

概念普及

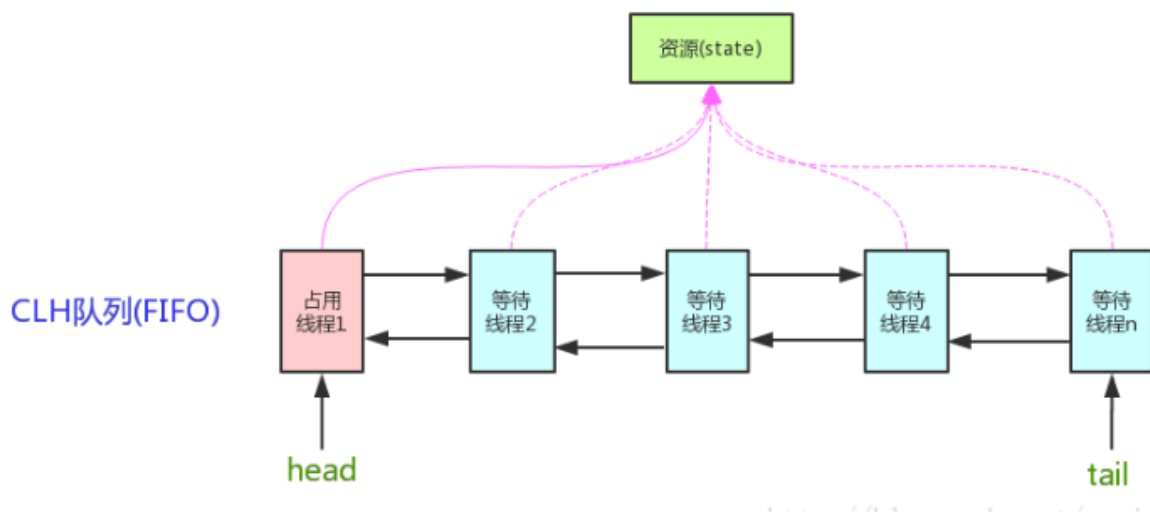
羊群效应：当有多个线程去竞争同一个锁的时候，假设锁被某个线程占用，那么如果有成千上万个线程在等待锁，有一种做法是同时唤醒这成千上万个线程去竞争锁，这时候就发生了羊群效应，海量的竞争必然造成资源的剧增和浪费。

ReentrantLock是可重入锁，可重入锁就是当前持有该锁的线程能够多次获取该锁，无需等待。可重入锁的实现是从ReentrantLock的一个内部类Sync的父类谈起，Sync的父类是AbstractQueuedSynchronizer（AQS）

AQS是一个基于FIFO等待队列实现的一个用于实现同步器的基础框架，JUC（current包）里面几乎所有有关锁、多线程并发以及线程同步器等重要组件的实现都是基于AQS这个框架的。

AQS的核心思想是：如果被请求的共享资源空闲，则将当前请求资源的线程设置为有效工作线程，并且将共享资源设置为锁定状态。如果被请求的共享资源被占用，那么就需要一套线程阻塞等待以及被唤醒时锁分配机制，这个机制AQS是用CLH队列实现锁的，即将暂时获取不到的锁线程加入队列中。

CLH队列是一个虚拟的双向队列（虚拟的双向队列即不存在队列实例，仅存在节点之间的关联关系）。AQS是将每条请求共享资源的线程封装成一个CLH锁队列的一个结点（Node）来实现锁的分配。



AQS，它维护了一个volatile in state（代表共享资源）和一个FIFO线程等待队列（多线程争用资源的时候被阻塞进入此队列）。资源的访问方式：

getState() setState() compareAndSetState()。

AQS定义两种资源共享方式：Exclusive（独占，只有一个线程能执行，如ReentrantLock）和share（共享，多个线程可同时执行，如Semaphore）。

- `isHeldExclusively()`：该线程是否正在独占资源。只有用到condition才需要去实现它。
- `tryAcquire(int)`：独占方式。尝试获取资源，成功则返回true，失败则返回false。
- `tryRelease(int)`：独占方式。尝试释放资源，成功则返回true，失败则返回false。
- `tryAcquireShared(int)`：共享方式。尝试获取资源。负数表示失败；0表示成功，但没有剩余可用资源；正数表示成功，且有剩余资源。
- `tryReleaseShared(int)`：共享方式。尝试释放资源，成功则返回true，失败则返回false。

有一些细节问题在里面：（重点关注并发问题）

<http://www.importnew.com/24006.html>