

咱们谈谈 Java 并发编程

作者: [JackLei](#)

原文链接: <https://ld246.com/article/1660663005568>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

咱们谈谈Java并发编程

1.为什么要有并发编程?

在现在计算机硬件在高速发展，多核计算机已经成为时代的发展主流，**为了提高多核CPU的利用率**。之提升程序的性能，如：响应时间、吞吐量、计算机资源使用率等。

2.什么是并发编程?

所谓并发编程是指在一台处理器上“**同时**”处理多个任务。

3.我们了解了解Java并发编程!

3.1首先咱们要谈谈什么是进程、什么是线程!

进程：是正在运行的程序;是系统进行资源分配和调度的基本单位!

线程：单个线程是进程（正在运行的程序）的一个**任务**；是系统进行资源能够进行调度的**最小单位**!

进程和线程的**区别**：

1.先有**进程**，后有**线程**；

2.**进程**是基本单位；**线程**是最小单位；一个进程中包含**多个**线程；

3.默认情况下，**进程**之间的内存**无法共享**，**线程**之间的**共享**父进程的内存。

3.2咱们接下来谈谈什么是串行、并行、并发?

串行：

按序执行；第一个任务没有执行完第二个任务不能执行；

并行：

同时执行，多管齐下；在多核CPU下，有两个或两个以上的任务同时执行；

并发：

穿插执行，减少等待；多个任务穿插轮流执行，实质是一个CPU在若干个程序之间**多路复用**，目的就是高有限资源的利用率。

3.3并发编程的三个核心（分工、协作、互斥）

分工：

****把一个任务分解成多个步骤（小任务），在什么时候谁完成那个任务。**任务分解和分工对于项目成败是非常关键，不过在并发领域中，分工更加重要，它直接决定了并发程序的性能。（在Java中为分提供很多方法：Executor、Fork/Join、Future；总结的设计模式：生产者-消费者、WorkerThread）；**

协作:

实质是一个任务完成了，如何通知后续的任务开始执行。（例如：我们经常使用的**生产者-消费者**式，当队列是满时，生产者就需要等待，当队列不满时，**通知**生产者开始工作；当队列空时，消费者需要等待，当队列不空时，**通知**消费者开始工作）；

互斥:

分工、协作主要强调的是性能，但并发程序里还有一部分关于**正确性的，线程安全**。当多个线程访问个共享变量时，结果往往不能确定的。就可能导致一些问题。导致问题的只要三个主要源头是：可性、有序性、原子性；所以Java引入了**内存模型**，模型提供了一些规则，利用规则，我们可以**避免可性问题、有序性问题、但是不能完全解决线程安全问题**；解决线程安全的问题的核心还是**互斥**。实现互斥的核心还是“**锁**”；（Java里的锁 synchronized、Lock），互斥的思想是：**同一时刻，只允许一个线程“访问共享变量**。