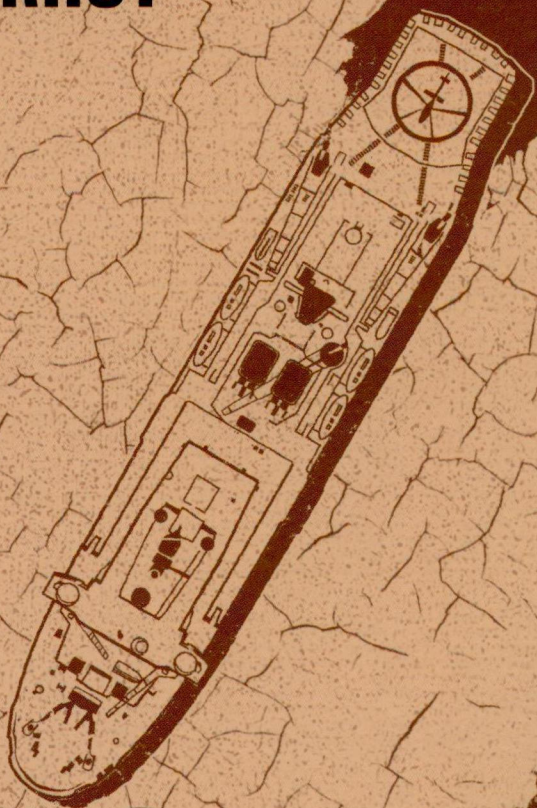


ПАРАЛЛЕЛЬНОЕ ПРОГРАММИРОВАНИЕ — ТАК ЛИ ЭТО СЛОЖНО?



Пол Маккинни



Пол Маккинни

**ПАРАЛЛЕЛЬНОЕ
ПРОГРАММИРОВАНИЕ —
ТАК ЛИ ЭТО
СЛОЖНО?**

Санкт-Петербург
«БХВ-Петербург»
2024

УДК 004.272
ББК 32.973.26-018.1
М15

Маккинни П.

М15 Параллельное программирование — так ли это сложно? : Пер. с англ. — СПб.: БХВ-Петербург, 2024. — 1072 с.: ил.

ISBN 978-5-9775-1927-4

Книга исследует различные низкоуровневые механизмы и алгоритмы, лежащие в основе современных параллельных и конкурентных вычислений, в частности реализованные в ядре Linux. Рассмотрены примитивы синхронизации (мьютексы и блокировки), владение данными, валидация, копирование и запись, эвристические методы разработки параллельных и конкурентных алгоритмов, подбор аппаратного обеспечения и другие малоизвестные находки в области параллелизма. Также уделено внимание упрощению и оптимизации параллельных вычислений. Наконец, спрогнозированы возможные тенденции развития параллельного программирования с учётом современных разработок нового аппаратного обеспечения.

*Для специалистов по параллельному программированию,
Linux, работе с памятью и ресурсами в операционных системах*

УДК 004.272
ББК 32.973.26-018.1

Группа подготовки издания:

Руководитель проекта	<i>Олег Сивченко</i>
Зав. редакцией	<i>Людмила Гауль</i>
Перевод с английского	<i>Михаила Райтмана</i>
Редактор	<i>Дарья Кустовская</i>
Компьютерная верстка	<i>Натальи Смирновой</i>
Корректор	<i>Светлана Крутоярова</i>
Оформление обложки	<i>Зои Канторович</i>

Copyright © 2022 by Paul E. McKenney
Translation Copyright © 2023 by BHV. All rights reserved.

Перевод © 2023 BHV. Все права защищены.

Подписано в печать 08.11.23.
Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 86,43.
Тираж 1500 экз. Заказ № 10237.
"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.

Отпечатано с готовых файлов заказчика
в АО «Первая Образцовая типография»,
филиал «УЛЬЯНОВСКИЙ ДОМ ПЕЧАТИ»
432980, Россия, г. Ульяновск, ул. Гончарова, 14

ISBN 978-5-9775-1927-4 (рус.)

© Paul E. McKenney, 2022
© Перевод на русский язык, оформление.
ООО "БХВ-Петербург", ООО "БХВ", 2023

Оглавление

ПРЕДИСЛОВИЕ К РУССКОМУ ИЗДАНИЮ.....	15
ОТ АВТОРА	16
ПАРАЛЛЕЛЬНОЕ ПРОГРАММИРОВАНИЕ: СЛОЖНО ИЛИ ВОЗМОЖНО?	17
Нормативные положения.....	17
Цветные изображения.....	18
ГЛАВА 1. КАК ПОЛЬЗОВАТЬСЯ ЭТОЙ КНИГОЙ.....	19
1.1 Структура книги.....	20
1.2 Экспресс-тесты.....	21
1.3 Альтернативы этой книге.....	22
1.4 Пример исходного кода.....	24
1.5 О чем эта книга?.....	24
ГЛАВА 2. Введение.....	27
2.1 Исторические трудности параллельного программирования.....	28
2.2 Цели параллельного программирования.....	29
2.2.1 Производительность.....	30
2.2.2 Продуктивность.....	32
2.2.3 Универсальность.....	33
2.3 Альтернативы параллельному программированию.....	37
2.3.1 Несколько экземпляров последовательного приложения.....	37
2.3.2 Использование существующего параллельного ПО.....	37
2.3.3 Оптимизация производительности.....	38
2.4 Почему параллельное программирование такое сложное?.....	39
2.4.1 Разделение работы.....	40
2.4.2 Управление параллельным доступом.....	41
2.4.3 Разделение и репликация ресурсов.....	42
2.4.4 Взаимодействие с оборудованием.....	42
2.4.5 Комбинированные возможности.....	43
2.4.6 Как языки и среда помогают в решении этих задач?.....	43
2.5 Обсуждение.....	44
ГЛАВА 3. АППАРАТНОЕ ОБЕСПЕЧЕНИЕ И С ЧЕМ ЕГО ЕДЯТ.....	45
3.1 Обзор.....	45
3.1.1 Конвейерные процессоры.....	46
3.1.2 Ссылки на память.....	49

3.1.3 Атомарные операции	50
3.1.4 Барьеры памяти	51
3.1.5 Промахи кэша	51
3.1.6 Операции ввода-вывода	52
3.2 Накладные расходы	53
3.2.1 Архитектура аппаратной системы	53
3.2.2 Затраты на операции	55
3.2.3 Аппаратная оптимизация	59
3.3 Бесплатные обеды	60
3.3.1 3D-интеграция	61
3.3.3 Свет вместо электронов	62
3.3.4 Ускорители специального назначения	63
3.3.5 Существующее параллельное программное обеспечение	64
3.4 Результаты проектирования программного обеспечения	64
ГЛАВА 4. ИНСТРУМЕНТАРИЙ	66
4.1 Языки сценариев	66
4.2 Многопроцессорность в системах POSIX	68
4.2.1 Создание и уничтожение процессов POSIX	68
4.2.2 Создание и уничтожение потоков POSIX	71
4.2.3 Блокировка в POSIX	72
4.2.4 Блокировка чтения-записи POSIX	77
4.2.5 Атомарные операции (GCC Classic)	82
4.2.6 Атомарные операции (C11)	83
4.2.7 Атомарные операции (Современный GCC)	83
4.2.8 Поточковые переменные	83
4.3 Альтернативы операциям POSIX	84
4.3.1 Организация и инициализация	84
4.3.2 Создание, уничтожение потоков и управление ими	85
4.3.3 Блокировка	88
4.3.4 Доступ к общим переменным	89
4.3.5 Атомарные операции	102
4.3.6 Переменные для каждого процессора	103
4.4 Как выбрать подходящий инструмент?	105
ГЛАВА 5. ПОДСЧЕТ	106
5.1 Что сложного в конкурентном счете?	107
5.2 Статистические счетчики	111
5.2.1 Дизайн	111
5.2.2 Реализация на основе массива	111
5.2.3 Реализация на основе переменной потока	113
5.2.4. Реализация с достижимой согласованностью	116
5.2.5 Обсуждение	118
5.3 Приблизительные предельные счетчики	119
5.3.1 Дизайн	119
5.3.2 Реализация простого счетчика пределов	121

5.3.3 Простой счетчик пределов	127
5.3.4 Реализация приближенного счетчика предельных значений	128
5.3.5 Приближенный счетчик предельных значений.....	129
5.4 Точные счетчики предельных значений.....	129
5.4.1 Реализация атомарного предельных значений	130
5.4.2 Атомарный счетчик предельных значений: обсуждение	138
5.4.3 Дизайн счетчика предельных значений с кражей сигналов.....	138
5.4.4 Реализация счетчика предельных значений на основе кражи сигналов	139
5.4.5 Счетчик предельных значений с кражей сигналов: обсуждение.....	146
5.4.6 Применение точных счетчиков предельных значений.....	146
5.5 Обсуждение параллельного подсчета.....	148
5.5.1 Валидация параллельного подсчета	148
5.5.2 Производительность параллельного счета	149
5.5.3 Специализации параллельного счета	150
5.5.4 Уроки параллельного подсчета.....	151
Глава 6. ПРОЕКТИРОВАНИЕ РАЗДЕЛЕНИЯ И СИНХРОНИЗАЦИИ	154
6.1 Упражнения на разделение.....	155
6.1.1 Проблема обедающих философов	155
6.1.2 Двусторонняя очередь	157
6.1.3 Пример разделения: обсуждение.....	168
6.2 Критерии дизайна	168
6.3 Детализация синхронизаций	171
6.3.1 Последовательная программа	172
6.3.2 Кодовая блокировка.....	174
6.3.3 Блокировка данных.....	175
6.3.4 Владение данными.....	178
6.3.5 Детализация и производительность блокировки	179
6.4 Параллельный быстрый путь	182
6.4.1 Блокировка чтения-записи	183
6.4.2 Иерархическая блокировка	184
6.4.3 Кэши распределителя ресурсов	185
6.5 Методы без разделения?	192
6.5.1 Решатель параллельных лабиринтов с очередью работ	192
6.5.2 Альтернативный решатель параллельного лабиринта	195
6.5.3 Проверка лабиринта.....	198
6.5.4 Сравнение производительности I	198
6.5.5 Альтернативный последовательный решатель лабиринта.....	201
6.5.6 Сравнение производительности II.....	201
6.5.7 Направления дальнейшего развития и выводы	203
6.6 Разделение, параллелизм и оптимизация	204
Глава 7. БЛОКИРОВКА	205
7.1 Остаться в живых	206
7.1.1 Взаимная блокировка	206
7.1.2 Динамическая блокировка и голодание.....	218

7.1.3	Несправедливость	220
7.1.4	Неэффективность	221
7.2	Типы блокировок.....	222
7.2.1	Эксклюзивные блокировки	222
7.2.2	Блокировки чтения-записи.....	223
7.2.3	Прочие средства, помимо блокировок чтения-записи.....	225
7.2.4	Блокировка с ограниченной областью действия.....	226
7.3	Проблемы с реализацией блокировки	230
7.3.1	Пример реализации эксклюзивной блокировки на основе атомарного обмена	230
7.3.2	Другие реализации эксклюзивной блокировки.....	231
7.4	Гарантии существования на основе блокировки	234
7.5	Блокировка: герой или злодей?	237
7.5.1	Блокировка приложений: герой!.....	237
7.5.2	Блокировка в параллельных библиотеках: еще один инструмент	238
7.5.3	Блокировка при распараллеливании последовательных библиотек: злодей!.....	242
7.6	Резюме	245
ГЛАВА 8. ВЛАДЕНИЕ ДАННЫМИ.....		246
8.1	Несколько процессов.....	247
8.2	Частичное владение данными и pthreads.....	247
8.3	Доставка функций	248
8.4	Назначенный поток	249
8.5	Приватизация	249
8.6	Другие разновидности владения данными.....	250
ГЛАВА 9. ОТЛОЖЕННАЯ ОБРАБОТКА		252
9.1	Пример запуска.....	252
9.2	Подсчет ссылок.....	255
9.3	Указатели опасности	259
9.4	Блокировки последовательности	267
9.5	Механизм Read-Copy-Update (RCU).....	274
9.5.1	Введение в RCU	275
9.5.2	Основы RCU	285
9.5.3	API RCU ядра Linux.....	297
9.5.4	Использование RCU.....	314
9.5.5	Работа, связанная с RCU	343
9.6	Что же выбрать?	348
9.6.1	Что выбрать? Обзор.....	348
9.6.2	Что выбрать? Подробный обзор	350
9.6.3	Что выбрать? Использование в производстве.....	354
9.7	А что насчет обновлений?	356

ГЛАВА 10. СТРУКТУРЫ ДАННЫХ	357
10.1 Мотивирующее приложение	357
10.2 Разделяемые структуры данных.....	358
10.2.1 Дизайн хэш-таблицы	358
10.2.2 Реализация хэш-таблицы.....	359
10.2.3 Производительность хэш-таблицы.....	363
10.3 Структуры данных преимущественно для чтения	365
10.3.1 Реализация хэш-таблицы с защитой RCU	365
10.3.2 Проверка защищенной RCU хэш-таблицы	367
10.3.3 Производительность защищенной RCU хэш-таблицы.....	368
10.3.4 Хэш-таблицы с защитой RCU: обсуждение	372
10.4 Неразделяемые структуры данных	374
10.4.1 Дизайн хэш-таблицы с изменяемым размером	374
10.4.2 Реализация хэш-таблицы с изменяемым размером	376
10.4.3 Хэш-таблицы изменяемого размера: обсуждение	385
10.4.4 Другие хэш-таблицы с изменяемым размером	386
10.5 Прочие структуры данных.....	390
10.6 Микрооптимизация	392
10.6.1 Специализация	392
10.6.2 Биты и байты	393
10.6.3 Аппаратные соображения	394
10.7 Резюме	396
ГЛАВА 11. ВАЛИДАЦИЯ	397
11.1 Введение	398
11.1.1 Откуда берутся ошибки?	398
11.1.2 Необходимый образ мышления	400
11.1.3 Когда должна начинаться валидация?	402
11.1.4 Путь с открытым исходным кодом	403
11.2 Трассировка.....	405
11.3 Утверждения	406
11.4 Статический анализ	407
11.5 Рецензирование кода	408
11.5.1 Инспекция	408
11.5.2 Пошаговый разбор	409
11.5.3 Самопроверка	409
11.6 Вероятности и гейзенбаги.....	412
11.6.1 Статистика в дискретном тестировании	413
11.6.2 Злоупотребление статистикой для дискретного тестирования.....	415
11.6.3 Статистика непрерывного тестирования	416
11.6.4 Охота на гейзенбаги.....	418
11.7 Оценка производительности.....	424
11.7.1 Эталонные тесты	425
11.7.2 Профилирование	425
11.7.3 Дифференциальное профилирование.....	426

11.7.4 Микроэталонные тесты	426
11.7.5 Изоляция	428
11.7.6 Обнаружение помех	429
11.8 Резюме	434
ГЛАВА 12. ФОРМАЛЬНАЯ ВЕРИФИКАЦИЯ	436
12.1 Поиск в пространстве состояний	436
12.1.1 Promela и Spin	437
12.1.2 Как использовать Promela	443
12.1.3 Пример Promela: блокировка	447
12.1.4 Пример Promela: QRCU	450
12.1.5 Promela Parable: dynticks и RCU с вытесняющим выполнением	461
12.1.6 Проверка RCU с вытесняющим выполнением и dynticks	467
12.2 Поиск в пространстве состояний специального назначения	496
12.2.1 Анатомия лакмусовой бумажки	497
12.2.2 Что означает эта лакмусовая бумажка?	499
12.2.3 Запуск лакмусовой бумажки	500
12.2.4 Обсуждение PPCMEM	501
12.3 Аксиоматические подходы	503
12.3.1 Аксиоматические подходы и блокировка	505
12.3.2 Аксиоматические подходы и RCU	507
12.4 SAT-решатели	511
12.5 Средства проверки моделей без сохранения состояния	513
12.6 Резюме	514
12.7 Выбор плана проверки	516
ГЛАВА 13. СОБИРАЕМ ВСЕ ВМЕСТЕ	520
13.1 Головоломки со счетчиками	520
13.1.1 Подсчет обновлений	520
13.1.2 Подсчет поисковых запросов	521
13.2 Вернемся к подсчету ссылок	521
13.2.1 Реализация категорий подсчета ссылок	523
13.2.2 Оптимизации счетчика	530
13.3 Помощники указателя опасности	530
13.3.1 Масштабируемый счетчик ссылок	530
13.4 Блокировка последовательности	531
13.4.1 Дуэльные блокировки последовательности	531
13.4.2 Коррелированные элементы данных	532
13.4.3 Атомарное перемещение	533
13.4.4 Превращение в писателя	535
13.5 RCU спешит на помощь	535
13.5.1 RCU и статистические счетчики на основе переменных потока	535
13.5.2 RCU и счетчики съемных устройств ввода-вывода	539
13.5.3 Массив и длина	540
13.5.4 Коррелированные поля	542

13.5.5 Удобный для обновления обход	543
13.5.6 Масштабируемый счетчик ссылок 2	543
13.5.7 Перезапущенные периоды простоя	544
Глава 14. Продвинутая синхронизация	548
14.1 Как избежать блокировки	548
14.2 Неблокирующая синхронизация	549
14.2.1 Простой NBS	550
14.2.2 Применимость преимуществ NBS	554
14.2.3 Обсуждение NBS	557
14.3 Параллельные вычисления в реальном времени	558
14.3.1 Что такое вычисления в реальном времени?	558
14.3.2 Кому нужен режим реального времени?	565
14.3.3 Кому нужен параллельный режим реального времени?	566
14.3.4 Реализация параллельных систем реального времени	567
14.3.5 Реализация параллельных операционных систем реального времени	569
14.3.6 Реализация параллельных приложений реального времени	586
14.3.7. Реальное время против реального быстрого: как выбрать?	591
Глава 15. Продвинутая синхронизация: порядок памяти	594
15.1 Упорядочение: зачем и как?	595
15.1.1 Почему аппаратное обеспечение нарушает порядок?	596
15.1.2 Как принудительно выполнить упорядочение?	599
15.1.3 Базовые эмпирические правила	603
15.2 Трюки и ловушки	605
15.2.1 Переменные с несколькими значениями	605
15.2.2 Переупорядочивание ссылок на память	609
15.2.3 Адресные зависимости	612
15.2.4 Зависимости данных	615
15.2.5 Зависимости управления	616
15.2.6 Когерентность кэша	618
15.2.7 Атомарность множественных копий	620
15.3 Ужасы времени компиляции	635
15.3.1 Ограничения по ссылке на память	636
15.3.2 Проблемы с зависимостями от адреса и от данных	637
15.3.3. Проблемы с зависимостями управления	642
15.4 Примитивы более высокого уровня	648
15.4.1 Выделение памяти	648
15.4.2 Блокировка	649
15.4.3 RCU	656
15.5 Особенности аппаратного обеспечения	670
15.5.1 Alpha	673
15.5.2 Armv7-A/R	677
15.5.3 Armv8	678
15.5.4 Itanium	679

15.5.5 MIPS	680
15.5.6 POWER / PowerPC	681
15.5.7 SPARC TSO	683
15.5.8 x86	684
15.5.9 z Systems	685
15.6 Где требуется упорядочение памяти?	685
ГЛАВА 16. ПРОСТОТА ИСПОЛЬЗОВАНИЯ	688
16.1 Что такое «простота»?	688
16.2 Шкала Расти при проектировании API	689
16.3 Стрижка множества Мандельброта	690
ГЛАВА 17. ПРОТИВОРЕЧИВЫЕ ВЗГЛЯДЫ НА БУДУЩЕЕ	693
17.1 Будущее процессорных технологий стало не таким, каким должно было	693
17.1.1 Однопроцессорность превыше всего	695
17.1.2 Потокотомания	695
17.1.3 Атака клонов	696
17.1.4 Краш-тест о стену памяти	696
17.1.5 Поразительные ускорители	698
17.2 Транзакционная память	698
17.2.1 Внешний мир	699
17.2.2 Модификация процесса	704
17.2.3 Синхронизация	710
17.2.4 Обсуждение	716
17.3 Аппаратная транзакционная память	719
17.3.1 Преимущества НТМ при блокировке WRT	720
17.3.2 Слабые стороны НТМ по сравнению с блокировкой	722
17.3.3. Слабые стороны НТМ по сравнению с блокировкой и не только	731
17.3.4 Где лучше всего использовать НТМ?	735
17.3.5 Что может повлиять на ситуацию	736
17.3.6 Выводы	740
17.4 Формальное регрессионное тестирование?	741
17.4.1 Автоматический перевод	741
17.4.2 Среда	743
17.4.3 Накладные расходы	743
17.4.4 Поиск ошибок	745
17.4.5 Минимальный набор инструментов	746
17.4.6 Релевантные ошибки	747
17.4.7 Формальная система оценки регрессии	748
17.5 Функциональное программирование в параллелизме	750
17.6 Резюме	752
ГЛАВА 18. ПОДВЕДЕМ ИТОГИ И НАМЕТИМ ПЕРСПЕКТИВЫ	753

ПРИЛОЖЕНИЕ А. ВАЖНЫЕ ВОПРОСЫ.....	757
A.1 Почему параллельные программы не всегда быстрее?	757
A.2 Почему бы не убрать блокировку?	758
A.3 Который сейчас час?	758
A.4 Что значит «после»?	760
A.5 Сколь сильным должно быть упорядочение?	764
A.5.1 Где находятся определяющие данные?	765
A.5.2 Согласованные данные используются последовательно?	766
A.5.3 Разделяема ли задача?	766
A.5.4 А если все это неверно?	766
A.6 В чем разница между «конкурентным» и «параллельным»?	767
A.7 Почему программа глючит?	768
ПРИЛОЖЕНИЕ Б. «ИГРУШЕЧНЫЕ» РЕАЛИЗАЦИИ RCU	769
B.1 RCU на основе блокировки	769
B.2 RCU на основе блокировки потока.....	770
B.3 Простой RCU на основе счетчика.....	772
B.4 RCU без голодания на основе счетчика	774
B.5 Масштабируемый RCU на основе счетчиков	778
B.6 Масштабируемый RCU на основе счетчиков с общими периодами простоя.....	781
B.7 RCU на основе автономного счетчика	784
B.8 RCU с вложенностью на основе автономного счетчика.....	787
B.9 RCU на основе состояний покоя.....	790
B.10 Краткий обзор «игрушечных» реализаций RCU	793
ПРИЛОЖЕНИЕ В. ЗАЧЕМ НУЖНЫ БАРЬЕРЫ ПАМЯТИ?	795
V.1 Структура кэша	795
V.2 Протоколы когерентности кэша	798
V.2.1 Состояние MESI	798
V.2.2 Сообщения протокола MESI	799
V.2.3 Диаграмма состояний MESI	800
V.2.4 Пример протокола MESI.....	802
V.3 Сохранения порождают ненужные простои.....	803
V.3.1 Буферы сохранения	804
V.3.2 Переадресация хранения.....	805
V.3.3. Буферы хранения и барьеры памяти.....	806
V.4 Последовательности сохранения приводят к ненужным задержкам	809
V.4.1 Очереди недействительности	810
V.4.2 Очереди недействительности и подтверждения недействительности	810
V.4.3 Очереди недействительности и барьеры памяти.....	811
V.5 Чтение и запись барьеров памяти.....	814
V.6 Примеры последовательностей барьера памяти	815
V.6.1 Враждебная порядку архитектура	815
V.6.2 Пример 1.....	816

В.6.3 Пример 2.....	817
В.6.4 Пример 3.....	818
В.7 Барьеры памяти с нами навсегда?	818
В.8 Советы разработчикам аппаратного обеспечения	819
ПРИЛОЖЕНИЕ Г. ОТВЕТЫ НА БЫСТРЫЕ ТЕСТЫ.....	821
Г.1 Как пользоваться этой книгой.....	821
Г.2 Введение.....	822
Г.3 Аппаратное обеспечение и его привычки.....	829
Г.4 Инструментарий	835
Г.5 Подсчет	845
Г.6 Разбиение на разделы и синхронизация.....	868
Г.7 Блокировка.....	877
Г.8 Владение данными	889
Г.9 Отложенная обработка	891
Г.10 Структуры данных	915
Г.11 Проверка	924
Г.12 Формальная верификация	935
Г.13 Собираем все вместе	947
Г.14 Продвинутая синхронизация.....	953
Г.15 Продвинутая синхронизация: упорядочение памяти.....	956
Г.16 Простота использования.....	976
Г.17 Противоречивые взгляды на будущее.....	977
Г.18 Важные вопросы.....	987
Г.19 «Игрушечные» реализации RCU	988
Г.20 Зачем нужны барьеры памяти?	998
ГЛОССАРИЙ.....	1004
БИБЛИОГРАФИЯ	1014
БЛАГОДАРНОСТИ.....	1058
Советник LATEX.....	1058
Рецензенты	1058
Владельцы машин.....	1059
Оригинальные публикации.....	1059
Авторство рисунков	1060
Прочая поддержка	1061
АКРОНИМЫ	1062
ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ	1063
УКАЗАТЕЛЬ API.....	1070