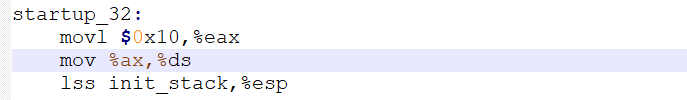
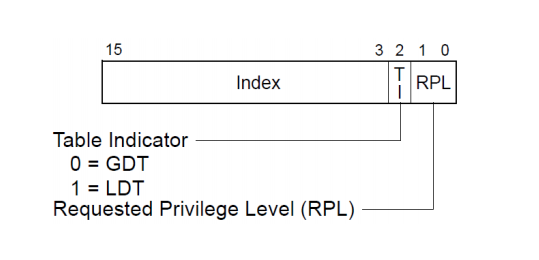
boot.s最后一句将代码段指向内存0x0的位置，并且将cs:eip指向基地址0x0，故此开始执行head.s

根据操作系统运行的原理，不论是否使用，所以数据段都必须要进行初始化，设置好代码段，紧接着就要设置数据段以及栈段



首先前两句将0x10存储到ds中，ds标识代码段的基地址，在32位保护模式下寻址方式

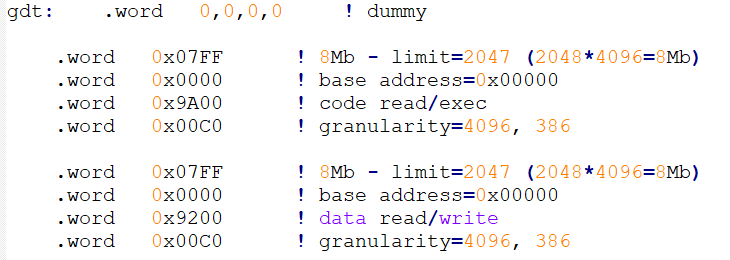
Segment Selector:



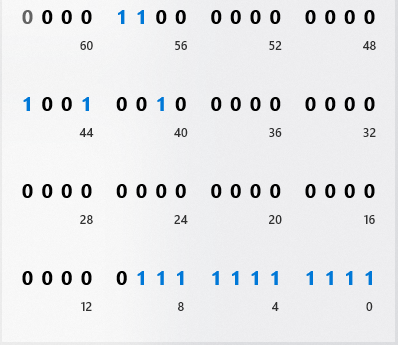
段选择子占16位，将0x10转换位16位二进制：0000 0000 0001 0000

对应上面段选择子的结构图可以发现，bit 0 bit 1为0表示特权等级为0，bit 2为0表示应该查看GDT表，bit 15-3为0000 0000 0001 0,表示索引GDT的第二项

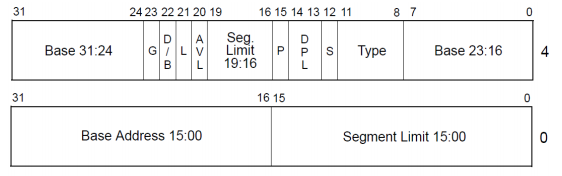
由于内存只有一个，GDTR还没有被修改过，所以当前的GDT表仍然为boot.s运行时所设置的GDT表，该表的内容如下：



索引(Index)2的内容为

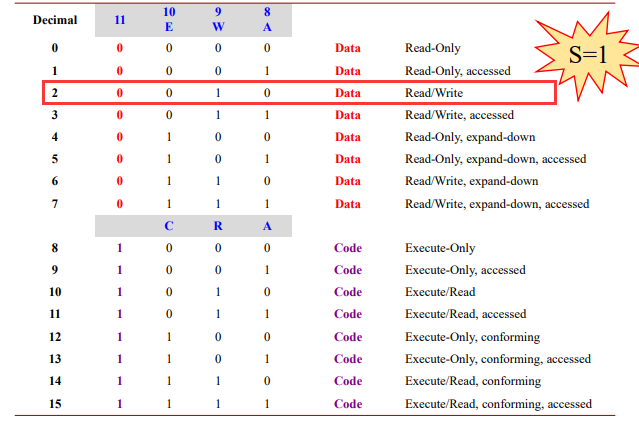


这是一个段描述符，需要转换为二进制与段描述符结构进行对比：



段描述符的结构如上图所示：

可以发现bit 12为1，对比如下表格可知



Bit 11-8为0010，为说明该段为数据段，可读可写，基地址为0x0

再下一句 lss init\_stack ,%esp

初始化栈段的ss与esp