几天前一个学生问我ARM中断嵌套的问题，我才发现原来在我心中理所当然的事对学生来说理解实属不易。   
ARM有七种模式，我们这里只讨论SVC、IRQ和FIQ模式。   
我们可以假设ARM核心有两根中断引脚（实际上是看不见的），一根叫 irq pin, 一根叫fiq pin。在ARM的cpsr中，有一个I位和一个F位，分别用来禁止IRQ和FIQ。

先不说中断控制器，只说ARM核心。正常情况下，ARM核都只是机械地随着pc的指示去做事情，当CPSR中的I和F位为1时，IRQ和FIQ全部处于禁止状态。无论你在irq pin和fiq pin上面发什么样的中断信号，ARM不会理你，你根本不能打断它，因为它“耳聋”，"眼瞎"了。 

当I位和F位为0时，irq pin上有中断信号过来时，就会打断arm的当前工作，并且切换到IRQ模式下，跳到相应的异常向量表（vector)位置去执行代码。这个过程是自动的，但是返回到被中断打断的地方就得您亲自动手。

当你跳到异常向量表，处于IRQ的模式的时候，此时如果irq pin上面又来中断信号，此时ARM是不会理你的，irq pin就像秘书，ARM核心就像老板，老板本来在做事，然后来了一个客户，秘书打断它，让客户进去。而此时再来一个客户，要么秘书不断去敲门问，要么客户走人。老板第一个客户没有会见完，不会理你。   
但是有一种情况例外，当ARM处在IRQ模式，这个时候fiq pin来了一个中断信号，fiq pin是什么？快速中断，好比公安局的来查刑事案件，才不管老板是不是在会见客户，直接打断，进入到fiq模式，跳到相应的fiq的异常向量表处去执行代码。那如果当ARM处理FIQ模式，fiq pin又来中断信号，也就是又一批公安来了，那没戏，都是执法人员，你打不断我。如果此时irq pin来了呢？来了也不理，正在办案，还敢来妨碍公务。   
  
所以得出一个结论： **IRQ模式只能被FIQ模式打断，FIQ模式下谁也打不断**。  
在打不断的情况下，irq pin 或fiq pin随便你怎么发中断信号，都是白发。   
所以**除了fiq能打断irq以外，根本没有所谓中断嵌套的情况**。   
但是再怎么说irq pin 和fiq pin加起来也就2根引脚，这么多中断源，怎么办呢？不可能谁来了都直接敲门吧。   
接下来该说谁来给irq pin或者 fiq pin发信号。从上文可以看到，可能是老板客户，也可能是公安。在ARM中，这个事情由中断控制器管理。   
拿最简单的2410/2440的中断控制器举例，中断控制器加一个子中断控制器，还有一个外部中断控制器管理了50多个中断资源，说穿了有50多个脚。这些脚除了外部中断都是规定了功能的，比如WDT、LCD、DMA等，这个功能不能改，因为2410/2440内部硬件连线已经决定了。   
当WDT和DMA中断都到来时，会被送到SRCPND寄存器中，两个中断都在里面，到底把哪一个送给ARM呢？这个时候先看INTMOD，也就是中断模式寄存器：哪个中断被设置成快速中断，哪个就被送上去；如果两个都被设置为快速中断呢？这不可能，因为**同一时间只能有一个中断可以被设成快速中断**。所以，如果有快速中断，这个时候直接给fiq pin发中断信号，打断ARM。   
要是没有快速中断呢，这个时候就看INTMSK，看WDT和DMA有没有被屏蔽，如果DMA在INTMSK被屏蔽，只有WDT继续向上送，如果都没有屏蔽，那么他们两个同时进入优先级寄存器PRIORITY，在这里根据优先级设置，一定会分出一个高一个低的优先级出来，优先级高的那个被送到INTPND寄存器，所以INTPND随时随地有且只有一个中断在里面。只要INTPND里面有中断，irq pin就不会一直不断给ARM发中断信号，当第一次发的时候，中断了ARM，这个时候ARM进入相应的异常向量并处于IRQ模式。

此时，INTPND仍然不断的通过irq pin向ARM发中断信号，但是此时ARM已经处于IRQ模式，不会理睬你。当你中断处理完，要退出IRQ模式，这个时候要小心，如果退出IRQ模式之前不清除INTPND里面的中断位，刚退出IRQ模式，又会被中断，因为INTPND一直在发中断信号。所以在退出IRQ模式前一定要清除INTPND里面的中断位**。**

光清除INTPND里面的位还不行，因为SRCPND里面WDT和DMA的中断在，当你刚清除完INTPND，结果SRCPND里面又选了一个中断出来送到INTPND里面。所以正确的处理方法是**退出IRQ模式之前，先清除SRCPND里相应的中断位，再清除INTPND里相应的位。**请注意，SRCPND里面可能有多个位，所以清除你已处理过的中断就行，而INTPND里面只可能有一位，直接清掉即可。

再说说Linux的情况。**Linux不用FIQ，只用到了IRQ**。但是我们有时候一个中断需要处理很长时间，我们需要占用IRQ模式那么长的时间吗？不需要，linux在IRQ模式下只是简单的记录是什么中断，马上切换回SVC模式，换句话说，**linux的中断处理都是在SVC模式下处理的。**

那么中断号是怎么来的呢？它在ARM上固定死了，相应的中断号只有一个办法得到：查询irqs.h 。先用一个中断号注册一个中断处理程序，当中断发生的时候，Linux怎么知道是我这个中断号发生的中断呢？在处理中断的时候，先读取INTPND，根据需要再读取EINTPEND或SUBSRCPND计算出一个中断号，相应的处理算法在get\_irq\_nr\_base这个宏中。irqs.h中的中断号就是根据这个算法把每个中断号算出来的。