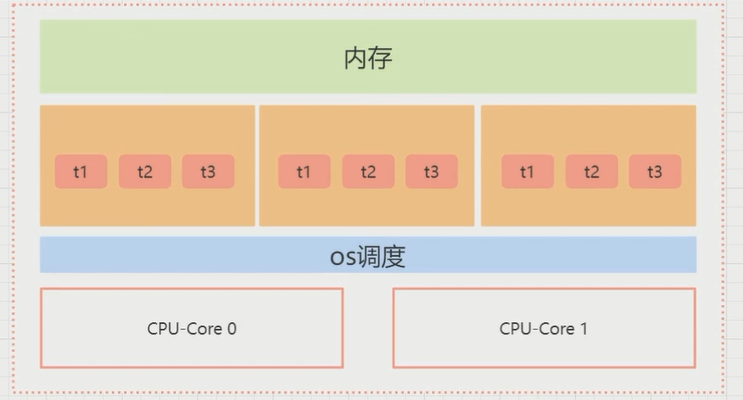
并发编程基础：



什么情况下使用多线程

1. 通过并行计算提高程序性能
2. 等待网络、IO响应导致的耗时问题

Thread /Runable /Callable,future

线程基础

线程的六种状态：

***NEW 没有调用start方法***

***RUNNABLE 运行状态 可运行的 可能在执行 也可能在等待执行***

***BLOCKED 阻塞 发生等待 等待monitor***

***等待阻塞 wait***

***同步阻塞 synchronized***

***其他阻塞 sleep/join***

***WAITING 等待***

***TIMED\_WAITING 时间等待 超时等待***

***TERMINATED 终止***

线程的启动和中断

Start native

Interrupt

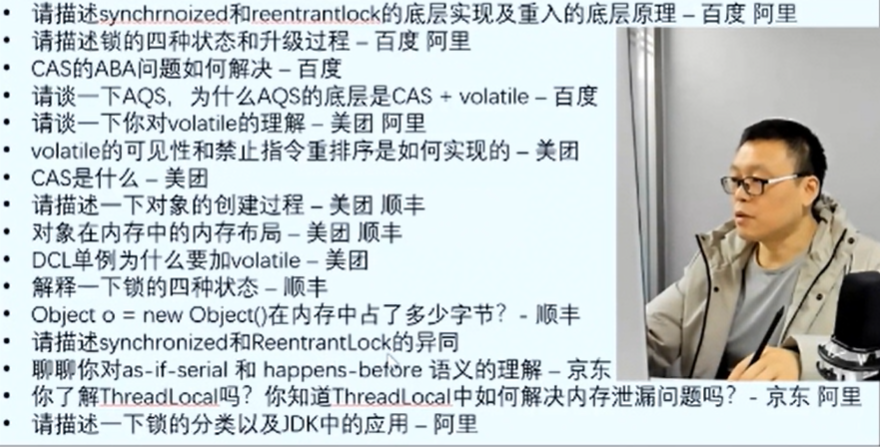
通过指令的方式：volatile boolean isStop = false

Thread.interrupted 对设置的中断标识的线程进行复位

线程的安全问题

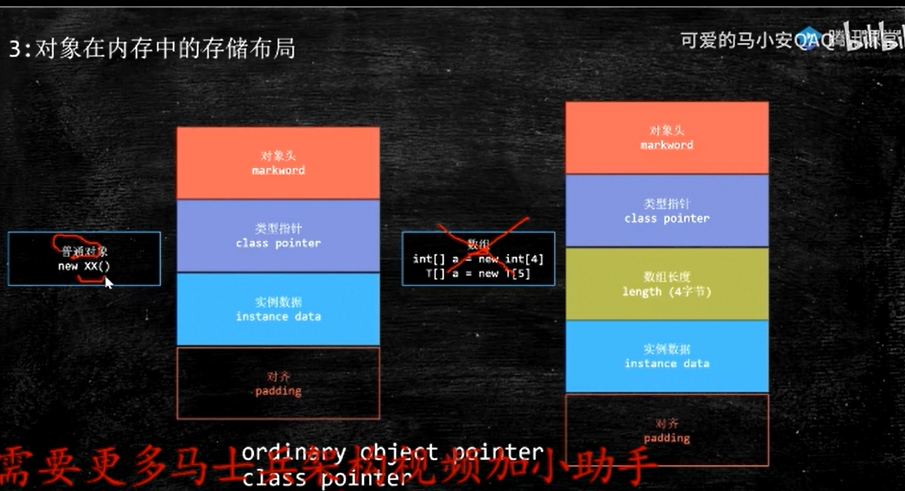
可见性、原子性、有序性

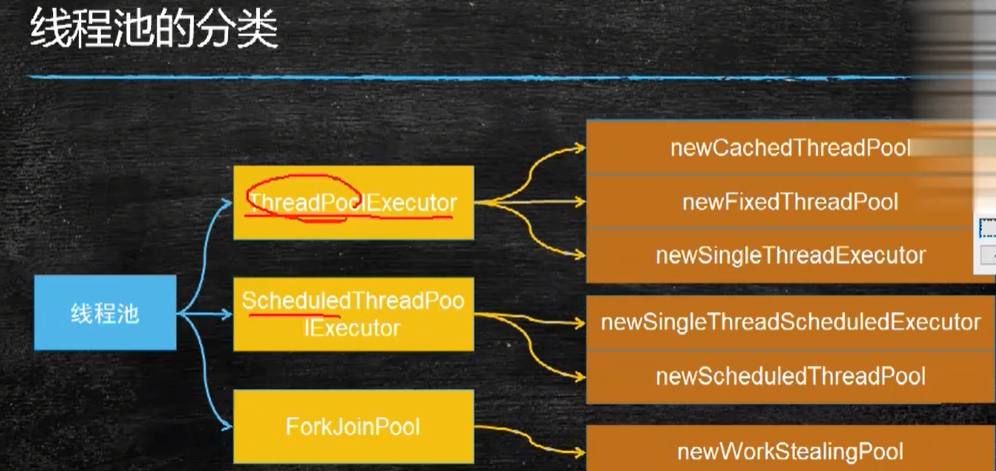
为了屏蔽硬件和操作系统的差异，所以定义了JMM 规范

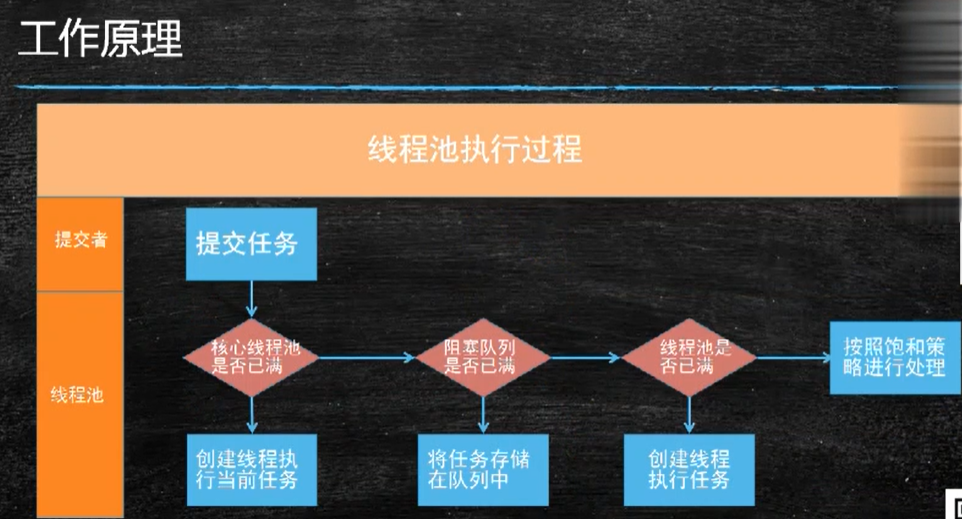


Cas 最终实现

Lock cmpxchg 指令







NewWorkStealingPool 可以根据当前的cpu核心

