任务切换的方法

1. 中断门和陷阱门

在实模式下，内存最低端的1M是中断向量表，保存着256个中断处理过程的段地址和偏移。当中断发生时，处理器把中断号乘以4，作为索引访问中断向量表，从相应的位置取出中断处理过程的段地址和偏移地址，并转移到哪里执行。

在保护模式下，中断向量表不再使用，取而代之的是中断描述符表（Interrupt Descriptor Table，IDT）。不要害怕，它和GDT、LDT是一样的，用于保存描述符。唯一不同的地方是，它保存的是门描述符，包括中断门、陷阱门和任务门。

当中断发生时，处理器用中断号乘以8（因为每个描述符占8个字节），作为索引访问IDT，找到某个门描述符。如果是中断门或陷阱门，那么就取出门描述符中的段选择子和段内偏移量，然后转移到相应的位置去执行。

中断门和陷阱门允许在任务内实施中断处理，转到全局空间执行一些系统级的管理工作。本质上，是任务内的控制转移行为。

2. 通过中断执行任务切换

但是，如果访问IDT时遇到的是任务门，那么就会引发任务切换。即，要中断当前任务的执行并保护现场，然后后切换到另一个任务去执行。

1. 任务门描述符中最重要的字段是段选择子，它应该指向新任务的TSS；
2. P位指示该门是否有效，当P=0时，不允许通过此门实施任务切换；
3. DPL是任务门描述符的特权级，但是对于因中断而发起的任务切换不起作用。

注意：任务门描述符可以安装在IDT、GDT和LDT中。

通过中断执行任务切换的过程，可以参考《Intel Architecture Software Developer’s Manual Volume 3:System Programming》的TASK SWITCHING这一节。   
我把其中的关键点总结如下（序号不代表发生的先后顺序）：   
1. 保存当前任务状态到它的TSS（由TR寄存器指向）中；   
2. 处理器访问新任务的TSS，从中恢复现场；   
3. TR寄存器指向新任务的TSS；   
4. 把旧任务的TSS选择子填写到新任务TSS中的“任务链接域”，将新任务EFLAGS寄存器的NT位置1，同时还要把新任务TSS描述符的B位置1，表示该任务状态为忙。（旧任务的TSS的B位不变，仍然为1.）

3. iret指令

当中断发生时，可以执行常规的中断处理过程，也可以执行任务切换。尽管性质不同，但是它们都要使用iret指令返回。前者是返回到同一任务内的断点处；后者是返回到上一层任务。

每个任务的TSS中都有一个任务链接域，其内容可以是前一个任务的TSS描述符的选择子。如果当前任务EFLAGS寄存器的NT位是1，则表示当前正在执行的任务嵌套于其他任务内，并且能够通过TSS任务链接域的指针返回到前一个任务。

可以使用iret指令从当前任务返回（切换）到前一个任务，前提是当前任务的NT位必须是1.无论何时处理器遇到iret指令，它都要检查NT位，如果是0，表明是一般中断过程，则按照一般的中断返回处理；如果是1，则表明当前任务之所以能够执行，是因为它打断了别的任务。因此，应当返回到原先被中断的任务继续执行，并且由处理器固件把当前任务EFLAGS的NT位改成0，并把TSS描述符的B位改成0.在保存了当前任务的状态后，接着用被中断的任务的TSS恢复现场。

4. 通过call或者jmp指令发起任务切换

在这两种情况下，call指令或者jmp指令的操作数是任务的TSS描述符选择子或者任务门。例如：

call 0x10:0x00000000

jmp 0x10:0x00000000

当处理器执行这两条指令时，首先用选择子索引GDT（对于本例，是GDT），分析得到的描述符类型   
1. 对于代码段描述符，就按照普通的段间转移执行；   
2. 对于调用门，按照调用门的规则执行；   
3. 对于TSS描述符或者任务门（下图中粉色部分），则执行任务切换。此时，指令中给出的32位偏移量被忽略，因为在执行任务切换时，所有处理器的状态都可以从TSS中获得。

使用CALL指令发起的任务切换类似于因中断发起的任务切换。也就是说，由CALL指令发起的任务切换是“嵌套”的。如下图所示：

5. 任务是不可重入的

执行任务切换时，新任务的状态不能为忙。处理器是通过TSS描述符的B位来检测是否重入的。由中断、iret、call和jmp指令发起任务切换时，处理器固件会检测新任务TSS描述符的B位，如果为1，则不允许执行这样的切换。

6. 总结

处理器可以通过以下四种方法实施任务切换：   
1. call指令或者jmp指令的操作数是GDT内的某个TSS描述符；   
2. call指令或者jmp指令的操作数是GDT或者LDT内某个任务门描述符；   
3. 一个异常或者中断发生时，中断号指向IDT内的某个任务门；   
4. 在EFLAGS寄存器的NT位置位的情况下，当前任务执行了一个iret指令。