链滴

java n 个线程顺序打印 n 个字符的通用解决方案

作者：mainlove

原文链接：https://ld246.com/article/1475310303434

来源网站：链滴

```
<p>闲来无事，练习一下，主要用重入锁的 condition 解决了 notify 的效率问题</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">import java.util.concurrent.locks.Condition;
</span></span><span class="highlight-line"><span class="highlight-cl">import java.util.co
current.locks.ReentrantLock;
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">public class Main {
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">public static Reent
antLock reentrantLock =new ReentrantLock();
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">//从某个字符开始
</span></span><span class="highlight-line"><span class="highlight-cl"> public static Char
cter next = 'A';
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">public static void
main(String[] args) {
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">    // input
</span></span><span class="highlight-line"><span class="highlight-cl"> Character[] chara
ters = {'A', 'B', 'C', 'D'};
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">    Integer length
 characters.length;
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">    Condition[] con
itions = new Condition[length];
</span></span><span class="highlight-line"><span class="highlight-cl">    for (int j = 0; j &
t; length; j++) {
</span></span><span class="highlight-line"><span class="highlight-cl">        conditions[j]
= reentrantLock.newCondition();
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">    for (int j = 0; j &
t; length; j++) {
</span></span><span class="highlight-line"><span class="highlight-cl">        new Thread(
ew Irunable(conditions[j], characters[j], characters[j + 1 &gt;= length ? 0 : j + 1], conditions[j +
1 &gt;= length ? 0 : j + 1])).start();
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span></code></pre>
<p>}</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">class Irunable implements Runnable {
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">public Condition
urrentCondition;
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">public Character c;
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">public Character
extC;
</span></span><span class="highlight-line"><span class="highlight-cl">
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">public Condition a
terCondition;
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">public Irunable(C
ndition currentCondition, Character c, Character nextC, Condition afterCondition) {
</span></span><span class="highlight-line"><span class="highlight-cl">    this.currentCon
ition = currentCondition;
</span></span><span class="highlight-line"><span class="highlight-cl">    this.c = c;
</span></span><span class="highlight-line"><span class="highlight-cl">    this.nextC = nex
C;
</span></span><span class="highlight-line"><span class="highlight-cl">    this.afterCondit
on = afterCondition;
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">@Override
</span></span><span class="highlight-line"><span class="highlight-cl"> public void run() {
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">    while (true) {
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">        try {
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">            Main.reent
antLock.lock();
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">            if (c != Ma
n.next) {
</span></span><span class="highlight-line"><span class="highlight-cl">                current
ondition.await();
</span></span><span class="highlight-line"><span class="highlight-cl">            }
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">            Thread.sle
p(1000);
</span></span><span class="highlight-line"><span class="highlight-cl">            System.out
println(c);
</span></span><span class="highlight-line"><span class="highlight-cl">            Main.next
= nextC;
</span></span><span class="highlight-line"><span class="highlight-cl">            afterCondit
on.signal();
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">        } catch (Inter
uptedException e) {
</span></span><span class="highlight-line"><span class="highlight-cl">            e.printStac
Trace();
</span></span><span class="highlight-line"><span class="highlight-cl">        } finally {
</span></span><span class="highlight-line"><span class="highlight-cl">            Main.reent
antLock.unlock();
</span></span><span class="highlight-line"><span class="highlight-cl">        }
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span></code></pre>
```