Посвящается моим жене и сыну,
Кирти Танвар и Шашвату Сингх Танвар.
Они мой жизненно необходимый кислород и стимул,
вдохновляющий меня на достижение выдающихся успехов

Содержание

От издательства

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв на нашем сайте www.dmkpress.com, зайдя на страницу книги и оставив комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу dmkpress@gmail.com; при этом укажите название книги в теме письма.

Если вы являетесь экспертом в какой-либо области и заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу http://dmkpress.com/authors/publish_book/ или напишите в издательство по адресу dmkpress@gmail.com.

Скачивание исходного кода примеров

Скачать файлы с дополнительной информацией для книг издательства «ДМК Пресс» можно на сайте www.dmkpress.com на странице с описанием соответствующей книги.

Список опечаток

Хотя мы приняли все возможные меры для того, чтобы обеспечить высокое качество наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг, мы будем очень благодарны, если вы сообщите о ней главному редактору по адресу dmkpress@gmail.com. Сделав это, вы избавите других читателей от недопонимания и поможете нам улучшить последующие издания этой книги.

Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательства «ДМК Пресс» и Packt Publishing очень серьезно относятся к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконной публикацией какой-либо из наших книг, пожалуйста, пришлите нам ссылку на интернет-ресурс, чтобы мы могли применить санкции.

Ссылку на подозрительные материалы можно прислать по адресу электронной почты dmkpress@gmail.com.

Мы высоко ценим любую помощь по защите наших авторов, благодаря которой мы можем предоставлять вам качественные материалы.

Об авторе

Шакти Танвар является генеральным директором Techpro Compsoft Pvt Ltd, глобального поставщика консалтинговых услуг в области информационных технологий. Шакти – IT-евангелист и архитектор программного обеспечения с 15-летним опытом работы в области разработки программного обеспечения и корпоративного обучения. Он также является сертифицированным преподавателем Microsoft и проводит обучение в сотрудничестве с Microsoft на Ближнем Востоке.

Шакти Танвар специализируется в таких областях, как .NET, машинное обучение в Azure, искусственный интеллект, применение чистого функционального программирования для построения отказоустойчивых систем и параллельные вычисления.

Его любовь к преподаванию привела к тому, что он запустил специальную программу «Обучение профессоров» с целью улучшения работы колледжей в Индии.

Эта книга была бы невозможна без неоценимой помощи моей жены Кирти и моего сына Шашвата, разделивших со мной все взлеты и падения. Благодаря их поддержке и желанию действовать я продолжал двигаться вперед в тяжелое время.

Я бесконечно благодарен своим родителям, братьям и сестрам, которые всегда побуждали меня к достижению новых высот.

Огромное спасибо моим друзьям, наставникам и команде Packt, которые сопровождали меня на протяжении всего пути.

О переводе

Данная книга далась нам непросто и потребовала больше года на перевод и кропотливую редактуру, но оно того стоило. Тема разработки параллельных программ сейчас актуальна как никогда, хотя и является очень непростой для понимания. В этой книге автор изложил не только сложные технические аспекты написания параллельных программ, но и объяснил механизмы реализации многопоточности в .NET, а также особенности языка C#.

Над переводом этой книги работали специалисты компании Devs Universe:

- **Алина Воронина** – переводчик, специалист по обучению разработчиков английскому языку в компании Devs Universe;
- **Вячеслав Черников** – редактор перевода, к. т. н. в области разработки ПО, основатель компании Devs Universe, автор книги «Разработка мобильных приложений на C# для iOS и Android», в прошлом – один из Microsoft MVP, Nokia Champion, Qt Certified Specialist, Qt Ambassador, автор статей для «Хабрахабра», «Хакера», Microsoft Developer Blogs, говоритель для конференций;
- **Максим Веркошанский, Дмитрий Милкин, Марина Королькова** – помощь с редактурой, специалисты компании Devs Universe.

Мы надеемся, что наши переводы помогут вам глубже понять суть современных технологий и стать суперразработчиками.

О рецензентах

Элвин Эшкрафт – разработчик, живущий недалеко от Филадельфии. Он провел свою 23-летнюю карьеру, создавая программное обеспечение при помощи C#, Visual Studio, WPF, ASP.NET и т. д. Был удостоен награды Microsoft Most Valuable Professional девять раз. Вы можете увидеть его ежедневные подборки ссылок в блоге для .Net-разработчиков *Morning Dew*. Ранее Элвин работал в софтверных компаниях, включая Oracle, сейчас он является главным специалистом по разработке ПО в Allscripts, создавая программное обеспечение для здравоохранения. Для Packt Publishing им также были написаны и другие рецензии на такие книги, как *Mastering ASP.NET Core 2.0, Mastering Entity Framework Core 2.0* и *Learning ASP.NET Core 2.0*.

Я хотел бы поблагодарить свою замечательную жену Стелену и трех наших очаровательных дочерей за их поддержку и понимание. Многие вечера и выходные дни были посвящены чтению и пересмотру глав этой книги, для того чтобы она смогла выйти в свет – первоклассная и полезная книга для разработчиков .NET.

Видья Врат Агарвал – любитель книг, спикер, автор публикаций для Apress и технический редактор более чем дюжины книг Apress, Packt и O'Reilly. Он является прикладным разработчиком с 20-летним опытом в области проектирования, создания и разработки распределенных программных решений для крупных предприятий. Будучи главным архитектором в T-Mobile, Видья Врат Агарвал работал с проектами B2C и B2B. Сейчас он также продолжает сотрудничество с другими архитекторами с целью разработки решений и дорожных карт для различных проектов T-Mobile для миллионов клиентов компании. Он рассматривает разработку программного обеспечения как ремесло и является большим сторонником архитектуры программного обеспечения и практики чистого кода (clean code).

Предисловие

Прошел почти год с момента, когда издательская компания Packt впервые связалась со мной по поводу книги. Я и предположить не мог, что этот долгий путь будет таким сложным, однако за это время я смог многому научиться. Книга, которую вы сейчас держите в руках, – это результат, который стоил долгих дней трудов, и я горжусь тем, что наконец-то представляю ее вам.

Процесс создания этой книги очень много для меня значит, так как я всегда мечтал написать о языке, с которого начинал свою карьеру. Язык программирования C# развивался очень стремительно, а платформа .NET Core еще сильнее улучшила его репутацию в сообществе разработчиков.

Для того чтобы книга была полезна широкому кругу разработчиков, мы поговорим как о классическом подходе, построенном на потоках (threads), так и о разработке с использованием библиотеки TPL (Task Parallel Library). Вначале будут рассмотрены основные концепции операционных систем (ОС), которые позволяют писать многопоточный код. Затем мы проанализируем различия между классическим подходом и TPL.

В этой книге я постараюсь подойти к параллельному программированию со стороны современных реалий. Все примеры будут короткими и простыми, чтобы облегчить ваше понимание. Содержание глав построено так, чтобы информация легко усваивалась, даже если вы не обладаете специальными знаниями.

Надеюсь, вы получите такое же удовольствие от чтения этой книги, как и я от ее написания.

Целевая аудитория

Данная книга предназначена для программистов C#, которые хотят изучить концепции параллельного программирования и многопоточности, а также использовать полученные знания для своих приложений, построенных на базе .NET Core. Книга будет полезна студентам и специалистам, желающим познакомиться с принципами работы параллельного программирования на современном оборудовании.

Предполагается, что вы уже имеете представление о C# и базовые знания о том, как работают операционные системы.

Краткий обзор

Глава 1 «Введение в параллельное программирование», в которой представлены важные понятия многопоточности и параллельного программирования, включает в себя описание того, как развивались операционные системы

для поддержки современных подходов параллельного программирования.

Глава 2 «Параллелизм задач» демонстрирует возможности разделения программы на задачи (tasks) с целью эффективного использования ресурсов процессора и повышения производительности.

В главе 3 «Реализация параллелизма данных», в которой основное внимание уделяется реализации параллелизма данных с использованием параллельных циклов, также рассматриваются дополнительные методы (extension methods), помогающие в достижении параллелизма, а также разделении данных.

Глава 4 «Использование PLINQ» рассказывает о преимуществах использования PLINQ, включая отмену запросов. Также рассматриваются особенности применения PLINQ.

В главе 5 «Примитивы синхронизации» рассматриваются конструкции, доступные в C# для работы с разделяемыми (shared) ресурсами в многопоточном коде.

Глава 6 «Использование параллельных коллекций» описывает использование преимуществ параллельных коллекций, доступных в .NET Core.

Глава 7 «Повышение производительности с помощью отложенной инициализации» посвящается повышению производительности с помощью паттерна отложенной (lazy, «ленивой») инициализации.

В главе 8 «Введение в асинхронное программирование» рассматривается то, как нужно писать асинхронный код в более ранних версиях .NET.

В главе 9 «Основы асинхронного программирования с помощью async, await и задач» рассказывается о том, как использовать преимущества на новых конструкциях в .NET Core для реализации асинхронного кода.

Глава 10 «Отладка задач с Visual Studio» посвящена различным инструментам, доступным в Visual Studio 2019, которые облегчают отладку параллельных задач (tasks).

В главе 11 «Создание модульных тестов для параллельного и асинхронного кодов» рассматриваются различные способы написания тестов в Visual Studio и .NET Core.

Глава 12 «IIS и Kestrel в ASP.NET Core» расскажет, что такое IIS и Kestrel и как этим пользоваться. В этой главе также рассматривается поддержка асинхронных потоков.

Глава 13 «Шаблоны параллельного программирования» описывает различные паттерны (patterns), уже реализованные в языке C#. Она также включает в себя примеры реализации таких шаблонов.

В главе 14 «Управление распределенной памятью» рассматривается совместное использование памяти в распределенных программах.

Прежде чем начать

Вам необходимо установить Visual Studio 2019 вместе с .NET Core 3.1.

Также рекомендуется иметь базовые знания в языке C# и механизмах работы операционных систем.

Скачайте примеры исходных кодов

Вы можете скачать примеры исходных кодов для этой книги по адресу www.packtpub.com. Если вы приобрели эту книгу в другом месте, то можете перейти на сайт https://www.packtpub.com/support и получить примеры по электронной почте.

Шаги по загрузке файлов:

1. Войдите в систему или зарегистрируйтесь на сайте www.packtpub.com.
2. Выберите вкладку **Support**.
3. Нажмите на кнопку **Code Downloads**.
4. Введите название книги на английском языке «Hands-On Parallel Programming with C# 8 and .NET Core 3» в поле поиска и следуйте инструкциям.

Как только файл будет загружен, убедитесь, что вы его распаковываете, используя последнюю версию программ:

1) WinRAR/7-Zip для Windows;
2) Zipeg/iZip/UnRarX для Mac;
3) 7-Zip/PeaZip для Linux.

Примеры кода для этой книги также размещены на GitHub: https://github.com/PacktPublishing/Hands-On-Parallel-Programming-with-C-8-and-NET-core-3.

При обновлении исходного кода он тоже меняется на GitHub.

В нашем обширном каталоге книг и видео, доступных по ссылке ниже, представлены и другие примеры: https://github.com/PacktPublishing/. Обязательно их посмотрите.

Изображения

Также предоставляем вам PDF-файл с использованными в книге цветными изображениями скриншотов/диаграмм, которые вы можете скачать по ссылке https://static.packt-cdn.com/downloads/9781789132410_ColorImages.pdf.

Условные обозначения

В данной книге используется целый ряд условных обозначений.

`CodeInText`: указывает кодовые слова, например названия таблиц в базах данных, имена папок, имена файлов и их расширения, имена путей, вводимую пользователем информацию или имена пользователей Twitter. Допустим: «Подключите загруженный образ диска `WebStorm-10*.dmg` в качестве нового диска вашей системы».

Блок кода обозначается следующим образом:

```
private static void PrintNumber10Times() {
  for (int i = 0; i < 10; i++) {
```

```
    Console.Write(1);
  }
  Console.WriteLine();
}
```

Некоторые строки или элементы кода могут быть выделены жирным шрифтом с целью привлечения внимания к ним:

```
private static void PrintNumber10Times() {
  for (int i = 0; i < 10; i++) {
    Console.Write(1);
  }
  onsole.WriteLine();
}
```

Жирным шрифтом обозначается новый термин, важное слово или слова, которые вы видите в диалоговых сообщениях. Пример: «Вместо того чтобы самим находить оптимальное количество потоков, мы можем предоставить это среде **Common Language Runtime**».

Предупреждение или важная информация.

Советы и рекомендации.

Часть I

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ РАБОТЫ С ПОТОКАМИ, МНОГОЗАДАЧНОСТИ И АСИНХРОННОСТИ

В данной части вы познакомитесь с понятиями потока, многозадачности и асинхронного программирования.

Содержание части I включает в себя следующие главы:

- глава 1 «Введение в параллельное программирование»;
- глава 2 «Параллелизм задач»;
- глава 3 «Реализация параллелизма данных»;
- глава 4 «Использование PLINQ».

Глава 1

Введение в параллельное программирование

Возможность использования параллельного программирования реализована в .NET с самого начала и после появления **Task Parellel Library** (TPL) в .NET Framework 4.0 получила широкое распространение.

Многопоточность (multithreading) является подмножеством параллельного программирования и выступает одной из самых сложных тем для начинающих разработчиков. Язык программирования C# заметно эволюционировал с момента своего появления и сейчас может быть использован не только для «классической» многопоточности, но и для «современного» асинхронного программирования. Многопоточность C# уходит своими корнями в версию 1.0. Язык C# является преимущественно синхронным, однако в версии 5.0 была добавлена поддержка асинхронности, которая сделала его отличным выбором для прикладных программистов. В то время как многопоточность имеет дело только с распараллеливанием внутри самих процессов, параллельное программирование также имеет дело с механизмами межпроцессного взаимодействия (inter-process communication, IPC).

До появления TPL использовались классы `Thread`, `BackgroundWorker` и `ThreadPool`, которые позволяли реализовать многопоточность. C# 1.0 опирался на потоки (threads) для отделения фоновой работы от **обработки событий пользовательского интерфейса** (user interface, или UI), что позволяло разрабатывать отзывчивые приложения. Эта модель теперь называется классической работой с потоками (classic threading). Со временем она уступила место другой модели программирования, называемой TPL, которая опирается на задачи (tasks) и скрывает от разработчика работу с потоками.

В этой главе мы познакомимся с различными концепциями, которые помогут вам научиться писать многопоточный код с нуля.

В главе 1 будут освещены следующие темы:

- основные понятия многоядерных вычислений, начиная с общих концепций и процессов **операционной системы** (ОС);
- потоки и разница между многопоточностью и многозадачностью;

- преимущества и недостатки написания параллельного кода и сценариев, в которых целесообразно использование параллельного программирования.

Технические требования

Продемонстрированные в данной книге примеры были созданы в Visual Studio 2019 при помощи C# 8. Со всеми исходниками вы можете ознакомиться на сайте: https://github.com/PacktPublishing/Hands-On-Parallel-Programming-with-C-8-and-.NET-Core-3/tree/master/Chapter01.

Подготовка к многоядерным вычислениям

В этом разделе вам будут представлены основные концепции ОС, начиная с процесса (process), внутри которого живут и исполняются потоки (threads). Далее мы рассмотрим развитие многозадачности с момента появления аппаратных возможностей, которые способствовали развитию параллельного программирования. После этого мы попытаемся разобраться в различных способах создания потоков в коде приложений.

Процессы

Говоря простым языком, слово «процесс» (process) относится к программе, которая запущена на компьютере. Однако с точки зрения ОС процесс – это адресное пространство в оперативной памяти. Каждому приложению нужны процессы для запуска, вне зависимости от того, написано ли это приложение для смартфона, Windows или веб-браузера. Процессы обеспечивают защиту одних программ от других, работающих в той же системе: выделенные одной программой данные не могут случайным образом быть доступными для другой. Кроме этого, процессы также обеспечивают изоляцию, при которой программы могут запускаться и останавливаться независимо друг от друга и от самой ОС.

Дополнительно об ОС

Производительность приложений во многом зависит от конфигурации аппаратного обеспечения, которая включает в себя следующее:

- скорость центрального процессора;
- объем оперативной памяти;
- скорость жесткого диска (HDD);
- тип диска, то есть HDD или SSD.

За последние несколько десятилетий мы стали свидетелями существенного прогресса в области аппаратных технологий. Например, микропроцес-