# 中断

中断（英语：Interrupt）是指 **处理器接收到来自硬件或软件的信号，提示发生了某个事件，应该被注意，这种情况就称为中断**。

通常，在接收到来自外围硬件（相对于中央处理器和内存）的异步信号，或来自软件的同步信号之后，处理器将会进行相应的 *硬件／软件* 处理。发出这样的信号称为进行中断请求（interrupt request，IRQ）。**硬件中断导致处理器通过一个运行信息切换（context switch）来保存执行状态（以程序计数器和程序状态字等寄存器信息为主）；软件中断则通常作为CPU指令集中的一个指令，以可编程的方式直接指示这种运行信息切换，并将处理导向一段中断处理代码**。中断在计算机多任务处理，尤其是即时系统中尤为有用。

## 中断分类

### 硬件中断

由硬件发出或产生的中断称为硬中断，按硬中断事件的来源和实现手段可将中断划分为外中断和内中断：

* **外中断**：又称为中断或异步中断，是指 **来自处理器以外的中断信号，包括时钟中断、键盘中断、外部设备中断等**。外中断又分为可屏蔽中断和不可屏蔽中断，各个中断具有不同的优先级，表示事件的紧急程度，在处理高一级中断时，往往会部分或全部屏蔽低等级中断。
* **内中断**：又称为异常或同步中断（产生时必须考虑与处理器时钟同步），是指 **来自处理器内部的中断信号，通常是由于程序执行过程中，发现与当前指令关联的、不正常的或错误的事件**。内中断可以细分为：
	+ 访管中断，由执行系统调用而引起的。
	+ 硬件故障中断，如电源失效、总线超时等。
	+ 程序性中断，如非法操作、地址越界、除数为0和浮点溢出等。

### 软件中断

软件中断：是一条CPU指令，用以自陷一个中断。由于 **软中断指令通常要运行一个切换CPU至内核态（Kernel Mode/Ring 0）的子例程，它常被用作实现系统调用（System call）**。

处理器通常含有一个内部中断屏蔽位，并允许通过软件来设定。一旦被设定，所有外部中断都将被系统忽略。这个屏蔽位的访问速度显然快于中断控制器上的中断屏蔽寄存器，因此可提供更快速地中断屏蔽控制。

中断尽管可以提高计算机处理性能，但 **过于密集的中断请求／响应反而会影响系统性能。这类情形被称作中断风暴（interrupt storm）**。