#### 练习3.分析bootload进入保护模式的过程

0. BIOS通过读取硬盘主引导扇区到内存，并跳转到对应内存中的位置，也就是从’%cs=0 $pc=0x7c00‘进入并执行bootloader，bootloader要完成：

　　（1） 切换到保护模式，启用分段机制;

　　（2） 读取磁盘中的ELF执行文件格式的ucore操作系统到内存;

　　（3） 显示字符串信息;

　　（4） 把控制权交给ucore操作系统;

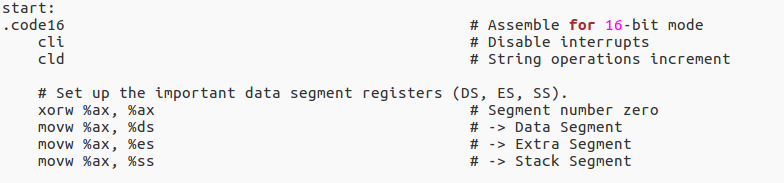
　　对应于boot目录下的 asm.h,bootasm.S,bootmain.c文件。bootasm.S的start函数文件是最先执行的，此函数进行了一定的初始化，完成从实模式到保护模式的转换，并调用bootmain.c中的bootmain函数。bootmain函数实现了屏幕、串口和并口显示字符串，加载ucore到内存，然后跳转到ucore的入口处执行。asm.h中是bootasm.S所需要的头文件，主要是一些与x86保护模式的段访问方式相关的宏定义。

　　所谓实模式，是将整个物理内存看成分段的取余，程序的数据和代码位于不同区域，操作系统和用户程序没有区别对待，而且每个指针都是指向实际的物理地址，若通过指针更改了操作系统或其他用户程序取余，会带来很大的灾难，此模式下的80386只是一个快速的8086。

　　所谓保护模式，80386的全部32根地址线全部有效，可寻址高达4GB的线性地址空间和物理地址空间，可访问64TB的逻辑地址空间，可采用分段存储管理机制和分页存储管理机制，提供4个特权级和完善的特权检查机制，既能实现资源共享又能保证代码数据的安全及任务的隔离。

 1. 初始化：

　　　首先调用bootasm.S中的start函数进行初始化，屏蔽中断，置位向量标志位，置位几个重要的段寄存器。



2. 开启A20：

　　　当A20地址线控制禁止时，程序就像运行在8086上，1MB以上的地址是不可访问的，为了使能所有地址位的寻址能力，必须向键盘控制器8082发送一个命令，键盘控制器8042会将A20线置于高电位，使全部32条地址线可用，实现访问4GB内存。开启A20的具体步骤如下：

　　（1） 等待8042 Input Buffer为空。

　　（2） 发送Write 8042 Output Port （P2）命令到8042 Input Buffer。

　　（3） 等待8042 Input Buffer为空。

　　（4） 将8042 Outpput Port （P2）得到字节的第2位置1,然后写入8042 Input Buffer。

　　具体实现在bootasm.S文件中：



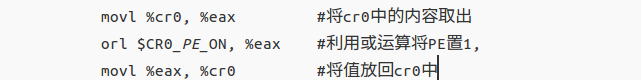
3. 初始化全局描述符表：

　　为了使分段存储管理机制正常运行，需要建立好段描述符和段描述符表，全局描述符表是一个保存多个段描述符的“数组”，其起始地址保存在全局描述符表寄存器GDTR中。GDTR长48位，其中高32位为基地址，低16位为段界限。这里只需要载入已经静态存储在引导区的GDT表和其描述符到GDTR寄存器：



4. 进入保护模式：

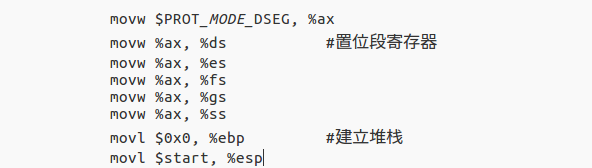
　　CR0的位0（PE）是启用保护（Protection Enable）标志。当设置该位时即开启了保护模式；当复位时即进入实地址模式。这个标志仅开启段级保护，而并没有启用分页机制。若要启用分页机制，那么PE和PG标志都要置位。所以这里需要将cr0的PE位置1：



5. 通过长跳转指令进入保护模式：



6. 置位段寄存器，建立堆栈：



7. 完成实模式到保护模式的转换，调用bootmain.c中的bootmain函数：

