Maya Posch

Mastering C++ Multithreading

A comprehensive guide to developing effective multithreading applications in C++

Packt>

Mastering C++ Multithreading

Multithreaded applications execute multiple threads in a single processor environment, allowing developers achieve concurrency. This book will teach you the finer points of multithreading and concurrency concepts and how to apply them efficiently in C++.

Divided into three modules, we start with a brief introduction to the fundamentals of multithreading and concurrency concepts. We then take an in-depth look at how these concepts work at the hardware level as well as how both operating systems and frameworks use these low-level functions.

In the next module, you will learn about the native multithreading and concurrency support available in C++ since the 2011 revision, synchronization and communication between threads, debugging concurrent C++ applications, and the best programming practices in C++.

In the final module, you will learn about atomic operations before moving on to apply concurrency to distributed and GPGPU-based processing. The comprehensive coverage of essential multithreading concepts means you will be able to efficiently apply multithreading concepts while coding in C++.

Things you will learn:

- Deep dive into the details of how various operating systems currently implement multithreading
- Choose the best multithreading APIs when designing a new application
- Explore the use of mutexes, spin-locks, and other synchronization concepts, and see how to safely pass data between threads
- Understand the level of API support provided by various C++ toolchains
- Resolve common issues in multithreaded code and recognize common pitfalls
- Discover the nature of atomic operations and understand how they can be useful in optimizing code
- Implement a multithreaded application in a distributed computing environment
- Design a C++-based GPGPU application that employs multithreading

Packt>
www.packtpub.com

$ **44.99** US
£ **37.99** UK

Prices do not include local sales Tax or VAT where applicable

ISBN 978-1-78712-170-6
54499
9 781787 121706

Mastering C++ Multithreading

Maya Posch

Mastering C++ Multithreading

A comprehensive guide to developing effective multithreading applications in C++

Maya Posch

Packt

BIRMINGHAM - MUMBAI

Mastering C++ Multithreading

Copyright © 2017 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: July 2017

Production reference: 1270717

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham
B3 2PB, UK.

ISBN 978-1-78712-170-6

www.packtpub.com

试读结束：需要全本请在线购买： www.ertongbook.com

Credits

Author
Maya Posch

Reviewer
Louis E. Mauget

Commissioning Editor
Aaron Lazar

Acquisition Editor
Chaitanya Nair

Content Development Editor
Rohit Kumar Singh

Technical Editors
Ketan Kamble

Copy Editor
Sonia Mathur

Project Coordinator
Vaidehi Sawant

Proofreader
Safis Editing

Indexer
Francy Puthiry

Graphics
Abhinash Sahu

Production Coordinator
Nilesh Mohite

About the Author

Maya Posch is a software engineer by trade and a self-professed electronics, robotics, and AI nut, running her own software development company, Nyanko, with her good friend, Trevor Purdy, where she works on various game development projects and some non-game projects. Apart from this, she does various freelance jobs for companies around the globe. You can visit her LinkedIn profile for more work-related details.

Aside from writing software, she likes to play with equations and write novels, such as her awesome reimagining of the story of the Nintendo classic, Legend of Zelda: Ocarina of Time, and the survival-horror novel she recently started, Viral Desire. You can check out her Scribd profile for a full listing of her writings.

Maya is also interested in biochemistry, robotics, and reverse-engineering of the human body. To know more about her, visit her blog, Artificial Human. If there's anything she doesn't lack, it has to be sheer ambition, it seems.

About the Reviewer

Louis E. Mauget learned to program a long time ago at the Michigan State University as a physics major learning to use software in designing a cyclotron. Later, he worked for 34 years at IBM. He went on to work for several consulting firms, including a long-term engagement with the railroad industry. He is currently consulting for Keyhole Software at Leawood, Kansas.

Lou has coded in C++, Java, JavaScript, Python, and newer languages, as each was conceived. His current interests include reactive functional programming, containers, Node JS, NoSQL, geospatial systems, mobile, and so on, in any new language or framework.

He occasionally blogs about software technology for Keyhole Software. He has coauthored three computer books and authored two IBM DeveloperWorks XML tutorials and a WebSphere Journal LDAP tutorial. Lou co-wrote several J2EE certification tests for IBM. He has also worked as a reviewer for Packt Publishing and others.

www.PacktPub.com

For support files and downloads related to your book, please visit www.PacktPub.com.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.PacktPub.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at service@packtpub.com for more details.

At www.PacktPub.com, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.

https://www.packtpub.com/mapt

Get the most in-demand software skills with Mapt. Mapt gives you full access to all Packt books and video courses, as well as industry-leading tools to help you plan your personal development and advance your career.

Why subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print, and bookmark content
- On demand and accessible via a web browser

Customer Feedback

Thanks for purchasing this Packt book. At Packt, quality is at the heart of our editorial process. To help us improve, please leave us an honest review on this book's Amazon page at `https://www.amazon.com/dp/1787121704`.

If you'd like to join our team of regular reviewers, you can e-mail us at `customerreviews@packtpub.com`. We award our regular reviewers with free eBooks and videos in exchange for their valuable feedback. Help us be relentless in improving our products!

试读结束：需要全本请在线购买： www.ertongbook.com

Table of Contents

Preface 1

Chapter 1: Revisiting Multithreading 7

- **Getting started** 7
- **The multithreaded application** 8
  - Makefile 12
- **Other applications** 14
- **Summary** 15

Chapter 2: Multithreading Implementation on the Processor and OS 17

- **Defining processes and threads** 18
  - Tasks in x86 (32-bit and 64-bit) 20
  - Process state in ARM 23
- **The stack** 24
- **Defining multithreading** 25
  - Flynn's taxonomy 27
  - Symmetric versus asymmetric multiprocessing 28
  - Loosely and tightly coupled multiprocessing 29
  - Combining multiprocessing with multithreading 29
  - Multithreading types 29
    - Temporal multithreading 29
    - Simultaneous multithreading (SMT) 30
- **Schedulers** 30
- **Tracing the demo application** 32
- **Mutual exclusion implementations** 34
  - Hardware 35
  - Software 35
- **Summary** 37

Chapter 3: C++ Multithreading APIs 39

- **API overview** 39
- **POSIX threads** 40
  - Windows support 42
  - PThreads thread management 43
  - Mutexes 45
  - Condition variables 46

Synchronization 48
Semaphores 49
Thread local storage (TLC) 49
Windows threads 50
Thread management 51
Advanced management 53
Synchronization 54
Condition variables 54
Thread local storage 55
Boost 55
Qt 55
QThread 56
Thread pools 57
Synchronization 57
QtConcurrent 58
Thread local storage 58
POCO 58
Thread class 59
Thread pool 59
Thread local storage (TLS) 60
Synchronization 61
C++ threads 62
Putting it together 62
Summary 63
Chapter 4: Thread Synchronization and Communication 65
Safety first 65
The scheduler 66
High-level view 66
Implementation 67
Request class 69
Worker class 71
Dispatcher 73
Makefile 77
Output 78
Sharing data 81
Using r/w-locks 82
Using shared pointers 82
Summary 82
Chapter 5: Native C++ Threads and Primitives 83

The STL threading API 83
Boost.Thread API 83
The 2011 standard 84
C++14 85
C++17 85
STL organization 86
Thread class 87
Basic use 87
Passing parameters 88
Return value 89
Moving threads 89
Thread ID 90
Sleeping 91
Yield 92
Detach 92
Swap 92
Mutex 93
Basic use 93
Non-blocking locking 95
Timed mutex 96
Lock guard 97
Unique lock 98
Scoped lock 99
Recursive mutex 99
Recursive timed mutex 100
Shared mutex 100
Shared timed mutex 101
Condition variable 101
Condition_variable_any 104
Notify all at thread exit 104
Future 105
Promise 106
Shared future 107
Packaged_task 108
Async 109
Launch policy 110
Atomics 110
Summary 110
Chapter 6: Debugging Multithreaded Code 111

When to start debugging 111
The humble debugger 112
GDB 113
Debugging multithreaded code 114
Breakpoints 115
Back traces 116
Dynamic analysis tools 118
Limitations 119
Alternatives 119
Memcheck 120
Basic use 120
Error types 123
Illegal read / illegal write errors 123
Use of uninitialized values 123
Uninitialized or unaddressable system call values 125
Illegal frees 127
Mismatched deallocation 127
Overlapping source and destination 127
Fishy argument values 128
Memory leak detection 128
Helgrind 129
Basic use 129
Misuse of the pthreads API 134
Lock order problems 135
Data races 136
DRD 136
Basic use 136
Features 138
C++11 threads support 139
Summary 140
Chapter 7: Best Practices 141
Proper multithreading 141
Wrongful expectations - deadlocks 142
Being careless - data races 146
Mutexes aren't magic 151
Locks are fancy mutexes 153
Threads versus the future 154
Static order of initialization 154
Summary 157
Chapter 8: Atomic Operations - Working with the Hardware 159

Atomic operations 159
Visual C++ 160
GCC 166
Memory order 169
Other compilers 170
C++11 atomics 170
Example 173
Non-class functions 174
Example 175
Atomic flag 177
Memory order 177
Relaxed ordering 178
Release-acquire ordering 178
Release-consume ordering 179
Sequentially-consistent ordering 179
Volatile keyword 180
Summary 180
Chapter 9: Multithreading with Distributed Computing 181
Distributed computing, in a nutshell 181
MPI 183
Implementations 184
Using MPI 185
Compiling MPI applications 186
The cluster hardware 187
Installing Open MPI 191
Linux and BSDs 191
Windows 191
Distributing jobs across nodes 193
Setting up an MPI node 194
Creating the MPI host file 194
Running the job 195
Using a cluster scheduler 195
MPI communication 196
MPI data types 197
Custom types 198
Basic communication 200
Advanced communication 201
Broadcasting 201
Scattering and gathering 202
MPI versus threads 203

Potential issues 205
Summary 205
Chapter 10: Multithreading with GPGPU 207
The GPGPU processing model 207
Implementations 208
OpenCL 209
Common OpenCL applications 209
OpenCL versions 210
OpenCL 1.0 210
OpenCL 1.1 210
OpenCL 1.2 211
OpenCL 2.0 212
OpenCL 2.1 212
OpenCL 2.2 213
Setting up a development environment 214
Linux 214
Windows 214
OS X/MacOS 215
A basic OpenCL application 215
GPU memory management 219
GPGPU and multithreading 221
Latency 222
Potential issues 222
Debugging GPGPU applications 223
Summary 224
Index 225

试读结束：需要全本请在线购买： www.ertongbook.com