

Оглавление

| | |
|---|----|
| Предисловие | 17 |
| Вступление | 19 |
| Целевая аудитория и необходимые предпосылки | 19 |
| Краткое содержание | 20 |
| Версии, рассмотренные в книге | 21 |
| Условные обозначения | 22 |
| Работа с примерами кода | 24 |
| Благодарности | 25 |
| От издательства | 26 |
| Глава 1. Введение и основополагающие концепции | 27 |
| Системное программирование | 27 |
| Зачем изучать системное программирование | 28 |
| Краеугольные камни системного программирования | 29 |
| Системные вызовы | 29 |
| Библиотека C | 30 |
| Компилятор C | 31 |
| API и ABI | 31 |
| API | 32 |
| ABI | 32 |
| Стандарты | 33 |
| История POSIX и SUS | 34 |
| Стандарты языка C | 34 |
| Linux и стандарты | 35 |
| Стандарты и эта книга | 36 |
| Концепции программирования в Linux | 37 |
| Файлы и файловая система | 37 |
| Процессы | 45 |
| Пользователи и группы | 47 |
| Права доступа | 48 |

| | |
|--|-----------|
| Сигналы | 49 |
| Межпроцессное взаимодействие | 50 |
| Заголовки | 50 |
| Обработка ошибок | 50 |
| Добро пожаловать в системное программирование | 53 |
| Глава 2. Файловый ввод-вывод | 54 |
| Открытие файлов | 55 |
| Системный вызов <code>open()</code> | 55 |
| Владельцы новых файлов | 58 |
| Права доступа новых файлов | 58 |
| Функция <code>creat()</code> | 60 |
| Возвращаемые значения и коды ошибок | 61 |
| Считывание с помощью <code>read()</code> | 61 |
| Возвращаемые значения | 62 |
| Считывание всех байтов | 63 |
| Неблокирующее считывание | 64 |
| Другие значения ошибок | 64 |
| Ограничения размера для <code>read()</code> | 65 |
| Запись с помощью <code>write()</code> | 65 |
| Случаи частичной записи | 66 |
| Режим дозаписи | 67 |
| Неблокирующая запись | 67 |
| Другие коды ошибок | 68 |
| Ограничения размера при использовании <code>write()</code> | 68 |
| Поведение <code>write()</code> | 68 |
| Синхронизированный ввод-вывод | 70 |
| <code>fsync()</code> и <code>fdatasync()</code> | 70 |
| <code>sync()</code> | 72 |
| Флаг <code>O_SYNC</code> | 73 |
| Флаги <code>O_DSYNC</code> и <code>O_RSYNC</code> | 74 |
| Непосредственный ввод-вывод | 74 |
| Закрытие файлов | 75 |
| Значения ошибок | 76 |
| Позиционирование с помощью <code>lseek()</code> | 76 |
| Поиск с выходом за пределы файла | 78 |
| Ограничения | 79 |
| Позиционное чтение и запись | 79 |
| Усечение файлов | 80 |

| | |
|---|------------|
| Мультиплексный ввод-вывод | 81 |
| select() | 83 |
| Системный вызов poll() | 88 |
| Сравнение poll() и select() | 92 |
| Внутренняя организация ядра | 93 |
| Виртуальная файловая система | 93 |
| Страничный кэш | 94 |
| Страничная отложенная запись | 96 |
| Резюме | 98 |
| Глава 3. Буферизованный ввод-вывод | 99 |
| Ввод-вывод с пользовательским буфером | 100 |
| Стандартный ввод-вывод. | 102 |
| Открытие файлов | 103 |
| Открытие потока данных с помощью файлового дескриптора | 104 |
| Закрытие потоков данных | 105 |
| Считывание из потока данных | 106 |
| Считывание одного символа в момент времени | 106 |
| Считывание целой строки | 107 |
| Считывание двоичных данных | 109 |
| Запись в поток данных | 110 |
| Запись отдельного символа | 110 |
| Запись строки символов. | 111 |
| Запись двоичных данных | 111 |
| Пример программы, в которой используется буферизованный ввод-вывод | 112 |
| Позиционирование в потоке данных | 113 |
| Сброс потока данных. | 115 |
| Ошибки и конец файла | 116 |
| Получение ассоциированного файлового дескриптора. | 117 |
| Управление буферизацией | 117 |
| Безопасность программных потоков. | 119 |
| Блокировка файлов вручную | 120 |
| Неблокируемые потоковые операции | 122 |
| Недостатки стандартного ввода-вывода. | 123 |
| Резюме | 123 |
| Глава 4. Расширенный файловый ввод-вывод | 125 |
| Фрагментированный ввод-вывод | 125 |
| Системные вызовы readv() и writev() | 126 |

| | |
|--|------------|
| Возвращаемые значения | 127 |
| Пример использования <code>writev()</code> | 128 |
| Пример использования <code>readv()</code> | 129 |
| Реализация | 130 |
| Опрос событий | 131 |
| Создание нового экземпляра <code>epoll</code> | 131 |
| Управление <code>epoll</code> | 132 |
| Ожидание событий с помощью <code>epoll</code> | 135 |
| Сравнение событий, запускаемых по фронту и по уровню сигнала | 136 |
| Отображение файлов в память | 137 |
| <code>mmap()</code> | 137 |
| Системный вызов <code>mmap()</code> | 142 |
| Пример отображения. | 143 |
| Преимущества <code>mmap()</code> | 144 |
| Недостатки <code>mmap()</code> | 145 |
| Изменение размеров отображения. | 145 |
| Изменение защиты отображения | 147 |
| Синхронизация файла с помощью отображения | 147 |
| Извещения об отображении. | 149 |
| Извещения об обычном файловом вводе-выводе | 151 |
| Системный вызов <code>posix_fadvise()</code> | 151 |
| Системный вызов <code>readahead()</code> | 153 |
| Рекомендации почти ничего не стоят. | 153 |
| Синхронизированные, синхронные и асинхронные операции | 154 |
| Планировщики и производительность ввода-вывода | 156 |
| Адресация диска | 156 |
| Жизненный цикл планировщика ввода-вывода. | 157 |
| Помощь при считывании | 158 |
| Выбор и настройка планировщика ввода-вывода | 162 |
| Оптимизация производительности ввода-вывода | 163 |
| Резюме | 170 |
| Глава 5. Управление процессами | 171 |
| Программы, процессы и потоки | 171 |
| Идентификатор процесса | 172 |
| Выделение идентификатора процесса | 173 |
| Иерархия процессов | 173 |
| <code>pid_t</code> | 174 |
| Получение идентификаторов процесса и родительского процесса | 174 |

| | |
|--|------------|
| Запуск нового процесса | 174 |
| Семейство вызовов <code>exec</code> | 175 |
| Системные вызовы <code>fork()</code> | 179 |
| Завершение процесса | 182 |
| Другие способы завершения | 183 |
| <code>atexit()</code> | 184 |
| <code>on_exit()</code> | 185 |
| <code>SIGCHLD</code> | 185 |
| Ожидание завершенных дочерних процессов | 185 |
| Ожидание определенного процесса | 188 |
| Еще больше гибкости при ожидании | 190 |
| На сцену выходит BSD: <code>wait3()</code> и <code>wait4()</code> | 192 |
| Запуск и ожидание нового процесса | 193 |
| Зомби | 195 |
| Пользователи и группы | 196 |
| Реальные, действительные и сохраненные идентификаторы пользователя и группы | 197 |
| Изменение реального или сохраненного идентификатора пользователя или группы | 198 |
| Изменение действительного идентификатора пользователя или группы | 199 |
| Изменение идентификаторов пользователя и группы согласно стилю BSD | 199 |
| Изменение идентификаторов пользователя и группы согласно стилю HP-UX | 200 |
| Действия с предпочтительными идентификаторами пользователя или группы | 201 |
| Поддержка сохраненных пользовательских идентификаторов | 201 |
| Получение идентификаторов пользователя и группы | 201 |
| Сессии и группы процессов | 202 |
| Системные вызовы сессий | 204 |
| Системные вызовы групп процессов | 205 |
| Устаревшие функции для группы процессов | 206 |
| Демоны | 207 |
| Резюме | 209 |
| Глава 6. Расширенное управление процессами | 210 |
| Планирование процессов | 210 |
| Кванты времени | 211 |
| Процессы ввода-вывода против ограниченных процессором | 212 |

| | |
|--|------------|
| Приоритетное планирование | 213 |
| Completely Fair Scheduler | 213 |
| Высвобождение ресурсов процессора | 215 |
| Приоритеты процессов | 216 |
| nice() | 217 |
| getpriority() и setpriority() | 218 |
| Приоритеты ввода-вывода | 219 |
| Привязка процессов к процессору | 220 |
| Системы реального времени | 223 |
| Мягкие и жесткие системы реального времени. | 224 |
| Задержка, колебание и временное ограничение. | 225 |
| Поддержка реального времени в Linux. | 226 |
| Политики планирования и приоритеты в Linux | 226 |
| Установка параметров планирования. | 230 |
| sched_rr_get_interval() | 233 |
| Предосторожности при работе с процессами реального времени | 235 |
| Детерминизм | 235 |
| Лимиты ресурсов. | 238 |
| Лимиты по умолчанию. | 242 |
| Установка и проверка лимитов. | 243 |
| Коды ошибок | 244 |
| Глава 7. Поточность | 245 |
| Бинарные модули, процессы и потоки | 245 |
| Многопоточность. | 246 |
| Издержки многопоточности | 248 |
| Альтернативы многопоточности. | 248 |
| Поточные модели | 249 |
| Поточность на уровне пользователя | 249 |
| Комбинированная поточность | 250 |
| Сопрограммы и фиберы. | 251 |
| Шаблоны поточности | 251 |
| Поток на соединение | 251 |
| Поток, управляемый событием. | 252 |
| Конкурентность, параллелизм и гонки. | 253 |
| Синхронизация | 256 |
| Мьютексы | 257 |
| Взаимные блокировки | 258 |
| P-потоки | 260 |

| | |
|---|------------|
| Реализация поточности в Linux | 261 |
| API для работы с P-потокami | 261 |
| Связывание P-потокoв | 262 |
| Создание потокoв | 262 |
| Идентификаторы потокoв | 264 |
| Завершение потокoв | 265 |
| Самозавершение | 265 |
| Завершение других потокoв | 266 |
| Присоединение и отсоединение потокoв | 268 |
| Присоединение потокoв | 268 |
| Отсоединение потокoв | 269 |
| Пример поточности | 269 |
| Мьютексы P-потокoв | 270 |
| Инициализация мьютексов | 270 |
| Запирание мьютексов | 271 |
| Отпирание мьютексов | 271 |
| Пример использования мьютексов | 272 |
| Дальнейшее изучение | 273 |
| Глава 8. Управление файлами и каталогами | 275 |
| Файлы и их метаданные | 275 |
| Семейство stat | 276 |
| Разрешения | 280 |
| Владение | 281 |
| Расширенные атрибуты | 284 |
| Перечисление расширенных атрибутов файла | 289 |
| Каталоги | 292 |
| Текущий рабочий каталог | 293 |
| Создание каталогов | 298 |
| Удаление каталогов | 299 |
| Чтение содержимого каталога | 300 |
| Ссылки | 303 |
| Жесткие ссылки | 304 |
| Символические ссылки | 305 |
| Удаление ссылки | 307 |
| Копирование и перемещение файлов | 308 |
| Копирование | 308 |
| Перемещение | 309 |
| Узлы устройств | 311 |

| | |
|---|------------|
| Специальные узлы устройств | 311 |
| Генератор случайных чисел | 312 |
| Внеполосное взаимодействие | 312 |
| Отслеживание файловых событий | 314 |
| Инициализация inotify | 315 |
| Стражи | 316 |
| События inotify | 317 |
| Расширенные события отслеживания | 321 |
| Удаление стража inotify | 321 |
| Получение размера очереди событий | 322 |
| Уничтожение экземпляра inotify | 323 |
| Глава 9. Управление памятью | 324 |
| Адресное пространство процесса | 324 |
| Страницы и их подкачка | 324 |
| Области памяти | 326 |
| Выделение динамической памяти | 327 |
| Выделение массивов | 329 |
| Изменение размера выделенных областей | 331 |
| Освобождение динамической памяти | 332 |
| Выравнивание | 335 |
| Управление сегментом данных | 339 |
| Анонимные отображения в памяти | 340 |
| Создание анонимных отображений в памяти | 342 |
| Отображение /dev/zero | 344 |
| Расширенное выделение памяти | 345 |
| Отладка при операциях выделения памяти | 348 |
| Выделение памяти на основе стека | 349 |
| Дублирование строк в стеке | 351 |
| Массивы переменной длины | 352 |
| Выбор механизма выделения памяти | 353 |
| Управление памятью | 354 |
| Установка байтов | 354 |
| Сравнение байтов | 355 |
| Перемещение байтов | 356 |
| Поиск байтов | 357 |
| Перещелкивание байтов | 358 |
| Блокировка памяти | 358 |
| Блокировка части адресного пространства | 359 |

| | |
|--|------------|
| Блокировка всего адресного пространства | 360 |
| Разблокировка памяти | 361 |
| Лимиты блокировки | 362 |
| Находится ли страница в физической памяти | 362 |
| Уступающее выделение памяти | 363 |
| Глава 10. Сигналы | 365 |
| Концепции, связанные с сигналами | 366 |
| Идентификаторы сигналов | 366 |
| Сигналы, поддерживаемые в Linux | 367 |
| Основы управления сигналами | 372 |
| Ожидание любого сигнала | 373 |
| Примеры | 374 |
| Выполнение и наследование | 376 |
| Сопоставление номеров сигналов и строк | 377 |
| Отправка сигнала | 378 |
| Права доступа | 378 |
| Примеры | 379 |
| Отправка сигнала самому себе | 379 |
| Отправка сигнала целой группе процессов | 380 |
| Реентерабельность | 380 |
| Наборы сигналов | 382 |
| Блокировка сигналов | 384 |
| Получение сигналов, ожидающих обработки | 385 |
| Ожидание набора сигналов | 385 |
| Расширенное управление сигналами | 385 |
| Структура <code>siginfo_t</code> | 388 |
| Удивительный мир <code>si_code</code> | 389 |
| Отправка сигнала с полезной нагрузкой | 391 |
| Изъян в UNIX? | 393 |
| Глава 11. Время | 394 |
| Структуры данных, связанные с представлением времени | 397 |
| Оригинальное представление | 397 |
| А теперь — с микросекундной точностью! | 397 |
| И еще лучше: наносекундная точность | 398 |
| Разбиение времени | 398 |
| Тип для процессного времени | 399 |

| | |
|---|------------|
| Часы POSIX | 400 |
| Получение текущего времени суток | 401 |
| Более удобный интерфейс | 402 |
| Продвинутый интерфейс | 403 |
| Получение процессного времени | 404 |
| Установка текущего времени суток | 405 |
| Установка времени с заданной точностью | 405 |
| Продвинутый интерфейс для установки времени | 406 |
| Эксперименты с временем | 406 |
| Настройка системных часов | 408 |
| Засыпание и ожидание | 411 |
| Засыпание с микросекундной точностью | 412 |
| Засыпание с наносекундной точностью | 413 |
| Продвинутая работа со спящим режимом | 415 |
| Переносимый способ засыпания | 416 |
| Превышение пределов | 417 |
| Альтернативы засыпанию | 418 |
| Таймеры | 418 |
| Простые варианты сигнализации | 418 |
| Интервальные таймеры | 419 |
| Функции для расширенной работы с таймерами | 421 |
| Приложение А. Расширения GCC для языка C | 427 |
| GNU C | 427 |
| Встраиваемые функции | 428 |
| Подавление встраивания | 429 |
| Чистые функции | 429 |
| Постоянные функции | 430 |
| Невозвращаемые функции | 430 |
| Функции, выделяющие память | 430 |
| Принудительная проверка возвращаемого значения вызывающей стороной | 431 |
| Как пометить функцию как устаревшую | 431 |
| Как пометить функцию как используемую | 431 |
| Как пометить функции или параметры как неиспользуемые | 432 |
| Упаковка структуры | 432 |
| Увеличение границы выравнивания переменной | 433 |
| Помещение глобальных переменных в регистр | 434 |

| | |
|--|------------|
| Аннотирование ветвей. | 434 |
| Получение типа выражения. | 435 |
| Получение границы выравнивания типа | 436 |
| Смещение члена внутри структуры | 437 |
| Получение возвращаемого адреса функции | 437 |
| Диапазоны оператора case | 438 |
| Арифметика указателей типа void и указателей на функции. | 438 |
| Более переносимо и красиво | 439 |
| Приложение Б. Библиография | 441 |
| Книги по языку программирования C | 441 |
| Книги по программированию в Linux | 442 |
| Книги, посвященные ядру Linux | 443 |
| Книги об организации операционных систем | 443 |