Эта книга посвящается
моей любимой маме
Аише Хармуш.

Моей опоре и поддержке

Оглавление

Об авторе 20
О техническом рецензенте 21
Признательности 22
Предисловие 23
Скачивание архивов с исходным кодом и цветными изображениями 24
От издательства 25

Глава 1. Системное программирование на языке Rust 26

Введение 26
Содержание главы 26
Защищенность и производительность 27
Защита памяти 28
Обращение к памяти по пустому указателю 29
Переполнение буфера 34
Сборщик мусора 36
Многопоточность и параллелизм 39
Сопоставление с шаблоном 44
Времена жизни 45
Абстракции с нулевой стоимостью 46
Интерфейс с внешними функциями (FFI) 48
Обработка ошибок 49
Контролируемые незащищенные операции 50
Инструментарий Rust 50
Практические применения 51
Будущее вместе с Rust 51
Отзывчивое сообщество 52
За пределами основ 52
Заметные проекты на Rust 52
Установка Rust 54
Установка языка в Windows 54
Установка в Linux 56
Установка Rust в MacOS 57
Все готово к работе! 57
Интегрированная среда разработки и инструменты 57
Выбор интегрированной среды разработки 58
Visual Studio Code (VS Code) 58
Основные инструменты Rust 59
Менеджер пакетов 59
Анализатор исходного кода 59
Форматирование исходного кода 59

Написание первой программы на Rust ... 60
Начало работы ... 60
Написание исходного кода ... 60
Компиляция и запуск ... 60
Cargo: менеджер пакетов исходного кода на Rust ... 62
Знакомство с менеджером Cargo ... 62
Создание нового проекта ... 62
Навигация по структуре проекта ... 62
Сборка и запуск ... 63
Управление зависимостями ... 63
Тестирование и документация ... 64
Обследование других функциональных возможностей менеджера Cargo ... 64
Заключение ... 64
Дополнительные ресурсы ... 65
Вопросы для самопроверки ... 65
Ответы ... 68
Ключевые термины ... 68

Глава 2. Основы Rust ... 71

Введение ... 71
Содержание главы ... 71
Переменные и типы данных ... 72
Введение в переменные ... 72
Мутируемость ... 74
Затенение ... 74
Константы ... 76
Числовые примитивы ... 77
Фиксированная ширина ... 78
Целочисленные типы ... 78
Символы и числовые типы ... 79
Строки и символы ... 81
Итерация по символам и операции со строковыми значениями ... 81
Использование числовых конвертаций ... 82
Арифметические операции ... 83
Сложение ... 84
Вычитание ... 84
Умножение ... 84
Деление ... 85
Взятие остатка от деления ... 85
Числа с плавающей точкой ... 86
f32 для эффективности ... 86
f64 для глубины точности ... 87
Литералы чисел с плавающей точкой ... 88
Математические операции ... 88
Ориентация в сложностях ... 89

Приближенное равенство 89
Освоение чисел с плавающей точкой 90
Булевы значения 90
Сложные типы данных 94
Строковые значения 94
Подстроки 95
Комбинирование строковых значений 95
Массивы 96
Преимущества фиксированного размера 97
Синтаксис и объявление 97
Надежность в сложных условиях 97
Кортежи 98
Деструктуризация кортежей 98
Управляющие конструкции 99
Условные инструкции 100
Циклы while 100
Циклы for 100
Конструкция loop 101
Выражения match 101
Функции 102
Явные сигнатуры 102
Вложенные функции для модульности 102
Замыкания 103
Владение, заимствование и времена жизни 103
Владение 103
Контролер заимствования 104
Повышение производительности 105
Заключение 105
Вопросы для самопроверки 105
Ответы 108
Ключевые термины 109

Глава 3. Общие черты и обобщения 110

Введение 110
Содержание главы 110
Общие черты 111
Реализация общих черт 111
Общая черта с дефолтным поведением 113
Границы общих черт 114
Расширение с помощью обобщений 115
Обобщения 115
Адаптивные структуры 116
Ассоциированные функции 117
Ассоциированные типы 118
Объекты динамической диспетчеризации 120
Практические применения 121
Границы общих черт в стандартной библиотеке 123

Продвинутые шаблоны общих черт ... 124
Ассоциированные константы ... 124
Перегрузка операторов ... 124
Маркерные общие черты ... 125
Комбинирование общих черт ... 126
Избегание конфликтов общих черт ... 127
Всеохватывающие реализации ... 128
Суперчерты ... 129
Шаблон-обертка ... 130
Типы с динамически изменяемым размером ... 132
Условное соблюдение ... 134
Программирование на уровне типов ... 135
Соображения производительности ... 137
Заключение ... 137
Вопросы для самопроверки ... 138
Ответы ... 141

Глава 4. Встроенные в Rust структуры данных ... 142

Введение ... 142
Содержание главы ... 143
Массивы в Rust ... 143
Создание массивов ... 144
Доступ к элементам массива ... 144
Изменение элементов массива ... 146
Выполнение итераций по элементам массива в цикле ... 147
Взятие срезов из массивов ... 147
Многомерные массивы ... 149
Работа с методами массивов ... 150
Инициализация массивов и дефолтные значения ... 151
Инициализация дефолтными значениями ... 151
Генерирование шаблонов с помощью итераторов ... 151
Инициализация вычисляемыми значениями ... 151
Длина массива и проверка границ ... 152
Получение длины ... 152
Проверка границ ... 153
Копирование и клонирование массивов ... 153
Копирование массивов ... 153
Клонирование массивов ... 153
Векторы ... 154
Создание векторов ... 155
Доступ к элементам и их изменение ... 155
Изменение векторов ... 156
Добавление элементов ... 156
Обновление элементов ... 157
Удаление элементов ... 158
Выполнение итераций по элементам в цикле ... 159
Использование цикла for ... 159

Использование метода iter_mut для мутируемой итерации 160
Использование метода enumerate для получения индекса и значения 161
Кортежи 161
Создание и инициализация кортежей 162
Доступ к элементам кортежа 162
Деструктуризация кортежей 163
Кортежные шаблоны и продвинутое использование 163
Варианты использования из реальной практики 164
Координатные вычисления 164
Обработка ошибок с помощью кортежей Result 164
Ограничения кортежей 165
Кортежи в операции сопоставления с шаблоном 165
Сопоставление кортежа с шаблонами 165
Игнорирование элементов кортежа 166
Вложенные кортежи и шаблоны 166
Опровержимость кортежных шаблонов 167
Владение и заимствование в кортежах 167
Владение кортежами 168
Заимствование кортежей 168
Взятие срезов кортежей и оператор развертывания 169
Создание срезов элементов кортежей 169
Доступ к срезам кортежей 170
Кортежи как аргументы функций и возвращаемые из них значения 170
Использование кортежей в качестве аргументов функций 170
Возврат кортежей из функций 171
Кортежные варианты перечислений 172
Создание вариантов перечислений с использованием кортежей 172
Срезы 173
Понимание срезов 174
Создание срезов 175
Доступ к элементам среза 175
Изменение элементов среза 176
Практические применения 177
Манипуляция строковыми литералами 177
Обработка данных 178
Токенизация текста 179
Работа с двоичными данными 179
Отображение в память 180
Хеш-множества в Rust 181
Создание множества 181
Обновление множества 182
Добавление элементов 182
Удаление элементов 182
Продвинутые операции на множествах 182
Симметрическая разность 183
Проверка на подмножество и надмножество 183

Практические применения ... 184
Дедупликация ... 184
Проверка на вхождение в множество ... 185
Кеширование ... 185
Алгоритмы на графах ... 186
Операции на множествах ... 188
Тегирование и фильтрация ... 189
Отслеживание уникальных значений ... 190
Вопросы производительности ... 190
Вопросы конкурентного исполнения ... 191
Умные указатели Mutex ... 191
Атомарные типы ... 192
Сериализация и десериализация ... 194
Нагрузочное тестирование и оптимизация ... 195
Хеш-словари ... 196
Создание хеш-словаря ... 196
Обновление хеш-словаря ... 197
Добавление элементов ... 197
Удаление элементов ... 198
Обновление элементов ... 198
Доступ к значениям ... 199
Выполнение итераций по элементам хеш-словаря в цикле ... 199
Продвинутые операции на хеш-словарях ... 201
Проверка существования ключа ... 201
API внесения записей ... 201
Очистка хеш-словаря ... 202
Емкость хеш-словаря ... 202
Практические применения ... 203
Подсчет вхождений ... 203
Мемоизация ... 204
Кеширование ... 204
Управление конфигурацией ... 205
Преобразование данных ... 206
Группировка данных ... 206
Графовые алгоритмы ... 207
Индексация баз данных ... 207
Заключение ... 208
Дополнительные ресурсы ... 209
Вопросы для самопроверки ... 211
Ответы ... 215

Глава 5. Обработка ошибок и восстановление после отказов ... 217
Введение ... 217
Содержание главы ... 218
Обработка ошибок с использованием типов Result и Option ... 218
Понимание типов Result и Option ... 218
Обработка ошибок с помощью типа Result ... 219
Обработка ошибок с помощью типа Option ... 220

Передача ошибок по стеку вызовов ... 221
Оператор ? ... 221
Обработка нескольких ошибок с помощью типа Result ... 223
Передача ошибок с использованием типов Result и Option ... 224
Создание конкретно-прикладных типов ошибок ... 225
Продвинутая обработка ошибок ... 226
Библиотека anyhow ... 226
Конкретно-прикладные типы ошибок с пакетом thiserror ... 227
Асинхронная обработка ошибок ... 229
Асинхронные функции и экземпляры типа Result ... 229
Конкретно-прикладные типы ошибок для асинхронного исходного кода ... 230
Обработка ошибок конкурентности ... 231
Обработка ошибок в веб-приложениях ... 232
Собираем все воедино ... 233
Разбор файлов и обработка ошибок ... 233
Обработка ошибок в консольных приложениях ... 234
Использование библиотеки clap ... 235
Обработка сигналов и прерываний ... 236
Обработка ошибок при файловом вводе-выводе ... 237
Чтение и запись файлов ... 238
Работа с директориями ... 239
Обработка ошибок в сетевом программировании ... 239
Выполнение HTTP-запросов ... 240
Построение сетевых сервисов ... 240
Обработка ошибок в многопоточном исходном коде ... 242
Межпотоковый обмен данными ... 244
Тестирование и обработка ошибок ... 245
Написание тестов для обработки ошибок ... 245
Тестирование на основе свойств ... 246
Заключение ... 247
Справочные и дополнительные материалы ... 248
Вопросы для самопроверки ... 248
Ответы ... 252

Глава 6. Управление памятью и указатели ... 253

Введение ... 253
Содержание главы ... 254
Роль управления памятью ... 254
Трудности ручного управления памятью ... 255
Владение и заимствование ... 255
Висячие ссылки ... 257
Незащищенный исходный код ... 258
Гонки за доступом к данным ... 259
Управление ресурсами ... 260
Стек и куча ... 260
Роль стека в Rust ... 262

Роль кучи в Rust ... 262
Модель владения ... 263
Заимствование в Rust ... 264
Времена жизни в Rust ... 264
Сырые и умные указатели ... 266
Умный указатель Box ... 266
Умный указатель Rc ... 268
Умный указатель Arc ... 269
Умный указатель RefCell ... 271
Умный указатель Mutex ... 272
Умный указатель RwLock ... 275
Атомарные операции ... 278
Незащищенный Rust ... 280
Незащищенные функции ... 280
Незащищенные пакеты ... 280
Конкретно-прикладные незащищенные абстракции ... 281
Интерфейс с внешними функциями ... 282
Рекомендации по написанию незащищенного исходного кода ... 283
Лучшие образцы практики управления памятью ... 283
Приоритет выделения памяти в стеке ... 284
Активное задействование ссылок ... 284
Разумное использование умных указателей ... 284
Использование сопоставления с шаблоном ... 285
Аннотирование времен жизни ... 285
Надлежащее управление ресурсами ... 286
Многопоточность и конкурентность ... 286
Осторожное использование незащищенного исходного кода ... 286
Профилирование и оптимизация ... 287
Продвинутое управление памятью ... 288
Конкретно-прикладные выделители памяти ... 288
Отображение файлов в память ... 290
Операционная совместимость с другими языками ... 291
Лучшие образцы практики управления памятью ... 292
Заключение ... 293

Глава 7. Управление конкурентностью ... 294

Введение ... 294
Содержание главы ... 294
Понимание конкурентного программирования ... 294
Параллелизм против конкурентности ... 295
Порождение потоков исполнения и управление ими ... 296
Коллективное использование данных между потоками исполнения ... 297
Конкурентные структуры данных ... 297
Базовое использование контейнера-синхронизатора Mutex ... 298
Коллективное использование данных между потоками ... 300
Предотвращение взаимных блокировок ... 301
Владение сторожами мьютекса ... 303

Обработка скомпрометированных мьютексов 304
Продвинутое использование мьютекса 305
Условное запирание на замок 305
Реализация защищенной мьютексом очереди 307
Реализация защищенной мьютексом очереди с приоритетом 308
Обработка взаимных блокировок с помощью мьютекса 310
Мьютекс в многопоточном сценарии производитель–потребитель 311
Обследование контейнера-синхронизатора RwLock 313
Управление коллективными данными с помощью примитива синхронизации RwLock 313
Продвинутые варианты использования контейнера-синхронизатора RwLock в Rust 315
Динамическое количество читателей 315
Запирание на замок по времени 316
Предотвращение взаимных блокировок 318
Реализация объединения ресурсов в пул 319
Обмен данными между потоками исполнения и передача сообщений 321
Использование каналов для обмена данными 321
Передача сообщений с использованием структур 322
Атомарные операции 323
Продвинутый обмен данными между потоками исполнения 324
Координация потоков исполнения с помощью барьеров 325
Локальное для потока хранение данных 326
Библиотека crossbeam 328
Асинхронное программирование 329
Лучшие образцы практики программирования конкурентности 330
Заключение 331
Дополнительные ресурсы 332
Вопросы для самопроверки 333
Ответы 336
Ключевые термины 336

Глава 8. Консольные программы 339

Введение 339
Содержание главы 339
Введение в разбор аргументов 339
Преимущества использования библиотеки clap 340
Начало работы с библиотекой clap 341
Обработка нескольких аргументов 343
Обработка аргументов-флагов 345
Обработка продвинутых аргументов 346
Реализация логики поиска и замены текста 349
Обзор функциональности утилиты поиска и замены текста 349
Чтение входного файла 350
Логика поиска и замены 354
Запись измененного содержимого 357
Поиск без учета регистра символов (опционально) 360
Поддержка нескольких входных файлов 364

Заключение 367
Дополнительные ресурсы 368
Вопросы для самопроверки 369
Ответы 371
Ключевые термины 371

Глава 9. Работа с вводом-выводом с устройств на Rust 373

Введение 373
Содержание главы 374
Чтение из файлов и запись в файлы 374
Открытие файла 374
Чтение из файла 376
Запись в файл 377
Закрытие файла 380
Выполнение распространенных операций с файловой системой 381
Создание директорий 381
Переименование и перемещение файлов 382
Проверка метаданных из файлов 383
Удаление файлов и директорий 385
Обход директорий 386
Работа с путями к файлам 388
Чтение и запись двоичных данных 390
Обработка ошибок и типы Result 391
Работа с аппаратными устройствами 393
Продвинутые файловые операции 395
Символические и жесткие ссылки 395
Запирание файлов на замок 397
Отображаемые файлы 401
Лучшие образцы практики файлового ввода-вывода 402
Драйверы устройств и модули ядра 405
Сетевое программирование и ввод-вывод с устройств 406
Сокеты и протоколы 407
Асинхронное сетевое взаимодействие и ввод-вывод с устройств 410
Библиотеки для сетевого взаимодействия и ввод-вывод с устройств 414
Вопросы защищенности и кибербезопасности 415
Параллелизм, конкурентность и ввод-вывод с устройств 418
Реально-временные системы и ввод-вывод с устройств 420
Заключение 423
Дополнительные ресурсы 424
Вопросы для самопроверки 425
Ответы 427
Ключевые термины 428

Глава 10. Итераторы и замыкания 430

Введение 430
Содержание главы 430
Итераторы 430
Анатомия итератора 431

Ключевые общие черты: Iterator и IntoIterator 433
Создание итераторов с помощью функций from_fn и successors 435
Метод drain 438
Другие источники итераций 439
Адаптерные методы итераторов 442
Построение конкретно-прикладных итераторов 444
Ленивые вычисления 446
Продвинутые методы: flat_map, take, skip и peekable 447
Адаптерный метод flat_map 447
Адаптерные методы take и skip 448
Адаптерный метод peekable 449
Адаптерные методы filter_map, fuse и flatten 451
Адаптерный метод filter_map 451
Адаптерный метод fuse 452
Адаптерный метод flatten 454
Волшебство замыканий 455
Синтаксис замыканий 456
Захват переменных в замыканиях 456
Заимствование переменных 457
Перемещение переменных 458
Замыкания и общие черты Fn, FnMut, FnOnce 459
Замыкания Fn 459
Замыкания FnMut 460
Замыкания FnOnce 461
Итераторы для эффективной обработки данных 462
Накопление данных с помощью метода fold 463
Выстраивание итераторов в цепочку с помощью метода chain 464
Применение итераторов и замыканий в практических примерах 465
Вариант использования 1: анализ текста 465
Вариант использования 2: обработка изображений 467
Вариант использования 3: фильтрация и преобразование данных 468
Вариант использования 4: обработка сенсорных данных 470
Вариант использования 5: финансовые расчеты 471
Вариант использования 6: сортировка 473
Заключение 474
Дополнительные ресурсы 475
Вопросы для самопроверки 476
Ответы 479
Ключевые термины 479

Глава 11. Модульное тестирование на Rust 481

Введение 481
Содержание главы 481
Модульное тестирование 482
Написание тестовых функций и модулей 482
Написание простейшей тестовой функции 483
Тестовые модули 484
Тесты и результаты тестирования 485

Выполнение тестов ... 485
Интерпретация результатов тестирования ... 487
Сверки ... 489
Конкретно-прикладные сообщения об ошибках ... 491
Выполнение тестов с опциями ... 491
Фикстуры и исходный код подготовки ... 492
Условные тесты ... 498
Асинхронное тестирование ... 500
Создание заглушек для зависимостей ... 502
Тестирование приватных функций ... 503
Замеры производительности ... 505
Покрытие исходного кода ... 506
Непрерывная интеграция ... 506
Заключение ... 507
Дополнительные ресурсы ... 508
Вопросы для самоконтроля ... 509
Ответы ... 512
Ключевые термины ... 512

Глава 12. Сетевое программирование ... 515

Введение ... 515
Содержание главы ... 516
Сетевое программирование ... 516
Разработка сетевых приложений ... 516
Протоколы связи: TCP и UDP ... 520
Протокол управления передачей (TCP) ... 521
Протокол пользовательских датаграмм (UDP) ... 524
Асинхронные сетевые операции ... 526
Асинхронное программирование ... 527
Асинхронный TCP-сервер ... 530
Варианты использования из реальной практики: обработка HTTP-запросов ... 531
Приложение для обмена сообщениями по сети ... 533
Сервер передачи файлов ... 536
Сервер дистанционного исполнения команд ... 538
Одноранговый обмен файлами ... 540
Реально-временное коллаборативное редактирование ... 542
Заключение ... 546
Дополнительные ресурсы ... 547
Вопросы для самопроверки ... 548
Ответы ... 550
Ключевые термины ... 551

Глава 13. Незащищенное программирование на Rust ... 552

Введение ... 552
Содержание главы ... 552

Незащищенный исходный код ... 553
Сценарии использования блоков unsafe ... 554
Вопросы производительности и защищенности ... 556
Примеры из реальной практики ... 557
Пример 1: взаимодействие с базой данных через FFI ... 557
Пример 2: продвинутая обработка изображений ... 563
Пример 3: конкретно-прикладное управление памятью ... 566
Лучшие образцы практики использования незащищенного исходного кода ... 568
Нарушения защищенности памяти ... 570
Практические примеры ... 572
Практический пример 1: уязвимость Heartbleed ... 572
Практический пример 2: авария первого полета ракеты Ариан-5 ... 575
Риски использования незащищенного исходного кода ... 580
Риск 1: обращение к памяти по пустому указателю ... 580
Риск 2: переполнение буфера ... 581
Риск 3: ошибки использования памяти после ее высвобождения ... 582
Заключение ... 583
Дополнительные ресурсы ... 584
Вопросы для самопроверки ... 585
Ответы ... 587
Ключевые термины ... 588

Глава 14. Асинхронное программирование ... 590

Введение ... 590
Содержание главы ... 590
Основы синтаксиса async/await Rust ... 591
Обследование динамики синтаксиса async/await ... 593
Использование библиотеки tokio ... 595
Использование ключевого слова async для повышения отзывчивости ... 595
Обработка ошибок в асинхронном исходном коде ... 597
Времена жизни конкурентных заданий ... 599
Продвинутые шаблоны асинхронного программирования ... 601
Асинхронные потоки данных ... 602
Управление ресурсами с помощью асинхронного завершения жизненного цикла ... 606
Разветвление и сведение в асинхронных потоках данных ... 608
Обработка отмен заданий и реализации тайм-аутов ... 611
Управление динамическими заданиями ... 612
Интеграция асинхронного и синхронного исходных кодов ... 614
Заключение ... 616
Дополнительные ресурсы ... 617
Вопросы для самопроверки ... 618
Ответы ... 620
Ключевые термины ... 620

Глава 15. Web Assembly на Rust 622
Введение 622
Содержание главы 622
Преимущества стандарта WebAssembly 623
Ограничения стандарта WebAssembly 623
Ограниченная поддержка веб-браузерами 624
Косвенная манипуляция объектной моделью документа 624
Сложности управления памятью 625
Rust для разработки в сопряжении со стандартом WebAssembly 626
Установка инструментария Rust и стандарта WebAssembly 626
Настройка нового проекта на Rust в связке с двоичным форматом инструкций Wasm 626
Сборка и тестирование модулей WebAssembly 627
Написание функций на Rust с привязкой к Wasm 628
Тестирование модуля WebAssembly 629
Примеры применения стандарта WebAssembly из реальной практики 630
Вариант использования 1: обработка изображений 631
Вариант использования 2: криптографические операции 635
Вариант использования 3: обработка сетевых запросов 638
Заключение 641
Дополнительные ресурсы 641
Вопросы для самопроверки 642
Ответы 644

Предметный указатель 645

Об авторе

Махмуд Хармуш – ориентированный на результат инженер-программист с впечатляющим набором навыков. Будучи универсальным разработчиком и исследователем данных, он обладает подтвержденным опытом создания программных продуктов и сервисов высочайшего уровня. Страсть Махмуда к программированию зажглась еще в студенческие годы и не угасает по сей день. Написание чистого исходного кода и решение сложных алгоритмических задач приносят ему огромное удовольствие и чувство реализации.

Компетенция Махмуда охватывает различные языки программирования и фреймворки, включая Python, C#, Java, Rust и C/C++. Он выделяется глубоким знанием фреймворков FastAPI, Django, NodeJS, Yew и ReactJS. Его навыки технического письма не имеют себе равных: он умеет делать сложную информацию доступной благодаря четкому и лаконичному документированию. Квалификация, эквивалентная степени магистра в области электротехники (Ливанский университет), делает его профессиональный профиль еще более весомым.

Путь Махмуда как программиста начался в колледже, и его энтузиазм продолжает пылать и по сей день. Он эффективно работает как в коллективе сотрудников, так и в качестве самостоятельного специалиста, всегда находясь в поиске новых вызовов для оттачивания мастерства. Махмуд не только быстро обучается, но и активно применяет лучшие образцы практики, такие как разработка через тестирование и методы DevOps (непрерывная разработка), следуя принципам SOLID и DRY.

Помимо технических навыков, Махмуд обладает талантом представлять сложную информацию в простом и понятном виде. В сочетании со стремлением к выстраиванию долгосрочных отношений с клиентами это позволяет ему создавать отказоустойчивые программные продукты, которые эффективно решают бизнес-задачи.

Вне профессиональной деятельности Махмуд любит проводить время с близкими. Будь то активный отдых на природе – езда на велосипеде и походы в горах рядом с домом – или погружение в хорошую книгу, он находит радость в каждом моменте жизни.

Конец ознакомительного фрагмента.
Приобрести книгу можно
в интернет-магазине
«Электронный универс»
e-Univers.ru