

CountDownLatch

C-2 创建: 张林伟, 最后修改: 张林伟 03-30 20:24

CountDownLatch是一个同步工具类。

作用:

能够使一个线程等待其他线程完成各自的工作后再执行。

原理:

CountDownLatch是通过一个计数器来实现的, 计数器的初始值为线程的数量。每当一个线程完成了自己的任务后, 计数器的值就会减1。当计数器值到达0时, 它表示所有的线程已经完成了任务, 然后在闭锁上等待的线程 (调用await()方法阻塞的线程) 就可以恢复执行任务。

使用:

构造方法: CountDownLatch(int)传入计数值

countDown()使计数值减一

await()使调用该方法的线程处于等待状态

await(long, TimeUnit)

demo

^ 代码块

Java

```
1 public class CountDownLatchDemo {
2
3     // count为3则main线程会一直等待
4     static CountDownLatch countDownLatch = new CountDownLatch(2);
5
6     public static void main(String[] args) throws InterruptedException {
7         Thread t1 = new Thread() -> {
8             System.out.println("t1 count start: " + countDownLatch.getCount());
9             countDownLatch.countDown();
10            System.out.println("t1 count end: " + countDownLatch.getCount());
11        };
12        Thread t2 = new Thread() -> {
13            System.out.println("t2 count start: " + countDownLatch.getCount());
14            countDownLatch.countDown();
15            System.out.println("t2 count end: " + countDownLatch.getCount());
16        };
17        t1.start();
18        t2.start();
19        countDownLatch.await();
20        System.out.println("main end");
21    }
22 }
```

参考:

《Java并发编程的艺术》第8章