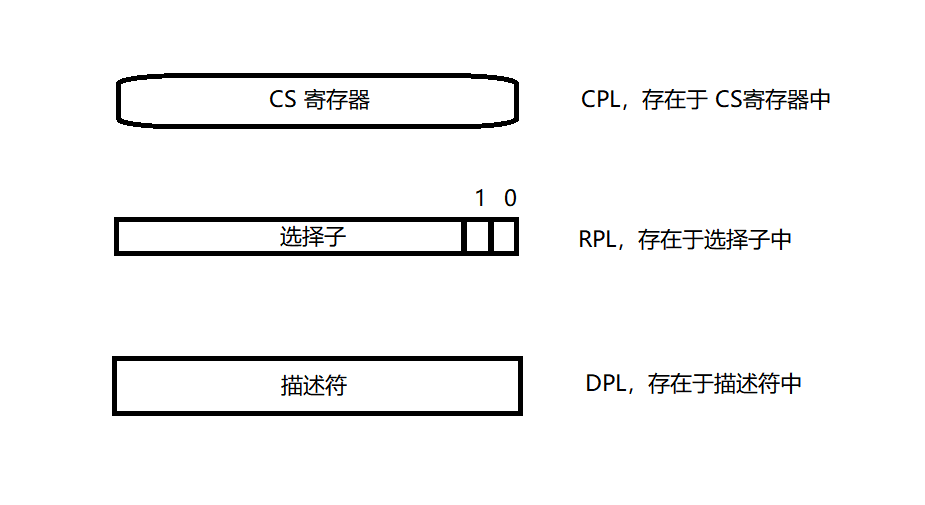
CPL是当前进程的权限级别(Current Privilege Level)，是当前正在执行的代码所在的段的特权级，存在于cs寄存器的低两位。  
  RPL说明的是进程对段访问的请求权限(Request Privilege Level)，是对于段选择子而言的，每个段选择子有自己的RPL，它说明的是进程对段访问的请求权限，有点像函数参数。而且RPL对每个段来说不是固定的，两次访问同一段时的RPL可以不同。RPL可能会削弱CPL的作用，例如当前CPL=0的进程要访问一个数据段，它把段选择符中的RPL设为3，这样虽然它对该段仍然只有特权为3的访问权限。  
 DPL存储在段描述符中，规定访问该段的权限级别(Descriptor Privilege Level)，每个段的DPL固定。  
当进程访问一个段时，需要进程特权级检查，一般要求DPL >= max {CPL, RPL}  
 下面打一个比方，中国官员分为6级国家主席1、总理2、省长3、市长4、县长5、乡长6，假设我是当前进程，级别总理（CPL=2）,我去聊城市(DPL=4)考察(呵呵),我用省长的级别(RPL=3 这样也能吓死他们:-))去访问,可以吧，如果我用县长的级别，人家就不理咱了(你看看电视上的微服私访，呵呵)，明白了吧！为什么采用RPL，是考虑到安全的问题，就好像你明明对一个文件用有写权限，为什么用只读打开它呢，还不是为了安全！

————>>



1. 对数据段和堆栈段访问时的特权级控制：  
    要求访问数据段或堆栈段的程序的CPL≤待访问的数据段或堆栈段的DPL，同时选择子的RPL≤待访问的数据段或堆栈段的DPL，即程序访问数据段或堆栈段要遵循一个准则：只有相同或更高特权级的代码才能访问相应的数据段。这里，RPL可能会削弱CPL的作用，访问数据段或堆栈段时，默认用CPU和RPL中的最小特权级去访问数据段，所以max {CPL, RPL} ≤ DPL，否则访问失败。