
Ядро Linux 2.6

Краткое описание новинок ядра 2.6

Джозеф Праневич, Воскресенье, 18 Июль 2004, 21:11

Казалось бы, только вчера мы запускали наши первые Linux-системы с ядром 2.4, время летело, контора писала, вот уж озимые заколосились и? команда разработчиков ядра приблизилась к завершению работы над новой версией ядра - 2.6. В этом документе будут описаны новые особенности ядра 2.6 (с сильным уклоном к i386 порту Linux.). В отличие от объявлений предварительных выпусков закрытого ПО, все особенности, описанные здесь, доступны уже сейчас (одни более, другие менее свободны от багов) в экспериментальном ядре Linux 2.5 (по традиции Linux, четно пронумерованные ядра "" и предназначены для готовых систем, в то время как нечетные ядра - для смельчаков)). Учитывая сказанное следует иметь в виду, что некоторые из особенностей, описанные здесь, могут быть удалены или помечены как "экспериментальн" в актуальных выпусках ядра 2.6. Однако на текущий момент развитие ядра имеет статус "feature-freeze", т.е. никакие новые возможности более не добавляются в ядро, и это радавает нас, патаму ша карашо, ибо финальный релиз не будет сильно отличаться от того что описано тут. Также следует учесть, что некоторые из "" особенностей, обсуждаемых здесь, возможно, были портированы задним числом в Linux 2.4 после их появления в Linux 2.6, официально ли или дистрибутором, опять же ли.

При создании данного документа возникли некоторые сложности, точный номер ожидаемой версии Linux еще не был определен. Наиболее вероятный номер - v2.6, но были предложения, чтобы v3.0 стал следующим номером из-за включения NUMA и MMU-less поддержки в будущей версии (подробнее об этом позже). Для удобства, этот документ будет всегда, что касается будущей версии ядра, ссылаться на него как на v2.6 и в настоящем времени.

То что вы читаете - первый проект этого документа, датированный 13.07.2003 и основывается на экспериментальном ядре 2.5.75. Будут выпущены и другие версии этого проекта, по мере завершения работы над ядром, но этот документ "" не будет закрыт до выхода финального релиза ядра. Поскольку это первый черновик, то могут быть ошибки или упущения; если вы найдете их, сообщите мне, я буду счастлив устранить их (а иначе ты будешь несчастен до конца дней своих?)).

Скоро сказка сказывается...

Ядро Linux было зачато в 1991 году Линусом Торвальдсом как Minix-подобная операционная система для его кампика, i386. Изначально Линус хотел назвать проект Freax, но нам всем крупно повезло, потому что он передумал (спрашивается: **ПОЧЕМУ???** - двойной смысл вопроса))).

Первый официальный выпуск Linux 1.0 был представлен в марте 1994-го, официально он поддерживал только i386 и только однопроцессорные персональные компьютеры. Linux 1.2 был выпущен в марте 1995-го и стал первой версией, которая официально поддерживала различные архитектуры (в частности Alpha, Sparc, и Mips). Linux 2.0 появился в июне 1996-го и поддерживал много новых архитектур, но что наиболее важно, это была первая версия, которая поддерживала многопроцессорные машины (SMP). Linux 2.2 вышел в январе 1999-го как дополнительное усовершенствование, которое существенно повысило производительность

Ядро Linux 2.6

<http://www.osrc.info/plugins/content/content.php?content.48>

на многопроцессорных машинах и снова был расширен диапазон поддерживаемых аппаратных средств. И наконец, Linux 2.4 был выпущен в январе 2001-го как следующее основное усовершенствование масштабируемости с SMP, при этом много десктопных особенностей было интегрировано в основную ветку ядра, включая USB, PC Card (aka PCMCIA), встроенный plug-and-play и т.д. Linux 2.6 базируется не только на этих особенностях, но также делает другой "большой шаг" в улучшении поддержки как больших систем (полагаю немеряных мейнфреймов), так и встраиваемых (PDA и прочие embedded Linux`с).

Расширение списка поддерживаемых архитектур.

Одной из многих сильных сторон ОС основанных на Linux (типа GNU ОС, ОС и различные окружение, так часто распространяемые с Linux, что уже не воспринимаются как отдельные объекты) - является то, что они поддерживает широкий диапазон аппаратных средств и платформ. Каждая версия начиная с 1.2 включала поддержку новых типов процессоров и особенностей. Linux 2.6 не является исключением в этой тенденции, и хотя этот факт прямо не влияет на использование Linux на Intel-е, он очень важен для Linux, для его как можно более широкого распространения.

Масштабированиевниз - Linux для Встроенных Систем

Одним из главных направлений роста Linux 2.6 была поддержка множества платформ, в частности посредством включения многих возможностей из проекта uClinux в основное ядро. Проект uClinux (возможно произношение "you-see-Linux", но более верно произносить с греческими символами "mu" (это как это???) - это проект Linux для микроконтроллеров. Много лет это ответвление Linux было основой поддержки многих встраиваемых процессоров, и очень приятно получить более высокую степень интеграции в ядре основной ветки.

В отличие от обычных портов Linux, встраиваемые порты описанные здесь не имеют всех возможностей обычного Linux из-за аппаратных ограничений. Основное различие в том, что процессоры, под которые эти порты созданы, не поддерживают MMU. (В мире Intel, MMU был представлен с i386). В то время как эти Linux системы истинно многозадачны, они не имеют защиты памяти (так что любая программа может разрушить любая другую) и некоторые системные вызовы, имеющие дело с клонированием процессов, заблокированы. И поскольку эти системы не имеют защиты памяти (в самом деле, какая уж тут безопасность), они оказываются бесполезными в качестве многопользовательских систем.

Есть четыре основных линейки встраиваемых процессоров, поддерживаемых Linux 2.6. Первый из этих портов - для новейшей линейки встраиваемых процессоров m68k фирмы Motorola. Эти процессоры выпускаются с именами типа Dragonball и ColdFire и включены в системы и "" платы, изготовленные фирмой Motorola, Lineo, Arcturus, и другими. Большинство пользователей Linux знакомы с этими процессорами как лежащими в основе Palm Pilot-ов, начиная с первого (Palm 1000), и до Palm III. Печально, но поддержка старших m68k процессоров без MMU (типа 68000, используемый в ранних Macintosh) еще не включена в это ПО. Среди новых поддерживаемых встраиваемых платформ такие как серия H8/300 от Хитачи (H8S еще не поддерживается, но вскоре могут быть включены) и NEC v850 процессор.

Акцентирую внимание на том, что это серьезное смещение приоритетов в сторону поддержки систем без MMU под Linux 2.6. Все предыдущие версии Linux были получены,

однако косвенно, ограниченным наследованием начальной разработки Линуса на Intel 80386.

Экстраполируя в этом направлении, будем надеяться что поддержка и других устаревших аппаратных средств будет реализована в будущем (ацтой!!! даешь вечный апгрейд!!!!)) (действительно, есть проекты, которые работают в этом направлении), а также много новых и захватывающих дверей будут открыты? (смысл понятен). В отличие от работы на современных и разрабатываемых/встраиваемых процессорах, перечисленных здесь, поддержка устаревших моделей в значительной степени определяется ""))))) и наиболее вероятно бесполезна для конечных пользователей (и возможно не достаточно важна для включения в официальные релизы Линукса в будущем).

Несмотря на то, что часть uClinux интегрирована, не технически (потому что включает MMU), последние версии Линукса также включает поддержку процессоров Axis Communications' ETRAX CRIS ("Code Reduced Instruction Set - код с уменьшенным количеством "). (В частности поддерживаются ETRAX 100LX и более новые). Фактическая поддержка этих процессоров появилась в процессе разработки ветки 2.4, но данная возможность была представлена после релиза 2.4.0, поэтому заслуживает упоминания здесь. Axis Communications' ETRAX CRIS - это встраиваемый процессор, который используется прежде всего в сетевом оборудовании. ETRAX100 родственник MMU-less процессоров, которые поддерживаются в uClinux, но его поддержка в основной ветке дерева ядер Линукса не была интегрирована.

Opteron Поддержка - 64 бит Линукс для Потребителя

Другой процессор, поддержка которого была интегрирована при развитии ветки 2.4.x, но заслуживающая упоминания здесь - поддержка Линуксом процессоров AMD Opteron (основанных на архитектуре AMD64.) Это новый чип обратно совместим с существующими процессорами-клонами Intel и получивший даже "обратную" поддержку от Microsoft. Станет ли этот процессор или же член Itanium-семейства Intel стандартом де-факто для 64-битных пользовательски продуктов покажет время.

В то время как более поздние редакции ядра 2.4 поддерживали этот процессор, все же были ограничения не позволявшие широкое использование этих систем. Наиболее критичным при использовании в высокопроизводительных системах является то, что каждая задача ограничена 512 Мбайтами памяти.. Кроме того была улучшена поддержка x86 (32-разрядных) программ.

[newpage]

Поддержка "".

В дополнение к нескольким новым процессорным архитектурам, поддерживаемым ядром 2.6., новая версия Линукса включает также новую концепцию, называемую """. Раньше Линукс чаще всего предполагал что тип процессора и тип его аппаратного окружения совпадают. То есть потомки i386 процессоров используются исключительно в PC/AT серверах. В ядре 2.4. это допущение было нарушено в отношении i386, с добавлением поддержки Визуальной Рабочей Станции от SGI, "без " платформы, работающей на чипах Intel (фактически, данный перелом произошел задолго до этого на многих других архитектурах. Например, m68k длительное время поддерживался на Amiga, Macintosh и других платформах). Существенным изменением в ядре 2.6. стало то, что данные возможность и концепция были стандартизованы так, чтобы все архитектуры оперировали ими одинаково и способом, позволяющим четко разделять только те компоненты, которые требуют данного разделения.

С этой стандартизацией приходит две новые платформы поддерживающие i386. Первая - архитектура NCR Voyager. Это SMP система (разработанная ранее и являющаяся сейчас стандартной спецификацией Intel MP) поддерживающая 486-686 процессоры в конфигурации вплоть до 32-х процессоров. Количество проданных конфигураций данной архитектуры довольно мало, и еще не все машины поддерживаются (наиболее старые не поддерживаются). Вторая поддерживаемая архитектура - более широко распространена платформа PC-9800, разработанная фирмой NEC, донедавна практически доминирующая платформа PC в Японии. Оригинальные PC-9800 машины строились на базе 8086 процессорах и дальнейшем их развитии (параллельно с AT-потомками) в итоге выросшем в процессоры Pentium-класса и появившаяся поддержка SMP (естественно поддержка для Линукса ограничена i386 или лучше (PA3BE????)). Эта платформа была абсолютно не известна в США, до тех пор пока Microsoft не выпустила Windows 95, портированный на нее)))))). Выпуск этой линейки был официально прекращен производителем в пользу более "" ПК-шек.

Формализация поддержки Линуксом этих "незначительно" аппаратные типов позволит более легко портировать ОС на другие системы, такие как выделенные хранилища данных и другие компоненты, используемые преимущественно в индустриальных типах процессоров. Для ясности скажем, что нельзя не учитывать эту категорию. Эта субархитектура была выделена потому, что низкоуровневые компоненты системы (такие как IRQ маршрутизация) более или менее различны. Вышесказанное отличается от Линукса, запущенного, например, на X-Vox, потому что не драйвера или другие особенности, а нечто другое отличает эту систему от основных i386 систем. Поддержка X-Vox не является субархитектурой

Масштабирование вверх - NUMA и Большое Железо.

В дополнение к поддержке перечисленных новых типов оборудования, новый выпуск ядра Линукса включает особенности, которые делают более приемлемым его использование на все более крупных серверах (некоторые работают на i386 процессорах, некоторые нет).

Поскольку данная поддержка в новинку для Линукса, многое еще предстоит сделать (в направлении оптимизации). Это область, в которой Линукс очень быстро растет и созревает, и можно ожидать, что Линукс довольно скоро станет здесь сильным соперником.

Важнейшим изменением в этом отношении стала поддержка Линуксом серверов с NUMA. NUMA (или "Non-Uniform Memory Access" - неоднородный доступ к памяти) является шагом в сторону от традиционного SMP в многопроцессорном мире, и это шаг вперед к эффективности систем, имеющих много процессоров. SMP системы были разработаны с некоторыми из тех же ограничений, что и их однопроцессорные коллеги. Одним из существенных ограничений при выборе такого дизайна является то, что он основан на едином пуле памяти, который разделяют между собой все процессоры системы. В многопроцессорных системах это приводит к сильной конкуренции между процессорами, работающими на одной шине памяти, и является узким местом системы. NUMA серверы обходят данную проблему, представляя следующую концепцию: отдельному процессору предоставляется часть памяти, не доступная другим процессорам. Один простой путь (технически довольно корректный) состоит в том, чтобы представить, что вы имеете систему с платами расширения, содержащими процессоры, память и, возможно, другие компоненты (ввода-вывода и др.). Некоторое количество таких плат в системе могут "" общаться между собой, каждый процессор имеет свободный доступ к локальной памяти (кроме того доступ к памяти на плате процессора быстрее чем к памяти на

<http://www.osrc.info/plugins/content/content.php?content.48>
выделенной плате). Во многом новая NUMA архитектура является примером очень сильно
связанного кластера.

Чтобы должным образом поддерживать новые NUMA машины, необходимо было адаптировать Линукс сразу в нескольких направлениях, чтобы новая модель стала эффективной. Для начала был создан API внутренней топологии, позволяющий ядру понять, один процессор или один пул памяти связаны с устройствами ввода-вывода и как они связаны между собой. Опираясь на эту информацию, планировщик процессов Линукса теперь может понять эти взаимосвязи и оптимизировать выполнение задач с наиболее выгодным распределением локальных ресурсов. Кроме того, многие NUMA машины построены таким образом, что имеют "" в линейном адресном пространстве между узлами. Новое ядро решает эту проблему разумным способом. Было сделано множество других внутренних изменений, которые позволят Линуксу работать с этими новыми хай-энд машинами, и в целом - это направление, определяющее дальнейшее совершенствование ядра. В течение следующего года мы можем ожидать новые доработки и усовершенствования поддержки Линуксом этих высокопроизводительных систем.

Внутренняя организация Линукса

В дополнение к поддержке нового оборудования, в ядре 2.6. также усовершенствована поддержка уже существующих платформ. Сюда входит оптимизация, специфичная для процессоров Transmeta Crusoe; Intel Pentium 4 Xeon, Pentium 3-M, и Pentium 4-M, а также мобильные процессоры AMD. Новые версии Линукса также могут обходить проблемы, имеющие место в некоторых Athlon-ax. И хотя большинство пользователей не замечают некоторых проблем, Линукс 2.6. решает проблему, возникающую в некоторых системах с более чем 16-ю процессорами при загрузке.

Hyperthreading

Хотя это не новинка Линукса 2.6., поддержка Hyperthreading появилась еще в ядре 2.4.17 (данная возможность описывается здесь потому, что начальный релиз 2.4. не поддерживал Hyperthreading, а также потому, что с того времени сделано много серьезных изменений). Hyperthreading - это способность одного процессора маскироваться под два (или больше) процессора от ОС. Удивительно то, что Линукс - первая ОС, которая вышла на рынок с поддержкой этой возможности, несмотря на то, что Intel выпустил процессоры с Hyperthreading всего около года назад. Говорят что Майкрософт колеблется в вопросе лицензирования таких процессоров: лицензировать как один или же как два (уже парешили). Открытая модель Линукса (и отсутствие лицензирования за процессор) позволили различным дистрибам стать первыми в поддержке Hyperthreading (ура товарищи!). Естественно процессор претендующий быть "двумя", все же остается одним процессором и производительность растет не сильно.

Одна вещь действительно новая в 2.5. - это планировщик, позволяющий получить выигрыш за счет Hyperthreading всегда. В 2.4. так было не всегда, и в некоторых случаях включение данного режима работы процессора приводило к замедлению работы.

Вопросы Размера - Усовершенствование Масштабируемость

Ядро Linux 2.6

<http://www.osrc.info/plugins/content/content.php?content.48>

Кроме поддержки NUMA, Линукс 2.6. реализовал и другие изменения касающиеся серверов

Intel, стоящих на вершущке "пищевой ". Первое - это полная поддержка PAE ("Physical Address Extension" - расширение физического адреса), которая позволяет новейшим 32-битным x86 системам адресовать до 64 Гбайт ОЗУ, но, в страничном режиме. Кроме того, посредством улучшенной поддержки APIC и других изменений, было существенно улучшено распределение IRQ в многопроцессорных системах.

Во многих других отношениях были расширены пределы других внутренних ограничений. Так, количество уникальных пользователей и групп системы Линукс возросло с 65000 до более чем 4 миллиардов (с 16 до 32 бит). Это позволит Линуксу работать на больших файл-серверах и серверах аутентификации, где ранее, возможно, поджимали предыдущие ограничения. Точно так, максимальное количество PID-ов (ID процесса) было увеличено с 32000 до 1 миллиарда. Это изменение, совместно с другими усовершенствованиями в PID-подсистеме, улучшит производительность запуска приложений на очень занятых или долго работающих системах. Хотя максимальное количество открытых файлов не было увеличено, ядро 2.6. более не требует вашего вмешательства в установление этого предела заблаговременно а самомасштабируе это количество. И, наконец, Линукс 2.6. включает улучшенную поддержку 64 бит на блочных устройствах, поддерживающих работу в данном режиме, даже на 32-битных платформах, таких как i386. Это позволяет создавать файловые системы, размером до 2 Тбайт.

[newpage]

Интерактивностью скорость реакции ядра.

Одним из направлений, в котором сфокусировано развитие ядра 2.6., является создание для пользователей настольных систем, и других задач, требующих высокого уровня контроля над происходящими событиями, более интерактивной системы. Каждое из этих целевых направлений имеет свои проблемы, но было внесено много изменений, которые принесут пользу в обоих случаях.

Главное внутреннее изменение, представленное в ядре 2.6. и значение которого не может быть преуменьшено, это то, что теперь само ядро является выгружаемым. В предыдущих версиях Линукса, когда ОС выполняла что-либо в ядре она не могла быть прервана (на многопроцессорных машинах это было истинно в отношении каждого процессора). Начиная с ядра 2.6., пользовательская задача может прервать работу ядра даже если оно выполняет нечто сложное (чтобы избежать вызванных этим явных коллизий, ядро имеет части, которые не могут быть прерваны в процессе выполнения). Основным выигрышем, достигнутым этими изменениями, заключается в том, что улучшается интерактивная производительность (например при работе пользователя настольной системы) и таким образом система становится более чувствительной к таким вещам как пользовательский ввод.

Другим изменением, позволившим Линуксу стать более интерактивной ОС в отношении приложений, поддерживающих такую интерактивность стала поддержка новых "futexes" (или "Fast User-Space Mutexes" - быстрые семафоры пользовательского пространства). Futexes - это возможность, с помощью которой множественные процессы и нити могут выстраивать последовательность событий так, что они не будут наступать друг другу на пятки ("состояние"). В отличие от традиционных операций взаимного исключения, поддерживаемых большинством библиотек распараллеливания процессов, данный способ частично базируется на поддержке ядра (но только в случае возникновения разногласий) и он также поддерживает систему

приоритетов, позволяющую приложениям или нитям с более высоким приоритетом, получить первоочередной доступ к запрошенным ресурсам. Программное определение приоритета задач, позволило получить более интерактивную систему, что может быть очень важно в критичных ко времени приложениях.

Серьезным изменениям подверглась подсистема ввода-вывода Линукса, сделав его более чувствительным независимо от степени загрузки. Эти изменения включают полностью переписанную подсистему планирования ввода-вывода, кода ядра, определяющего какие процессы будут производить чтение с устройств и когда. Учитывая более раннюю оптимизацию, которая обеспечивала чтение в порядке, наилучшем относительно оборудования, переписанный код обеспечивает исключение долговременного простоя процессов в ожидании ввода-вывода.

И хотя разработчики приложений "реального " (RTOS) получают максимальный выигрыш от этих изменений, Линукс 2.6. все же не является ядром "реального " исполнения. Однако эти и другие изменения делают возможным создание основы для полностью RTOS-версии Линукса, и уже есть сторонние патчи (пока не утвержденные и не включенные в официальный выпуск ядра), которые обеспечивают поддержку пользователей и разработчиков, нуждающихся в этом сейчас.

Подсистема Модулей - Драйверы Устройств

Другой измененной частью, с выходом ядра 2.6., стала подсистема модулей. Большая часть кода была переписана с целью повышения стабильности и создания более прозрачной системы. Кроме очевидных внешних изменений, многое было изменено "", то есть то как ядро видит и использует модули.

Первое и наиболее очевидное изменение (хотя и наименее функциональное) в драйверах ядра 2.6. - это изменение расширений файлов. Вместо бывшего "." (общее расширение для "" файлов, обычно создаваемые в результате компиляции любой программы до линковки в исполняемый файл) теперь используется расширение ".k" ("kernel object"). Это лишь косметическое изменение, делающее более ясным то, что модули не являются промежуточными объектными файлами.

Абсолютно не косметическое изменение - огромная работа, проделанная для устранения состояний гонки, которые имели место в коде многих предыдущих ревизий. Загвоздка в том, что какое-либо устройство может обратиться к модулю в процессе его выгрузки, но уже после того, как модуль проверит не используется ли он. Новый код модульной части ядра сводит такие состояния к минимуму. Следующий шаг данного решения - возможность простого отключения выгрузки модулей в целом.

Улучшенная прозрачность - еще одна возможность новой подсистемы модулей. Почти во всех предыдущих версиях Линукса модули были достаточно умными для определения устройств, поддерживающих определение ID устройства сканированием шин (таких как PCI, ISA PnP, и PC Card). В ядре 2.6. большая часть такой поддержки была стандартизована и вынесена вне ядра, что позволило внешним программам и загрузчикам модулей определять какие устройства поддерживает конкретный модуль. Это позволит различным программам управления оборудованием (таким как "kudzu") принимать интеллектуальные решения об оборудовании,

даже если они его не "знаю". И если вы уверены, вы можете "" драйвер попытаться работать с устройством, которое он не поддерживает (посредством интерфейса новой файловой системы `sys`, описанной ниже).

Другие Усовершенствования

В дополнение ко многим изменениям, описанным выше, было сделано много других основных изменений внутри Линукса, повышающих производительность во многих случаях. В них вошло много устранений "Big Kernel Lock" ("большой блокировки, использовавшиеся в ранних версиях Линукса, поддерживающих несколько процессоров), оптимизация упреждающего чтения файловой системы, отложенная запись, манипуляции с маленькими файлами и другое.

В Линуксе 2.6. решена одна из серьезных проблем стабильности: более невозможно выделить памяти больше, чем есть в системе ОЗУ (плюс своп). Раньше, в некоторых случаях, Линукс мог позволить успешно вызвать malloc() ("выделение памяти") даже если память была исчерпана. Была пересмотрена логика выделения ресурсов и подобные случаи сделаны невозможными (конечно если вы исчерпаете системное ОЗУ - даже без превышения максимума - у вас возникнут серьезные проблемы).

Линукс всегда был выгодной платформой открытых стандартов. Одним из больших внутренних изменений стала полностью переписанная инфраструктура нитей, позволившая использовать поверх себя Native POSIX Thread Library (NPTL). Это дало серьезное увеличение производительности на процессорах Pentium Pro и лучше в сильно "распараллеленных" приложениях, многие крупные игроки уровня предприятия требовали этого (фактически, Red Hat портировала эту возможность в ядро 2.4., входящее в состав Red Hat Linux начиная с версии 9.0). Это изменение привнесло новые концепции в Линукс включая группы нитей, локальная память отдельных нитей, сигналы в стиле POSIX и многое другое. Одним из главных недостатков является то, что приложения (такие как Sun Java) написанные без учета данной спецификации и опирающиеся на старый Linux-isms, не будут поддерживаться. Поскольку преимущества оказываются довольно дороги (и учитывая множество крупных игроков в игре), становится ясным, что большинство важных приложений включают поддержку изменений задолго до выхода нового ядра.

И, наконец, хотя большинство приложений Линукса не получают прямой выгоды, ядро 2.6. теперь включает возможность полностью отказаться от поддержки свопа при компиляции ядра. Это позволит Линуксу работать в немного меньшем объеме памяти, что может быть полезно во встраиваемых устройствах, где маловероятно использование свопа.

Модель Унифицированного Устройства

Очень часто принимаемая за отдельный компонент ОС, принятая в ней модель устройств очень важна для ОС, спроектированной для работы с широким спектром оборудования. Если коротко, то модель устройств представляет собой внутреннюю инфраструктуру ядра, обнаруживающую и определяющую использование ресурсов всех дополнительных компонент системы. Всем ОС (и разновидностям Линукса) свойственно иметь какую-либо концепцию представления устройства. Предыдущие версии Линукса (2.2. и старше) имели низкоуровневое представление об устройствах. Существовали драйверы для различных

Ядро Linux 2.6

<http://www.osrc.info/plugins/content/content.php?content.48>

устройств с шинной архитектурой и различные драйверы устройств знали как опрашивать
поддерживаемую ими шину для определения подключенных к ней устройств. Такая система
постоянно расширялась и различные шинные архитектуры имели различные и не
совместимые API для выполнения разнообразных операций. Линукс 2.4. стал первым шагом к
"" модели устройств объединив PCI, PC Card и ISA Plug-and-Play в единую структуру устройств
с общим интерфейсом. Линукс 2.6. приложил значительные усилия для того, чтобы сделать
завершенное новое видение того, как ядро Линукса видит оборудование с которым работает -
универсальное видение для всей системы.

[newpage]

Абстракция Объектов Ядра

Ядро новой инфраструктуры модели устройств - новый объектно-ориентированный интерфейс, который должны использовать все низкоуровневые устройства. Эта объектная структура устройств ядра (называемая "К-") включает все интерфейсы, необходимые для обеспечения связей и обработки подустройств. Низкоуровневые устройства (такие как системные шины) теперь используют эту общую прослойку для представления системы в унифицированном виде, как в ядре, так и в пользовательском пространстве. Теперь, когда все это централизовано, Линукс может делать множество полезных вещей используя эту информацию.

С этой новой информацией, полностью размещенной в ядре, Линукс способен лучше поддерживать системы, где требуется глубокое знание оборудования. Очевидным примером использования этой информации является управление электропитанием В последние годы стандартом управления питанием является ACPI. ACPI - "Advanced Configuration and Power Interface" (продвинутая конфигурация и интерфейс питания) - поддерживается начиная с Линукса 2.4. В отличие от APM, системы с этим новым интерфейсом должны индивидуально сообщать всем совместимым устройствам о том, что они должны изменить свое состояние питания. Новая система ядра это то, что доктор прописал; она позволяет подсистеме контролировать устройства, которые должны быть отключены или переведены в другое состояние питания. Другой пример использования данной возможности - шины с "горя" подключением. Возможность подключать устройства после загрузки системы сейчас воспринимается как должное, но до Линукса 2.2. включительно ядро не поддерживало "горя" подключение. Начиная с ядра 2.4. эта возможность была отработана и включена поддержка "горячего подключения" PCI, PC Card, USB, и Firewire устройств. Новая ревизия централизованно системы устройств ядра развивает данную поддержку и практически стирает разницу между устройствами с "горя" подключением и без него. Когда вы загружаете компьютер, подсистема обнаружения устройств систематически "подклю" все устройства вашей системы. После того, как были созданы представления в ядре для всех объектов системы после их обнаружения, независимо от того когда это произошло - во время загрузки или позже, становится гораздо проще работать со съемными устройствами.

Системная Файловая Система.

Возможно наиболее очевидным видимым для пользователя в новой модели устройств является создание новой системной файловой системы (дополняющей `proc` для процессов, `devfs` для устройств и `devpts` для псевдотерминалов UNIX98), называемой "sysfs". Эта файловая система (монтируемая в `/sys`) является визуальным представлением дерева

устройств, каким его видит ядро. Ядро получает эту информацию непосредственно от

подсистемы объектов ядра: когда создается объект ядра, также создается файл или каталог (возможен явный отказ объекта ядра от создания записи в sysfs, если в этом есть необходимость).

После того как каждое устройство в системе (точнее К-объект) получает уникальную структуру каталогов, следующим логическим шагом становится экспорт различных известных атрибутов устройства (имя устройства, режим питания, прерывания и подобное) в дерево таким образом, что они могли быть прочитаны и записаны администратором. Это выгодный побочный эффект (вначале слегка запутывающий) позволяющий многие аппаратно-зависимые случаи использования /proc/sys перенести в /sys уже сегодня или же в будущем.

Поддержка Оборудования в Ядре

За последние годы Линукс существенно продвинулся вперед и в направлении мэйнстрима (широкого потребления), каждый этап развития ядра был скачком, лучшим чем предыдущий в контексте поддерживаемых устройств - как вновь появившихся технологий (USB в 2.4.), так и старых технологий (MCA в 2.2.). Таким образом мы пришли к тому, что число устройств, не поддерживаемых ядром 2.6. относительно мало. Осталось очень мало, если вообще осталось, основных направлений PC оборудования, требующего исследований. По этой причине большинство (но, конечно, не все) улучшений в поддержке оборудования PC (включая вышеупомянутую модель устройств) были связаны с упрочнением уже существующей поддержки.

Внутренние Шинные Архитектуры Устройств.

Не менее важными чем процессор, являются системные шинные архитектуры, служащие связующим звеном между устройствами. Мир PC прошел длинный путь системных шинных архитектур, от старейшей ISA (разработанной в оригинальных IBM PC) до современных внешних последовательных и беспроводных шин. Линукс всегда быстро реагировал на появление новых типов шин и устройств, по мере их распространения среди потребителей, и значительно менее быстро адаптируется к технологиям, относительно меньше распространенным.

Примером того, как Линукс включил поддержку нового типа шины позднее, чем коммерческие ОС, стало расширение ISA Plug-and-Play (PnP) в ядре 2.4. (вы могли ранее иметь данную поддержку сделанными наспех пользовательскими утилитами). В Линукс 2.6. включено существенное обновление данной подсистемы, делая ее практически законченной и хорошо интегрированной в остальную часть новой модели устройств. Среди новых возможностей - полная поддержка PnP BIOS, базы данных имен устройств и другие изменения, делающие поддержку более устойчивой. Результатом всех изменений стало то, что Линукс теперь является полностью Plug-and-Play ОС и может быть установлен в машины с совместимым BIOS.

Альтернативами PnP в ISA (довольно не популярными) были MCA (архитектура микро канала) и EISA (расширенная стандартная архитектура промышленного применения). Обе эти подсистемы были улучшены в процессе разработки Линукса 2.6. с целью улучшения

поддержки новой модели устройств. Более того, EISA была далее стандартизирована с другими подсистемами посредством включения базы данных имен устройств.

Другими, не столь кардинальными, стали несколько примечательных изменений Линукса в поддержке шинных архитектур. Шина PCI - наиболее распространена и важна из всех шин - получила существенно улучшенную поддержку в процессе разработки ядра 2.6, включая усовершенствование "горя" подключения и поддержки управления питанием. В этой же строке, новое ядро теперь поддерживает системы с несколькими AGP ("accelerated graphics ports" - ускоренный графический порт - выделенная высокоскоростная шина, также использующая протокол PCI), такие как высокопроизводительные графические станции. В свете поддержки описанных типов внутренней организации PC, Линукс идет в ногу с общими тенденциями на рынке.

Кроме поддержки всех "" шинных архитектур, Линукс 2.6. привнес концепцию "" шины (по крайней мере внутренней). Шина этого типа определена в каждой архитектуре и содержит все мыслимые устройства, которые вы ожидаете увидеть. В PC, например, она может включать наборные последовательные, параллельный и PS/2 порты - устройства, которые имеют место быть, но не привязаны к какой-либо физической шине в системе. На некоторых платформах, данная функциональная поддержка может включать множество сложных вещей (таких как запрос встроенного ПО), но чаще всего это лишь обертка, позволяющая устройству быть обработанным стандартным образом согласно новой парадигме драйвера.

Шины Внешних Устройств

Все вышеперечисленные "" стандарты созревали и добавляли относительно немного новых возможностей в систему, чего нельзя сказать о USB. Универсальная Последовательная Шина претерпела множество изменений в течение недавнего времени. Самое заметное изменение - поддержка новым ядром устройств USB 2.0. USB2 - это новый стандарт, поддерживающий передачу данных со скоростью 480 Мбит/с (по сравнению с 12 Мбит/с у USB 1.1.). Устройства поддерживающие этот стандарт и маркируемые как "high speed" USB ("высокоскоростно USB) постепенно завоевывают рынок. Новый родственный стандарт, называемый USB On-The-Go (или USB OTG) - вариант точка-точка протокола USB, для соединения устройств напрямую друг с другом (например, для подключения цифровой камеры к принтеру без участия PC), в настоящее время не поддерживается ядром 2.6. (патчи с возможностью такой поддержки существуют, но пока не включены в официальный выпуск ядра). В дополнение к поддержке новых устройств, был пересмотрен внутренний пересчет USB устройств таким образом, что стало возможным иметь доступ к намного большему числу однотипных устройств в Линуксе. В основном при этом выиграли большие принт- и файл-серверы (хотя последние, вероятно, будут использовать выделенную шину памяти). Это определенно область технологий, существенно повзрослевших за последние несколько лет, и Линукс работает над тем, чтобы успевать за развитием устройств на рынке.

[newpage]

Беспроводные Устройства

В течение нескольких предыдущих лет технология беспроводного доступа существенно продвинулась вперед. Все чаще кажется что кабели (исключая кабель питания!) канут в лету в ближайшие годы. Ныне беспроводные устройства охватывают как область сетей (наиболее

В области беспроводных сетей устройства можно разделить на дальнотействующиe (например, устройства AX.25 поверх любительского радио) и короткотействующиe (обычно 802.11, но есть и более старые протоколы). Линукс поддерживал устройства обоих типов начиная с ядра 1.2. и обе подсистемы были усовершенствованы в процессе разработки ядра 2.6. Главное изменение заключается в том, что основные компоненты "короткотействующей" подсистемы, поддерживающие различные платы и протоколы, были объединены в единую "беспроводную" подсистему и API. Такое объединение решило проблему мелких несовместимостей различных поддерживаемых устройств, и усилило подсистему поддержки Линуксом единого набора пользовательского инструментария, позволяющего работать со всеми поддерживаемыми устройствами. В дополнение к стандартизации, Линукс 2.6. предлагает множество полноценных усовершенствований, включая улучшенную способность уведомления в случае изменения состояния устройства (например состояние перемещающихся устройств ("roaming")) и изменения в TCP, позволяющие лучше обрабатывать периодические задержки, возникающие в беспроводной сети. Поскольку ядро 2.4. обеспечивало хорошую поддержку беспроводных устройств, многие изменения уже присутствуют в готовом ядре.

В области беспроводных устройств также были сделаны существенные подвижки. IrDA ("протокол названный так группой Infrared Data Associates) был улучшен со времени последнего релиза, в частности были добавлены управление питанием и интеграция с новой моделью драйверов ядра. Значительный прогресс был достигнут Линуксом в обеспечении поддержки Bluetooth устройств. Bluetooth - это новый протокол беспроводного доступа, разработанный для работы на коротких расстояниях и с низким энергопотреблением, но не имеющий в отличие от IrDA такого ограничения как условие прямой видимости. Bluetooth это протокол, спроектированный чтобы "быть", и им оснащаются такие устройства как PDA, cell-телефоны, принтеры и более экзотические устройства типа автомобильного оборудования. Сам протокол работает с двумя различными типами передачи данных: SCO, или Synchronous Connection Oriented (синхронно-ориентированное соединение), предназначенное для аудио приложений не чувствительных к потерям при передаче; и L2CAP, или "Logical Link Control and Adaptation Protocol" (протокол контроля логического соединения и адаптации), для более устойчивого соединения, поддерживающий повторную передачу и подобное. Кроме того протокол L2CAP поддерживает различные под-протоколы (включая RFCOMM для соединения точка-точка и BNEP для Ethernet-подобного соединения в сеть). Поддержка Линуксом таких вещей как Bluetooth продолжает развиваться и мы можем ожидать значительно большей поддержки потребительских устройств. Следует упомянуть, что поддержка Bluetooth также была интегрирована в последние выпуски ядра 2.4.

Поддержка блочных устройств

Шинные архитектуры систем хранения данных (хранилищ)

Выделенные шины хранилищ, такие как IDE/ATA и SCSI, также были усовершенствованы в процессе разработки ядра 2.6. Большинство основных изменений касается подсистемы IDE, которая была переписана (и переписана опять))) в процессе разработки нового ядра, решив много проблем масштабирования и других ограничений. Например, приводы IDE CD-RW теперь могут писать непосредственно реальным драйвером дисков IDE, что более аккуратно

чем ранее (прежде требовался специальный драйвер, эмулирующий SCSI, который был

запутан и труден в понимании). Также, IDE система теперь способна запрашивать у BIOS машины информацию о неподдерживаемом контроллере и использовать разумные значения таймингов и другие данные. В отношении SCSI было сделано множество мелких улучшений относящихся к расширенной поддержке и масштабированию. Одним специфичным усовершенствованием стала поддержка Линуксом многоканальных SCSI-2 устройств, которые имеют более чем 2 LUN на борту (SCSI-2 это предыдущая версия стандарта SCSI устройств, приблизительно 1994 года). Другим важным изменением Линукса стала возможность отката для проверки смены носителя, как делает Microsoft Windows, для большей совместимости с устройствами, не полностью следующим спецификации. Поскольку данные технологии со временем были стабилизированы Линукс включил их поддержку.

Теперь Линукс включает поддержку хоть и не самой шины хранилища, но прямой доступ к EDD BIOS новейших машин (EDD - Enhanced Disk Device - усовершенствованное дисковое устройство), что позволяет ему "" глазами самого storage-сервера на его дисковые накопители. EDD BIOS включает всю известную ему информацию о том, какие шины хранилищ есть в системе (включая IDE и SCSI). Кроме "" определения конфигурации и другой полученной информации об имеющихся накопителях, имеется ряд других преимуществ. Например, новый интерфейс позволяет Линуксу определить с какого именно устройства была загружена система, что может оказаться полезным в новейших ОС, в которых это часто не очевидно. Например, это позволяет продвинутым программам инсталляции использовать данную информацию чтобы определить, где разместить GRUB (загрузчик Линукса).

Дополняя все эти изменения, следует снова подчеркнуть, что все типы шин (накопителей, беспроводных и хранилищ) были интегрированы в новую подсистему Линукса модели устройств. В некоторых случаях эти изменения были чисто косметическими. В других, они были более существенны (в некоторых случаях, например, необходимо было изменять даже логику определения устройства).

Файловые системы (ФС).

Наиболее очевидное использование блочных устройств в Линуксе (да и в других ОС) - смонтировать на нем ФС, и поддерживаемые Линуксом ФС были значительно усовершенствованы в нескольких отношениях начиная с ядра 2.4. Ключевым моментом стала поддержка расширенных атрибутов и контроля доступа в стиле POSIX.

Вторая (и третья) расширенные ФС (имеются в виду ext2fs и ext3fs), чаще всего устанавливаемые Линуксом по умолчанию, были наиболее улучшены в процессе разработки Линукса 2.6. Основным изменением стала поддержка "расширенных", или метаданных, которые могут быть встроены в ФС для каждого файла. Некоторые из этих атрибутов используются системой и могут быть изменены только рутом (все понимают о ком речь?)). Многие другие ОС уже активно используют этот вид атрибутов. К сожалению, наследники Unix обычно не имели достаточной поддержки расширенных атрибутов, и многие пользовательски программы (такие как "tar") должны быть обновлены, прежде чем смогут сохранять и читать эти атрибуты. Это еще одно направление, в котором Линуксу следует развиваться, и полная поддержка расширенных атрибутов близится к завершению.

Первым использованием подсистемы расширенных атрибутов являются списки контроля

<http://www.osrc.info/plugins/content/content.php?content.48>

доступа (СКД) POSIX. Контроль доступа POSIX - это надстройка над стандартными правами доступа Unix, которая позволяет осуществлять более четкий контроль. При необходимости (например при экспорте файлов в NFS), эти СКД могут быть отображены (но с осторожностью) в стандартные права доступа пользователь/группа.

В дополнение к этим изменениям, ext3fs получил другие, более мелкие. Время фиксации в журнале теперь может быть настроено для ФС, чтобы удовлетворять потребности пользователей ноутбуков (когда диску надо набрать обороты, если он был в состоянии сохранения энергии), кроме этого опции монтирования ФС теперь могут быть сохранены непосредственно в ФС (так что вам не надо их указывать во время загрузки), и, наконец, вы можете пометить каталог как "" для увеличения скорости поиска файлов и каталогов.

Линукс также получил ряд усовершенствований на уровне ФС для повышения совместимости с превалирующими PC OC. Во-первых, Линукс 2.6. теперь поддерживает Windows Logical Disk Manager (или "Dynamic Disks"). Это новая схема таблицы разделов, которую WindowsXP и следующие версии приняли для упрощения создания и изменения множества разделов (конечно маловероятно, что Линукс в ближайшее время будет использовать новую схему при установке). В Линуксе 2.6. была изменена (переписана) поддержка NTFS (ФС, используемая теперь и в потребительских продуктах), которая теперь может быть смонтирована в режиме чтение/запись. Поддержка записи все еще находится в экспериментальной стадии и постепенно совершенствуется; она может быть включена или нет в финальный релиз. И, наконец, поддержка FAT12 (используемая в древних системах, а также дискетах) была улучшена в работе над ошибками, присутствовавшими в некоторых mp3-проигрывателях, использующий данный формат. Не отставать от других технологий PC - главное направление, в котором разрабатываются новые версии ядра Линукса.

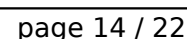
Совместимость с другими ОС также была улучшена. Были сделаны изменения в поддержке расширенных атрибутов ФС HPFS (используемой в OS/2 и других ОС). Атрибуты в стиле HPFS были выделены в отдельное пространство имен. Была улучшена совместимость ФС XFS с ОС IRIX (экзотика).

Кроме вышеописанного, было сделано множество более мелких изменений в поддержке ФС. Была переписана quota, поддерживающая теперь гораздо большее количество пользователей. Отдельные каталоги могут быть помечены как синхронные, таким образом позволяя сделать все изменения (добавление файлов и прочее) "" (положил сюда, появилось там))). Данная возможность наиболее востребована в почтовых системах и базах данных, основанных на каталогах, что немного облегчает восстановление после сбоя диска. В ФС ISO9660 (используемую на CD-ROM) было добавлено "прозрачное " (расширение - только для Линукса). И последнее, исключительно для улучшения поддержки баз данных с разделяемой памятью, была создана ФС, базирующаяся на памяти ("hugetlbfs").

[newpage]

Устройства ввода/вывода.

На переднем фронте любой компьютерной системы находятся устройства ввода и вывода, важнейшие составляющие, однако никогда не воспринимаемые таковыми. К ним относятся такие часто встречающиеся устройства как мышь и клавиатура, звуковые и видео платы, и реже встречающиеся джойстики и другие устройства доступа. Многие подсистемы Линукса, ответственные за взаимодействие с конечным пользователем, были расширены в процессе



разработки ядра 2.6., но большинство распространенных устройств уже имели достаточно зрелую поддержку. Улучшенная поддержка этих устройств Линуксом 2.6., в значительной степени явилась следствием усовершенствованной поддержки ядром внешних шин данных (например, беспроводные Bluetooth клавиатуры и подобное). Однако есть несколько подобных областей, в которых Линукс преуспел более значительно.

Интерфейсные устройства для пользователей

Главным внутренним изменением в Линукс 2.6. стал сильно переработанный уровень интерфейса пользователя (ИП). Уровень ИП является центром в исследовании пользователем системы Линукс, в него входят видеовыход, мышь и клавиатура. В новой версии ядра этот уровень был переработан и "" (от "", не от "модуляция") намного больше чем прежде. Теперь стало возможным создать "безголовую" систему Линукс, исключив поддержку дисплея и всего прочего. Основное преимущество подобной модульности получают разработчики встроенных систем, создающих устройства, которые могут быть администрированы посредством сети или консоли, а для конечных пользователей преимуществом стал перенос в модули предположений по умолчанию об устройствах и архитектурах. Например прежде всегда предполагалось, что если вы работаете с РС, то вам необходима поддержка стандартного контроллера клавиатуры АТ (i8042); новая версия Линукса устранила это требование, и ненужный код может быть удален из системы.

Были сделаны некоторые изменения в поддержке Линуксом вывода на монитор, хотя большинство из них применимы только в конфигурациях, которые используют подсистему внутреннего фрейм-буфера ядра (большинство систем Линукс на Intel-е сконфигурирован иначе, однако это не относится ко многим другим архитектурам). ИМХО, лучшей "", появившейся вследствие этого, стала поддержка во время загрузки разрешений вплоть до 24bpp (а глубину цвета автор вероятно измеряет в количестве точек, 800x600?))) (симпатичный пингвин, если обращали внимание). С другой стороны, к другим изменениям в работе с консолью можно отнести изменение размеров и вращение (для PDA и подобных устройств), расширенная поддержка аппаратного ускорения многих устройств. И, наконец, в ядро включена поддержка запросов информации о совместимости к VESA мониторам, хотя Xfree86 и многие программы инсталляции различных дистрибутивов имели такую возможность в пользовательско окружении ядра.

Кроме значительных изменений, Линукс 2.6. включил множество более мелких изменений взаимодействия с пользователем. Например, теперь поддерживаются сенсорные экраны. Драйверы мыши и клавиатуры были улучшены и стандартизованы таким образом, чтобы независимо от устройства или протокола, работа велась посредством единой записи об устройстве (/dev/input/mouse0, например). Теперь поддерживаются более экзотические мыши (с несколькими колесиками прокрутки, например). Были изменены переназначения расширенных клавиш PC клавиатуры, чтобы соответствовать стандарту расширенных клавиш Windows)))))). Поддержка джойстиков улучшена не только за счет новых драйверов (например X-Box ""), но также за счет добавления новых возможностей, таких как отдача (force-feedback). И последнее, но не менее важное, новое ядро поддерживает терминальное устройство Брайля (Tietan Voyager), позволяющее слепым работать с Линуксом (это важное дополнение, поэтому оно уже портировано в ядро 2.4.).

Отметим, что Линукс изменил интерфейс "системного ", дабы обеспечить лучшую поддержку

<http://www.osrc.info/plugins/content/content.php?content.48>
систем без локальной клавиатуры. Интерфейс системного запроса ("sysrq") это способ, который позволяет системному администратору посредством локальной консоли получить отладочную информацию, перезагрузить систему, перемонтировать ФС в режиме для чтения и много других чудесных штук. Начиная с ядра 2.6., Линукс поддерживает полностью безголовые системы (дивысь выщэ), также он поддерживает инициацию перечисленных событий посредством ФС /proc (ясно, что если ваша система подвисла и вы вынуждены делать подобные вещи, то вряд ли это вам поможет).

Аудио

Одним из самых ожидаемых пользователями изменений было включение в ядро ALSA ("Advanced Linux Sound Architecture") вместо устаревшей аудио подсистемы. Старая подсистема, известная как OSS ("Open Sound System"), поддерживалась Линуксом с древних времен и имела множество архитектурных ограничений. Первым основным улучшением в новой системе стало ее проектирование от начала до конца безопасной по отношению к системным нитям и SMP, устранены проблемы со многими старыми драйверами, которые "глю" если не работали с ожидаемой парадигмой "ПК-означает-один-". Более важно то, что драйверы изначально спроектированы модульными (пользователи ранних версий Линукса помнят как модульность была воскрешена в ветке ядра 2.2.), и поэтому поддерживает системы с несколькими аудио картами, более того, с несколькими разнотипными аудио картами. Независимо от того насколько хороша система внутренне, она не была бы улучшенной для пользователей, если бы они не критиковали ее, и аудио подсистема критиковалась довольно сильно. Ключевыми были поддержка нового оборудования (включая USB аудио и MIDI устройства), полнодуплексное воспроизведение и запись, оборудование и не чередующееся микширование, поддержка "" звуковых устройств и многое другое. Кем бы вы ни были, аудиофилом или кем-то, кто просто слушает mp3, улучшенная поддержка звука в Линуксе сделала существенный шаг навстречу к вам.

Мультимедиа

Сегодня кроме поддержки звука, пользователям нужна поддержка более интересных устройств, таких как веб-камеры, радио- и теле адаптеры, цифровые камеры. Во всех трех случаях была улучшена поддержка этих устройств в ядре 2.6.

В то время как поддержка радио карт (в большей или меньшей степени) уже была (часто в пользовательско окружении), поддержка теле тюнеров и видеокамер была добавлена лишь в последние одну-две ревизии. Эта подсистема, известная как Video4Linux (V4L), была усовершенствована в процессе работы над новой редакцией ядра, в частности был "" API и расширена поддерживаемая функциональность карт. Новый API не совместим со старым, поэтому приложения написанные под старый API, должны быть переписаны.

Новинкой стала встроенная поддержка в ядре 2.6. оборудования Цифрового Широковещательного Видео (Digital Video Broadcasting (DVB)). Это оборудование, обычное для сильнонаполнены приставок, может быть использовано для того, чтобы сделать Линукс-сервер в виде Tivo-устройства, с соответствующим ПО.

Усовершенствования ПО

Усовершенствования Линукса не сфокусированы исключительно на инфраструктуре и оборудовании. Без поддержки ПО (такого как ФС и сетевые протоколы), поддержка оборудования была бы просто мертвым грузом.

[newpage]

Сети

Передовая сетевая инфраструктура всегда была главным козырем Линукса. Линукс - это ОС, уже поддерживающая большинство доминирующих в мире сетевых протоколов включая TCP/IP (v4 и v6), AppleTalk, IPX и другие (фактически из распространенных и не поддерживаемых на ум приходит лишь старый и сильно запутанный протокол NetBIOS/NetBEUI от Майкрософта).

Как и изменения во многих других подсистемах, большинство изменений в ядре 2.6. касающихся сетевого оборудования скрыты от глаз и не очевидны. Сюда относятся изменения, позволяющие воспользоваться преимуществами модели устройств и изменения во многих драйверах устройств. Например, теперь Линукс имеет выделенную подсистему MII (Media Independent Interface (интерфейс, не зависящий от среды передачи) или IEEE 802.3u), которая используется драйверами некоторых устройств. Эта подсистема исключает множество незначительно различных экземпляров MII, которые поддерживались различными драйверами и различными способами, тем самым исключив дублирование кода и ненужную работу. К другим изменениям в основном относится усовершенствование ISDN и прочее.

Одним из главных нововведений стала поддержка Линуксом протокола IPSec. IPSec, или IP Security (безопасность IP), это пул протоколов для IPv4 ("IP) и IPv6, обеспечивающих криптографическую безопасность на сетевом уровне. В связи с этим приложения могут не беспокоиться об обеспечении безопасности. Он очень схож с SSL и другими туннелирующими протоколами обеспечения безопасности, но работает на более нижнем уровне. На сегодняшний день поддерживается шифрование в ядре включая разнообразные реализации SHA ("secure hash algorithm" - хэш-алгоритм безопасности), DES ("data encryption standard" - стандарт шифрования информации) и другие.

Касательно сетевых протоколов, была улучшена поддержка работы с сетевыми группами (multicast networking). Сетевые группы это сети, в которых один пакет может быть получен несколькими компьютерами (в традиционном сетевом взаимодействии точка-точка вы можете одновременно связаться лишь с одним компьютером). В основном эта возможность используется пейджинговыми системами (такими как Tibco) и ПО организации аудио/видео конференций. Теперь поддержка этих возможностей Линуксом 2.6. обеспечивается реализациями таких SSM (Source Specific Multicast - группа обуславливаемая источником) протоколов как MLDv2 (Multicast Listener Discovery - групповое открытие слушателя) и IGMPv3 (Internet Group Messaging Protocol - Internet-протокол групповой рассылки). Это стандартные протоколы, поддерживаемые оборудованием таких гигантов как Cisco).

В ядре 2.6. протокол LLC выделен в отдельный стек LLC. LLC, или Logical Link Control (управление логическими связями) (IEEE 802.2), это низкоуровневый протокол, используемый протоколами более высокого уровня, такими как Microsoft NetBeui, IPX и AppleTalk. Частью данного изменения стало то, что драйверы протоколов IPX, AppleTalk, и Token Ring были переписаны с учетом новой общей подсистемы. Кроме того, внешним образом обеспечена ее

связь с работой стека NetBEUI, остается отметить, что вряд ли эта связь будет интегрирована в ядро.

Среди мелких изменений - IPv6 был подправлен и теперь способен работать в сетях Token Ring. Реализация NAT/masquerading была расширена для лучшей поддержки протоколов, требующих нескольких соединений (H.323, PPTP и прочее). На фронте маршрутизаторов поддержка Линуксом конфигурирования VLAN-ов теперь не является "экспериментальн".

Сетевые файловые системы.

Следствием качественной поддержки Линуксом сетевых протоколов является такая же качественная поддержка сетевых файловых систем. Монтирование (и иногда экспорт) сетевой ФС является одной из не многих высокоуровневых операций поддерживаемых ядром непосредственно (наиболее очевидно что "сетевые блочные" не претерпели существенных изменений в ядре 2.6. и они в основном используются в специализированных приложениях, в которых вы так или иначе работаете с ними как с подобием ФС). Все остальные сетевые операции вынесены в пользовательское пространство и не относятся к области разработчиков ядра.

В мире Линукса и ЮНИКС клонов наиболее распространена сетевая ФС метко названная Сетевая ФС (Network File System - NFS). NFS это сложный протокол расшаривания файлов (общего использования), имеющий глубокие корни в ЮНИКС (и особенно в прекрасной его реализации в Sun Solaris). Основным транспортным протоколом могут быть TCP и UDP, кроме этого необходимы несколько под-протоколов, каждый из которых выполняется поверх протокола RPC ("remote procedure call - удаленный вызов"). К ним относится выделенный протокол "монтирования" для аутентификации и блокировки файлов NLM-ом ("network lock manager - менеджер сетевых"). (Вообще реализация тесно связана с другими протоколами, основанными на RPC, включая NIS (network information service - сервис сетевой информации) для аутентификации. NIS обычно не используется в Линуксе из-за его "ненадежности"). Ситуация осложнена тем, что NFS не был всесторонне адаптирован к работе в качестве протокола Internet.

В Линуксе 2.6. эта ФС получила множество изменений и дополнений. Среди них экспериментальная поддержка нового и пока не распространенного протокола NFSv4 в реализации как клиентской, так и серверной части (предыдущие версии Линукса включали поддержку версий v2 и v3 этого протокола). Новая версия поддерживает более сильную и безопасную аутентификацию (с шифрованием), более интеллектуальную блокировку, псевдо-ФС и прочее. Не все возможности NFSv4 реализованы в Линуксе на сегодня, но имеющаяся поддержка довольно стабильна и может быть использована в коммерческих приложениях. Кроме этого реализация серверной части NFS в Линуксе усовершенствована в отношении масштабирования (вплоть до 64-кратного увеличения одновременно обслуживаемых пользователей и очередей запросов), завершенности (посредством поддержки обслуживания с помощью TCP, в дополнение к UDP), устойчивости (отдельные драйверы ФС могут адаптировать путь к файлу в системах, которые их экспортируют), сопровождаемость (вместо системных вызовов, управление ФС новым `nfsd`). Среди менее очевидных изменений - разделение lockd и nfsd, поддержка нуль-копии (zero-copy) сети на поддерживаемых интерфейсах. Реализация клиентской части NFS была улучшена за счет протокола RPC, включающего инфраструктуру кэширования, управление соединением

посредством UDP, и другие улучшения TCP. Была усовершенствована поддержка томов root-ФС (для бездисковых систем), теперь ядро поддерживает NFS поверх TCP.

Кроме улучшенной поддержки сетевых ФС в стиле ЮНИКС, Линукс 2.6. принес множество усовершенствований сетевых ФС Windows-стиля. В Windows серверах (также как и в OS/2 и других ОС) стандартным протоколом "расшаривания" ФС является SMB ("server message block" - блочные сообщения), и все последние версии Линукса прекрасно поддерживают клиентскую часть этого протокола. Однако Windows 2000 опирается на новую надстройку этого протокола называемую CIFS ("common internet filesystem" - разделяемая internet-ФС) . Цель такого существенного обновления состояла в упрощении и совершенствовании некоторых аспектов протокола SMB, что привело к окончательной путанице (сам протокол не был четко определен и часто расширялся, что стало причиной несовместимости даже между его реализациями в Windows 95/98/ME и WindowsNT/2000). CIFS призван добавить поддержку Unicode, улучшенную блокировку файлов, жесткие ссылки, устранить остатки зависимостей от NetBIOS и нескольких других возможностей для пользователей Windows. Так как пользователи Линукса не любят сидеть в темноте подолгу, в ядро 2.6. включена полностью переписанная поддержка монтирования ФС CIFS. Линукс 2.6. также поддерживает расширение протоколов SMB и CIFS именуемое SMB-UNIX, которое позволяет ему посредством SMB-серверов, поддерживающих эти расширения (например Samba), получать доступ к файлам не-Windows типа (такими как узлы и символические ссылки).

Хотя сегодня на это не обращают внимание, все же Линукс не забыл о пользователях Novell NetWare. Ядро 2.6. позволяет клиентам Линукса монтировать до 256 "" с одного тома NetWare, используя при этом встроенный драйвер ФС NCP ("NetWare Core Protocol - протокол ядра NetWare").

Линукс 2.6. включает улучшенную поддержку относительно новых доменов сетевых ФС, систем, в которых файлы хранящиеся на одном логическом томе, могут быть распределены между несколькими узлами. Кроме представленной в Линукс 2.4. ФС CODA, Линукс включил некоторую поддержку еще двух распределенных ФС: AFS и InterMezzo. AFS, Andrew ФС (названная так потому как изначально была разработана в CMU), в настоящем ограничен лишь операциями чтения. Вторая ФС, InterMezzo (тоже разработанная в CMU), также поддерживается начиная с ядра 2.6., но позволяет делать больше, например операция разрыва соединения (таким образом вы работаете с кэшированными файлами локально), поэтому эту ФС можно использовать в приложениях высокой доступности, когда вам необходим постоянный гарантированный доступ к хранилищу (или имитация при отключении). InterMezzo также пригоден для использования в приложениях, нуждающихся в синхронизации файлов между несколькими компьютерами, такими как ноутбуки, наладонники и ПК. Многие проекты обеспечивают поддержку этих типов ФС, позволяя Линуксу развивать их и дальше.

[newpage]

Другие изменения

Защита

Другими серьезными изменениями в ядре 2.6., не получивших достаточного внимания, стали изменения, связанные с безопасностью. Наиболее существенное заключается в том, что система безопасности, полностью обеспечиваемая ядром (права супер-пользователя в Unix

безопасности (сегодня, однако, существует лишь одна модель системы защиты по умолчанию и пример, как создать собственную). Вследствие этого изменения все части ядра были изменены, и теперь используют новую систему как базовую, в отличие от старой системы "суперпользователя". Все Линукс системы и в дальнейшем будут иметь учетную запись суперпользователя, но при этом будет учитываться возможность ее отсутствия. Другим изменением в безопасности стало то, что исполняемые модули (например драйверы, поставляемые производителем оборудования) теперь не могут "" системные вызовы по собственному усмотрения, а также не смогут модифицировать таблицу системных вызовов. Эти меры существенно ограничивают возможности модулей с не-открытыми исходниками в ядре и, возможно, прикрывают некоторые юридические лазейки обхода GPL. Последнее изменение, относящееся к безопасности, - теперь Линукс способен работать с аппаратными генераторами случайных чисел (присутствующих в некоторых современных процессорах), лучшего чем основанный на энтропии пул (довольно хороший по общему признанию), базирующийся на случайных аппаратных флуктуациях.

Одно из интереснейших нововведений в Линуксе 2.6. - поддержка архитектуры "user mode" (непривилегированного режима). По существу это портирование Линукса (как на различные аппаратные архитектуры), реализующее полностью виртуальную среду выполнения Линукса-в-Линуксе. Новая инстанция Линукса выполняется также как обычное приложение. "" приложения вы можете конфигурировать"" сетевые интерфейсы, ФС и другие устройства посредством специальных драйверов, которые безопасным образом связаны с головной копией Линукса. Такой подход полезен как в задачах разработки (профилирование и прочее), так и для анализа системы безопасности и забавы ради))). И хотя большинство пользователей не нуждаются в такой поддержке, все же эта "прикольная " достойна быть запущенной на вашем "" (Поразите своих друзей!))).

Кроме описанных ранее общих усовершенствований (APM и ACPI, улучшенная поддержка беспроводного оборудования и т.п.), Линукс также включил две трудно классифицируемые возможности для пользователей ноутбуков. Первая из них заключается в полной поддержке новой редакцией ядра режима приостановки-ПО-на-диск (software-suspend-to-disk). Вторая - это поддержка возможности современных мобильных процессоров изменять частоту (и энергопотребление) основываясь на том, питается ноут от сети или от батарей.

Хотя Линукс 2.6. является серьезным апгрейдом, пользовательски приложения почти не почувствуют разницы в работе. Единственное исключение касается вопроса распараллеливания (threading): некоторые вещи, которые делали приложения под 2.4. или 2.2, теперь не позволительны. Однако подобные приложения являются скорее исключением из правила. Естественно низкоуровневые приложения, такие как утилиты модулей, определенно не будут работать. Кроме того, некоторые файлы и форматы в каталогах /proc и /dev были изменены, и приложения имеющие зависимости от них, также могут работать не корректно (особенно это касается тех частей, которые были перемещены в виртуальную ФС /sys. В

случае с /dev, имена устройств могут быть легко сконфигурированы для обратной совместимости).

В дополнение к изменениям оговоренным в стандартах, есть несколько мелких изменений, которые затрагивают окружение ядра. Во-первых, древние "" (файл или раздел подкачки в Линукс 2.0 и более ранних версиях) должны быть переформатированы для использования с ядром 2.6. (поскольку своп не содержит постоянной информации, это не станет проблемой для обычных пользователей). Демон kHTTPd, позволяющий обрабатывать веб-содержимое непосредственно ядром, был перемещен, также как и многие другие, влияющие на производительность сервисы, включая Apache, Zeus, решив таким образом проблему достижения максимальной производительности ядра. Были исключены: поддержка древних ДОСовских диск-менеджеров (OnTrack и EzDrive) для старых жестких дисков; а также поддержка совместимости с ДОС. Из ядра была удалена поддержка специального загрузочного сектора, предназначенного для загрузки с дискет, вместо этого следует использовать SysLinux. И последнее, поддержка "soundmodem" (для игр со звуком и модемом) была вынесена из ядра в пользовательское пространство, где имеется лучшая поддержка и более своевременное обновление этого ПО.

Материал лежащий в основании

Этот документ составлен прежде всего на основании долгих часов изучения логов изменений (changelogs), изучения и экспериментирования с исходниками, чтения рассылок и многих и долгих поисков на Google и Lycos документации по данной теме. В некоторых местах могут быть неточности и ошибки, в некоторых - изменения, сделанные уже после описания (автор с особенной осторожностью говорил о двух версиях реализации поддержки IDE над которыми велась работа при написании данного слога, но есть и другие примеры). Также автор старался осторожно использовать информацию с веб-страниц проектов, интегрированных в ядро, но существенно преуспевших по сравнению с тем, что было включено в ядро официально. Если у вас возникли вопросы, замечания или что еще угодно, можете связаться со мной по адресу jpranevich@kniggit.net.

Для технически продвинутых людей, которым нужен лишь список изменений и возможностей, Дэйв Джонс (Dave Jones) сделал превосходное резюме процесса разработки ядра 2.6. доступное по адресу <http://www.codemonkey.org.uk/post-halloween-2.5.txt>. Все совпадения в этих двух документах вполне естественны, так как мы оба работали с исходниками, причем версии для i386.

Правом на этот документ обладает Джозеф Праневич (Joseph Pranevich). Разрешается распространение "" этого документа без изменений, но, пожалуйста, сообщите по электронной почте автору о ваших намерениях иным способом распространения (журналы и т.п.). В любом случае автор будет благодарен за письмо, поскольку может быть доступна более свежая версия этого документа.

Перевод: [Death_Mokar](#).

