

使用Java多线程实现任务分发Java认证考试 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/644/2021\\_2022\\_\\_E4\\_BD\\_BF\\_E7\\_94\\_A8Java\\_c104\\_644506.htm](https://www.100test.com/kao_ti2020/644/2021_2022__E4_BD_BF_E7_94_A8Java_c104_644506.htm) 多线程下载由来已久，如FlashGet、NetAnts等工具，它们都是依赖于HTTP协议的支持（Range字段指定请求内容范围），首先能读取请求内容（即欲下载的文件）的大小，划分出若干区块，把区块分段分发给每个线程去下载，线程从本段起始处下载数据及至段尾，多个线程下载的内容最终会写入到同一个文件中。只研究有用的，工作中的需求：要把多个任务分派给Java的多个线程去执行，这其中就会有一个任务列表指派到线程的策略思考：已知：1. 一个待执行的任务列表，2. 指定要启动的线程数；问题是：每个线程实际要执行哪些任务。使用Java多线程实现这种任务分发的策略是：任务列表连续按线程数分段，先保证每线程平均能分配到的任务数，余下的任务从前至后依次附加到线程中只是数量上，实际每个线程执行的任务都还是连续的。如果出现那种僧多（线程）粥（任务）少的情况，实际启动的线程数就等于任务数，一挑一。这里只实现了每个线程各扫自家门前雪，动作快的完成后眼见别的线程再累都是爱莫能助。实现及演示代码如下：由三个类实现，写在了一个Java文件中：TaskDistributor为任务分发器，Task为待执行的任务，WorkThread为自定的工作线程。代码中运用了命令模式，如若能配以监听器，用上观察者模式来控制UI显示就更绝妙不过了，就能实现像下载中的区块着色跳跃的动感了，在此定义下一步的着眼点了。代码中有较为详细的注释，看这些注释和执行结果就很容易理解的

。 main ( ) 是测试方法 package com.unmi.common. import java.util.ArrayList. import java.util.List. /\*\* \* 指派任务列表给线程的分发器 \*/ public class TaskDistributor { /\*\* \* 测试方法 \* @param args \*/ public static void main(String[] args) { //初始化要执行的任务列表 List taskList = new ArrayList(). for (int i = 0. i

100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)