行云流水的博客

⋮ 目录视图 ₩ 摘要视图



个人资料



starfe-geek

访问: 353145次 积分: 3638 等级: BLOC > 5 排名: 第9126名

原创: 64篇 转载: 52篇 译文: 0篇 评论: 29条

文章搜索

文章分类

java (37) hadoop (9)

shell (13)

安全、加解密 (1)

c语言 (11)

ubuntu (3)

网络 (3)

计算机实用技术 (7)

http (1)

hive (4)

hbase (1) nlp (2)

java web服务器剖析 (5)

JVM (1)

solr (2)

并发编程 (1)

web前端 (2)

netty (2)

分布式计算 (1)

编码 (0)

maven (7) sql (1)

python (3)

linux (8)

分布式系统 (0)

flume (1)

文章存档

2014年10月 (3) 2014年09月 (10) 异步赠书: Kotlin领衔10本好书 免费直播: AI时代,机器学习如何入门? 程序员8月书讯 项目管理+代码托管+文档协作,开发 更流畅

Thread.setDaemon详解

17914人阅读 2014-05-19 16:10

Ⅲ 分类: java (36) 🕶

▮ 版权声明:本文为博主原创文章,未经博主允许不得转载。

java中线程分为两种类型:用户线程和守护线程。通过Thread.setDaemon(fall 户线程;通过Thread.setDaemon(true)设置为守护线程。如果不设置次属性,默认为用户 线程。

用户线程和守护线程的区别:

1. 主线程结束后用户线程还会继续运行,JVM存活;主线程结束后守护线程和JVM的状态 又下面第2条确定。

2.如果没有用户线程,都是守护线程,那么JVM结束(随之而来的是所有的一切烟消云 散,包括所有的守护线程)。

补充说明:

定义:守护线程--也称"**服务线程**",在没有用户线程可服务时会自动离开。 优先级:守护线程的优先级比较低,用于为系统中的其它对象和线程提供服务。

设置:通过setDaemon(true)来设置线程为"守护线程";将一个用户线程设置为守护线程的 方式是在线程启动用线程对象的setDaemon方法。

example: 垃圾回收线程就是一个经典的守护线程,当我们的程序中不再有任何运行的 Thread,程序就不会再产生垃圾,垃圾回收器也就无事可做,所以当垃圾回收线程是JVM 上仅剩的线程时,垃圾回收线程会自动离开。它始终在低级别的状态中运行,用于实时监 控和管理系统中的可回收资源。

生命周期:守护进程(Daemon)是**运行在后台**的一种特殊进程。它独立于控制终端并且 周期性地执行某种任务或等待处理某些发生的事件。也就是说守护线程不依赖于终端,但 是依赖于系统,与系统"同生共死"。那Java的守护线程是什么样子的呢。当JVM中所有的 线程都是守护线程的时候,JVM就可以退出了;如果还有一个或以上的非守护线程则JVM 不会退出。

例子代码:

```
import java.io.IOException;
class TestMain4 extends Thread {
                                                                                关闭
  public void run() {
                              //永真循环线程
      for(int i=0;;i++){
         try {
             Thread.sleep(1000);
         } catch (InterruptedException ex) { }
         System.out.println(i);
      }
  public static void main(String [] args){
                                                  雍和宫招聘会
     TestMain4 test = new TestMain4();
     test.setDaemon(true);
                           //调试时可以设置为false,那么
                                                         true,即可主线程结束,test线程也结束。
```

2014年08月 (6) 2014年07月 (12) 2014年06月 (9) 展开

阅读排行

java中匹配中文的正则表 书写优雅的shell脚本(一 solr安装-tomcat+solrClo Thread.setDaemon详解

免安装版tomcat6.0的配置 (16500)

maven实战 (4) -- mave http的安全方法和幂等性

java socket 以及 流 关闭

maven实战(7)-- 设置n

书写优雅的shell脚本(匹 (9925)

评论排行

光女装版lomcalo.U的配置	(5)
Thread.setDaemon详解	(4)
内部类 final变量的生命质	(2)
solr安装-tomcat+solrClou	(2)
无线连接wzc服务	(2)
书写优雅的shell脚本(匹	(2)
vc6.0卸载重装后缺少项E	(2)
maven实战(7) 设置r	(2)
maven实战(4) mave	(2)
java中匹配中文的正则表	(1)

推荐文章

- * CSDN日报20170828——《4 个方法快速打造你的阅读清单》
- * Android检查更新下载安装
- * 动手打造史上最简单的 Recycleview 侧滑菜单
- * TCP网络通讯如何解决分包粘
- *程序员的八重境界
- * 四大线程池详解

最新评论

maven实战(7)-- 设置maven编

Thread.setDaemon详解 ducati2: Good!

maven实战 (7) -- 设置maven编 不称臣: 我一直在找maven的编 译器,我不知道setting.xml在哪 个位置,能告诉我么

Thread.setDaemon详解 另一只蝴蝶: 感谢

http的安全方法和幂等性 我是一头奔跑的猪: thx

Thread.setDaemon详解 湖心亭看雪: 不错,解决了菜鸟

maven实战(4)-- maven构建自 zytdoit: 按第二种方式 运行命令 后repo里找不到对应文件,按第

```
test.start();
      System.out.println("isDaemon = " + test.isDaemon());
         System.in.read(); // 接受输入,使程序在此停顿,一旦接收到用户输入,main线程结束,守护线
程自动结束
      } catch (IOException ex) {}
  1
```

踩 顶

上一篇 书写优雅的shell脚本(五)-shell中(())双括号运算符

下一篇 jetty源码剖析

相关文章推荐

- Thread.setDaemon设置说明
- Hadoop生态系统零基础入门
- java多线程之守护线程(daemon thread)
- 系统集成工程师必过冲刺!
- 如何向AOSP中添加根证书
- 征服React Native我有妙招
- Python中threading的join和setDaemon的区别及用...
- FFmpeg音视频高级开发实战

- java的守护进程setDaemon
- 5天搞定深度学习框架-Caffe
- Thread.setDaemon设置说明
- Python数据分析经典案例解析
- 搭建Maven私服
- 书写优雅的shell脚本(五)- shell中(())双括号运算...
- 高并发 简述
- Thread.setDaemon详解











杳看评论

4楼 ducati2 2017-07-21 14:46发表



Good!

3楼 另一只蝴蝶 2017-04-26 10:25发表



感谢

2楼 湖心亭看雪 2017-01-02 18:21发表



不错,解决了菜鸟的疑惑



1楼 chs0113 2016-05-06 20:36发表



不错

发表评论

用户名: u011291205 43 评论内容:



雍和宫招聘会

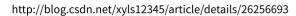












关闭

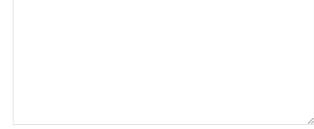
2017/9/7

三种方式 这行报错,哎到底该怎

maven实战(4)--maven构建自 zhhz418418: 按照你的方法测试 成功了! 请问怎么知道第三方包 所依赖的包呢,或者万一第三方 包又依赖另外一个,是不是逐

Thread.setDaemon详解 chs0113: 不错

内部类 final变量的生命周期 任长江: 学习了,final关键字



Thread.setDaemon详解 - 行云流水的博客 - CSDN博客

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

单身公寓



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 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 | webmaster@csdn.net

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

关闭

