段寄存器中的组成：https://www.cnblogs.com/flatcc/archive/2019/09/30/11613649.html

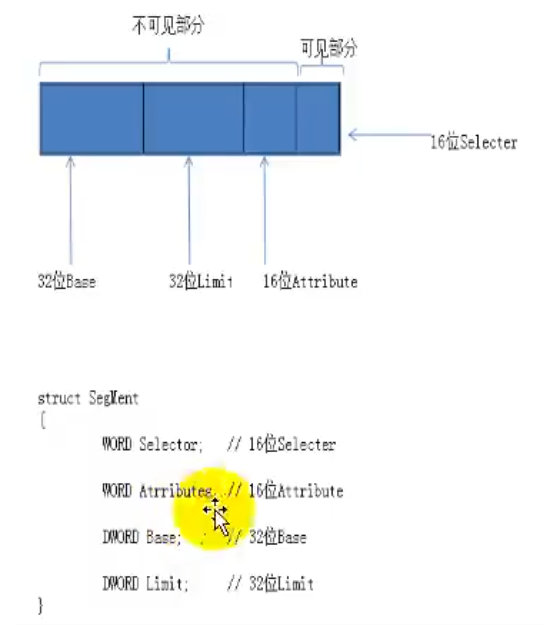
段选择子 2字节

属性 2字节

base基址 4字节

limit大小 4字节

一共96位，12个字节。

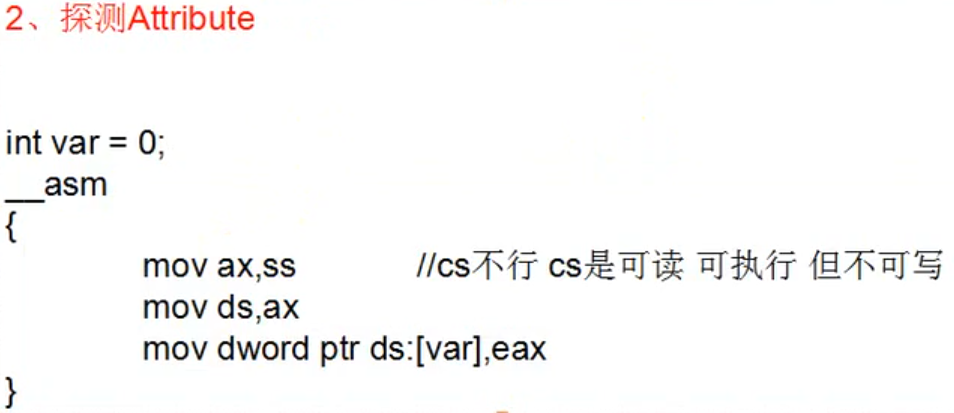




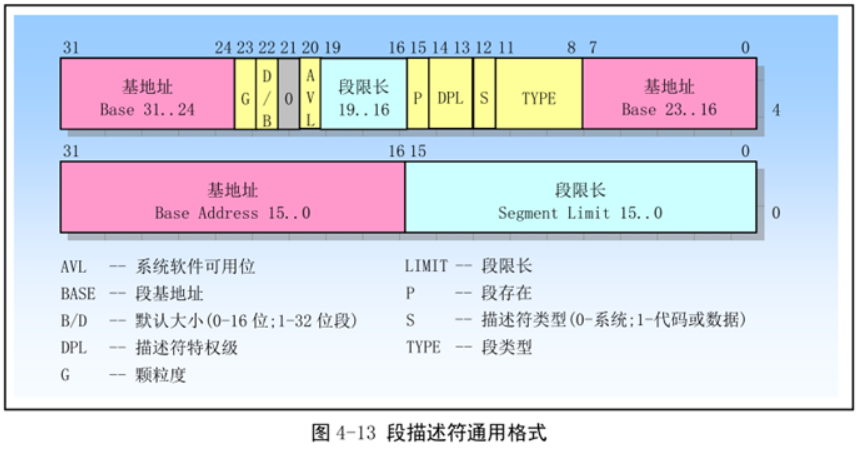
段选择子的结构：



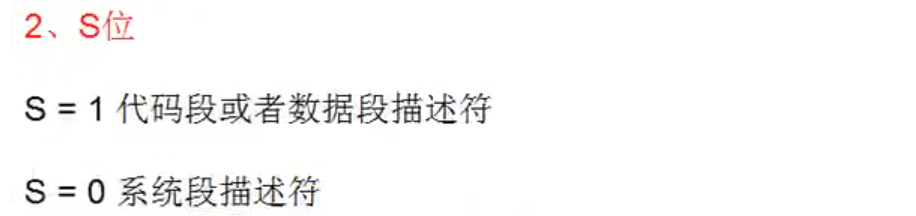
还要注意：每次对段寄存器操作时，实际是对段寄存器的低2个字节的段选择子来进行操作的。



GDT全局描述符表，其中每一项（也就是每一个描述符）的大小为8个字节（32位）。并且全局描述符表中的每一项，是通过段选择子的高13位来索引的。具体的步骤：先是通过gdtr寄存器来找到GDT的基址，之后通过2个字节的段选择子来索引具体的描述符，每个描述符的大小为8字节，也就是每次从GDT中索引得到8个字节。

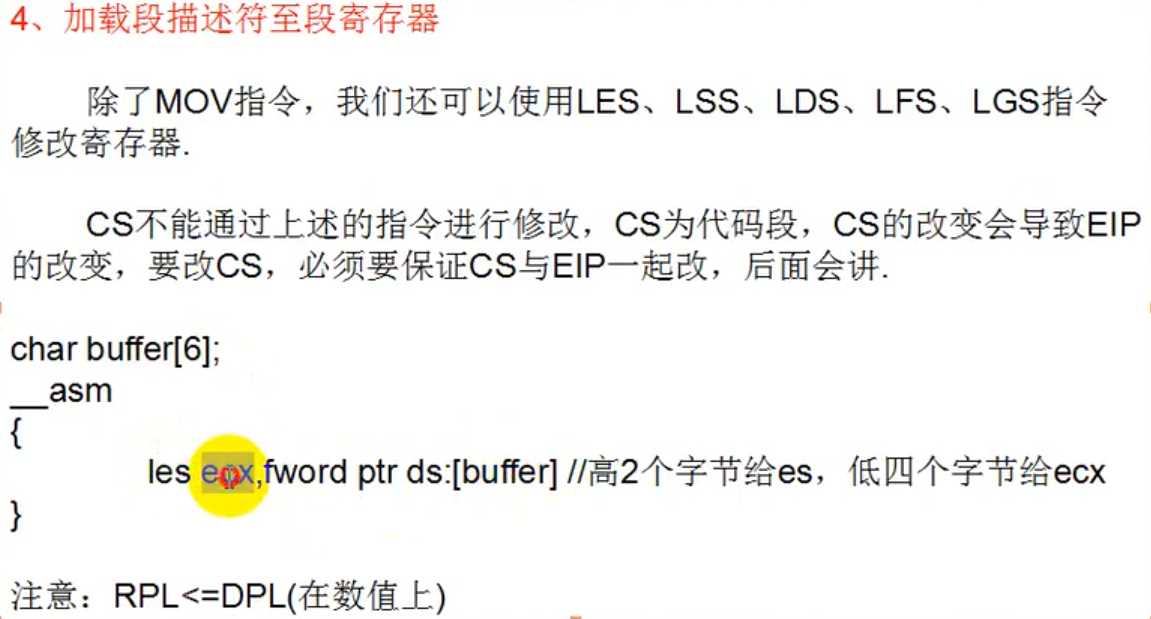








Les,lss,lds,lfs,lgs来直接修改段寄存器的方法汇编：

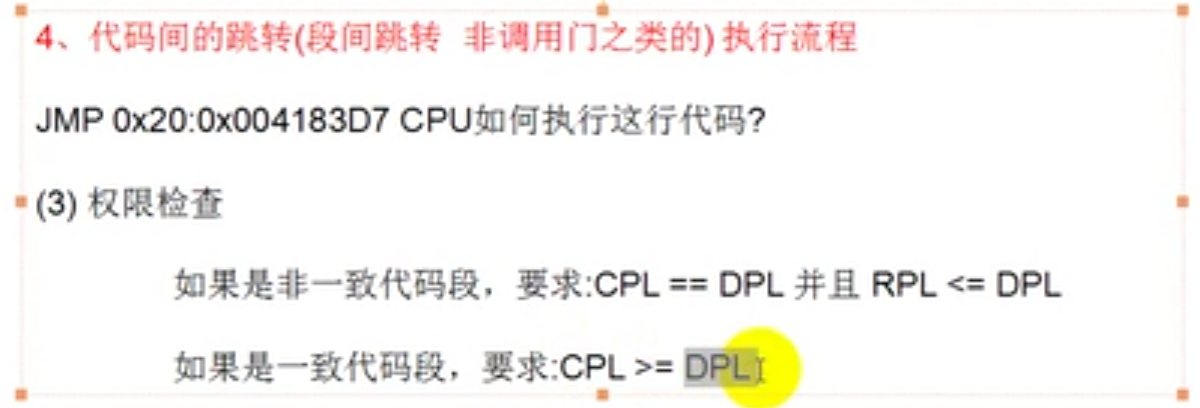
CPL：当前程序处于哪一个环，当前CPU处于哪一个环。CS寄存器的段选择子(共16bit)的后两个bit，表示的是CPL(Current Privilege Level):CPU当前特权等级  
CS和SS中存储的段选择子后2个bit。

也就是当前程序的CPL=CS.RPL 。CPL只存在于cs段寄存器的段选择子中。



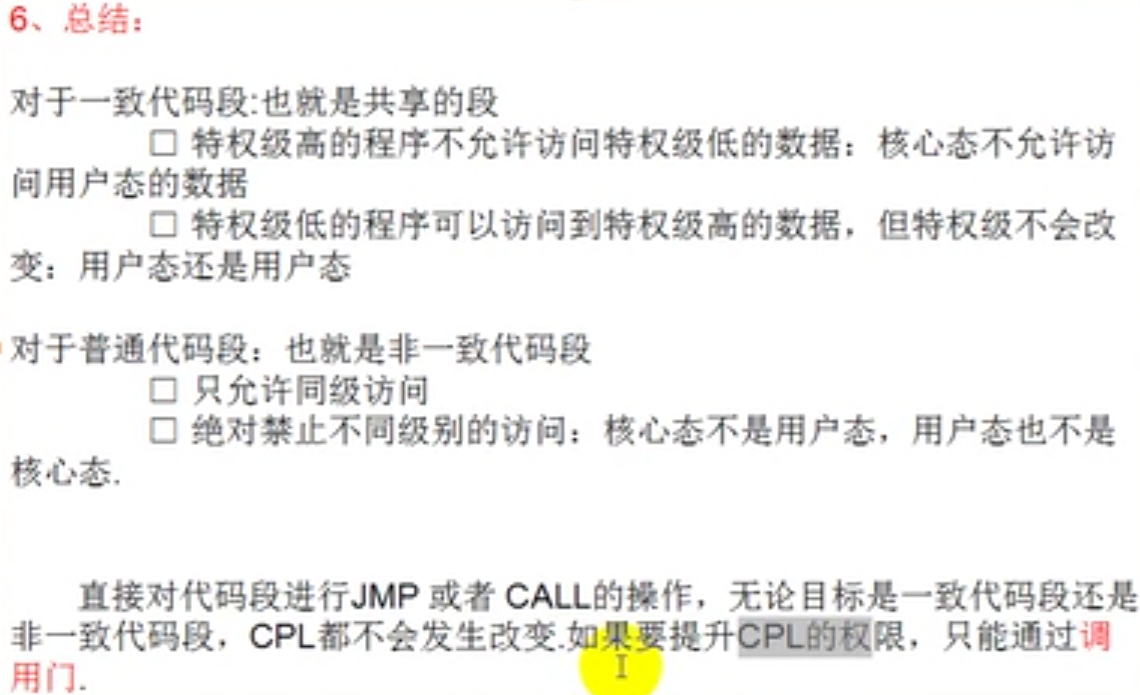
一致代码段：共享段，就比如操作系统提供了一段代码，这段代码提供一些通用的功能，这些功能不会对内核的数据构成破坏，系统希望这些功能可以直接被应用层访问的。

非一致代码段：不希望应用层代码可以直接访问的数据，就使用非一致代码段来修饰，



一致代码段中：CPL>=DPL说明CPL权限可以比DPL低，也就是说这段代码就是给低权限的人用的，（并且非常注意，还不能让高权限的访问，也即是内核不能访问，只能地权限的访问，仅仅供低权限的访问）因为不会对内核造成影响，是确定的安全的代码，并且一般是通用的代码，这样方便，并不是所有的内核的代码都是为内核准备的，一些代码放在内核，是为了更方便，更具有通用性，这样的代码，可以被应用层来访问，并且不会对内核造成影响的。

如果一些内核数据，不让低权限的来访问，就是要非一致代码段来修饰，防止应用层越权访问内核数据。大多数代码都是非一致代码段。



直接对代码段的jmp和call操作，不会权限提升（CPL不会改变），一致代码段可以访问到高权限的数据，打死你hi特权级不会改变，还是用户态。非一致代码段只允许同级访问，也不会改变权限和特权级。

但是可以通过调用门来提升CPL权限的。

代码的跨段执行：也就是修改cs寄存器。

由于cs段寄存器中，存放的是可执行代码的属性，基址和长度。当修改cs之后，相应的执行代码的基址和长度也会发生改变。因此只修改cs段寄存器会出现问题，例如：只修改cs寄存器的值，那么在之后执行程序时，会检查地址不在范围内，就会出问题。

JMP 0x20:0x004183D7

1.先段选择子拆分为0000 0000 0010 0000 ===》RPL=00 Ti=0 index=4

2.之后查询GDT表，找到对应段描符。

3．之后进行权限的检查，区分一直代码段和非一直代码段，

非一直代码段：CPL==DPL 并且 RPL<= DPL

一直代码段: CPL>=DPL

4.加载段描述符：通过上面检查，CPU会将段描述符加载到CS段寄存器中，也就是修改cs段寄存器

5.之后代码执行：也就是修改EIP

CPU将CS.base+Offset的值写入到EIP然后执行CS:EIP处的代码。

不能通过JMP 形式来跳转到系统的代码段来执行代码的（如果能得话就会修改），原因就是会对非一致代码段进行权限的检查。

**描述符的类型决定因素：**由S和Type来决定的

描述符的类型为：代码段，调用门，TSS任务段，任务门

S=0 系统段的各种描述符

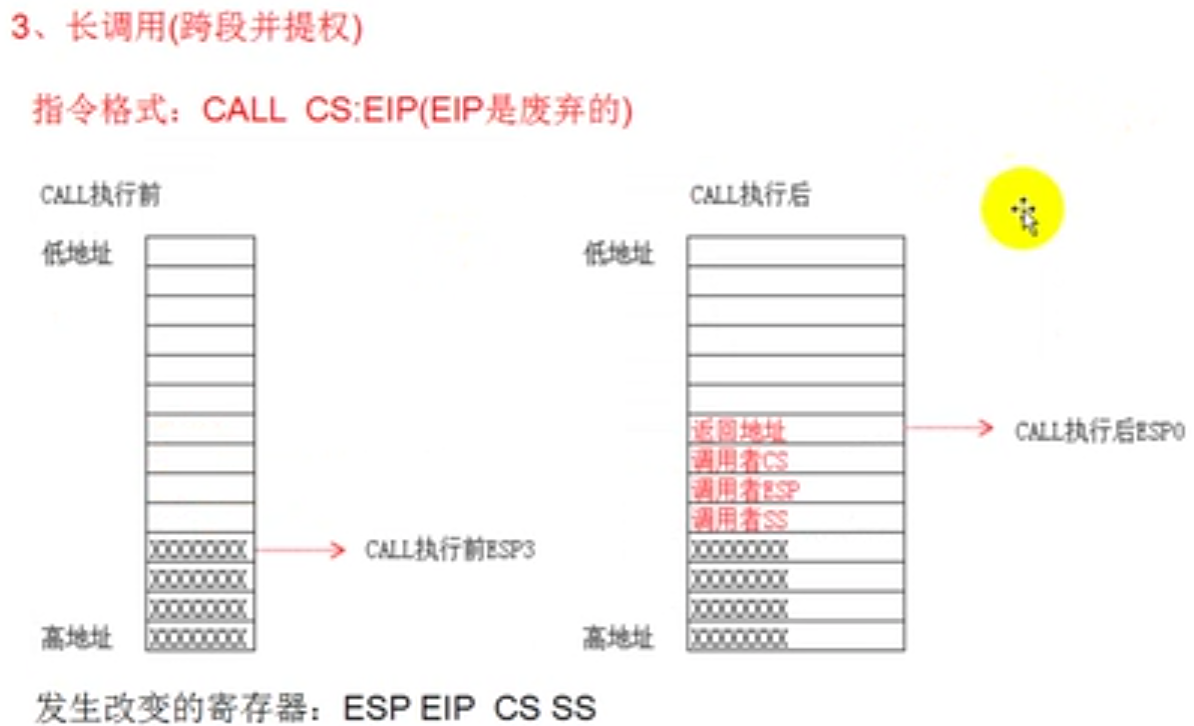
S=1 Type最高一位为0 数据段

S=1 Type最高一位为1 代码段



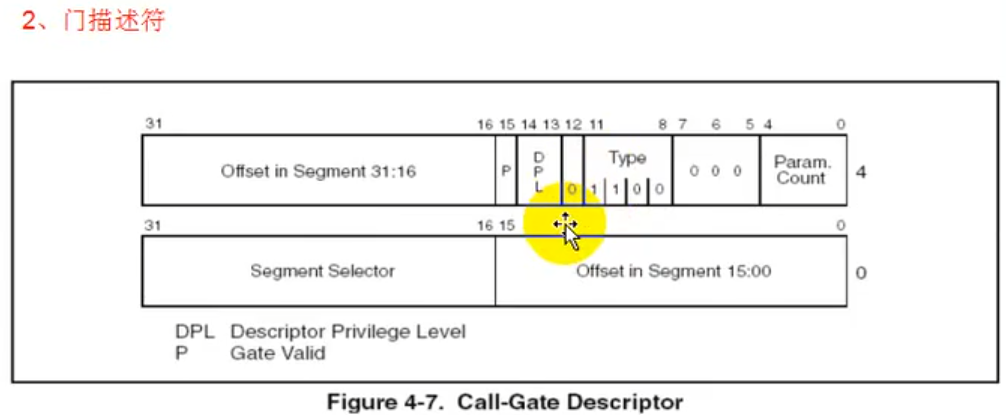
Jmnp指令只能跳转到代码段，而call指令可以调用调用门，因此call可以来提取，而jmp指令不能来提取，是jmp的功能限制了。因此jmp的cs的段描述符的S=1，且Type的最高一位为1。





提权长调用是两个不同的堆栈并且cs指向的描述符为调用门

门描述符的结构有些不同：



其中说明一下：低四个字节的Offset in Segment和高四个字节的Offset in Segment形成一个偏移，低四个字节中的Segment Selector是一个新的段选择子。并且S=0 Type=1 1 0 0

长调用时一共需要6个字节，第四个字为EIP，高两个字节为CS。当调用的cs为门描述符时，EIP没用到。

