇区读到指定内存dst处，这里是只能读一个扇区，详细步骤见下图代码及注释：   
   
　　(3)readseg函数：包装readsect，在offset处读取count个字节从内核到虚拟内存va，实现可以读取任意长度的内容，但可能比要求的复制得更多。详细见下图代码及注释：   
   
　　至此利用readseg函数和readsect函数就可以实现bootloader读取硬盘扇区了。   
   
2.bootloader是如何加载ELF格式的OS？   
（1）ELF header在文件开始处描述了整个文件的组织，ELF的文件头包含了整个执行文件的控制结构，其定义在elf.h中，如下图（含注释）：   
 　　  
（2）bootmain函数，作为bootloader的入口，从硬盘读取8个扇区到内存0x10000处，并将其强制转换为elfhdr使用。然后根据头部的e\_magic是否等于ELF\_MAGIC判断ELF格式是否正确。之后根据描述表中的progeam header表的偏移量分别把程序段的数据读到内存中。详细代码及注释见下图：   
