

文章搜索

Q

个人资料



涡轮5

+ 加关注

发私信

访问：656178次

积分：5858

等级：BLOG 5

排名：第4308名

原创：86篇 转载：104篇

译文：0篇 评论：73条

文章分类

- 计算机软设计师 (17)
- 法律常识 (3)
- java基础知识 (59)
- 随笔 (9)
- 数据库 (26)
- Myeclipse+Tomcat+SQLserver (14)
- 数据挖掘 (20)
- SSH (7)
- 设计模式 (6)
- Webservice (2)
- 项目经理必备系列 (2)
- 测试 (1)
- 面试 (1)
- 笔试 (10)
- 数据结构系列 (3)
- Linux (1)
- JSP (1)
- http (4)
- sql (2)
- Apache Tomcat (2)

推荐文章

- * CSDN日报20170725——《新的开始，从研究生到入职亚马逊》
- * 深入剖析基于并发AQS的重入锁(ReentrantLock)及其Condition实现原理
- * Android版本的"Wannacry"文件加密病毒样本分析(附带锁机)
- * 工作与生活真的可以平衡吗？

原 Thread的run () 与start () 的区别

标签：thread 多线程 class jdk java 游戏

2012-06-22 20:01 52265人阅读 评论(12) 收藏 举报

分类：

java基础知识 (58)

版权声明：本文为博主原创文章，未经博主允许不得转载。

Java的线程是通过java.lang.Thread类来实现的。VM启动时会有一个由主方法所定义的线程。可以创建新的线程。每个线程都是通过某个特定Thread对象所对应的方法run () 来完成的，方法run()称为线程体。通过调用Thread类的start()方法来启动一个线程。

在Java当中，线程通常都有五种状态，创建、就绪、运行、阻塞和死亡。

第一是创建状态。在生成线程对象，并没有调用该对象的start方法，这是线程处于创建状态。

第二是就绪状态。当调用了线程对象的start方法之后，该线程就进入了就绪状态，但是此时线程调度程序还没有把该线程设置为当前线程，此时处于就绪状态。在线程运行之后，从等待或者睡眠中回来之后，也会处于就绪状态。

第三是运行状态。线程调度程序将处于就绪状态的线程设置为当前线程，此时线程就进入了运行状态，开始运行run函数当中的代码。

第四是阻塞状态。线程正在运行的时候，被暂停，通常是为了等待某个时间的发生(比如说某项资源就绪)之后再继续运行。sleep,suspend, wait等方法都可以导致线程阻塞。

第五是死亡状态。如果一个线程的run方法执行结束或者调用stop方法后，该线程就会死亡。对于已经死亡的线程，无法再使用start方法令其进入就绪。

实现并启动线程有两种方法

- 1、写一个类继承自Thread类，重写run方法。用start方法启动线程
- 2、写一个类实现Runnable接口，实现run方法。用new Thread(Runnable target).start()方法来启动

多线程原理：相当于玩游戏机，只有一个游戏机(cpu)，可是有很多人要玩，于是，start是排队！等CPU选中你就是轮到你，你就run ()，当CPU的运行的时间片执行完，这个线程就继续排队，等待下一次的run ()。

调用start () 后，线程会被放到等待队列，等待CPU调度，并不一定要马上开始执行，只是将这个线程置于可行动状态。然后通过JVM，线程Thread会调用run () 方法，执行本线程的线程体。先调用start后调用run，这么麻烦，为了不直接调用run？就是为了实现多线程的优点，没这个start不行。

1.start () 方法来启动线程，真正实现了多线程运行。这时无需等待run方法体代码执行完毕，可以直接继续执行下面的代码；通过调用Thread类的start()方法来启动一个线程，这时此线程是处于就绪状态，并没有运行。然后通过此Thread类调用方法run()来完成其运行操作的，这里方法run()称为线程体，它包含了要执行的这个线程的内容，Run方法运行结束，此线程终止。然后CPU再调度其它线程。

2.run () 方法当作普通方法的方式调用。程序还是要顺序执行，要等待run方法体执行完毕后，才可继续执行下面的代码；程序中只有主线程——这一个线程，其程序执行路径还是只有一条，这样就没有达到写线程的目的。

记住：多线程就是分时利用CPU，宏观上让所有线程一起执行，也叫并发

[java] view plain copy print ?

```
01. public class Test {
02.     public static void main(String[] args) {
03.         Runner1 runner1 = new Runner1();
04.         Runner2 runner2 = new Runner2();
05.         // Thread(Runnable target) 分配新的 Thread 对象。
```



微信关注CSDN

快速回复

☆ 我要收藏

* 《Real-Time Rendering 3rd》
提炼总结——高级着色：BRDF
及相关技术
* 《三体》读后思考-泰勒展开/维
度打击/黑暗森林

文章存档

2016年10月 (1)

2016年07月 (1)

2016年01月 (1)

2015年12月 (2)

2015年11月 (6)

展开

```
06.         Thread thread1 = new Thread(runner1);
07.         Thread thread2 = new Thread(runner2);
08.         // thread1.start();
09.         // thread2.start();
10.         thread1.run();
11.         thread2.run();
12.     }
13. }
14.
15. class Runner1 implements Runnable { // 实现了Runnable接口, jdk就知道这个类是一个线程
16.     public void run() {
17.         for (int i = 0; i < 100; i++) {
18.             System.out.println("进入Runner1运行状态—————" + i);
19.         }
20.     }
21. }
22.
23. class Runner2 implements Runnable { // 实现了Runnable接口, jdk就知道这个类是一个线程
24.     public void run() {
25.         for (int i = 0; i < 100; i++) {
26.             System.out.println("进入Runner2运行状态===== " + i);
27.         }
28.     }
29. }
```

顶

25

踩

18

上一篇

String类型

下一篇

JAVA集合类总结

相关文章推荐

- java之_Thread(线程)
- python 多线程 start_new_thread()
- new Thread(t);和new Thread(new ThreadStart(t));
- java中thread的start()和run()的区别
- 线程启动start后, 如何停止的?
- Android Thread第二次Thread.start()报错的疑问
- 介绍new Thread的弊端及Java四种线程池的使用
- 使用匿名内部类实现线程, new Thread (new Ru...
- Thread中start()和run()的区别
- 关于多线程run和start的区别

猜你在找

- 【直播】机器学习&数据挖掘7周实训--韦玮
- 【直播】3小时掌握Docker最佳实战-徐西宁
- 【直播】计算机视觉原理及实战-屈教授
- 【直播】机器学习之矩阵--黄博士
- 【直播】机器学习之凸优化--马博士
- 【套餐】系统集成项目管理工程师顺利通关--徐朋
- 【套餐】机器学习系列套餐（算法+实战）--唐宇迪
- 【套餐】微信订阅号+服务号Java版 v2.0--翟东平
- 【套餐】微信订阅号+服务号Java版 v2.0--翟东平
- 【套餐】Javascript 设计模式实战--曾亮

查看评论

12楼 张思思 2017-06-15 20:53发表



博主写的简单易懂, 谢谢

11楼 冯尧 2017-05-22 08:12发表



解释很详细, 简单易懂, 谢谢博主分享。

10楼 superGrit 2017-03-25 10:17发表



博主, 我可以这样认为吗? 多线程增大了程序抢到cpu的概率, cpu的执行速度其实是不变的(这是由硬件决定), 但是从宏观上看本程序在单位时间内被执行的次数变多了, 所以我们感觉程序变快了, 其实这也是量变到质变的一种体现吧

9楼 [superGrit](#) 2017-03-25 10:15发表

博主，我可以这样认为吗？多线程增大了程序抢到cpu的概率，cpu的执行速度其实是不变的（这是由硬件决定），但是从宏观上看本程序在单位时间内被执行的次数变多了，所以我们感觉程序变快了，其实这也是量变到质变的一种体现吧

8楼 [superGrit](#) 2017-03-25 10:08发表

谢谢，看了你的文章解决了我困惑：
“这时主线程无需等待新启动的线程的run方法体代码执行完毕，可以直接继续执行下面的代码；”这句话总结的太到位了

7楼 [chuan_wu](#) 2017-03-24 16:44发表



还不错！

6楼 [AD_Liang](#) 2017-03-24 10:11发表



直接使用下边的：
thread1.run();
thread2.run(); 是不是有问题呀？？？

5楼 [那个发布](#) 2016-12-22 11:44发表



不错

4楼 [sureballse](#) 2016-08-31 09:57发表



写的很好，但是其中有一句话感觉说的不是很清楚，我说一下我的理解。
1.start () 方法来启动线程，真正实现了多线程运行。
“这时主线程无需等待新启动的线程的run方法体代码执行完毕，可以直接继续执行下面的代码；”

3楼 [cyqnaruto](#) 2016-03-24 11:26发表



好文，解决了我的困惑!

2楼 [zhulong2020](#) 2014-06-17 10:49发表



线程就像百米赛跑，start()就是法令枪,每个人都有不同的反应时间，出发也就有了不同的先后顺序，每个人速度不同运行时间也不同。CPU就像赛道宽度，有人在赛道运行，只能等他跑完，才轮到下一个。

1楼 [insistEveryDay](#) 2014-02-11 16:54发表



好文要顶，辛苦LZ

发表评论

用户名: qq_36596145

评论内容:

提交

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-660-0108 | 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 |

江苏乐知网络技术有限公司

京 ICP 证 09002463 号 | Copyright © 1999-2017, CSDN.NET, All Rights Reserved