JDK1.6之前syn是一个重量级锁，他需要调用操作系统的函数。

操作系统将线程停止，这个过程系统需要从{**用户态}**转换为{**内核态}**。所以执行效率低。

JDK1.7之后，线程同步概念尽量在JVM级别解决了，减少了操作系统的操作。

ReentrantLock

维护静态内部类aqs队列（公平锁，和非公平锁），用成员变量sync储存，在构造方法中构建。

lock方法空壳方法，直接调用acquire(int args)方法。

tryAcquire(args)方法，

getState获取锁状态，

hasQueuePredecessors()//判断队列是否为空

compareAndSetState()//加锁

setExclusiveOwnerThread(current);//将当前拿到锁的线程记录到aqs中的exclusiveOwnerThread。

可重用锁，获取锁，如果锁已经被占用，查看拿到锁的线程exclusiveOwnerThread是否为当前线程

是当前线程就把锁自加

获取锁失败时，入队，通过cas将当前线程节点设置为尾节点。在和原尾节点连接

aqs

维护一个Thread exclusiveOwnerThread，用来存储当前持有锁的线程

维护静态内部类Node对象，用head（头）和tail（尾）

Node中的属性，prev和next的node。和thread节点存储的线程。

waitStatus等待情况，在加解锁是判断等级

hasQueuedPredecessors方法，判断aqs队列是否为空方法

1.判断头和尾是否相等