关于编程路上的一些杂谈 由线程的通信原理想到的 (一)

作者：whxiaobu

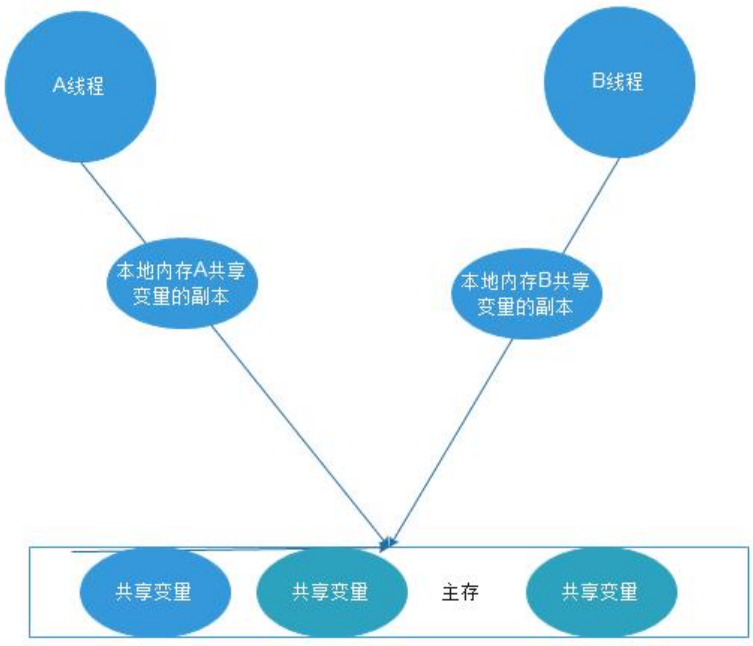
原文链接：https://ld246.com/article/1483854785581

来源网站：链滴

- 来源： 关于编程路上的一些杂谈 由线程的通信原理想到的(一)
- 作者：知秋

写这个其实想了很多，到底该用什么方式来写，最后还是按照自己的随意来随意的拉拉家常算了。时听说身边和网友说自己遇到瓶颈，接下来不知道该往哪里走，我给的建议就是：请回归基础。

为何这样说，先拿一个大家都知道的东西来说，并发编程中 ，线程之间如何通信及线程之间如何同步 ，线程之间的通信机制有两种：共享内存和消息传递 ，如图：



如果线程A与线程B之间要通信的话，必须要经历下面2个步骤：

1. 线程A把本地内存A中更新过的共享变量刷新到主内存中去。
2. 线程B到主内存中去读取线程A之前已更新过的共享变量。

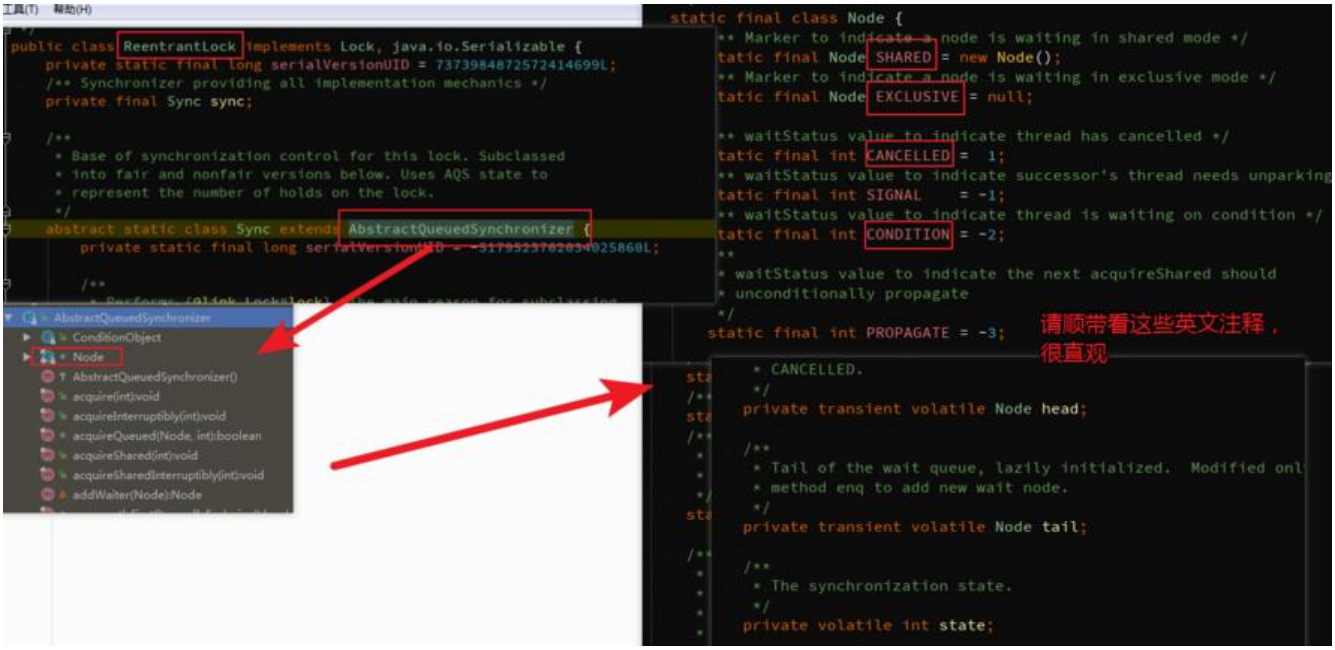
好了，就言仅于此，接下来我们跳过这里看一个让人感觉离的比较远的东西。

就拿大家觉得比较在上层的**zookeeper**来讲，**分布式锁**这个概念相信不少人都听过，Linux文件系统组成相信大家也一定知道点，最起码的是这个文件系统是以树的形式组织起来的,知道这个其实你就已理解了三分之一了，因为**zookeeper**的数据结构也是这么组织的，也是一棵树的形式，其实大家很自就想到了b+树这个概念，这也是面试里为何会经常问到b+树到底是个什么东西，回到正题，当大家谈分布式这个概念的时候，往往就会觉得头大，不想触碰，我觉得无非就是几份同样的代码给分放到同的机器里来运行，各自有各自的web服务器，而我们可以很自然的把相应的服务器看成是一个**容器**也就是上下文，其实也就是上图里A和B线程相当于是两台机器，而共享变量的副本其实就是各自机器的上下文而已，这么看来，**所谓的分布式不也就是我们在单机上操作的多线程么**，又有何高大上可言。

既然搞分布式，那么我们接下来要解决的就是两台机器间的通信问题，也就是线程之间如何通信及线之间如何同步变成了分布式环境下两台机器如何进行通信和同步，同样的，我们需要一个所谓的**主存**这里的主存就是zookeeper,这里先对大致的原理说下，在以后的文章里会对细节各种说道的，使用过ookeeper的都知道，我们可以通过相应的API(个人推荐Curator)在zookeeper里创建一个临时的节，一个操作session结束这个节点的生命周期也就结束销毁了,利用这个特性，我们在A机器修改某一

共享变量的时候，会先在zookeeper里获取下有没有这个节点，没有则创建，有则等待(因为是同一份码，所以创建的节点也会相同)，这样便不会对共享变量同时进行多次操作，具体的实现细节以后再说其实这个api的写法和Java底层的源码没多大区别，用法也没多少区别。

其实所说的就是分布式锁的原理，其实我们是可以**从volatile 的语义联系到锁的语义**的，拿Reentrantock来说，其底层也无非是维护一个节点(请看下图)，用的也是volatile语义，再回头看看zookeeper这种操作方式，难道还有疑虑么，所谓的抽象出来的高度的东西，其实还是底层的一些实现，用的代也逃不出底层代码那些套路，包括zookeeper里的barrier，计数器等等。



这次就暂时说到这里了，篇幅长了大家也没人看，下一篇想想，最近两个月一直在用通用mapper，如来讲下通用mapper是如何根据mybatis生成的东西做的一系列组合封装，方便大家项目中使用相的思想，后面依然会回到基础和上层一些架构框架之间的联系上来，不仅仅是多线程这块的东西。