C++ Concurrency IN ACTION

Practical Multithreading

Anthony Williams

MANNING

Table of Contents

Preface. C++ Concurrency in Action: Practical Multithreading, ISBN: 9781933988771

preface

I encountered the concept of multithreaded code while working at my first job after I left college. We were writing a data processing application that had to populate a database with incoming data records. There was a lot of data, but each record was independent and required a reasonable amount of processing before it could be inserted into the database. To take full advantage of the power of our 10-CPU UltraSPARC, we ran the code in multiple threads, each thread processing its own set of incoming records. We wrote the code in C++, using POSIX threads, and made a fair number of mistakes—multithreading was new to all of us—but we got there in the end. It was also while working on this project that I first became aware of the C++ Standards Committee and the freshly published C++ Standard.

I have had a keen interest in multithreading and concurrency ever since. Where others saw it as difficult, complex, and a source of problems, I saw it as a powerful tool that could enable your code to take advantage of the available hardware to run faster. Later on I would learn how it could be used to improve the responsiveness and performance of applications even on single-core hardware, by using multiple threads to hide the latency of time-consuming operations such as I/O. I also learned how it worked at the OS level and how Intel CPUs handled task switching.

Meanwhile, my interest in C++ brought me in contact with the ACCU and then the C++ Standards panel at BSI, as well as Boost. I followed the initial development of the Boost Thread Library with interest, and when it was abandoned by the original developer, I jumped at the chance to get involved. I have been the primary developer and maintainer of the Boost Thread Library ever since.

As the work of the C++ Standards Committee shifted from fixing defects in the existing standard to writing proposals for the next standard (named C++0x in the hope that it would be finished by 2009, and now officially C++11, because it was finally published in 2011), I got more involved with BSI and started drafting proposals of my own. Once it became clear that multithreading was on the agenda, I jumped in with both feet and authored or coauthored many of the multithreading and concurrency-related proposals that shaped this part of the new standard. I feel privileged to have had the opportunity to combine two of my major computer-related interests—C++ and multithreading—in this way.

This book draws on all my experience with both C++ and multithreading and aims to teach other C++ developers how to use the C++11 Thread Library safely and efficiently. I also hope to impart some of my enthusiasm for the subject along the way.