

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 неписанных), 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 решение¹ 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-

ращена к людям, которые интересуются и тем и другим. Я всегда придерживался мнения, что для понимания практических приложений необходима теория, тем не менее я считаю, что эта книга не сильно углубляется в оба этих направления. Надеюсь, что, независимо от мотиваций необходимости понимания ядра операционной системы Linux, эта книга сможет объяснить особенности устройства и реализации в достаточной степени.

Таким образом, данная книга освещает как использование основных подсистем ядра, так и особенности их устройства и реализации. Думаю, что эти вопросы важны и достойны обсуждения. Хороший пример — глава 8, “Нижняя половина обработчика и отложенные действия”, посвященная компонентам драйверов устройств, называемых *нижними половинами* (bottom half). В этой главе рассказывается о принципах работы и об особенностях реализации механизмов обработки нижних половин (эта часть может быть интересна разработчикам основных механизмов ядра), а также о том, как на практике использовать экспортируемый интерфейс ядра для реализации собственных обработчиков нижних половин (это может быть интересно разработчикам драйверов устройств, а также другим программистам-профессионалам). На самом деле мне кажется, что обе эти стороны обсуждения будут интересны всем группам разработчиков. Разработчик основных механизмов ядра, который, конечно, должен понимать принципы работы внутренних частей ядра, должен также понимать и то, как интерфейсы ядра будут использоваться на практике. В то же самое время разработчик драйверов устройств получит большую пользу от хорошего понимания того, что стоит за этим интерфейсом.

Все это сродни изучению программного интерфейса некоторой библиотеки наряду с изучением того, как эта библиотека реализована. На первый взгляд, разработчик прикладных программ должен понимать лишь интерфейс (API). И действительно, интерфейсы часто предлагают рассматривать в виде “черного ящика”. Разработчик библиотеки, наоборот, обычно интересуется лишь принципом работы и реализации функций библиотеки. Я уверен, что обе группы разработчиков должны потратить некоторое время на изучение другой стороны предмета. Разработчик программ, который хорошо понимает операционную систему, сможет значительно лучше использовать ее. Аналогично разработчик библиотеки должен иметь хотя бы малое представление о том, что происходит в реальной жизни, и, в частности, о тех программах, в которых будет использоваться его библиотека. Поэтому я старался коснуться как устройства, так и использования подсистем ядра не только в связи с тем, что эта книга может быть полезна одной или другой группе разработчиков, а в надежде, что *весь материал* книги будет полезен всем разработчикам.

Предполагается, что читатель знаком с языком программирования C и операционной системой Linux. Некоторые знания принципов построения операционных систем также желательны. Я старался объяснять все понятия, однако в случае проблем в списке литературы можно найти несколько отличных книг, которые посвящены основам построения операционных систем.

Эта книга будет полезна студентам, изучающим основы построения операционных систем, в качестве *прикладного* пособия и вводного материала по соответствующей теории. Книга пригодна как для расширенных специальных курсов, так и для общих специальных курсов, причем в последнем случае без дополнительных материалов.

Благодарности

Как и большинство авторов, я писал эту книгу, не сидя в пещере (что само по себе хорошо, потому что в пещерах могут водиться медведи), и, следовательно, многие люди оказали мне поддержку в создании рукописи своим сердцем и умом. Поскольку невозможно привести полный список этих людей, хочу поблагодарить всех своих друзей и коллег за помощь, поддержку и конструктивную критику.

В первую очередь хотел бы высказать благодарность моим коллегам из издательства Addison–Wesley и компании Pearson, которые долго и упорно трудились над тем, чтобы сделать эту книгу лучше. В особенности хочу поблагодарить Марка Табера (Mark Taber) за руководство, благодаря которому третье издание книги превратилось из идеи в конечный продукт, а также редактора-консультанта (development editor) Майкла Гарстона (Michael Thurston) и редактора проектов (project editor) Тонию Симпсон (Tonya Simpson).

Особые благодарности я хотел бы выразить техническому редактору данного издания Роберту П. Дж. Дею (Robert P. J. Day). Его проницательность, опыт и правки помогли сделать эту книгу неизмеримо лучше. Тем не менее, если, несмотря на все его титанические усилия, некоторые ошибки все же остались в книге, то это — вина автора! Такое же огромное спасибо Адаму Билаю (Adam Belay), Заку Брауну (Zak Brown), Мартину Пулу (Martin Pool) и Крису Ривера (Rivera), которые славно потрудились над техническим редактированием первого и второго издания этой книги и не ударили в грязь лицом при работе над третьим изданием.

Многие разработчики ядра отвечали на вопросы, предоставляли поддержку или просто писали программный код, интересный настолько, что по нему можно было бы написать отдельную книгу. Среди них Андреа Аркангели (Andrea Arcangely), Алан Кокс (Alan Cox), Грег Кроа-Хартман (Greg Kroah-Hartman), Дейв Миллер (Dave Miller), Патрик Мочел (Patrick Mochel), Эндрю Мортон (Andrew Morton), Ник Пиггин (Nick Piggin) и Линус Торвальдс (Linus Torvalds).

Огромное спасибо хочу сказать моим коллегам из Google, самой творческой и интеллектуальной группе людей, с которыми мне когда-либо приходилось сталкиваться и иметь удовольствие работать. Если я здесь начну перечислять их всех, мне не хватит объема книги. Поэтому я хочу выделить Алана Блоунта (Alan Blount), Джея Крима (Jay Crim), Криса Дэниса (Chris Danis), Криса Дибона (Chris DiBona), Эрика Флэтта (Eric Flatt), Майка Локвуда (Mike Lockwood), Сана Мехата (San Mehat), Брайана Рогана (Brian Rogan), Брайана Свитланда (Brian Swetland), Джона Тробрайда (Jon Trowbridge) и Стива Винтера (Steve Vinter) за их дружелюбие, эрудицию и помощь.

Я хочу выразить свою любовь и признательность Полу Амичи (Paul Amichi), Мики Бэббиту (Mikey Babbitt), Киту Бэрбегу (Keith Barbag), Джейкобу Беркману (Jacob Berkman), Нату Фридману (Nat Friedman), Дастину Холлу (Dustin Hall), Джойсу Хокинсу (Joyce Hawkins), Мигуэлю де Иказа (Miguel de Icaza), Джимми Крелу (Jimmy Krehl), Дорис Лав (Doris Love), Линде Лав (Linda Love), Бретти Лаку (Brette Luck), Рэнди О’Дауду (Randy O’Dowd), Сальваторэ РибAUDO (Salvatore RibAUDO) и его чудесной маме, Крису Ривера (Chris Rivera), Кэролайн Родон (Carolyn Rodon), Джой Шо (Joey Shaw), Саре Стьюарт (Sarah Stewart), Джереми ВанДорену (Jeremy VanDoren) и его семье, Льюису Вилле (Luis Villa), Стиву Вейсбергу (Steve Weisberg) и его семье, а также Хелен Винснант (Helen Whinsnant).

И в заключение хочу поблагодарить моих родителей за то, что по наследству от них мне достались хорошо сложенные ушные раковины. Желаю им большого хакерского счастья!

*Роберт Лав,
Бостон.*