

内核态用户态是什么？

操作系统对程序的执行权限进行分级, 分别为用户态和内核态。用户态相比内核态有较低的执行权限, 很多操作是不被操作系统允许的, 简单来说就是用户态只能访问内存, 防止程序错误影响到其他程序, 而内核态则是可以操作系统的程序和普通用户程序

内核态: cpu可以访问计算机所有的软硬件资源

用户态: cpu权限受限, 只能访问到自己内存中的数据, 无法访问其他资源

为什么要有用户态和内核态？

系统需要限制不同的程序之间的访问能力, 防止程序获取不相同程序的内存数据, 或者外围设备的数据, 并发送到网络, 所有cpu划分出两个权限等级用户态和内核态

Java线程是用户态还是内核态

Java线程在jdk1.2之前, 是基于称为“绿色线程”的用户线程实现, 但从JDK1.3起, 主流平台上的“主流”商用Java虚拟机的线程模型普遍都被替换为基于操作系统原生线程模型来实现, 即采用1:1的线程模型

Java线程用户态内核态切换

cpu采用时间轮片机制给每个线程分发执行时间片来让线程执行, 如果线程在进行上下文切换, 堵塞, 时间片用完, 睡眠, \等情况, 就会发生用户态和内核态的切换

线程切换的调度器存放于内核中, 中断处理也在内核。 , 从这个角度来说, 如果线程当前处于用户态, 若要发生线程切换, 必然是要先进入内核态, 发生状态切换的。线程切换的原因还可能是其他类型的中断, 或者线程自身主动进入等待和睡眠. 这些情况无一例外会进入内核。

但是线程当前就处于内核态, 也可能一样中断, 或者主动进入等待或者睡眠, 这也是一样会发送的, 但就是没有了用户态和内核态的状态切换了

总而言之线程切换和状态切换没有直接的关联, , 只不过线程切换一般都放到内核实现而已, 当然也还有所谓的用户级线程, 也就是在用户态直接切换线程的栈和寄存器而已, 这也无需切换到内核态

