



链滴

线程的上下文切换 -- 《深入理解 java 虚拟机》

作者: [HuanYuanHe](#)

原文链接: <https://ld246.com/article/1576372221766>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

对于单核CPU来说（对于多核CPU，此处就理解为一个核），CPU在一个时刻只能运行一个线程。当在运行一个线程的过程中转去运行另外一个线程，这个叫做**线程上下文切换**（对于进程也是类似）。> 在 Windows 操作系统上，每个线程要预留出 1m 的内存空间，意味着 2G 的内存理论上最多只能创建 2048 个线程。而在 Linux 上，最大线程数由常量 `PTHREAD_THREADS_MAX` 决定，一般为 024。

由于可能当前线程的任务并没有执行完毕，所以在切换时需要保存线程的运行状态，以便下次重新切换回来时能够继续切换之前的状态运行。举个简单的例子：比如一个线程A正在读取一个文件的内，正读到文件的一半，此时需要暂停线程A，转去执行线程B，当再次切换回来执行线程A的时候，我不希望线程A又从文件的开头来读取。

在理解这个之前，要理解程序计数器的作用（以下摘抄来自《深入理解java虚拟机》）：

程序计数器是一块较小的内存空间，它可以看作是当前线程所执行字节码的行号指示器。在虚拟的概念模型里（仅是概念模型，各种虚拟机可能会通过一些更高效的方式去实现），字节解释器工作就是通过改变这个计数器的值来选取下一条需要执行的字节码指令。且由于java虚拟机的多线程是通过线程轮流切换并分配器执行时间的方式来实现的，在任何一个确定的时刻，一个处理器（对于多核处理器是一个内核）都只会执行一条线程中的指令，因为为了线程切换后能恢复到正确的执行位置，每条线程都需要有一个独立的程序计数器，各条线程之间计数器互不影响，我们称这类区域为“线程私有”内存。

线程切换时需要知道在这之前当前线程已经执行到哪条指令了，所以需要记录程序计数器的值，外比如说线程正在进行某个计算的时候被挂起了，那么下次继续执行的时候需要知道之前挂起时变量值时多少，因此需要记录CPU寄存器的状态。所以一般来说，线程上下文切换过程中会记录程序计数、CPU寄存器状态等数据。