

СОВРЕМЕННЫЕ Операционные СИСТЕМЫ

СОВРЕМЕННЫЕ ОПЕРАЦИОННЫЕ СИСТЕМЫ

4-е ИЗДАНИЕ



Э. ТАНИЕНБАУМ
Х. БОС

PH
PTR

ИИТЕР

Краткое содержание

| | |
|--|------|
| Предисловие | 17 |
| Глава 1. Введение | 22 |
| Глава 2. Процессы и потоки | 111 |
| Глава 3. Управление памятью | 214 |
| Глава 4. Файловые системы | 301 |
| Глава 5. Ввод и вывод информации | 380 |
| Глава 6. Взаимоблокировка | 488 |
| Глава 7. Виртуализация и облако | 527 |
| Глава 8. Многопроцессорные системы | 576 |
| Глава 9. Безопасность | 659 |
| Глава 10. Изучение конкретных примеров: Unix, Linux и Android | 784 |
| Глава 11. Изучение конкретных примеров: Windows 8 | 931 |
| Глава 12. Разработка операционных систем | 1058 |
| Глава 13. Библиография | 1110 |

Оглавление

| | |
|---|-----------|
| Предисловие | 17 |
| От издательства | 20 |
| Об авторах | 20 |
| Глава 1. Введение | 22 |
| 1.1. Что такое операционная система? | 24 |
| 1.1.1. Операционная система как расширенная машина | 24 |
| 1.1.2. Операционная система в качестве менеджера ресурсов | 26 |
| 1.2. История операционных систем | 27 |
| 1.2.1. Первое поколение (1945–1955): электронные лампы | 28 |
| 1.2.2. Второе поколение (1955–1965): транзисторы и системы пакетной обработки .. | 29 |
| 1.2.3. Третье поколение (1965–1980): интегральные схемы и многозадачность | 31 |
| 1.2.4. Четвертое поколение (с 1980 года по наши дни): персональные компьютеры ... | 36 |
| 1.2.5. Пятое поколение (с 1990 года по наши дни): мобильные компьютеры | 41 |
| 1.3. Обзор аппаратного обеспечения компьютера | 42 |
| 1.3.1. Процессоры | 43 |
| 1.3.2. Многопоточные и многоядерные микропроцессоры | 45 |
| 1.3.3. Память | 47 |
| 1.3.4. Диски | 50 |
| 1.3.5. Устройства ввода-вывода | 51 |
| 1.3.6. Шины | 55 |
| 1.3.7. Загрузка компьютера | 58 |
| 1.4. Зоопарк операционных систем | 59 |
| 1.4.1. Операционные системы мейнфреймов | 59 |
| 1.4.2. Серверные операционные системы | 60 |
| 1.4.3. Многопроцессорные операционные системы | 60 |
| 1.4.4. Операционные системы персональных компьютеров | 60 |
| 1.4.5. Операционные системы карманных персональных компьютеров | 61 |
| 1.4.6. Встроенные операционные системы | 61 |
| 1.4.7. Операционные системы сенсорных узлов | 61 |
| 1.4.8. Операционные системы реального времени | 62 |
| 1.4.9. Операционные системы смарт-карт | 63 |
| 1.5. Понятия операционной системы | 63 |
| 1.5.1. Процессы | 63 |
| 1.5.2. Адресные пространства | 66 |
| 1.5.3. Файлы | 66 |
| 1.5.4. Ввод-вывод данных | 69 |
| 1.5.5. Безопасность | 70 |
| 1.5.6. Оболочка | 70 |
| 1.5.7. Онтогенез повторяет филогенез | 72 |
| 1.6. Системные вызовы | 75 |
| 1.6.1. Системные вызовы для управления процессами | 80 |
| 1.6.2. Системные вызовы для управления файлами | 82 |
| 1.6.3. Системные вызовы для управления каталогами | 83 |

| | |
|--|-----|
| 1.6.4. Разные системные вызовы | 85 |
| 1.6.5. Windows Win32 API | 85 |
| 1.7. Структура операционной системы | 88 |
| 1.7.1. Монолитные системы | 88 |
| 1.7.2. Многоуровневые системы | 90 |
| 1.7.3. Микроядра | 91 |
| 1.7.4. Клиент-серверная модель | 94 |
| 1.7.5. Виртуальные машины | 94 |
| 1.7.6. Экзоядра | 99 |
| 1.8. Устройство мира согласно языку C | 99 |
| 1.8.1. Язык C | 99 |
| 1.8.2. Заголовочные файлы | 100 |
| 1.8.3. Большие программные проекты | 101 |
| 1.8.4. Модель времени выполнения | 102 |
| 1.9. Исследования в области операционных систем | 103 |
| 1.10. Краткое содержание остальных глав этой книги | 104 |
| 1.11. Единицы измерения | 105 |
| 1.12. Краткие выводы | 106 |
| Вопросы | 107 |

Глава 2. Процессы и потоки 111

| | |
|--|-----|
| 2.1. Процессы | 111 |
| 2.1.1. Модель процесса | 112 |
| 2.1.2. Создание процесса | 114 |
| 2.1.3. Завершение процесса | 116 |
| 2.1.4. Иерархии процессов | 117 |
| 2.1.5. Состояния процессов | 118 |
| 2.1.6. Реализация процессов | 120 |
| 2.1.7. Моделирование режима многозадачности | 122 |
| 2.2. Потоки | 123 |
| 2.2.1. Применение потоков | 124 |
| 2.2.2. Классическая модель потоков | 129 |
| 2.2.3. Потоки в POSIX | 133 |
| 2.2.4. Реализация потоков в пользовательском пространстве | 135 |
| 2.2.5. Реализация потоков в ядре | 138 |
| 2.2.6. Гибридная реализация | 140 |
| 2.2.7. Активация планировщика | 140 |
| 2.2.8. Всплывающие потоки | 142 |
| 2.2.9. Превращение однопоточного кода в многопоточный | 143 |
| 2.3. Взаимодействие процессов | 146 |
| 2.3.1. Состязательная ситуация | 147 |
| 2.3.2. Критические области | 148 |
| 2.3.3. Взаимное исключение с активным ожиданием | 150 |
| 2.3.4. Приостановка и активизация | 155 |
| 2.3.5. Семафоры | 158 |
| 2.3.6. Мьютексы | 161 |
| 2.3.7. Мониторы | 167 |
| 2.3.8. Передача сообщений | 172 |
| 2.3.9. Барьеры | 175 |
| 2.3.10. Работа без блокировок: чтение — копирование — обновление | 177 |
| 2.4. Планирование | 178 |
| 2.4.1. Введение в планирование | 179 |
| 2.4.2. Планирование в пакетных системах | 186 |

| | |
|---|------------|
| 2.4.3. Планирование в интерактивных системах | 188 |
| 2.4.4. Планирование в системах реального времени | 195 |
| 2.4.5. Политика и механизмы | 196 |
| 2.4.6. Планирование потоков | 197 |
| 2.5. Классические задачи взаимодействия процессов | 198 |
| 2.5.1. Задача обедающих философов | 199 |
| 2.5.2. Задача читателей и писателей | 202 |
| 2.6. Исследования, посвященные процессам и потокам | 203 |
| 2.7. Краткие выводы | 205 |
| Вопросы | 205 |
| Глава 3. Управление памятью | 214 |
| 3.1. Память без использования абстракций | 215 |
| 3.2. Абстракция памяти: адресные пространства | 218 |
| 3.2.1. Понятие адресного пространства | 218 |
| 3.2.2. Свопинг | 221 |
| 3.2.3. Управление свободной памятью | 224 |
| 3.3. Виртуальная память | 227 |
| 3.3.1. Страничная организация памяти | 229 |
| 3.3.2. Таблицы страниц | 233 |
| 3.3.3. Ускорение работы страничной организации памяти | 235 |
| 3.3.4. Таблицы страниц для больших объемов памяти | 239 |
| 3.4. Алгоритмы замещения страниц | 243 |
| 3.4.1. Оптимальный алгоритм замещения страниц | 244 |
| 3.4.2. Алгоритм исключения недавно использовавшейся страницы | 245 |
| 3.4.3. Алгоритм «первой пришла, первой и ушла» | 246 |
| 3.4.4. Алгоритм «второй шанс» | 246 |
| 3.4.5. Алгоритм «часы» | 247 |
| 3.4.6. Алгоритм замещения наименее востребованной страницы | 248 |
| 3.4.7. Моделирование LRU в программном обеспечении | 248 |
| 3.4.8. Алгоритм «рабочий набор» | 250 |
| 3.4.9. Алгоритм WSClock | 254 |
| 3.4.10. Краткая сравнительная характеристика алгоритмов замещения страниц | 256 |
| 3.5. Разработка систем страничной организации памяти | 257 |
| 3.5.1. Сравнительный анализ локальной и глобальной политики | 258 |
| 3.5.2. Управление загрузкой | 260 |
| 3.5.3. Размер страницы | 261 |
| 3.5.4. Разделение пространства команд и данных | 263 |
| 3.5.5. Совместно используемые страницы | 264 |
| 3.5.6. Совместно используемые библиотеки | 265 |
| 3.5.7. Отображаемые файлы | 268 |
| 3.5.8. Политика очистки страниц | 268 |
| 3.5.9. Интерфейс виртуальной памяти | 269 |
| 3.6. Проблемы реализации | 270 |
| 3.6.1. Участие операционной системы в процессе подкачки страниц | 270 |
| 3.6.2. Обработка ошибки отсутствия страницы | 271 |
| 3.6.3. Перезапуск команды | 272 |
| 3.6.4. Блокировка страниц в памяти | 273 |
| 3.6.5. Резервное хранилище | 274 |
| 3.6.6. Разделение политики и механизма | 276 |
| 3.7. Сегментация | 277 |
| 3.7.1. Реализация чистой сегментации | 281 |

| | |
|---|------------|
| 3.7.2. Сегментация со страничной организацией памяти: система MULTICS | 281 |
| 3.7.3. Сегментация со страничной организацией памяти: система Intel x86 | 285 |
| 3.8. Исследования в области управления памятью | 290 |
| 3.9. Краткие выводы | 290 |
| Вопросы | 291 |
| Глава 4. Файловые системы | 301 |
| 4.1. Файлы | 303 |
| 4.1.1. Имена файлов | 303 |
| 4.1.2. Структура файла | 305 |
| 4.1.3. Типы файлов | 306 |
| 4.1.4. Доступ к файлам | 308 |
| 4.1.5. Атрибуты файлов | 309 |
| 4.1.6. Операции с файлами | 310 |
| 4.1.7. Пример программы, использующей файловые системные вызовы | 312 |
| 4.2. Каталоги | 314 |
| 4.2.1. Системы с одноуровневыми каталогами | 315 |
| 4.2.2. Иерархические системы каталогов | 315 |
| 4.2.3. Имена файлов | 316 |
| 4.2.4. Операции с каталогами | 318 |
| 4.3. Реализация файловой системы | 319 |
| 4.3.1. Структура файловой системы | 320 |
| 4.3.2. Реализация файлов | 321 |
| 4.3.3. Реализация каталогов | 326 |
| 4.3.4. Совместно используемые файлы | 329 |
| 4.3.5. Файловые системы с журнальной структурой | 332 |
| 4.3.6. Журналируемые файловые системы | 334 |
| 4.3.7. Виртуальные файловые системы | 336 |
| 4.4. Управление файловой системой и ее оптимизация | 339 |
| 4.4.1. Управление дисковым пространством | 339 |
| 4.4.2. Резервное копирование файловой системы | 346 |
| 4.4.3. Непротиворечивость файловой системы | 352 |
| 4.4.4. Производительность файловой системы | 355 |
| 4.4.5. Дефрагментация дисков | 360 |
| 4.5. Примеры файловых систем | 361 |
| 4.5.1. Файловая система MS-DOS | 361 |
| 4.5.2. Файловая система UNIX V7 | 365 |
| 4.5.3. Файловые системы компакт-дисков | 367 |
| 4.6. Исследования в области файловых систем | 373 |
| 4.7. Краткие выводы | 374 |
| Вопросы | 374 |
| Глава 5. Ввод и вывод информации | 380 |
| 5.1. Основы аппаратного обеспечения ввода-вывода | 380 |
| 5.1.1. Устройства ввода-вывода | 381 |
| 5.1.2. Контроллеры устройств | 382 |
| 5.1.3. Ввод-вывод, отображаемый на пространство памяти | 383 |
| 5.1.4. Прямой доступ к памяти | 387 |
| 5.1.5. Еще раз о прерываниях | 391 |

| | |
|--|------------|
| 5.2. Принципы создания программного обеспечения ввода-вывода | 395 |
| 5.2.1. Задачи, стоящие перед программным обеспечением ввода-вывода | 395 |
| 5.2.2. Программный ввод-вывод | 397 |
| 5.2.3. Ввод-вывод, управляемый прерываниями | 398 |
| 5.2.4. Ввод-вывод с использованием DMA | 399 |
| 5.3. Уровни программного обеспечения ввода-вывода | 400 |
| 5.3.1. Обработчики прерываний | 400 |
| 5.3.2. Драйверы устройств | 402 |
| 5.3.3. Программное обеспечение ввода-вывода, не зависящее от конкретных устройств | 406 |
| 5.3.4. Программное обеспечение ввода-вывода, работающее в пространстве пользователя | 412 |
| 5.4. Диски | 414 |
| 5.4.1. Аппаратная часть дисков | 414 |
| 5.4.2. Форматирование диска | 421 |
| 5.4.3. Алгоритмы планирования перемещения блока головок | 425 |
| 5.4.4. Обработка ошибок | 429 |
| 5.4.5. Стабильное хранилище данных | 432 |
| 5.5. Часы | 435 |
| 5.5.1. Аппаратная составляющая часов | 436 |
| 5.5.2. Программное обеспечение часов | 437 |
| 5.5.3. Программируемые таймеры | 440 |
| 5.6. Пользовательский интерфейс: клавиатура, мышь, монитор | 442 |
| 5.6.1. Программное обеспечение ввода информации | 442 |
| 5.6.2. Программное обеспечение вывода информации | 448 |
| 5.7. Тонкие клиенты | 466 |
| 5.8. Управление энергопотреблением | 467 |
| 5.8.1. Роль оборудования | 468 |
| 5.8.2. Роль операционной системы | 470 |
| 5.8.2. Роль прикладных программ | 476 |
| 5.9. Исследования в области ввода-вывода данных | 477 |
| 5.10. Краткие выводы | 479 |
| Вопросы | 480 |
| Глава 6. Взаимоблокировка | 488 |
| 6.1. Ресурсы | 489 |
| 6.1.1. Выгружаемые и невыгружаемые ресурсы | 489 |
| 6.1.2. Получение ресурса | 490 |
| 6.2. Введение во взаимоблокировки | 492 |
| 6.2.1. Условия возникновения ресурсных взаимоблокировок | 493 |
| 6.2.2. Моделирование взаимоблокировок | 493 |
| 6.3. Страусиный алгоритм | 496 |
| 6.4. Обнаружение взаимоблокировок и восстановление работоспособности | 497 |
| 6.4.1. Обнаружение взаимоблокировки при использовании одного ресурса каждого типа | 497 |
| 6.4.2. Обнаружение взаимоблокировки при использовании нескольких ресурсов каждого типа | 499 |
| 6.4.3. Выход из взаимоблокировки | 502 |
| 6.5. Уклонение от взаимоблокировок | 504 |
| 6.5.1. Траектории ресурса | 504 |
| 6.5.2. Безопасное и небезопасное состояние | 506 |
| 6.5.3. Алгоритм банкира для одного ресурса | 507 |
| 6.5.4. Алгоритм банкира для нескольких типов ресурсов | 508 |

| | |
|---|------------|
| 6.6. Предотвращение взаимоблокировки | 510 |
| 6.6.1. Атака условия взаимного исключения | 510 |
| 6.6.2. Атака условия удержания и ожидания | 511 |
| 6.6.3. Атака условия невыгружаемости | 512 |
| 6.6.4. Атака условия циклического ожидания | 512 |
| 6.7. Другие вопросы | 513 |
| 6.7.1. Двухфазное блокирование | 513 |
| 6.7.2. Взаимные блокировки при обмене данными | 514 |
| 6.7.3. Активная взаимоблокировка | 516 |
| 6.7.4. Зависание | 518 |
| 6.8. Исследования в области взаимоблокировок | 518 |
| 6.9. Краткие выводы | 519 |
| Вопросы | 520 |
| Глава 7. Виртуализация и облако | 527 |
| 7.1. История | 529 |
| 7.2. Требования, применяемые к виртуализации | 530 |
| 7.3. Гипервизоры первого и второго типа | 533 |
| 7.4. Технологии эффективной виртуализации | 535 |
| 7.4.1. Виртуализация оборудования, не готового к виртуализации | 536 |
| 7.4.2. Цена виртуализации | 539 |
| 7.5. Являются ли гипервизоры настоящими микроядрами? | 540 |
| 7.6. Виртуализация памяти | 543 |
| 7.6.1. Аппаратная поддержка вложенных таблиц страниц | 545 |
| 7.6.2. Возвращение памяти | 546 |
| 7.7. Виртуализация ввода-вывода | 547 |
| 7.7.1. Блоки управления памятью при вводе-выводе | 548 |
| 7.7.2. Домены устройств | 549 |
| 7.7.3. Виртуализация ввода-вывода в отдельно взятом физическом устройстве | 550 |
| 7.8. Виртуальные устройства | 551 |
| 7.9. Виртуальные машины на мультиядерных центральных процессорах | 551 |
| 7.10. Вопросы лицензирования | 552 |
| 7.11. Облака | 553 |
| 7.11.1. Облака в качестве услуги | 554 |
| 7.11.2. Миграция виртуальных машин | 554 |
| 7.11.3. Установка контрольных точек | 555 |
| 7.12. Изучение конкретных примеров: VMWARE | 556 |
| 7.12.1. Ранняя история VMware | 556 |
| 7.12.2. VMware Workstation | 558 |
| 7.12.3. Сложности внедрения виртуализации в архитектуру x86 | 559 |
| 7.12.4. VMware Workstation: обзор решения | 560 |
| 7.12.5. Развитие VMware Workstation | 570 |
| 7.12.6. ESX-сервер: гипервизор первого типа компании VMware | 571 |
| 7.13. Исследования в области виртуализации и облаков | 573 |
| Вопросы | 573 |
| Глава 8. Многопроцессорные системы | 576 |
| 8.1. Мультипроцессоры | 579 |
| 8.1.1. Мультипроцессорное аппаратное обеспечение | 579 |
| 8.1.2. Типы мультипроцессорных операционных систем | 591 |

| | |
|--|------------|
| 8.1.3. Синхронизация мультипроцессоров | 595 |
| 8.1.4. Планирование работы мультипроцессора | 600 |
| 8.2. Мультикомпьютеры | 607 |
| 8.2.1. Аппаратное обеспечение мультикомпьютеров | 608 |
| 8.2.2. Низкоуровневые коммуникационные программы | 612 |
| 8.2.3. Коммуникационные программы пользовательского уровня | 615 |
| 8.2.4. Вызов удаленной процедуры | 619 |
| 8.2.5. Распределенная совместно используемая память | 621 |
| 8.2.6. Планирование мультикомпьютеров | 626 |
| 8.2.7. Балансировка нагрузки | 627 |
| 8.3. Распределенные системы | 630 |
| 8.3.1. Сетевое оборудование | 633 |
| 8.3.2. Сетевые службы и протоколы | 636 |
| 8.3.3. Связующее программное обеспечение на основе документа | 640 |
| 8.3.4. Связующее программное обеспечение на основе файловой системы | 641 |
| 8.3.5. Связующее программное обеспечение, основанное на объектах | 646 |
| 8.3.6. Связующее программное обеспечение, основанное на взаимодействии | 648 |
| 8.4. Исследования в области многопроцессорных систем | 651 |
| 8.5. Краткие выводы | 652 |
| Вопросы | 653 |
| Глава 9. Безопасность | 659 |
| 9.1. Внешние условия, требующие принятия дополнительных мер безопасности | 661 |
| 9.1.1. Угрозы | 662 |
| 9.1.2. Злоумышленники | 665 |
| 9.2. Безопасность операционных систем | 665 |
| 9.2.1. Можно ли создать защищенные системы? | 666 |
| 9.2.2. Высоконадежная вычислительная база | 667 |
| 9.3. Управление доступом к ресурсам | 669 |
| 9.3.1. Домены защиты | 669 |
| 9.3.2. Списки управления доступом | 671 |
| 9.3.3. Перечни возможностей | 674 |
| 9.4. Формальные модели систем безопасности | 677 |
| 9.4.1. Многоуровневая защита | 679 |
| 9.4.2. Тайные каналы | 682 |
| 9.5. Основы криптографии | 686 |
| 9.5.1. Шифрование с секретным ключом | 687 |
| 9.5.2. Шифрование с открытым ключом | 688 |
| 9.5.3. Односторонние функции | 689 |
| 9.5.4. Цифровые подписи | 689 |
| 9.5.5. Криптографические процессоры | 691 |
| 9.6. Аутентификация | 693 |
| 9.6.1. Слабые пароли | 695 |
| 9.6.2. Парольная защита в UNIX | 697 |
| 9.6.3. Одноразовые пароли | 698 |
| 9.6.4. Схема аутентификации «клик — отзыв» | 700 |
| 9.6.5. Аутентификация с использованием физического объекта | 700 |
| 9.6.6. Аутентификация с использованием биометрических данных | 703 |

| | |
|--|------------|
| 9.7. Взлом программного обеспечения | 706 |
| 9.7.1. Атаки, использующие переполнение буфера | 708 |
| 9.7.2. Атаки, использующие форматированную строку | 717 |
| 9.7.3. Указатели на несуществующие объекты | 720 |
| 9.7.4. Атаки, использующие разыменованное нулевое указателя | 721 |
| 9.7.5. Атаки, использующие переполнение целочисленных значений | 722 |
| 9.7.6. Атаки, использующие внедрение команд | 723 |
| 9.7.7 Атаки, проводимые с момента проверки до момента использования | 724 |
| 9.8. Инсайдерские атаки | 725 |
| 9.8.1. Логические бомбы | 725 |
| 9.8.2. Лазейки | 726 |
| 9.8.3. Фальсификация входа в систему | 727 |
| 9.9. Вредоносные программы | 728 |
| 9.9.1. Троянские кони | 731 |
| 9.9.2. Вирусы | 733 |
| 9.9.3. Черви | 743 |
| 9.9.4. Программы-шпионы | 745 |
| 9.9.5. Руткиты | 749 |
| 9.10. Средства защиты | 754 |
| 9.10.1. Брандмауэры | 754 |
| 9.10.2. Антивирусные и антиантивирусные технологии | 756 |
| 9.10.3. Электронная подпись двоичных программ | 763 |
| 9.10.4. Тюремное заключение | 764 |
| 9.10.5. Обнаружение проникновения на основе модели | 765 |
| 9.10.6. Инкапсулированный мобильный код | 767 |
| 9.10.7. Безопасность в системе Java | 771 |
| 9.11. Исследования в области безопасности | 773 |
| 9.12. Краткие выводы | 775 |
| Вопросы | 776 |
| Глава 10. Изучение конкретных примеров: Unix, Linux и Android | 784 |
| 10.1. История UNIX и Linux | 785 |
| 10.1.1. UNICS | 785 |
| 10.1.2. PDP-11 UNIX | 786 |
| 10.1.3. Переносимая система UNIX | 787 |
| 10.1.4. Berkeley UNIX | 788 |
| 10.1.5. Стандартная система UNIX | 789 |
| 10.1.6. MINIX | 790 |
| 10.1.7. Linux | 791 |
| 10.2. Обзор системы Linux | 794 |
| 10.2.1. Задачи Linux | 794 |
| 10.2.2. Интерфейсы системы Linux | 795 |
| 10.2.3. Оболочка | 797 |
| 10.2.4. Утилиты Linux | 800 |
| 10.2.5. Структура ядра | 802 |
| 10.3. Процессы в системе Linux | 804 |
| 10.3.1. Фундаментальные концепции | 805 |
| 10.3.2. Системные вызовы управления процессами в Linux | 807 |
| 10.3.3. Реализация процессов и потоков в Linux | 811 |
| 10.3.4. Планирование в Linux | 818 |
| 10.3.5. Загрузка Linux | 823 |

| | |
|---|------------|
| 10.4. Управление памятью в Linux | 826 |
| 10.4.1. Фундаментальные концепции | 826 |
| 10.4.2. Системные вызовы управления памятью в Linux | 829 |
| 10.4.3. Реализация управления памятью в Linux | 830 |
| 10.4.4. Подкачка в Linux | 837 |
| 10.5. Ввод-вывод в системе Linux | 840 |
| 10.5.1. Фундаментальные концепции | 840 |
| 10.5.2. Работа с сетью | 841 |
| 10.5.3. Системные вызовы ввода-вывода в Linux | 843 |
| 10.5.4. Реализация ввода-вывода в системе Linux | 844 |
| 10.5.5. Модули в Linux | 847 |
| 10.6. Файловая система UNIX | 848 |
| 10.6.1. Фундаментальные принципы | 848 |
| 10.6.2. Вызовы файловой системы в Linux | 853 |
| 10.6.3. Реализация файловой системы Linux | 857 |
| 10.6.4. Файловая система NFS | 866 |
| 10.7. Безопасность в Linux | 872 |
| 10.7.1. Фундаментальные концепции | 872 |
| 10.7.2. Системные вызовы безопасности в Linux | 875 |
| 10.7.3. Реализация безопасности в Linux | 875 |
| 10.8. Android | 876 |
| 10.8.1. Android и Google | 877 |
| 10.8.2. История Android | 878 |
| 10.8.3. Цели разработки | 881 |
| 10.8.4. Архитектура Android | 883 |
| 10.8.5. Расширения Linux | 885 |
| 10.8.6. Dalvik | 888 |
| 10.8.7. Binder IPC | 890 |
| 10.8.8. Приложения Android | 898 |
| 10.8.9. Намерения | 909 |
| 10.8.10. Песочницы приложений | 911 |
| 10.8.11. Безопасность | 912 |
| 10.8.12. Модель процесса | 918 |
| 10.9. Краткие выводы | 923 |
| Вопросы | 924 |
| Глава 11. Изучение конкретных примеров: Windows 8 | 931 |
| 11.1. История Windows вплоть до Windows 8.1 | 931 |
| 11.1.1. 80-е годы прошлого века: MS-DOS | 932 |
| 11.1.2. 90-е годы прошлого столетия: Windows на базе MS-DOS | 933 |
| 11.1.3. 2000 год: Windows на базе NT | 933 |
| 11.1.4. Windows Vista | 936 |
| 11.1.5 2010-е годы: Современная Windows | 937 |
| 11.2. Программирование в Windows | 938 |
| 11.2.1. Собственный интерфейс прикладного программирования NT | 942 |
| 11.2.2. Интерфейс прикладного программирования Win32 | 946 |
| 11.2.3. Реестр Windows | 950 |
| 11.3. Структура системы | 952 |
| 11.3.1. Структура операционной системы | 952 |
| 11.3.2. Загрузка Windows | 968 |
| 11.3.3. Реализация диспетчера объектов | 970 |
| 11.3.4. Подсистемы, DLL и службы пользовательского режима | 980 |

| | |
|---|-------------|
| 11.4. Процессы и потоки в Windows | 983 |
| 11.4.1. Фундаментальные концепции | 983 |
| 11.4.2. Вызовы API для управления заданиями, процессами, потоками и волокнами | 990 |
| 11.4.3. Реализация процессов и потоков | 996 |
| 11.5. Управление памятью | 1003 |
| 11.5.1. Фундаментальные концепции | 1004 |
| 11.5.2. Системные вызовы управления памятью | 1008 |
| 11.5.3. Реализация управления памятью | 1009 |
| 11.6. Кэширование в Windows | 1019 |
| 11.7. Ввод-вывод в Windows | 1020 |
| 11.7.1. Фундаментальные концепции | 1020 |
| 11.7.2. Вызовы интерфейса прикладного программирования ввода-вывода | 1022 |
| 11.7.3. Реализация ввода-вывода | 1024 |
| 11.8. Файловая система Windows NT | 1029 |
| 11.8.1. Фундаментальные концепции | 1029 |
| 11.8.2. Реализация файловой системы NTFS | 1030 |
| 11.9. Управление электропитанием в Windows | 1040 |
| 11.10. Безопасность в Windows 8 | 1042 |
| 11.10.1. Фундаментальные концепции | 1044 |
| 11.10.2. Вызовы интерфейса прикладного программирования безопасности | 1046 |
| 11.10.3. Реализация безопасности | 1046 |
| 11.10.4. Облегчение условий безопасности | 1049 |
| 11.11. Краткие выводы | 1052 |
| Вопросы | 1053 |
| Глава 12. Разработка операционных систем | 1058 |
| 12.1. Природа проблемы проектирования | 1058 |
| 12.1.1. Цели | 1058 |
| 12.1.2. Почему так сложно спроектировать операционную систему? | 1060 |
| 12.2. Разработка интерфейса | 1062 |
| 12.2.1. Руководящие принципы | 1062 |
| 12.2.2. Парадигмы | 1064 |
| 12.2.3. Интерфейс системных вызовов | 1068 |
| 12.3. Реализация | 1071 |
| 12.3.1. Структура системы | 1071 |
| 12.3.2. Механизм и политика | 1075 |
| 12.3.3. Ортогональность | 1076 |
| 12.3.4. Именованье | 1077 |
| 12.3.5. Время связывания | 1079 |
| 12.3.6. Статические и динамические структуры | 1080 |
| 12.3.7. Реализация системы сверху вниз и снизу вверх | 1081 |
| 12.3.8. Сравнение синхронного и асинхронного обмена данными | 1082 |
| 12.3.9. Полезные методы | 1083 |
| 12.4. Производительность | 1089 |
| 12.4.1. Почему операционные системы такие медленные? | 1089 |
| 12.4.2. Что следует оптимизировать? | 1090 |
| 12.4.3. Выбор между оптимизацией по скорости и по занимаемой памяти | 1091 |
| 12.4.4. Кэширование | 1094 |
| 12.4.5. Подсказки | 1095 |
| 12.4.6. Использование локальности | 1095 |
| 12.4.7. Оптимизируйте общий случай | 1096 |

| | |
|--|-------------|
| 12.5. Управление проектом | 1097 |
| 12.5.1. Мифический человек-месяц | 1097 |
| 12.5.2. Структура команды | 1098 |
| 12.5.3. Роль опыта | 1100 |
| 12.5.4. Панацеи нет | 1101 |
| 12.6. Тенденции в проектировании операционных систем | 1102 |
| 12.6.1. Виртуализация и облако | 1102 |
| 12.6.2. Многоядерные микропроцессоры | 1103 |
| 12.6.3. Операционные системы с большим адресным пространством | 1103 |
| 12.6.4. Беспрепятственный доступ к данным | 1104 |
| 12.6.5. Компьютеры с автономным питанием | 1105 |
| 12.6.6. Встроенные системы | 1105 |
| 12.7. Краткие выводы | 1106 |
| Вопросы | 1106 |
| Глава 13. Библиография | 1110 |
| 13.1. Дополнительная литература | 1110 |
| 13.1.1. Введение и общие труды | 1110 |
| 13.1.2. Процессы и потоки | 1111 |
| 13.1.3. Управление памятью | 1111 |
| 13.1.4. Файловые системы | 1112 |
| 13.1.5. Ввод-вывод | 1112 |
| 13.1.6. Взаимоблокировка | 1113 |
| 13.1.7. Виртуализация и облако | 1114 |
| 13.1.8. Многопроцессорные системы | 1114 |
| 13.1.9. Безопасность | 1115 |
| 13.1.10. Изучение конкретных примеров 1: Unix, Linux и Android | 1117 |
| 13.1.11. Изучение конкретных примеров 2: Windows 8 | 1118 |
| 13.1.12. Проектирование операционных систем | 1118 |
| 13.2. Алфавитный список литературы | 1119 |