УДК 004.45
ББК 32.972.1
А79

Арпачи-Дюссо Р. Х., Арпачи-Дюссо А. К.

А79 Операционные системы: Три простых элемента / пер. с англ. А. А. Слинкина. – М.: ДМК Пресс, 2021. – 730 с.: ил.

ISBN 978-5-97060-932-3

В книге рассматриваются три фундаментальные концепции операционных систем: виртуализация (процессора и памяти), конкурентность (блокировки и условные переменные) и долговременное хранение (диски, RAID-массивы, файловые системы). В каждой главе представлена одна конкретная проблема и описано ее решение. Приводятся советы, которые могут пригодиться читателю при создании собственных систем.

Выполняя задания, предложенные авторами, и работая над серьезными проектами, читатели приходят к более глубокому пониманию современных ОС. Задания-эмуляторы способны генерировать практически бесконечное множество задач, благодаря чему можно многократно перепроверять свои знания. Все проекты, а также примеры кода написаны на языке программирования C.

Издание адресовано студентам технических вузов и всем, кто интересуется программированием. Преподаватели технических вузов могут использовать книгу в рамках курса информатики.

УДК 004.45
ББК 32.972.1

ISBN 978-1-4842-6466-9 (англ.)
ISBN 978-5-97060-932-3 (рус.)

ŋqqbyeY bE dedEbYse d ecabcY Jqqbyqyè

Содержание

От издательства 27

Предисловие 28

Глава 1. Диалог о книге 36

Глава 2. Введение в операционные системы 38
2.1. Виртуализация процессора 40
2.2. Виртуализация памяти 42
2.3. Конкурентность 44
2.4. Хранение 46
2.5. Цели проектирования 48
2.6. Немного истории 50
Первые операционные системы: просто библиотеки 50
Не только библиотеки: защита 50
Эра мультипрограммирования 51
Современность 52
2.7. Резюме 54
Литература 55
Домашнее задание 56

Часть I. ВИРТУАЛИЗАЦИЯ 58

Глава 3. Диалог о виртуализации 59

Глава 4. Абстракция: процесс 61
4.1. Абстракция: процесс 62
4.2. API процессов 63
4.3. Создание процесса: подробности 64
4.4. Состояния процесса 65
4.5. Структуры данных 67
4.6. Резюме 69
Литература 70
Домашнее задание (эмуляция) 70
Вопросы 71

Глава 5. Интерлюдия: API процессов 72
5.1. Системный вызов `fork()` 72
5.2. Системный вызов `wait()` 74
5.3. И наконец, системный вызов `exec()` 75
5.4. Почему? Мотивация API 77

5.5. Управление процессами и пользователи 79
5.6. Полезные инструменты 80
5.7. Резюме 81
Литература 82
Домашнее задание (кодирование) 83
Вопросы 83

Глава 6. Механизм: ограниченное прямое выполнение 85
6.1. Базовая техника: ограниченное прямое выполнение 86
6.2. Проблема 1: запрещенные операции 86
6.3. Проблема 2: переключение между процессами 91
Кооперативный подход: дождаться системного вызова 91
Некооперативный подход: ОС силой забирает управление 92
Сохранение и восстановление контекста 93
6.4. Сомневаетесь насчет конкурентности? 96
6.5. Резюме 97
Литература 98
Домашнее задание (измерение) 99

Глава 7. Планирование: введение 101
7.1. Предположения о рабочей нагрузке 101
7.2. Метрики планирования 102
7.3. Первым пришел, первым ушел (FIFO) 103
7.4. Сначала самое короткое 104
7.5. Сначала с наименьшим временем до завершения 106
7.6. Новая метрика: время отклика 106
7.7. Циклическое планирование 107
7.8. Учет ввода-вывода 110
7.9. Долой оракулов 111
7.10. Резюме 111
Литература 112
Домашнее задание (эмуляция) 113
Вопросы 113

Глава 8. Планирование: многоуровневая аналитическая очередь 114
8.1. MLFQ: основные правила 115
8.2. Попытка 1: как изменять приоритеты 116
Пример 1: одно долго работающее задание 117
Пример 2: к нам приходит короткое задание 117
Пример 3: а как насчет ввода-вывода? 118
Проблемы текущей реализации MLFQ 119
8.3. Попытка 2: повышение приоритета 120
8.4. Попытка 3: улучшенный учет 121
8.5. Настройка MLFQ и другие вопросы 122
8.6. MLFQ: резюме 124

Литература 124
Домашнее задание (эмуляция) 126
Вопросы 126

Глава 9. Планирование: пропорциональная доля 127

9.1. Основная идея: ваша доля представлена билетом 127
9.2. Механизмы обращения с билетами 129
9.3. Реализация 130
9.4. Пример 131
9.5. Как раздавать билеты? 132
9.6. Зачем отказываться от детерминированности? 132
9.7. Вполне равномерный планировщик в Linux 134
Принцип работы 134
Взвешивание (уровень nice) 136
Использование красно-черных деревьев 137
Обращение со спящими процессами 138
Другие возможности CFS 138
9.8. Резюме 139
Литература 140
Домашнее задание (эмуляция) 141
Вопросы 141

Глава 10. Планирование в многопроцессорных системах (материал повышенной сложности) 142

10.1. Введение: многопроцессорная архитектура 143
10.2. Не забывайте о синхронизации 145
10.3. Последняя проблема: привязка к процессору 146
10.4. Планирование с одной очередью 147
10.5. Планирование с несколькими очередями 148
10.6. Планировщики мультипроцессоров в Linux 151
10.7. Резюме 152
Литература 152
Домашнее задание (эмуляция) 153
Вопросы 154

Глава 11. Заключительный диалог о виртуализации процессора 156

Глава 12. Диалог о виртуализации памяти 158

Глава 13. Абстракция: адресное пространство 160

13.1. Ранние системы 160
13.2. Мультипрограммирование и разделение времени 161
13.3. Адресное пространство 162
13.4. Цели 164

13.5. Резюме 166
Литература 167
Домашнее задание (код) 168
Вопросы 168

Глава 14. Интерлюдия: API памяти 170
14.1. Типы памяти 170
14.2. Вызов `malloc()` 171
14.3. Вызов `free()` 173
14.4. Типичные ошибки 173
Забыли выделить память 173
Выделили недостаточно памяти 174
Забыли инициализировать выделенную память 175
Забыли освободить память 175
Освободили память раньше, чем закончили с ней работать 175
Освободили память несколько раз 176
Неправильно вызвали `free()` 176
Итоги 177
14.5. Поддержка со стороны ОС 177
14.6. Другие вызовы 177
14.7. Резюме 178
Литература 178
Домашнее задание (код) 179
Вопросы 179

Глава 15. Механизм: трансляция адресов 181
15.1. Предположения 182
15.2. Пример 182
15.3. Динамическое (аппаратное) перемещение 185
Пример трансляции 187
15.4. Аппаратная поддержка: итоги 188
15.5. Требования к операционной системе 189
15.6. Резюме 192
Литература 193
Домашнее задание (эмуляция) 194
Вопросы 194

Глава 16. Сегментация 195
16.1. Сегментация: обобщение идеи базы и границы 195
16.2. К какому сегменту мы обращаемся? 198
16.3. А что насчет стека? 200
16.4. Поддержка разделения 200
16.5. Мелкоструктурная и крупноструктурная сегментация 201
16.6. Поддержка со стороны ОС 202
16.7. Резюме 203
Литература 204

Домашнее задание (эмуляция) ... 205
Вопросы ... 205

Глава 17. Управление свободным пространством ... 207
17.1. Предположения ... 208
17.2. Низкоуровневые механизмы ... 209
Разделение и объединение ... 209
Запоминание размеров выделенных блоков ... 211
Встраивание списка свободных ... 212
Увеличение размера кучи ... 217
17.3. Основные стратегии ... 218
Лучший подходящий ... 218
Худший подходящий ... 218
Первый подходящий ... 219
Следующий подходящий ... 219
Примеры ... 219
17.4. Другие подходы ... 220
Сегрегированные списки ... 220
Метод близнецов ... 221
Другие идеи ... 222
17.5. Резюме ... 223
Литература ... 223
Домашнее задание (эмуляция) ... 224
Вопросы ... 224

Глава 18. Страничная организация: введение ... 226
18.1. Простой пример и общий обзор ... 226
18.2. Где хранятся таблицы страниц? ... 230
18.3. Что хранится в таблице страниц? ... 231
18.4. Страничная организация: тоже слишком медленно ... 232
18.5. Трассировка доступа к памяти ... 234
18.6. Резюме ... 237
Литература ... 237
Домашнее задание (эмуляция) ... 238
Вопросы ... 238

Глава 19. Страничная организация: более быстрая трансляция (TLB) ... 240
19.1. Основной алгоритм TLB ... 241
19.2. Пример: доступ к массиву ... 242
19.3. Кто обрабатывает непопадание в TLB? ... 245
19.4. Содержимое TLB: что там хранится? ... 247
19.5. Проблема TLB: контекстные переключения ... 248
19.6. Проблема: политика вытеснения ... 250
19.7. Реальная запись TLB ... 251
19.8. Резюме ... 252

Литература ... 253
Домашнее задание (измерение) ... 254
Вопросы ... 256

Глава 20. Страничная организация: уменьшенные таблицы ... 257
20.1. Простое решение: увеличенные страницы ... 257
20.2. Гибридный подход: страничная организация и сегменты ... 258
20.3. Многоуровневые таблицы страниц ... 261
Подробный пример работы с многоуровневой таблицей страниц ... 264
Больше двух уровней ... 267
Процесс трансляции: вспомним про TLB ... 268
20.4. Инвертированные таблицы страниц ... 269
20.5. Выгрузка таблиц страниц на диск ... 270
20.6. Резюме ... 270
Литература ... 270
Домашнее задание (эмуляция) ... 271
Вопросы ... 271

Глава 21. За пределами физической памяти: механизмы ... 273
21.1. Область подкачки ... 274
21.2. Бит присутствия ... 275
21.3. Отказ страницы ... 276
21.4. А что, если память заполнена? ... 277
21.5. Поток управления при обработке отказа страницы ... 278
21.6. Когда на самом деле происходит замещение ... 279
21.7. Резюме ... 280
Литература ... 281
Домашнее задание (измерение) ... 281
Вопросы ... 282

Глава 22. За пределами физической памяти: политики ... 284
22.1. Управление кешем ... 284
22.2. Оптимальная политика замещения ... 286
22.3. Простая политика: FIFO ... 288
22.4. Еще одна простая политика: случайная ... 289
22.5. Учет истории: LRU ... 290
22.6. Примеры рабочей нагрузки ... 292
22.7. Реализация алгоритмов, учитывающих историю ... 295
22.8. Аппроксимация LRU ... 296
22.9. Учет модифицированных страниц ... 297
22.10. Другие политики ВП ... 298
22.11. Пробуксовка ... 298
22.12. Резюме ... 299
Литература ... 299
Домашнее задание (эмуляция) ... 301
Вопросы ... 301

Глава 23. Полные примеры систем виртуальной памяти 302
23.1. Виртуальная память в VAX/VMS 303
Оборудование управления памятью 303
Реальное адресное пространство 304
Замещение страниц 306
Другие хитрости 308
23.2. Система виртуальной памяти в Linux 309
Адресное пространство Linux 310
Структура таблицы страниц 312
Поддержка больших страниц 313
Страничный кеш 314
Безопасность и переполнение буфера 316
Другие проблемы безопасности: Meltdown и Spectre 318
23.3. Резюме 319
Литература 320

Глава 24. Заключительный диалог о виртуализации памяти 322

Часть II. КОНКУРЕНТНОСТЬ 325

Глава 25. Диалог о конкурентности 326

Глава 26. Конкурентность: введение 328
26.1. Зачем нужны потоки? 329
26.2. Пример: создание потока 330
26.3. Почему становится хуже: разделяемые данные 333
26.4. Суть проблемы: неконтролируемое планирование 335
26.5. Жажда атомарности 337
26.6. Еще одна проблема: ожидание другого потока 339
26.7. Резюме: почему на курсе по ОС? 339
Литература 340
Домашнее задание (эмуляция) 341
Вопросы 341

Глава 27. Интерлюдия: API потоков 343
27.1. Создание потока 343
27.2. Завершение потока 345
27.3. Блокировки 348
27.4. Условные переменные 350
27.5. Компиляция и выполнение 352
27.6. Резюме 352
Литература 353
Домашнее задание (код) 354
Вопросы 354

Глава 28. Блокировки 355
28.1. Блокировки: основная идея 355
28.2. Блокировки в pthread 356
28.3. Конструирование блокировки 357
28.4. Оценивание блокировок 357
28.5. Управление прерываниями 358
28.6. Неудачная попытка: пробуем обойтись командами загрузки и сохранения 359
28.7. Построение работоспособных спин-блокировок с помощью команды проверки и установки 361
28.8. Оценка спин-блокировок 363
28.9. Сравнить и обменять 364
28.10. Загрузить по связи и сохранить условно 366
28.11. Выбрать и прибавить 368
28.12. Слишком много активного ожидания: и как с этим быть? 369
28.13. Простой подход: уступи 369
28.14. Очереди: засыпание вместо активного ожидания 371
28.15. Разные ОС, разная поддержка 374
28.16. Двухфазная блокировка 375
28.17. Резюме 376
Литература 376
Домашнее задание (эмуляция) 378
Вопросы 378

Глава 29. Конкурентные структуры данных с блокировками 380
29.1. Конкурентные счетчики 380
Простой, но немасштабируемый 380
Масштабируемый подсчет 382
29.2. Конкурентные связные списки 385
Масштабирование связных списков 388
29.3. Конкурентные очереди 389
29.4. Конкурентная хеш-таблица 390
29.5. Резюме 392
Литература 392
Домашнее задание (код) 393
Вопросы 393

Глава 30. Условные переменные 395
30.1. Определение и функции 396
30.2. Задача о производителе и потребителе (об ограниченном буфере) 399
Неправильное решение 401
Лучше, но все равно неправильно: While, а не If 404
Решение задачи о производителе и потребителе с буфером на один элемент 406
Правильное решение задачи о производителе и потребителе 407

30.3. Покрывающие условия 408
30.4. Резюме 410
Литература 410
Домашнее задание (код) 411
Вопросы 411

Глава 31. Семафоры 413

31.1. Семафоры: определение 413
31.2. Двоичные семафоры (блокировки) 415
31.3. Использование семафоров для упорядочения 416
31.4. Задача о производителе и потребителе (об ограниченном буфере) 418
Первая попытка 419
Решение: добавление взаимного исключения 421
Предотвращение взаимоблокировки 422
Наконец-то правильное решение 422
31.5. Блокировки чтения-записи 422
31.6. Обедающие философы 425
Неправильное решение 426
Решение: разрыв зависимости 427
31.7. Как реализуются семафоры 428
31.8. Резюме 429
Литература 429
Домашнее задание (код) 431
Вопросы 431

Глава 32. Типичные ошибки в конкурентных программах 433

32.1. Какие бывают ошибки? 433
32.2. Ошибки, не связанные с взаимоблокировкой 434
Ошибки нарушения атомарности 434
Ошибка нарушения порядка 435
Ошибки, не связанные с взаимоблокировкой: резюме 437
32.3. Ошибки, связанные с взаимоблокировкой 437
Почему возникают взаимоблокировки? 438
Условия возникновения взаимоблокировки 439
Предотвращение 440
Циклическое ожидание 440
Ожидание с удержанием 441
Отсутствие вытеснения 441
Взаимное исключение 442
Избегание взаимоблокировок с помощью планирования 444
Найди и исправь 446
32.4. Резюме 446
Литература 447
Домашнее задание (код) 448
Вопросы 448

Глава 33. Событийно-управляемая конкурентность (материал повышенной сложности) 450
33.1. Основная идея: цикл событий 451
33.2. Важный API: **select()** (или **poll()**) 451
33.3. Использование **select()** 452
33.4. Почему проще? Потому что не нужны блокировки 454
33.5. Проблема: блокирующие системные вызовы 454
33.6. Решение: асинхронный ввод-вывод 455
33.7. Еще одна проблема: управление состоянием 457
33.8. Какие еще трудности сопряжены с событиями 458
33.9. Резюме 459
Литература 459
Домашнее задание (код) 460
Вопросы 460

Глава 34. Итоговый диалог о конкурентности 462

Часть III. ХРАНЕНИЕ 464

Глава 35. Диалог о хранении 465

Глава 36. Устройства ввода-вывода 466
36.1. Архитектура системы 466
36.2. Каноническое устройство 468
36.3. Канонический протокол 469
36.4. Прерывания помогают снизить затраты CPU 470
36.5. Более эффективное перемещение данных с помощью ПДП 472
36.6. Методы взаимодействия с устройствами 473
36.7. Сопряжение с ОС: драйвер устройства 474
36.8. Практический пример: простой драйвер IDE-диска 475
36.9. Исторические замечания 478
36.10. Резюме 478
Литература 479

Глава 37. Жесткие диски 481
37.1. Интерфейс 481
37.2. Базовая геометрия 482
37.3. Простой диск 483
Одна дорожка: задержка вращения 483
Несколько дорожек: время поиска 484
Дополнительные детали 485
37.4. Время ввода-вывода: немного арифметики 487
37.5. Планирование диска 490
SSTF: с наименьшим временем поиска первым 490
Лифт (он же SCAN или C-SCAN) 491

SPTF: с наименьшим временем позиционирования первым 492
Другие проблемы планирования 493
37.6. Резюме......494
Литература......494
Домашнее задание (эмуляция)......495

Глава 38. Избыточный массив недорогих дисков (RAID) 498

38.1. Интерфейс и внутреннее устройство RAID......499
38.2. Модель отказов......500
38.3. Как оценивать RAID......500
38.4. RAID уровня 0: чередование......501
Размеры порций......502
Возвращаясь к анализу RAID-0......503
Оценка производительности RAID......503
Снова возвращаемся к анализу RAID-0......505
38.5. RAID уровня 1: зеркалирование......505
Анализ RAID-1......506
38.6. RAID уровня 4: экономия места за счет четности......508
Анализ RAID-4......509
38.7. RAID уровня 5: ротация четности......512
Анализ RAID-5......512
38.8. Сравнение RAID: итоги......513
38.9. Другие интересные вопросы RAID......514
38.10. Резюме......514
Литература......515
Домашнее задание (эмуляция)......516
Вопросы......516

Глава 39. Интерлюдия: файлы и каталоги......517

39.1. Файлы и каталоги......518
39.2. Интерфейс файловой системы......519
39.3. Создание файлов......519
39.4. Чтение и запись файлов......521
39.5. Непоследовательные чтение и запись......522
39.6. Разделяемые записи таблицы файлов: `fork()` и `dup()`......525
39.7. Безотлагательная запись с помощью `fsync()`......527
39.8. Переименование файлов......528
39.9. Получение информации о файлах......529
39.10. Удаление файлов......530
39.11. Создание каталога......530
39.12. Чтение каталогов......531
39.13. Удаление каталогов......532
39.14. Жесткие ссылки......533
39.15. Символические ссылки......534
39.16. Биты полномочий и списки контроля доступа......536
39.17. Создание и монтирование файловой системы......539

39.18. Резюме ... 541
Литература ... 541
Домашнее задание (код) ... 542
Вопросы ... 543

Глава 40. Реализация файловой системы ... 544
40.1. Ход мыслей ... 544
40.2. Общая организация ... 545
40.3. Организация файла: индексный дескриптор ... 548
Многоуровневый индекс ... 550
40.4. Организация каталогов ... 552
40.5. Управление свободным местом ... 553
40.6. Пути доступа: чтение и запись ... 554
Чтение файла с диска ... 554
Запись на диск ... 556
40.7. Кеширование и буферизация ... 558
40.8. Резюме ... 560
Литература ... 561
Домашнее задание (эмуляция) ... 562
Вопросы ... 562

Глава 41. Локальность и быстрая файловая система ... 563
41.1. Проблема: низкая производительность ... 563
41.2. FFS: решение – осведомленность о диске ... 565
41.3. Организационная структура: группа цилиндров ... 565
41.4. Политики: как выделять место для файлов и каталогов ... 567
41.5. Измерение локальности файлов ... 569
41.6. Исключение для больших файлов ... 571
41.7. Другие аспекты FFS ... 573
41.8. Резюме ... 575
Литература ... 575
Домашнее задание (эмуляция) ... 576
Вопросы ... 576

Глава 42. Согласованность после отказа: FSCK и журналирование ... 578
42.1. Подробный пример ... 579
Сценарии отказа ... 581
Проблема согласованности после отказа ... 582
42.2. Решение 1: средство проверки файловой системы ... 582
42.3. Решение 2: журналирование (или упреждающая запись в журнал) ... 584
Журналирование данных ... 585
Восстановление ... 588
Группировка обновлений журнала ... 589
Ограничение размера журнала ... 590
Журналирование метаданных ... 591

Интересный случай: повторное использование блока 593
Подводя итоги: хронология журналирования 594
42.4. Решение 3: другие подходы 595
42.5. Резюме 596
Литература 597
Домашнее задание (эмуляция) 599
Вопросы 599

Глава 43. Файловые системы со структурой журнала 600
43.1. Записывать на диск последовательно 601
43.2. Записывать последовательно и эффективно 602
43.3. Сколько буферизовать? 603
43.4. Проблема: нахождение индексных дескрипторов 604
43.5. Решение дает косвенность: карта индексных дескрипторов 605
43.6. Полное решение: область контрольной точки 606
43.7. Чтение файла с диска: повторение пройденного 607
43.8. А как насчет каталогов? 607
43.9. Новая проблема: сборка мусора 608
43.10. Нахождение живых блоков 610
43.11. Политика: какие блоки очищать и когда? 611
43.12. Структура журнала и восстановление после аварии 612
43.13. Резюме 613
Литература 614
Домашнее задание (эмуляция) 615
Вопросы 615

Глава 44. SSD-диски на основе флеш-памяти 618
44.1. Сохранение одного бита 619
44.2. От битов к банкам и плоскостям 619
44.3. Основные операции с флеш-памятью 620
Подробный пример 621
Резюме 622
44.4. Производительность и надежность флеш-памяти 622
44.5. От голой флеш-памяти к SSD на ее основе 624
44.6. Организация FTL: неправильный подход 625
44.7. FTL со структурой журнала 625
44.8. Сборка мусора 628
44.9. Размер таблицы отображения 631
Блочное отображение 631
Гибридное отображение 632
Страничное отображение плюс кеширование 635
44.10. Выравнивание износа 635
44.11. Производительность и стоимость SSD 636
Производительность 636
Стоимость 637
44.12. Резюме 638

Литература 639
Домашнее задание (эмуляция) 641
Вопросы 642

Глава 45. Целостность и защита данных 644
45.1. Виды отказа дисков 644
45.2. Обработка скрытых ошибок секторов 646
45.3. Обнаружение искажения: контрольная сумма 647
Распространенные функции вычисления контрольной суммы 648
Хранение контрольных сумм 649
45.4. Использование контрольных сумм 650
45.5. Новая проблема: запись не по адресу 651
45.6. Последняя проблема: потерянные записи 652
45.7. Очистка 653
45.8. Накладные расходы контрольных сумм 653
45.9. Резюме 654
Литература 654
Домашнее задание (эмуляция) 656
Вопросы 656
Домашнее задание (код) 657
Вопросы 657

Глава 46. Итоговый диалог о долговременном хранении 658

Глава 47. Диалог о распределенности 660

Глава 48. Распределенные системы 662
48.1. Основы коммуникации 663
48.2. Ненадежные уровни коммуникации 664
48.3. Надежные коммуникационные уровни 666
48.4. Абстракции коммуникации 669
48.5. Удаленный вызов процедур (RPC) 670
Генератор заглушек 670
Библиотека времени выполнения 672
Другие проблемы 673
48.6. Резюме 675
Литература 675
Домашнее задание (код) 676
Вопросы 676

Глава 49. Сетевая файловая система Sun (NFS) 678
49.1. Простая распределенная файловая система 679
49.2. Вперед к NFS 680
49.3. Акцент на простом и быстром восстановлении после аварии файлового сервера 680

49.4. Ключ к быстрому восстановлению: отсутствие информации о состоянии 681
49.5. Протокол NFSv2 682
49.6. От протокола к распределенной файловой системе 684
49.7. Обработка отказов сервера благодаря идемпотентным операциям 686
49.8. Повышение производительности: кеширование на стороне клиента 688
49.9. Проблема согласованности кешей 689
49.10. Оценка согласованности кешей в NFS 690
49.11. Последствия для буферизации записи на стороне сервера 691
49.12. Резюме 693
Литература 694
Домашнее задание (измерение) 695
Вопросы 695

Глава 50. Файловая система Andrew (AFS) 697
50.1. AFS версии 1 697
50.2. Проблемы версии 1 699
50.3. Улучшение протокола 700
50.4. AFS версии 2 700
50.5. Согласованность кешей 702
50.6. Восстановление после аварии 703
50.7. Масштабируемость и производительность AFSv2 705
50.8. AFS: другие усовершенствования 707
50.9. Резюме 708
Литература 708
Домашнее задание (эмуляция) 709
Вопросы 709

Глава 51. Заключительный диалог о распределенных файловых системах 711

Предметный указатель 713