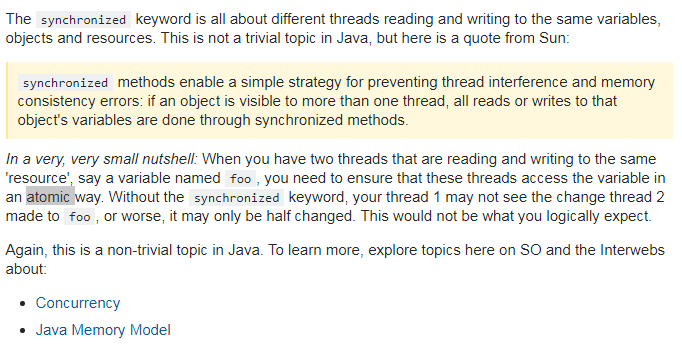
原子性:

这些操作是不可被中断的，要么执行，要么不执行.

y = x; // 这个赋值操作不是原子性,因为首先要读取x的值然后再赋值给y,是两个操作.

Synchronized 阻塞式同步



synchronized方法启用了一个简单的策略来防止线程干扰和内存一致性错误：如果一个对象对多个线程可见，则对该对象变量的所有读取或写入都是通过同步方法完成的。

多线程有一个共享变量a=0 线程A首先获取锁,修改a=1,在线程A操作结束后将a=1传给线程B.否则线程不能进入,保持同步.

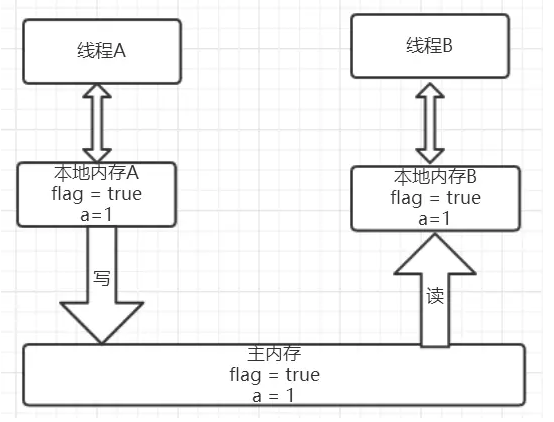
Volatile

被volatile修饰的变量能够保证每个线程能够获取该变量的最新值，从而避免出现数据脏读的现象**。**

实现原理:

1. Lock前缀的指令会引起处理器缓存写回内存
2. 一个处理器的缓存回写到内存会导致其他处理器的缓存失效
3. 当处理器发现本地缓存失效后,就会从内存中重读该变量数据,即可以获取当前最新值

图例:



问题:

Volatile不能保证对变量的任何操作都是原子性,如果线程阻塞就不会去修改内存,最终也会出现问题.