线程调度

计算机通常只有一个CPU,在任意时刻只能执行一条机器指令,每个[线程](https://baike.baidu.com/item/%E7%BA%BF%E7%A8%8B" \t "_blank)只有获得CPU的使用权才能执行指令.所谓多线程的[并发运行](https://baike.baidu.com/item/%E5%B9%B6%E5%8F%91%E8%BF%90%E8%A1%8C" \t "_blank),其实是指从宏观上看,各个线程轮流获得CPU的使用权,分别执行各自的任务.在运行池中,会有多个处于就绪状态的线程在等待CPU,JAVA[虚拟机](https://baike.baidu.com/item/%E8%99%9A%E6%8B%9F%E6%9C%BA)的一项任务就是负责线程的调度,线程调度是指按照特定机制为多个线程分配CPU的使用权.

简介

有两种调度模型：分时调度模型和[抢占式](https://baike.baidu.com/item/%E6%8A%A2%E5%8D%A0%E5%BC%8F" \t "_blank)调度模型。

分时调度模型是指让所有的[线程](https://baike.baidu.com/item/%E7%BA%BF%E7%A8%8B" \t "_blank)轮流获得cpu的使用权,并且平均分配每个线程占用的CPU的时间片这个也比较好理解。

java虚拟机采用抢占式调度模型，是指优先让可运行池中优先级高的线程占用CPU，如果可运行池中的线程优先级相同，那么就随机选择一个线程，使其占用CPU。处于运行状态的线程会一直运行，直至它不得不放弃CPU。

放弃CPU原因

一个[线程](https://baike.baidu.com/item/%E7%BA%BF%E7%A8%8B)会因为以下原因而放弃CPU。

1 java虚拟机让当前线程暂时放弃CPU，转到就绪状态，使其它线程获得运行机会。

2 当前线程因为某些原因而进入阻塞状态

3 线程结束运行

需要注意的是，线程的调度不是跨平台的，它 不仅仅取决于java虚拟机，还依赖于操作系统。在某些操作系统中，只要运行中的线程没有遇到阻塞，就不会放弃CPU；在某些操作系统中，即使线程没有遇到阻塞，也会运行一段时间后放弃CPU，给其它线程运行的机会。

java的线程调度是不分时的，同时启动多个线程后，不能保证各个线程轮流获得均等的CPU时间片。

如果希望明确地让一个[线程](https://baike.baidu.com/item/%E7%BA%BF%E7%A8%8B)给另外一个线程运行的机会，可以采取以下办法之一。

调整各个线程的优先级

让处于运行状态的线程调用Thread.sleep()方法

让处于运行状态的线程调用Thread.yield()方法

让处于运行状态的线程调用另一个线程的join()方法

线程切换：不是所有的线程切换都需要进入内核模式