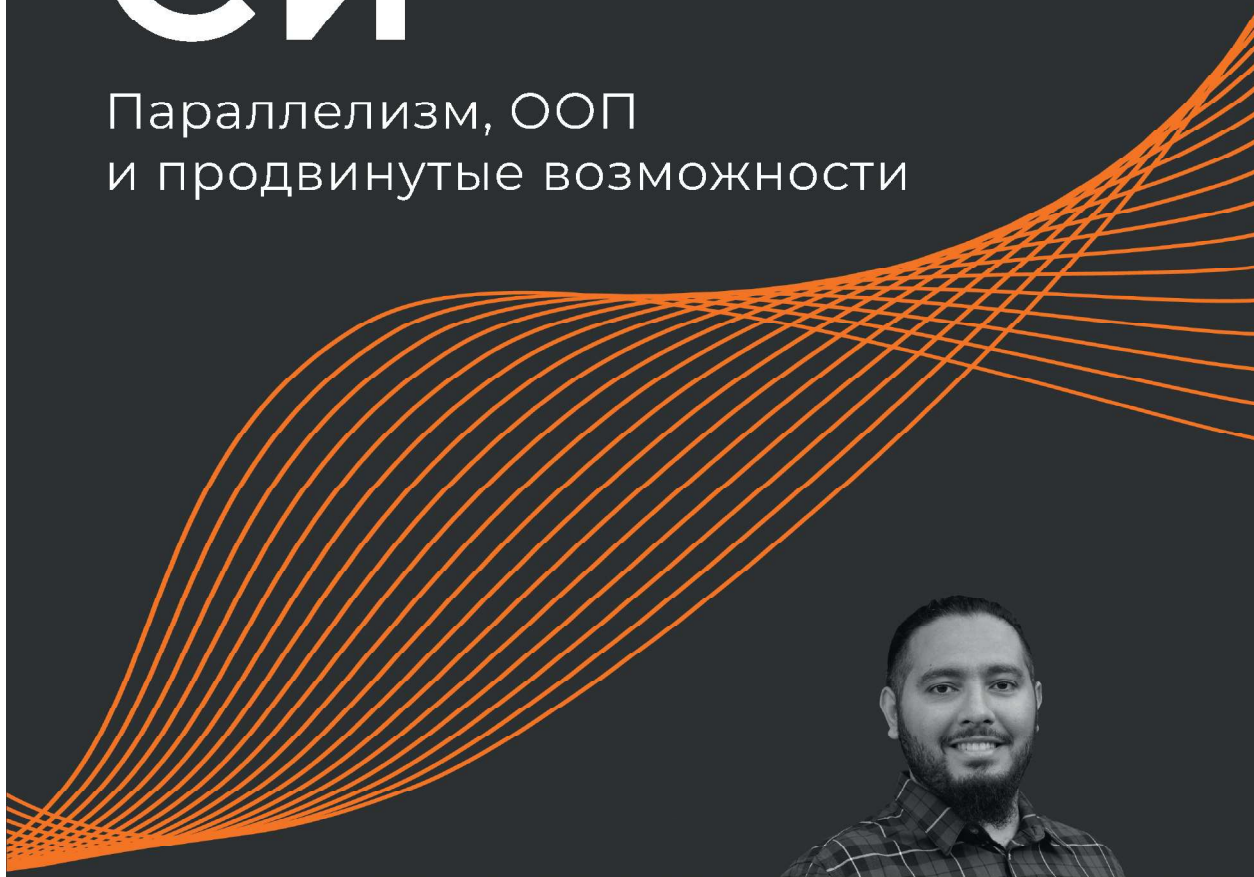




EXPERT INSIGHT

Экстремальный СИ

Параллелизм, ООП
и продвинутые возможности



Камран Амини

Packt>

Extreme C

Taking you to the limit in Concurrency, OOP,
and the most advanced capabilities of C

Kamran Amini

Packt>

BIRMINGHAM - MUMBAI

Камран Амини

Экстремальный СИ

Параллелизм, ООП
и продвинутые возможности



Санкт-Петербург • Москва • Минск

2021

ББК 32.973.2-018.1
УДК 004.43
А62

Амини Камран

А62 Экстремальный Си. Параллелизм, ООП и продвинутые возможности. — СПб.: Питер, 2021. — 752 с.: ил. — (Серия «Для профессионалов»).

ISBN 978-5-4461-1694-2

Для того чтобы овладеть языком Си, знания одного лишь синтаксиса недостаточно. Специалист в области разработки должен обладать четким научным пониманием принципов и методик. Книга «Экстремальный Си» научит вас пользоваться продвинутыми низкоуровневыми возможностями языка для создания эффективных систем, чтобы вы смогли стать экспертом в программировании на Си.

Вы освоите директивы препроцессора, макрокоманды, условную компиляцию, указатели и многое другое. Вы по-новому взглянете на алгоритмы, функции и структуры. Узнаете, как выжимать максимум производительности из приложений с ограниченными ресурсами.

В XXI веке Си остается ключевым языком в машиностроении, авиации, космонавтике и многих других отраслях. Вы узнаете, как язык работает с Unix, как реализовывать принципы объектно-ориентированного программирования, и разберетесь с многопроцессной обработкой.

Камран Амини научит вас думать, сомневаться и экспериментировать. Эта книга просто необходима для всех, кто хочет поднять знания Си на новый уровень.

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

ББК 32.973.2-018.1
УДК 004.43

Права на издание получены по соглашению с Packt Publishing. Все права защищены. Никакая часть данной книги не может быть воспроизведена в какой бы то ни было форме без письменного разрешения владельцев авторских прав.

Информация, содержащаяся в данной книге, получена из источников, рассматриваемых издательством как надежные. Тем не менее, имея в виду возможные человеческие или технические ошибки, издательство не может гарантировать абсолютную точность и полноту приводимых сведений и не несет ответственности за возможные ошибки, связанные с использованием книги. Издательство не несет ответственности за доступность материалов, ссылки на которые вы можете найти в этой книге. На момент подготовки книги к изданию все ссылки на интернет-ресурсы были действующими.

ISBN 978-1789343625 англ.

© Packt Publishing 2019.
First published in the English language under the title 'Extreme C – (9781789343625)'

ISBN 978-5-4461-1694-2

© Перевод на русский язык ООО Издательство «Питер», 2021
© Издание на русском языке, оформление ООО Издательство «Питер», 2021
© Серия «Для профессионалов», 2021
© Павлов А., перевод с английского языка, 2020

Краткое содержание

Об авторе.....	15
О научных редакторах	16
Введение	17
Глава 1. Основные возможности языка.....	25
Глава 2. Компиляция и компоновка	74
Глава 3. Объектные файлы	117
Глава 4. Структура памяти процесса.....	146
Глава 5. Стек и куча.....	171
Глава 6. ООП и инкапсуляция	208
Глава 7. Композиция и агрегация	244
Глава 8. Наследование и полиморфизм.....	260
Глава 9. Абстракция данных и ООП в C++	289
Глава 10. История и архитектура Unix.....	307
Глава 11. Системные вызовы и ядра.....	335
Глава 12. Последние нововведения в C	365
Глава 13. Конкурентность	381
Глава 14. Синхронизация	404
Глава 15. Многопоточное выполнение	446
Глава 16. Синхронизация потоков	469
Глава 17. Процессы	501

Глава 18. Синхронизация процессов	532
Глава 19. Локальные сокеты и IPC	573
Глава 20. Программирование сокетов.....	613
Глава 21. Интеграция с другими языками.....	655
Глава 22. Модульное тестирование и отладка.....	693
Глава 23. Системы сборки	730
Послесловие	751

Оглавление

Об авторе	15
О научных редакторах	16
Введение	17
Для кого эта книга.....	18
Структура издания.....	19
Условия, при соблюдении которых книга будет максимально полезной	21
Скачивание файлов с примерами кода	22
Условные обозначения	22
От издательства.....	24
Глава 1. Основные возможности языка	25
Директивы препроцессора	27
Макросы.....	28
Условная компиляция	41
Указатели на переменные.....	44
Синтаксис	45
Арифметические операции с указателями на переменные	47
Обобщенные указатели.....	50
Размер указателей.....	53
Висячие указатели.....	53
Общая информация о функциях	56
Анатомия функции.....	56
Роль функций в архитектуре приложений	57
Управление стеком.....	57
Передача по значению и передача по ссылке.....	58
Указатели на функции.....	60
Структуры.....	63
Зачем нужны структуры.....	63
Зачем нужны пользовательские типы	64
Принцип работы структур	65

Размещение структур в памяти	66
Вложенные структуры	70
Указатели на структуры	71
Резюме	72
Глава 2. Компиляция и компоновка	74
Процесс компиляции	75
Сборка проекта на языке C	77
Этап 1: предобработка	83
Этап 2: компиляция в ассемблерный код	85
Этап 3: компиляция в машинные инструкции	88
Этап 4: компоновка	90
Препроцессор	93
Компилятор	97
Дерево абстрактного синтаксиса	98
Ассемблер	100
Компоновщик	101
Принцип работы компоновщика	102
Компоновщик можно обмануть!	110
Декорирование имен в C++	114
Резюме	116
Глава 3. Объектные файлы	117
Двоичный интерфейс приложений	118
Форматы объектных файлов	119
Переносимые объектные файлы	121
Исполняемые объектные файлы	125
Статические библиотеки	129
Динамические библиотеки	138
Ручная загрузка разделяемых библиотек	142
Резюме	145
Глава 4. Структура памяти процесса	146
Внутреннее устройство памяти процесса	147
Исследование структуры памяти	148
Исследование статической схемы размещения в памяти	149
Сегмент BSS	151
Сегмент Data	153
Сегмент Text	157
Исследование динамической схемы размещения в памяти	159
Отражение памяти	160
Стек	164
Куча	166
Резюме	169

Глава 5. Стек и куча	171
Стек	172
Исследование содержимого стека.....	173
Рекомендации по использованию стековой памяти.....	179
Куча	183
Выделение и освобождение памяти в куче	185
Принцип работы кучи	193
Управление памятью в средах с ограниченными ресурсами	197
Среды с ограниченной памятью	198
Высокопроизводительные среды	200
Резюме	206
Глава 6. ООП и инкапсуляция	208
Объектно-ориентированное мышление	210
Как мы мыслим.....	211
Диаграммы связей и объектные модели	212
В коде нет никаких объектов	214
Атрибуты объектов.....	216
Предметная область	216
Отношения между объектами	217
Объектно-ориентированные операции	218
Объекты имеют поведение	221
Почему язык С не является объектно-ориентированным	221
Инкапсуляция	222
Инкапсуляция атрибутов.....	223
Инкапсуляция поведения	225
Принцип сокрытия информации	235
Резюме	242
Глава 7. Композиция и агрегация	244
Отношения между классами	244
Объекты и классы.....	245
Композиция.....	247
Агрегация	253
Резюме	259
Глава 8. Наследование и полиморфизм	260
Наследование.....	260
Природа наследования.....	261
Полиморфизм	277
Что такое полиморфизм	277
Зачем нужен полиморфизм.....	280
Полиморфное поведение в языке С	280
Резюме	288

Глава 9. Абстракция данных и ООП в C++	289
Абстракция данных.....	289
Объектно-ориентированные концепции в C++.....	293
Инкапсуляция.....	293
Наследование	296
Полиморфизм	302
Абстрактные классы.....	305
Резюме.....	306
Глава 10. История и архитектура Unix.....	307
История Unix	308
Multics OS и Unix	308
BCPL и B.....	309
Путь к C.....	310
Архитектура Unix.....	312
Философия.....	312
Многослойная структура Unix.....	314
Интерфейс командной оболочки для пользовательских приложений	317
Интерфейс ядра для кольца командной оболочки	322
Ядро.....	327
Аппаратное обеспечение.....	332
Резюме.....	334
Глава 11. Системные вызовы и ядра.....	335
Системные вызовы.....	335
Тщательное исследование системных вызовов.....	336
Выполнение системного вызова напрямую, в обход стандартной библиотеки C.....	337
Внутри функции syscall.....	340
Добавление системного вызова в Linux	342
Ядра Unix	355
Монолитные ядра и микроядра	356
Linux	357
Модули ядра	358
Резюме.....	364
Глава 12. Последние нововведения в C	365
C11	366
Определение поддерживаемой версии стандарта C.....	366
Удаление функции gets	368
Изменения в функции fopen.....	368
Функции с проверкой диапазона.....	370

Невозвращаемые функции.....	371
Макрос для обобщенных типов.....	372
Unicode.....	372
Анонимные структуры и анонимные объединения.....	378
Многопоточность	380
Немного о C18	380
Резюме.....	380
Глава 13. Конкурентность	381
Введение в конкурентность	381
Параллелизм	383
Конкурентность	384
Планировщик заданий.....	385
Процессы и потоки.....	387
Порядок выполнения инструкций	388
Когда следует использовать конкурентность.....	390
Разделяемые состояния.....	397
Резюме.....	402
Глава 14. Синхронизация	404
Проблемы с конкурентностью.....	404
Естественные проблемы с конкурентностью	406
Постсинхронизационные проблемы.....	416
Методы синхронизации	417
Холостые циклы и циклические блокировки	418
Механизм ожидания/уведомления	421
Семафоры и мьютексы.....	424
Системы с несколькими вычислительными блоками.....	429
Циклические блокировки.....	434
Условные переменные	436
Конкурентность в POSIX.....	438
Ядра с поддержкой конкурентности	438
Многопроцессность	440
Многопоточность	443
Резюме.....	444
Глава 15. Многопоточное выполнение	446
Потоки.....	447
POSIX-потоки	450
Порождение POSIX-потоков	452
Пример состояния гонки	457
Пример гонки данных.....	465
Резюме.....	468

Глава 16. Синхронизация потоков	469
Управление конкурентностью в POSIX	470
POSIX-мьютексы.....	470
Условные переменные POSIX	473
POSIX-барьеры	477
POSIX-семафоры.....	480
POSIX-потоки и память.....	488
Сегмент стека	488
Сегмент кучи	493
Видимость памяти	498
Резюме.....	500
Глава 17. Процессы	501
API для выполнения процессов	501
Создание процесса.....	504
Выполнение процесса.....	509
Разные методы создания и выполнения процессов.....	512
Этапы выполнения процесса.....	512
Разделяемые состояния.....	513
Методы разделения ресурсов	514
Разделяемая память в POSIX.....	516
Файловая система.....	526
Сравнение многопоточности и многопроцессности	528
Многопоточность.....	528
Локальная многопроцессность	529
Распределенная многопроцессность.....	530
Резюме.....	531
Глава 18. Синхронизация процессов	532
Локальное управление конкурентностью.....	533
Именованные POSIX-семафоры	534
Именованные мьютексы	538
Первый пример.....	538
Второй пример.....	542
Именованные условные переменные	552
Этап 1: класс разделяемой памяти.....	553
Этап 2: класс разделяемого 32-битного целочисленного счетчика.....	556
Этап 3: класс разделяемого мьютекса.....	558
Этап 4: класс разделяемой условной переменной.....	562
Этап 5: основная логика.....	565
Распределенное управление конкурентностью	570
Резюме.....	572

Глава 19. Локальные сокеты и IPC	573
Методы межпроцессного взаимодействия	574
Коммуникационные протоколы.....	576
Характеристики протоколов.....	578
Взаимодействие в рамках одного компьютера.....	581
Файловые дескрипторы.....	581
POSIX-сигналы	582
POSIX-каналы	586
Очереди сообщений POSIX.....	588
Сокеты домена Unix.....	591
Введение в программирование сокетов	592
Компьютерные сети	592
Что такое программирование сокетов	605
У сокетов есть собственные дескрипторы!	611
Резюме.....	612
Глава 20. Программирование сокетов.....	613
Краткий обзор программирования сокетов	614
Проект «Калькулятор»	616
Иерархия исходного кода	617
Сборка проекта	620
Запуск проекта.....	621
Прикладной протокол	622
Библиотека сериализации/десериализации	625
Сервис калькулятора.....	630
Сокеты домена Unix	632
Потоковый сервер на основе UDS.....	632
Потоковый клиент на основе UDS.....	640
Датаграммный сервер на основе UDS.....	643
Датаграммный клиент на основе UDS	647
Сетевые сокеты	649
TCP-сервер.....	650
TCP-клиент.....	651
UDP-сервер.....	652
UDP-клиент.....	653
Резюме.....	654
Глава 21. Интеграция с другими языками.....	655
Что делает интеграцию возможной	656
Получение необходимых материалов	657
Библиотека для работы со стеком.....	658
Интеграция с C++	664

Декорирование имен в C++	665
Код на C++	667
Интеграция с Java.....	672
Написание кода на Java.....	672
Написание машинно-зависимой части	677
Интеграция с Python.....	685
Интеграция с Go	689
Резюме	691
Глава 22. Модульное тестирование и отладка.....	693
Тестирование программного обеспечения	694
Уровни тестирования	695
Модульное тестирование.....	696
Тестовые дублеры.....	704
Компонентное тестирование	706
Библиотеки тестирования для C	707
CMocka	708
Google Test.....	717
Отладка	721
Категории программных ошибок	722
Отладчики	723
Средства проверки памяти	725
Средства отладки потоков	726
Профилировщики производительности.....	727
Резюме	728
Глава 23. Системы сборки.....	730
Что такое система сборки	731
Make	732
CMake — не система сборки!.....	740
Ninja.....	744
Bazel.....	746
Сравнение систем сборки	749
Резюме	749
Послесловие	751

Об авторе

Камран Амини специализируется на ядре и встроенных системах. Он работал инженером, архитектором, консультантом и техническим директором во множестве известных иранских компаний. В 2017 году переехал в Европу, где трудился старшим архитектором и инженером в таких солидных компаниях, как Jeppesen, Adecco, TomTom и ActiveVideo Networks. Во время своего пребывания в Амстердаме Камран и написал эту книгу. Его больше всего интересуют следующие темы: теория алгоритмов, распределенные системы, машинное обучение, теория информации и квантовые вычисления.

Хочу поблагодарить маму Эхтирам, которая посвятила жизнь воспитанию меня и моего брата Ашкана. Она всегда нас поддерживает.

Хочу также поблагодарить мою прекрасную и любимую жену Афсанех, которая поддерживала меня на каждом этапе, особенно во время работы над этой книгой. Без ее терпения и поддержки я бы никогда не справился.

О научных редакторах

Алиакбар Аббаси — разработчик ПО с более чем восьмилетним опытом использования различных технологий и языков программирования. Специалист в ООП, C/C++ и Python. Любит читать техническую литературу и расширять свой кругозор в области программирования. В настоящее время проживает в Амстердаме и работает старшим программистом в компании TomTom.

Рохит Талвалкар — очень опытный специалист в программировании на языках C, C++ и Java. На его счету разработка приложений, драйверов и сервисов для проприетарной версии *RTOS (Real Time OS)*, Windows, устройств на основе Windows Mobile и платформы Android.

Рохит получил диплом бакалавра технических наук в индийском Технологическом институте города Мумбаи, а также диплом магистра CS. В настоящее время занимает должность ведущего разработчика приложений в сфере смешанной реальности. Рошит успел поработать в Motorola и BlackBerry и сейчас является сотрудником компании Magic Leap, которая выпускает очки смешанной реальности и специализируется на пространственных вычислениях. В свое время редактировал книгу *C++ for the Impatient* Брайана Оверленда.

Хочу поблагодарить доктора Кловиса Тондо, который научил меня C, C++, Java и многому другому.

Введение

В современном мире мы регулярно имеем дело с умопомрачительными технологиями, которые невозможно было представить еще несколько десятилетий назад. На улицах начинают появляться беспилотные автомобили. Достижения в физике и других науках меняют наши представления о реальности как таковой. Мы читаем новости о том, как исследователи делают первые шаги в квантовых вычислениях, о блокчейне и криптовалютах, о планах колонизации других планет. Невероятно, но в основе таких разнообразных по своей природе достижений лежат всего несколько технологий, одной из которых посвящена эта книга. Речь идет о языке C.

Я начал программировать на C++ в девятом классе, присоединившись к команде юных разработчиков, занимавшейся созданием 2D-симулятора игры в футбол. Вскоре после C++ я начал изучать Linux и C. Стоит признать, что в те времена важность C и Unix не была для меня столь очевидной, но, постепенно получая опыт использования этих технологий в разных проектах и все больше узнавая о них, я осознал, насколько важную роль они играют. Чем ближе я знакомился с C, тем сильнее уважал этот язык программирования. В конце концов я решил стать профессиональным программистом на C. Мне хотелось делиться своими знаниями с другими людьми, чтобы они тоже понимали всю важность этой технологии. Данная книга стала результатом моих амбиций.

Бытует заблуждение, будто C — мертвый язык, и в целом многие технические специалисты имеют о нем туманное представление. Чтобы убедиться в обратном, достаточно взглянуть на рейтинг TIOBE по адресу www.tiobe.com/tiobe-index. На самом деле C наряду с Java — один из самых известных языков за прошедшие 15 лет, и в последние годы он только набирал популярность.

Я подошел к написанию этой книги, имея многолетний опыт в разработке и проектировании с помощью C, C++, Golang, Java и Python на разных платформах, включая различные версии BSD Unix, Linux и Microsoft Windows. Моей основной целью было вывести навыки читателей на новый уровень, поделиться с ними опытом, полученным тяжелым трудом. Легкой прогулки ждать не стоит, именно поэтому книга называется «Экстремальный Си. Параллелизм, ООП и продвинутые возможности». Мы не станем отвлекаться и сравнивать C с другими языками программирования.

Я попытался сделать текст максимально практичным, однако он все равно содержит большое количество фундаментального теоретического материала, имеющего реальное применение. Множество примеров, представленных здесь, помогут вам подготовиться к тому, с чем вам предстоит столкнуться в реальных проектах.

Возможность взяться за столь весомую тему — большая честь для меня. Это сложно выразить словами, и потому скажу лишь, что мне было очень приятно писать о том, что так близко моему сердцу. Это моя первая книга, и возможностью быть ее автором я обязан Эндрю Валдрону.

Заодно хочу передать привет и поблагодарить Йена Хью, редактора-консультанта, с которым мы бок о бок трудились над каждой главой, Алиакбара Аббаси за его замечания и предложения, а также Кишора Рита, Гаурава Гаваса, Веронику Пэйс и многих других людей, внесших ценный вклад в подготовку и издание этой книги.

Я предлагаю вам стать моими спутниками в этом длинном путешествии. Надеюсь, чтение книги изменит ваш кругозор, поможет вам увидеть C в новом свете и заодно сделает вас отличным программистом.

Для кого эта книга

Книга предназначена для читателей, уже имеющих минимальный уровень знаний в области разработки на C и C++. Основная аудитория — начинающие и middle-программисты на C/C++; именно они смогут извлечь максимальную пользу из прочитанного материала, применяя полученные навыки и знания. Надеюсь, книга поможет им ускорить карьерный рост и стать старшими разработчиками. Кроме того, прочитав ее, они смогут претендовать на большое количество вакансий с высокими требованиями и, как правило, хорошей зарплатой. Те или иные темы могут пригодиться и опытным программистам на C/C++, хотя я ожидаю, что эти люди в целом знакомы с изложенным здесь материалом и могут почерпнуть для себя лишь некоторые полезные детали.

Еще одна категория читателей, которым может пригодиться эта книга, — студенты и научные сотрудники. Возможно, вы получаете высшее образование или учитесь в аспирантуре в любой научной или технической области, такой как информатика, разработка ПО, искусственный интеллект, *Интернет вещей* (Internet of Things, IoT), астрономия, физика частиц и космология, либо занимаетесь исследованиями в этих сферах. Книга позволит повысить уровень ваших знаний о C/C++ и Unix-подобных операционных системах, а также поможет отточить соответствующие навыки программирования. Представленный материал пригодится инженерам и ученым, работающим над сложными, многопоточными или даже многопроцессными системами для удаленного управления устройствами, моделирования, обработки больших объемов данных, машинного/глубокого обучения и т. д.

Структура издания

Книга состоит из семи условных частей, каждая из которых посвящена определенным аспектам программирования на С. В первой части рассматривается создание проекта на С, во второй обсуждается память, в третьей — объектная ориентированность, а в четвертой речь в основном идет о системах Unix и их связях с языком С. В пятой части мы поговорим о конкурентности, в шестой — о межпроцессном взаимодействии, а в седьмой, заключительной части речь пойдет о тестировании и сопровождении кода. Ниже приводится краткое описание каждой из 23 глав книги.

Глава 1 «Основные возможности языка» посвящена основным возможностям С, которые определяют то, как мы используем данный язык. В число главных тем входят определение макросов с помощью препроцессора и директив, указатели на переменные и функции, механизмы вызова функций, а также структуры.

Глава 2 «Компиляция и компоновка» содержит описание сборки проектов на С. Во всех подробностях будут рассмотрены как процесс компиляции в целом, так и отдельные его элементы.

Глава 3 «Объектные файлы» посвящена результатам компиляции проекта на С. Вы познакомитесь с объектными файлами, заглянете внутрь этих файлов и посмотрите, какую информацию из них можно извлечь.

Глава 4 «Структура памяти процесса» исследует внутреннее устройство памяти процесса. Вы узнаете, из каких сегментов состоит память и чем статическая память отличается от динамической.

Глава 5 «Стек и куча» содержит информацию о таких сегментах, как стек и куча. Мы поговорим о переменных, которые в них хранятся, и об управлении их жизненным циклом в С. Вы научитесь передовым практикам работы с кучей.

Глава 6 «ООП и инкапсуляция» — первая из четырех глав, относящихся к объектной ориентированности в С. Мы пройдемся по теории, стоящей за ООП, и будет дано определение важным терминам, которые часто встречаются в технической литературе.

Глава 7 «Композиция и агрегация» посвящена композиции и ее специальной разновидности — агрегации. Мы обсудим их отличия, которые проиллюстрируем с помощью примеров.

Глава 8 «Наследование и полиморфизм» исследует один из самых важных аспектов *объектно-ориентированного программирования (ООП)* — наследование. Я покажу, как происходит наследование между двумя классами и как это можно реализовать в С. Вдобавок мы поговорим еще об одной обширной теме — полиморфизме.

Глава 9 «Абстракция данных и ООП в С++» является заключительной для третьей части книги и отведена теме абстракции. Вы познакомитесь с абстрактными типами

данных и узнаете, как они реализованы в С. Мы обсудим С++ и рассмотрим объектно-ориентированные концепции на примере этого языка.

При обсуждении языка С нельзя не упомянуть о Unix. Глава 10 «История и архитектура Unix» объясняет, почему эти две технологии так тесно связаны между собой и как данный симбиоз способствует их живучести. Мы также рассмотрим архитектуру операционной системы Unix и узнаем, как программы используют предоставляемые ею возможности.

Глава 11 «Системные вызовы и ядра» посвящена пространству ядра в архитектуре Unix. Мы подробно обсудим системные вызовы и рассмотрим, как их можно создавать в Linux. Вдобавок поговорим о разных видах ядер и увидим принцип работы модулей ядра Linux на примере простого модуля.

Глава 12 «Последние нововведения в С» представляет последнюю версию стандарта С под названием C18. Вы увидите, чем она отличается от предыдущей версии, C11. Я также продемонстрирую ряд новых возможностей, которые появились с момента выхода C99.

Глава 13 «Конкурентность» открывает пятую часть и посвящена конкурентности. Мы в основном обсудим среды конкурентного выполнения и их различные свойства, такие как чередование. Я объясню, почему эти системы недетерминистические и каким образом данная особенность может вызывать проблемы конкурентности, такие как состояние гонки.

Глава 14 «Синхронизация» побуждает продолжить наше обсуждение сред конкурентного выполнения и обращает внимание на разные проблемы, которые в них встречаются, включая состояние гонки, конкуренцию за данные и взаимную блокировку. Вы познакомитесь с методиками, позволяющими преодолеть эти проблемы, такими как семафоры, мьютексы и условные переменные.

Глава 15 «Многопоточное выполнение» демонстрирует одновременное выполнение нескольких потоков и способы управления ими. Я также приведу реальные примеры с проблемами конкурентности в С, перечисленные в предыдущей главе.

Глава 16 «Синхронизация потоков» описывает методы синхронизации нескольких потоков. Среди представленных здесь тем можно выделить семафоры, мьютексы и условные переменные.

Глава 17 «Процессы» представляет способы создания или порождения новых процессов. Мы также обсудим пассивные и активные методики разделения состояния между разными процессами и рассмотрим проблемы с конкурентностью, затронутые в главе 14, используя реальные примеры на языке С.

Глава 18 «Синхронизация процессов» в основном посвящена имеющимся механизмам для синхронизации разных процессов, находящихся на одном и том же компьютере, включая межпроцессные семафоры, мьютексы и условные переменные.

Глава 19 «Локальные сокеты и IPC» основное внимание уделяет пассивным методам *межпроцессного взаимодействия* (interprocess communication, IPC). Основной акцент делается на взаимодействии процессов, находящихся на одном компьютере. Вы также познакомитесь с программированием сокетов и научитесь создавать каналы между процессами, принадлежащими разным сетевым узлам.

Глава 20 «Программирование сокетов» представляет сетевое программирование и содержит примеры кода, которые проиллюстрируют разные типы сокетов, включая сокеты домена Unix, а также TCP- и UDP-сокеты, основанные на поточных и датаграммных каналах.

Глава 21 «Интеграция с другими языками» демонстрирует, как библиотеку на C, собранную в виде динамического объектного файла, можно загружать и использовать в программах, написанных на C++, Java, Python и Golang.

Глава 22 «Модульное тестирование и отладка» посвящена тестам разных видов, но мы сосредоточимся на модульном тестировании в C. Вы познакомитесь с библиотеками CMocka и Google Test, предназначенными для написания наборов тестов в C. Применительно к отладке мы пройдемся по различным инструментам, которые позволяют отлаживать разного рода программные ошибки.

Глава 23 «Системы сборки», заключительная, представляет системы сборки, такие как Make, Ninja и Bazel, и один генератор сборочных скриптов, CMake.

Условия, при соблюдении которых книга будет максимально полезной

Как уже объяснялось, от читателя требуется определенный уровень знаний и навыков в области программирования.

- Общее понимание архитектуры компьютера. Вы должны иметь представление о памяти, центральном процессоре, периферийных устройствах и их характеристиках и понимать, как компьютерные программы взаимодействуют с этими элементами.
- Владение основами программирования. Вы должны знать, что такое алгоритм, как проследить за его выполнением, что собой представляет исходный код и как в двоичной системе выполняются арифметические операции.
- Навыки работы с *терминалом* и основными *утилитами командной строки* в Unix-подобных операционных системах, таких как Linux или macOS.
- Понимание таких тем, как условные выражения, разные виды циклов, структуры или классы минимум в одном языке программирования, указатели в C или C++, функции и т. д.

- Знание основ ООП. Требование необязательное, поскольку ООП подробно рассматривается в книге, однако знание этой темы поможет вам лучше понять материал, изложенный в главах, посвященных *объектной ориентированности*.



Кроме того, настоятельно рекомендуется скачать репозиторий с кодом и самостоятельно выполнять команды, которые приводятся в листингах командной оболочки. Пожалуйста, используйте компьютер под управлением Linux, macOS или другой операционной системы, совместимой с POSIX.

Скачивание файлов с примерами кода

Вы можете скачать файлы с кодом, выполнив следующие шаги.

1. Войдите или зарегистрируйтесь на сайте <http://www.packt.com>.
2. Выберите вкладку **Support** (Поддержка).
3. Щелкните на ссылке **Code Downloads** (Загрузка кода).
4. Введите английское название книги (Extreme C) в поле **Search** (Искать) и следуйте инструкциям, которые появятся на экране.

Скачав файл, не забудьте его распаковать, используя последнюю версию одного из следующих инструментов:

- WinRAR/7-Zip для Windows;
- Zipeg/iZip/UnRarX для Mac;
- 7-Zip/PeaZip для Linux.

Архив с кодом для этой книги также доступен на GitHub по адресу github.com/PacktPublishing/Extreme-C. Все обновления кода вносятся в существующий GitHub-репозиторий.

У нас есть богатая библиотека книг и видеороликов. Примеры кода для них можно найти на сайте github.com/PacktPublishing/.

Условные обозначения

В книге используются листинги кода и командной оболочки. Первые содержат либо код на языке C, либо псевдокод. Пример листинга кода показан ниже (листинг 17.1).

Листинг 17.1. Создание дочернего процесса с помощью API fork (ExtremeC_examples_chapter17_1.c)

```
#include <stdio.h>
#include <unistd.h>
```

```

int main(int argc, char** argv) {
    printf("This is the parent process with process ID: %d\n",
        getpid());
    printf("Before calling fork() ...\n");
    pid_t ret = fork();
    if (ret) {
        printf("The child process is spawned with PID: %d\n", ret);
    } else {
        printf("This is the child process with PID: %d\n", getpid());
    }
    printf("Type CTRL+C to exit ...\n");
    while (1);
    return 0;
}

```

Как видите, приведенный выше код можно найти в файле `ExtremeC_examples_chapter17_1.c`, который находится в архиве с примерами для этой книги, в каталоге главы 17. Архив с кодом доступен по ссылке github.com/PacktPublishing/Extreme-C.

Если листинг не связан ни с каким файлом, он содержит псевдокод или код на языке C, который не вошел в архив. Вот пример.

Листинг 13.1. Простое задание с пятью инструкциями

```

Task P {
    1. num = 5
    2. num++
    3. num = num - 2
    4. x = 10
    5. num = num + x
}

```

Иногда некоторые строки в листингах выделены жирным шрифтом. Они обычно обсуждаются перед листингом или после него, а выделяются для того, чтобы вам было легче их найти.

Листинги командной оболочки используются для иллюстрации вывода консольных команд, запускаемых в терминале. Сами команды, как правило, напечатаны жирным шрифтом, а их вывод — обычным. Вот пример.

Терминал 17.6. Чтение из объекта разделяемой памяти, созданного в примере 17.4, и его удаление

```

$ ls /dev/shm
shm0
$ gcc ExtremeC_examples_chapter17_5.c -lrt -o ex17_5.out
$ ./ex17_5.out
Shared memory is opened with fd: 3
The contents of the shared memory object: ABC

$ ls /dev/shm
$

```

Команды начинаются либо с \$, либо с #. В первом случае команду следует выполнять от имени обычного пользователя, а во втором — от имени администратора.

Консольные команды обычно выполняют в каталоге соответствующей главы, который находится в архиве с кодом. Если нужно перейти в определенный рабочий каталог, то я вам об этом сообщу.

Курсивом выделены новые термины или слова, на которых нужно акцентировать внимание. Шрифт без засечек используется для ссылок и названий каталогов, а также для элементов интерфейса. Например: «Выберите раздел **System info** (Системная информация) на панели **Administration** (Администрирование)». Названия файлов оформляются моноширинным шрифтом.



Предупреждения и важные замечания оформлены так.



Советы и приемы оформлены таким образом.

От издательства

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

Мы будем рады узнать ваше мнение!

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