Ядро Linux

Описание процесса разработки

Linux Kernel Development

Third Edition

Robert Love

Addison
Wesley

Upper Saddle River, NJ • Boston • Indianapolis • San Francisco
New York • Toronto • Montreal • London • Munich • Paris • Madrid
Cape Town • Sydney • Tokyo • Singapore • Mexico City

Ядро Linux

Описание процесса разработки

Третье издание

Роберт Лав

Москва • Санкт-Петербург • Киев
2013

ББК 32.973.26-018.2.75
Л13
УДК 681.3.07

Издательский дом “Вильямс”

Зав. редакцией *С.Н. Тригуб*

Перевод с английского и редакция канд. физ.-мат. наук *С.Г. Тригуб*

По общим вопросам обращайтесь в Издательский дом “Вильямс” по адресу:
info@williamspublishing.com, http://www.williamspublishing.com

Лав, Роберт.

Л13 Ядро Linux: описание процесса разработки, 3-е изд. : Пер. с англ. — М. : ООО “И.Д. Вильямс”, 2013. — 496 с. : ил. — Парал. тит. англ.

ISBN 978-5-8459-1779-9 (рус.)

ББК 32.973.26-018.2.75

Научно-популярное издание

Роберт Лав

Ядро Linux: описание процесса разработки

3-е издание

Литературный редактор *И.А. Попова*
Верстка *М.А. Удалов*
Художественный редактор *В.Г. Павлютин*
Корректор *Л.А. Гордиенко*

Подписано в печать 29.06.2012. Формат 70x100/16.
Гарнитура Times. Печать офсетная.
Усл. печ. л. 39,9. Уч.-изд. л. 32,1.
Тираж 1000 экз. Заказ № 0000.

Первая Академическая типография “Наука”
199034, Санкт-Петербург, 9-я линия, 12/28

ООО “И. Д. Вильямс”, 127055, г. Москва, ул. Лесная, д. 43, стр. 1

ISBN 978-5-8459-1779-9 (рус.)
ISBN 978-0-672-32946-3 (англ.)

Оглавление

Содержание

♦

Посвящается Дорис (Doris) и Элен (Helen)

♦

Предисловие

В связи с тем, что ядро и приложения операционной системы Linux используются все более широко, возрастает число разработчиков системного программного обеспечения, желающих заняться разработкой и поддержкой операционной системы Linux. Часть из этих разработчиков руководствуется исключительно собственным интересом, часть — работает в компаниях, которые занимаются операционной системой Linux, часть — работает на производителей компьютерных аппаратных средств, часть — занята в проектах по разработке программного обеспечения на дому.

Однако все они сталкиваются с общей проблемой: кривая затрат на изучение ядра становится все длиннее и круче. Система становится все более сложной и, кроме того, очень большой по объему. Годы проходят, и нынешние члены команды разработчиков ядра приобретают все более широкие и глубокие знания, что увеличивает разрыв между ними и разработчиками-новичками.

Я уверен, что понимание основного кода ядра Linux уже сейчас является проблемой, приводящей к ухудшению качества ядра, и в будущем эта проблема станет еще более серьезной. Все, кому нравится операционная система Linux, несомненно, заинтересованы в увеличении числа разработчиков, которые смогут внести свой вклад в развитие ядра этой операционной системы.

Один из возможных подходов к решению данной проблемы — ясность исходного кода: удобные интерфейсы, четкая структура, следование принципу “Лучше меньше, да лучше” и т.д. Такое решение предложено Линусом Торвальдсом (Linus Torvalds).

Подход, который предлагаю я, состоит в использовании большего числа комментариев в исходном коде, что поможет читателю понять, чего хотел достичь программист. (Процесс выявления расхождений между поставленной программистом целью и полученной в результате реализацией называется *отладкой*. Этот процесс значительно затрудняется, если не известно, чего хотели достичь.)

Однако комментарии все же не дают представления о том, для чего предназначено большинство подсистем и как разработчики приступали к их реализации. Именно печатное слово лучше всего подходит для отправной точки такого понимания.

Вклад Роберта Лава (Robert Love) состоит в предоставлении возможности, благодаря которой опытные разработчики смогут получить полную информацию о том, какие функции должны выполнять различные подсистемы ядра и каким образом предполагается выполнение этих функций. Этой информации должно быть достаточно для многих людей: для любопытных, для разработчиков прикладного программного обеспечения, для тех, кто хочет ознакомиться с устройством ядра, и т.д.

Кроме того, данная книга является ступенькой, которая поможет начинающим разработчикам перейти на новый уровень, где изменения в ядро вносятся для того, чтобы достичь определенной цели. Я хотел бы посоветовать начинающим разработчикам, чтобы они не боялись испачкать свои руки: наилучший способ понять какую-либо часть ядра —

это внести в нее изменения. Внесение изменений повышает понимание разработчика до уровня, которого нельзя достичь простым чтением кода ядра. Серьезный разработчик ядра присоединится к спискам рассылки разработчиков и будет контактировать с другими коллегами. Это основной способ, позволяющий разработчикам учиться и быть на высоком уровне. Роберт очень хорошо осветил механизмы и культуру этой важной части жизни сообщества разработчиков ядра.

Пользуйтесь книгой Роберта и учитесь по ней! Может быть, и вы решите сделать следующий шаг и вступить в сообщество разработчиков ядра, куда мы вас и приглашаем. Людей ценят по важности их дел, поэтому, помогая развитию операционной системы Linux, знайте, что ваша работа — небольшая, но непосредственная помощь десяткам и даже сотням миллионов людей.

Эндрю Мортон (Andrew Morton)
Open Source Development Labs

Введение

Сделав первую попытку превратить свой опыт работы с ядром Linux в текст книги, я понял, что не знаю, куда двигаться дальше. Не хотелось просто писать еще одну книгу о ядре операционной системы. Конечно, на эту тему написано *не так уж и много* книг, но все же я хотел сделать что-то такое, благодаря чему моя книга была бы особенной. Как достичь этой цели? Я не могу успокоиться, пока не сделаю что-нибудь особенное, лучшее в своем роде.

Наконец я решил, что смогу предложить достаточно уникальный подход к данной теме. Моя работа — изучение и разработка ядра операционной системы. Мое увлечение — изучение и разработка ядра операционной системы. Моя любовь — ядро операционной системы. Конечно, за многие годы я успел собрать много интересных анекдотов и полезных советов. С моим опытом я смог бы написать книгу о том, как нужно разрабатывать программный код ядра и как этого делать *не нужно*. Прежде всего, эта книга о структуре и практической реализации ядра операционной системы Linux. Информация в ней представлена так, чтобы получить достаточно знаний для решения реальных практических задач и решать эти задачи правильно. Я человек прагматичный, и книга имеет практический уклон. Она должна быть полезной, интересной и легко читаться.

Я надеюсь, что читатели после прочтения этой книги получат хорошее понимание тех правил (писанных и неписаных), которые действуют в ядре операционной системы. Надеюсь также, что читатели сразу после прочтения этой книги смогут начать действовать и писать полезный, правильный и хороший код ядра. Конечно, эту книгу можно читать и просто ради интереса.

Это то, что касалось еще первого издания книги. Однако время идет, и снова приходится возвращаться к рассмотренным вопросам. В этом, третьем по счету, издании представлено несколько больше информации по сравнению с двумя остальными: материал серьезно пересмотрен и доработан, появились новые разделы и главы. С момента выхода второго издания в ядро были внесены изменения. Но, что более важно, сообщество разработчиков ядра Linux приняло решение[1] в ближайшем будущем не начинать разработку ядра версии 2.7. Было решено заняться стабилизацией ядра версии 2.6. Стабилизация включает в себя много моментов, тем не менее есть один важный, касающийся данной книги, — книга, которая посвящена ядру серии 2.6, остается актуальной до сих пор. Поскольку изменения происходят не слишком быстро, существует большой шанс, что "моментальный снимок" ядра останется актуальным и в будущем. Эта книга, по сути, стала канонической документацией по ядру, в которой отражены как история, так и взгляд в будущее.

[1] Это решение было принято на саммите разработчиков ядра Linux (Linux Kernel Development Summit), который состоялся летом 2004 года в Оттаве, Канада. Меня пригласили туда в качестве участника.

Итак...

Разработка программного кода ядра операционной системы не требует наличия гениальной, волшебной или густой бороды Unix-хакера. Хотя ядро операционной системы и имеет некоторые свои особенности, оно незначительно отличается от любого большого программного продукта. Так же как и в случае любой сложной программы, здесь есть, что изучать, но в программировании ядра не намного больше священных или непонятных вещей, чем в создании любой другой программы.

Очень важно, чтобы вы читали программный код. Доступность открытого исходного кода операционной системы Linux — это подарок, который встречается очень редко. Однако недостаточно *только* читать исходный код. Необходимо взяться за дело серьезно и изменять этот программный код. Находите ошибки и исправляйте их! Улучшайте драйверы для своего аппаратного обеспечения! Добавляйте новые функциональные возможности в ядро, даже если они на первый взгляд кажутся весьма простыми. Находите слабые места и закрывайте их! У вас все получится, если вы будете сами *писать* программный код.

Версия ядра

В этой книге рассмотрено ядро Linux версии 2.6. Я не стал описывать устаревшие версии ядра, за исключением случаев, представляющих чисто исторический интерес. Например, мы рассмотрим, как были реализованы определенные подсистемы в ядре версии 2.4, поскольку тогда они были намного проще, и сможем легче во всем разобраться. Что касается данной книги, то она была написана на момент, когда актуальным являлось ядро Linux версии 2.6.34. Ядро — это "движущийся объект", и никакая книга не в состоянии передать динамику во все моменты времени. Тем не менее базовые внутренние структуры ядра уже сформировались, и основные усилия по представлению материала были направлены на то, чтобы этот материал можно было использовать и в будущем.

Несмотря на то что в этой книге описано ядро версии 2.6.34, я постарался сделать так, чтобы в ней также были максимально корректно отражены сведения по ядру версии 2.6.32. Дело в том, что эта устаревшая версия была официально одобрена в качестве "корпоративного" ядра, которое продолжает использоваться во многих дистрибутивах Linux. А это значит, что мы еще долго будем встречать его во многих производственных системах и оно еще много лет будет находиться в состоянии активной разработки. (Существовавшие ранее ядра версий 2.6.9, 2.6.18 и 2.6.27 — аналогичные примеры "долгожителей".)

Читательская аудитория

Эта книга предназначена для разработчиков программного обеспечения, которые хотят понять, как устроено ядро операционной системы Linux. Тем не менее она *не* является сборником построчных комментариев, извлеченных из исходного кода ядра. Ее также нельзя считать руководством по разработке драйверов или справочником по программному интерфейсу (API) ядра. Целью книги является предоставление достаточной информации о структуре и реализации ядра, чтобы подготовленный программист смог начать разработку программного кода. Разработка ядра может быть увлекательным и полезным занятием, и я хочу ознакомить читателя с этой сферой деятельности по возможности быстро. В книге обсуждаются как вопросы теории, так и практические приложения, она об-