

请求CPU关注的两种方式：

1. 基于中断
2. 基于轮询

基于轮询：对于像操作杆（或称为游戏手柄）上的CPU，它并不会等待设备发送中断。因为操作杆主要用于游戏，操作杆的移动必须非常快，因此使用轮询的方式检测设备是否需要CPU的关注还是比较理想的。使用轮询方式的缺点是CPU就处于了忙等状态，因为CPU会不停的多次检查设备。但是需要注意的是在linux中，这种处理信号的方式也是必不可少的。

硬中断（硬件产生）

对于上文所讨论的场景都是属于硬中断的例子。硬中断主要分为两种类别：

1. 非屏蔽中断(Non-maskable interrupts,即NMI)：就像这种中断类型的字面意思一样，这种中断是不可能被CPU忽略或取消的。NMI是在单独的中断线路上进行发送的，它通常被用于关键性硬件发生的错误，如内存错误，风扇故障，温度传感器故障等。
2. 可屏蔽中断 (Maskable interrupts)：这些中断是可以被CPU忽略或延迟处理的。当缓存控制器的外部针脚被触发的时候就会产生这种类型的中断，而中断屏蔽寄存器就会将这样的中断屏蔽掉。我们可以将一个比特位设置为0，来禁用在此针脚触发的中断。

软中断（软件产生）

这些中断是在CPU执行指令（也就是说在进程正在运行的时候）的时候产生的，因为在执行指令时，CPU（确切的说是CPU中的运算器）自身会产生一个异常（此处的异常也可理解为软中断）。

例如，一个数字除以0（当然这是不可能的），此时就会导致一个divide-by-zero的异常，从而导致计算机将此计算取消或者显示一个错误的信息。