

Java高并发编程详解



[Java高并发编程详解_下载链接1](#)

著者:汪文君

出版者:机械工业出版社

出版时间:2018-6

装帧:平装

isbn:9787111599937

本书共分为四个部分：部分详细地介绍了Java多线程的基本用法和各个API的使用，并且着重介绍了线程与Java虚拟机内存之间的关系。第二部分由线程上下文类加载器方法引入，介绍为什么在线程中要有上下文类加载器的方法函数，从而掌握类在JVM的加载和初始化的整个过程。第三部分主要围绕着volatile关键字展开，在该部分中我们将会了解到现代CPU的架构以及Java的内存模型（JMM）。后一部分，主要站在架构设计的高度看待如何巧妙地开发出真正具备并发能力的系统。

作者介绍:

汪文君，汇丰软件（广东）研发中心数据服务中心技术经理、技术专家，目前专注于实时数据data pipeline平台的构建与架构，在加入汇丰软件以前，曾有7年多的移动通信工作经验，以及移动互联网、云计算和B2C电子商务平台的开发架构经验，热衷于技术分享、技术细节锤炼。目前已录制10余套技术视频，在互联网上广泛传播。

目录: 推荐序一

推荐序二

推荐序三

推荐序四

前言

第一部分 多线程基础

第1章 快速认识线程 3

1.1 线程的介绍 3

1.2 快速创建并启动一个线程 3

1.2.1 尝试并行运行 4

1.2.2 并发运行交替输出 5

1.2.3 使用Jconsole观察线程 6

1.3 线程的生命周期详解 7

1.3.1 线程的NEW状态 8

1.3.2 线程的RUNNABLE状态 8

1.3.3 线程的RUNNING状态 8

1.3.4 线程的BLOCKED状态 8

1.3.5 线程的TERMINATED状态 9

1.4 线程的start方法剖析：模板设计模式在Thread中的应用 9

1.4.1 Thread start方法源码分析以及注意事项 9

1.4.2 模板设计模式在Thread中的应用 11

1.4.3 Thread模拟营业大厅叫号机程序 13

1.5 Runnable接口的引入以及策略模式在Thread中的使用 16

1.5.1 Runnable的职责 16

1.5.2 策略模式在Thread中的应用 16

1.5.3 模拟营业大厅叫号机程序 18

1.6 本章总结 19

第2章 深入理解Thread构造函数 20

2.1 线程的命名 20

2.1.1 线程的默认命名 21

2.1.2 命名线程 21

2.1.3 修改线程的名字 22

2.2 线程的父子关系 22

2.3 Thread与ThreadGroup 23

2.4 Thread与Runnable 24

2.5 Thread与JVM虚拟机栈 25

2.5.1 Thread与Stacksize 25

2.5.2 JVM内存结构 27

2.5.3 Thread与虚拟机栈 30

2.6 守护线程 33

2.6.1 什么是守护线程 33

2.6.2 守护线程的作用 34

2.7 本章总结 34

第3章 Thread API的详细介绍 35

3.1 线程sleep 35

3.1.1 sleep方法介绍 35

3.1.2 使用TimeUnit替代Thread.sleep 36

3.2 线程yield	37
3.2.1 yield方法介绍	37
3.2.2 yield和sleep	37
3.3 设置线程的优先级	38
3.3.1 线程优先级介绍	38
3.3.2 线程优先级源码分析	39
3.3.3 关于优先级的一些总结	40
3.4 获取线程ID	40
3.5 获取当前线程	41
3.6 设置线程上下文类加载器	41
3.7 线程interrupt	42
3.7.1 interrupt	42
3.7.2 isInterrupted	43
3.7.3 interrupted	45
3.7.4 interrupt注意事项	46
3.8 线程join	47
3.8.1 线程join方法详解	48
3.8.2 join方法结合实战	50
3.9 如何关闭一个线程	53
3.9.1 正常关闭	54
3.9.2 异常退出	56
3.9.3 进程假死	56
3.10 本章总结	58
第4章 线程安全与数据同步	59
4.1 数据同步	59
4.1.1 数据不一致问题的引入	59
4.1.2 数据不一致问题原因分析	61
4.2 初识 synchronized关键字	62
4.2.1 什么是synchronized	63
4.2.2 synchronized关键字的用法	63
4.3 深入synchronized关键字	65
4.3.1 线程堆栈分析	65
4.3.2 JVM指令分析	67
4.3.3 使用synchronized需要注意的问题	70
4.4 This Monitor和Class Monitor的详细介绍	72
4.4.1 this monitor	72
4.4.2 class monitor	74
4.5 程序死锁的原因以及如何诊断	77
4.5.1 程序死锁	77
4.5.2 程序死锁举例	77
4.5.3 死锁诊断	80
4.6 本章总结	81
第5章 线程间通信	82
5.1 同步阻塞与异步非阻塞	82
5.1.1 同步阻塞消息处理	82
5.1.2 异步非阻塞消息处理	83
5.2 单线程间通信	84
5.2.1 初识wait和notify	84
5.2.2 wait和notify方法详解	87
5.2.3 关于wait和notify的注意事项	89
5.2.4 wait和sleep	90
5.3 多线程间通信	90
5.3.1 生产者消费者	90
5.3.2 线程休息室wait set	93

- 5.4 自定义显式锁BooleanLock 94
 - 5.4.1 synchronized关键字的缺陷 94
 - 5.4.2 显式锁BooleanLock 95
- 5.5 本章总结 104
- 第6章 ThreadGroup详细讲解 105
 - 6.1 ThreadGroup与Thread 105
 - 6.2 创建ThreadGroup 105
 - 6.3 复制Thread数组和ThreadGroup数组 106
 - 6.3.1 复制Thread数组 106
 - 6.3.2 复制ThreadGroup数组 109
 - 6.4 ThreadGroup操作 109
 - 6.4.1 ThreadGroup的基本操作 110
 - 6.4.2 ThreadGroup的interrupt 113
 - 6.4.3 ThreadGroup的destroy 114
 - 6.4.4 守护ThreadGroup 115
 - 6.5 本章总结 116
- 第7章 Hook线程以及捕获线程执行异常 117
 - 7.1 获取线程运行时异常 117
 - 7.1.1 UncaughtExceptionHandler的介绍 117
 - 7.1.2 UncaughtExceptionHandler实例 118
 - 7.1.3 UncaughtExceptionHandler源码分析 119
 - 7.2 注入钩子线程 121
 - 7.2.1 Hook线程介绍 121
 - 7.2.2 Hook线程实战 122
 - 7.2.3 Hook线程应用场景以及注意事项 124
 - 7.3 本章总结 124
- 第8章 线程池原理以及自定义线程池 125
 - 8.1 线程池原理 125
 - 8.2 线程池实现 126
 - 8.2.1 线程池接口定义 127
 - 8.2.2 线程池详细实现 131
 - 8.3 线程池的应用 139
 - 8.4 本章总结 142
- 第二部分 Java ClassLoader
- 第9章 类的加载过程 144
 - 9.1 类的加载过程简介 144
 - 9.2 类的主动使用和被动使用 145
 - 9.3 类的加载过程详解 148
 - 9.3.1 类的加载阶段 148
 - 9.3.2 类的连接阶段 149
 - 9.3.3 类的初始化阶段 154
 - 9.4 本章总结 156
- 第10章 JVM类加载器 158
 - 10.1 JVM内置三大类加载器 158
 - 10.1.1 根类加载器介绍 159
 - 10.1.2 扩展类加载器介绍 159
 - 10.1.3 系统类加载器介绍 160
 - 10.2 自定义类加载器 161
 - 10.2.1 自定义类加载器，问候世界 161
 - 10.2.2 双亲委托机制详细介绍 165
 - 10.2.3 破坏双亲委托机制 167
 - 10.2.4 类加载器命名空间、运行时包、类的卸载等 170
 - 10.3 本章总结 175
- 第11章 线程上下文类加载器 177

11.1 为什么需要线程上下文类加载器	177
11.2 数据库驱动的初始化源码分析	178
11.3 本章总结	180
第三部分 深入理解volatile关键字	
第12章 volatile关键字的介绍	182
12.1 初识volatile关键字	182
12.2 机器硬件CPU	184
12.2.1 CPU Cache模型	184
12.2.2 CPU缓存一致性问题	186
12.3 Java内存模型	187
12.4 本章总结	188
第13章 深入volatile关键字	189
13.1 并发编程的三个重要特性	189
13.1.1 原子性	189
13.1.2 可见性	190
13.1.3 有序性	190
13.2 JMM如何保证三大特性	191
13.2.1 JMM与原子性	192
13.2.2 JMM与可见性	193
13.2.3 JMM与有序性	194
13.3 volatile关键字深入解析	195
13.3.1 volatile关键字的语义	195
13.3.2 volatile的原理和实现机制	197
13.3.3 volatile的使用场景	198
13.3.4 volatile和synchronized	199
13.4 本章总结	200
第14章 7种单例设计模式的设计	201
14.1 饿汉式	201
14.2 懒汉式	202
14.3 懒汉式+同步方法	203
14.4 Double-Check	204
14.5 Volatile+Double-Check	206
14.6 Holder方式	206
14.7 枚举方式	207
14.8 本章总结	208
第四部分 多线程设计架构模式	
第15章 监控任务的生命周期	212
15.1 场景描述	212
15.2 当观察者模式遇到Thread	212
15.2.1 接口定义	212
15.2.2 ObservableThread实现	215
15.3 本章总结	217
15.3.1 测试运行	217
15.3.2 关键点总结	219
第16章 Single Thread Execution设计模式	220
16.1 机场过安检	220
16.1.1 非线程安全	221
16.1.2 问题分析	223
16.1.3 线程安全	225
16.2 吃面问题	225
16.2.1 吃面引起的死锁	226
16.2.2 解决吃面引起的死锁问题	228
16.2.3 哲学家吃面	229
16.3 本章总结	230

- 第17章 读写锁分离设计模式 231
 - 17.1 场景描述 231
 - 17.2 读写分离程序设计 232
 - 17.2.1 接口定义 232
 - 17.2.2 程序实现 234
 - 17.3 读写锁的使用 239
 - 17.4 本章总结 242
- 第18章 不可变对象设计模式 244
 - 18.1 线程安全性 244
 - 18.2 不可变对象的设计 244
 - 18.2.1 非线程安全的累加器 245
 - 18.2.2 方法同步增加线程安全性 247
 - 18.2.3 不可变的累加器对象设计 248
 - 18.3 本章总结 249
- 第19章 Future设计模式 251
 - 19.1 先给你一张凭据 251
 - 19.2 Future设计模式实现 251
 - 19.2.1 接口定义 252
 - 19.2.2 程序实现 253
 - 19.3 Future的使用以及技巧总结 256
 - 19.4 增强FutureService使其支持回调 257
 - 19.5 本章总结 258
- 第20章 Guarded Suspension设计模式 259
 - 20.1 什么是Guarded Suspension设计模式 259
 - 20.2 Guarded Suspension的示例 259
 - 20.3 本章总结 261
- 第21章 线程上下文设计模式 262
 - 21.1 什么是上下文 262
 - 21.2 线程上下文设计 263
 - 21.3 ThreadLocal详解 264
 - 21.3.1 ThreadLocal的使用场景及注意事项 265
 - 21.3.2 ThreadLocal的方法详解及源码分析 265
 - 21.3.3 ThreadLocal的内存泄漏问题分析 270
 - 21.4 使用ThreadLocal设计线程上下文 274
 - 21.5 本章总结 276
- 第22章 Balking设计模式 277
 - 22.1 什么是Balking设计 277
 - 22.2 Balking模式之文档编辑 278
 - 22.2.1 Document 278
 - 22.2.2 AutoSaveThread 280
 - 22.2.3 DocumentEditThread 281
 - 22.3 本章总结 283
- 第23章 Latch设计模式 284
 - 23.1 什么是Latch 284
 - 23.2 CountdownLatch程序实现 285
 - 23.2.1 无限等待的Latch 285
 - 23.2.2 有超时设置的Latch 289
 - 23.3 本章总结 291
- 第24章 Thread-Per-Message设计模式 293
 - 24.1 什么是Thread-Per-Message模式 293
 - 24.2 每个任务一个线程 293
 - 24.3 多用户的网络聊天 296
 - 24.3.1 服务端程序 296
 - 24.3.2 响应客户端连接的Handler 297

24.3.3 聊天程序测试 299
24.4 本章总结 300
第25章 Two Phase Termination设计模式 301
25.1 什么是Two Phase Termination模式 301
25.2 Two Phase Termination的示例 302
25.2.1 线程停止的Two Phase Termination 302
25.2.2 进程关闭的Two Phase Termination 303
25.3 知识扩展 304
25.3.1 Strong Reference及LRUCache 304
25.3.2 Soft Reference及SoftLRUCache 308
25.3.3 Weak Reference 311
25.3.4 Phantom Reference 312
25.4 本章总结 314
第26章 Worker-Thread设计模式 315
26.1 什么是Worker-Thread模式 315
26.2 Worker-Thread模式实现 315
• • • • • ([收起](#))

[Java高并发编程详解_下载链接1](#)

标签

多线程

并发

java

编程

Java

计算机

计算科学

JVM

评论

这是照抄《图解Java多线程设计模式》吧。。。

一下午读完。对juc包的各个核心类讲解还不深入，很多内容点到为止。看过《Java多线程编程实战指南（核心篇）》，本书前14都可以快速翻过。第四部分设计模式有点意思，不少所谓设计模式其实就是对juc包中的一个类的使用。

有许多底层原理之外的东西，比如设计模式。很值得一读。

简洁清晰流畅

写的很好。循序渐进的。~~

好的点：对Java源码的部分API解读的很细致。不好的点：1. 对于糟糕的代码没有明确的提示，很容易让人产生误导，可以学学《Java并发编程实践》那样，给与表情提示。2. 缺少理论支撑，不适合入门级选手 3. 代码风格有不一致的地方，像Thread.sleep(), TimeUnit.SECONDS.sleep()交叉出现，其实作者是推崇后者的··
总之，现在让我对机械出版社出版的图书印象分-1

从底层到顶层均有涉及，深入程度适中。对构建知识体系是个较好的补充。

8年架构之路的我收集了一些java架构资料，现在免费分享给大家~里面有高可用、高并发、高性能及分布式、Jvm性能调优、Spring源码，MyBatis, Netty, Redis, Kafka, Mysql, Zookeeper, Tomcat, Docker, Dubbo, Nginx等多个知识点的架构资料都是免费获取的~加Q群8093/89099 免费获取!!!

我觉得覆盖挺全的，前面基础阶段讲的挺详细，全面的，第二部分蜻蜓点水，不懂，第三部分volatile讲的不错，源码不懂，后面设计模式有点难懂

[Java高并发编程详解_下载链接1](#)

书评

个人感觉这个书的缺点挺多的，有些概念并没有讲得很清楚，目前还在看。不过我觉得里面使用的lambda是一个很好的亮点，虽然不是什么很新的东西，但是对于我来讲，又接触了一些新东西了。总得来讲好坏参半吧，阅读没有什么问题。OK，到此为止了，我只是简单的写一下目前我看这...

[Java高并发编程详解_下载链接1](#)