链滴

CyclicBarrier 与 CountDownLatch

作者：614756773

原文链接：https://ld246.com/article/1595747988466

来源网站：链滴

原文链接：CyclicBarrier 与 CountDownLatch

功能

- CyclicBarrier
  - 调用await方法休眠线程,当所有线程都执行到达await点时,唤醒所有线程然后才执行后续代码
  - 唤醒所有线程后,CyclicBarrier进入下一个周期　意味着可以重复使用
- CountDownLatch
  - 调用await休眠线程,当调用指定次数countDown后唤醒所有await的线程
  - 执行了唤醒所有线程后,await方法就会失效　意味着不可以重复使用

CyclicBarrier可以重复使用的原因

- 关键代码　->await->dowait->nextGeneration

```
private int dowait(boolean timed, long nanos)
    throws InterruptedException, BrokenBarrierException,
           TimeoutException {
    final ReentrantLock lock = this.lock;
    lock.lock();
    try {
        final Generation g = generation;
        if (g.broken)
            throw new BrokenBarrierException();
        if (Thread.interrupted()) {
            breakBarrier();
            throw new InterruptedException();
        }

        int index = --count;
        if (index == 0) {  // tripped
            boolean ranAction = false;
            try {
                final Runnable command = barrierCommand;
                if (command != null)
                    command.run();
                ranAction = true;
                nextGeneration();
                return 0;
            } finally {
                if (!ranAction)
                    breakBarrier();
            }
        }
        ......
}
-----------------------------------------------
private void nextGeneration() {
    // signal completion of last generation
    trip.signalAll();
    // set up next generation
```

原文链接：CyclicBarrier 与 CountDownLatch

```
    count = parties;
    generation = new Generation();
}
```

- 最终会执行　nextGeneration方法
  - 该方法会唤醒所有的休眠线程
  - 然后又把count重置为初始值　count值用来计数休眠的线程,当count为0时表示该唤醒所有线了

CountDownLatch

Syn

CountDownLatch使用的AQS，主要定义了一个内部类 Sync实现AQS

因为CountDownLatch是共享锁，所以Sync重载的 tryAcquireShared和 tryReleaseShared

当初始化CountDownLatch时需要指定数量，最后这个数量会被 Syn使用，作为AQS的state

流程

- 当调用await方法时，会生成节点插入到AQS的同步队列中然后使用LockSupport挂起线程
- 使用countDown方法时会调用Syn的tryReleaseShared方法将state的值减一，当state被减为0时AQS会调用doReleaseShared唤醒所有的线程