

Различие между операционными системами семейств BSD и Linux

Оглавление

1. Вступление
2. Действующие лица
3. Базовая система
4. Коллекция Портов
5. Релизы
6. Обновление базовой системы
7. Обновление коллекции портов
8. Философия
9. Мифы
10. Заключение
11. Ответы на вопросы

Вступление

Что это?

Я использую FreeBSD на своих компьютерах, а многие из моих друзей используют дистрибутивы Linux. Естественно, мы считаем, что Unix-подобные операционные системы - это правильный выбор, но мы не можем прийти к согласию какую именно систему стоит использовать.

У меня сложилось впечатление, что сообщество BSD, в целом, понимает Linux гораздо лучше, чем сообщество Linux понимает BSD. У меня есть несколько теорий на этот счет, но это не столь важно. Я думаю, что многие пользователи Linux прекращают использовать BSD-системы, только потому что не совсем понимают принцип их работы. Как человек использующий BSD, я хочу попытаться объяснить, как BSD-системы устроены, чтобы пользователи Linux могли их лучше понять.

Хотя в большинстве случаев существует огромное сходство между обсуждаемыми операционными системами, одновременно существует и множество различий. По мере того, как мы будем углубляться в их различия, станет понятным, что они основаны на глубоких разногласиях. Некоторые из них - разногласия по поводу методологии разработки, некоторые по поводу развертывания и использования, некоторые по поводу того, что важно, некоторые по поводу того, кто важен. Простое сравнение поверхностных различий ничего не даст, именно глубокие различия объясняют и оправдывают действия и взгляды каждой системы.

Чем это не является

В этом повествовании вы не найдете ответы на следующие вопросы:

- Список соответствий команд; «netstat -rnfinet» в BSD = «netstat -rnAinet» в Linux и тому подобное.
- Руководство по администрированию BSD.
- Почему вы должны использовать BSD вместо Linux?
- Почему вы должны использовать Linux вместо BSD?
- Почему вы должны использовать этот BSD вместо другого BSD?
- Почему вы должны использовать этот Linux вместо того Linux?
- Почему BSD прав, а Linux неправ?
- Почему Linux прав, а BSD неправ?

Лично я считаю (естественно), что выбрал правильную операционную систему. Но это я. Я не говорю вам, что нужно следовать за мной. Узнайте факты и происхождение этих фактов, и примите решение. Это ваше право.

Немного рассуждений

Существует много философских разговоров и споров по поводу различий между Linux и BSD. Так же существует и не мало шуток. Одна из них мне особенно нравится и звучит примерно так:

BSD - это то, что вы получаете, когда кучка Unix-хакеров садится, чтобы попытаться перенести систему Unix на ПК.
Linux - это то, что вы получаете, когда кучка компьютерных хакеров садится и пытается написать систему Unix для ПК.

Мне нравится эта шутка не потому, что она является окончательным и верным ответом в споре о различии обсуждаемых систем, но она хороша дает понять о существенных различиях. В целом, BSD-системы гораздо больше похожи на традиционные Unix, чем на Linux. Во многом это связано с тем, что они являются прямыми потомками BSD из Беркли, который был прямым потомком оригинальной AT & T Unix. Товарный знак Unix является товарным знаком The Open Group, а Unix-the-code принадлежит SCO, поэтому на самом деле нельзя утверждать, что BSD это действительно Unix (подобного рода утверждение в свое время явилось причиной судебного иска USL / UCB, что на самом деле феерично). Но во многих отношениях BSD-системы являются прямыми производными традиционного Unix.

Это проявляется во многих отношениях. Это проявляется в дизайне «базовой системы» и дополнениях. Это проявляется при разметке запоминающих устройств. Это проявляется во многих деталях команд. Так же это проявляется в отношении, поведении и предрассудках разработчиков, что в свою очередь отражается в коде и влияет на работу пользователей.

BSD разработан. Линукс взращен. Возможно, это единственный краткий способ описать различие, и, возможно, самый правильный.

С чего же начать?

В первую очередь я хочу вернуться к истории обсуждаемых систем и определениям.

Действующие лица

Здесь я предоставляю краткое описание обсуждаемых систем, о которых буду говорить дальше. Договоримся, что представленная информация не является исчерпывающей и единственно верной, потому что изначально такая цель и не преследовалась.

Unix

Unix не является операционной системой.

Ну, это так и не совсем так.

Unix - это операционная система, изначально разработанная Кеном Томпсоном и Деннисом Ритчи в конце 60-х годов в Bell Labs. С тех пор она разрабатывалась и распространялась как коммерческая и исследовательская операционная система компаниями Bell Labs, USG, USDL, ATTIS, USL, Novell и SCO и всеми другими, кто мог себе придумать аббревиатуру.

Без преувеличения можно сказать, что Unix является единственной наиболее влиятельной операционной системой в современном компьютерном мире. Каждое вычислительное устройство общего назначения, которое вы найдете и множество вычислительных устройств специального назначения будут использовать как идеи и концепции, так и программный код, основанный на семейном древе Unix.

Когда я использую слово «Unix», тогда я чаще имею в виду общее представление данных, чем конкретную ОС, которая носит название Unix. Общее представление данных означает «Любая операционная система», которая по дизайну и исполнению, а также по интерфейсу и общей концепции в значительной степени похожа на систему Unix. Это относится к BSD, Linuxen, SunOS, Tru64, SCO, Irix, AIX, HP / UX и к сотне других систем.

Меня не интересуют философские дискуссии из разряда сколько ангелов уместится на кончике иглы. Теперь вы точно знаете, какой смысл я вкладываю при использовании фразы типа «Unix system».

Linux

Linux также означает несколько вещей. Это ядро, изначально написанное Линусом Торвальдсом, когда он был студентом в Финляндии. С тех пор это ядро били кулаками, подправляли, тыкали, подталкивали, подбрасывали, переваривали, выплевывали, топтали, жевали, выбрасывали и другим образом манипулировали (не обязательно в

таким порядке, конечно) столько раз и такое количество людей, которое сложно сосчитать.

Linux - это также термин для семейства операционных систем. Хотя в эту самую секунду (я гарантирую это) происходят увлекательные метафизические дискуссии в десятках мест по всему миру о том, что «Linux на самом деле не операционная система, а просто ядро» или «Его следует называть «GNU/Linux» или аналогичные темы, то я также буду аккуратно избегать подобных семантических рассуждений. Когда я говорю «Linux», я имею в виду [Red Hat](#), [Slackware](#), [Mandrake](#), [Debian](#), [SuSe](#), [Gentoo](#). Я имею в виду каждый из двух квадриллионов дистрибутивов, основанных на ядре Linux с практически одинаковыми пользовательскими областями, и в большинстве своем основанных на инструментах GNU, которые распространяются по всему миру.

BSD

BSD расшифровывается как «Berkeley Software Distribution». Первоначально это был набор исправлений и дополнительных утилит для официальной системы Bell Unix, которые были разработаны CSRG (Computer Systems Research Group) в Университете Калифорнии в Беркли. Со временем этот набор изменений стал становиться больше изначальной системы, пока в какой-то неопределенный момент он не превратился в свою собственную ОС, которая просто обменивалась частями кода с Unix-системой Bell.

Конечно, для использования системы все еще требовалось иметь лицензию Bell, так как большая ее часть все еще оставалась кодом Bell. Однако весь код, написанный в Беркли, был выпущен под так называемой лицензией BSD, которую кратко описать можно примерно так: «Делайте что хотите с кодом, просто поблагодарите нас за его написание». И большая часть кода BSD в конечном итоге вернулась в «официальную» систему Unix, в System III и System V. И многие из этих патчей и дополнений стали использоваться в различных коммерческих форках Unix.

Последующий распад CSRG привел к остановке разработки системы BSD, в то же время часть разработчиков продолжила работу. Результатом их работы стал проект 386BSD, в котором был использован код BSD и портирован на платформу Intel i386. Когда проект 386BSD был свернут, две основные группы разработчиков использовали код 386BSD и разделились. Результатом работы одной из групп стал проект FreeBSD, другой - NetBSD. Позже, внутренние разногласия проекта NetBSD привели к уходу части разработчиков, которые создали проект OpenBSD.

Когда я говорю «BSD», я имею в виду общую философию BSD и подход к системам. В общем, все вышесказанное привело к появлению трех современных свободно доступных систем BSD:

[FreeBSD](#) нацелен на получение максимальной производительности на платформе 386. Позже система была портирована на ряд других платформ, включая Alpha и SPARC, а также на i386, Intel Itanium и AMD Opteron. Основная цель - быть максимально надежной и эффективной на различных платформах как для серверов, так и для настольных компьютеров.

[NetBSD](#) предназначен для работы на максимально возможном количестве платформ. Цель - стать самой портативной ОС на планете, и она, похоже, справляется с этим довольно неплохо.

[OpenBSD](#) ориентирован в первую очередь (некоторые сказали бы «исключительно») на вопросы безопасности и смежные с ней вопросы. Тесная интеграция безопасности и криптографии является основной целью.

Все эти цели, конечно, взаимозаменяемы. Каждая система BSD заботится о безопасности и работает над ней. Каждая BSD заботится о производительности и работает над ней. Каждая BSD заботится о мобильности и работает над ней. Существует общий код, который присутствует во всех системах. Так же многие разработчики одновременно работают над несколькими системами.

Проницательные читатели могут заметить, что я не упоминаю [Mac OS X](#) или более низкий слой в виде [Darwin](#) как BSD. Хотя они и основаны в основном на BSD, но верхние уровни OS X это чистый Apple. Работая с OS X, как пользователь, вы обычно используете его как MacOS, а не как BSD. Поэтому, хотя некоторые из них могут быть применимы академически к OS X, это не имеет никакой практической ценности. Darwin ближе к стандартному ощущению BSD, но большая часть его базы пользователей - это люди, пришедшие из BSD, так что это тоже немного выходит за рамки данного эссе.

При обсуждении специфики в этом эссе я в основном буду ссылаться на FreeBSD, потому что эту систему использую лично я, и безусловно, знаком с ней лучше чем с другими. По очень детальным моментам будут существенные различия между системами BSD. Чем шире общий смысл, тем более вероятно, что он будет одинаковым всех систем. С философской точки зрения все BSD очень похожи, в отличие от методологии Linux. И вообще, философия - это то, о чем в первую очередь написано данное эссе.

Итак, давайте начнем сравнение

Следующее на чем мы остановимся, это различия в архитектуре, а именно такое понятие как «базовая система».

Базовая система

Концепция «базовой системы» - это то, что, как мне кажется, вызывает большинство вопросов у людей, привыкшим к методологии Linux. Что совершенно понятно, потому что сама идея «базовой системы» просто не существует в мире Linux.

Linux с самого начала был просто ядром. Не вдаваясь в вечные споры о том, из чего именно состоит операционная система, очевидно, что ядро само по себе мало чем полезно. Вам нужны дополнительные утилиты и программы, чтобы работать с ним. Linux всегда был конгломератом: ядро отсюда, ls оттуда, ps из другого места, vim, perl, gzip, tar из третьего и так далее.

В Linux никогда не было никакого разделения между тем, что является «базовой системой» и тем, что является «дополнительными программами и пакетами». Вся система, по сути, является набором «дополнительных программ и пакетов». MySQL ничем не отличается от ls, KDE от whois, от dc, от GnuCash... Каждый бит системы - это просто та или иная «дополнительная программа или пакет».

В BSD-системах, напротив, всегда была централизованная модель развития. BSD не использует GNU ls или GNU libc, она использует BSD ls и BSD libc, которые являются прямыми потомками ls и libc, которые были в распространяемых CSRG выпусках BSD. Они никогда не были разработаны сторонними разработчиками. Вы не можете пойти куда-нибудь и «скачать BSD libc», потому что в мире BSD библиотека libc сама по себе бессмысленна, ls сам по себе бессмысленен, ядро само по себе бессмысленно. Система в целом представляет собой единое целое, а не кучку маленьких кусочков.

X Window не является частью базовой системы FreeBSD. Это «дополнительный пакет». Поскольку X Window не является частью базовой системы, то и приложения X Window и такие пакеты как xterm и KDE, а также Gnome и Mozilla, gaim и xmms, очевидно, тоже не могут быть частью базовой системы и рассматриваются как дополнительные пакеты.

NetBSD и OpenBSD имеют базовую реализацию X Window, но из-за особенностей интеграции со своим драйверами, они используют сильно модифицированные, очень нестандартные версии, поэтому не представляется возможным использовать сторонний пакет X Window.

Вся «базовая система» разрабатывается едино. Конечно, есть части «базовой системы», такие как sendmail, BIND, tcpdump, ssh и тому подобное, которые на самом деле являются отдельными пакетами, которые разрабатываются сторонними разработчиками. Есть даже некоторые пакеты GNU, такие как groff, gcc, gzip и другие, которые будут сразу узнаваемы любым пользователем Linux. Но они обрабатываются специально, а затем формируются в соответствии с остальной частью системы. Фактически, многие из этих пакетов раньше были доступны исключительно для BSD; BIND и sendmail были первоначально разработаны в Беркли как часть BSD, и только позже стали доступны отдельно. Моя система FreeBSD утверждает, что на данный момент работает с gcc версии 3.2.2. Технически это не совсем gcc 3.2.2, это компилятор FreeBSD, основанный на gcc 3.2.2. У меня есть версия tcpdump, технически не 3.7.2, это версия tcpdump для FreeBSD, основанная на tcpdump 3.7.2.

Конечно, в большинстве случаев версии программ и пакетов «базовой системы» FreeBSD практически неотличимы от версий, разработанных сторонними разработчиками. Обычно при сборке программ и пакетов вносятся некоторые изменения, например в Makefiles, чтобы обеспечить более гладкую и чистую сборку, а иногда и некоторые необходимые исправления, чтобы заставить их скомпилироваться и работать правильно. Одни изменения более глубокие, другие более поверхностные, но все они направлены на то, чтобы все элементы базовой системы работали слажено и дать понять, что системы BSD спроектированы верно.

Основная причина того, что поддерживаемый извне пакет импортируется и отслеживается в «базовой системе», заключается в том, что он в некотором роде является достаточно базовым для функционирования системы, поэтому его проще всего иметь по умолчанию. В настоящее время FreeBSD использует ssh-сервер и клиент OpenSSH, которые интегрированы в «базовую систему», поскольку в наше время метод безопасного удаленного доступа является достаточно «базовым» для любой системы, то это является практической необходимостью. gcc и binutils являются частью «базовой системы», потому что... ну, они необходимы для построения «базовой системы». X и Gnome, PostgreSQL и Apache и другие не являются частью «базовой системы» и, вероятно, никогда не будут, потому что они не требуются для запуска и развертывания системы, не будут использоваться во многих системах и не нуждаются в такой тесной интеграции.

Важно подчеркнуть эту идею «базовой системы», так как она, я думаю, является одним из наиболее заметных признаков различий в методологиях разработки между BSD и Linux. Разработчики дистрибутивов Linux могут собрать кусочки программ и проверить их соответствие перед выпуском. Они могут даже иметь некоторые настройки для некоторых пакетов. Но это довольно сильно отличается от интенсивной интеграции BSD. Особенно в связи с тем, что многие из компонентов «базовой системы» изначально не происходят «откуда-то еще», они

прожили всю свою жизнь как часть системы BSD.

Некоторые дистрибутивы Linux, особенно Debian и Gentoo, стараются учитывать разделение на «базовую систему» и дополнительные пакеты, которые вы, возможно, захотите установить позже. Что отличает это от «базовой системы» BSD, так это то, что большинство этих пакетов все еще берутся извне и просто «собираются» для каждого дистрибутива. Например, нет «Gentoo tftp»; Хотя многие пакеты в BSD так же «берутся откуда-то еще», но большая часть «базовой системы» никогда не была получена от сторонних разработчиков.

«Базовую систему» иногда воспринимают как «урезанную», с тем подтекстом, что она разрабатывается только с целью быть как можно меньше. Но это действительно не так, «базовая система» представляет собой в действительности основу для будущей системы. Я никогда в своей жизни не использовал систему BSD, работающую только в «базовом виде», и, вероятно, никогда не буду. И я не знаю никого, кто использует. «Базовая система» просто должна предоставить инструменты для запуска системы и позволить вам обновить ее и установить необходимые пакеты. Далее вы устанавливаете то, что вам нужно, для конкретной роли вашей системы.

Мне сказали, что вышеупомянутое более справедливо для FreeBSD, чем для NetBSD или OpenBSD, которые во многих отношениях имеют гораздо более широкие критерии для определения того, что включать в базовую систему. Естественно, вам имеет смысл обратиться к руководствам каждой из систем, чтобы точно определить, что является «базовой системой» для каждой из них.

Что же дальше?

После углубления в понятие «базовая система» пришло время поговорить о «коллекции портов».

Коллекция Портов

После «базовой системы» в мире BSD существует вторая категория программ. Это те программы, которые являются дополнительными пакетами и называются «коллекцией портов». Это имя выбрано по определенной причине.

Традиционно, когда вы хотите запустить пакет в своей системе, первое, что вам нужно было сделать, то это скомпилировать его. И часто, прежде чем вы сможете скомпилировать его, вам придется повозиться с ним. Вашей системе потребуются дополнительные файлы и библиотеки. Иногда все гораздо сложнее, и вам даже нужно внести изменения в программный код устанавливаемого пакета.

Или, другими словами, вам придется «портировать» его на свою ОС и/или на конкретную систему. Основная цель «коллекции портов» состоит в том, чтобы сделать все эти «портирующие» действия для вас. В то же время «коллекция портов» также автоматизирует сборку, установку и удаление пакетов, хотя это не так хорошо отражено в названии.

В текущей коллекции портов FreeBSD содержится около 10 000 пакетов (это число, конечно, быстро устареет, но такова природа разработки). Самая очевидная особенность «коллекции портов» заключается в том, что установка пакетов происходит из исходного кода, а не просто из предварительно собранных файлов. Это, похоже, еще одно явное различие, которое сбивает людей с толку, когда они пытаются взглянуть на BSD с точки зрения Linux. То, что пакеты из «коллекции портов» строятся из исходного кода, является просто побочным эффектом, а не основной целью или различием. Бинарные пакеты также доступны, фактически бинарные пакеты создаются из «коллекции портов»!

Большинство пользователей Linux устанавливают бинарные пакеты, а большинство пользователей BSD с помощью сборки из исходного кода. «Коллекция портов» разработана на основе концепции сборки из исходного кода, но так же с возможностью создания и установки бинарных пакетов, в то время как пакеты Linux, такие как RPM и dpkg и другие, основаны на концепции установки бинарного пакета. Так исторически сложилось, что установка бинарных пакетов не является доминирующей в системах Unix и является в этом плане своего рода новшеством.

Gentoo - это дистрибутив Linux довольно популярный в наши дни. Одним из главных его преимуществ является система портежей, которая часто считается очень похожей на «коллекцию портов» BSD. В Gentoo пакеты также компилируются из исходного кода, что позволяет избежать многих проблем. Я никогда не использовал этот дистрибутив, но отзывы и впечатления, которыми со мной поделились люди, использующие Gentoo, дают понять, что в этом дистрибутиве собраны идеи от разных систем. Будет очень интересно посмотреть, как этот дистрибутив будет продолжать развиваться в течение следующих лет. Конечно, Gentoo это все еще намного больше Linux, чем BSD, но больше всех из других популярных дистрибутивов Linux соответствует стилю BSD.

У бинарных пакетов есть свои преимущества, в основном это меньшее время для установки и меньшее требование к свободному месту по сравнению с установкой из исходного кода. С другой стороны, у сборки из исходного кода

тоже есть преимущества, например, избегание всякого рода нестандартных версий библиотек (моя личная головная боль с бинарными пакетами). Не смотря на то, что вы можете установить бинарные пакеты как на Linux так и на BSD, вы так же можете собрать эти же пакеты из исходного кода как Linux так на и BSD.

Я думаю, здесь важно понять, что разница между «коллекцией портов» и бинарными пакетами заключается не только в том, что порты компилируются, а последние просто устанавливаются. «Коллекция портов» спроектирована таким образом, чтобы охватывать как можно больше вариантов и деталей установки; кодирование, отслеживание и установка зависимостей, установка и удаление, локальные изменения, необходимые для сборки в вашей системе, настройки во время компиляции и тому подобное. RPM - это просто бинарный пакет. Если вы хотите автоматически устанавливать зависимости, вам нужен высокоуровневый инструмент, например, `dnf` или `apt`. При использовании бинарных файлов, вам приходится иметь дело с конфликтами версий библиотек, отсутствующими опциями компиляции или любыми другими ограничениями, которые возникают у вас из-за того, что пакет был собран не в вашей собственной системе.

Кроме того, «коллекция портов», как и остальные инструменты BSD, централизована. «Коллекция портов» на самом деле представляет собой просто большое дерево с кучей последовательно упорядоченных каталогов, каждый из которых содержит Makefile с некоторыми определениями переменных, файл контрольной суммы, список упаковки и другие компоненты. Каждый из этих каталогов представляет собой отдельную программу, которая описывается имеющимися файлами. За кулисами происходит куча черной магии, так что когда вы запускаете `make`, он находит и загружает дистрибутивный файл, где бы он ни находился, вносит любые необходимые изменения или исправления, рекурсивно находит и устанавливает все зависимости, вызывает GNU `configure` или `imake` или что-то другое, что необходимо для подготовки программы к установке, запускает компиляцию с правильными параметрами и т. д.

Но все эти файлы в этом большом дереве каталогов поддерживаются самим проектом FreeBSD. Например, когда был разработан KDE, он не сразу появлялся в «коллекции портов». Кто-то должен был переработать этот пакет, чтобы построить для него порт, а затем зафиксировать файлы в репозитории FreeBSD CVS, чтобы он появился в «коллекции портов». Поэтому, еще раз, есть некоторые гарантии того, что KDE будет работать с другими пакетами из «коллекции портов». Любые зависимости, которые у него есть, будут там, потому что он не может ссылаться на зависимость, которой нет в «коллекции портов».

С другой стороны, вещи имеют свойство ломаться. Возможно, основной репозиторий может быть недоступен, поэтому никто не сможет скачать исходный файл и начать его установку. Возможно, новая версия какой-то библиотеки будет настолько изменена, что не позволит другим пакетам, которые зависят от нее, работать. Конечно, это не решает всех проблем. Но частота «я хочу А, для которого требуется В, которого я не могу найти», намного меньше, чем в системах, использующих RPM.

За более детальной информацией (впрочем, как всегда), следует обратиться к [Руководству FreeBSD](#).

Время для релиза!

Теперь, после осознания таких понятий как «базовая система» и «коллекция портов», давайте поговорим о релизах и обновлениях.

Релизы

Все BSD используют систему контроля версий; все бесплатные BSD используют [CVS](#). Контроль версий (вкратце) - это система, регистрирующая изменения в одном или нескольких файлах для того, чтобы в дальнейшем была возможность обратиться к определённым старым версиям этих файлов. Полная история всех изменений хранится в системе контроля версий, поэтому вы можете просмотреть историю изменений, проверить старую версию, посмотреть на различия между произвольными версиями и т. д.

Все BSD так или иначе предоставляют открытый доступ к своим хранилищам CVS; как правило, через анонимный CVS или CVSup или зеркалирование, или через все сразу. Это означает, что как пользователь, вы можете увидеть какие изменения были внесены. Кем, когда и почему были они внесены. Вы также всегда можете получить последние изменения (конечно, в течение нескольких часов, в зависимости от частоты обновления зеркал). Все бесплатные BSD имеют списки рассылки, на которые вы можете подписаться и видеть изменения по мере их внесения. На самом деле, у всех них есть веб-интерфейсы; вы можете просмотреть все дерево исходных текстов FreeBSD по адресу <http://cvsweb.freebsd.org/src/> и также просмотреть всю историю каждого файла.

Исторически в Linux не использовалось никакого контроля версий для ядра. Где-то в середине верии 2.4 ядро стало храниться в общедоступном репозитории [BitKeeper](#). Многие другие программы и утилиты, включенные в Linux разрабатывались используя контроль версий, но, поскольку они все разрабатывались отдельно, то и не было централизованного места, куда можно было бы перейти, чтобы просмотреть все изменения. Поэтому иногда трудно получить историческую достоверность всех этапов разработки.

Примечание. По поводу последнего абзаца возникли разногласия. Хотя многие разработчики использовали CVS для некоторых частей ядра, другая информация говорит, что Линус никогда сам не использовал его для ядра, поэтому до перехода на Bitkeeper все изменения не фиксировались в единой системе контроля версий. Смотрите эту [статью](#) для подробностей.

Это приводит к большому количеству различий. Системы BSD постоянно развиваются; Я всегда могу обновить свою систему до самой последней версии, независимо от релизов. В Linux это не имеет особого значения, потому что процесс разработки совершенно другой. Я думаю, что наиболее подходящее определение для релизов Linux, это то, что они «собранны». Релиз Linux собран из версии A.B этой программы, плюс версия C.D этой программы, плюс версия E.F этой программы ... все вместе с версией X.Y.Z ядра Linux. В BSD же, поскольку все части разрабатываются вместе, понятие «срез» имеет гораздо больше смысла; релиз «обрезается» в определенное время.

Linux выпускает ядра параллельно (ну, часто больше, чем два, но мы немного упростим); версия с нечетным второстепенным номером выпускается в качестве версии «для разработчиков», а версия с четным второстепенным номером в качестве версии в качестве «продакшена». У BSD также есть версии для «разработки» и «продакшена», но разрабатываются немного по другому.

CVS, как и большинство систем контроля версий, имеет концепцию «ветвей». Это легко понять, но немного сложно объяснить. По сути, когда вы «разветвляете» файл или набор файлов (или целое дерево каталогов), вы создаете новую версию файла, которая существует параллельно с основной версией. Когда вы вносите изменения в основную версию, это не влияет на ответвленную версию. В то же время вы можете вносить изменения в ответвленную версию, не влияя на основную.

Во FreeBSD обычно есть два активных направления разработки; одно называется «-CURRENT» и является версией «для разработчиков», а другое называется «-STABLE», что ближе к версии для конечного пользователя. Обе версии, конечно, находятся в стадии разработки, и обе готовы для использования. -STABLE, как правило, получает исправления багов и обновления безопасности, которые в первую очередь протестированы сначала на -CURRENT. -CURRENT, в свою очередь, получает новые функции и большие архитектурные изменения. Следует отметить, что название ветвей не обязательно описывают их суть; -STABLE обычно является «стабильной» версией не в плане «надежности», термин «стабильный» больше относится к тому факту, что в самой кодовой базе не было внесено серьезных изменений.

В мире Linux разработка релизов Debian следует примерно по тому же пути. Есть «стабильный» выпуск», который в основном получает только основные исправления. Это примерно то же самое, что и FreeBSD-RELEASE. Затем существует «тестовая» версия, которая получает больше новых функций и возможностей, но только после того, как эти нововведения тестировались и не имеют серьезных проблем. Это похоже на ветку FreeBSD-STABLE. Далее «нестабильный выпуск», в котором разрабатываются новые функции и появляются новые версии пакетов. Это соответствует ветке FreeBSD-CURRENT. (Примечание: я не очень хорошо знаю процесс выпуска Debian; это мое понимание и видение, которое может быть совершенно неверным. Надеюсь, кто-нибудь пришлет мне правильную информацию, если я ошибаюсь)

Я повторяю, потому что это важно; это параллельные ветки разработки. Не релизы. Это отдельные направления разработки, в каждой из которых кодовая база меняется от дня ко дню, от часа к часу и часто от минуты к минуте. Если я возьму -STABLE сейчас и -STABLE завтра, они, скорее всего, будут отличаться. Однако, поскольку разработка находится под контролем версий, то я могу сказать что-то вроде «Дай мне -STABLE по состоянию на 23:30 13 октября 2003 года» и всегда буду получать один и тот же код.

На самом деле, это и есть релиз; зафиксированная кодовая база в определенный момент времени одной из веток разработки. Например, то, что мы называем «2.2.6-RELEASE», на самом деле является просто зафиксированным кодом того, как ветка 2.2-STABLE выглядела 24 марта 1998 года. 25 марта она называлась «2.2.6-STABLE», хотя практически ничего не изменилось. Далее релиз продолжает называться «2.2.6-STABLE» до 21 июля, после чего были внесены изменения и код в очередной раз был зафиксирован и получил название «2.2.7-RELEASE». И так далее.

Можно заметить, что у веток разработки так же есть нумерация. Есть 2.1-STABLE, и 2.2-STABLE, и 3-STABLE, и 4-STABLE. Чтобы понять из-за чего это, посмотрим, откуда берутся ветки. В какой-то момент был 3-CURRENT. В терминах CVS это было основное дерево; не ответвление, а основное направление разработки. В какой-то момент было решено начать готовить это направление к релизу, поэтому кодовая база была зафиксирована и для этой ветки было установлено название 3.0-RELEASE. В это время ветвь 3-CURRENT была все еще -CURRENT и одновременно существовала версия 2.2-STABLE, разработанная ранее. Когда настало время для 3.1-RELEASE, было решено, что пришло время создать ветку 3-STABLE. Поэтому было создано ответвление от 3-CURRENT и названо 3-STABLE, а 3-CURRENT был переименован в 4-CURRENT. То же самое происходило (примерно), когда 4 стало -STABLE и -CURRENT стало 5, и то же самое произойдет снова, когда 5 станет -STABLE и -CURRENT станет 6. Иногда перед выпуском изменяется только минорная версия x.0 -STABLE. 5.0-RELEASE это сохраненное состояние 5-CURRENT.

Вы заметите, что несмотря на то, что 4.x или 4-STABLE все еще (пока что) находятся в «продакшене», версия 3-STABLE все еще существует, хотя и не получает никаких изменений в течение длительного времени. В этом отношении 2.2-STABLE и 2.1-STABLE так же существуют, хотя они не получили никаких изменений еще дольше. Условно, -STABLE без номера относится к последней -STABLE ветке. На самом деле, путаница возникает только тогда, когда новая ветвь только что была создана, поэтому многие люди все еще используют предыдущую.

Также обратите внимание, что 5.1-RELEASE разработан до 4.9-RELEASE. И 5.0-РЕЛИЗ был до 4.8-РЕЛИЗ. Это то самое время, когда одна ветка находится на этапе завершения, а другая ветвь - на этапе создания, в таком случае все становится по-настоящему запутанным. Именно тогда возникает разница между -CURRENT и -STABLE. Чтобы сделать очень грубую аналогию, 5-CURRENT похож на Linux 2.5, а 4-STABLE - на Linux 2.4. Но до этого 4-CURRENT был похож на Linux 2.3, а в будущем 5-STABLE будет похож на Linux 2.6. Конечно, это не идеальная аналогия, отчасти потому, что мы говорим о полной системе со всеми ее частями, а не только с ядром. Но это достаточно близко, чтобы понять.

Так. Что это все значит? Возможно, немного. Но на этом фоне, возможно, вы сможете лучше понять, что происходит, когда и что означают названия веток и номера выпусков.

Еще немного информации о -CURRENT vs -STABLE доступно в [руководстве FreeBSD](#). В документации также есть [статья](#) о подготовке релизов.

Релизы устаревают

Итак, теперь вы лучше понимаете, откуда берутся релизы, и что это за ерунда: -CURRENT и -STABLE. Но все релизы со временем устаревают, и их нужно обновлять. Итак, давайте перейдем к обновлению.

Обновление базовой системы

В результате того, что «базовая система» BSD разработана как одно целое, можно с легкостью получить весь ее исходный код. А благодаря архитектуре, можно выполнить любую команду на верхнем уровне, чтобы скомпилировать все что угодно. Для большинства пользователей BSD это нормальный способ обновления; В Linux обновление немного сложнее: изменения в исходный код могут быть внесены кем угодно, обновление происходит дольше, так как они должны быть скомпилированы.

Впрочем, необязательно устанавливать самые последние обновления, существует возможность установить обновления, например, за прошлую неделю. Как правило, весь процесс выполняется в четыре этапа. Вы начинаете с команды «make buildworld», которая компилирует все пользовательское окружение, затем «make buildkernel», которая компилирует ядро. Затем вы делаете глубокий вдох и делаете «installkernel» для установки нового ядра и в конечном итоге - «make install world» для установки нового пользовательского окружения. Каждый шаг автоматизируется с помощью Makefile.

Конечно, я опускаю множество деталей. Такие вещи, как конфигурация ядра, объединение файлов конфигурации системы, очистка системы... Если вы хотите прочитать об этом, обратитесь к [Руководству FreeBSD](#) и в частности к разделам по обновлению, сборке и настройке ядра, или к другим доступным формам документации. Но такие вещи довольно быстро запоминаются если вы делаете их несколько раз. Действительно, процесс обновления системы сводится к этим четырем командам. Я нахожу это намного проще, чем разрешать перекрестные зависимости и изменять версии библиотек и тому подобное в миллиарде бинарных пакетов.

Приведенная информация в основном относится к FreeBSD. NetBSD использует другую модель для сборки системы. OpenBSD гораздо больше склоняется к переустановке, по крайней мере для серьезных изменений версии.

Дополнительное программное обеспечение

Ну, это было легко. Но как насчет всех этих дополнительных пакетов и программ? Как мы управляем ими? Давайте поговорим об установке и обновлении коллекции портов.

Обновление коллекции портов

Дополнительно установленные вами пакеты и программы так же нуждаются в периодическом обновлении после первичной установки; путем установки пакета более новой версии или создания порта более новой версии. Само дерево портов также хранится в CVS, как и дерево исходных текстов, поэтому вы можете обновить его через CVS или CVSup, как и дерево исходных текстов. Также обратите внимание на утилиту portupgrade, которая представляет собой набор скриптов, написанных на Ruby, чтобы упростить обновление уже установленных портов.

Дерево портов представляет огромную структуру. Каждый отдельный пакет имеет свой собственный каталог, который содержит ряд файлов. И у каждого есть Makefile, который определяет некоторые переменные,

относящиеся к этому пакету, например, адрес для скачивания, номера версий, специальные инструкции для autoconf и список зависимостей. Так же сюда включены любые исправления, которые могут понадобиться для компиляции или запуска. Вот из этого и состоит дерево портов.

В гигантской паутине очень сложных Makefiles есть все виды магии. К счастью, вам не нужно ничего знать об этом. Все, что вам нужно знать, - это то, что если вы запустите make, он автоматически извлечет исходный код для этого пакета, автоматически исправит его, автоматически сконфигурирует его с использованием любой необходимой магии и автоматически скомпилирует. И если для этого требуются другие пакеты, он автоматически проверит, установлены ли они, а если нет, то он автоматически соберет, скомпилирует и установит их. Затем, когда вы запустите make install, он автоматически установит его и автоматически зарегистрирует как пакет, чтобы позже вы могли обновить или удалить его с помощью инструментария управления пакетами.

Это чертовски сложно в деталях, но удивительно просто в использовании.

Или вы можете использовать бинарные пакеты. Здесь вы прежде всего будете использовать команду pkg_add. И если вы используете его с ключом -r, он автоматически попытается получить запрошенный пакет и его зависимости с зеркал FreeBSD.

Множество дополнительной информации обо всем процессе доступно, как всегда, в справочнике; в частности, [глава](#) о портах и пакетах.

Философия

Это тоже не сложно. Несмотря на то, что существует еще больше технических деталей, давайте отойдем от них перейдем к различиям в философии обсуждаемых систем.

Философия

Основная часть

Теперь та часть, которая мне нравится. Много обобщений без конкретных фактов и отсутствие каких-либо исследований. Я собираюсь обсудить здесь реальные и воображаемые философские различия, которые как и вызывают, так и вызваны некоторыми техническими и организационными различиями, которые мы обсуждали ранее. Поймите, я должен подчеркнуть, что многое из обсуждаемого является очень общим. Обе системы часто не «следуют правилам» или не соответствуют их же собственным ожиданиям и заявлениям. Это скорее вопрос склонности к исключительной реализации. Я просто говорю это сейчас, поэтому мне не нужно продолжать квалифицировать и переквалифицировать каждое высказывание, которое я сделаю.

Хаос против Порядка

Методология Linux - это живое воплощение хаоса, тогда как методология BSD гораздо больше относится к контролю и порядку. В значительной степени это правда. Linux вырос из опыта программирования в свободное время, а BSD вырос из контролируемого инженерного образования. Конечно, как и BSD-код может быть написан энтузиастами в выходные, так и много профессиональных программистов, занятых полный рабочий день, могут разрабатывать Linux. Но все равно ощущается раскол между этими системами.

Мы уже обсуждали архитектуру; BSD создает единую «базовую систему», в то время как дистрибутивы Linux собраны из уже существующих частей. Естественно, метод BSD склонен упорядочивать вещи, в то время как метод Linux практически пребывает в полном хаосе. Это не значит, что хаос изначально плох, или порядок по сути хорош. Это просто разные вещи.

Linux, как правило, будет нацелен на более новые версии программ и пакетов, в отличие от BSD. При этом четкое разделение на «базовую систему» и «порты» в BSD, а также структура самого дерева портов упрощают использование нескольких параллельных версий одних и тех же пакетов и программ. Иногда даже возможно установить несколько версий одновременно. В Linux, не имея такого разделения, очень трудно иметь параллельные версии и вместо этого почти всегда требуется одна «благословенная».

Компиляции исходного кода также облегчает работу с несколькими версиями. Например, PHP должен быть скомпилирован по-разному, в зависимости от того, используете ли вы Apache 1.3 или Apache 2. С помощью пакетов из исходного кода, таких как порты, я могу определить переменную окружения при компиляции и установке PHP, чтобы указать, использовать ли Apache 1.3 или Apache 2. С бинарными пакетами у вас должно быть два отдельных пакета, что рано или поздно приведет к путанице.

Правда или Ложь

Разницу можно также увидеть в подходе к разработке. BSD всегда уклоняется от «хакерских» решений на коленке, поэтому любые изменения четко запланированы и спроектированы. Поэтому, с одной стороны, с таким подходом разработка стремится к лучшему, а с другой, если архитектура один раз была упрощена и сделана абы как, то она никогда не сможет избавиться от этих изменений. Конечно, вы скажете, что любые изменения могут быть временными. Но вы знаете, что нет ничего более постоянного, чем временное. Единственный способ избежать такого подхода - это изначально не сделать первый шаг. Это как налоги; когда в последний раз вы видели новый налог, который сразу же отменили?

Поэтому Linux интегрирует поддержку аппаратного обеспечения раньше, чем BSD. Но когда это происходит в BSD, это работает и это стабильно. Драйверы Linux, как правило, имеют гораздо больше различий, потому что они разработаны ранее, в то время как BSD имеет очень жестко контролируруемую «базовую систему» и может быть очень свободен при настройке дополнительного программного обеспечения, поскольку контроль происходит со стороны пользователя. У Linux очень слабая и плавкая связь между ядром и пользовательским пространством, но пользовательское пространство в целом, из-за отсутствия разделения на «базовую систему» и дополнительные пакеты, требует гораздо больше работы и ресурсов для поддержания согласованности, что предъявляет гораздо более высокие требования к настройке и поддержке различных версий пакетов. Дополнительные сложности в большинстве дистрибутивов так же вызваны использованием бинарных пакетов вместо исходного кода.

Аудитория

Linux, с ее хаотичной разработкой, естественно, привлекает типы личностей, отличных от BSD с его академическим и строгим опытом. Лично я думаю, что это сдерживает Linux в некоторых направлениях, которые становятся все более очевидными в последнее время. Случайный хаос, безусловно, имеет свои преимущества, но не избавлен и от недостатков.

Последствия дизайна

Методика поэтапной сборки в Linux позволяет (теоретически) просто обновить один компонент. В BSD легко обновить отдельные части, которые являются «надстройкой».

Например, в случае BSD, OpenSSL импортируется и поддерживается в «базовой системе», но я не могу просто обновить его самостоятельно; Я должен обновить всю «базовую систему». И это включает в себя ядро, а также ls и gzip и gcc и less, а также ssh, sendmail и BIND. (Конечно, вы всегда можете установить разные версии sendmail и BIND, или Postfix и djbdns через порты, и просто запустить другую версию вместо базовой).

В теории это проблема. На практике я обнаружил, что обычно все гораздо легче. Я не могу сосчитать, сколько раз я блокировал себя из удаленного Linux, обновляя OpenSSL, затем пытаюсь обновить или перезапустить ssh из-за другой версии OpenSSL. С интегрированной системой это не проблема, поскольку они (ssh и OpenSSL) будут автоматически синхронизированы. А с их сборкой из исходного кода они все равно будут автоматически синхронизированы, даже если они не были интегрированы вместе в «базовую систему».

В результате этой и подобных ситуаций не так-то просто поддерживать системы Linux в актуальном состоянии. Отдельные выпуски данного дистрибутива гораздо более независимы друг от друга, поэтому сложнее превратить вашу систему RedHat 7.3 в систему RedHat 8. Новые пакеты для 7.3 создаются и обновляются в течение определенного периода времени, и часто действительно новые версии пакетов никогда не поступают. И если вы пытаетесь использовать пакеты из другой версии вашего дистрибутива или из другого дистрибутива, вы часто столкнетесь с отсутствующими зависимостями и проблемами с различными версиями других программ. И как бы трудно ни было поддерживать систему в актуальном состоянии, еще сложнее привести их в актуальное состояние после долгого периода отсутствия обновлений.

С другой стороны, у меня есть BSD-системы, которые я установил в 1997 году и они всегда были (и остаются на сегодняшний день) с последним ядром, последней версией Apache, последней Mozilla и самой последней версией всего. Однако в системе Linux вы, как правило, пытаетесь обновлять необходимые пакеты, тогда как другие отстают все дальше и дальше, пока в конечном итоге вы просто не переустановите последнюю версию, чтобы обновить ее. Эти методологии привлекают довольно разных администраторов.

Мифы

Мы потратили много времени на обсуждение различий между BSD и Linux. Мы потратили много времени на разговоры о том, как работает BSD. И теперь мы как бы раскрыли, что такое BSD. Итак, давайте кратко рассмотрим, чем не является BSD, и развеем некоторые мифы.

Мифы

Вступление

Существует множество мифов и заблуждений, а также «общеизвестных» и «общепринятых мнений» о BSD. Я всегда немного удивляюсь тому, как быстро некоторые люди из мира Linux заикливаются на таких чрезмерных упрощениях и давних заявлениях о BSD, тем более что они тратят столько усилий, крича о людях, которые делают то же самое в отношении Linux. Ну что ж. Давайте разрушим несколько стереотипов.

Аппаратные средства

«BSD не поддерживает стандартное оборудование».

Поддерживает ли Linux аппаратное обеспечение, которого нет у BSD? Наверное. Это имеет значение? Только если у вас есть это оборудование.

Бьюсь об заклад, Windows поддерживает аппаратное обеспечение, которое поддерживает Linux. В этом отношении MacOS, вероятно, поддерживает аппаратное обеспечение, которое не поддерживают остальные операционные системы. BSD поддерживает наиболее распространенное аппаратное обеспечение, которое вы используете на сервере, на рабочей станции... Существуют пробелы, но многое меняется от выпуска к выпуску, как и любой другой системе.

Например, поддержка видеокарт вряд ли когда-либо будет востребована в какой-либо документации BSD, а в документации Linux об этом много говорится. Это кажется странным, пока вы не поймете, что в мировоззрении BSD ОС не поддерживает ни одну из этих видеокарт напрямую; есть X Window, который представляет собой отдельный пакет. Таким образом, вы можете использовать любую видеокарту под BSD, которую вы можете использовать под Linux, поскольку ни ядро BSD, ни ядро Linux не поддерживают видеокарту напрямую. С другой стороны, более глубокая работа с 3D и DRI требуют более прямых аппаратных связей и большего количества «храбрости» в самом ядре. Конечно, я не слежу за этой сферой, поэтому я даже не знаю, каково текущее состояние мира во FreeBSD, не говоря уже о Linux. Может быть, BSD не имеет поддержки наравне с Linux. Я не знаю этого, вероятно что-то может измениться со временем.

Но большая часть оборудования проста. Большинство обычных контроллеров и запоминающих устройств IDE и SCSI работают просто отлично. Даже большинство контроллеров RAID поддерживаются в некоторой степени. Большинство сетевых карт, проводных и беспроводных, большинство звуковых карт, некоторые крипто-карты ...

Все не так сложно. Вам не важно, какое оборудование поддерживает ОС, если оно поддерживает то, что у вас есть. Посмотрите списки поддержки оборудования и/или попробуйте загрузить его. Вы можете быть удивлены.

Если есть сомнения, проверьте списки поддерживаемого оборудования.

Доступность программ

«Но в Linux больше программ, чем в BSD!»

Большинство из этих «программ», которые вам так интересны, - это в любом случае вещи с открытым исходным кодом или с доступным исходным кодом. Если код переносимый и качественный, то с вероятностью 95% или больше он может быть скомпилирован в любой POSIX-совместимой системе. Тем более просто посмотрите «коллекцию портов»; там более 10000 программ и пакетов.

Конечно, есть много программного обеспечения, которое не будет компилироваться ни на чем, кроме Linux. Иногда это происходит потому, что для этого действительно требуются средства, которые есть только в Linux, или выполняются вещи, имеющие значение только для Linux. Иногда это означает, что вам нужно взять и найти автора, потому что он сделал что-то совершенно несовместимое из-за злобы или лени. Есть люди, которые делают то же самое и в BSD, но быстро растущее сообщество Linux, в сочетании с количеством людей, пишущих программы и имеющих меньший опыт в разработке программного обеспечения, делают это гораздо более заметным.

Конечно, есть вещи, которые не будут кроссплатформенными, некоторым требуется лишь небольшая работа по переносу, некоторые даже не имеют никакого смысла в других системах (Когда кто-нибудь когда-либо переносил Microsoft Defrag на Linux?) Но если программа полезна, то возможно, она либо портирована или есть какой-то эквивалент или ее аналог.

И еще есть программы, которые не доступны в виде исходного кода. Браузеры Netscape, Opera, некоторые офисные программы, RealPlayer и т. д. Большинство из перечисленных также имеют версии BSD, но даже в этом случае, в частности, FreeBSD и (я полагаю) все BSD имеют уровень эмуляции Linux, который обеспечивает двоичную совместимость. Это не всегда помогает, особенно, когда что-то ломается на низком уровне реализации. Но это работает на удивление часто. Бинарные программы Linux могут быть просто подобраны и запущены в системе BSD

(по крайней мере, между системами i386, а иногда и на других архитектурах). Более подробная информация выходит за рамки данного эссе, но если вам нужна дополнительная информация, вы можете прочитать об этом в справочнике.

Стоит взять за правило поиск в коллекции портов. Скорее всего, вы найдете то, что ищете. А если нет, то, вероятно, среди этих 10000 и других вариантов есть что-то, что будет отвечать вашим требованиям.

Популярность

«Но Linux более популярен».

И что? Windows еще более популярна, значит нужно использовать ее?

Обычно аргумент популярности на самом деле означает такие вещи, как «легче найти поддержку», или он связан с проблемой доступности программ. Но есть много мест и людей, предоставляющих коммерческую поддержку BSD, как и сообщества в различных списках рассылки и новостных группах. Это можно противопоставить пользователям Linux, которые приводят аргумент «Кому нужно платить за поддержку? Посмотрите, что мы можем получить бесплатно!»

Юзабилити

«BSD сложно использовать, он более продвинутый, более сложный, менее удобный для пользователя».

Ну что я могу противопоставить этому? Что насчет этого «сложнее»? Конечно, он отличается от Linux, но что с того? Во многих отношениях я считаю, что BSD гораздо более логичен и понятен. Вот где все усилия и снобизм окупаются. Linux тоже намного «отличается» и «сложнее в использовании», и «более продвинутый», и «более сложный», и «менее удобный для пользователя», чем Windows. BSD не сложнее, просто заставляет думать немного по-другому.

Философия BSD всегда считала более важным позволить продвинутому пользователю делать больше, чем новичку. Теория, конечно, заключается в том, что новички становятся продвинутыми пользователями очень быстро (год или два), в то время как продвинутые пользователи используют систему долгое время (десятилетия). Так что, вероятно, в систему встроено меньше «пушистых и мягких» внешних интерфейсов. С другой стороны, пакеты дополнений более дружелюбны. Если в KDE есть какая-то пользовательская справка или учебник или что-то еще встроенное (я не знаю, я им не пользуюсь), то оно будет и на BSD, поскольку KDE - это KDE, где бы вы его ни использовали. Так что отсутствие дружелюбности не такая уж и большая проблема, как кажется.

Элитарность

«Пользователи BSD - это группа элитарных эгоистичных грубых снобов».

Ага. И мы гордимся этим. :-)

Вы не можете охарактеризовать целое сообщество одной фразой. В сообществе FreeBSD есть множество людей, которые не хотят слышать о вашей проблеме, если вы сами не изучили исходный код и не нашли точное место ее возникновения, и лучше это будет настоящая проблема, а не глупая ошибка пользователя! В сообществе FreeBSD также множество людей, которые готовы выпрыгнуть из своих штанов, чтобы помочь вам решить самую простую проблему.

Ответы типа «RTFM», конечно, случаются. Но они случаются и в сообществе Linux. Никто не любит, чтобы кто-то задавал вопрос, который описан в четвертой главе FAQ, который все должны были прочитать. Документация FreeBSD охватывает множество тем и довольно подробна. И поскольку система разрабатывается централизованно, большая часть документации также разрабатывается централизованно и находится прямо на сайте FreeBSD.

Никто не хочет пытаться ответить на вопрос, где спрашивающий также не дает никакой полезной информации. И когда вы ответили на один и тот же вопрос, который уже был в документах сто раз и попросили одну и ту же дополнительную информацию о проблеме сто раз, то вы будете склонны довольно резко относиться к таким вопросам в сто первый раз. Это не совсем гуманно, но каждое техническое сообщество имеет дело с тем же самым.

Вполне возможно, что BSD в силу своей более технической и инженерной родословной, в конечном итоге сильнее влияет на мировоззрение и философию своей аудитории, чем другие операционные системы. Я не уверен, что это действительно так, мне кажется, что это просто более заметно. Одним из последствий более централизованной и организованной разработки является то, что конечными пользователями, в большинстве случаев, являются профессиональные системные администраторы и программисты. Также большинство разработчиков, имеют одинаковое «пользовательское пространство». В случае с Linux, у вас есть отдельный список рассылки linux-kernel, который есть практически у всех разработчиков ядра и сопровождающих, а у разработчиков и сопровождающих

make, cron, locate и tar существуют собственные списки рассылки где-то еще... Большинство обсуждений во FreeBSD так же происходят либо в списках рассылки, либо в одной из нескольких групп новостей. Неопытные пользователи нередко используют неправильный список рассылки для вопросов.

Итак, являются ли пользователи BSD элитарными? Хорошо, может быть. Может быть, это утверждение было более распространенным несколько лет назад; подобные ярлыки сохраняются надолго, даже после того, как причины их существования стали историей. Лично я думаю, что элитарность BSD намного более заметна из-за более тесного сообщества.

Ну, долго еще?

Давайте попробуем подвести итоги.

Заключение

Почему я выбираю BSD?

BSD просто работает. Когда я устанавливаю систему, система работает. Когда я обновляюсь, обновление работает. Когда я устанавливаю дополнительные пакеты, они работают.

BSD поддерживает практически все устройства, которые я использовал за последнее десятилетие. Поддержка устройств просто работает и не пропадает от одной версии к другой.

Я могу выбирать - использовать проверенные версии «-STABLE» или разрабатываемую ветку -CURRENT, то есть могу выбрать как близко подойти к краю, за которым возможны программные ошибки. И даже самая последняя разрабатываемая версия обычно работает просто отлично. Я могу взять версию трехлетней давности и обновить ее до текущей без лишних проблем.

Любая проблема, которая связана с безопасностью, стабильностью, производительностью или даже эстетикой может быть исправлена. Из-за методологии разработки, я могу получить эти исправления в тот же вечер. Когда будет добавлена новая функция, я могу получить ее прямо сейчас, а не ждать недели, месяца или полгода до следующего выпуска. Я вижу в списке рассылки все изменения по мере их внесения. Или, если что-то сломается, я могу исправить это сам и сразу же отправить исправление кому-то, кто может сразу же включить исправление в обновление. Я делал это раньше, я сделаю это снова, как и тысячи других людей.

Компоненты системы разработаны, чтобы соответствовать друг другу. И это не случайность, что все работает слажено, потому что изначально система спроектирована цельно и едино.

В то же время случаются ошибки и недоработки, бывает программное обеспечение работает необъяснимым образом, сообщения об ошибках могут вводить в заблуждение, а иногда ломаются вещи, которые работали вчера. Читая списки рассылки и я вижу каким образом все эти проблемы решаются. Сначала проблема обнаруживается, заинтересованные и компетентные люди обсуждают ее, предлагают разные решения, некоторые решения отклоняются, другие проверяются и согласуются, и в итоге я вижу, что исправления помещены в основной репозиторий.

Это всего лишь вопрос уровня согласованности, прозрачности и удобства использования, который превосходит все, что я видел. Путь развития BSD нацелен на технические достоинства и правильные решения, а не только на новомодные фишки и работоспособные хаки. И я думаю, что это единственный способ поддерживать операционную систему в долгосрочной перспективе.

Почему вы должны установить и использовать BSD?

Ну, я не знаю. То, чем вы пользуетесь сейчас, отвечает вашим потребностям? Тогда вам, вероятно, не нужно что-либо менять.

Существует множество статей и рассказов о различиях между разными операционными системами, например, почему X работает лучше, чем Y. Я не хочу писать еще одно. Если у вас есть время, желание и ресурсы, чтобы попробовать BSD - действуйте. И я хочу, чтобы это эссе помогло вам понять, почему существуют эти различия и каковы результаты этих различий.

Больше информации

Все сайты BSD (FreeBSD, NetBSD и OpenBSD) имеют обширную документацию. Поскольку существует понятие как «базовая система», все что с ней связано, может быть достоверно и подробно задокументировано в одном месте.

Документация описывают сильные и слабые стороны отдельных систем, как их устанавливать, как их обновлять, как их администрировать ... Все, что вам нужно.

В частности, в документации по FreeBSD статья об объяснении принципов BSD похожа на это эссе в том смысле, что она описывает BSD и его исторический контекст. Есть все виды документации, от вводных документов, описывающих основы систем Unix-подобных систем и FreeBSD до подробных описаний внутренней работы ядра.

И, конечно же, Руководство FreeBSD и FAQ по FreeBSD наполнены информацией, о которой вы даже не подозревали.

У BSD также есть списки рассылки и группы новостей, доступные для того, чтобы задавать вопросы. Обычно в них полно отзывчивых людей, которые прошли тот же процесс обучения, что и вы. Смотрите отдельные сайты для получения дополнительной информации.

Конец

Может быть сложно понять BSD, когда вы привыкли к Linux, потому что вас может сбить с толку множество базовых различий. Хотя обе системы удивительно похожи благодаря своему наследию Unix, стратегии разработки и внедрения сильно различаются. Я надеюсь, что помог вам лучше понять, как работают системы BSD и что их объединяет.

Пожалуйста, не стесняйтесь, напишите мне, если в этом эссе есть что-то, что вы не понимаете или думаете, что можете улучшить;

Компьютеры это весело. Наслаждайтесь.

Ответы на вопросы

Я также добавил еще одну страницу в конце, содержащую некоторые ответы на критику, которая была направлена об этом эссе.

Ты не прав!

Я получил огромное количество отзывов на это эссе, гораздо больше, чем ожидал. Многие люди написали мне, что им действительно понравилось или что эта информация действительно помогла им. Многие люди также написали предложения с улучшениями и разъяснениями, многие из которых я использовал и за которые я благодарен. Другие указали на различные ошибки.

Я попытаюсь ответить на некоторые критические замечания. Конечно, это не удовлетворит урожденных спорщиков (флейм - это весело!), но может помочь прояснить мою позицию.

«Вы предвзяты и снисходительны»

Ну, я предвзят. Мне очень нравится FreeBSD и в большинстве систем Linux, над которыми я работал, есть много вещей, которые меня бесят.

Но это не делает недействительными заявления, которые я сделал. Каждый в какой-то степени предвзят. Честно говоря, я бы не стал доверять высказываниям тех, кто был абсолютно объективен; обычно, чтобы быть абсолютно беспристрастным и объективным, вы не должны понимать, о чем идет речь. И это не значит, что я всегда знаю, о чем говорю, но подождите, мы общаемся в сети :)

Что касается снисхождения, то я стараюсь не допустить, чтобы оно попало в мои записи. Все, что я могу сделать, это извиниться и пожать плечами. Я думаю, что за исключением предвзятости и снисходительности, здесь в моих записях есть некоторая полезная информация. Если вам не нравится мой тон или стиль настолько, что вы не можете получить информацию, я искренне надеюсь, что вы найдете ее где-то еще.

«Почему вы продолжаете оскорблять Linux, называя его «взросленным» или «хаотичным»?

Вот этого я не понимаю. Это не оскорбление, люди! Это самая большая сила Linux !!! Это то, чем можно гордиться!

Хаос означает, что вы движетесь во все стороны одновременно. Linux (и его различные дистрибутивы) движутся в самых разных направлениях, практически без реальной координации. Это означает, что Linux окажется во многих местах, где BSD, с его более осторожным и упорядоченным развитием, будет добираться гораздо медленнее (если вообще доберется). Многие из этих мест могут быть отвратительными и ужасными, но некоторые из них станут абсолютными жемчужинами, которых вы никогда не сможете достичь другим способом.

Таким образом, Linux может достигать совершенно разных целей и огромное количество людей и проектов, продвигающихся в самых разных направлениях, практически гарантирует вам удачу. Но не можете достичь такого результат, не приняв хаос и не сделав множества неправильных поворотов на пути к цели.

Linux также вырос очень органично и развивается в разных направлениях с помощью различных групп разработчиков. Никогда не было никакого вдохновителя, который планировал, куда пойдет Linux. Все произошло почти случайно, это все равно, что идти по дороге и просто смотреть, куда она вас ведет, а не пытаться добраться до определенного места.

Когда я говорю, что Linux взращен и хаотичен, это не оскорбление. Это высокий комплимент!

«Почему вы сравниваете BSD с Foo Linux, когда Bar Linux имеет функцию X, которая намного лучше?»

Ну, для начала я пытаюсь объяснить, а не убеждать. Я знаю, что уже говорил это десятки раз, но некоторые люди, кажется, не верят мне. Я хочу противопоставить и показать, в чем заключаются различия, и почему они есть. И я могу сравнивать только с тем, что я знаю. Я не знаю все о Linux. У меня нет глубоких знаний о каждом дистрибутиве. Я даже не знаю досконально NetBSD и OpenBSD.

У меня есть хорошее знание FreeBSD и общее понимание философии проектирования BSD. У меня есть достаточный опыт работы с несколькими дистрибутивами Linux. Я получил много хороших комментариев от людей о дистрибутивах Linux, таких как Gentoo и Debian, а также о других BSD. И я попытался интегрировать большую часть этой информации, чтобы лучше продемонстрировать и объяснить различия.

Извините, если я игнорирую флагманское направление вашего любимого дистрибутива. Если это очень похоже на то, что делает BSD или иным образом позволяет лучше и яснее объяснить, как BSD работает по отношению к Linux, напишите мне об этом с достаточным количеством информации, чтобы я мог дополнить это эссе. Если это просто добавляет контраст между дистрибутивами Linux, не проливая больше света на то, как работает BSD, то это не очень поможет.

Оставайтесь в курсе

Эссе будет дополнено, когда я получу больше отзывов и предложений.