链滴

高并发程序设计（1）——基础

作者：wanglei0622
原文链接：https://ld246.com/article/1488425635036
来源网站：链滴

<h3>同步(synchronous)与异步(asynchronous)</h3>

同步：同步方法一旦开始执行，调用者必须等到方法调用返回后，才能继续后续行为。
异步：通常会在另外一个线程中真实的执行，不会阻碍调用者继续工作。

<h2>并发(concurrency)与并行(parallelism)</h2>

并发：偏重与多个任务交替执行，而多个任务中可能还是串行的。
并行：真正意义上的同时执行。

<h2>原子性(atomicity)</h2>

原子性是指一个操作是不可中断的，即使是多个线程一起执行，一个操作一旦开始就不会被其他程干扰。

int类型的变量的赋值操作就是原子性的，就算两个线程对他进行赋值，一个1一个-1，那他的结不是1就是-1。
long类型的变量在32位的操作系统里就不是原子的，因为long是8个字节，赋值需要2步，读写不是原子的，值就有可能被另外的线程干扰。

<h2>可见性(visibility)</h2>

多个线程对一个共享变量进行操作时，由于编译器或者硬件优化的缘故，a程修改了变量的值，但是b线程缓存了变量原来的值，读取的就是cache中或者寄存器里的据。
指令重排也会导致一个线程的修改，另一个线程不会立即察觉。一个线程去观察另外一个线程的变量，他们的值能否或者何时能读取到是不能保证的！

<h2>指令重排与有序性(ordering)</h2>

对单线程的代码，我们习惯性的认为代码执行的顺序是按代码顺序执行，其实并非如此，编译器代码编译时会进行指令重排，重拍后的指令顺序不一定和原来一样。
指令重排可以保证串行程序语义一致，但是不保证多线程间语义也一致。
指令重排有原则

<h3>线程与进程</h3>

线程是轻量级进程，是程序（进程）执行的最小单位
线程生命周期，NEW,RUNNABLE,BLOCKED,WAITING,TIMED_WAITING,TERMINATED 6个态

NEW:刚创建的线程，还没有执行，也就是还没有调用start()方法。
RUNNABLE：调用start()后，进入该状态。
BLOCKED:如果执行过程中遇到syschronized同步块，且是锁已经别先进入的线程获得，没有释，则进入阻塞状态。
WAITING:进入无时间限制的等待，直到被notify()，如果是通过join()方法则等目标线程终止后继续执行,进入RUNNABLE状态。
TIMED_WAITING:等待时间到了后，自动执行。
TERMINATED:线程执行完毕，或者调用stop()，进入该状态，但是不建议用stop()，太暴力，会成代码执行逻辑错误，影响数据，建议通过interrupt()方法设置中断标志，通过判断是否设置了中断志来结束线程。

