Энтони Уильямс

C++

Практика многопоточного программирования

ВТОРОЕ ИЗДАНИЕ

MANNING

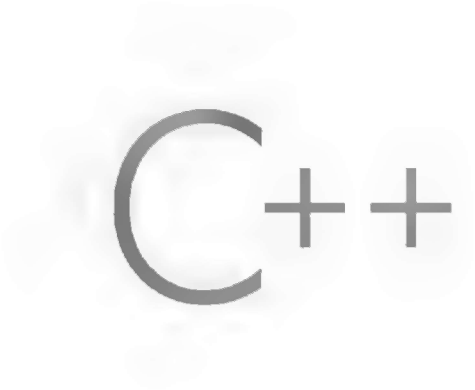
C++ Concurrency in Action

Second Edition

ANTHONY WILLIAMS

MANNING
SHELTER ISLAND

Энтони Уильямс



Практика многопоточного программирования

ПИТЕР®
Санкт-Петербург • Москва • Екатеринбург • Воронеж
Нижний Новгород • Ростов-на-Дону • Самара • Минск
2020

Энтони Уильямс

C++. Практика многопоточного программирования

Серия «Для профессионалов»

Перевел с английского *Н. Вильчинский*

Заведующая редакцией	*Ю. Сергиенко*
Руководитель проекта	*М. Колесников*
Ведущий редактор	*Н. Гринчик*
Литературные редакторы	*А. Дубейко, Н. Рощина*
Художественный редактор	*В. Мостипан*
Корректоры	*О. Андриевич, Е. Павлович, Т. Радецкая*
Верстка	*Г. Блинов*

ББК 32.973.2-018.1
УДК 004.43

Уильямс Энтони

У36 C++. Практика многопоточного программирования. — СПб.: Питер, 2020. — 640 с.: ил. — (Серия «Для профессионалов»).

ISBN 978-5-4461-0831-2

Язык C++ выбирают, когда надо создать по-настоящему молниеносные приложения. А качественная конкурентная обработка сделает их еще быстрее. Новые возможности C++17 позволяют использовать всю мощь многопоточного программирования, чтобы с легкостью решать задачи графической обработки, машинного обучения и др.

Энтони Уильямс, эксперт конкурентной обработки, рассматривает примеры и описывает практические задачи, а также делится секретами, которые пригодятся всем, в том числе и самым опытным разработчикам. Теперь вам доступны все аспекты конкурентной обработки на C++17 — от создания новых потоков до проектирования полнофункциональных многопоточных алгоритмов и структур данных.

16+ (В соответствии с Федеральным законом от 29 декабря 2010 г. № 436-ФЗ.)

ISBN 978-1617294693 англ.
ISBN 978-5-4461-0831-2

Изготовлено в России. Изготовитель: ООО «Прогресс книга».
Место нахождения и фактический адрес: 194044, Россия, г. Санкт-Петербург, Б. Сампсониевский пр., д. 29А, пом. 52. Тел.: +78127037373.

Дата изготовления: 12.2019. Наименование: книжная продукция. Срок годности: не ограничен.

Налоговая льгота — общероссийский классификатор продукции ОК 034-2014, 58.11.12 — Книги печатные профессиональные, технические и научные.

Импортер в Беларусь: ООО «ПИТЕР М», 220020, РБ, г. Минск, ул. Тимирязева, д. 121, э. к. 214, тел. факс: 208 80 01.

Подписано в печать 05.12.19. Формат 70×100/16. Бумага офсетная. Усл. п. л. 51,600. Тираж 1000. Заказ 0000.

Краткое содержание

Приложения

Оглавление

Приложения

Посвящается Ким, Хью и Эрин

Предисловие

С понятием многопоточного кода я столкнулся на своей первой работе, на которую устроился по окончании колледжа. Мы разрабатывали приложение для обработки данных, предназначенное для наполнения базы данных поступающими записями. Данных было много, но записи были независимы друг от друга, и требовалось немного обработать их перед добавлением. Чтобы по максимуму нагрузить наш компьютер UltraSPARC, имеющий десять центральных процессоров, мы запускали код в нескольких потоках и каждый поток обрабатывал собственный набор поступающих записей. Мы написали код на C++, используя потоки POSIX, допустили кучу ошибок — многопоточность для всех нас была в новинку, — но все же справились. Кроме того, работая над проектом, я узнал о существовании Комитета по стандартизации C++ и только что вышедшем стандарте C++.

С тех пор я очень заинтересовался многопоточностью и конкурентностью[1]. В том, что другим представлялось чем-то весьма сложным, запутанным и приносящим одни проблемы, я видел весьма эффективное средство, позволяющее программе за-

[1] В русскоязычной литературе нередко путаются термины «параллелизм» и «конкурентность». Оба термина означают одновременность процессов, но первый — на физическом уровне (параллельное исполнение нескольких процессов, нацеленное только на повышение скорости исполнения за счет использования соответствующей аппаратной поддержки), а второй — на логическом (парадигма проектирования систем, идентифицирующая процессы как независимые, что в том числе позволяет их исполнять физически параллельно, но в первую очередь нацелено на упрощение написания многопоточных программ и повышение их устойчивости) (источник: «Википедия»). В этой книге термин concurrency переводится как «конкурентность» и подразумевает одновременные вычисления, обеспечиваемые средствами конкретного языка и среды выполнения. Под одновременностью понимается не только и не столько выполнение сразу нескольких вычислений в одно и то же физическое время (что больше похоже на параллелизм), но и выполнение разных взаимосвязанных вычислений за один и тот же период времени в рамках одного приложения с использованием программных средств синхронизации (которая не нужна параллельным вычислениям в силу их независимости друг от друга). — *Примеч. ред.*

действовать все доступное оборудование и увеличить скорость своей работы. Чуть позже я освоил приемы использования этой технологии для повышения отзывчивости и производительности приложений даже на одноядерном оборудовании за счет применения нескольких потоков, позволяющих не замечать ожидания завершения таких продолжительных процессов, как ввод-вывод данных. Я также разобрался в том, как это все работает на уровне операционной системы и как центральные процессоры Intel справляются с переключением задач.

Интересуясь C++, я начал общаться с участниками Ассоциации пользователей языков C и C++ (ACCU), а затем и с представителями Комиссии по стандартизации C++ при Институте стандартов Великобритании (BSI), а также с разработчиками библиотеки Boost. Я с интересом наблюдал за началом разработки Boost Thread Library, а когда автор забросил проект, не упустил своего шанса принять участие в этом процессе. На протяжении нескольких лет я остаюсь основным разработчиком и сопровождающим Boost Thread Library.

По мере того как Комитет по стандартизации C++ перешел от устранения недочетов в существующем стандарте к выработке предложений по стандарту C++11 (обозначенному C++0x в надежде, что его разработка завершится до 2009-го, а позже официально названному C++11 по причине окончательной публикации в 2011 году), я еще больше втянулся в работу BSI и начал вносить собственные предложения. Как только стало ясно, что многопоточность приобрела особую актуальность, я всецело отдался ее продвижению и стал автором и соавтором многих предложений, касающихся многопоточности и конкурентности, сформировавших соответствующую часть стандарта. Совместно с группой по конкурентности участвовал в работе над внесением изменений для выработки стандарта C++17, спецификации Concurrency TS и предложений на будущее. Мне повезло в том, что таким образом удалось объединить две основные сферы моих компьютерных интересов — C++ и многопоточность.

В этой книге отражен весь мой опыт как в программировании на C++, так и в применении многопоточности, она предназначена для обучения других разработчиков программных средств на C++ приемам безопасного и эффективного использования C++17 Thread Library и спецификации Concurrency TS. Я также надеюсь заразить читателей своим энтузиазмом по части решения рассматриваемых проблем.

Благодарности

Прежде всего хочу сказать огромное спасибо своей жене Ким за ее любовь и поддержку, которую она мне оказывала, когда я писал книгу. Работа над первым изданием отнимала изрядную долю моего свободного времени на протяжении четырех лет. Над вторым изданием также пришлось немало потрудиться. Без терпения, поддержки и понимания Ким я бы не справился.

Далее хочу поблагодарить коллектив издательства Manning, сделавший возможным выпуск данной книги: издателя Марьяна Бэйса (Marjan Bace), соиздателя Майкла Стивенса (Michael Stephens), моего редактора-консультанта Синтию Кейн (Cynthia Kane), редактора-рецензента Александара Драгосавлевича (Aleksandar Dragosavljević), моих корректоров, компанию Safis Editing и Хайди Уорд (Heidi Ward), корректора Мелоди Долаб (Melody Dolab). Без их участия вы бы сейчас эту книгу не читали.

Хочу также поблагодарить представителей Комитета по стандартизации C++, составлявших документы по многопоточным средствам: Андрея Александреску (Andrei Alexandrescu), Пита Беккера (Pete Becker), Боба Блейнера (Bob Blainer), Ханса Бёма (Hans Boehm), Бемана Доуса (Beman Dawes), Лоуренса Кроула (Lawrence Crowl), Питера Димова (Peter Dimov), Джеффа Гарланда (Jeff Garland), Кевлина Хенни (Kevlin Henney), Ховарда Хиннанта (Howard Hinnant), Бена Хатчингса (Ben Hutchings), Яна Кристофферсона (Jan Kristofferson), Дага Ли (Doug Lea), Пола Маккенни (Paul McKenney), Ника Маклорена (Nick McLaren), Кларка Нельсона (Clark Nelson), Билла Пью (Bill Pugh), Рауля Сильвера (Raul Silvera), Херба Саттера (Herb Sutter), Детлефа Фольманна (Detlef Vollmann) и Майкла Вонга (Michael Wong), а также всех, кто комментировал документы, обсуждал их на заседаниях Комитета и иными путями помогал настроить поддержку многопоточности и конкурентности в C++11, C++14, C++17 и спецификации Concurrency TS.

И наконец, хочу с благодарностью отметить тех, чьи предложения позволили существенно улучшить книгу: доктора Джейми Оллсопа (Jamie Allsop), Питера Димова (Peter Dimov), Говарда Хиннанта (Howard Hinnant), Рика Моллоя (Rick Molloy), Джонатана Уэйкли (Jonathan Wakely) и доктора Рассела Уиндера (Russel

Winder). В особенности Рассела за его подробные рецензии и Фредерика Флайоля (Frédéric Flayol), технического корректора, в процессе производства тщательно проверившего окончательный вариант рукописи на явные ошибки. (Разумеется, все оставшиеся в тексте ляпы целиком на моей совести.) Кроме того, хочу поблагодарить группу рецензентов второго издания книги: Ала Нормана (Al Norman), Андрея де Араужо Формигу (Andrei de Araújo Formiga), Чада Брюбейкера (Chad Brewbaker), Дуайта Уилкинса (Dwight Wilkins), Хьюго Филипе Лопеса (Hugo Filipe Lopes), Виейру Дурана (Vieira Durana), Юру Шикина (Jura Shikin), Кента Р. Спиллнера (Kent R. Spillner), Мария Джемини (Maria Gemini), Матеуша Малента (Mateusz Malenta), Маурицио Томази (Maurizio Tomasi), Ната Луенгнарюмитчая (Nat Luengnaruemitchai), Роберта С. Грина II (Robert C. Green II), Роберта Трауесмута (Robert Trausmuth), Санчира Картиева (Sanchir Kartiev) и Стивена Парра (Steven Parr). Спасибо также читателям предварительного издания, не пожалевшим времени и указавшим на ошибки или отметившим текст, требующий пояснения.

О книге

Эта книга представляет собой подробное руководство по средствам поддержки конкурентности и многопоточности из нового стандарта C++, начиная от базового использования классов `std::thread`, `std::mutex` и `std::async` и заканчивая сложностями атомарных операций и модели памяти.

Структура издания

В главах 1–4 дается введение в различные библиотечные средства и порядок их возможного применения.

В главе 5 рассматриваются низкоуровневые подробности модели памяти и атомарные операции, включая порядок использования атомарных операций для наложения ограничений на порядок доступа к памяти со стороны другого кода. Ею завершаются вводные главы.

В главах 6 и 7 начинается изучение программирования на высоком уровне и приводится ряд примеров того, как использовать основные средства для создания более сложных структур данных — основанных на блокировках (в главе 6) и без блокировок (в главе 7).

В главе 8 продолжается рассмотрение высокоуровневых тем и даются рекомендации по разработке многопоточного кода. Здесь охвачены факторы, влияющие на производительность, и приведены примеры реализации различных параллельных алгоритмов.

В главе 9 речь идет об управлении потоками — пулах потоков, очередях работ и операциях, прерывающих выполнение потоков.

В главе 10 рассматриваются новые аспекты поддержки параллелизма, появившиеся в C++17 и представленные в виде дополнительных переопределений для многих алгоритмов стандартной библиотеки.

В главе 11 разговор идет о тестировании и отладке — типах ошибок, приемах обнаружения мест возникновения ошибок, порядке проведения тестирования на наличие ошибок и т. д.

В приложения включено краткое описание некоторых новых средств языка, введенных новым стандартом и имеющих отношение к многопоточности. Здесь также рассмотрены детали библиотеки передачи сообщений, упомянутой в главе 4, и приведен полный справочник по C++17 Thread Library.

Для кого предназначена эта книга

Эта книга для тех, кто создает многопоточные приложения на языке C++. Если вы пользуетесь новыми средствами многопоточности из стандартной библиотеки C++, то издание послужит руководством по основным вопросам. Если вы работаете с другими библиотеками многопоточности, то описанные рекомендации и приемы тоже могут стать полезным подспорьем.

Предполагается, что читатели обладают практическими навыками программирования на языке C++, возможно не зная о новых свойствах языка — их описание дано в приложении A. Также не требуются знания и навыки в области многопоточного программирования, хотя они были бы полезными.

Порядок чтения

Если ранее вам не приходилось создавать многопоточный код, советую читать книгу последовательно от начала до конца, возможно, опуская некоторые детали из главы 5. Материал главы 7 основан на информации, изложенной в главе 5, поэтому, если вы ее пропустите, то изучение главы 7 нужно отложить до тех пор, пока не будет изучена глава 5.

Если ранее вам не приходилось пользоваться новыми возможностями языка, появившимися в стандарте C++11, то, возможно, сначала стоит просмотреть приложение A, чтобы убедиться, что вам понятны примеры, приводимые в книге. Впрочем, в основном тексте упоминания о новых средствах графически выделены, поэтому, встретив что-то незнакомое, вы всегда можете обратиться к приложению.

Даже при наличии большого опыта создания многопоточного кода в других средах начальные главы все же стоит прочитать, чтобы увидеть, как уже известные вам понятия переносятся на те средства, что вошли в новый стандарт языка C++. Если есть намерение поработать с низкоуровневыми средствами с применением атомарных переменных, нужно обязательно изучить главу 5. А главу 8 стоит просмотреть, чтобы убедиться, что вы знакомы с такими вопросами, как безопасность выдачи исключений в многопоточном коде на C++. Если перед вами стоит конкретная задача, то быстро найти соответствующий раздел поможет оглавление.

Даже после освоения C++ Thread Library вам все равно пригодится материал приложения Г, например для поиска конкретных сведений о каждом классе или вызове функции. Также можно будет время от времени заглядывать в основные главы, освежая в памяти конкретные конструкции или просматривая примеры кода.

Условные обозначения и загрузка кода

Исходный код, приводимый в листингах и в тексте книги, набран моноширинным шрифтом. Многие листинги сопровождаются примечаниями, выделяющими важные понятия. Иногда примечания сопровождаются пронумерованными метками, фигурирующими в пояснениях, которые приводятся сразу за листингами.

Исходный код всех работоспособных примеров, приводимых в книге, доступен для загрузки с веб-сайта издательства по адресу www.manning.com/books/c-plus-plus-concurrency-in-action-second-edition. Исходный код можно загрузить также из хранилища GitHub по адресу https://github.com/anthonywilliams/ccia_code_samples.

Требования к программным средствам

Чтобы воспользоваться кодом, приведенным в книге, не внося в него каких-то изменений, понадобится самая последняя версия компилятора C++, поддерживающая средства языка C++17, перечисленные в приложении A, кроме того, нужна копия C++ Standard Thread Library.

Во время написания книги с реализацией C++17 Standard Thread Library уже поставлялись самые последние версии g++, clang++ и Microsoft Visual Studio. Они поддерживали большинство особенностей языка, рассмотренных в приложении A, а поддержка остальных возможностей ожидалась в ближайшее время.

Моя компания Just Software Solutions Ltd продает полную реализацию C++11 Standard Thread Library для ряда более старых компиляторов, а также реализацию спецификации Concurrency TS для новых версий clang, gcc и Microsoft Visual Studio[1]. Эта реализация использовалась для тестирования примеров, приведенных в книге.

Boost Thread Library[2], переносимая на многие платформы, предоставляет API, основанный на предложениях, касающихся развития C++11 Standard Thread Library. В большинство примеров из книги можно внести изменения для работы с Boost Thread Library, для чего нужно аккуратно заменить префикс `std::` на `boost::` и воспользоваться соответствующими директивами `#include`. Есть ряд средств, которые в Boost Thread Library либо не поддерживаются (например, `std::async`), либо называются иначе (например, `boost::unique_future`).

Форум, посвященный книге

Купив второе издание книги, вы получите свободный доступ к закрытому веб-форуму, организованному издательством Manning Publications, где можно оставить комментарий, касающийся текста, задать технические вопросы и получить помощь от автора или других пользователей. Для этого нужно перейти по адресу www.manning.com/books/c-plus-plus-concurrency-in-action-second-edition. Дополнительная информация

[1] Реализация just::thread библиотеки C++ Standard Thread Library, http://www.stdthread.co.uk.

[2] Коллекция библиотек Boost C++, http://www.boost.org.

о форумах издательства Manning и правилах поведения на них содержится по адресу https://forums.manning.com/forums/about.

В соответствии с обязательствами издательства Manning читателям предоставляется площадка, где они могут вести содержательный диалог между собой и с автором. При этом автор не обязан участвовать в диалоге, для него эта деятельность добровольная (и неоплачиваемая). Чтобы заинтересовать автора в разговоре с вами, задавайте ему сложные вопросы.

Форум и архивы предыдущих обсуждений будут доступны на сайте издательства на весь период допечатывания книги.

Об авторе

Энтони Уильямс (Anthony Williams) — британский разработчик, консультант и преподаватель с более чем 20-летним опытом программирования на C++. С 2001 года он принимает активное участие в деятельности группы по выработке стандартов BSI C++ и является автором или соавтором многих документов Комитета по стандартизации C++, благодаря которым библиотека потоков была включена в стандарт C++11. Он продолжает работать над новыми функциями, позволяющими усовершенствовать инструментарий для конкурентности на C++, а также над предложениями по стандартам и реализацией соответствующих средств расширения just::thread Pro для библиотеки потоков C++ от компании Just Software Solutions Ltd. Энтони живет на крайнем западе Англии, в графстве Корнуолл.

Об иллюстрации на обложке

Рисунок на обложке книги называется «Традиционный костюм японской женщины»[1]. Это репродукция из четырехтомника Томаса Джефферис (Thomas Jefferys) «Коллекция костюмов разных народов»[2], изданного в Лондоне в 1757–1772 годах. Коллекция включает отпечатанные с медных пластин и раскрашенные вручную гравюры с изображением одежды народов со всего мира. Издание существенно повлияло на дизайн театральных костюмов. Разнообразие рисунков в коллекции ярко свидетельствует о великолепии костюмов на Лондонской сцене более 200 лет назад. Можно было увидеть традиционную историческую и современную одежду людей, живших в разное время в разных странах, сделав их понятнее и ближе зрителям, посещавшим лондонские театры.

С тех пор стиль одежды сильно изменился и исчезло разнообразие, свойственное различным областям и странам. Теперь трудно различить по одежде даже жителей разных континентов. Если взглянуть на это с оптимистичной точки зрения, мы пожертвовали культурным и внешним разнообразием в угоду более насыщенной личной жизни или более разнообразной и интересной интеллектуальной и технической деятельности.

В наше время, когда трудно отличить одну компьютерную книгу от другой, издательство Manning проявляет инициативу и деловую сметку, украшая обложки книг изображениями, которые показывают богатое разнообразие жизни в регионах два века назад.

---

[1] Habit of a Lady of Japan.

[2] Collection of the Dress of Different Nations.

От издательства

Ваши замечания, предложения, вопросы отправляйте по адресу comp@piter.com (издательство «Питер», компьютерная редакция).

Мы будем рады узнать ваше мнение!

На веб-сайте издательства www.piter.com вы найдете подробную информацию о наших книгах.