

华章程序员书库

[PACKT]
PUBLISHING

HZ BOOKS
华章IT

C#多线程编程实战

Multithreading in C# 5.0 Cookbook



[美] Eugene Agafonov 著

黄博文 黄辉兰 译



机械工业出版社
China Machine Press

华章程序员书库



Multithreading in C# 5.0 Cookbook

C#多线程编程实战

[美] Eugene Agafonov 著

黄博文 黄辉兰 译



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

C# 多线程编程实战 / (美) 阿格佛温 (Agafonov, E.) 著; 黄博文等译. —北京: 机械工业出版社, 2015.3

(华章程序员书库)

书名原文: Multithreading in C# 5.0 Cookbook

ISBN 978-7-111-49348-8

I. C… II. ①阿… ②黄… ③黄… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2015) 第 029657 号

本书版权登记号: 图字: 01-2014-2038

Multithreading in C# 5.0 Cookbook (ISBN: 978-1-84969-764-4) .

Copyright © 2013 Packt Publishing. First published in the English language under the title "Multithreading in C# 5.0 Cookbook" .

All rights reserved.

Chinese simplified language edition published by China Machine Press.

Copyright © 2015 by China Machine Press.

本书中文简体字版由 Packt Publishing 授权机械工业出版社独家出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

C# 多线程编程实战

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 陈佳媛

责任校对: 董纪丽

印 刷: 三河市宏图印务有限公司

版 次: 2015 年 3 月第 1 版第 1 次印刷

开 本: 186mm × 240mm 1/16

印 张: 13.5

书 号: ISBN 978-7-111-49348-8

定 价: 49.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzjsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

不久前，典型的个人计算机的 CPU 还只有一个计算核心，并且功耗足以煎熟鸡蛋。2005 年，英特尔推出了其首款多核心 CPU，从此计算机开始向不同的方向发展。低耗电量及多个计算核心变得比提高行计算（row computing）核心性能更重要。这也导致了编程范式的改变。现在我们需要学习如何有效地使用所有 CPU 核心来最优化性能，并同时通过在特定时间只运行需要的程序来节省电池电量。除此之外，我们在编写服务器端应用程序时需要有效地利用多个 CPU 核心，甚至多台计算机来支持尽可能多的用户。

为了创建这样的应用程序，你需要学习如何在程序中有效地使用多个 CPU 核心。如果你使用的是 Microsoft .NET 开发平台以及 C# 编程语言，那么本书将是一个编写高性能、高响应性的应用程序的完美起点。

本书的目的是给你提供 C# 中多线程以及并行编程的详尽指导。我们将从基本概念开始，每章主题比前一章都有所拔高，最后展示现实世界中的并行编程模式以及 Windows 商店应用示例。

本书内容

第 1 章介绍了 C# 中基本的线程操作。本章解释了什么是线程，使用线程的优缺点，以及与线程相关的其他重要方面。

第 2 章描述了线程交互细节。你将了解为何我们需要协调线程，以及协调组织线程的不同方式。

第 3 章解释了线程池概念。本章展示了如何使用线程池，如何执行异步操作，以及使用线程池的好的和不好的实践。

第 4 章深入讲解了任务并行库（Task Parallel Library, TPL）框架。本章讲述了 TPL 的所有重要方面，包括任务组合、异常管理及取消操作等。

第 5 章深入解释了 C# 5.0 中的新功能——异步方法。你将了解 `async` 和 `await` 关键字，如何在不同的场景中使用它们，以及 `await` 底层工作机制。

第 6 章描述了 .NET 框架中并行算法的标准数据结构，并为每种数据结构展示了示例编程场景。

第 7 章深入讲解了并行 LINQ 基础设施。本章讲述了任务和数据并行度，并行化 LINQ 查询，调整并行选项，分割查询，集合并行查询结果等内容。

第 8 章解释了如何以及何时使用 Reactive Extensions 框架。你将学习如何组合事件，如何对事件序列执行 LINQ 查询等。

第 9 章深入讲解了异步 I/O 进程，包括文件、网络及数据库等场景。

第 10 章列出了针对常见的并行编程问题的解决方案。

第 11 章覆盖了基于 Windows 8 的编写异步应用程序的场景。你将学习如何使用 Windows 8 异步 API，以及如何在 Windows 商店应用中完成后台工作。

准备事项

我们需要 Windows 桌面版本的 Microsoft Visual Studio Express 2012 来完成大多数章节的学习。第 11 章中的小节还需要 Windows 8 操作系统和 Visual Studio Express 2012 编译 Windows 商店应用。

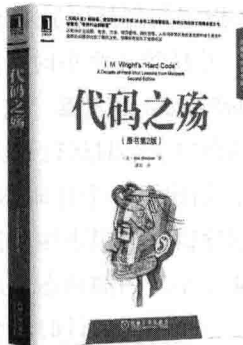
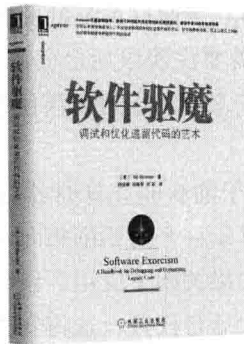
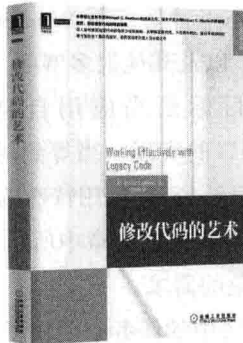
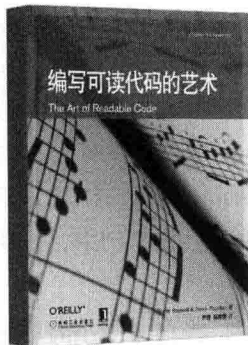
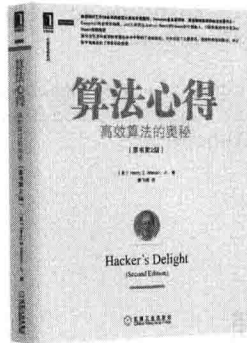
读者对象

本书的读者对象为没有或只有少量多线程及异步和并发编程背景的 C# 开发人员。本书涵盖了 C# 和 .NET 生态系统中从基础概念到复杂编程模式及算法的很多与多线程相关的主题。

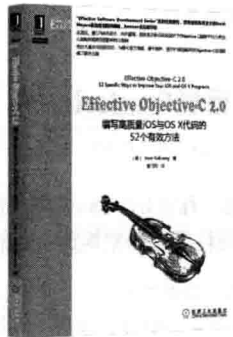
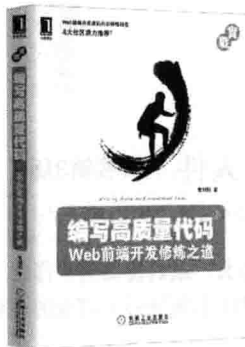
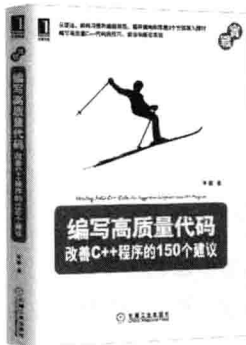
下载示例代码

本书的示例代码可登录华章网站 (<http://www.hzbook.com>) 中的本书页面下载。

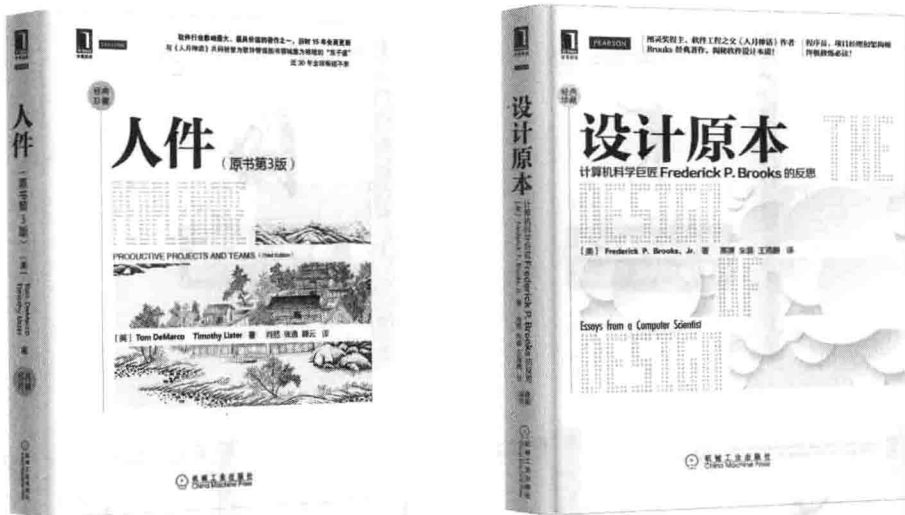
推荐阅读



推荐阅读



推荐阅读



人件 (原书第3版)

作者: (美) Tom DeMarco 等 ISBN: 978-7-111-47436-4 定价: 69.00元

公认对软件行业影响最大、最具价值的著作之一，历时15年全面更新
与《人月神话》共同被誉为软件图书领域最为璀璨的“双子星”，近30年全球畅销不衰

在软件管理领域，很少有著作能够与本书媲美。全书从管理人力资源、创建健康的办公环境、雇用并留用正确的人、高效团队形成、改造企业文化和快乐工作等多个角度阐释了如何思考和管理软件开发的重大问题——人（而不是技术），以得到高效的项目和团队。

设计原本——计算机科学巨匠Frederick P. Brooks的反思 (经典珍藏)

作者: (美) Frederick P. Brooks, Jr. ISBN: 978-7-111-41626-5 定价: 79.00元

图灵奖得主、《人月神话》作者Brooks封笔之作，揭秘软件设计神话！
程序员、项目经理和架构师必读的一本书！

《设计原本》开启了软件工程全新的“后理性时代”，完成了从破到立的圆满循环，具有划时代的重大里程碑意义，是每位从事软件行业的程序员、项目经理和架构师都应该反复研读的经典著作。全书以设计理念为核心，从对设计模型的探讨入手，讨论了有关设计的若干重大问题：设计过程的建立、设计协作的规划、设计范本的固化、设计演化的管控，以及设计师的发现和培养。

前 言

第 1 章 线程基础	1
1.1 简介.....	1
1.2 使用 C# 创建线程.....	2
1.2.1 准备工作.....	2
1.2.2 实现方式.....	2
1.2.3 工作原理.....	4
1.3 暂停线程.....	4
1.3.1 准备工作.....	5
1.3.2 实现方式.....	5
1.3.3 工作原理.....	5
1.4 线程等待.....	6
1.4.1 准备工作.....	6
1.4.2 实现方式.....	6
1.4.3 工作原理.....	7
1.5 终止线程.....	7
1.5.1 准备工作.....	7
1.5.2 实现方式.....	7
1.5.3 工作原理.....	8
1.6 检测线程状态.....	8
1.6.1 准备工作.....	8

1.6.2	实现方式	8
1.6.3	工作原理	9
1.7	线程优先级	10
1.7.1	准备工作	10
1.7.2	实现方式	10
1.7.3	工作原理	11
1.8	前台线程和后台线程	12
1.8.1	准备工作	12
1.8.2	实现方式	12
1.8.3	工作原理	13
1.9	向线程传递参数	13
1.9.1	准备工作	13
1.9.2	实现方式	13
1.9.3	工作原理	15
1.9.4	更多信息	15
1.10	使用 C# 中的 lock 关键字	16
1.10.1	准备工作	16
1.10.2	实现方式	16
1.10.3	工作原理	18
1.11	使用 Monitor 类锁定资源	18
1.11.1	准备工作	18
1.11.2	实现方式	18
1.11.3	工作原理	20
1.12	处理异常	20
1.12.1	准备工作	20
1.12.2	实现方式	21
1.12.3	工作原理	22
第 2 章	线程同步	23
2.1	简介	23
2.2	执行基本的原子操作	24

2.2.1	准备工作	24
2.2.2	实现方式	24
2.2.3	工作原理	26
2.3	使用 Mutex 类	27
2.3.1	准备工作	27
2.3.2	实现方式	27
2.3.3	工作原理	27
2.4	使用 SemaphoreSlim 类	28
2.4.1	准备工作	28
2.4.2	实现方式	28
2.4.3	工作原理	29
2.4.4	更多信息	29
2.5	使用 AutoResetEvent 类	29
2.5.1	准备工作	30
2.5.2	实现方式	30
2.5.3	工作原理	31
2.6	使用 ManualResetEventSlim 类	31
2.6.1	准备工作	31
2.6.2	实现方式	31
2.6.3	工作原理	32
2.6.4	更多信息	33
2.7	使用 CountDownEvent 类	33
2.7.1	准备工作	33
2.7.2	实现方式	33
2.7.3	工作原理	34
2.8	使用 Barrier 类	34
2.8.1	准备工作	34
2.8.2	实现方式	34
2.8.3	工作原理	35
2.9	使用 ReaderWriterLockSlim 类	35
2.9.1	准备工作	36

2.9.2 实现方式	36
2.9.3 工作原理	37
2.10 使用 SpinWait 类	38
2.10.1 准备工作	38
2.10.2 实现方式	38
2.10.3 工作原理	39
第 3 章 使用线程池	40
3.1 简介	40
3.2 在线程池中调用委托	41
3.2.1 准备工作	42
3.2.2 实现方式	42
3.2.3 工作原理	43
3.3 向线程池中放入异步操作	44
3.3.1 准备工作	44
3.3.2 实现方式	44
3.3.3 工作原理	45
3.4 线程池与并行度	45
3.4.1 准备工作	46
3.4.2 实现方式	46
3.4.3 工作原理	47
3.5 实现一个取消选项	47
3.5.1 准备工作	47
3.5.2 实现方式	48
3.5.3 工作原理	49
3.6 在线程池中使用等待事件处理器及超时	50
3.6.1 准备工作	50
3.6.2 实现方式	50
3.6.3 工作原理	51
3.6.4 更多信息	51
3.7 使用计时器	52

3.7.1	准备工作	52
3.7.2	实现方式	52
3.7.3	工作原理	53
3.8	使用 BackgroundWorker 组件	53
3.8.1	准备工作	53
3.8.2	实现方式	53
3.8.3	工作原理	55
第 4 章	使用任务并行库	56
4.1	简介	56
4.2	创建任务	57
4.2.1	准备工作	58
4.2.2	实现方式	58
4.2.3	工作原理	59
4.3	使用任务执行基本的操作	59
4.3.1	准备工作	60
4.3.2	实现方式	60
4.3.3	工作原理	61
4.4	组合任务	61
4.4.1	准备工作	61
4.4.2	实现方式	61
4.4.3	工作原理	63
4.5	将 APM 模式转换为任务	63
4.5.1	准备工作	64
4.5.2	实现方式	64
4.5.3	工作原理	66
4.6	将 EAP 模式转换为任务	66
4.6.1	准备工作	66
4.6.2	实现方式	67
4.6.3	工作原理	68
4.7	实现取消选项	68

4.7.1	准备工作	68
4.7.2	实现方式	68
4.7.3	工作原理	69
4.8	处理任务中的异常	69
4.8.1	准备工作	70
4.8.2	实现方式	70
4.8.3	工作原理	71
4.8.4	更多信息	71
4.9	并行运行任务	72
4.9.1	准备工作	72
4.9.2	实现方式	72
4.9.3	工作原理	73
4.10	使用 TaskScheduler 配置任务的执行	73
4.10.1	准备工作	73
4.10.2	实现方式	74
4.10.3	工作原理	76
第 5 章	使用 C# 5.0	78
5.1	简介	78
5.2	使用 await 操作符获取异步任务结果	80
5.2.1	准备工作	80
5.2.2	实现方式	80
5.2.3	工作原理	81
5.3	在 lambda 表达式中使用 await 操作符	82
5.3.1	准备工作	82
5.3.2	实现方式	82
5.3.3	工作原理	83
5.4	对连续的异步任务使用 await 操作符	83
5.4.1	准备工作	84
5.4.2	实现方式	84
5.4.3	工作原理	85

5.5	对并行执行的异步任务使用 <code>await</code> 操作符	86
5.5.1	准备工作	86
5.5.2	实现方式	86
5.5.3	工作原理	87
5.6	处理异步操作中的异常	88
5.6.1	准备工作	88
5.6.2	实现方式	88
5.6.3	工作原理	89
5.7	避免使用捕获的同步上下文	90
5.7.1	准备工作	90
5.7.2	实现方式	90
5.7.3	工作原理	92
5.8	使用 <code>async void</code> 方法	93
5.8.1	准备工作	93
5.8.2	实现方式	93
5.8.3	工作原理	95
5.9	设计一个自定义的 <code>awaitable</code> 类型	96
5.9.1	准备工作	96
5.9.2	实现方式	96
5.9.3	工作原理	97
5.10	对动态类型使用 <code>await</code>	98
5.10.1	准备工作	99
5.10.2	实现方式	99
5.10.3	工作原理	101
第 6 章	使用并发集合	102
6.1	简介	102
6.2	使用 <code>ConcurrentDictionary</code>	103
6.2.1	准备工作	104
6.2.2	实现方式	104
6.2.3	工作原理	105

6.3	使用 ConcurrentQueue 实现异步处理	105
6.3.1	准备工作	106
6.3.2	实现方式	106
6.3.3	工作原理	107
6.4	改变 ConcurrentStack 异步处理顺序	108
6.4.1	准备工作	108
6.4.2	实现方式	108
6.4.3	工作原理	109
6.5	使用 ConcurrentBag 创建一个可扩展的爬虫	110
6.5.1	准备工作	110
6.5.2	实现方式	110
6.5.3	工作原理	113
6.6	使用 BlockingCollection 进行异步处理	113
6.6.1	准备工作	113
6.6.2	实现方式	113
6.6.3	工作原理	115
第 7 章	使用 PLINQ	116
7.1	简介	116
7.2	使用 Parallel 类	117
7.2.1	准备工作	118
7.2.2	实现方式	118
7.2.3	工作原理	119
7.3	并行化 LINQ 查询	119
7.3.1	准备工作	119
7.3.2	实现方式	120
7.3.3	工作原理	122
7.4	调整 PLINQ 查询的参数	122
7.4.1	准备工作	122
7.4.2	实现方式	123
7.4.3	工作原理	124

7.5 处理 PLINQ 查询中的异常	125
7.5.1 准备工作	125
7.5.2 实现方式	125
7.5.3 工作原理	126
7.6 管理 PLINQ 查询中的数据分区	126
7.6.1 准备工作	127
7.6.2 实现方式	127
7.6.3 工作原理	128
7.7 为 PLINQ 查询创建一个自定义的聚合器	129
7.7.1 准备工作	129
7.7.2 实现方式	129
7.7.3 工作原理	131
第 8 章 使用 Reactive Extensions	132
8.1 简介	132
8.2 将普通集合转换为异步的可观察集合	133
8.2.1 准备工作	133
8.2.2 实现方式	133
8.2.3 工作原理	135
8.3 编写自定义的可观察对象	136
8.3.1 准备工作	136
8.3.2 实现方式	136
8.3.3 工作原理	138
8.4 使用 Subject	138
8.4.1 准备工作	138
8.4.2 实现方式	138
8.4.3 工作原理	140
8.5 创建可观察的对象	141
8.5.1 准备工作	141
8.5.2 实现方式	141
8.5.3 工作原理	143