

Даниэль Куссвюрм

Профессиональное программирование на ассемблере

x64

С расширениями
AVX, AVX2 и AVX-512

Modern X86 Assembly Language Programming

**Covers x86 64-bit, AVX, AVX2,
and AVX-512**

Second Edition

Daniel Kusswurm

Профессиональное программирование на ассемблере x64 с расширениями AVX, AVX2 и AVX-512

Даниэль Куссьюрм



Москва, 2021

УДК 004.4
ББК 32.972
К94

Даниэль Куссуорм

К94 Профессиональное программирование на ассемблере x64 с расширениями AVX, AVX2 и AVX-512 / пер. с англ. В. С. Яценкова. – М.: ДМК Пресс, 2021. – 628 с.: ил.

ISBN 978-5-97060-928-6

В книге рассматривается программирование для 64-разрядной архитектуры x86 и использование расширенного набора векторных команд (AVX). Изучив этот материал, вы сможете кодировать быстродействующие функции и алгоритмы с использованием 64-разрядного языка ассемблера x86 и расширений набора команд AVX, AVX2 и AVX-512.

Примеры исходного кода разработаны с использованием Visual Studio C++ и MASM; для их запуска рекомендуется ПК на базе x86 с 64-разрядной ОС Windows 10 и процессором, поддерживающим AVX.

Предполагается, что читатели имеют опыт программирования на языках высокого уровня и базовые знания C++.

Книга предназначена разработчикам, которые хотят научиться писать код с использованием языка ассемблера x64.

УДК 004.4
ББК 32.972

Modern X86 Assembly Language Programming; Covers x86 64-bit, AVX, AVX2, and AVX-512 by Daniel Kusswurm, edition: 2

This edition has been translated and published under licence from APress Media, LLC, part of Springer Nature.

APress Media, LLC, part of Springer Nature takes no responsibility and shall not be made liable for the accuracy of the translation.

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

ISBN (англ.) 978-1-4842-4062-5
ISBN (рус.) 978-5-97060-928-6

© 2018 by Daniel Kusswurm
© Оформление, издание, перевод, ДМК Пресс, 2021

Оглавление

Предисловие от издательства	11
Об авторе	12
О техническом редакторе	13
Благодарности	14
Вступление	15
О чем эта книга	16
Глава 1. Архитектура ядра x86-64	19
1.1. Исторический обзор.....	19
1.2. Типы данных.....	22
1.2.1. Основные типы данных.....	22
1.2.2. Числовые типы данных.....	24
1.2.3. Типы данных SIMD.....	24
1.2.4. Прочие типы данных.....	26
1.3. Внутренняя архитектура.....	26
1.3.1. Регистры общего назначения.....	27
1.3.2. Регистр RFLAGS.....	29
1.3.3. Указатель команд.....	31
1.3.4. Операнды команд.....	32
1.3.5. Адресация памяти.....	33
1.4. Различия между программированием x86-64 и x86-32.....	35
1.4.1. Недопустимые команды.....	37
1.4.2. Устаревшие команды.....	38
1.5. Обзор набора команд.....	38
1.6. Заключение.....	41
Глава 2. Программирование ядра x86-64. Часть 1	42
2.1. Простая целочисленная арифметика.....	42
2.1.1. Сложение и вычитание.....	43
2.1.2. Логические операции.....	46
2.1.3. Операции сдвига.....	49
2.2. Расширенная целочисленная арифметика.....	53
2.2.1. Умножение и деление.....	53
2.2.2. Вычисления с использованием смешанных типов.....	57

2.3. Команды адресации памяти и состояния	63
2.3.1. Режимы адресации памяти	63
2.3.2. Условные команды	68
2.4. Заключение	72
Глава 3. Программирование ядра x86-64. Часть 2	74
3.1. Массивы	74
3.1.1. Одномерные массивы.....	74
3.1.2. Двумерные массивы	81
3.3. Строки.....	94
3.3.1. Подсчет символов	94
3.3.2. Конкатенация строк.....	97
3.3.3. Сравнение массивов	103
3.3.4. Обращение массива	106
3.4. Заключение	110
Глава 4. Векторное расширение набора команд AVX	111
4.1. Обзор AVX.....	111
4.2. Концепции программирования SIMD	113
4.3. Арифметика с переносом или арифметика с насыщением?	114
4.4. Среда выполнения AVX	116
4.4.1. Набор регистров.....	116
4.4.2. Типы данных	117
4.4.3. Синтаксис команд	118
4.5. Скалярные вычисления AVX с плавающей запятой	119
4.5.1. Концепция программирования с плавающей запятой.....	119
4.5.2. Набор скалярных регистров с плавающей запятой.....	123
4.5.3. Регистр управления и состояния	123
4.5.4. Обзор набора команд.....	125
4.6. Операции с упакованными числами с плавающей запятой в AVX	126
4.6.1. Обзор набора команд.....	129
4.7. Операции с упакованными целыми числами в AVX.....	130
4.7.1. Обзор набора команд	131
4.8. Различия между x86-AVX и x86-SSE	133
4.9. Заключение	135
Глава 5. Программирование AVX – скалярные вычисления с плавающей запятой	137
5.1. Скалярная арифметика с плавающей запятой	138
5.1.1. Арифметика с плавающей запятой одинарной точности.....	138
5.1.1. Арифметика с плавающей запятой двойной точности.....	141
5.2. Скалярные сравнения и преобразования с плавающей запятой	146
5.2.1. Сравнение с плавающей запятой	147
5.2.2. Преобразования чисел с плавающей запятой	156

5.3. Скалярные массивы и матрицы с плавающей запятой	163
5.3.1. Массивы значений с плавающей запятой	163
5.3.2. Матрицы значений с плавающей запятой	167
5.4. Соглашение о вызовах	171
5.4.1. Базовые фреймы стека	172
5.4.2. Использование энергонезависимых регистров общего назначения	176
5.4.3. Использование энергонезависимых регистров XMM	181
5.4.4. Макросы для прологов и эпилогов	187
5.5. Заключение	194

Глава 6. Программирование AVX – упакованные числа

с плавающей запятой	196
6.1. Упакованная арифметика с плавающей запятой	196
6.2. Сравнение упакованных чисел с плавающей запятой	203
6.3. Преобразования упакованных чисел с плавающей запятой	208
6.4. Массивы упакованных чисел с плавающей запятой	213
6.4.1. Квадратные корни из упакованных чисел с плавающей запятой	213
6.4.2. Поиск минимального и максимального значений массива упакованных значений с плавающей запятой	217
6.4.3. Наименьшие квадраты упакованных чисел с плавающей запятой	222
6.5. Упакованные матрицы значений с плавающей запятой	228
6.5.1. Транспонирование матрицы	229
6.5.2. Умножение матриц	236
6.6. Заключение	242

Глава 7. Программирование AVX –

упакованные целые числа.....	244
7.1. Сложение и вычитание упакованных целых чисел	244
7.2. Сдвиг упакованных целых чисел.....	250
7.3. Умножение упакованных целых чисел	255
7.4. Обработка изображений с применением упакованных целых чисел ..	261
7.4.1. Минимальные и максимальные значения пикселей.....	262
7.4.2. Средняя интенсивность пикселей.....	270
7.4.3. Преобразования пикселей	275
7.4.4. Гистограммы изображений	283
7.4.5. Пороговая обработка изображений	290
7.5. Заключение	302

Глава 8. Подробнее про AVX2

8.1. Среда выполнения AVX2.....	304
8.2. Команды AVX2 для упакованных чисел с плавающей запятой	305
8.3. Команды AVX2 для упакованных целых чисел	307

8.4. Расширения набора команд X86	308
8.4.1. Числа с плавающей запятой половинной точности	308
8.4.2. Слитное умножение-сложение (FMA)	309
8.4.3. Расширения набора команд для регистров общего назначения....	311
8.5. Заключение	312

Глава 9. Программирование AVX2 – упакованные числа

с плавающей запятой	314
9.1. Арифметика упакованных чисел с плавающей запятой	315
9.2. Массивы упакованных чисел с плавающей запятой	321
9.2.1. Простые вычисления	321
9.2.2. Среднее арифметическое значение столбца	328
9.2.3. Коэффициент корреляции.....	334
9.3. Умножение и транспонирование матриц	341
9.4. Обращение матриц	349
9.5. Команды смешивания и перестановки	361
9.6. Команды извлечения данных	367
9.7. Заключение	373

Глава 10. Программирование AVX2 –

упакованные целые числа	375
10.1. Основные операции над упакованными целыми числами	375
10.1.1. Основные арифметические операции	376
10.1.2. Упаковка и распаковка.....	380
10.1.3. Увеличение размера.....	386
10.2. Обработка изображений с упакованными целочисленными пикселями	391
10.2.1. Усечение пикселей	391
10.2.2. Поиск минимального и максимального значений RGB	396
10.2.3. Преобразование RGB в оттенки серого.....	403
10.3. Заключение.....	411

Глава 11. Программирование AVX2 –

расширенные команды	412
11.1. Программирование операций FMA	412
11.1.1. Свертки	413
11.1.2. Скалярные операции FMA.....	415
11.1.3. Операции FMA с упакованными операндами	424
11.2. Команды для работы с регистрами общего назначения	431
11.2.1. Бесфлаговое умножение и сдвиги.....	432
11.2.2. Расширенные манипуляции битами	436
11.3. Преобразования с плавающей запятой половинной точности	440
11.4. Заключение.....	443

Глава 12. Система векторных команд AVX-512	445
12.1. Обзор AVX-512	445
12.2. Среда выполнения AVX-512.....	446
12.2.1. Наборы регистров	447
12.2.2. Типы данных	448
12.2.3. Синтаксис команды	448
12.3. Обзор набора команд.....	452
12.3.1. AVX512F	453
12.3.2. AVX512CD.....	455
12.3.3. AVX512BW	456
12.3.4. AVX512DQ	456
12.3.5. Регистры маски операции	457
12.4. Заключение.....	458
 Глава 13. Программирование AVX-512 –	
числа с плавающей запятой	459
13.1. Скалярные операнды с плавающей точкой	459
13.1.1. Маскирование слиянием	460
13.1.2. Маскирование нулем	463
13.1.3. Округление на уровне команды.....	466
13.2. Упакованные числа с плавающей запятой.....	470
13.2.1. Арифметика упакованных чисел с плавающей запятой.....	471
13.2.2. Сравнение упакованных чисел с плавающей запятой	478
13.2.3. Средние значения столбца упакованных чисел с плавающей запятой	483
13.2.4. Векторные перекрестные произведения	491
13.2.5. Умножение матрицы на вектор	501
13.2.6. Свертки	512
13.3. Заключение.....	516
 Глава 14. Программирование AVX-512 –	
упакованные целые числа.....	517
14.1. Базовая арифметика	517
14.2. Обработка изображений.....	523
14.2.1. Пиксельные преобразования	523
14.2.2. Пороговая обработка изображений	530
14.2.3. Статистика изображений	535
14.2.4. Преобразование формата RGB в оттенки серого	546
14.3. Заключение.....	553
 Глава 15. Стратегии и методы оптимизации	555
15.1. Микроархитектура процессора	555
15.1.1. Обзор архитектуры процессора	556

15.1.2. Функциональная схема конвейера микроархитектуры	557
15.1.3. Механизм выполнения	560
15.2. Оптимизация кода на языке ассемблера	561
15.2.1. Основные методы	562
15.2.2. Операции с плавающей запятой	563
15.2.3. Ветвление программы	564
15.2.4. Выравнивание данных	565
15.2.5. Методы SIMD	566
15.3. Заключение	567
Глава 16. Продвинутое программирование	568
16.1. Команда CPUID	568
16.2. Постоянные хранилища в памяти	584
16.3. Предварительная выборка данных	589
16.4. Многопоточность	596
16.5. Заключение	609
Приложение	611
П.1. Программные утилиты для процессоров x86	611
П.2. Visual Studio	611
П.2.1. Запуск примера исходного кода	612
П.2.2. Создание проекта Visual Studio C ++	612
П.3. Основные и дополнительные материалы	620
П.3.1. Справочные руководства по программированию на X86	620
П.3.2. Справочные материалы по x86 и микроархитектуре	621
П.3.3. Вспомогательные ресурсы	622
П.3.4. Ссылки на ресурсы по алгоритмам	622
П.3.5. Ссылки на ресурсы по C ++	623
Предметный указатель	624

Предисловие 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**.

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 – возможно, ошибку в основном тексте или программном коде, – мы будем очень благодарны, если вы сообщите нам о ней. Сделав это, вы избавите других читателей от недопонимания и поможете нам улучшить последующие издания этой книги.

Если вы найдете какие-либо ошибки в коде, пожалуйста, сообщите о них главному редактору по адресу **dmkpress@gmail.com**, и мы исправим это в следующих тиражах.

Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательство «ДМК Пресс» очень серьезно относится 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 **dmkpress@gmail.com**.

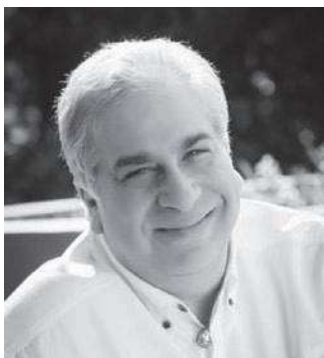
Мы высоко ценим любую помощь по защите наших авторов, благодаря которой мы можем предоставлять вам качественные материалы.

Об авторе



Даниэль Куссвюрм более 30 лет занимается разработкой программного обеспечения и информационными технологиями. За свою карьеру он разработал инновационное программное обеспечение для различного медицинского и научного оборудования и приложения для обработки изображений. Во многих из этих проектов он успешно использовал язык ассемблера x86 для значительного повышения быстродействия ресурсоемких алгоритмов и решения уникальных задач программирования. Он получил степень бакалавра электроники и электротехники в Университете Северного Иллинойса, а также степень магистра и доктора информатики в Университете ДеПола.

О техническом редакторе



Пол Коэн пришел в корпорацию Intel на самых первых этапах развития архитектуры x86, начиная с 8086, и ушел из Intel после 26 лет работы в сфере продаж, маркетинга и управления. В настоящее время он является партнером Douglas Technology Group и специализируется на написании технической литературы по заказу Intel и других корпораций. Пол также совместно с Академией молодых предпринимателей (YEA) ведет учебный курс, который превращает учеников средних и старших классов в настоящих, уверенных в себе предпринимателей. Он также является комиссаром по вопросам дорожного движения города Бивертон, штат Орегон, и входит в совет директоров нескольких некоммерческих организаций.

просам дорожного движения города Бивертон, штат Орегон, и входит в совет директоров нескольких некоммерческих организаций.

Благодарности

Производство фильма и издание книги в чем-то похожи. Трейлеры к фильму чествуют актеров, играющих главных героев. На обложке книги звучат имена авторов. Актеры и авторы в конечном итоге получают признание публики за свои усилия. Однако невозможно создать фильм или опубликовать книгу без преданности делу, опыта и творческого подхода профессиональной закадровой команды. Эта книга не исключение.

Я хочу поблагодарить за прекрасную работу талантливый редакционный коллектив Apress, особенно Стива Энглина, Марка Пауэрса и Мэтью Муди. Пол Коэн заслуживает отдельной похвалы за его тщательный технический обзор и практические предложения. Корректор Эд Куссвюрм заслуживает аплодисментов и признательности за его упорный труд и конструктивные отзывы. Я беру на себя полную ответственность за любые оставшиеся недостатки.

Я также хочу поблагодарить Нирмала Селвараджа, Дулси Нирмала, Кезию Эндсли, Дханеша Кумара и весь редакционный персонал Apress за их вклад, а моих профессиональных коллег за их одобрение и поддержку. Наконец, я хотел бы поблагодарить родителей Армина (вечная ему память) и Мэри, а также двоюродную сестру Мэри и двоюродных братьев Тома, Эда и Джона, которые вдохновляли меня на написание этой книги.

Вступление

С момента изобретения персонального компьютера (ПК) разработчики программного обеспечения использовали язык ассемблера x86 для создания инновационных решений применительно к широкому спектру алгоритмических задач. В первые дни эры ПК было обычной практикой писать большие фрагменты программного кода или полные приложения с использованием языка ассемблера x86. Учитывая преобладание в XXI веке языков высокого уровня, таких как C++, C#, Java и Python, может быть удивительно узнать, что многие разработчики программного обеспечения все еще используют язык ассемблера для кодирования критически важных в плане быстродействия участков своих программ. И хотя с годами компиляторы заметно улучшились с точки зрения генерации машинного кода, который стал эффективнее по объему и производительности, все еще остаются ситуации, когда разработчику программного обеспечения имеет смысл использовать преимущества программирования на языке ассемблера.

Архитектура современных процессоров x86 с одним потоком команд и множеством потоков данных (SIMD, single instruction stream – multiple data stream) является еще одним объяснением постоянного интереса к программированию на языке ассемблера. Процессор с поддержкой SIMD обладает вычислительными ресурсами, которые упрощают параллельные вычисления с использованием нескольких значений данных, что может значительно повысить быстродействие приложений, обеспечивающих высокую скорость отклика в реальном времени. Архитектуры SIMD также хорошо подходят для ресурсоемких проблемных областей, таких как обработка изображений, кодирование аудио и видео, автоматизированное проектирование, компьютерная графика и интеллектуальный анализ данных. К сожалению, многие языки высокого уровня и инструменты разработки по-прежнему не могут полностью или хотя бы частично использовать возможности SIMD современного процессора x86. С другой стороны, язык ассемблера открывает разработчику программного обеспечения полный доступ к ресурсам SIMD процессора.

О чем эта книга

Эта книга рассказывает о программировании на 64-битном (x86-64) языке ассемблера x86. Содержание и структура книги призваны помочь вам быстро освоить программирование на языке ассемблера x86-64 и вычислительные ресурсы векторных расширений набора команд Advanced Vector Extensions (AVX)¹. Она также содержит множество примеров исходного кода, которые способствуют ускоренному изучению и пониманию основных конструкций языка ассемблера x86-64 и концепций программирования SIMD. После прочтения этой книги вы сможете кодировать быстродействующие функции и алгоритмы с помощью языка ассемблера x86-64 и наборов инструкций AVX, AVX2 и AVX-512.

Прежде чем продолжить, я должен обратить ваше внимание, что *в этой книге не рассматривается программирование на языке ассемблера x86-32*. В нем также не обсуждаются устаревшие технологии x86, такие как модуль с плавающей запятой x87, MMX и Streaming SIMD Extensions. Если вы заинтересованы в изучении этих тем, прочтите первое издание книги (в переводе не издавалось – прим. перев.). Эта книга не объясняет архитектурные особенности x86 или привилегированные команды процессора, которые используются в операционных системах. Однако вам все равно придется досконально изучить материал, представленный в данной книге, чтобы разрабатывать код на языке ассемблера x86 для использования в операционной системе.

Хотя теоретически возможно написать целую прикладную программу, используя только язык ассемблера, высокие требования к разработке современного программного обеспечения делают такой подход непрактичным и нецелесообразным. Вместо этого в данной книге основное внимание уделяется кодированию функций языка ассемблера x86-64, вызываемых из C++. Каждый пример исходного кода был создан с использованием Microsoft Visual Studio C++ и Microsoft Macro Assembler (MASM).

Целевая аудитория

Целевая аудитория этой книги – разработчики программного обеспечения, в том числе:

- разработчики, которые создают прикладные программы для платформ на базе Windows и хотят научиться писать алгоритмы и функции высокого быстродействия с использованием языка ассемблера x86-64;
- разработчики, которые создают прикладные программы для сред, отличных от Windows, и хотят изучить программирование на языке ассемблера x86-64;

¹ Расширение системы команд x86 для микропроцессоров Intel и AMD, предложенное Intel в марте 2008. – Прим. перев.

- разработчики, которые хотят научиться создавать вычислительные функции SIMD с использованием наборов команд AVX, AVX2 и AVX-512;
- разработчики и студенты, изучающие информатику, которые хотят добиться глубокого понимания платформы x86-64 и ее архитектуры SIMD.

Основная аудитория данной книги – это разработчики программного обеспечения на базе Windows, поскольку примеры исходного кода были разработаны с использованием Visual Studio C++ и MASM. Разработчики, ориентированные на платформы, отличные от Windows, также могут извлечь пользу из этой книги, поскольку большая часть информативного контента излагается независимо от какой-либо конкретной операционной системы. Предполагается, что читатели данной книги уже имеют опыт программирования на языках высокого уровня и базовые знания C++. Знакомство с программированием в Visual Studio или Windows PowerShell не требуется.

Обзор содержания

Основная цель этой книги – помочь вам изучить программирование на 64-битном языке ассемблера x86, а также набор команд AVX, AVX2 и AVX-512. Главы и содержание книги построены таким образом, чтобы достичь этой цели. Ознакомьтесь с кратким обзором того, что вы найдете в этой книге.

В главе 1 рассматривается основная архитектура платформы x86-64. Она включает в себя описание основных типов данных платформы, внутренней архитектуры, наборов регистров, операндов команд и режимов адресации памяти. В этой главе также описывается основной набор команд x86-64. В главах 2 и 3 объясняются основы программирования на языке ассемблера x86-64 с использованием базового набора команд и общих программных конструкций, включая массивы и структуры. Примеры исходного кода, представленные в этих (и последующих) главах, оформлены как рабочие программы, которые в процессе обучения вы можете запускать, изменять или иным образом экспериментировать с кодом.

Глава 4 посвящена архитектурным ресурсам AVX, включая наборы регистров, типы данных и набор команд. В главе 5 объясняется, как использовать набор команд AVX для выполнения скалярных арифметических операций с плавающей запятой, со значениями как с одинарной, так и с двойной точностью. В главах 6 и 7 рассказано про программирование AVX SIMD с использованием упакованных операндов, как целочисленных, так и с плавающей запятой.

Глава 8 знакомит с AVX2 и исследует его расширенные возможности, включая широковегательную передачу, сбор и перестановку данных. Здесь также объясняются операции *слиянного умножения-сложения* (FMA, fused multiply-add). Главы 9 и 10 содержат примеры исходного кода, которые иллюстрируют различные вычислительные алгоритмы, использующие AVX2 с упакованными операндами. Глава 11 включает примеры исходного кода, демонстрирующие программирование FMA. В этой главе также рассмотрены примеры, иллюстрирующие недавние расширения платформы x86 с использованием регистров общего назначения.

В главе 12 детально рассмотрен набор команд AVX-512. В этой главе описаны наборы регистров и типы данных AVX-512. В ней также рассмотрены основные усовершенствования AVX-512, включая условное выполнение и слияние, встроенные широковегательные операции и округление на уровне команд.

В главах 13 и 14 содержится множество примеров исходного кода, демонстрирующих, как использовать эти расширенные функции.

В главе 15 представлен обзор современного многоядерного процессора x86 и лежащей в его основе микроархитектуры. В этой главе также описаны конкретные стратегии и методы программирования, которые можно использовать для повышения быстродействия кода на языке ассемблера x86. В главе 16 приведен обзор нескольких примеров исходного кода, иллюстрирующего передовые методы программирования на языке ассемблера x86, включая обнаружение функций процессора, ускоренный доступ к памяти и многопоточные вычисления.

В приложении А рассказано, как выполнить примеры исходного кода с помощью Visual Studio и MASM. Оно также содержит список ссылок и ресурсов, к которым вы можете обратиться для получения дополнительной информации о программировании на языке ассемблера x86.

Исходный код примеров

Примеры исходного кода этой книги доступны на веб-сайте Apress по адресу <https://www.apress.com/us/book/9781484240625>. Для каждой главы есть ZIP-архив, содержащий файлы исходного кода C++ и ассемблера, а также файлы проекта Visual Studio. Процедура установки не требуется. Вы можете просто извлечь содержимое ZIP-архива главы в папку по вашему выбору.

Внимание! Единственное назначение исходного кода – показать примеры программирования, которые напрямую связаны с темами, обсуждаемыми в этой книге. В этих примерах уделяется минимальное внимание важным проблемам разработки программного обеспечения, таким как надежная обработка ошибок, риски безопасности, вычислительная стабильность, ошибки округления или плохо согласованные функции. Вы принимаете на себя всю ответственность за решение этих проблем, если решите использовать исходный код каких-либо примеров в своих собственных программах.

Примеры исходного кода были созданы с помощью Visual Studio Professional 2017 (версия 15.7.1) на ПК с 64-разрядной Windows 10 Pro. На веб-сайте Visual Studio (<https://visualstudio.microsoft.com>) содержится дополнительная информация об этом и других выпусках Visual Studio. Технические сведения об установке, настройке и разработке приложений Visual Studio доступны по адресу <https://docs.microsoft.com/en-us/visualstudio/?view=vs-2017>.

Рекомендуемой аппаратной платформой для запуска примеров исходного кода является ПК на базе x86 с 64-разрядной ОС Windows 10 и процессором, поддерживающим AVX. Для запуска примеров исходного кода, в которых используются эти наборы команд, требуется процессор, совместимый с AVX2 или AVX-512. Вы можете использовать одну из свободно доступных утилит, перечисленных в приложении, чтобы определить, какие расширения набора команд x86-AVX поддерживает процессор вашего компьютера.