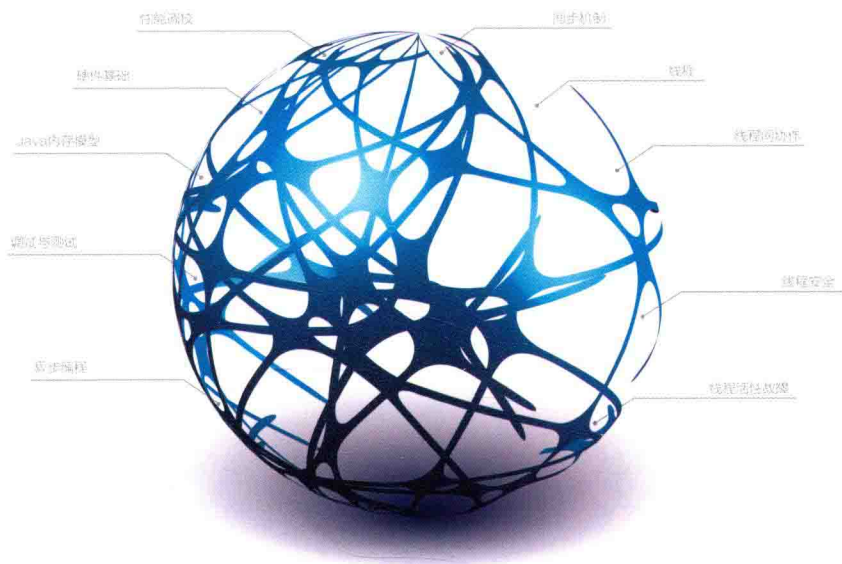


本书以基本概念、原理与方法为主线,辅以丰富的实战案例和生活化实例,从 Java 虚拟机、操作系统和硬件多个层次与角度出发循序渐进、系统地介绍 Java 平台下的多线程编程核心技术及相关工具。

Java 多线程编程 实战指南

(核心篇)

黄文海 / 著



Java多线程编程 实战指南

(核心篇)

黄文海 / 著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

随着现代处理器的生产工艺从提升处理器主频频率转向多核化,即在一块芯片上集成多个处理器内核(Core),多核处理器(Multicore Processor)离我们越来越近了——如今就连智能手机这样的消费类设备都已配备了4核乃至8核的处理器,更何况商用系统!在此背景下,以往靠单个处理器自身处理能力的提升所带来的软件计算性能提升的那种“免费午餐”已不复存在,这使得多线程编程在充分利用计算资源、提高软件服务质量方面扮演了越来越重要的角色。故而,掌握多线程编程技能对广大开发人员的重要性亦由此可见一斑。本书以基本概念、原理与方法为主线,辅以丰富的实战案例和生活化实例,并从Java虚拟机、操作系统和硬件多个层次与角度出发,循序渐进、系统地介绍Java平台下的多线程编程核心技术及相关工具。

本书适合有一定Java语言基础的读者作为入门多线程编程之用,也适合有一定多线程编程经验的读者作为重新梳理知识结构以提升认知层次和参考之用。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

Java多线程编程实战指南.核心篇 / 黄文海著. —北京:电子工业出版社, 2017.4

(Java多线程编程实战系列)

ISBN 978-7-121-31065-2

I. ①J… II. ①黄… III. ①JAVA语言—程序设计—指南 IV. ①TP312.8-62

中国版本图书馆CIP数据核字(2017)第047559号

策划编辑:付 睿

责任编辑:李云静

印 刷:三河市良远印务有限公司

装 订:三河市良远印务有限公司

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编:100036

开 本:787×980 1/16 印张:30 字数:618千字

版 次:2017年4月第1版

印 次:2017年4月第1次印刷

印 数:3000册 定价:89.00元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010)88254888,88258888。

质量投诉请发邮件至 zltts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式:010-51260888-819, faq@phei.com.cn。

随着现代处理器的生产工艺从提升处理器主频频率转向多核化,即在一块芯片上集成多个处理器内核(Core),多核处理器(Multicore Processor)离我们越来越近了——如今就连智能手机这样的消费类设备都已配备了4核乃至8核的处理器,更何况商用系统!在此背景下,以往靠单个处理器自身处理能力的提升所带来的软件计算性能提升的那种“免费午餐”已不复存在,这使得多线程编程在充分利用计算资源、提高软件服务质量方面扮演了越来越重要的角色。故而,掌握多线程编程技能对广大开发人员的重要性亦由此可见一斑。

本书内容及特色

本书以基本概念、原理与方法为主线,辅以丰富的实战案例和生活化实例,并从Java虚拟机、操作系统和硬件多个层次与角度出发,循序渐进、系统地介绍Java平台下的多线程编程核心技术及相关工具。

脉络清晰、循序渐进和系统性介绍。全书围绕多线程编程所要解决的问题(所要实现的目标)及其面临的各种挑战,由此展开介绍多线程编程中的相关概念、原理与技术。本书以先介绍相关问题及背景再给出相应的解决方案的方式来讲解新的概念、知识。并且,本书对概念、原理与技术的讲解会适当地深入到Java虚拟机、操作系统和硬件这三个层次与角度,而不仅仅停留在Java语言层面。全书章节是按照知识间的内在联系并依照认知程度上的由浅至深的顺序组织的。

以基本概念、原理与方法为主线。本书既注重实战又注重理论对实践的指导作用。本书以多线程编程的基本概念、原理与方法为主线,将Java平台中与多线程编程相关的关键字、Java标准库类(API)等知识串在其上进行讲解,并在讲解过程中适当穿插相关工具的介绍。本书在介绍相关Java标准库类时,不仅仅介绍其API,还适当介绍其内部实现

原理与实战注意事项。

辅以丰富的实战案例和生活化实例。本书配有丰富的实战案例，这些案例的配套源码都是可以实际运行的，以便读者进行实验。本书在介绍一些概念和原理的时候往往辅以一些生活化实例以增加读者的感性认识，降低理解难度。

答疑解惑。本书讲解过程中会穿插一些“扩展阅读”的内容，这部分内容以问答的形式来对多线程编程的初学者在学习和工作过程中经常遇到的一些疑惑和问题进行解答。

本书一共分为 12 章，各章的主要内容如下。

第 1 章主要介绍线程及多线程编程这两个基本概念，以及 Java 平台的线程 API，并通过一个实战案例使读者对多线程编程有个初步和感性的认识。

第 2 章主要介绍多线程编程所要实现的目标及其面临的挑战。明确多线程编程的目标有助于我们在实践中做到有的放矢，掌握多线程编程所面临的挑战使得我们在学习本书后续内容时能够做到知其然而且知其所以然。

第 3 章主要介绍 Java 平台所提供的能够用于应对多线程编程所面临的部分挑战的一些关键字和标准库类（API），以及这些关键字和 API 的性能开销、适用场景及注意事项。

第 4 章通过实战案例介绍具体如何玩转线程以实现多线程编程的目标，并通过这些实战案例展开介绍多线程编程实践中的一些实际问题及应对措施。

第 5 章主要介绍线程与线程之间通过哪些常见的协作形式来实现多线程编程的目标以及 Java 所支持的相应标准库类。

第 6 章主要从软件设计的角度出发介绍应对多线程编程所面临的一些挑战的常见技术。

第 7 章主要介绍多线程程序可能产生的一些常见的具有隐蔽性的故障以及相应的规避措施。

第 8 章主要介绍在多线程编程中如何更加有效和有效率地使用线程。

第 9 章主要从计算模型的角度出发介绍多线程编程中如何利用线程来提高软件的吞吐率和响应性。

第 10 章主要介绍 Java 平台中多线程程序的调试技巧与测试手段。

第 11 章主要介绍多线程编程的硬件基础以及 Java 平台为屏蔽不同硬件平台的差异而进行的抽象（Java 内存模型）。

第 12 章结合实战案例介绍与 Java 平台中的多线程程序紧密相关的常用性能优化方法与技术。

目标读者

本书适合有一定 Java 语言基础的读者作为入门多线程编程之用，也适合有一定多线程编程经验的读者作为重新梳理知识结构以提升认知层次和参考之用。

本书约定

1. 对于标题中带星号（*）的小节，读者可以选择先浏览一下章节标题就跳过它，之后在阅读后续内容遇到问题时再回头来阅读相应的小节。

2. 斜体格式的方法名表示相应方法为静态方法，例如 `System.currentTimeMillis()`（其中，方法名 `currentTimeMillis` 的字体格式为斜体）。非斜体格式的方法名表示相应方法为相应类的实例方法，例如 `StringBuilder.append(String)` 或者 `StringBuilder.append(String str)` 均表示类 `StringBuilder` 的实例方法 `append`。本书有时候也会省略方法中的形式参数列表。

3. 本书用“/”作为分隔符来表示同一个类的多个方法。例如，`StringBuilder.append(String)/toString()` 表示 `StringBuilder` 的 `append` 方法和 `toString` 方法。

4. 本书所指的 Java 虚拟机（JVM）如无特别说明均特指 Oracle 公司的 HotSpot Java 虚拟机。

5. 就 HotSpot Java 虚拟机而言，JIT（Just In Time）编译器是该 Java 虚拟机的一部分，因此本书有时候并不严格区分 Java 虚拟机和 JIT 编译器。

6. 本书涉及的命令如无特别说明均指 Linux 平台下的命令。

如何阅读本书

本书讲解过程中会涉及一些与多线程编程紧密相关的硬件知识，如果读者对这些知识不太熟悉，可以在阅读过程中参考或者直接先行阅读本书第 11 章前 4 节的内容。

读者也可以先阅读完本书前 4 章的内容，接着就开始集中实践。然后边实践边阅读本书的后续章节，或者在实践过程中遇到问题时再参考本书后续章节的内容。当然，这种阅读方法主要是便于读者尽快上手，并不是说本书后续章节的内容无足轻重。

学习一门新的技术、语言的一个行之有效的方法就是边学习边思考、带着问题在学习过程中寻找答案。因此，本书讲解过程中会穿插一些“扩展阅读”的内容，这些内容多涉及新手在学习多线程编程过程中经常会遇到的一些疑惑和问题。尽管如此，这并不能代替读者自己主动思考并从书中或者其他途径寻找答案。

本书介绍了 Java 标准库中与多线程编程紧密相关的一些类，但是这些内容并不能取代读者亲自阅读 Java 的 API 文档（<http://docs.oracle.com/javase/8/docs/api/>）。

配套源码下载

本书配套源码可以从下列网址下载：

<https://github.com/Viscent/javamtia>

或者，

<http://www.broadview.com.cn/31065>

与作者联系

读者在阅读本书过程中遇到问题或者有任何建议时，可以通过微信公众号 VChannel 与作者联系。

读者服务

轻松注册成为博文视点社区用户 (www.broadview.com.cn), 您即可享受以下服务。

- 下载资源: 本书所提供的示例代码及资源文件均可在【下载资源】处下载。
- 提交勘误: 您对书中内容的修改意见可在【提交勘误】处提交, 若被采纳, 将获赠博文视点社区积分 (在您购买电子书时, 积分可用来抵扣相应金额)。
- 与作者交流: 在页面下方【读者评论】处留下您的疑问或观点, 与作者和其他读者一同学习交流。

页面入口: <http://www.broadview.com.cn/31065>

二维码:



博文视点诚邀精锐作者加盟

十载耕耘奠定专业地位

以书为证彰显卓越品质

《C++Primer (中文版) (第5版)》、《淘宝技术这十年》、《代码大全》、《Windows内核情景分析》、《加密与解密》、《编程之美》、《VC++深入详解》、《SEO实战密码》、《PPT演义》……

“圣经”级图书光耀夺目,被无数读者朋友奉为案头手册传世经典。

潘爱民、毛德操、张亚勤、张宏江、咎辉Zac、李刚、曹江华……

“明星”级作者济济一堂,他们的名字熠熠生辉,与IT业的蓬勃发展紧密相连。

十年的开拓、探索和励精图治,成就博古通今、文圆质方、视角独特、点石成金之计算机图书的风向标杆:博文视点。

“凤翱翔于千仞兮,非梧不栖”,博文视点欢迎更多才华横溢、锐意创新的作者朋友加盟,与大师并列于IT专业出版之巔。

英雄帖

江湖风云起,代有才人出。
IT界群雄并起,逐鹿中原。
博文视点诚邀天下技术英豪加入,
指点江山,激扬文字
传播信息技术,分享IT心得

• 专业的作者服务 •

博文视点自成立以来一直专注于IT专业技术图书的出版,拥有丰富的与技术图书作者合作的经验,并参照IT技术图书的特点,打造了一支高效运转、富有服务意识的编辑出版团队。我们始终坚持:

善待作者——我们会把出版流程整理得清晰简明,为作者提供优厚的稿酬服务,解除作者的顾虑,安心写作,展现出最好的作品。

尊重作者——我们尊重每一位作者的技术实力和生活习惯,并会参照作者实际的工作、生活节奏,量身制定写作计划,确保合作顺利进行。

提升作者——我们打造精品图书,更要打造知名作者。博文视点致力于通过图书提升作者的个人品牌和技术影响力,为作者的事业开拓带来更多的机会。



联系我们

博文视点官网: <http://www.broadview.com.cn>

CSDN官方博客: <http://blog.csdn.net/broadview2006/>

投稿电话: 010-51260888 88254368

投稿邮箱: jsj@phei.com.cn



@博文视点Broadview



微信公众平台号 博文视点Broadview



第一部分 多线程编程基础

第 1 章 走近 Java 世界中的线程.....	2
1.1 进程、线程与任务	2
1.2 多线程编程简介	4
1.2.1 什么是多线程编程	4
1.2.2 为什么使用多线程	4
1.3 Java 线程 API 简介	5
1.3.1 线程的创建、启动与运行	5
1.3.2 Runnable 接口	9
1.3.3 线程属性	12
1.3.4 Thread 类的常用方法	14
1.3.5 Thread 类的一些废弃方法	16
1.4 无处不在的线程	17
1.5 线程的层次关系	19
1.6 线程的生命周期状态	21
1.7 线程的监视	22
1.8 多线程编程简单运用实例	26
*1.9 多线程编程的优势和风险	27
1.10 本章小结	29
第 2 章 多线程编程的目标与挑战	31
2.1 串行、并发与并行	31
2.2 竞态	33

2.2.1 二维表分析法：解释竞态的结果.....	37
2.2.2 竞态的模式与竞态产生的条件.....	39
2.3 线程安全性.....	42
2.4 原子性	43
2.5 可见性	49
2.6 有序性	56
2.6.1 重排序的概念	56
2.6.2 指令重排序	57
2.6.3 存储子系统重排序.....	63
2.6.4 貌似串行语义	66
2.6.5 保证内存访问的顺序性.....	68
2.7 上下文切换.....	69
2.7.1 上下文切换及其产生原因.....	69
2.7.2 上下文切换的分类及具体诱因	70
2.7.3 上下文切换的开销和测量.....	71
2.8 线程的活性故障.....	73
2.9 资源争用与调度.....	74
2.10 本章小结.....	77
第3章 Java 线程同步机制.....	80
3.1 线程同步机制简介.....	80
3.2 锁概述	81
3.2.1 锁的作用	82
3.2.2 与锁相关的几个概念.....	84
3.2.3 锁的开销及其可能导致的问题.....	86
3.3 内部锁：synchronized 关键字	86
3.4 显式锁：Lock 接口	89
3.4.1 显式锁的调度	91
3.4.2 显式锁与内部锁的比较.....	92
3.4.3 内部锁还是显式锁：锁的选用	95
*3.4.4 改进型锁：读写锁	95
3.5 锁的适用场景.....	99
3.6 线程同步机制的底层助手：内存屏障.....	99

*3.7 锁与重排序	102
3.8 轻量级同步机制: <code>volatile</code> 关键字	105
3.8.1 <code>volatile</code> 的作用	105
3.8.2 <code>volatile</code> 变量的开销	111
3.8.3 <code>volatile</code> 的典型应用场景与实战案例	111
3.9 实践: 正确实现看似简单的单例模式	120
3.10 CAS 与原子变量	126
3.10.1 CAS	127
3.10.2 原子操作工具: 原子变量类	129
3.11 对象的发布与逸出	135
3.11.1 对象的初始化安全: 重访 <code>final</code> 与 <code>static</code>	137
3.11.2 安全发布与逸出	142
3.12 本章小结	143
第 4 章 牛刀小试: 玩转线程	148
4.1 挖掘可并发点	148
4.2 新战场上的老武器: 分而治之	148
4.3 基于数据的分割实现并发化	149
4.4 基于任务的分割实现并发化	158
4.4.1 按任务的资源消耗属性分割	159
4.4.2 实战案例的启发	169
4.4.3 按处理步骤分割	171
4.5 合理设置线程数	172
4.5.1 Amdahl's 定律	172
4.5.2 线程数设置的原则	173
4.6 本章小结	177
第 5 章 线程间协作	179
5.1 等待与通知: <code>wait/notify</code>	179
5.1.1 <code>wait/notify</code> 的作用与用法	180
5.1.2 <code>wait/notify</code> 的开销及问题	188
5.1.3 <code>Object.notify()/notifyAll()</code> 的选用	191
*5.1.4 <code>wait/notify</code> 与 <code>Thread.join()</code>	191

5.2	Java 条件变量	192
5.3	倒计时协调器：CountDownLatch	198
5.4	栅栏（CyclicBarrier）	203
5.5	生产者—消费者模式	210
5.5.1	阻塞队列	213
5.5.2	限购：流量控制与信号量（Semaphore）	216
*5.5.3	管道：线程间的直接输出与输入	218
5.5.4	一手交钱，一手交货：双缓冲与 Exchanger	221
5.5.5	一个还是一批：产品的粒度	223
5.5.6	再探线程与任务之间的关系	224
5.6	对不起，打扰一下：线程中断机制	225
5.7	线程停止：看似简单，实则不然	228
5.7.1	生产者—消费者模式中的线程停止	233
5.7.2	实践：Web 应用中的线程停止	233
5.8	本章小结	236
第 6 章	保障线程安全的设计技术	240
*6.1	Java 运行时存储空间	240
6.2	大公无私：无状态对象	243
6.3	以“不变”应万变：不可变对象	248
6.4	我有我地盘：线程特有对象	254
6.4.1	线程特有对象可能导致的问题及其规避	258
6.4.2	线程特有对象的典型应用场景	264
6.5	装饰器模式	265
6.6	并发集合	267
6.7	本章小结	270
第 7 章	线程的活性故障	273
7.1	鹬蚌相争：死锁	273
7.1.1	死锁的检测	274
7.1.2	死锁产生的条件与规避	283
7.1.3	死锁的恢复	296

7.2 沉睡不醒的睡美人：锁死	301
7.2.1 信号丢失锁死	301
7.2.2 嵌套监视器锁死	301
7.3 巧妇难为无米之炊：线程饥饿	307
7.4 屡战屡败，屡败屡战：活锁	307
7.5 本章小结	308
第 8 章 线程管理	310
8.1 线程组	310
8.2 可靠性：线程的未捕获异常与监控	311
8.3 有组织有纪律：线程工厂	316
8.4 线程的暂挂与恢复	318
8.5 线程的高效利用：线程池	320
8.5.1 任务的处理结果、异常处理与取消	326
8.5.2 线程池监控	329
8.5.3 线程池死锁	330
8.5.4 工作者线程的异常终止	330
8.6 本章小结	331
第 9 章 Java 异步编程	333
9.1 同步计算与异步计算	333
9.2 Java Executor 框架	336
9.2.1 实用工具类 Executors	337
9.2.2 异步任务的批量执行：CompletionService	339
9.3 异步计算助手：FutureTask	344
9.3.1 实践：实现 XML 文档的异步解析	345
9.3.2 可重复执行的异步任务	349
9.4 计划任务	352
9.5 本章小结	358

第 10 章	Java 多线程程序的调试与测试	360
10.1	多线程程序的调试技巧	360
10.1.1	使用监视点	360
10.1.2	设置暂挂策略	361
10.2	多线程程序的测试	363
10.2.1	可测试性	364
10.2.2	静态检查工具：FindBugs	369
10.2.3	多线程程序的代码复审	370
10.2.4	多线程程序的单元测试：JCStress	372
10.3	本章小结	375

第二部分 多线程编程进阶

第 11 章	多线程编程的硬件基础与 Java 内存模型	378
11.1	填补处理器与内存之间的鸿沟：高速缓存	378
11.2	数据世界的交通规则：缓存一致性协议	382
11.3	硬件缓冲区：写缓冲器与无效化队列	386
11.3.1	存储转发	388
11.3.2	再探内存重排序	388
11.3.3	再探可见性	391
11.4	基本内存屏障	392
11.5	Java 同步机制与内存屏障	395
11.5.1	volatile 关键字的实现	395
11.5.2	synchronized 关键字的实现	397
11.5.3	Java 虚拟机对内存屏障使用的优化	398
11.5.4	final 关键字的实现	398
11.6	Java 内存模型	399
11.6.1	什么是 Java 内存模型	400
11.6.2	happen(s)-before 关系	401
11.6.3	再探对象的安全发布	407
11.6.4	JSR 133	411
11.7	共享变量与性能	411
11.8	本章小结	411

第 12 章 Java 多线程程序的性能调校	415
12.1 Java 虚拟机对内部锁的优化	415
12.1.1 锁消除	415
12.1.2 锁粗化	417
12.1.3 偏向锁	419
12.1.4 适应性锁	420
12.2 优化对锁的使用	421
12.2.1 锁的开销与锁争用监视	421
12.2.2 使用可参数化锁	424
12.2.3 减小临界区的长度	428
12.2.4 减小锁的粒度	432
12.2.5 考虑锁的替代品	438
12.3 减少系统内耗：上下文切换	438
12.4 多线程编程的“三十六计”：多线程设计模式	440
12.5 性能的隐形杀手：伪共享	441
12.5.1 Java 对象内存布局	442
12.5.2 伪共享的侦测与消除	445
12.6 本章小结	454
Web 参考资料	457
参考文献	463

第一部分

多线程编程基础

- 第 1 章 走近 Java 世界中的线程
 - 第 2 章 多线程编程的目标与挑战
 - 第 3 章 Java 线程同步机制
 - 第 4 章 牛刀小试：玩转线程
 - 第 5 章 线程间协作
 - 第 6 章 保障线程安全的设计技术
 - 第 7 章 线程的活性故障
 - 第 8 章 线程管理
 - 第 9 章 Java 异步编程
 - 第 10 章 Java 多线程程序的调试与测试
-