线程

用户级别线程：

内核级别线程：

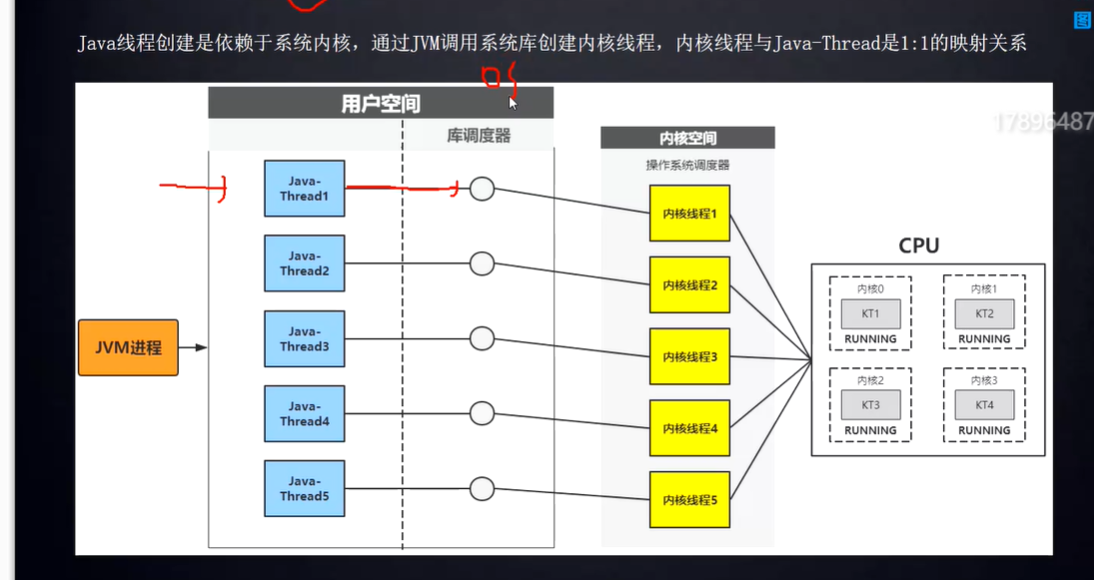
操作系统将内存分为内核空间和用户空间，内科空间为操作系统专用，用户空间是给软件用的，操作系统可以访问两个空间，软件只能访问用户空间，这样做防止外来软件访问内核空间从而伤害操作系统

每一个软件是一个进程，每个进程里面有多个线程，称为用户线程，用户线程的创建不依赖于操作系统，并且状态切换不依赖操作系统，所以切换比较快，缺点就是只能使用一个cpu的一个内核，而内核线程为操作系统的线程，每创建一个用户线程都会在内核线程里面维护一个线程表，内核线程用于实现操作系统的调度和工作，缺点是线程状态切换会需要cpu内核进行切换，优点就是可以利用cpu的多核心，可以多线程并行

简单说

用户线程（ult）是线程的创建阻塞销毁都由用户维护，内核线程（klt）则由内核维护

Java虚拟机用的是内核线程



因为java的线程都依赖于内核线程，直接和cpu相关，可以多线程并行也需要线程调度和线程切换而且消耗很大，所以在线程过多的时候反而会在线程调度的时候消耗大量资源，因此需要合理控制线程数量。

从上可知线程是非常宝贵的资源，并且需要合理管理线程数量，因此线程池非常重要，可以合理管理线程和线程复用，

线程池适用

单个任务处理时间短

任务数量多

ctrl+shift+alt+u 得到类的结构图

线程数量过多处理不过来的时候，拒绝策略里面可以将线程写入redis，高峰过去之后再取出执行

线程池执行业务代码的时候是否会有多线程的问题，

线程池线程参数如何设置