

# Как освоить современный linux

Полный справочник: от новичка  
до профессионала

Майкл Хаузенблас

УДК 004.4  
ББК 32.973.26-018.2

X26

Michael Hausenblas  
Learning Modern Linux: A Handbook for the Cloud Native Practitioner

© 2026 “Astana International Publishing” LLP Authorized Russian translation of the English edition of Learning Modern Linux ISBN 9781098108946

© 2022 Michael Hausenblas. This translation is published and sold by permission of O’Reilly Media, Inc., which owns or controls all rights to publish and sell the samee.

**Хаузенблас, Майкл.**

X26 Как освоить современный Linux. Полный справочник: от новичка до профессионала / Майкл Хаузенблас : [перевод с английского А. Пескова]. — Алматы : Астана иностранная пресса, 2026. — 304 с. — (O’Reilly. Книги по программированию).

ISBN 978-601-12-4679-8

Linux — это универсальная рабочая среда для разработчиков, архитекторов и специалистов по облачным технологиям.

Автор выстраивает ясный и последовательный маршрут: от понимания ключевых компонентов — ядра, оболочек, файловых систем — до уверенной работы с сетевым стеком, контейнерами и инструментами мониторинга. Особое внимание уделено безопасности, управлению зависимостями и организации процессов, что делает книгу ценным источником знаний для специалистов, которым важны надежность и масштабируемость решений.

Это не сухая энциклопедия команд, а практическое руководство, которое показывает, как Linux может стать основой продуктивной разработки и эксплуатации современных систем.

УДК 004.4  
ББК 32.973.26-018.2

ISBN 978-601-12-4679-8

© Песков А. Ю., перевод на русский язык, 2026  
© Издание на русском языке, оформление.  
ТОО «Издательство «Астана иностранная пресса», 2026

---

# Оглавление

<b>Предисловие</b> .....	9
Для кого эта книга .....	10
Как пользоваться книгой .....	10
Условные обозначения .....	11
Использование примеров кода .....	12
Благодарности .....	13
<b>Глава 1. Введение в Linux</b> .....	14
Что такое современные среды? .....	14
История Linux (далекая от завершения) .....	16
Так зачем же нужна операционная система? .....	17
Дистрибутивы Linux .....	19
Видимость ресурсов .....	19
Linux с высоты птичьего полета .....	22
Заключение .....	23
<b>Глава 2. Ядро Linux</b> .....	25
Архитектура Linux .....	25
Архитектуры процессоров .....	27
Компоненты ядра .....	31
Расширения ядра .....	43
Заключение .....	47
<b>Глава 3. Оболочки и скрипты</b> .....	50
Основы .....	51
Удобные для работы оболочки .....	69
Терминальный мультиплексор .....	78
Сценарии .....	85
Заключение .....	94
<b>Глава 4. Управление доступом</b> .....	97
Основы .....	97
Пользователи .....	100

Разрешения .....	106
Расширенное управление разрешениями .....	113
Полезные советы .....	116
Заключение .....	117
<b>Глава 5. Файловые системы</b> .....	<b>119</b>
Основы .....	120
Виртуальная файловая система .....	124
Псевдофайловые системы .....	132
Обычные файлы .....	137
Заключение .....	143
<b>Глава 6. Приложения, управление пакетами и контейнеры</b> .....	<b>145</b>
Основы .....	146
systemd .....	149
Цепочки поставок приложений в Linux .....	155
Пакеты и менеджеры пакетов .....	158
Контейнеры .....	164
Современные менеджеры пакетов .....	180
Заключение .....	181
<b>Глава 7. Сетевое взаимодействие</b> .....	<b>183</b>
Основы .....	183
Стек TCP/IP .....	185
DNS .....	206
Сетевое взаимодействие на прикладном уровне .....	215
Дополнительная информация о сетевом взаимодействии .....	226
Заключение .....	231
<b>Глава 8. Наблюдаемость</b> .....	<b>233</b>
Основы .....	234
Журналирование .....	238
Наблюдение (мониторинг) .....	245
Углубимся в наблюдаемость .....	254
Заключение .....	261
<b>Глава 9. Углубленные темы</b> .....	<b>263</b>
Межпроцессное взаимодействие .....	263
Виртуальные машины .....	268

Современные дистрибутивы Linux .....	271
Избранное в области безопасности .....	274
Другие современные и перспективные возможности .....	275
Заключение .....	278
<b>Приложение А. Полезные примеры .....</b>	<b>281</b>
Сбор системной информации .....	281
Работа с пользователями и процессами .....	282
Сбор информации о файлах .....	283
Работа с файлами и каталогами .....	283
Работа с перенаправлением и конвейерами .....	284
Работа с датой и временем .....	285
Работа с Git .....	286
Производительность системы .....	287
<b>Приложение Б. Инструментарий современной Linux .....</b>	<b>288</b>
<b>Об авторе .....</b>	<b>290</b>
<b>Животное на обложке .....</b>	<b>291</b>
<b>Алфавитный указатель .....</b>	<b>292</b>



---

# Предисловие

Добро пожаловать на страницы книги «Как освоить современный Linux»! С большой радостью составлю вам компанию в недалгом совместном путешествии. Эта книга окажется вам полезной, если вы только познакомились с Linux и искали упорядоченное практическое руководство для более глубокого изучения, или вы опытный пользователь, но хотите подсказок и советов, чтобы отточить навыки работы с этой операционной системой для разработки или профессиональной деятельности.

Мы сосредоточимся на повседневном использовании Linux — от написания скриптов до решения офисных задач, — оставив в стороне вопросы администрирования. Графическому интерфейсу мы предпочтем командную строку. И пусть 2022 год считается годом настольного Linux, но, избрав терминал средством общения с операционной системой, вы получите дополнительное преимущество: вы сможете применить знания в самых различных конфигурациях, включая Raspberry Pi и виртуальные машины облачного провайдера.

Прежде чем мы начнем, я поделюсь с вами этапами собственного развития: первой операционной системой, с которой я познакомился, был не Linux. В конце 80-х годов прошлого века мне довелось поработать с AmigaOS, а позже, в техническом колледже, я использовал MS DOS и свежевывшедшую Microsoft Windows для создания событийных систем и графических интерфейсов. В университетской лаборатории в конце 90-х мы использовали основанную на Unix операционную систему Solaris и рабочие станции Silicon Graphics. Полноценное знакомство с Linux состоялось лишь во втором десятилетии 2000-х, когда я занялся статистической обработкой больших данных (bigdata) и стал применять в работе контейнеры: сначала Apache Mesos (в 2015, когда я работал в Mesosphere), а затем Kubernetes (в составе команды OpenShift в Red Hat, а позже в команде разработки контейнерных сервисов в AWS). Именно тогда я понял, что добиться успеха в моей области можно только освоив Linux. Он особенный. Он уникален своей гибкостью, универсальностью, историей развития и мировой поддержкой сообщества.

У Linux интересная и растущая экосистема разработчиков и поставщиков ПО с открытым исходным кодом. Эту операционную систему можно запустить на чем угодно, от Raspberry Pi за 50 долларов до виртуальных машин вашего провайдера, и даже на марсоходе. За 30 лет разработки Linux ничуть не сдал позиций, так что самое время взглянуть на него поближе.

Для начала выделим несколько правил и условностей. В предисловии я расскажу, как извлечь из книги максимальную пользу, и затрону организационные вопросы, касающихся того, где и как можно попрактиковаться в темах, которые мы будем изучать сообща.

## Для кого эта книга

Эта книга пригодится тем, кто хочет или должен использовать Linux профессионально: разработчикам, архитекторам ПО, инженерам по тестированию, DevOps, SRE и другим похожим специалистам. Если вы любитель и столкнулись с Linux при решении задач, связанных с 3D печатью или обустройством умного дома, то, вероятно, мало знаете об операционных системах вообще, и семействе UNIX/Linux в частности. В таком случае вам стоит прочитать книгу от начала и до конца: главы, как правило, последовательно дополняют друг друга. Если же вы знакомы с Linux, эта книга заменит хороший справочник.

## Как пользоваться книгой

Этот курс скорее научит вас работать с Linux, чем администрировать его. Вопросы администрирования затронуты в огромном количестве других прекрасных книг.

Прочитав эту книгу, вы узнаете, что такое Linux (глава 1), и из каких компонентов она состоит (главы 2 и 3). Вы освоите механизмы управления доступом (глава 4), оцените роль файловых систем как основных кирпичиков Linux (глава 5) и поймете, что такое приложения (глава 6).

Вы также получите практический навык работы с сетевым стеком и его инструментами (глава 7), ознакомитесь с современными возможностями наблюдения за ОС (глава 8) и научитесь управлять рабочей нагрузкой.

Вы узнаете о современных способах запуска приложений Linux в контейнерах или в неизменяемых дистрибутивах (таких как Bottlerocket), сможете безопасно работать с сетью, скачивать файлы или обмениваться данными через «защищенную оболочку» (Secure Shell, или SSH), познакомитесь с расширенным инструментарием для одноранговой (peer-to-peer, или p2p) и облачной синхронизации (глава 9).

Ниже я дам несколько советов по практической подготовке к процессу обучения (и рекомендую вам не пренебрегать ими, поскольку освоение Linux очень похоже на изучение иностранного языка — чем больше практики, тем лучше).

- Найдите компьютер или ноутбук. У меня, к примеру, замечательная машинка с названием StarBook от компании StarLabs. Можно взять ноутбук с устаревшей версией Windows, которая больше не обновляется, и установить Linux на него.
- Если вы хотите запустить Linux внутри другой операционной среды — скажем, у вас MacBook или iMac, — используйте виртуальную машину (VM)<sup>1</sup>. Для macOS есть замечательная Linux-on-Mac<sup>2</sup>.
- Можно развернуть виртуальную машину на базе Linux в сети у вашего облачного провайдера.
- Если вы любите мастерить и хотите попробовать процессор с архитектурой, отличной от Intel (к примеру, ARM), можете купить одноплатный компьютер. Советую замечательный Raspberry Pi.

В любом случае полезно не просто читать, а иметь под рукой настроенную среду и практиковаться. Выполняйте команды и экспериментируйте. Пробуйте «сломать» программу, подавая на вход странные данные. Перед запуском команды попробуйте представить, какой получите результат.

И еще совет: всегда задавайте вопросы и докапывайтесь до сути. Глядя на команду или результат выполнения, подумайте, как он был сформирован и какой базовый компонент за это отвечал.

## Условные обозначения

В этой книге используется следующая типографская маркировка:

### *Курсив*

*Отмечает новые термины, адреса в интернете, имена файлов и их расширения.*

### **Блок фиксированной ширины**

Используется для отображения программного кода, а также внутри параграфов, где упоминаются определенные программные элементы: переменные, названия функций, базы данных, типы данных, переменные окружения, операторы, константы и ключевые слова.

---

<sup>1</sup> Для современных компьютеров на базе Windows можно развернуть виртуальную машину Linux в VirtualBox, либо установить WS. (*Прим. ред.*)

<sup>2</sup> Репозиторий Lima (Linux-on-Mac) на Github: <https://github.com/lima-vm/lima>.

### *Курсив в блоках с фиксированной шириной*

Отмечает текст, который следует заменить на ваше или подстановочное значение.



Этот элемент означает подсказку или предложение.



Этот элемент означает общее примечание.



Этот элемент означает предупреждение или предостережение.

## Использование примеров кода

Сопроводительные материалы (в том числе примеры программного кода и упражнения) доступны для загрузки по адресу: <https://oreil.ly/learning-modern-linux-code>.

Если у вас возникли технические вопросы или затруднения при использовании примеров кода, пожалуйста, свяжитесь с нами по электронному адресу: [mbookquestions@oreilly.com](mailto:mbookquestions@oreilly.com).

Эта книга должна помочь вам в работе. Примеры из нее можно использовать в вашей программе или документации. Вам не нужно запрашивать разрешения до тех пор, пока вы не воспроизводите значительную часть кода. К примеру, если вы написали программу, включив в нее код из книги, разрешений не требуется. А для продажи и распространения примеров из книг издательства O'Reilly разрешения нужны. Ответ на вопрос путем цитирования выдержек из книги или примеров не требует разрешений. А на включение значительного объема примеров из этой книги в вашу документацию нужно их получить.

## Благодарности

Прежде всего я хочу поблагодарить замечательных рецензентов: Криса Негуса, Джона Бонесио и Павла Крупю. Без их отзывов книга была бы и вполнину не так хороша и полезна.

Также благодарю родителей, которые дали мне образование, заложив основы моего характера и профессии. Огромное спасибо моей старшей сестре Монике за то, что вдохновила пойти в IT.

Выражаю глубочайшую признательность своей замечательной семье за поддержку: детям — Сапфире, Ранье и Яннису; моей невероятно умной и веселой жене Аннелизе; Снупи — лучшей из всех собак, и коту Чарли, новому члену нашей семьи.

На пути знакомства с Unix и Linux я встретил огромное количество людей, оказавших влияние на мое мышление и многому меня научивших. С некоторыми мне выпала честь и счастье работать и плотно общаться (и перечислены далеко не все): Жером Петацони, Джесси Фразель, Брендан Грегг, Джастин Гаррисон, Макл Керриск и Дуглас Макилрой.

И последнее, но не менее важное: я благодарю всю команду издательства O'Reilly, в особенности моего технического редактора, Джеффа Блейла, за опеку и помощь в написании книги.

---

# Введение в Linux

Linux — одна из самых популярных операционных систем, используемых везде и повсюду, от мобильных устройств до облачных вычислений.

Возможно, вы не знакомы с концепцией ОС или работаете в Microsoft Windows, не вдаваясь в детали. Возможно, вы новичок в Linux. Чтобы настроить вас на нужный лад и дать некоторое представление об операционных системах, в том числе и Linux, мы начнем издаleка.

Для начала обсудим, что значит слово «современная» в названии книги, и приглядимся к общей предыстории Linux с важными вехами ее развития за последние 30 лет. Затем в этой главе вы узнаете, какова роль ОС в целом и как с ней справляется Linux. В заключение рассмотрим, что такое дистрибутивы Linux и что означает термин «видимость ресурсов».

Если операционные системы и Linux для вас в новинку, прочитайте главу от начала и до конца. Если вы знакомы с Linux, можете сразу перейти к разделу «Linux с высоты птичьего полета» и рисунку, сопоставляющему архитектуру Linux с главами книги.

Перед тем как углубиться в технические детали, отступим на шаг и подумаем о том, что понимается под выражением «современная Linux». Как ни странно, это не такой уж банальный вопрос.

## Что такое современные среды?

Так что кроется за словом «современная» в названии книги? Ну что же, в здешнем контексте — все что угодно, от облачных вычислений до Raspberry Pi. А недавний взлет платформы контейнеризации Docker и связанных с ней инноваций кардинально изменил расстановку сил для разработчиков и связанных с инфраструктурой инженеров.

Давайте же подробнее рассмотрим некоторые из современных сред, и место, которое занимает в них Linux.

### *Мобильные устройства*

Когда я упоминаю «мобильный телефон» при детях, они спрашивают: «А что, бывают другие?» В наши дни большинство устройств — больше 80%, в зависимости от того, у кого вы спросите, — это телефоны и планшеты на базе Android, системы, которая, в свою очередь, является вариацией Linux<sup>3</sup>. К мобильным средам выдвигаются самые жесткие требования по энергопотреблению и надежности, ведь мы зависим от них постоянно. Если вам интересно создание приложений для Android, посетите специализированные ресурсы для разработчиков Android<sup>4</sup>.

### *Облачные вычисления*

Ту же картину, что с мобильными и микросредами, только в более крупном масштабе, мы наблюдаем и в облаке. Здесь наряду с новыми, мощными, безопасными, энергосберегающими процессорными архитектурами, такими как Graviton на базе ARM (крайне успешное предложение от AWS), обитают устоявшиеся и зарекомендовавшие себя продукты по делегированию тяжелых вычислений облачным провайдерам, особенно в рамках ПО с открытым исходным кодом.

### *Интернет («умных») вещей*

Вам наверняка встречались проекты и продукты, связанные с интернетом вещей (Internet of Things, или IoT), от всевозможных датчиков до дронов. Сегодня мало кого удивляют умные домашние приборы и даже автомобили. В средах интернета вещей требования к энергопотреблению еще жестче, чем у мобильных устройств. Очень часто умные устройства должны работать не постоянно, а включаться по расписанию только раз в день, чтобы передать порцию данных. Однако и способность работать в реальном времени<sup>5</sup> — немаловажный фактор для этих сред. Если вам интересно попробовать Linux в связке с IoT, обратите внимание на IoT EduKit<sup>6</sup>, который предлагает AWS.

### *Разнообразие процессорных архитектур*

Очень долгое время, порядка 30 лет, Intel была ведущим производителем центральных процессоров (ЦП, CPU). Она доминировала на рынке персональных

---

<sup>3</sup> Is Android really just Linux? <https://www.androidauthority.com/android-linux-784964>.

<sup>4</sup> Сайт разработчиков на Android: <https://developer.android.com/?hl=ru>.

<sup>5</sup> Real-TimeLinux (Linux в реальном времени): <https://developer.toradex.com/software/real-time/real-time-linux/>.

<sup>6</sup> IoT EduKit отAWS: <https://aws.amazon.com/ru/iot/edukit/content>.

и микрокомпьютеров, а архитектура Intel x86 считалась золотым стандартом. Открытый подход, к которому прибегла IBM, опубликовав спецификации и разрешив производителям предлагать совместимые устройства, оправдал ожидания и привел к появлению клонов x86: они тоже использовали процессоры Intel, хотя бы на начальном этапе.

Хотя процессоры Intel широко используются в настольных компьютерах и ноутбуках, с ростом популярности мобильных устройств наблюдается активное распространение архитектуры ARM и совсем недавней RISC-V. Кроме того, в моду стремительно входят кросс-архитектурные языки программирования, такие как Go или Rust со своими инструментариями, и вносят в экосистему настоящий хаос.

Упомянутые выше среды я считаю «современными». Большинство из них, если не все, так или иначе работают с Linux или используют ее.

Теперь, когда мы узнали о современных (аппаратных) системах, можно задаться вопросом, как возникла и развилась сама Linux.

## История Linux (далекая от завершения)

В 2021 году Linux отпраздновала свое тридцатилетие. Развитие проекта, в который вовлечены миллиарды пользователей и тысячи разработчиков, без сомнения можно считать всемирной историей успеха (спасибо открытому исходному коду). Но как же все начиналось и как пришло к сегодняшней славе?

### *1990-е*

Точкой отсчета для проекта Linux, по крайней мере исходя из общедоступной информации, можно считать электронное письмо от 25 августа 1991 года, отправленное Линусом Торвалдсом в новостную группу `comp.os.minix`. Любительский проект заметили сразу: он выгодно отличался количеством строк кода и простотой внедрения. Не прошло и трех лет, как свет увидела Linux 1.0.0: в ней насчитывалось более 176 000 строк кода. Изначальная цель — возможность запуска большинства программ Unix/GNU — к тому моменту была достигнута. Тогда же появился и первый коммерческий дистрибутив: Red Hat Linux.

### *2000-2010 годы*

В «подростковом» возрасте Linux не только продолжал наращивать базу предлагаемых функций и список совместимого оборудования, но превзошел по возможностям и саму UNIX. Система привлекла и интерес, и признание

таких крупных игроков, как Google, Amazon, IBM и многих других. Вскоре наступил пик войны дистрибутивов, в результате которого целые предприятия меняли направление деятельности.

*с 2010 года по настоящее время*

Linux зарекомендовал себя в качестве «рабочей лошадки» в дата-центрах, в облаке, на бесчисленных типах IoT устройств и телефонах. В каком-то смысле закончилась даже война дистрибутивов (в наши дни большинство коммерческих систем основаны либо на Red Hat, либо на Debian), и немалую роль в этом сыграло развитие контейнерных технологий (с 2014/15 и по наши дни).

После сверхбыстрого исторического обзора, который задал тон и прояснил, как подбирались темы для книги, перейдем к невинному на первый взгляд вопросу: для чего нужна Linux и ОС в целом?

## Так зачем же нужна операционная система?

Представьте, что ОС у вас нет или она недоступна. Практически все вам придется делать самому: управлять памятью, обрабатывать прерывания, взаимодействовать с устройствами ввода/вывода (I/O), заботиться о файлах и настройках, обеспечивать взаимодействие с сетевым стеком — список длинный.



С технической точки зрения операционная система не обязательна. Есть системы, обычно крошечные и самодостаточные, где ОС не требуется. Представьте себе IoT-маячок. В нем не хватит ресурсов ни для чего, кроме единственной специфичной программы. К примеру, используя язык Rust и подключив лишь его ядро и стандартную библиотеку, вы можете запустить любое приложение прямо на голом железе.

ОС берет на себя всю неявную и сложную работу, отделяя вас от аппаратных компонентов и предоставляя (как правило) чистый и понятно оформленный программный интерфейс для взаимодействия с приложениями (Application Programm Interface, или API). Такой API есть и у ядра Linux; его мы рассмотрим в главе 2. Обращения к предоставленному системой API называют «*системными вызовами*» (*systemcalls*, для краткости *syscalls*). Языки верхнего уровня (Go, Rust, Python или Java) построены поверх таких вызовов и выполняют их «под капотом» своих библиотек.