

Linux для разработчиков

Практическое углубленное руководство по командной строке
и утилитам Linux для разработчиков программного обеспечения

Дэвид Коэн
Кристиан Штурм

SPRINT
book

2025

Краткое содержание

| | |
|---|-----|
| Предисловие | 20 |
| Глава 1. Как устроена командная строка | 27 |
| Глава 2. Как работать с процессами | 51 |
| Глава 3. Как управлять службами с помощью <code>systemd</code> | 69 |
| Глава 4. История команд в оболочке Linux | 78 |
| Глава 5. Файлы в Linux | 84 |
| Глава 6. Как редактировать файлы из командной строки | 107 |
| Глава 7. Пользователи и группы | 119 |
| Глава 8. Владельцы ресурсов и права доступа | 131 |
| Глава 9. Как устанавливать программное обеспечение | 141 |
| Глава 10. Как настраивать конфигурацию приложений | 154 |
| Глава 11. Конвейеры и перенаправление ввода-вывода | 166 |
| Глава 12. Как автоматизировать задачи с помощью сценариев командной оболочки | 192 |
| Глава 13. Безопасный удаленный доступ по протоколу SSH | 210 |
| Глава 14. Как управлять версиями с помощью Git | 230 |
| Глава 15. Контейнерная виртуализация приложений с помощью Docker | 248 |
| Глава 16. Журналы приложений | 269 |
| Глава 17. Балансировка нагрузки и протокол HTTP | 284 |

Оглавление

| | |
|--|----|
| Создатели книги | 17 |
| Об авторах | 17 |
| О рецензентах | 18 |
| От издательства | 19 |
| О научном редакторе русского издания | 19 |
| Предисловие | 20 |
| Для кого эта книга | 20 |
| Чего нет в этой книге | 21 |
| О чем эта книга | 21 |
| Как извлечь максимальную пользу из книги | 25 |
| Условные обозначения | 25 |
| Глава 1. Как устроена командная строка | 27 |
| Вначале был REPL | 27 |
| Синтаксис командной строки (чтение) | 29 |
| Командная строка и командная оболочка | 31 |
| Каким образом оболочка узнаёт, какую программу запускать (вычисление) | 32 |
| Краткое введение в POSIX | 34 |
| Элементарные навыки работы в командной строке | 35 |
| Введение в файловую систему Unix | 35 |
| Абсолютные и относительные пути к файлам | 36 |
| Сведения о файлах и каталогах | 38 |
| Навигация по файловой системе | 40 |
| Как вносить изменения в файлы | 43 |
| Как пользоваться справочными страницами | 46 |
| Автозавершение команд | 48 |
| Итоги | 50 |

| | |
|---|----|
| Глава 2. Как работать с процессами | 51 |
| Введение в процессы Linux | 52 |
| Из чего состоит процесс Linux | 53 |
| Идентификатор процесса (Process ID, PID) | 55 |
| Эффективные идентификаторы пользователя (EUID) и группы (EGID) | 55 |
| Переменные окружения | 55 |
| Рабочий каталог | 56 |
| Команды для работы с процессами Linux | 56 |
| Продвинутое понятие и инструменты для работы с процессами | 59 |
| Сигналы | 59 |
| lsuf — дескрипторы файлов, которые открыл процесс | 64 |
| Наследование | 66 |
| Практическая работа: сеанс устранения неполадок | 66 |
| Итоги | 68 |
| Глава 3. Как управлять службами с помощью systemd | 69 |
| Введение в службы Linux | 70 |
| Подсистема инициализации | 72 |
| Процессы и службы | 72 |
| Команды systemctl | 73 |
| Как проверить состояние службы | 73 |
| Как запустить и остановить службу | 74 |
| Как перезагрузить службу | 75 |
| Как управлять автозапуском служб | 76 |
| Службы Linux и контейнеры Docker | 76 |
| Итоги | 77 |
| Глава 4. История команд в оболочке Linux | 78 |
| История команд в командной оболочке | 78 |
| Конфигурационные файлы оболочки | 79 |
| Файлы истории | 80 |
| Поиск по истории команд | 81 |
| Исключения поиска | 81 |
| Как запустить предыдущую команду с помощью восклицательного знака (!) | 81 |
| Как повторно запустить команду с прежними аргументами | 82 |
| Как включить предыдущую команду в состав текущей | 82 |

| | |
|---|------------|
| Как перейти в начало или конец текущей строки | 82 |
| Итоги | 83 |
| Глава 5. Файлы в Linux | 84 |
| Самое главное о файлах в Linux | 85 |
| Текстовые файлы | 85 |
| Двоичные файлы | 86 |
| Переводы строк | 86 |
| Дерево файловой системы | 87 |
| Основные операции с файлами и каталогами | 88 |
| ls — содержимое каталога | 89 |
| pwd — текущий каталог | 90 |
| cd — сменить текущий каталог | 90 |
| touch — создать пустой файл или обновить метки времени для существующего файла | 90 |
| less — постраничный просмотр файла | 91 |
| tail — просмотреть последние строки файла | 91 |
| mv — переместить или переименовать файл или каталог | 92 |
| cp — копировать файлы и каталоги | 92 |
| mkdir — создать каталог | 93 |
| rm — удалить файл или каталог | 93 |
| Как редактировать файлы | 93 |
| Типы файлов | 95 |
| Символические ссылки | 97 |
| Жесткие ссылки | 98 |
| Команда file | 98 |
| Продвинутые операции с файлами | 99 |
| Поиск по содержимому файлов с помощью grep | 99 |
| Как искать файлы с помощью find | 100 |
| Расширенные возможности файловой системы в реальной практике | 103 |
| FUSE: новые горизонты файловых систем Unix | 105 |
| Итоги | 106 |
| Глава 6. Как редактировать файлы из командной строки | 107 |
| Nano | 108 |
| Как установить nano | 109 |
| Шпаргалка по сочетаниям клавиш nano | 109 |

| | |
|---|------------|
| vi и Vim | 110 |
| Команды vi и Vim | 111 |
| Режимы | 111 |
| Как эффективно освоить vi или Vim | 114 |
| Настройки Vim в других программах | 116 |
| Как редактировать файл, если у вас недостаточно прав | 117 |
| Как настроить редактор по умолчанию | 117 |
| Итоги | 118 |
| Глава 7. Пользователи и группы | 119 |
| Что представляет собой пользователь | 119 |
| Root и остальные пользователи | 120 |
| sudo | 121 |
| Что такое группа | 123 |
| Мини-проект: как управлять пользователями и группами | 123 |
| Как создать пользователя | 124 |
| Как создать группу | 125 |
| Как управлять пользователями Linux | 126 |
| Дополнительный материал: что представляет собой пользователь на самом деле | 127 |
| Метаданные (атрибуты) пользователей | 128 |
| О сценариях командной оболочки | 129 |
| Итоги | 130 |
| Глава 8. Владельцы ресурсов и права доступа | 131 |
| Перечень файлов в расширенном формате | 131 |
| Атрибуты файлов | 132 |
| Тип файла | 132 |
| Права доступа (разрешения) | 133 |
| Количество жестких ссылок | 133 |
| Пользователь-владелец | 133 |
| Группа-владелец | 134 |
| Размер файла | 134 |
| Время последнего изменения | 134 |
| Имя файла | 135 |
| Владельцы файлов | 135 |

| | |
|---|-----|
| Права доступа (разрешения) | 135 |
| Права доступа в восьмеричном формате | 136 |
| Распространенные права доступа | 137 |
| Как сменить владельцев файла (<code>chown</code>) и редактировать права доступа к нему (<code>chmod</code>) | 138 |
| <code>Chown</code> — назначить владельцев файла | 138 |
| <code>Chmod</code> — задать права доступа к файлу | 139 |
| Итоги | 140 |
| Глава 9. Как устанавливать программное обеспечение | 141 |
| Как работать с системами управления пакетами | 142 |
| Как обновить локальный кэш пакетов | 143 |
| Как найти пакет | 144 |
| Как установить пакет | 144 |
| Как обновить все пакеты, для которых доступны обновления | 145 |
| Как удалить пакет (и все его зависимости, если они не нужны другим пакетам) | 145 |
| Как анализировать список установленных пакетов | 146 |
| Будьте осторожны с конвейером <code>curl bash</code> | 147 |
| Как компилировать стороннее ПО из исходного кода | 148 |
| Пример: компилируем и устанавливаем <code>htop</code> | 149 |
| Итоги | 153 |
| Глава 10. Как настраивать конфигурацию приложений | 154 |
| Иерархия конфигурации | 155 |
| Аргументы командной строки | 157 |
| Переменные окружения | 158 |
| Конфигурационные файлы | 161 |
| Конфигурация системного уровня в <code>/etc</code> | 161 |
| Конфигурация пользовательского уровня в <code>~/config</code> | 161 |
| Модули <code>systemd</code> | 162 |
| Как создать собственную службу | 162 |
| Дополнительно: конфигурация в <code>Docker</code> | 164 |
| Итоги | 165 |
| Глава 11. Конвейеры и перенаправление ввода-вывода | 166 |
| Файловые дескрипторы | 167 |
| На что ссылаются файловые дескрипторы | 168 |
| Перенаправление ввода-вывода | 168 |

| | |
|---|------------|
| Перенаправление ввода: < | 169 |
| Перенаправление вывода: > | 169 |
| Перенаправление ошибок: 2> | 171 |
| Как объединять команды с помощью конвейеров () | 172 |
| Многоступенчатые конвейеры | 173 |
| Полезные инструменты командной строки | 174 |
| cut — разбить строку на части | 174 |
| sort — отсортировать строки | 175 |
| uniq — обработать повторяющиеся строки | 175 |
| wc — подсчитать текстовые компоненты | 177 |
| head — вывести первые строки потока данных | 178 |
| tail — вывести последние строки потока данных | 178 |
| tee — копировать стандартный ввод в стандартный вывод и файл | 178 |
| awk — расширенная обработка входного потока | 179 |
| sed | 179 |
| Практические идиомы конвейеров | 180 |
| Как вывести «топ-Х» (рейтинг с подсчетом) | 180 |
| Как устанавливать ПО с помощью curl bash | 181 |
| Как фильтровать и искать данные с помощью grep | 183 |
| Как анализировать журналы с помощью grep и tail | 184 |
| Как обрабатывать множество файлов с помощью find и xargs | 184 |
| Как анализировать данные с помощью sort, uniq и числовой сортировки по убыванию | 185 |
| Как форматировать данные и обрабатывать поля записей с помощью awk и sort | 186 |
| Как редактировать файлы и создавать их резервные копии с помощью sed и tee | 187 |
| Как управлять процессами с помощью p s, grep, awk, xargs и kill | 187 |
| Как создавать сжатые резервные копии с помощью tar и gzip | 187 |
| Продвинутый материал: как анализировать файловые дескрипторы | 188 |
| Итоги | 190 |
| | |
| Глава 12. Как автоматизировать задачи с помощью сценариев командной оболочки | 192 |
| Зачем нужно умение писать сценарии для Bash | 193 |
| Основы языка сценариев Bash | 193 |
| Переменные | 194 |

| | |
|---|------------|
| Bash и другие командные оболочки | 195 |
| Шебанг и исполняемые текстовые файлы (они же сценарии) | 195 |
| Распространенные настройки Bash (параметры/аргументы) | 196 |
| /usr/bin/env | 197 |
| Специальные символы и экранирование | 198 |
| Подстановка команд | 199 |
| Проверка условий | 199 |
| Операторы проверки условий | 200 |
| [[Условия со строками и файлами]] | 200 |
| ((Условия с арифметическими выражениями)) | 201 |
| Условные инструкции: if/then/else | 202 |
| Циклы | 203 |
| Циклы в стиле C | 203 |
| for ... in | 203 |
| Цикл while | 204 |
| Экспорт переменных | 204 |
| Функции | 205 |
| Предпочитайте локальные переменные | 206 |
| Перенаправление ввода-вывода | 206 |
| Перенаправление ввода с помощью < | 207 |
| Перенаправление вывода с помощью > и >> | 207 |
| Как перенаправить стандартный поток ошибок и стандартный вывод с помощью 2>&1 | 207 |
| Интерполяция переменных с помощью \${ } | 208 |
| Ограничения сценариев командной оболочки | 208 |
| Итоги | 209 |
| Глава 13. Безопасный удаленный доступ по протоколу SSH | 210 |
| Введение в криптографические системы с открытым ключом | 211 |
| Как шифровать сообщения | 211 |
| Как подписывать сообщения | 212 |
| Ключи SSH | 212 |
| Когда можно распространять закрытые ключи | 214 |
| Аутентификация по SSH | 214 |
| Практическая работа: настраиваем подключение к удаленному серверу с помощью ключей | 215 |

| | |
|--|------------|
| Шаг 1. Откройте терминал на стороне клиента (не сервера) SSH | 215 |
| Шаг 2. Создайте пару ключей | 215 |
| Шаг 3. Скопируйте открытый ключ на сервер | 216 |
| Шаг 4. Проверьте, что все работает | 216 |
| Как преобразовывать ключи SSH2 в формат OpenSSH | 216 |
| Чего мы хотим добиться | 217 |
| Как преобразовать открытый ключ из формата SSH2 в OpenSSH | 217 |
| Как преобразовать открытый ключ из формата OpenSSH в SSH2 | 218 |
| Агент SSH | 218 |
| Распространенные ошибки SSH и вывод диагностических сообщений с флагом -v | 220 |
| Передача файлов | 222 |
| SFTP | 222 |
| SCP | 223 |
| Продвинутые примеры | 224 |
| Туннелирование | 226 |
| Переадресация на локальный порт | 226 |
| Проксирование | 226 |
| Конфигурационные файлы SSH | 227 |
| Итоги | 228 |
| Глава 14. Как управлять версиями с помощью Git | 230 |
| Git: немного истории | 231 |
| Что такое распределенная система управления версиями | 231 |
| Основы работы с Git | 232 |
| Как настроить Git с самого начала | 232 |
| Как инициализировать новый репозиторий Git | 232 |
| Как вносить изменения и просматривать их | 233 |
| Как индексировать и фиксировать изменения | 233 |
| Дополнительно: как добавить удаленный репозиторий Git | 233 |
| Как передавать и принимать изменения | 234 |
| Как клонировать репозиторий | 234 |
| Термины и понятия Git | 235 |
| Репозиторий | 235 |
| Ветки | 235 |

| | |
|--|------------|
| Теги | 236 |
| Слияние | 236 |
| Конфликт слияния | 237 |
| Буфер | 237 |
| Запрос на принятие изменений | 237 |
| Выборочное извлечение коммитов | 238 |
| Бисекция | 238 |
| Перебазирование | 239 |
| Как писать комментарии к коммитам | 241 |
| Как писать хорошие комментарии | 242 |
| Системы с графическим интерфейсом | 243 |
| Полезные псевдонимы оболочки | 243 |
| GitHub на коленке | 244 |
| Общие соображения | 244 |
| Шаг 1. Подключитесь к серверу | 244 |
| Шаг 2. Установите Git | 245 |
| Шаг 3. Инициализируйте репозиторий | 245 |
| Шаг 4. Клонировать репозиторий | 245 |
| Шаг 5. Внесите изменения в проект и выгрузите их | 246 |
| Итого | 246 |
| Глава 15. Контейнерная виртуализация приложений с помощью Docker | 248 |
| Как контейнеры выступают в качестве пакетов | 249 |
| Как установить Docker | 250 |
| Введение в Docker | 250 |
| Как создавать образы с помощью Docker-файлов | 253 |
| Команды для управления контейнерами | 257 |
| docker run | 257 |
| docker ps | 258 |
| docker exec | 258 |
| docker stop | 259 |
| Проект с использованием Docker: контейнер для приложения на Python/Flask | 259 |
| 1. Как создать приложение | 259 |
| 2. Как создать образ Docker | 261 |
| 3. Как развернуть контейнер из образа | 261 |

| | |
|--|------------|
| Чем контейнеры отличаются от виртуальных машин | 263 |
| Репозитории образов Docker | 264 |
| Контейнеры: советы от наученных горьким опытом | 264 |
| Не используйте большие образы | 264 |
| Обращайте внимание на стандартную библиотеку C | 265 |
| Продакшен-среда — не ваш компьютер: не полагайтесь на внешние зависимости | 265 |
| Немного теории: контейнеры и пространства имен | 266 |
| Как контейнеры участвуют в операциях технического обслуживания | 267 |
| Итоги | 267 |
| Глава 16. Журналы приложений | 269 |
| Введение в протоколирование и журналы в Linux | 270 |
| Журналы в Linux: не все так просто | 271 |
| Как отправлять сообщения в журнал | 272 |
| Журнал подсистемы инициализации systemd | 272 |
| Некоторые команды journalctl | 273 |
| Как выводить активные записи журнала для модуля в реальном времени | 273 |
| Как фильтровать журналы по времени | 273 |
| Как фильтровать журналы по определенному уровню протоколирования | 274 |
| Как анализировать журналы от предыдущих загрузок | 275 |
| Сообщения ядра | 275 |
| Журналы в контейнерах Docker | 275 |
| Введение в Syslog | 276 |
| Категории | 277 |
| Уровни важности сообщений | 278 |
| Конфигурация и реализации | 279 |
| Как вести журналы | 279 |
| Тщательно выбирайте ключевые слова при структурированном протоколировании | 279 |
| Уровень важности | 280 |
| Централизованные журналы | 280 |
| Итоги | 282 |

| | |
|---|-----|
| Глава 17. Балансировка нагрузки и протокол HTTP | 284 |
| Основные понятия | 286 |
| Шлюз | 286 |
| Восходящий узел | 286 |
| Распространенные заблуждения об HTTP | 287 |
| Состояния HTTP | 287 |
| Заголовки HTTP | 290 |
| Версии HTTP | 291 |
| Балансировка нагрузки | 295 |
| CORS | 301 |
| Итоги | 303 |