多核编程与单核多线程编程的区别

# 1. 锁竞争

单核中，如果单个线程取得锁的控制权，则会获得CPU的运行时间，其它等待获取锁的线程就会阻塞。使用了锁，影响的只是加锁和解锁的耗时，CPU始终运行。

多核中，若2个（更多）线程使用同一把锁，则会导致CPU饥饿。实际还是串行化执行！

# 2. 线程分解和执行的区别

对单核CPU，对客户端软件，采用多线程，主要是创建多线程将一些计算放在后台执行，而不影响用户交互操作。（用户界面&其它计算并发执行）提高用户的操作性能！

多核中，分解多个线程，不在限于将用户界面操作和其它计算分离。是为了让计算分配到各个CPU上执行，执行线程数量与CPU核数有关！如果线程数小于核数，某些CPU肯定处于空闲状态

# 3. CPU多核负载均衡

单核中不考虑负载均衡，各个线程计算量相差很大，也不会影响程序总计算时间。

多核中，必须考虑将各个线程计算量均衡到各CPU上。

# 4. 任务调度策略区别

单核中，任务调度主要考虑分时，某些任务的优先执行！常用的方法有：时间片轮转，优先级抢占！

多核中，任务调度比单核有新的需求。要考虑不同任务的耗时，和计算均衡！不能采用简单的时间片轮转和优先级抢占。而总的调用操作系统不能掌握，需要程序员实现！

# 5. CPU cache存取

CPU读取Cache时，以行为单位读取。如果2个硬件线程的两块不同内存位于同一Cache行。当2个硬件线程同时对各自的内存进行写操作，导致2个线程同时写一个Cache行。冲突！！[伪内存问题]。

单核机，不存在伪内存问题

多核机，存在伪内存问题。想办法，使不同的内存块映射到不同的Cache行。

# 6. 任务优先级抢先的问题

单核CPU中，优先级抢占很常见的任务调度策略。

多核CPU中，多核可导致，低优先级和高优先级的任务同时运行。需要进一步改进优先级调度策略。

# 7. 串行计算，并行计算与分布式计算的区别

单核多线程编程中，都是串行算法。用不到并行计算，更用不到分布式计算。

多核编程中，多采用并行计算和分布式计算。

并行计算=并行设计模式+并行算法

分布式计算复杂度 > 并行计算的复杂度 > 串行计算复杂度

并行计算：只考虑计算并行地执行，不考虑线程间的竞争导致CPU饥饿

分布式计算：相比并行计算，能更好解决CPU饥饿，使计算均衡地分配任务到各内核中