

Linux-машина следующего поколения

Silicon Graphics готовится к демонстрации новой высокопроизводительной системы Altix 4000, вобравшей в себя мощь 512 процессоров Itanium и операционной системы с ядром Linux (Novell SUSE Linux Enterprise Server 9 или Red Hat Enterprise Linux Advanced

Server 4), как и нынешняя Altix 3000. В отличие от предшественницы новая машина содержит 128 Тбайт памяти, превосходя ее по этому параметру на 24 Тбайт. Предыдущая разработка имела блоковую серверную структуру по восемь процессоров со связанной памятью, теперь Altix исполнена в традициях blade-серверов, но при этом



она сохранила свою приверженность мощью. Видимо, Silicon Graphics окончательно отказалась от использования собственных MIPS-процессоров и ОС Irix. Сегодня большую часть (75%) списка пятисот самых мощных в мире компью-

теров составляют машины под управлением Linux. Silicon Graphics удалось занять четвертую позицию с системой SGI Altix 3700, используемой NASA. Компания IBM пока не сдает лидирующих позиций:

на самой вершине расположился eServer Blue Gene Solution ее производства.

www.news.com

www.top500.org

Linux в Китае: прогнозируемое и неожиданное

Будущее программной индустрии Китая — Linux. Выбор уже сделан, но именно он определил и новую проблему для Поднебесной. SUSE от Novell, завоевавший лидирующие позиции на внутреннем рынке, а также успешное продвижение Red Hat, с одной стороны, сталкивает гигантов с интересами китайских разработчиков, а с другой — является мощным стимулом для развития последних. Когда в конце прошлого года первые шаги на новом рынке стала осваивать Red Hat, специалисты подчеркивали, что, прежде чем добиться успеха, ей необходимо будет проделать долгий путь к пониманию Китайских правил, в которых индивидуальные превосходства смешаны с политической властью и личными связями.

Собственно китайские Linux-компании имеют солидную государственную поддержку: символ Linux-индустрии Китая Red Flag Software — со стороны Академии наук Китая, а Turbolinux — со стороны правительства Шанхая. Орбита интересов этих компаний недостижима для посторонних, тем не менее Китай-

ские правила не являются неким амулетом для отечественных производителей — хорошим примером может служить и успех Novell, и даже продвижение Microsoft в отдельно взятой провинции Синьхуа.

Однако не все так гладко. Главной проблемой, сдерживающей развитие Linux в Китае, как всегда оказывается законодательство, регулирующее взаимоотношения иностранных представителей со своими собственными, что теперь вызывает серьезные опасения за будущее Linux у последних более, чем у первых. Сможет ли индустрия, уже пережившая стремительный взлет, избежать губительного периода застоя, зависит от того, взвесит ли государство все преимущества нынешней политики и дивиденды с отказа от излишней опеки. История пока зательна еще и тем, что ни какое-либо сообщество, ни коммерческие гиганты не имеют в своем распоряжении совершенных механизмов, гарантирующих беспрепятственное распространение идеи «невзирая на границы».

www.business.newsforge.com

Между двумя средами

В отличие от ситуации в Китае сообщество в Европе может оказывать влияние даже на принятие решений гигантами в мире Linux. После сообщения о сворачивании поддержки KDE в SUSE Linux Enterprise Server и Novell Linux Desktop в пользу GNOME (предполагалось, что отныне поддержкой KDE будет вестись исключительно проект openSUSE) сервисная служба Novell испытала беспрецедентное давление со стороны пользователей и разработчиков. Под напором писем компания была вынуждена отказаться от спешного решения и опять повернуться на 180°, чтобы успокоить клиентов. Некоторые аналитики считают, что



первое решение Novell можно хоть как-то объяснить владением Ximian с 2003 года и давним замыслом на более плотную интеграцию GNOME со своими коммерческими дистрибутивами. Как разъясняют эти маневры в Novell, компания продолжает спонсировать оба проекта и будет включать в дистрибутивы все новейшие разработки в области Open Desktop, но все же отдает формальное предпочтение GNOME. Именно эта интегрированная среда устанавливается в готовящихся дистрибутивах по умолчанию, если пользователь не указывает в процессе инсталляции другой вариант.

www.novell.com/prblogs

Nessus отказывается от GPL



Еще не вышедший третий большой релиз одного из самых популярных открытых решений для автоматизации проверки и обнаружения уязвимостей и брешей в защите системы Nessus и без того привлек бы к себе внимание, но после заявления группы разработчиков Tenable Network Security о переходе на закрытый код резонанс последовал нешуточный. Сооснователь Tenable Network Security Рон Гула дал обстоятельное интервью Security Focus, где, объясняя причины отказа от открытой модели разработки, сослался на требования клиентов и коммерческие интересы проекта, подчеркивая, что «свобода» значит очень много для разных стран, при разных дворах и в бизнесе, и Nessus 3 останется свободным — кроме той части, где до сих пор кому угодно можно было извлекать выгоду

из труда Tenable Network Security кроме самой Tenable. Коммерческая поддержка Nessus 3 обойдется в \$1200 в год за один обслуживаемый сканер. Приоткрыта завеса тайны и над самим Nessus 3. Ожидаемая скорость работы сканера при анализе серверов Windows поднимется в пять раз и, по самым оптимистичным прогнозам, составит 17 крат, что будет достигнуто благодаря новому, полностью переписанному «сердцу» сканера. Между тем Nessus 2 по-прежнему остается свободным. В версии 2.2.6 исправлены обнаруженные на сегодняшний день ошибки, проведены дополнительные манипуляции с NASL и включен улучшенный сканер Nessus TCP — пользователям Nessus 2.2 настоятельно рекомендуется произвести обновления.

Вокруг Nessus складываются не только негативные, но и вполне благоприятные отношения специалистов. С небольшой поддержкой от Tenable Network Security группа разработчиков Intevation GmbH представила первого кандидата в релиз 1.0.0 — NessusClient с графическим интерфейсом под X11/GTK, возможностями одновременного запуска нескольких задач, поддержкой сохранения отчетов в PDF, HTML, XML и другими полезными функциями.

www.nessus.org

www.securityfocus.com

Open Source как социализм

Шей Агасси, исполнительный директор SAP, подверг критике движение Open Source, заявив, что оно не способствует росту новых идей, необходимых для развития общества: «Мы все говорим о том, насколько хороша Linux, но если вы взглянете на самый инновационный десктоп, то увидите, что Vista от Microsoft копирует не Linux, а Apple. Социализм в интеллектуальной собственности (IP) — худшее из всего, что может случиться в любом информационном обществе... Если у нас нет путей для защиты IP, то нет и смысла инвестировать в нее». Таким образом, Агасси последовал примеру

Билла Гейтса, в начале года назвавшего разработчиков свободного ПО коммунистами. Тенденция налицо — всего за год сообщество регрессировало до социализма. Что скажут в следующем году?

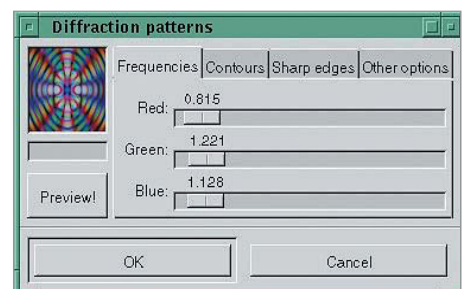
Слова Агасси можно было бы пропустить мимо ушей, как и речи других противников, если не принимать во внимание влияние, которое имеет SAP на новую организацию ESA (European Software Association). Именно она пытается организовать противодействие переходу европейских организаций и публичных администраций на открытые решения.

www.news.zdnet.co.uk

Короткой строкой

Fedora Project, спонсируемый Red Hat, открывает тестирование пятого по счету дистрибутива. Тестовый релиз включает модульный XOrg, ядро 2.6.15-rc1-git3, компилятор GCC 4.0.2, последние интегрированные среды GNOME 2.12 и KDE 3.4.92, слепок Xen 3.0 для i386, улучшенную открытую версию Java и значительные изменения в установщике. Окончательный релиз Fedora Core 5 должен состояться 27 февраля 2006 года.

www.redhat.com



The GNU Image Manipulation Program, более известный как GIMP, отметил свое десятилетие. Первый исторический анонс Питера Маттиса был сделан 21 ноября 1995 года следующими словами: «GIMP разработан с целью предоставления интуитивного графического интерфейса к разнообразным операциям редактирования изображений». Первая версия, увидевшая свет, имела номер 0.54.

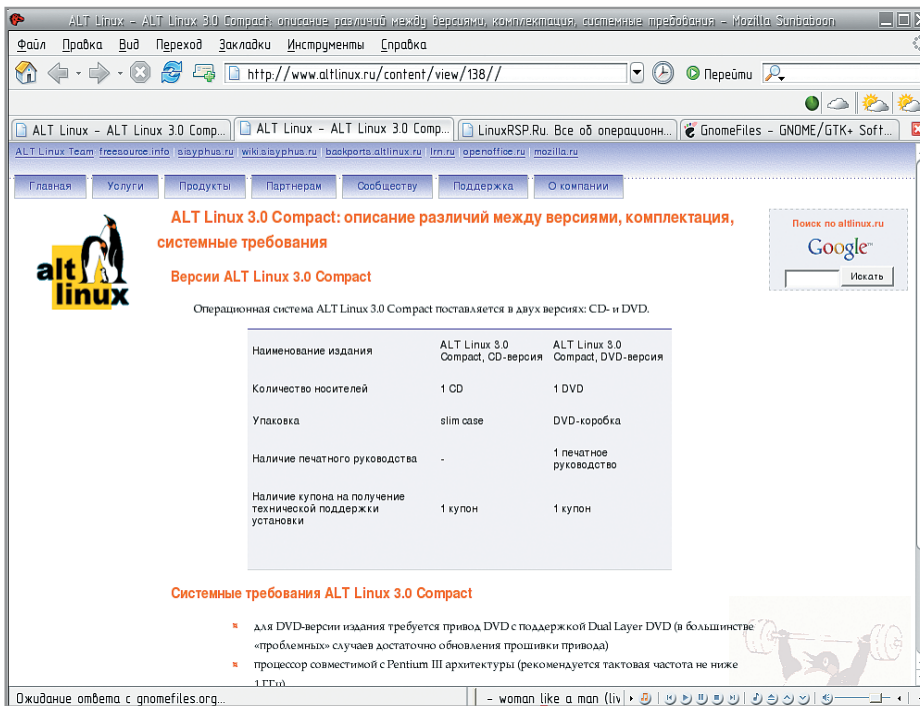
www.gimp.org

Проект Mozilla.ru стал лауреатом Премии Рунета 2005, учрежденной Федеральным агентством по печати и массовым коммуникациям, заняв первое место в народном голосовании, которое продолжалось в течение 100 дней. Mozilla.ru выражает глубокую признательность всем, кто проявил внимание к этому проекту: «Это наша общая победа, она стала возможна благодаря нашей сплоченности, преданности общему делу». В «народную десятку» попал и русскоязычный ресурс энциклопедии ru.wikipedia.org.

www.premiaruneta.ru



Новое звучание «Альта»



Самый ожидаемый отечественный дистрибутив ALT Linux 3.0 Compact готов к встрече со своими пользователями. «ALT Linux 3.0 Compact — простая и удобная операционная система для повседневной работы, основанная на последних разработках компании ALT Linux. В дистрибутив включено все необходимое пользователю домашнего

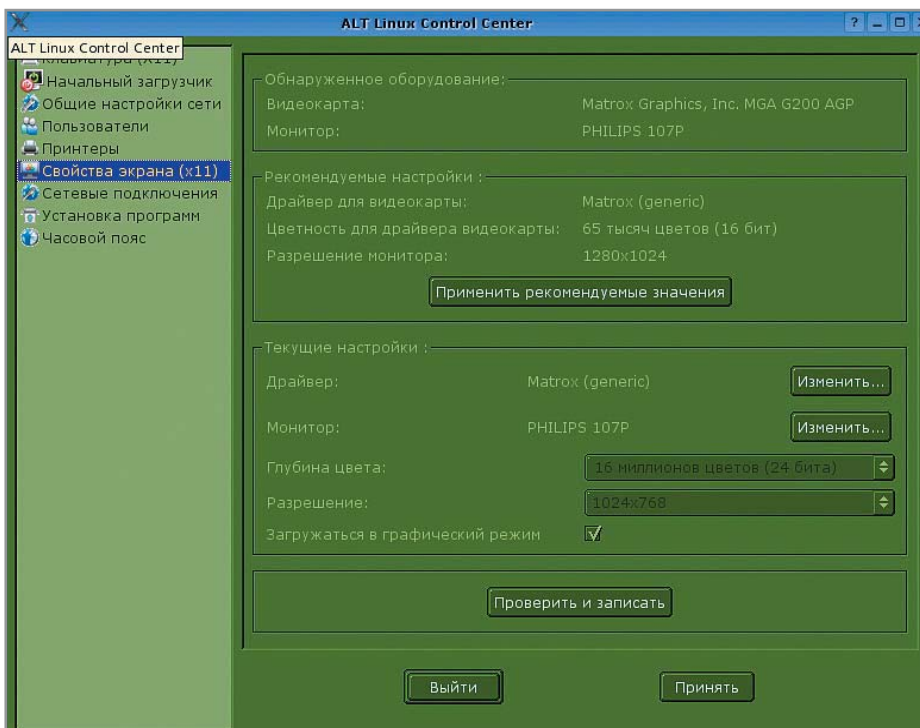
компьютера, в том числе программы для работы в Интернете, офисные приложения, утилиты для просмотра и прослушивания мультимедийной информации. В Compact реализована поддержка широкого спектра современного оборудования, в том числе процессоров, материнских плат, видеоускорителей, принтеров, цифровых камер, сканеров и т. д.» Произ-

водитель отмечает, что в очередном дистрибутиве пользователи смогут опробовать новую программу установки, познакомиться с системой настройки ALT Linux Control Center, основанной на модульной безопасной платформе Alterator, уже знакомой тем, кто использовал ALT Linux Sisyphus.

Дистрибутив содержит новую стабильную версию ядра Linux 2.6.12, графическую систему XOrg 6.8.2, интегрированную среду KDE и массу приложений, необходимых домашнему компьютеру.

ALT Linux 3.0 Compact доступен на одном CD или DVD в коробке с печатным руководством — в обоих случаях с купоном, дающим право на получение технической поддержки. Рекомендуемые системные требования укладываются в процессор совместимой с Pentium III архитектуры с тактовой частотой не ниже 1 ГГц, объем оперативной памяти — от 256 Мбайт. ALT Linux уже демонстрировала свою продукцию на выставках «Цифровой мир-2005» и ИТО-2005, где также была предоставлена возможность ознакомиться с самозагружающимся дистрибутивом ALT Linux 3.0 TravelCD, не требующим установки на жесткий диск.

www.altlinux.ru



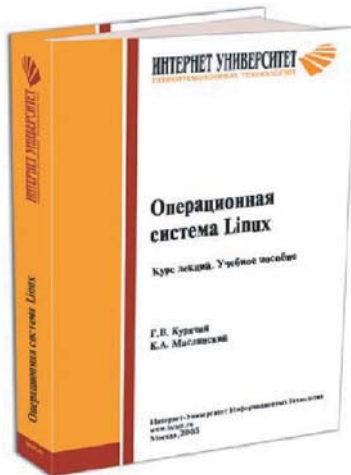
Предпраздничная артподготовка

Перед новогодними праздниками спамеры начинают готовить плацдарм для своих рассылок, создавая новые зомби-сети. В частности, в ноябре Trend Micro обнаружила три модификации Linux-червя под названием Lurper. Он случайным образом генерирует IP-адрес и пытается атаковать его как уязвимый веб-сервер. Первая модификация содержала троянский компонент, с помощью которого злоумышленник мог управлять компьютером через открытые порты 7111 и 7222. Два других варианта уже использовали стандартный для зомби механизм реверсивных команд, когда захваченный компьютер сам обращается по указанному захватчиком адресу.

www.osp.ru

Учиться никогда не поздно

Лучшим подтверждением этому изречению является существование Интернет-университета информационных технологий — проекта дистанционного образования, первого в России учебного заведения, в котором такой вид обучения доступен любому, так как получать его можно бесплатно. Суетливый и нервный ритм подготовки к зачетам и экзаменам? Забудьте. Теперь вы сами решаете, когда готовы отчитаться о приобретенных знаниях, к тому же вы вольны выбирать несколько курсов сразу — было бы по силам. «Это только начало. Возможно, название нашего проекта покажется вам нескромным, но мы на самом деле ставим своей целью создание университета информационных технологий», — обращается к посетителям ресурса один из авторов проекта Анатолий Шкред. Интернет-университет предлагает слушателям открытые курсы по архитектуре ЭВМ, информационной безопасности, интернет- и сетевым технологиям, операционным системам (Solaris, Linux, Unix), базам данных, разработке приложений (объектно-ориентированное программирование и программирование в стандарте POSIX), системам программирования (C, C++, Java, Pascal, PHP и другим) и даже



человеко-машинному взаимодействию и социальным вопросам. Открытость предлагаемой модели обучения объясняется тем, что «любые, даже самые сокровенные знания рано или поздно будут доступны в Сети».

Интернет-университет информационных технологий — это частная организация, которая ставит следующие цели:

- ▶ финансирование разработок учебных курсов по тематике информационно-коммуникационных технологий;
- ▶ координация учебно-методической деятельности предприятий компьютерной индустрии по созданию учебных курсов по ИКТ;
- ▶ обеспечение профессорско-преподавательских кадров вузов и библиотек учебниками и методическими материалами;
- ▶ содействие органам государственной власти в области развития образовательных программ, связанным с информационными технологиями.

Признание INTUIT.ru пришло незамедлительно, и проект стал лауреатом Национальной премии Российской Федерации за вклад в развитие российского Интернета в номинации «Наука и образование». У вас есть желание? Возможность уже предоставлена.

www.intuit.ru

Ужесточение регламента

Линус Торвалдс пригрозил разработчикам, что вносимые в последнюю минуту изменения со следующей версии ядра им рассматриваться не будут, а их авторам он от души посмеется в лицо. Последняя версия 2.6.14 постоянно откладывалась и вышла лишь спустя месяц после намеченного срока именно из-за ошибок, содержащихся в коде, предоставленном в самый последний момент. Комментарии разработчиков, как водится, разделились. Одни остались довольны жестким регламентом работы, другие посетовали на то,

что так хорошо в теории, а на практике многое зависит еще и от тех, кто подготавливает им работу таким образом, что уложиться в короткое временное окно сложно. В ответ Торвалдс рекомендует надавить на таких помощников и в подобных случаях вообще останавливать их работу, давая тем самым себе возможность подавать материалы в срок.

«Люди постоянно жалуются, что я становлюсь слишком мягким. Только не сейчас», — отметил Торвалдс.

news.zdnet.co.uk

Короткой строкой

C++ свергнут с неофициального трона короля SourceForge. Вершину занял Java. Мэтс Генринсон опубликовал результаты подсчета количества проектов, ведомых на разных языках, которые разделили первую шестерку следующим образом:

1. Java — 16 738 проектов;
2. C++ — 16 731 проект;
3. C — 15 934 проекта;
4. PHP — 12 175 проектов;
5. Perl — 6209 проектов;
6. Python — 4542 проекта.

www.jroller.com

По единодушному, но не простому мнению экспертов ZDNet, лучшим десктопом для малого бизнеса признан Ubuntu Linux 5.10. На «операционный стол» вместе с ним попали Mandriva Linux 2006, Novell Linux Desktop 9, Red Hat Desktop 4 и SUSE Linux 10. В оценку входили такие критерии как соединение почтового клиента с сервером Microsoft Exchange, печать на сетевом принтере и установка службы обмена мгновенными сообщениями.

reviews.zdnet.co.uk

ASPLinux совместно с фирмой 1C провела акцию по снижению цен на коробочные версии дистрибутива ASPLinux 10. Начиная с 7 ноября коробка с самым полным вариантом дистрибутива — ASPLinux v10 Deluxe — стоит 999 рублей, а коробку с вариантом ASPLinux v10 Standard можно приобрести за 499 рублей, при этом для всех версий техническая поддержка осталась в пределах прежнего объема. В случае отсутствия дистрибутива в магазинах по новым ценам его можно купить на сайтах

www.softkey.ru
www.asplinux.ru

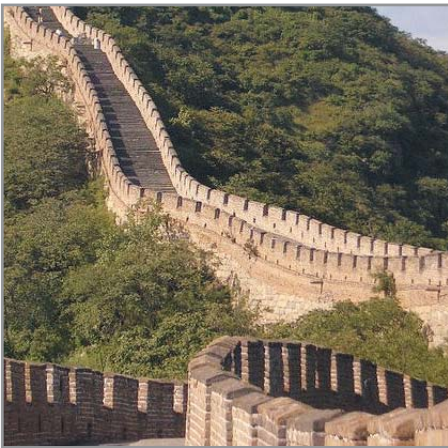


Белоруссия идет по китайскому пути?

Китай обладает самой совершенной в мире системой цензуры в Интернете, что само по себе не удивительно для страны, где даже доступ к информации считается сферой государственных интересов. «Великий Китайский firewall» включает в себя аппаратные средства контроля доступа граждан к содержимому Сети за счет фильтрации трафика — по сообщениям различных информационных ресурсов, именно к ним проявили интерес белорусские власти. Подробности в настоящее время неизвестны — возможно, ситуация прояснится после декабрьского визита руководства Белоруссии в Китай, однако тревога остается...

www.gazeta.pl

www.securitylab.ru

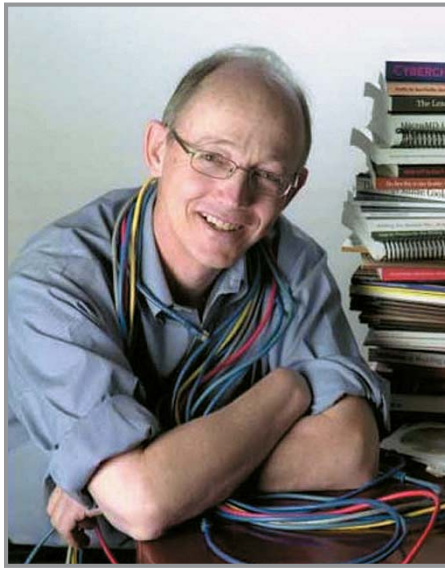


Быть ли единому стандарту Linux-mobile?

PalmSource и Orange от France Telecom объявляют о начале совместной работы по стандартизации интерфейса Linux для сотовой связи. В случае успеха союзники предполагают оказать существенное влияние на рынок и составить конкуренцию Microsoft и Symbian. В том же направлении работают FSM Labs с Motorola и MontaVista Software — с одной стороны, и Open Source Development Lab (OSDL) с собственным проектом Consumer Electronics Linux Forum — с другой.

www.news.com

Кадровая политика: кто, откуда, почему?



Уходящий год принес немало сюрпризов в отношении кадровых перемен. В июне стоявший у истоков Gentoo Даниель Роббинс перешел на работу в корпорацию Microsoft, оставив все имеющиеся у себя права на логотипы, сценарии ebuild и другую интеллектуальную собственность, прочно связанную со своим детищем, организации Gentoo Foundation, а на новом месте стал оказывать «помощь в понимании Open Source и проектов сообщества». Октябрь отозвался отказом Боба Янга от места в правлении Red Hat, который по-

святил себя проекту Lulu.com на новом независимом публикационном сетевом рынке, а ведь именно Янг был в 1993 году сооснователем компании.

В ноябре из SUSE ушел Хьюберт Мантель, один из основателей SUSE Linux. Главной причиной он назвал расхождение с диктуемым направлением развития и разочарованием в нынешних идеях компании, не соответствующих тем, что в SUSE разделяли 13 лет. Ранее, в мае, компанию покинул Ричард Зейбт, бывший президент SUSE Linux.

Не обходится без потрясений и в корпорации Microsoft. Дон Ганье, один из ведущих разработчиков Microsoft Office, заявил об отказе от дальнейшей работы в пользу своего увлечения автогонками, что было воспринято руководством как невосполнимая потеря. Единственным утешением стал только тот факт, что Ганье не переходит на работу в Google. Кроме него Microsoft к концу следующего года покинет крестный отец Windows Джим Олчин. Больше не работают в корпорации Уорд Каннингем, изобретатель Wiki, а также Хади Партови, известный своим участием в «браузерных войнах» в конце 90-х и инициативами портала MSN.

Лучше меньше, да лучше

Одним из главных доводов, который предъявляют пользователи, не спешащие переходить с Windows на Linux, является поддержка различного оборудования операционной системой от Microsoft. Дескать, зачем менять среду обитания, когда в ней работают все устройства, и подвергать себя риску возникновения проблем в новой операционной системе? Аргумент серьезный, бесспорно, но не столь однозначный. Во-первых, на стороне Linux полная прозрачность в отношении того, как работает драйвер какого-либо устройства, а также то, каким образом он создан. Во-вторых, Linux имеет множество встроенных методов тестирования некорректно настроенного оборудования и его мониторинга (достаточно взглянуть хотя бы на

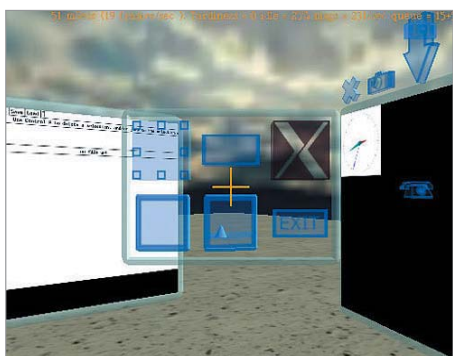
вывод `lspci` или `dmesg`). Наконец, что делать в том случае, если с вашим оборудованием не справилась и Windows? Конечно, это большая редкость, однако если это все же случилось, будьте уверены, что Microsoft сможет помочь вам — за \$129 в час. В столь экстремальном случае в Linux с горящей под ногами землей вы можете обратиться за поддержкой и поиском решения проблемы в LKML (Linux Kernel Mailing List). Наконец, есть простой момент, почему-то упущенный автором материала на pcburn.com, который всего-то и заключается в том, чтобы справиться о поддержке своего оборудования на домашней странице того дистрибутива, который вас заинтересовал.

www.pcburn.com

Три измерения на рабочем столе

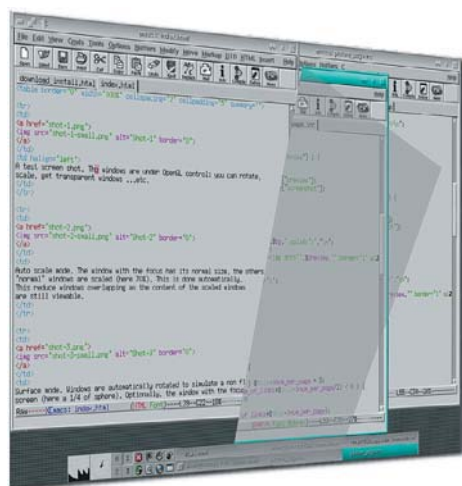


Принесет ли наступающий год 3D на рабочий стол пользователя, покажет, конечно, время. Прогнозы — дело неблагодарное. А вот на вопрос, можно ли уже сегодня присмотреться к чему-нибудь эдакому, отвечаем утвердительно. Project Looking Glass — открытая платформа для исследования инновационного трехмерного пользовательского интерфейса (<http://lg3d-core.dev.java.net>) Текущая версия — 0.7.1, доступен и самозагружающийся дистрибутив LG3D LiveCD в версии 2.3. Платформа основана на технологии Java, открыта и берет начало в недрах Sun Microsystems. Дистрибутив поддерживает запуск и работу даже немодифицированных приложений в трехмерном окружении. Существующая в настоящее время интеграция приложений поддерживает Linux и Solaris на



x86-платформах, среда разработки ПО включает еще и поддержку Windows. Проект открыт для свежих идей: авторы считают, что доступ всего сообщества к исходным текстам проекта — лучший спутник на пути к открытию «волнующих и грандиозных возможностей». Open Croquet — www.opencroquet.org. Этот открытый проект, как и Looking Glass, не ограничивается просто 3D, а добавляет к нему перспективные возможности совместной работы сразу нескольких пользователей в одном телеметрическом пространстве, что увеличивает его «сетевой вес». Архитектура синхронизации гарантирует немедленное отражение всех действий разных пользователей в трехмерном окружении. Здесь разработчики сосредоточили внимание и на взаимодействии в режиме реального времени. Графическая база Croquet — OpenGL, сращенный со скриптовым языком. В основе Croquet лежит фундаментальная идея разработки достаточно гибкой среды, в которой можно без труда и серьезных временных затрат воплотить любое представление о графическом интерфейсе. «Пределы этой системы равны пределам воображения», — утверждают ее авторы. Metisse (<http://insitu.lri.fr/metisse>) — проект не менее интересный, чем предыду-

щие. Во-первых, разработчики не ставят перед собой заоблачных целей, даже предостерегают от соблазна назвать результаты их работы рабочим столом в 3D. Буквально: вы видите не то, что вы видите, это только основанная на X оконная система с принципиально новыми возможностями управления, поддерживающая все мыслимые в пределах трех осей координат манипуляции. Авторы не предлагают нового десктопа, скорее это инструмент для создания новых типов окружения рабочего стола. В технических подробностях Metisse представляет собой совокупность инструментария nucleo, Xserver, Tight VNC и оконного менеджера FVWM, снова оказавшегося замечательной платформой для воплощения самых смелых идей (вспомните Symphony, fvwm-crystal). Предварительная версия системы доступна для скачивания, а совсем скоро ожидается выход новой, уже продемонстрированной на прошедшем в конце октября Восемнадцатом симпозиуме по программированию и технологиям пользовательского интерфейса UIST 2005 в Сиэтле. Сегодня разработчики имеют Xmetisse — сервер Metisse для приложений X Window, Xmetisse-config и/или Xmetisse.pc (для pkg-config), подробное описание протокола Metisse и



набор скриптов, облегчающих запуск и настройку Xmetisse (геометрия, глубина и пр.). Возможно, невзрачный на снимке, Metisse поразит ваше воображение, представ воочию, а с настройкой внешнего вида FVWM справится любой пылливый пользователь.

Сергей Мороз

Ворота в Сеть

Где бы мы ни находились, всегда хочется быть в курсе последних событий и знать прогноз погоды на следующий день. Владельцы смартфонов решают этот вопрос просто: в их телефонах есть браузер, который может просматривать любые веб-страницы с помощью GPRS. Обладателям телефонов более простых моделей приходится искать другие пути.

Первое, что приходит в голову, — воспользоваться WAP, благо его поддержка есть почти в каждом телефоне. Однако после недели использования от этого решения хочется отказаться: тарификация поминутная, скорость соединения не превышает 9600 бит/с, а цены и вовсе кусаются. Зачастую минута связи стоит столько же, сколько минута общения с абонентами городской сети. Тут уж не до вдумчивого чтения!

Вместо этого некоторые операторы советуют пользоваться услугой GPRS+WAP. Предложение разумное, но при анализе расценок выясняется, что провайдеры постарались сделать их такими, что выгода для абонента выглядит очень сомнительной. В результате он вынужден платить гораздо больше, чем если бы пользовался смартфоном и обычным браузером, оплачивая трафик GPRS. Чтобы понять, почему это происходит, и попытаться устранить несправедливость, надо разобратся в основах работы GPRS+WAP.

| Как работает WAP? |

WAP (Wireless Application Protocol) — протокол передачи данных по беспроводным сетям с очень ограниченной пропускной способностью. Однако в отличие от HTTP или FTP он не доставляет содержимое сайтов напрямую с сервера на телефон, а требует промежуточное звено — WAP-шлюз (в английской терминологии — WAP-Gate). Мини-браузер на телефоне использует шлюз в качестве посредника при обращении к сайтам. При этом шлюз обменивается данными с телефоном по протоколу WAP, а с сайтом — по HTTP. Это обусловлено техническими причинами: HTTP был изначально создан для компьютеров с мощными про-

цессорами и широкими каналами связи, но всего этого у телефонов нет. По той же причине для телефонов не подходит HTML. Его гибкость и сложность World Wide Web Consortium (www.w3c.org) принес в жертву строгости и простоте, разработав новый язык разметки — WML (Wireless Markup Language).

Именно таким шлюзом нам дает пользоваться оператор, когда требует плату за услугу GPRS+WAP. Обычно в руководствах по настройке телефона его обозначают предельно просто и ясно — «IP-адрес». Давайте настроим шлюз, обеспечив практически бесплатный доступ к WAP-ресурсам! Ведь даже при активном ежедневном просмотре новостей, чтении анекдотов и прогнозов погоды тратится не более 1 Мбайт трафика в месяц.

| Установка и настройка шлюза |

Нам понадобится специально предназначенная для этого программа kannel (www.kannel.org). Последняя версия на момент написания статьи — 1.4.0. Далее приведена последовательность действий для ASPLinux 10, но их можно с легкостью распространить и на любой другой дистрибутив. Создадим нового пользователя kannel. В дальнейшем будем работать от его имени. Скачаем и распакуем исходные тексты:

```
$ mkdir /usr/arc/kannel && cd /usr/arc/kannel
$ wget -c http://kannel.org/download/1.4.0/gateway-1.4.0.tar.bz2
$ tar -jxvf gateway-1.4.0.tar.bz2
$ cd gateway-1.4.0
```

Запустим скрипт ./configure со следующими параметрами:
 ▶ --disable-docs — отключить файлы документации. Опыт показывает, что при сборке документации установится большое

количество дополнительных пакетов, значительно превышающих по объему уже готовые файлы, которые можно получить на сайте производителя.

► `--enable-localtime` — отмечать в логах локальное время, а не по Гринвичу.

► `--disable-sms` — отключить возможность отправки SMS. Для работы этой опции потребуется доступ к SMS-центру оператора, которого у нас в данный момент нет.

Теперь соберем и установим исполняемые файлы в каталог `/usr/arc/kannel/bin/`:

```
$ make
```

```
$ sudo make bindir = /usr/arc/kannel/bin install
```

Синтаксис конфигурационного файла очень прост и чем-то напоминает файл конфигурации Samba. Весь файл разбит на три части, которые относятся к разным компонентам `kannel`: `bearerbox` (основной компонент шлюза), `wapbox` (компонент, отвечающий за поддержку WAP) и `smsbox` (модуль поддержки транспорта SMS). Его мы использовать не будем, поэтому соответствующую секцию оставим пустой. Каждая часть начинается со слова `group`. Команда `a = b` присваивает переменной `a` значение `b`, а если `b` строкового типа, значение следует заключить в кавычки. Строки-комментарии начинаются со знака «#». Создадим пустой файл `/usr/arc/kannel/kannel.conf` следующего содержания:

```
# Основная группа
```

```
group = core
```

```
# порт и пароль для управления kannel
```

```
admin-port = 13000
```

```
admin-password = password1
```

```
# порт, к которому будут подключаться wapbox
```

```
wapbox-port = 13002
```

```
# пароль для просмотра статуса шлюза
```

```
status-password = password2
```

```
# список разрешенных хостов для доступа к шлюзу
```

```
# адреса отделяются друг от друга точкой с запятой, «*» означает любой адрес
```

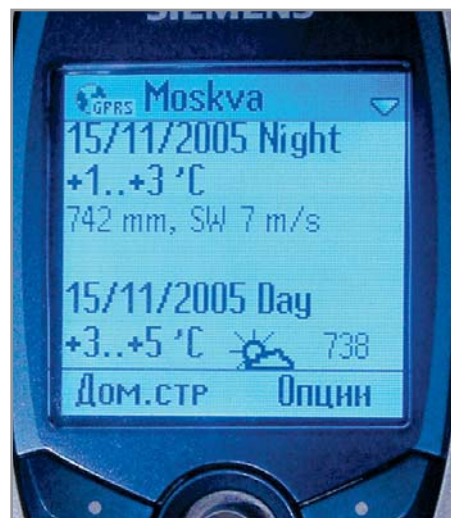
```
box-allow-ip = "*"
udp-allow-ip = "*"
# имя файла для записи логов
log-file = "/tmp/kannel.log"
# числами от 0 до 5 будут регистрироваться все события,
# уровень которых не ниже установленного
# рекомендуется поставить 0 на этапе настройки,
4 — при использовании
log-level = 0
# имя файла для регистрации сообщений о доступе к серверу
access-log = "/tmp/kannel.access.log"
# интерфейс, который будет «слушать» kannel
# укажите здесь внешний IP-адрес сервера, на который будут
приходить запросы
# формат такой же, как у параметра "box-allow-ip"
wdp-interface-name = "111.111.111.111"
# группа wapbox
group = wapbox
bearerbox-host = localhost
log-file = "/tmp/kannel.wap.log"
```

Здесь надо сделать несколько довольно важных замечаний. `Kannel`, как было сказано, состоит в нашем случае из двух модулей — `bearerbox` и `wapbox`. К одному имеющемуся `wapbox` может быть подключено несколько `bearerbox`. Например, для того чтобы распределить нагрузку на серверную машину, можно разместить `bearerbox` на одном сервере, а `wapbox` — на трех других. Директива `bearerbox-host` в группе `wapbox` указывает, на каком конкретном сервере работает `wapbox` и к какому из них следует подключаться.

Настало время проверить, все ли мы правильно сделали. Запустим `bearerbox` и `wapbox` с максимальным уровнем отладки в разных терминальных окнах:

```
$/usr/local/sbin/bearerbox -v 0 /usr/arc/kannel/gateway-1.4.0/kannel.conf
```

```
$/usr/local/sbin/wapbox -v 0 /usr/arc/kannel/gateway-1.4.0/kannel.conf
```



Не правда ли удобно всегда иметь под рукой последние новости и сводки погоды?

Внимательно просмотрите вывод этих команд. Должны появиться сообщения только типов INFO и DEBUG. Если вы видите сообщения ERROR или тем более PANIC, то постарайтесь найти причину. Это могло произойти, например, если вы неправильно указали адрес интерфейса в переменной wdp-interface-name.

Для проверки работоспособности всей связки в целом используется эмулятор WAP-клиента fakewap, входящий в состав kannel. В третьем терминальном окне перейдите в каталог с исходными текстами kannel и запустите следующую команду (вместо значения 111.111.111.111 впишите адрес сервера, на котором работает warbox):

```
$ test/fakewap -g 111.111.111.111 wap.rbc.ru
```

Если вы видите надпись «fakewap complete» — шлюз работает!

Создадим простой скрипт для запуска шлюза следующего содержания. Если вы хотите, чтобы шлюз запускался при старте системы, добавьте путь к скрипту в конец файла /etc/rc.local:

```
#!/bin/bash
sudo -u kannel /usr/local/sbin/bearerbox -v 5 -d -P
/usr/arc/kannel/gateway-1.4.0/kannel.conf
sleep 3
sudo -u kannel /usr/local/sbin/wapbox -v 5 -d -P /usr/arc/kannel/gateway-1.4.0/kannel.conf
```

Как вы, наверное, заметили, добавилось два параметра. Параметр -d запускает сервисы в виде демонов. Второй, дополнительный, параметр -P обеспечивает автоматический перезапуск

сервиса в случае его краха. К сожалению, это нужный аргумент. Бывает, warbox «падает» даже от сканирования портов с помощью nmap. Трехсекундная пауза между запуском bearerbox и warbox необходима, чтобы bearerbox успел проинициализироваться, перед тем как warbox начнет к нему подключаться.

Настройка телефона

Для работы со свеженастроенным шлюзом надо подходящим образом подготовить телефон. Подробное руководство вы можете найти на сайте своего оператора и выставить все настройки согласно его рекомендациям. Изменить нужно лишь один параметр — в одних аппаратах он называется «IP-адрес», в других — «шлюз». Сделайте его равным IP-адресу сервера, на котором установлена kannel. Теперь выберите GPRS как предпочтительный способ соединения. Телефон готов к работе!

Последний нюанс

Итак, теперь у вас есть полнофункциональная замена WAP-шлюзу оператора. Она не требует денег, ее работа зависит только от вас, что, согласитесь, немаловажно. Особо следует отметить, что использование этого шлюза абсолютно законно и не нарушает условий контракта. Однако в качестве меры предосторожности не стоит публиковать его IP-адрес, потому что заблокировать доступ к нему со стороны оператора очень просто (но на функциональности сервера это, конечно, не скажется).

Дополнительные услуги

Читаем почту

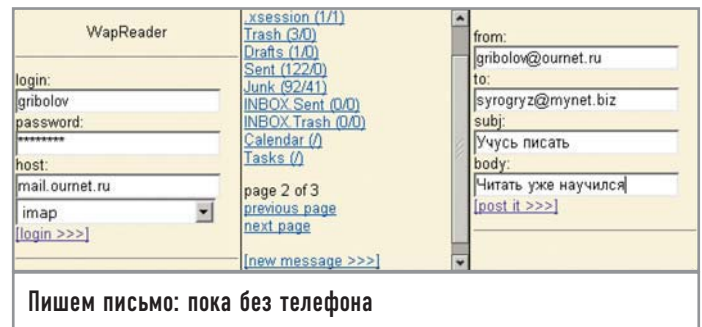
Пользователи сервиса Mail.ru вскоре обнаружат возможность чтения и отправки электронной почты через WAP-интерфейс. Но что делать, если администратор сервера, на котором располагается ваш почтовый ящик, не озаботился настройкой подобной службы? Сделать это самим! Для этого понадобится почтовый клиент с WAP-интерфейсом warreader. Его можно скачать с сайта <http://warreader.dp.ua/ru/about>. Он может работать по протоколам IMAP и POP3, поддерживает защищенные соединения SSL и позволяет не только читать, но и отправлять новые письма. Установите и настройте веб-сервер, например Apache. Вполне подойдут начальные настройки. Пусть корневой папкой для него станет /var/www/html. Создайте

в ней папку mail и распакуйте туда файлы из папки warreader следующего архива: <http://citkit.sourceforge.net/sourceforge/warreader/warreader-1.1.tar.bz2>. Установите модуль php-imap для поддержки протокола IMAP в PHP:

```
# yum install apache php-imag
```

Опять же будем считать адресом сервера, например, 111.111.111.111. Введите следующий адрес в строке вашего браузера (на компьютере): <http://111.111.111.111/mail/warreader.php?mode=html>. На экране должно появиться приглашение ввести имя почтового сервера, учетной записи и пароля. Именно это вы увидите на своем телефоне, когда warreader будет полностью готов к работе.

Если приглашение не появилось, попробуйте открыть



страничку <http://111.111.111.111/mail/test.php>. Там вы обнаружите более подробное описание найденных ошибок. После того как вы убедились, что все подобающим образом работает, можно приступить к чтению почты с помощью телефона. Для этого зайдите на страницу <http://111.111.111.111/mail/warreader.php>. Если не хотите вводить имя пользователя, пароль и другие данные при каждом

заходе на сервер, добавьте в закладки телефона ссылку следующего вида: <http://111.111.111.111/mail/warreader.php?<login>:<password>:<host>:<protocol>>, где <login> — имя учетной записи; <password> — пароль; <host> — доменное имя почтового сервера, <protocol> — IMAP или POP3. Она может выглядеть, например, так: <http://111.111.111.111/mail/warreader.php?gribolov:mukhomor@mail.ournet.ru:imap>.

Даниил Шидловский

Электронный управленец

Проблема поиска постоянного или дополнительного заработка с помощью разработки свободного программного обеспечения легко решается при наличии опыта, знаний и умения небанально мыслить.

Радость от совместной жизни с любимой девушкой длилась недолго. Остро встал финансовый вопрос. А если учесть, что я ленивый, к тому же еще и разгильдяй, не способный работать в команде, то проблема представлялась очень даже серьезной. Между прочим, мое трудоустройство на постоянную (даже высокооплачиваемую) работу не устраивает и мою девушку тоже. Кроме денег она требует постоянного внимания.

В раздумьях о заработке я отправился на книжный рынок и купил издание из серии «Бедный папа, богатый папа». Одна мысль меня очень заинтересовала: чтобы стать обеспеченным — лучше получать \$100 в месяц и почти ничего не делать, чем \$5000, вкалывая от зари до зари. Потому что \$100 пассивного дохода позволят найти следующую сотню. А потом еще. И еще.

Подготовка

Дальше я поведаю вам о том, как моей мечте удалось воплотиться в реальность. Мы вместе пройдем весь путь от поисков, робких экспериментов и закончим решением тех проблем, с которыми я столкнулся при коммерческой эксплуатации продукта, получившего название «Докотека».

Итак, начнем с конца. Нужны деньги. Не очень большие, но систематически. У кого есть деньги? У больших компаний. Имея определенный опыт делового общения, я сразу отказался от этой идеи. Ведь там сплошная политика. А я далеко не Лебедев.

Другой вариант — обычные пользователи. Также мимо кассы. Тот, кто настраивал компьютеры частным лицам, меня поймет. Работы очень много, денег — кот наплакал. Остановился на следующем варианте: мои клиенты — владельцы небольшого бизнеса. Зарегистрировав предприятие, я сам легко и непринужденно попал в этот круг. И сразу остро почувствовал необходимость в инструменте управления информацией. И теперь он у меня есть — «Докотека».

Используя личный опыт и несколько замечательных продуктов с открытым исходным кодом, вы сможете запустить прекрасный сервис практически не выходя из дома. Да, большинство клиентов привозят мне деньги прямо домой. Ценят. Вот неполный список необходимых в работе продуктов: Linux, Python, Roundup, PostgreSQL, Mtacs, Freemind.

Что необходимо разработчику?

Внутреннее содержание проекта должно отвечать нескольким критериям. Как оказалось, примерно следующим:

- ▶ очень простое (лень, знаете ли);
- ▶ распространенное;
- ▶ на Python;
- ▶ с открытыми исходными кодами;
- ▶ созданное профессионалами, желательно для себя.

Внешний облик должен иметь веб-интерфейс (но очень быстрый). Проект должен работать на небольшом сервере на основе недорогой машины под управлением, естественно, Linux. И еще обязательно нужно использовать MySQL с возможностью перехода на PostgreSQL.

В ходе поисков оптимального решения я перебрал несколько проектов. Перечислю их, вкратце упомянув плюсы и минусы.

- ▶ Roundup (roundup.sourceforge.net). Собственно, понравилось все: дизайн, документация, исходный код. Стабильное развитие.
- ▶ Zope2 (www.zope.org). Среда с непростой системой безопасности и требовательна к ресурсам. Использую для веб-разработок.
- ▶ Zope3 (www.zope.org). Не потянул. Отладка ZCML (это больше, чем конфигурационный файл) показалась для меня сущим адом.
- ▶ Quixote (www.mems-exchange.com). Хорошо. В связке с SQLAlchemy пишется очень легко. Но захотелось чего-нибудь нового (для меня) и более распространенного.

► Tiny erp (www.tinyerp.org). Сильный продукт. Обнаружил его, когда «Докотека» уже была на полпути к клиенту. Уверен, что еще вернусь к нему.

Остановился на Roundup по той причине, что этот продукт используют профессионалы, к тому же он позволяет менять схему данных прямо на лету. А это для меня самое важное, ведь вначале я не имел представления о том, какие структуры буду задействовать. Кроме того, Roundup автоматически ведет журнал операций, работает с dbm, MySQL, PostgreSQL, SQLite на выбор. Данными можно управлять из командной строки, через Интернет или SMTP.

Задачи, решаемые «Докотекой»

Какие же задачи должна уметь решать система? Руководителю бизнеса от нее требуется следующее:

- централизованное хранилище файлов с доступом через Глобальную сеть;
- отчет о том, кто какую работу выполнил и сколько времени это заняло;
- быстрый доступ к нужному файлу;
- резервное копирование всей информации;
- обмен сообщениями.

Разработка

Ну что, поехали? Для начала я поднял Linux на стареньком Celeron 500. Затем скачал и установил следующие компоненты:

- Python (интерпретатор языка программирования);
- MySQL (движок базы данных);
- MySQLdb (модуль Python для работы с MySQL);
- Roundup (bug tracking).

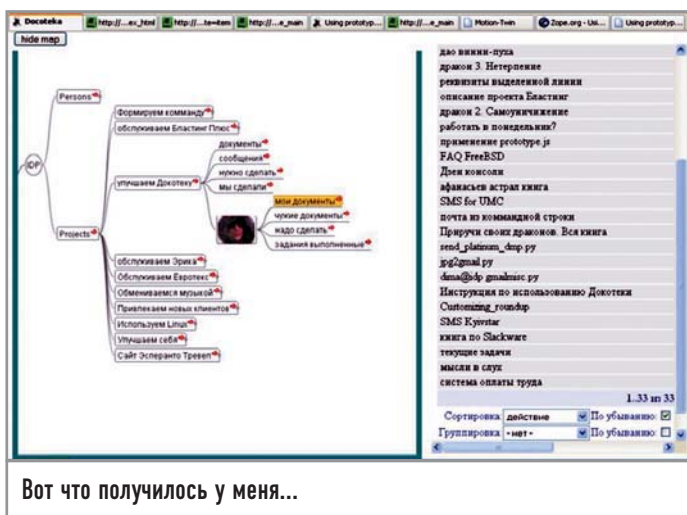
Затем установил администраторский модуль roundup, набрав в командной строке:

```
$ roundup-admin install
```

При установке выбрал шаблон (template) classic и MySQL в качестве backend:

```
$ emacs /trackers/classic/config.ini
```

Здесь я редактирую переменные. Самые важные из них — web, domain, host. Теперь необходимо указать пароль администратора и каталог с трекером:



```
$ roundup-admin initialise
```

```
$ roundup-server classic=/home/myhome/trackers/classic
```

И можно запускать roundup.

А затем открывать любимый Firefox на <http://localhost:8080/classic>.

Получился баг-трекинг, полностью готовый к работе. Уже локализованный! Очень хорошо. Но этого мало. Распечатал `customising.html`, `client.py` и поехал на неделю на дачу (компьютер не брал). Изучал до просветления. Стал думать над интерфейсом. Не нашел лучшего решения, чем дерево и список. Roundup замечательно работает со списками из коробки. А вот с деревом не все так просто. Может, есть варианты получше? Случайно наткнулся на продвинутую модель дерева — MindMap. Дальше MindMap я буду называть ментальной картой; подробнее о ней можно узнать на сайте www.mindmap.ru

Нашел свободный продукт для работы с ментальными картами — Freemind (<http://freemind.sourceforge.net>). На скриншоте вы можете видеть пример его работы.

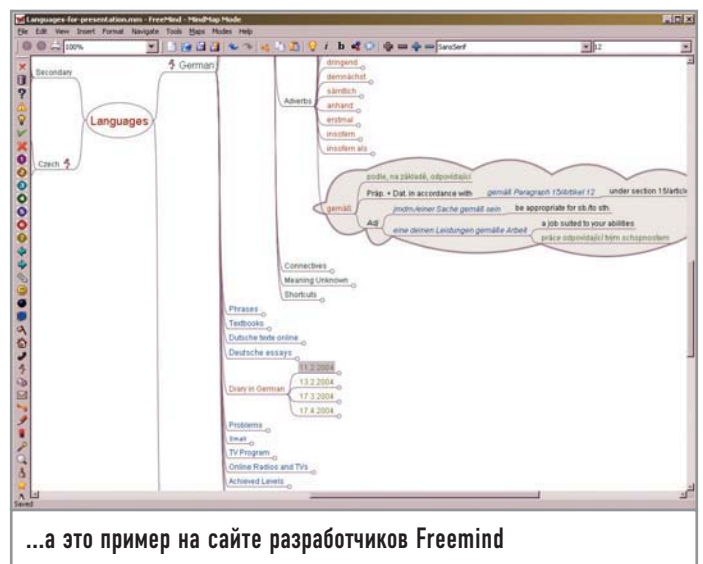
Потом написал скрипт на Python, который генерирует XML, на основе которого Freemind строит ментальную карту.

Жаль только, что для отображения ментальной карты в браузере приходится использовать java plugin. И тут удача улыбнулась мне. По адресу <http://cvs.sourceforge.net/viewcvs.py/freemind/flash> я нашел исходники к SWF-файлу (Macromedia Flash, если кто не знает), который отлично отображает ментальную карту в Flash plugin для браузера.

И что самое главное, этот `visorFreemind.swf` создан с помощью свободного компилятора (подробнее о нем вы можете узнать на сайте www.mtasc.org). Какая чудесная находка!

Видение того, каким должен быть интерфейс, пришло ко мне во сне: слева — ментальная карта, причем у каждого пользователя своя; справа — фрейм для отображения данных. При нажатии на узел слева или справа открывается форма для ввода данных или список.

Проснувшись, сразу же сел за компьютер. Помедитировал над исходниками `visorFreemind.swf` и внес два небольших изменения. В файле `Main.as` убрал строку `System.useCodepage = true`, а в `Node.as` во всех вызовах `getUrl` добавил еще один параметр —



right_frame, чтобы ссылки открывались в правом фрейме (да, да, я помню, что фреймы использовать не рекомендуется, но ничего лучше конструкции с тегом <iframe> не придумал).

Компилирую с учетом того, что исходники лежат в каталоге /home/myhome/freeMindFlash/visorFreeMind, следующей строкой:

```
$ mtask -cp ./freeMindFlash/visorFreeMind/Main.as -swf
visorFreemind.swf -header 600:600:24 -separate
```

Несколько слов о том, как будет храниться информация. В Roundup схема данных представлена множеством классов, размещенных в файле schema.py. Вот выдержка из моего schema.py, описывающая класс Document:

```
doc = Class (db, "doc", title=String(), file=Link("file"),
state=Link("docstate"), doctype=Link("doctype"),
project=Link("project"), interesants=Link("user"),
versions=Multilink("file"), messages=Multilink("msg"))
```

Здесь, doc, file, doctate, doctype, project, user, msg — наименования классов. String, Number, Date, Interval, Number — свойства экземпляров. Link, Multilink — связи между экземплярами классов. Если что-то не ясно, то документ customising.html — ваш лучший друг!

Стоит заметить, что добавить или удалить класс, свойство или связь я могу просто посредством редактирования файла schema.py. Таблицы в базе данных изменит за меня Roundup. Здорово, не так ли? Но нужно соблюдать известную осторожность.

Запросы — одна из удивительно приятных и мощных особенностей Roundup. Вот как будет выглядеть URL, для того чтобы вывести список документов созданных мной. Этот URL привязан к узлу «Мои документы», как показано на скриншоте:

```
http://mytracker:8080/docotheka/doc?@template=index&@columns=title,file&@filter=creator&creator=3&@pagesize=50&@sort=-activity
```

Необходимые пояснения:

- ▶ doc?@template=index — используется шаблон doc.index.html;
- ▶ @filter=creator&creator=3 — будут выведены документы, где свойство creator = 3;
- ▶ @columns=title,file — будет две колонки — заголовок и ссылка на файл;
- ▶ @pagesize=50 — 50 записей на странице;
- ▶ @sort=-activity — документы, изменявшиеся последними, будут расположены наверху.

Дополнительная информация

Легкий путь к AJAX

С содроганием вспоминаю, как сразу после изучения Perl столкнулся с необходимостью писать на JavaScript. Кошмар. Поэтому всегда старался как можно больше всего делать на стороне сервера. Но вот увидел, что эти парни из Google сделали из почтового клиента, и

теперь приобрел к модному слову AJAX. Трудностей было много, пока не наткнулся на замечательную статью одного специалиста, где он рассказывает о том, как использовать prototype.js. Теперь все наладилось. Рекомендую — www.sergio-pereira.com/articles/prototype.js.html.

А теперь поговорим о шаблонах, благодаря которым мы выводим формы для ввода данных и показываем списки. Кстати, практика показала, что вводить данные через формы практически не приходится. Нужно максимально интегрироваться в существующий документооборот. Но об этом немного позже.

Шаблоны представлены файлами в папке html и написаны с использованием ZPT (Zope Page Template). Изучить этот как бы язык оказалось для меня самым сложным. Но мотивировав себя тем, что это решение для предприятий, которое широко используется не только в Zope (а и в Roundup, например), и вооружившись отличной документацией (www.zope.org/Documentation/Books/ZopeBook/2_6Edition/ZPT.stx), осилил это дело. И оно пошло лучше, когда я осознал, что, используя ZPT, нужно придерживаться простого правила: каждый тег может быть рассмотрен отдельно; никаких else.

Дальнейшую радость открытий и удовольствие от постижения исходного кода оставляю вам. Подробности и документация находятся на <http://roundup.sourceforge.net>, а также на моей страничке www.idp.com.ua/docotheka.

| Внедрение |

После того как разработка приложения вошла в свою колею, настало время вернуться к основной задаче: как на этом заработать? Составил небольшой список того, что может программный комплекс, и пригласил потенциального клиента на кружку пива. Далее расскажу о выводах, которые мне пришлось сделать.

| Что вызывает интерес |

- ▶ Дистанционный контроль над деятельностью сотрудников.
- ▶ Скорость доступа к информации (надо показывать).
- ▶ Низкие расходы.
- ▶ Резервное копирование данных на Gmail.
- ▶ Трудовая дисциплина.

Интерфейс и особенно его скорость клиенту понравились. Все-таки технология Flash имеет в этой области неоспоримые преимущества. Еженедельный отчет о деятельности сотрудника стал своеобразной «бомбой». Выяснилось, что клиент может платить \$150 в месяц. Я подумал и решил понизить цену, чтобы оплата включала предоплату и происходила без задержек: €100 в месяц. Ударили по рукам.

Приехал, установил. В качестве движка базы данных использовал Postgresql. Первый день работы принес несколько проблем, среди которых — небольшой баг в генерации карты и непонимание пользователями принципов работы. Пришлось установить VNC (www.tightvnc.com) на каждый компьютер клиента. После этого дело пошло гораздо быстрее.

Главная проблема возникла через шесть недель. «Докотекой» перестали пользоваться. Сотрудники предпочитали по старинке закидывать файлами свои рабочие столы. Не стал с ними бороться, так как и сам такой же ленивый. Решением стал описанный в прошлом номере небольшой скрипт, который в конце дня обходит рабочие каталоги пользователя и все измененные файлы загружает в «Докотеку». Дело наладилось. А за одним клиентом потянулся другой... Теперь обкатываем учет платежей. В следующий раз отчитаюсь. |

Свободу хамелеону

Похоже, очередная версия SUSE Linux, финальный релиз которой не так давно вышел в свет, станет новой вехой в истории этого дистрибутива. Дело в том, что компания Novell (нынешний владелец SUSE Linux AG) приняла решение открыть для сообщества Open Source разработку SUSE Linux, так что теперь каждый пользователь может внести свою лепту в создание любимого дистрибутива.

Ранее никакой самостоятельности не допускалось, разработку вела сравнительно небольшая команда людей. Кроме того, ISO-образы очередного релиза дистрибутива выкладывались только после того, как его коробочная версия поступала в продажу.

Похоже, что Novell решила пойти по стопам компании Red Hat, основавшей в свое время проект Fedora, который занимается выпуском одноименного дистрибутива, используемого для тестирования новинок и сборки собственного решения для корпоративных заказчиков — Red Hat Enterprise Linux.

SUSE Linux 10 OSS — это первый релиз SUSE Linux, выпущенный в рамках проекта openSUSE, и именно на его основе Novell будет разрабатывать новую версию Novell Linux Desktop, рассчитанного на корпоративный рынок. Трудно предсказать, станет ли сам дистрибутив от этого более качественным, однако популярность SUSE после выхода последней версии значительно возросла. Есть мнение, что в openSUSE компания Novell будет обкатывать и тестировать различные технологические новинки и самые свежие версии ПО, прежде чем включать их в Novell Linux Desktop. Что ж, судя по наличию в SUSE 10 таких вещей как Xen (новая технология виртуализации, которую уже заочно назвали «убийцей виртуальных машин») и Novell AppArmor (которая, по идее, должна защищать от эксплоитов), с этим мнением можно согласиться.

Что же получает обычный пользователь? А ему достается свободный дистрибутив SUSE с самыми свежими версиями программного обеспечения, ISO-образы которого можно загрузить с сайта www.opensuse.org/download. Кстати, практиче-

ски сразу после выхода десятого релиза была анонсирована альфа-версия 10.1, которую можно загрузить по тому же адресу. Выход финальной версии 10.1 намечен на март-апрель 2006 года, а пока можно смело качать SUSE 10.

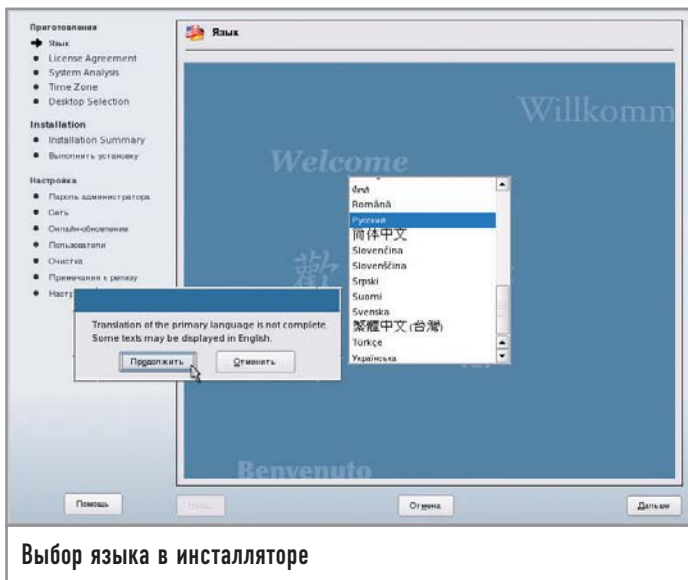
Сразу хочется предупредить, что качать придется много: для полноценной установки SUSE Linux 10 потребуется загрузить образы всех пяти дисков. Впрочем, при сегодняшних ценах на трафик это не так много. Кроме того, SUSE Linux 10 уже появился во многих российских и украинских интернет-магазинах, занимающихся продажей дистрибутивов Linux и BSD, где вы сможете приобрести его по символической цене.

| Дружелюбный YaST |

Инсталляционная программа в SUSE Linux (YaST) всегда была одной из лучших в своем классе, а по своей простоте и логичности с ней смог бы сравниться разве что установщик Mandriva/Mandrake. При этом YaST не кажется чем-то ущербным или примитивным, как, например, инсталляторы Linspire и Xandros, которые не обладают гибкостью YaST. В процессе установки вам всего лишь придется ответить на пару-тройку несложных вопросов.

Уже с момента запуска установочного диска заметно, что разработчики SUSE Linux попытались скрыть линуксовое нутро подальше от глаз пользователя — сообщений ядра вы не увидите, вместо них вам придется смотреть на красивую полосу загрузки.

Сам YaST выглядит очень неплохо (спасибо Trolltech за отличную тему для виджетов в Qt), хотя грузится очень неторопливо, в



Выбор языка в инсталляторе

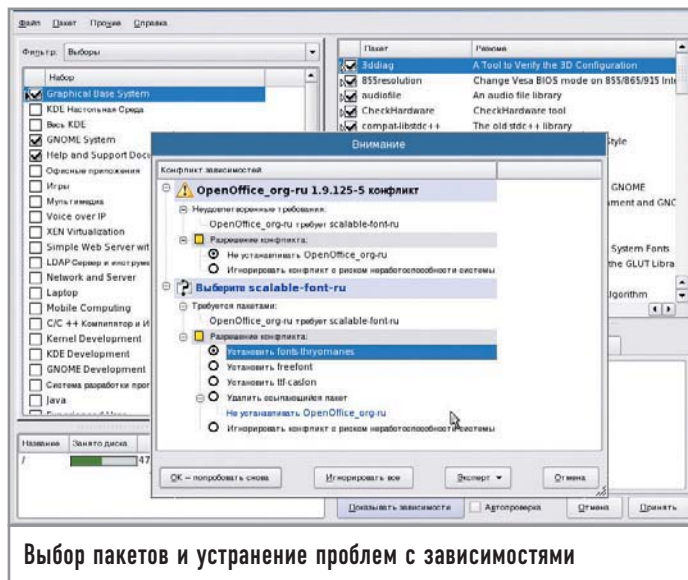
отличие от инсталлятора Mandriva, который тоже написан с помощью Qt. На первом этапе вам предложат выбрать язык — приятно, что среди чешского и монгольского присутствует русский (его даже отнесли к европейским). Только вот кодировку выбрать невозможно — по умолчанию для всех языков в SUSE Linux используется UTF-8. Что касается самого YaST, то на русский он переведен очень плохо, и если, начиная установку, вы еще сможете разглядеть русские слова, то ближе к ее завершению найти в интерфейсе YaST хоть один кириллический символ будет уже очень сложно. Такое ощущение, что русский перевод интерфейса установщика не редактировался с версии 8.2. На мой взгляд, лучше вообще вырезать русский интерфейс YaST, чем включать такой. Будем надеяться, что среди отечественных пользователей SUSE Linux найдется кто-нибудь, кто не поленился переработать перевод YaST.

Любопытные факты

Из истории дистрибутива

Невероятно, но факт: первые версии дистрибутива SUSE Linux были основаны на... Slackware! Да, небольшая немецкая фирма S.u.S.E. (сокращение от «Software und System Entwicklung») первоначально специализировалась на консультациях по Unix, а также выпуске печатной документации по этой системе. В 1994 году компания издала диск с немного измененной версией Slackware Linux, который был назван S.u.S.E. Linux 1.0. Первый же собственный дистрибутив был выпущен только в 1996 году, им стал S.u.S.E. Linux 4.2, который уже тогда он приобрел

значительную популярность в Германии и странах Западной Европы. Вскоре S.u.S.E был переименован в SUSE, а в 2003-м компания SUSE Linux AG была куплена Novell, имеющей планы по разработке собственного дистрибутива для корпоративного рынка. По данным www.distrowatch.com, SUSE уже третий год входит в десятку самых популярных дистрибутивов Linux в мире. Причем сейчас он находится на третьей позиции, уступая лишь амбициозному новичку Ubuntu Linux и сверхпопулярному в нашей стране Mandriva/Mandrake.



Выбор пакетов и устранение проблем с зависимостями

Если не обращать внимание на перевод, YaST — почти совершенство. Существуют два режима установки — для новичков и экспертов, причем между режимами можно переключаться в ходе установки: например, разметку диска проделать в режиме эксперта, а все остальное — в режиме новичка. Если у вас не запускается графический вариант YaST, систему можно установить из командной строки в псевдографическом интерфейсе YaST.

На моем ноутбуке (Compaq M700) YaST корректно распознал все устройства и даже справился с настройкой капризного PCMCIA Wi-Fi-адаптера D-Link DWL-D630 Rev. C (до сих пор ни один инсталлятор вот так, с ходу, его не признавал).

В ходе установки вас спросят, в какой графической среде вы будете работать — KDE или GNOME. К сожалению, других вариантов не будет предложено — несмотря на то, что в дистрибутиве присутствуют и blackbox, и icewm, и даже fwm. От этого выбора зависит набор программного обеспечения в стандартном варианте установки.

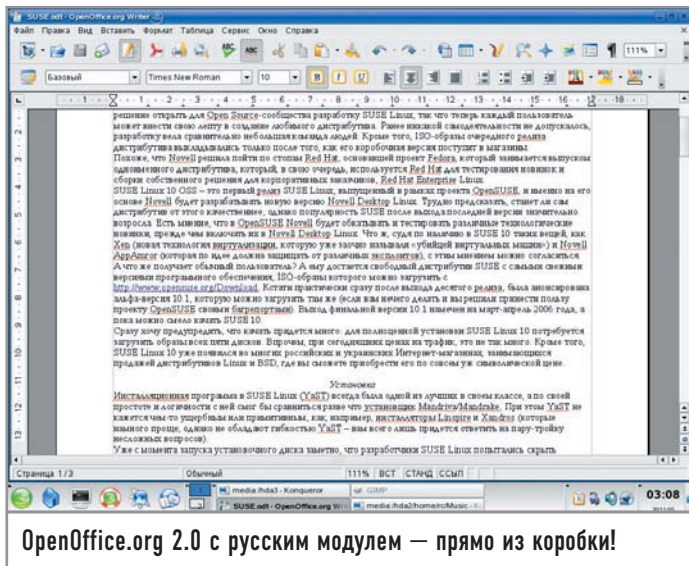
После того как пакеты с первого инсталляционного диска будут установлены, YaST выполнит перезагрузку, после которой установка будет продолжена (сам YaST будет работать уже с жесткого диска).

Стоит отметить удобную утилиту разметки диска, позволяющую изменять размеры разделов без потери информации, причем она работает даже с NTFS-разделами!

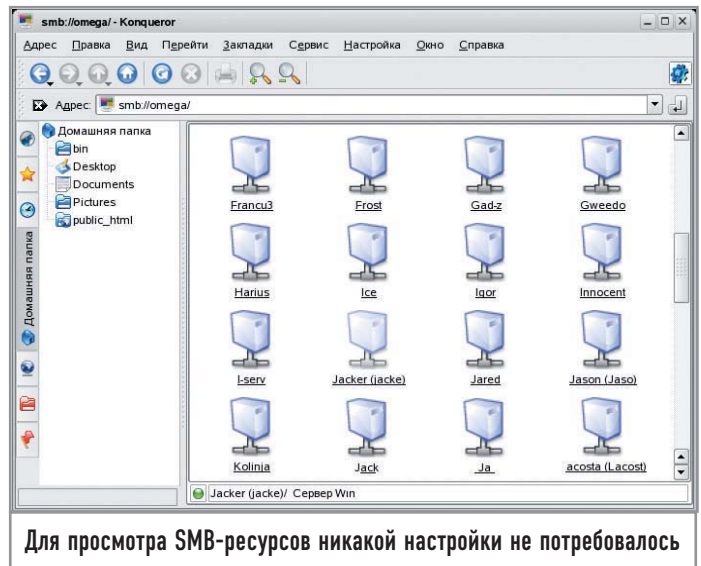
В целом YaST великолепно справляется со своей задачей, при грамотной установке никакой доработки или ручного редактирования конфигурационных файлов не потребуется. Даже недоработанная русификация не портит общей картины.

| SUSE — пользователю |

Если у вас не самый мощный по сегодняшним меркам компьютер, то во время первоначальной загрузки вы успеете выпить чашку кофе, а может, даже две. Причина такой неторопливости в слишком большом количестве служб, загружаемых по умолчанию. Например, демон печати (cups) стартует даже в том случае, если на этапе установки принтер обнаружен не был. К счастью, автоматический запуск всех ненужных сер-



OpenOffice.org 2.0 с русским модулем — прямо из коробки!



Для просмотра SMB-ресурсов никакой настройки не потребовалось

висов можно отключить в конфигураторе YaST2 («System → System Services (Runlevel)»).

Набор пользовательского программного обеспечения довольно стандартный: KDE 3.4.2, GNOME 2.12, Mozilla Firefox, Kopete, Gaim... Все это есть практически в каждом дистрибутиве. А вот что хотелось бы отметить, так это OpenOffice.org 2.0 сборки фирмы Novell, который выглядит как обычное KDE-приложение (сборка с Qt), а не как бедный родственник с интерфейсом наподобие Windows 98. Приятно удивило наличие русского модуля для OpenOffice.org с проверкой орфографии и расстановкой переносов, который устанавливается автоматически, если во время инсталляции вы выбрали русский язык. Кроме OpenOffice.org в дистрибутиве есть KOffice, GNOME Office (Abiword, Gnumeric, GIMP), а также коммерческие пакеты Textmaker и Planmaker от SoftMaker Software GMBH.

Кстати, если в качестве графического окружения в ходе установки вы выбрали KDE, то все GTK-программы будут использовать текущую цветовую схему и стиль KDE (это реализовано с помощью GTK Qt engine). Жаль, что Qt/KDE-программы так и не научились «маскироваться» под GTK.

Никаких проблем с воспроизведением MP3 обнаружено не было: и в AmaroK, и в RealPlayer 10 включены MP3-модули, причем по умолчанию используется движок Helix (который можно сменить на xine-lib). А вот для нормального просмотра DVD нужно будет скачать и установить пакет libdvdcss, так как xine в SUSE Linux 10 поставляется с ограниченным набором кодеков.

Популярный норвежский браузер Opera тоже вошел в состав SUSE 10, но устанавливать его не рекомендуется: в дистрибутиве присутствует версия 8.02, из которой еще не вырезан баннер (в бесплатной 8.50 его уже нет), поэтому лучше скачать свежий релиз с сайта www.opera.com/download.

Хотелось бы высказать претензии к разработчикам по поводу локализации в UTF-8, да не за что: русские шрифты в командной строке выглядят вполне адекватно, да и Midnight Commander, у которого всегда были проблемы с UTF-8, с помощью патчей заставили уважать Unicode.

Как вы думаете, за что неопытные пользователи так любят SUSE Linux? Верно, за потрясающий и практически всеобъемлющий си-

стемный конфигурактор YaST2, в котором можно настроить все — от загрузчика GRUB до соединения с Интернетом, от NFS-сервера до параметров Bluetooth. Установка и обновление программного обеспечения тоже производится с помощью YaST2. По количеству функций YaST2 опережает даже Mandriva Control Center! Но гвоздика в том, что Mandriva Control Center практически полностью переведен на русский язык, а вот уровень русификации конфигура-тора YaST2 даже хуже, чем у инсталляционной программы.

| SUSE — разработчику |

SUSE Linux 10 содержит последнюю экспериментальную версию GCC — 4.0.2, причем более ранних версий в дистрибутиве нет, поэтому GCC 3.4.3, например, придется скачивать отдельно. Не очень хороший шаг со стороны разработчиков, все же четвертая версия GCC еще слишком далека от идеала. Кроме традиционных Perl, Python, Ruby и Tcl в SUSE включен Clisp.

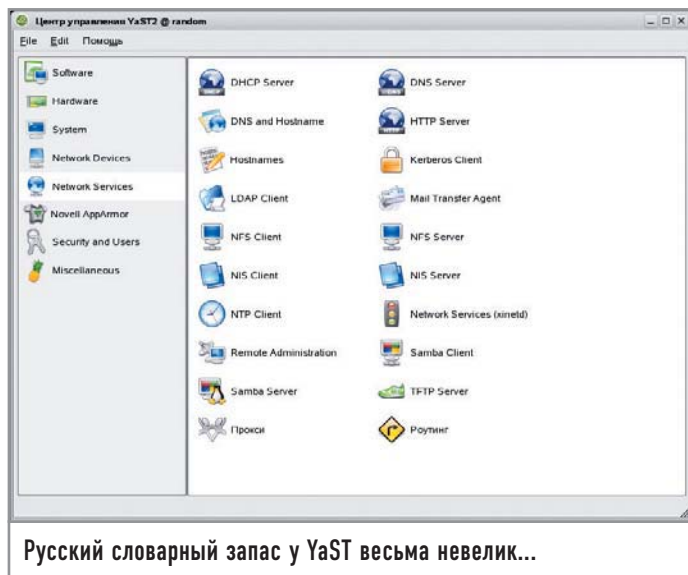
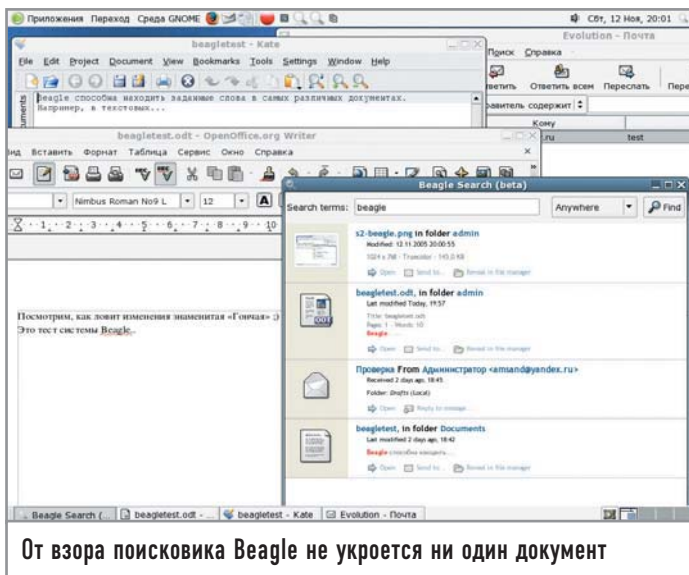
Есть и Java, причем как компилятор из пакета GCC (gcj), так и официальная версия от Sun, а вот Eclipse в дистрибутиве отсутствует, что по меньшей мере странно, особенно если учесть популярность этой IDE. Среду разработки для Python — Eric3 — почему-то тоже обделили вниманием (хорошо, что про Kdevelop и Glade не забыли). Отсутствует и FreePascal.

Взгляд IBM на Linux

Под присмотром «Голубого гиганта»

Как известно, корпорация IBM в последнее время уделяет очень большое внимание движению Open Source и, в частности, операционной системе Linux. Значительное число собственных продуктов компании поддерживает как Windows, так и Linux. Однако в IBM не считают целесообразным обеспечивать полноценную под-

держку всего многообразия имеющихся дистрибутивов. Политика компании нацелена на сотрудничество с двумя ведущими разработчиками, которыми считаются Red Hat и Novell. Такое признание со стороны «Голубого гиганта» еще раз подтверждает лидирующие позиции дистрибутивов Novell, в том числе и SUSE, на рынке Linux-систем.



Зато в релизе удалось обнаружить экспериментальную версию пакета Gambas (очень своеобразной реализации языка, подобного Visual Basic), содержащую IDE с визуальным дизайном форм, интерпретатор и отладчик. Зачем нужно было включать в дистрибутив нестабильную версию 1.9.13 (которая постоянно «падала»), нам до сих пор не понятно, тем более что доступен стабильный релиз 1.0.x.

Очень удивило наличие множества пакетов для mono: тут есть и mono-web (реализация ASP.NET), и mono-basic (runtime-библиотека VB.NET), и monodevelop, и даже mono-winforms (пока еще очень сырая реализация Windows Forms). При этом самих программ, написанных для Mono и не относящихся к разработке, в дистрибутиве целых пять: beagle, blam, f-spot, banshee и tomboy. Странно, не правда ли? Такое ощущение, что разработчики хотели превратить SUSE в испытательный полигон: кто сможет на этом писать — тому приз.

Порадовало, что все GTK-программы собраны с библиотекой GTK 2.8, которая для отрисовки виджетов использует векторный графический движок Cairo.

Технологии

KDE vs GNOME

Эти два мощнейших оконных менеджера издавна ведут борьбу за господство на рабочих столах пользователей. Судя по опросам, время от времени проводимым на различных форумах, KDE все же пользуется большей популярностью, по крайней мере у российских пользователей. Однако компания Novell в своих дистрибутивах SUSE Linux Enterprise Server и Novell Linux Desktop по умолчанию стала предлагать GNOME (что привело к уходу

из компании нескольких ведущих разработчиков из старой команды SUSE). Тем не менее право последнего голоса по-прежнему остается за пользователем, и полноценная поддержка обоих менеджеров неизменно включается практически в любой дистрибутив Linux. Кстати, в SUSE 10.0 выбор графического менеджера — пожалуй, единственный пункт установки, где разработчики вообще не предлагают ничего по умолчанию.

Ни обещанного Qt4, ни PHP5 в дистрибутиве нет. Думается, первый не включили лишь потому, что спектр приложений, написанных с помощью этой библиотеки, пока ограничивается лишь учебными примерами из комплекта самого Qt4. Вообще, подбор пакетов для разработки выглядит довольно интересно и своей контрастностью наводит на тягостные раздумья.

Прочие повадки хамелеона

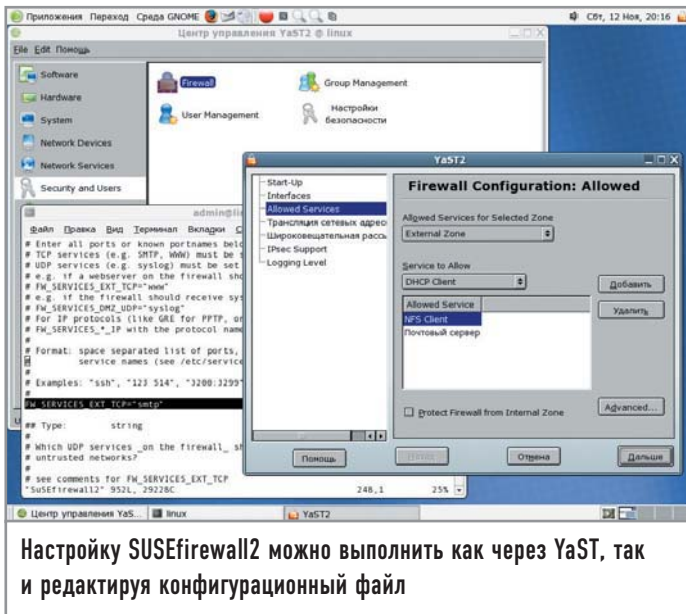
В эту версию SUSE наконец-то был включен art4rpm — популярный среди пользователей Debian/Ubuntu менеджер пакетов, адаптированный для работы с RPM. Вообще-то, art для SUSE существовал еще с версии 8.2, но его приходилось докачивать и устанавливать отдельно. Первичная адаптация art-get к RPM-дистрибутивам была проведена компанией Connectiva (ныне Connectiva входит в состав Mandriva). Странно, что в SUSE не включили какой-нибудь графический интерфейс для art, например тот же Synaptic.

Кроме art в дистрибутиве обнаружился yum, который изначально разрабатывался как аналог art-get для основанных на RPM дистрибутивов. Впрочем, с появлением art4rpm необходимость в нем вроде как отпала, но разработчики, видимо, решили оставить выбор пользователям. Стоит ли говорить, что одновременная работа с yum и art-get чревата непредсказуемыми последствиями вроде интересных глюков в самых неожиданных местах?

Очень порадовал широкий выбор разнообразных шрифтов — от популярного консольного terminus до Bitstream Vera от Bitstream Inc. Есть даже такая экзотика как tv-fonts, рассчитанный на использование в роли OSD-шрифта в ТВ-приложениях.

Неприятно удивило практически полное отсутствие научных пакетов, таких как maxima и octave (gnuplot все же нашелся), что непростительно для дистрибутива, занимающего пять компакт-дисков. Впрочем, это лишний раз подтвердило, что «русские физики выбирают Slackware».

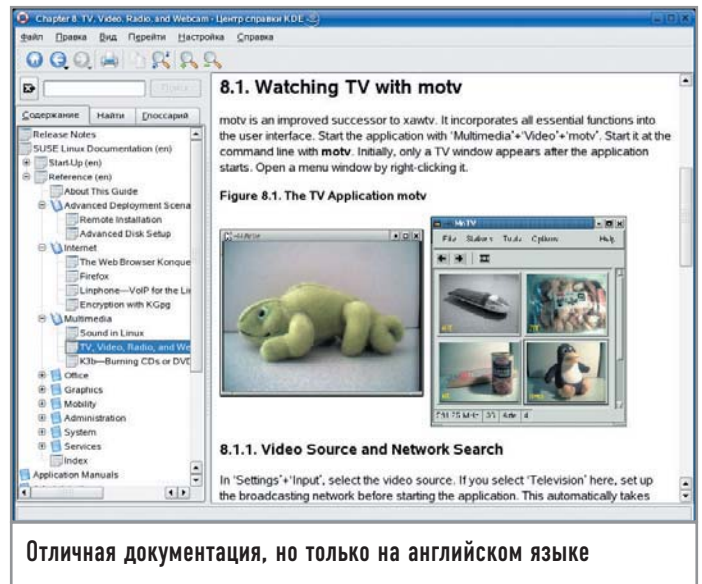
При вставке компакт-диска в CD-ROM или флеш-диска в USB-порт устройства монтируются автоматически в каталог /media, причем не важно, работаете вы в KDE, GNOME, WindowMaker или вообще в режиме командной строки: спасибо специалистам SUSE за грамотно настроенные udev и subsfs.



Кстати, выполнять `umount` для размонтирования CD/DVD и USB-носителей не обязательно. Для USB-носителей по умолчанию задается синхронный режим (`sync`). В этом режиме данные на диск записываются сразу же, а при асинхронном режиме небольшие файлы обычно сохраняются в виртуальной памяти и записываются на диск при его размонтировании. Обычно асинхронный режим имеет смысл устанавливать для дискет и флеш-дисков, то есть устройств, имеющих ограниченное количество циклов записи. Но, видимо, разработчики SUSE Linux готовы пожертвовать всем ради удобства пользователя — еще бы, его время дороже всего. Поэтому ни в коем случае не редактируйте документы в SUSE Linux прямо с USB-диска, лучше скопируйте их сначала на винчестер!

Заключение

Пора подводить итоги. Что можно сказать о новой версии SUSE? Качественный дистрибутив, не требующий абсолютно никакой доработки, все очень удобно и логично, хорошо подойдет для начинающего пользователя, знающего английский



язык. Благодаря поддержке большинства (в том числе и нестандартных) устройств это идеальный дистрибутив для установки на современный ноутбук. Почему на современный? Да потому что SUSE довольно жаден до ресурсов: комфортно работать в нем можно лишь в том случае, если на вашем компьютере установлено не менее 512 Мбайт оперативной памяти. Да, за удобство надо платить, и немало. Я же, посмотрев на свой старенький ноутбук и на то, как SUSE 10 работает на нем, вздохнув, вернулся к любимому Slackware.

В принципе SUSE можно было бы рекомендовать пользователям, переходящим с Windows на Linux, если бы не полное отсутствие русскоязычной документации и, что еще хуже, адекватно русифицированного YaST2. Надеюсь, в следующей версии ситуация изменится в лучшую сторону (может быть, именно вы переведете интерфейс YaST2). Если не обращать внимание на YaST2, то в целом русификация выполнена довольно хорошо (рекомендуйте SUSE Linux 10 противникам перехода на UTF-8).

Словом, если у вас достаточно мощный компьютер и вы цените удобство во всем — обязательно установите SUSE Linux 10, скорее всего, он станет вашей основной рабочей системой. |

Технические тонкости

Настройка «иксов»

Обратите внимание на предложенные настройки графической платы и при необходимости измените их — иногда инсталлятор ошибается, выставляя завышенные параметры. Например, для монитора Samsung SyncMaster 152s (который определился инсталлятором как SyncMaster 4s) было выбрано завышенное разрешение 1280x1024 при частоте 85 Гц, и после перезагрузки пришлось созерцать на экране надпись «Режим не поддержи-

вается». Если у вас приключится такая же беда, сделайте следующее:

- ▶ Переключитесь в текстовую консоль, нажав клавиши «Ctrl+Alt+F1».
- ▶ Введите логин (`root`) и пароль, указанные во время инсталляции.
- ▶ С помощью утилиты `gtf` определите строку параметров для поддерживаемого вашим монитором режима. В следующем примере возвращается строка «Modeline» для режи-

ма 1024x768 при частоте обновления 60 Гц:

```
$ gtf 1024 768 60
Modeline "1024x768_60.0"
64.11 1024 1080 1184 1344
768 769 772 795 -HSync
+Vsync
```

- ▶ Полученную строку впишите в файл `/etc/X11/xorg.conf` в секцию «Modes» (там уже будут строки «Modeline» для других режимов).
- ▶ В этом же файле задайте новый режим в секции «Screen». Нужную подсекцию

вам укажет значение параметра «DefaultDepth».

- ▶ Сохраните изменения в файле `xorg.conf`.
- ▶ Вернитесь в графическую консоль («Ctrl+Alt+F7») и нажмите «Ctrl+Alt+Backspace» для перезапуска X-сервера с новыми параметрами. После этого вы сможете войти в систему, введя логин и пароль созданного ранее пользователя (если не был выбран режим автоматического входа в систему).

Роман Химов

Гуманность для всех

Меньше чем за год с момента своего появления этот дистрибутив не просто смог завоевать заслуженную популярность пользователей, но и вытеснил с первого места рейтинга DistroWatch прочно до того времени обосновавшегося там Mandriva. Как ему это удалось и что в нем особенного?

Ubuntu появился на свет стараниями нашего дорогого (во всех смыслах) космонавта Марка Шаттлворта. Марк очень неравнодушен к Open Source, и, имея на руках несколько десятков миллионов долларов и желание потратить их на что-нибудь хорошее, он решил создать Ubuntu. Тем более что, по его мнению, индустрия ПО нынче имеет склонность к развитию Open Source, а значит из благотворительности может выйти неплохое дело.

Сказано — сделано, была основана компания Canonical Limited, в качестве основы взят Debian, и пошло дело. Конечно, в реальности все было не так просто, но Марку действительно удалось очень быстро развить проект и добиться признания Ubuntu как одного из интереснейших дистрибутивов.

Что же интересного предлагает Ubuntu? Регулярные релизы (раз в полгода), GNOME, живое воплощение принципа «Just works», нацеленность на пользователя, свободу и... бесплатность. Это, пожалуй, одно из самых интересных свойств, поскольку продукт не просто можно скачать с какого-нибудь FTP-сервера, но и заказать CD с дистрибутивом, который вам бесплатно доставят в любую точку земного шара. Заманчивое предложение, не так ли?

Наверное, это и стало одним из ключевых моментов, способствовавших популярности Ubuntu. Ведь, для того чтобы его опробовать, не нужен хороший/бесплатный канал для доступа в Интернет, не надо ничего скачивать/прожигать, достаточно просто оформить заказ на сайте.

Кстати, дистрибутив действительно занимает всего лишь один-единственный CD. Конечно, по подборке софта он не может тягаться с многодисковыми дистрибутивами, но со-

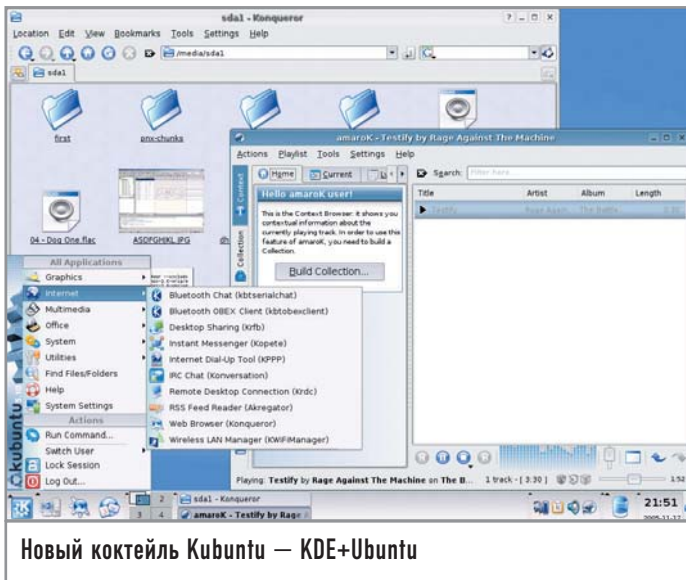
держит все необходимое для базовой работы. Правда, есть также и DVD-версия, в которой значительно больше полезных приложений. Но при наличии хорошего подключения к Сети это не играет решающей роли.

Ubuntu также гарантирует пользователям свободу (в классическом определении Free Software Foundation), что задекларировано в его манифесте, по духу напоминающем лицензию GPL. Кстати, само слово «ubuntu» пришло из одного африканского наречия и означает «гуманность», «человечность по отношению к другим». Поэтому направленность на конечного пользователя и обеспечение максимальных удобств — не просто одна из задач дистрибутива, а фактически его сущность.

| Установка |

Возможно, некоторых читателей спугнет малозначительный на самом деле факт того, что инсталлятор в Ubuntu — текстовый. То есть в действительности он псевдографический, но это уже детали. Однако, даже если вы привыкли к чему-либо в стиле Fedora Core, Mandriva или SUSE, не пугайтесь! На самом деле с текстовыми инсталляторами зачастую удается управляться еще быстрее и эффективнее, чем с графическими. Да и, если прикинуть, из более чем трех сотен существующих дистрибутивов GNU/Linux абсолютное большинство все-таки устанавливается с помощью текстовых инсталляторов. Так что сомнения прочь: это ничуть не препятствие.

Хочется отметить, что инсталлятор Ubuntu на самом деле до безобразия прост. Он даже говорит по-русски. После настройки языка, клавиатуры, сети и разделов (здесь будьте



Новый коктейль Kubuntu — KDE+Ubuntu

аккуратнее — по умолчанию Ubuntu предлагает использовать весь винчестер по своему усмотрению) он сразу переходит к копированию пакетов на винчестер. Что интересно, он же мне предложил скачать дополнительные пакеты для поддержки русского языка, которых на диске нет (единственная наклейка — это предложение было отображено на английском). С этим я согласился, однако реальное скачивание и установка этих пакетов откладываются до перезагрузки в свежую систему.

Еще мне понравился выбор часового пояса, который сразу привел к разделу «Europe/Kaliningrad», а пункт «Europe/Moscow» шел уже следующей строчкой. Мелочь, но такая догадливость инсталлятора весьма приятна. Затем создается новый пользователь.

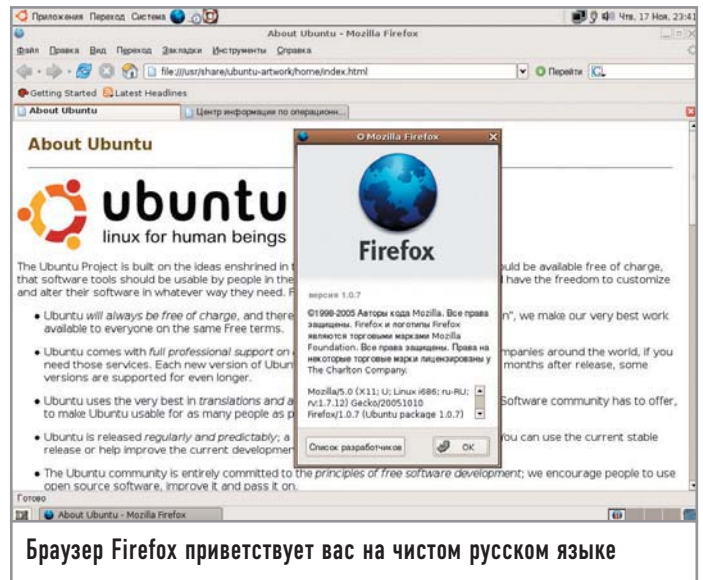
И, возможно, стоит предупредить тех, кто не собирается ставить загрузчик GRUB из Ubuntu, а хочет сконфигурировать уже имеющийся: помимо ядра /vmlinuz не забудьте прописать initrd /initrd.img.

На этом первая часть установки заканчивается, и мы подходим к первой загрузке в новую систему.

Первая загрузка

Здесь система сразу начинает ставить пакеты (на первом этапе она просто их копирует с диска), и видны мелкие недоработки: русский язык не отображается, в левом верхнем углу красуется кучка квадратиков и слово «Ubuntu». Причем это становится еще менее привлекательным, когда показывается диалог настройки разрешения экрана для X. Но все интуитивно понятно, а больше ни на какие вопросы вам отвечать не придется. Также из досадных мелочей: во время этого процесса на экран периодически вываливались сообщения ядра о разных модулях, но вреда от этого никакого нет.

Ближе к окончанию процесса установки пакетов были скачаны и установлены пакеты для поддержки русского языка. Скачивание пяти необходимых пакетов заняло менее 10 минут при скорости 256 кбит/с, то есть что-то около 20 Мбайт. Сразу после этого передо мной предстал экран логина. Разрешение — 1024x768, все работает, как и заказывали. Отмечаю это, поскольку не каждый дистрибутив способен нормально

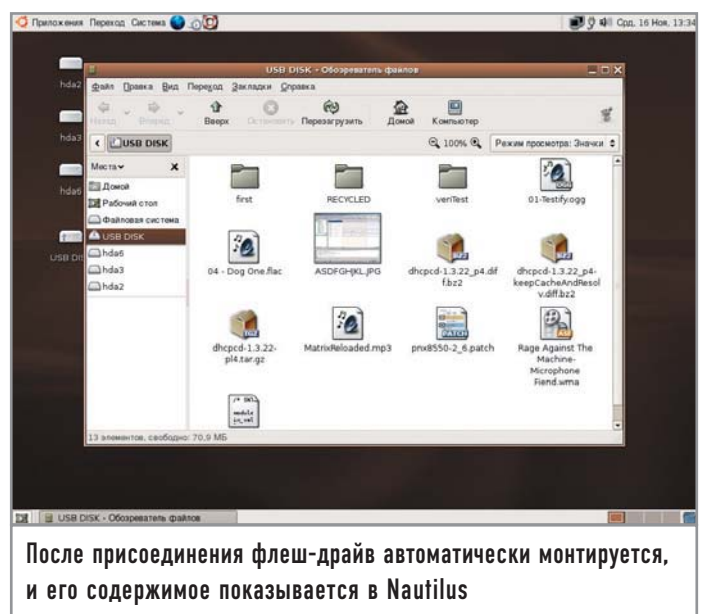


Браузер Firefox приветствует вас на чистом русском языке

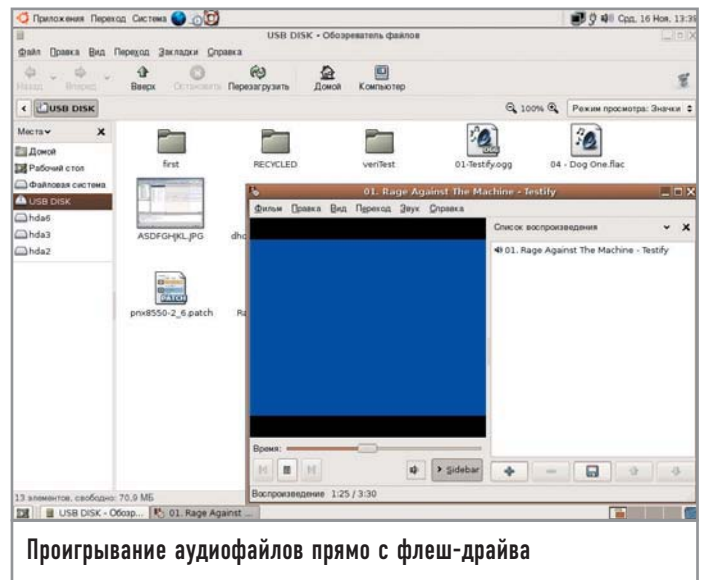
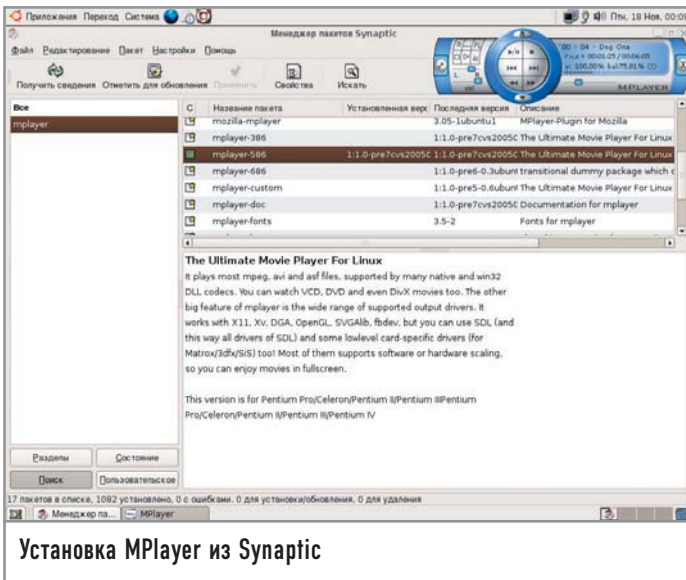
определить на моем ноутбуке видеоплату и работать с ней. Как правило, приходится копировать один и тот же проверенный временем конфигурационный файл X. Ubuntu все смог сделать сам, за что его нельзя не похвалить.

Великий и ужасный root

Если кто-то думает, что я пропустил что-то важное, он ошибается. Вначале мне тоже так показалось, поскольку никто не спрашивал пароль для root, однако на самом деле это не столь необходимо в Ubuntu. В этом дистрибутиве нет пользователя root. Но первый создаваемый пользователь может выполнять административные задачи, для которых, правда, от него потребуются ввести еще раз свой собственный пароль. Все это работает через sudo, схема очень похожа на Mac OS. Однако при желании можно получить консоль с правами root, используя команду sudo -s (или запуская графические терминалы от имени пользователя root). Можно даже завести нормального root, если вам совсем без него плохо — воспользуйтесь sudo passwd root.



После присоединения флеш-драйв автоматически монтируется, и его содержимое показывается в Nautilus



Настольная среда

Итак, заходим. Вот он, GNOME, давно не виделись. Надо отметить, что на самом деле я давно свыкся с KDE и не представляю себе без него комфортной жизни в графической среде, но тем интереснее изучать конкурирующие предложения. GNOME загрузился очень быстро и далее работал столь же бодро, что очень приятно (вообще, в плане скорости работы Ubuntu оставляет самые лучшие впечатления). Стиль оформления Ubuntu по умолчанию довольно интересный, главное его свойство — это, пожалуй, ненавязчивость. Как показала статистика, почему-то подобное оформление очень нравится девушкам.

Сразу после логина меня уведомили о доступности обновлений, и я воспользовался этим, для того чтобы немного обновить компоненты системы. Из приятного тут же обнаружилась полноценная поддержка тачпадов Synaptics, хотя по умолчанию почему-то не настроена горизонтальная прокрутка. Звук работает (используется ESD и GStreamer поверх ALSA), звуковая схема довольно оригинальная, с явными африканскими мотивами.

Медиатест

Теперь пройдемся по звуку. Вставляем USB-флеш — Ubuntu ведет себя как надо, HAL и D-Bus настроены на связку с GNOME, флеш-драйв тут же появляется на рабочем столе в виде накопителя, и открывается Nautilus, отображающий файлы. В памяти флеш-накопителя хранится специально подобранная коллекция музыкальных файлов в разных форматах. Сухой остаток — FLAC, MP3 и WMA сдались, проиграть я смог только лишь OGG. Это очень странно, поскольку плагины FLAC для GStreamer есть, и он был просто обязан проигрываться!

Совсем нехорошо получилось с DVD: Totem упрямо отказывался проигрывать их, а отличнейший диск Pink Floyd «The Wall» уже сам GNOME отказался воспринимать как видео, предложив на него что-нибудь записать. Медиатест должен быть пройден и, наверное, нам не хватает каких-то программ. Что у нас с пакетами?

Пакеты

Ubuntu, что вполне естественно, использует пакеты формата Debian, но, в отличие от его многочисленных потомков, не

стремится обеспечивать с ними бинарную совместимость. Ubuntu использует хитрую систему с несколькими репозиториями пакетов, официально поддерживается не очень большое их количество, однако в рамках сообщества представлены практически все пакеты Debian, а это более 16 000 свободных программ. По умолчанию пакетный менеджер настроен на использование именно официально поддерживаемого репозитория, и для получения всех вкусовостей на блюде с голубой каемочкой стоит сразу же включить использование репозитория пакетов, поддерживаемого сообществом (Community Maintained Packages). Где? Конечно же, в Synaptic — знакомому пользователям Debian графическому инструменту управления пакетами. Можно также пользоваться консольным apt-get, тоже известным пользователям Debian и нынче доступным в разных вариациях для управления пакетами RPM, но консоль мы оставим джигитам — Synaptic слишком удобен, чтобы его игнорировать. В нем я сразу поставил cmytsetup (а он доступен в виде пакета в репозитории сообщества), чтобы получить доступ к внешним винчестерам, а также решил попытаться счастья с totem-xine, который сразу же приказал удалить установленный по умолчанию totem-gstreamer.

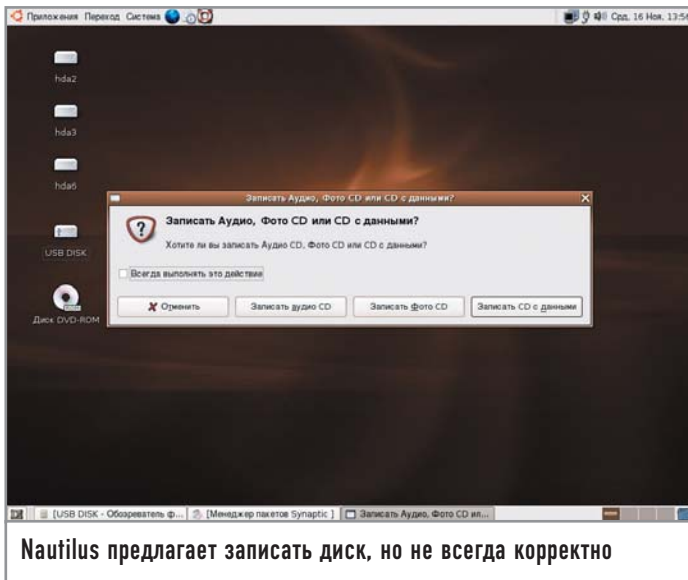
Медиатест 2

И что вы думаете? После этого Totem начал свободно воспроизводить DVD! Правда, с диском Pink Floyd проблема так и не решилась, но это уже не его вина.

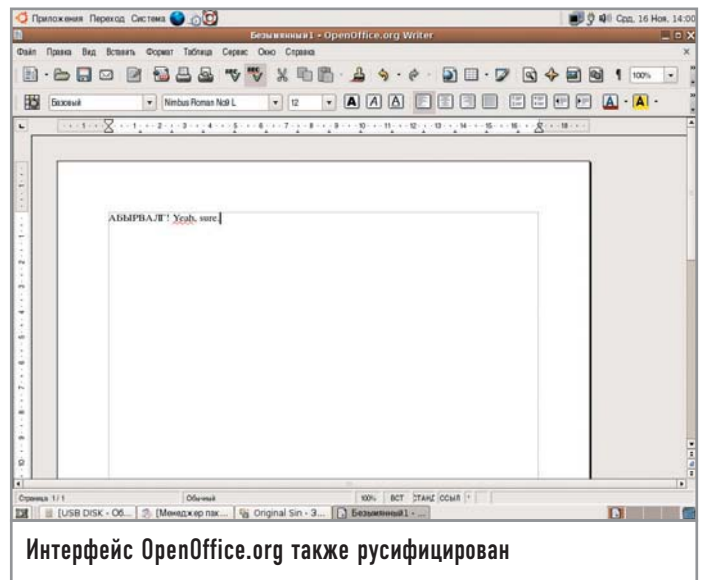
Однако как быть с музыкой? Да очень просто! Забываем про существование Totem и RhythmBox, ставим MPlayer (все доступно в Synaptic, все само скачается и установится). На этом медийные проблемы заканчиваются, поскольку MPlayer играет почти все, что существует в цифровом виде (ценой наличия нескольких несвободных бинарных библиотек).

hibernate

Увидев среди пунктов выключения компьютера возможность использования hibernate, я не мог ее не опробовать, поскольку знал, что на испытываемом железе с ним случаются проблемы. Однако мне удалось усыпить и пробудить компьютер как положено, даже



Nautilus предлагает записать диск, но не всегда корректно



Интерфейс OpenOffice.org также русифицирован

с подключенным флеш-драйвом. Suspend, он же hibernate — на сегодня все еще экспериментальная возможность, поскольку она требует четкого и слаженного взаимодействия аппаратуры и ПО. Разработчики Ubuntu смогли довести ее до рабочего состояния на дешевеньком ноутбуке, что бесспорно заслуживает уважения.

Русский язык

Это особенно приятный момент в рассказе об Ubuntu. Хотя и потребовалось скачать дополнительные пакеты, но результат того стоил: без каких-либо ухищрений в системе сразу же и полноценно начинают работать русскоязычный ввод (в «иксах» он переключается с помощью «Alt+Shift»), а также русскоязычная локализация. Система полностью (за редкими исключениями) говорит на родном языке, и это очень приятно. Да, кстати, как и все современные дистрибутивы, Ubuntu использует по умолчанию локаль UTF-8.

Семейство Ubuntu

Мастер перевоплощений

На самом деле Ubuntu — не только цельный дистрибутив, но и семейство дистрибутивов, использующих его в качестве основы. Первым стал Kubuntu, как можно догадаться, предлагающий в качестве основной настольной среды KDE. Затем поднялся Edubuntu, а теперь и ImpiLinux (дистрибутив с прицелом на африканских пользователей) также начал использовать Ubuntu в качестве основы. Аналогично и MoLinux, только он разрабатывается в Испании. Отличия между дистрибутивами заключаются в поставляемых

на дисках пакетах и некоторых настройках по умолчанию. Лично мне, конечно, особенно приятно отметить Kubuntu. Если вы привыкли работать в KDE, возможно, стоит обратить внимание именно на него. Мною был опробован LiveCD, и, несмотря на некоторые шероховатости, в целом у меня остались от него приятные впечатления. Edubuntu содержит дополнительные пакеты, касающиеся образовательного применения (в частности, быстрой организации учебного класса), и это довольно специфичная вещь.

Приложения

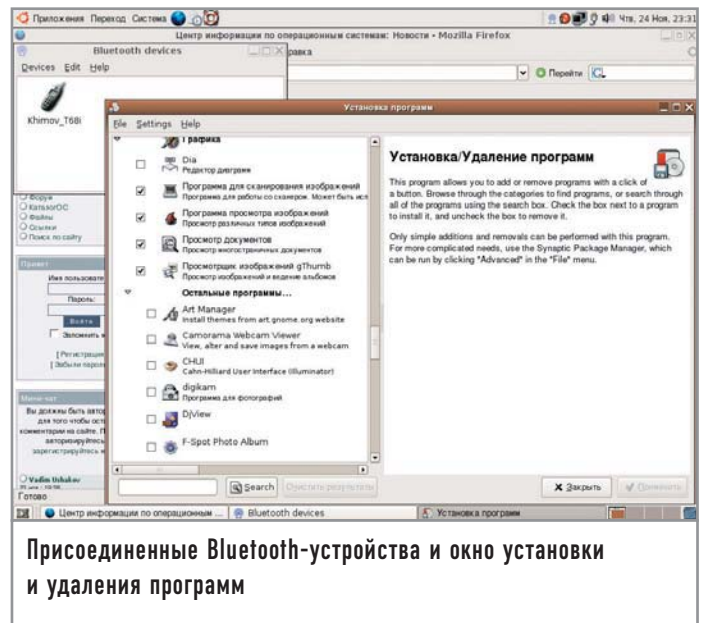
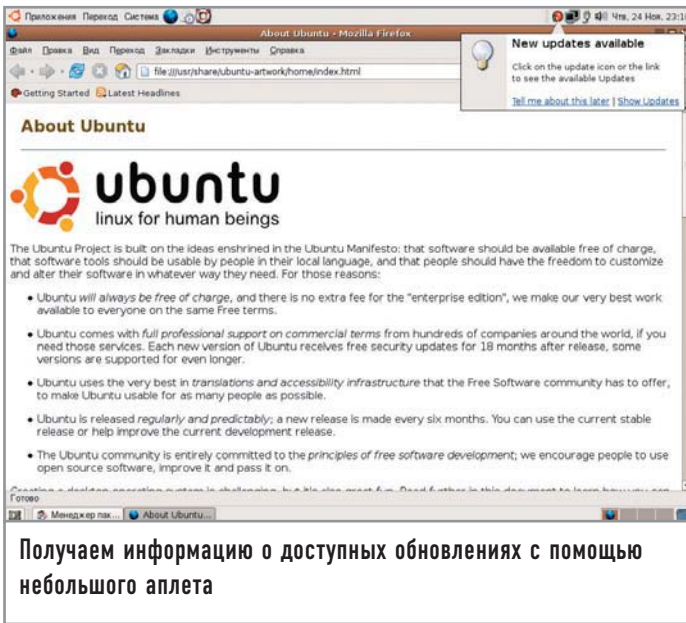
Исходя из основной рабочей среды в виде GNOME, приложения в базовом диске также подобраны соответствующие: Firefox в качестве основного браузера, Evolution в качестве почтового клиента, Gaim в качестве IM-клиента, графический редактор GIMP и т. д.

Естественное, в отличие от родителя (Debian), здесь используются самые свежие версии всех программ. Обычно это приветствуется для домашнего/настольного применения, но в случае серверов у вас могут быть свои соображения насчет стабильности. Хотя, какими бы ни были эти соображения, поддержка дистрибутива в течение 18 месяцев вам обеспечена. В том числе и коммерческая.

Кстати, если вы думаете, что Synaptic — единственное графическое средство управления пакетами в Ubuntu, то ошибаетесь. В меню GNOME Ubuntu есть простой пункт «Add applications» (недоработка со стороны переводчиков). Воспользуйтесь им, и вы увидите альтернативу Synaptic, которая не будет морочить вам голову пакетами, а покажет определенные приложения на своих конкретных местах в меню GNOME. Какие-то из них уже есть, а какие-то можно установить, просто поставив галочку (соответственно, для удаления галочку надо снять). Все зависимости, все необходимое для выбранного приложения установится автоматически!

Стоит сказать и о менее позитивных вещах. Некоторые пользователи и обозреватели отмечают нестабильность поведения Nautilus, который попросту любит падать (а совсем злые языки утверждают, что это общее свойство GNOME). Но за все время тестирования ничего подобного не произошло (хотя в Debian такое случалось не раз). Может быть, это связано с установленными обновлениями, в любом случае к стабильности работы Ubuntu нареканий не возникло.

В целом же рабочая среда в Ubuntu приятная и, это стоит отметить еще раз, очень быстрая. Если что-то и стоит подправить, так это шрифты: моноширинный шрифт по умолчанию выглядит не очень адекватно, для латиницы он смотрится нормально, но для кириллицы уже совсем плохо. Но это легко изменить в разделе «Система → Параметры → Шрифты». В остальном, не-



смотря на отсутствие разнообразия настроек, надо признать, что GNOME в Ubuntu вполне пригоден к использованию и чрезвычайно прост в освоении.

Разработчику

Ubuntu, как и полагается полноценному дистрибутиву GNU/Linux, имеет в своем составе все необходимое для разработчиков. Правда, на инсталляционный CD всего не уместить, поэтому некоторые инструменты придется скачивать отдельно.

Однако инструменты разработчика — также одна из интересных тем в разговоре об Ubuntu, поскольку Canonical спонсирует разработку системы контроля версий Bazaar, а Марк Шаттлворт заявляет о необходимости более тесного взаимодействия разработчиков и сообществ. Поэтому Ubuntu работает также над улучшением эффективности взаимодействия создателей свободного ПО. Помимо систем контроля версий это затрагивает баг-трекеры и сервисы для переводчиков.

Уже хотя бы поэтому (а также благодаря наличию свежайшего набора инструментов) разработчику может быть интересно начать знакомство с Ubuntu, но и это не все, что он может предложить. Как насчет вознаграждений за работу над свободным ПО? Да, в Ubuntu есть и такое: выбирайте работу по вкусу, это хороший способ попрактиковаться и познакомиться как с самим софтом, так и с его разработчиками.

Сообщество

На вопрос «Чем же так хорош Ubuntu?» многие иностранные пользователи закатывают глаза и с придыханием произносят: «О, да это же community!» И это действительно так, Ubuntu сумел собрать вокруг себя крепкое сообщество пользователей.

Иногда кажется, что, даже если бы Ubuntu мало чем отличался от других дистрибутивов с технической точки зрения, его все равно можно было бы четко определить по сообществу. Ubuntu просто окружает дух взаимопомощи и общения: к вашим услугам форумы, блоги разработчиков, почтовые рассылки и конференции, IRC, wiki-сайты, различные ресурсы любителей Ubuntu и многое другое.

Изучение структуры и взаимодействия сообщества, и просто наблюдение за его работой и развитием гипнотизирует! Не проникнуться этим невозможно, не влюбиться после этого в Ubuntu очень сложно. И только одно может остановить вас на пути знакомства с этим своеобразным миром — языковой барьер. Однако есть и наше, российское, сообщество пользователей (www.ubuntu.ru) — не столь многочисленное, но тоже вполне дружелюбное и имеющее в активе достаточно обширный и хороший форум, где решаются в том числе и переводческие проблемы. Есть также русскоязычный IRC-канал и почтовая конференция.

Развитие

Ubuntu развивается очень стремительно, следующий релиз выйдет уже в апреле! Ожидается, что в нем значительно улучшится поддержка Bluetooth (в GNOME действительно маловато соответствующих утилит), произойдет очень много изменений в области работы с устройствами на раннем этапе загрузки системы, а также будет произведено множество других доработок.

Но особое внимание стоит обратить на релиз 6.10, который должен выйти в октябре 2006 года. Марк Шаттлворт уже пообещал дать достойный ответ выходу Windows Vista (которая появится позднее). И, судя по сумасшедшему взлету популярности Ubuntu, это не просто слова.

Резюме

Ubuntu — интереснейший дистрибутив и еще более интересное сообщество разработчиков и пользователей. В его активе прекрасная поддержка аппаратуры и качественная локализация (но, напомним, требующая немного интернет-трафика), дружное сообщество и четкое видение собственного развития. Ubuntu идеально подходит на роль первого дистрибутива для неподготовленного пользователя, но может с успехом использоваться и специалистом (особенно если он имел дело с Debian). В любом случае, если вы находитесь в поиске нового/первого дистрибутива, обратите на него (или Kubuntu, или Edubuntu — по вкусу) внимание — может быть, это именно то, что вам нужно. |

Дармовой суперкомпьютер

Знаете ли вы, какой процесс из запущенных сейчас на вашем компьютере самый красивый? А самый вежливый? Вопрос неожиданный. Но именно так — «красивый», «вежливый» — можно перевести термин `nice`, которым обозначается показатель приоритета запущенного процесса в Linux. На моем, а также на компьютерах сотен тысяч людей по всему миру «самыми вежливыми» процессами — имеющими наибольшее значение `nice` (а значит самый низкий приоритет) — являются программы распределенных вычислений, использующие все свободное время вверенного им компьютера во благо науки.

Большую часть времени своего пребывания во включенном состоянии среднестатистический современный компьютер просто простаивает, ожидая ввода данных пользователем. Для многогигагерцевых процессоров время между нажатиями клавиш при печати, пока пользователь читает с экрана какой-нибудь текст или ненадолго отошел от компьютера, — целая вечность.

Но, к счастью, такая громадная вычислительная мощь — миллиарды операций в секунду, помноженные на сотни миллионов компьютеров — недолго оставалась невостребованной. В наши дни каждый желающий занять свободное время своего компьютера чем-нибудь полезным вполне может сделать это, присоединившись к одному из проектов распределенных вычислений. Сами распределенные вычисления (далее РВ) — это способ выполнения каких-либо расчетов путем их разделения между множеством компьютеров.

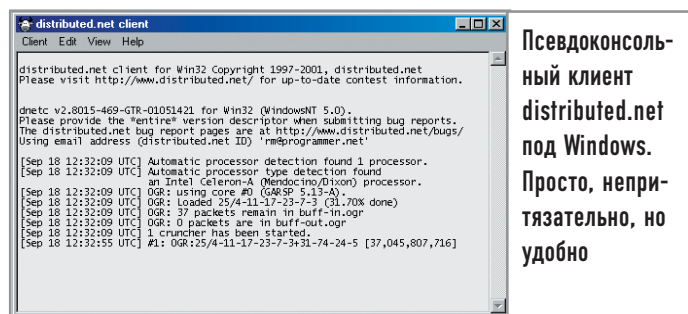
В этой статье речь пойдет о проектах РВ, работающих через Интернет и задействующих компьютеры обычных пользователей (с их ведома, разумеется) для выполнения разного рода сложных научных расчетов. Участвуя в подобных проектах, можно помочь исследовать строение белков и генетических структур, найти лекарство от рака и СПИДа или даже поспособствовать в поиске сигналов внеземных цивилизаций.

Обычной работе (или игре) на компьютере это абсолютно не мешает — используются только моменты бездействия.

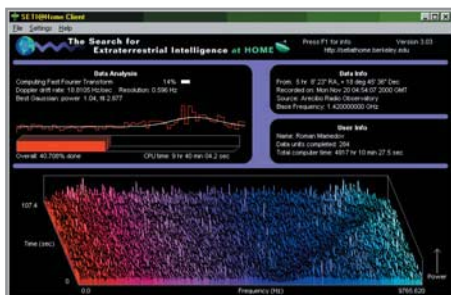
И, конечно же, говорить мы будем только о тех проектах, участие в которых доступно пользователям Linux.

С чего все началось

Можно сказать, что история массовых распределенных вычислений через Интернет началась с проекта `distributed.net`. Самый популярный из его подпроектов — `RC5` — занимается взломом зашифрованных одноименным криптоалгоритмом сообщений и обязан своим существованием компании `RSA Data Security`, которая время от времени с целью продемонстрировать надежность разрабатываемых ею алгоритмов шифрования устраивает публичные кон-



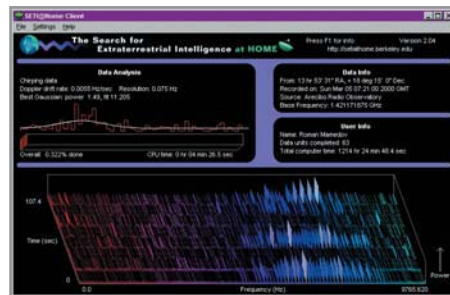
Псевдоконсольный клиент `distributed.net` под Windows. Просто, неприятно, но удобно



В поисках подобных аномалий на графике фанаты проекта просиживали за мониторами долгие часы



В данном случае мы засекли всего лишь понижение чувствительности приемника на опорной частоте 1420 МГц



А здесь — помехи от интерференции с наземными источниками или спутниками. Хотя, кто знает...

курсы по взлому зашифрованных ими сообщений с призами в несколько тысяч долларов. Участники проекта distributed.net уже выигрывали такие конкурсы, успешно подобрав ключ к зашифрованным алгоритмами RC5-56 и RC5-64 сообщениям. В данное время идет соревнование по взлому RC5-72, однако, по некоторым оценкам, из-за возросшей по сравнению с прежними проектами длины ключа получения результата, даже с учетом постоянного роста доступных проекту вычислительных мощностей, придется ждать несколько тысяч лет.

Хотя distributed.net существует очень давно, пределом его мечтаний до сих пор является внимание энтузиастов, либо интересующихся распределенными вычислениями как таковыми, либо криптографией и математикой. А первым проектом, получившим по-настоящему широкую известность и популярность среди обычных пользователей, стал, несомненно, SETI@home. Его руководители предложили всем желающим принять участие в поиске сигналов внеземной цивилизации. Принцип работы прост: радиотелескоп обсерватории в Аресибо постоянно записывает космический шум, ленты с ним пересылаются в «штаб-квартиру» SETI@home в Беркли, где специальные серверы (сплиттеры) нарезают данные на мелкие кусочки и рассылают компьютерам участников для обработки.

Однако высокая популярность — в проекте зарегистрировано более 4 млн ПК — обернулась для SETI@home и головной болью. Дело в том, что рост доступных проекту вычислительных мощностей шел гораздо быстрее, чем увеличение поступающих с телескопа данных для анализа и способностей сплиттеров по их пере-

работке. Поэтому, чтобы не отказывать в регистрации или блоках данных новым участникам, проекту пришлось начать рассылать одну и ту же, уже обработанную, информацию по много раз. По сведениям независимого сайта статистики (www.goving-mouse.com/setiathome), избыточность «классической» версии SETI@home достигала фантастических величин в 50–100 крат.

Куда идем?

Способ использования «лишних» вычислительных мощностей, заключающийся в повторном обсчете информации либо в искусственном и не приносящем особой пользы усложнении обработки одних и тех же данных, не понравился никому, включая самих авторов SETI@home. Гораздо логичнее было бы, если бы проект, у которого на момент обращения компьютера участника за новой порцией данных обсчитывать было бы особо нечего, смог вместо повторной отправки «старой», уже обработанной, порции честно заявить: «Работы сейчас нет, зайдите позже». А клиентская программа участника смогла бы этот ответ понять и переключиться на помощь какому-нибудь другому проекту — тому, которому дополнительные ресурсы в этот конкретный момент времени действительно необходимы. Данный подход и был реализован авторами SETI@home в их новой платформе для распределенных вычислений, получившей название BOINC.

Технические тонкости

Необходимые файлы

Самым большим среди россыпи закачиваемых проектом Rosetta@home файлов является собственно расчетный модуль Rosetta — около 7 Мбайт. Очевидно, что его загрузка потребует лишь при первоначальном подключении к проекту, ну и при выходе его более новых версий. Вторым по размеру является 6-мегабайтный файл с загадочным названием `bbdep02.May.sortlib.gz`, кото-

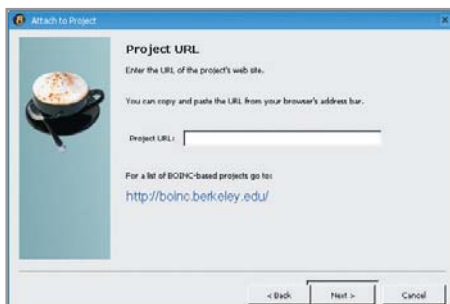
рый не обновлялся аж с мая месяца (возможно, даже 2002 года) и не собирается, будем надеяться, слишком часто обновляться и в дальнейшем. Такое поведение довольно характерно для распределенных проектов: скачанные при первоначальном подключении файлы в дальнейшем подвергаются повторному использованию настолько часто, насколько это является возможным.

Детали

Ингибитор для вычислений

Помимо рассылки одних и тех же данных для борьбы с «избытком» вычислительной мощности существует и другой способ. Руководители проекта SETI@home внесли в третью версию клиентского ПО новые, гораздо более тщательные и медленные алгоритмы анализа блоков данных. Но, по мнению неко-

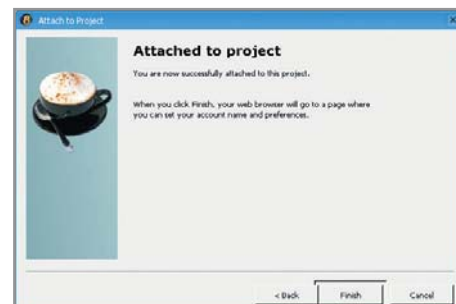
торых, с точки зрения возможности ими что-то обнаружить эти алгоритмы являлись довольно-таки бесполезными. При этом рост вычислительной мощности продолжался, и вскоре проект достиг показателей избыточности, многим уступающим тем, что существовали еще до выхода «замедленной» версии.



Чтобы присоединиться к вычислениям, прежде всего необходимо указать URL главной страницы проекта



Зарегистрировать новый аккаунт теперь можно и из самого клиента — ранее это приходилось делать через сайт



Компьютер готов трудиться во благо науки!

BOINC

| <http://boinc.berkeley.edu> |

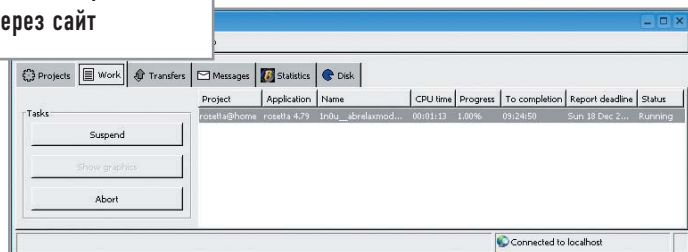
Berkeley Open Infrastructure for Internet Computing (BOINC) — это весьма успешная и уже очень популярная как среди авторов, так и среди участников вычислительная платформа, задача которой — унификация общих по функциональности модулей клиентского и серверного ПО различных проектов. Разработчикам распределенных научных проектов, ученым — специалистам в области биологии, медицины, криптографии и прочих наук — она пришлась по вкусу благодаря тому, что при ее использовании им больше не нужно было разбираться с такими скучными и бесконечно далекими от области их исследований вещами как организация системы регистрации пользователей, разработка системы статистики, отладка взаимодействия клиентского и серверного ПО и т. д. Все уже готово — придумано, написано и отлажено авторами BOINC. Остается брать и пользоваться. Предлагается использовать любое ПО BOINC — от клиентского модуля до серверных PHP-скриптов — в исходниках, под лицензией GNU GPL, а разобраться во всем этом поможет доступная на сайте BOINC подробнейшая документация по созданию на основе этой платформы собственного проекта PB с нуля.

Rosetta@home

| <http://boinc.bakerlab.org/rosetta> |

Итак, пора перейти от теории к практике. Рассмотрим процесс присоединения к работающему на платформе BOINC проекту на примере запущенного недавно Rosetta@home.

Это научное исследование, проводимое одной из лабораторий Вашингтонского университета, результатами которого должны стать новые методы (алгоритмы), помогающие с высокой точностью заниматься моделированием, предсказанием и дизайном белков. Наличие проверенных алгоритмов моделирования поможет исследователям разрабатывать лекарства для лечения различных болезней, вызываемых нарушениями структуры белка. Своим названием проект обязан лежащему в основе его клиентского ПО модулю Rosetta, который был изначально создан для предсказания трехмерной структуры белка из его аминокислотной последовательности. Со временем в Rosetta появились функции дизайна и предсказания белковых комплексов. Авторы проекта надеются продолжить



Окно BOINC Manager содержит информацию о назначенных данному компьютеру заданиях

постепенно улучшать используемые в Rosetta физическую модель и алгоритмы поиска. А помочь им в этом должны результаты исследований Rosetta@home.

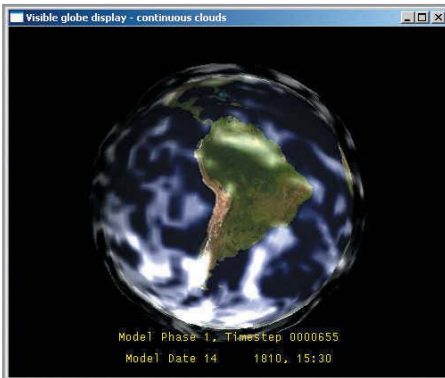
Для участия в проекте нам прежде всего понадобится базовый клиент BOINC. Скачиваем с сайта <http://boinc.berkeley.edu> файл с самой свежей версией — на момент написания статьи это boinc_5.2.7_i686-pc-linux-gnu размером 3,4 Мбайт — и запускаем. Дистрибутив создает в текущем каталоге подкаталог BOINC, куда и распаковывается все свое содержимое, после чего сообщает, что для запуска консольной версии BOINC следует использовать скрипт `run_client`, а для запуска версии с графическим интерфейсом — `run_manager`. Кроме того, он прописывает в этих скриптах полный путь к созданному подкаталогу, так что при необходимости куда-либо его переместить нужно это учесть и соответствующим образом их подправить. Загружаем графическую версию BOINC. Поскольку этот первый за-

Технические тонкости

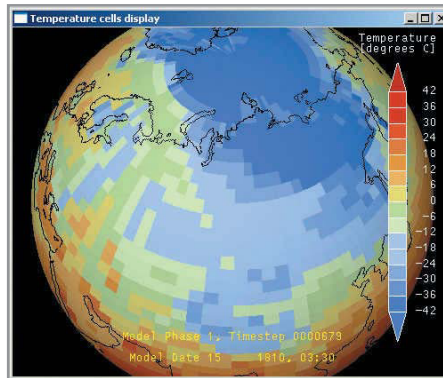
Особенности авторизации

Стоит отметить, что схема авторизации по e-mail и паролю для BOINC относительно нова. Ранее в основанных на нем проектах использовалась авторизация по секретному Account Key, выступавшему одновременно в роли и уникального идентификатора, и пароля на изменение информации. Кое-где такая схема используется и по сей день (в этом случае клиент BOINC

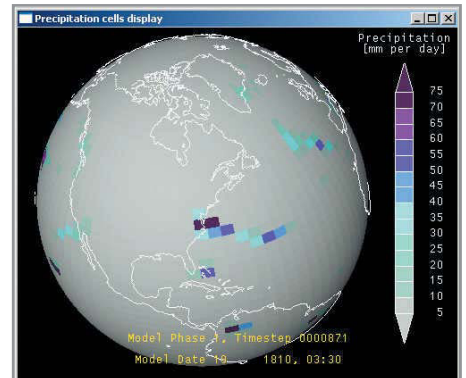
корректно ее обнаруживает, и вместо поля с вводом пароля появляется поле для Account Key). Существенным ее недостатком является отсутствие возможности зарегистрировать новый аккаунт прямо из клиента: необходимо зайти на сайт проекта, зарегистрироваться там, дождаться электронного письма с заветным Account Key и только после этого подключаться к проекту.



Climate Prediction показывает облачность...



...температуру...



...и влажность

пуск, нам будет предложено указать адрес главной страницы проекта, к которому мы желаем присоединиться.

В следующем окне мастера нужно либо указать e-mail и пароль к уже созданному в проекте аккаунту, либо зарегистрировать новый (в этом случае пароль придется ввести дважды).

После создания нового аккаунта BOINC предложит задать его параметры, открыв в браузере раздел настроек на сайте проекта. Впрочем, сделать это можно и позже, просто зайдя на сайте проекта в раздел «Your Account» и написав выбранные при регистрации адрес почты и пароль. Там же указывается выделяемое под проекты BOINC место на диске, устанавливается режим работы клиента (допустим, можно сделать, чтобы он работал только в определенные часы), настраивается отображаемая графика (если она есть) и т. д. Во всех проектах BOINC есть не только «личный зачет», в котором участники выстраиваются в порядке количества набранных ими за обработку блоков данных очков, но и командный, где соревнуются образованные ими команды. Поэтому там же, в разделе настроек аккаунта, есть возможность либо присоединиться к одной из команд (например, к «Russia», объединяющей российских участников проекта), либо создать свою.

Вернемся к подключаемому нами проекту. Сразу после присоединения можно открыть закладку «Transfers» и увидеть, что BOINC уже вовсю занялся закачкой модулей, необходимых для участия в Rosetta@home и занимающих около 17 Мбайт. Важно отметить, что подобный объем трафика довольно-таки не типичен для проектов PB. Обычно совокупный размер всех скачиваемых даже во время первоначального присоединения к какому-либо проекту файлов в разы меньше этого значения, а если он посвящен криптографии или математике, вовсе практически равен нулю.

На закладке «Work» уже отображается процесс обработки текущего блока данных (пока нулевой — состояние «Downloading»), затраченное на его обработку время и так называемый дедлайн, то есть финальная дата, до которой нужно обязательно посчитать этот блок и вернуть результат, иначе его отправят на обработку кому-нибудь другому.

К сожалению, каких-либо возможностей визуально оценить процесс моделирования (например, своими глазами увидеть создаваемую модель белка) Rosetta@home в данное время не предоставляет. Относится это не только к данному проекту, но и к большинству других, имеющих версии под Linux. Среди опробованных мною BOINC-проектов графику под Linux на текущий момент показывает только Einstein@home. Однако в скором времени ситуация наверняка изменится: последняя новость на сайте Rosetta@home сообщает о начавшемся тестировании новой версии скринсейвера (то есть версии расчетного модуля с полноценной графической частью, наверное, и под Linux тоже). От других проектов тоже логично ожидать выпуска в скором будущем Linux-версий с поддержкой визуализации расчетов, ведь в большинстве случаев она написана под OpenGL, который портируется между платформами довольно просто.

Но что-то мы отвлеклись. Возвращаемся к клиенту BOINC и видим, что закачка расчетных модулей Rosetta уже завершилась и проект начал расчеты. Теперь можно свернуть окно BOINC Manager и заняться своими делами. Rosetta, равно как и любой другой современный проект распределенных вычислений, другим запущенным на компьютере программам никогда не мешает. Чтобы удостовериться, можно открыть тор и увидеть, что процесс rosetta_4.79, хотя и занимает 98% CPU, при этом имеет самый низкий из возможных приоритетов, а значит, как только процессор становится нужен любой другой программе, Rosetta тут же в нужном объеме его освобождает.

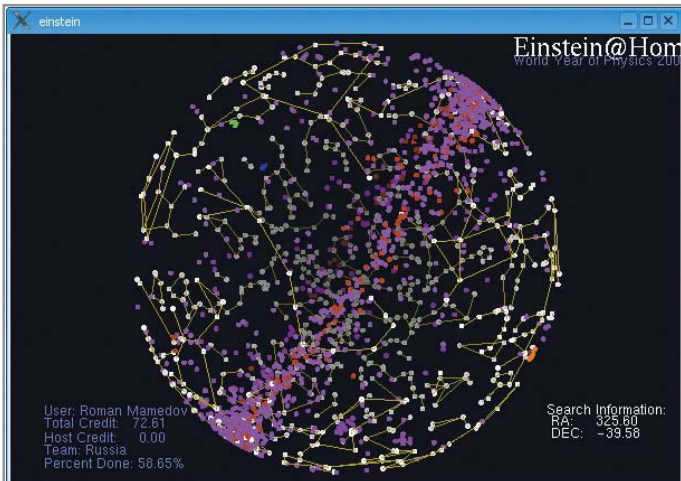
Чтобы воспользоваться основным преимуществом BOINC — динамическим переключением между проектами, — нужно

Визуализация в Linux

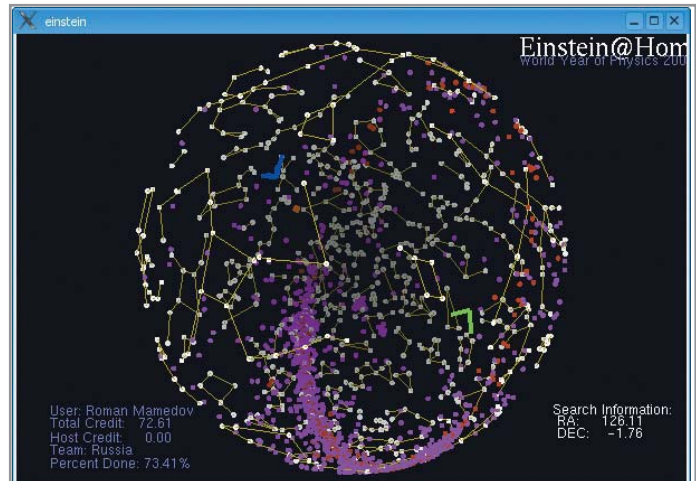
Временные недостатки

К сожалению, Linux-версии многих расчетных модулей основанных на BOINC проектов не предоставляют пока никакой возможности визуально оценить процесс расчета (покрутить модель молекулы, узнать погоду на виртуальном глобусе Земли, полюбоваться на пики графика с космическим шумом). В общем случае Windows-версия проекта имеет красивую визуализацию, а в Linux кнопка BOINC-менеджера «Show graphics» отключена. Это от-

носится к LHC@home, Climate Prediction и SETI@home и связано не только и, наверное, даже не столько с полным отсутствием внимания к Linux-версиям со стороны авторов, сколько с совсем недавним переходом BOINC со старой, четвертой, на новую — пятую версию платформы. По сообщениям участников, в четвертой версии BOINC графику показывали гораздо больше проектов, а в пятой кое-где ее нет даже под Windows.



Точки — это звезды, а полоса поперек сферы — Млечный путь



Видны два из трех маркеров, символизирующих позиции зенитов для собирающих информацию приборов

присоединиться еще хотя бы к одному проекту, а лучше к нескольким. Не знаю, кому как, но, по-моему, участвовать одновременно в нескольких проектах гораздо интереснее, чем в каком-то одном, тем более что ничего сложного в этом нет: процесс подключения у всех идентичен описанному на примере Rosetta@home.

| Climate Prediction |

| www.climateprediction.net |

Проект Climate Prediction, как следует из его названия, занимается моделированием и предсказанием климата на Земле. Упрощенно можно сказать, что результатом работы проекта должен стать «прогноз погоды» на 50 лет вперед. Его руководители хотят определить, насколько точны существующие методы долгосрочного предсказания погоды и насколько сильно на их точность влияют погрешности в исходных данных.

Работающий на платформе BOINC вычислительный модуль проекта занимается моделированием долгосрочных изменений климата Земли. Все участвующие компьютеры получают немного разные входные данные, поэтому итог работы каждого из них будет уникальным. Руководители проекта выберут среди полученных результатов наиболее точно отражающие происшедшее до 2000 года и сделают вывод о наиболее вероятных изменениях, ожидающих нас в будущем.

Этот проект является, несомненно, рекордсменом по количеству скачиваемых при подключении данных. Сразу после присоединения BOINC-закладка «Transfers» проинфор-

Цитата

Дэйв Фрейм о Climate Prediction

«Мы хотим исследовать небольшие изменения в общей модели, — говорит координатор проекта, доктор Дэйв Фрейм из Оксфордского университета. — Единственный способ сделать это — провести очень много эксперимен-

тов, проверить все возможные варианты. То есть сделать то, что заняло бы у суперкомпьютера несколько тысяч лет. Но, решая это как распределенную задачу, мы сможем сделать все гораздо быстрее.»

мировала меня о том, что проект climateprediction.net заказал получение файлов общим объемом аж в 44 Мбайт! Поэтому, если вам трафик дорог как память, возможно, стоит от участия в этом проекте отказаться. Справедливости ради стоит сказать, что единожды закачанный блок данных «Climate Prediction» будет считаться очень долго — как минимум месяц, и это на довольно быстром компьютере, причем работающем в круглосуточном режиме. А при расчете последующих блоков скачивать большую часть уже полученных данных повторно не требуется.

Скриншоты приводятся из Windows-версии этого клиента; Linux-графика, когда она появится (по слухам, для некоторых типов расчетных блоков она уже доступна), вряд ли будет сильно отличаться.

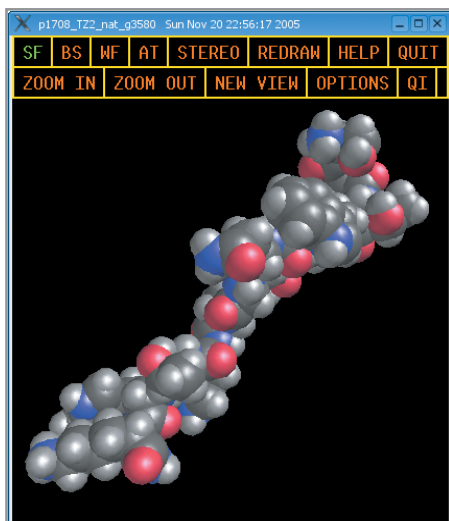
| Einstein@home |

| <http://einstein.phys.uwm.edu> |

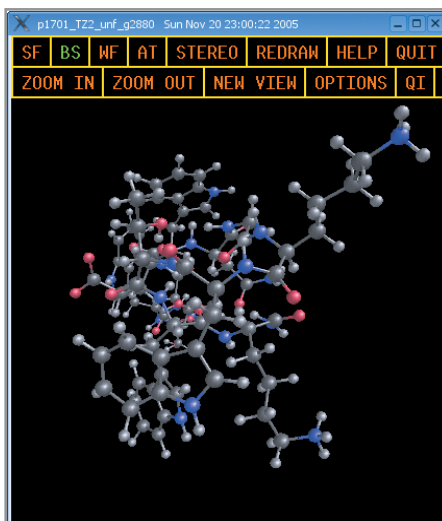
Проект Einstein@home занимается составлением для всего неба атласа излучаемых звездами-пульсарами гравитационных полей. Такой атлас должен помочь в проверке гипотезы Эйнштейна, предсказывающей теоретическую возможность существования гравитационных волн, возникающих при столкновениях черных дыр и взрывах звезд. Пока никому из ученых обнаружить эти волны не удалось.

Данные для анализа будут поступать из Лазерной интерферометрической гравитационно-волновой обсерватории (LIGO). Рабочая часть задействованной в проекте обсерватории LIGO состоит из двух точнейшим образом установленных зеркал и путешествующего между ними луча лазера. При наличии волн в гравитационном поле путь лазера чуть-чуть отклоняется, что позволяет ученым замечать и измерять эти волны. Большее расстояние между зеркалами означает большую чувствительность интерферометра, поэтому зеркала LIGO установлены расстояния четырех километров друг от друга. Это позволяет фиксировать изменения положения луча лазера вплоть до одной стомиллионной диаметра атома водорода.

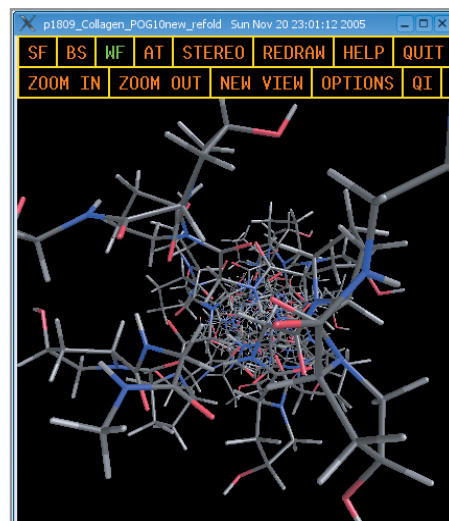
Этот проект, как упоминалось выше, на сегодняшний момент является единственным, имеющим полноценное отоб-



Внешнюю структуру белка проще всего оценить в режиме fpd по умолчанию



Второй режим позволяет лучше рассмотреть его внутренние связи



Исследуемые структуры часто бывают весьма сложными

ражение визуализации расчетов как под Windows, так и под Linux. Русский перевод описания того, что означают различные элементы этой визуализации, можно найти по адресу <http://distributed.ru/?einstein.starsphere>.

| Folding@home | | <http://folding.stanford.edu> |

Можно сколько угодно говорить о текущем глобальном переходе на BOINC, однако существуют как участники, так и разработчики PB, которым BOINC по разным причинам либо не нравится, либо просто не приносит, по их мнению, никакой пользы по сравнению с «традиционной» архитектурой проекта. Особенно в том случае, если последняя уже давно запущена и успешно функционирует.

Среди таких проектов наиболее примечательным является Folding@home. Его разработчики сразу же после появления BOINC заявили, что никаких преимуществ от перехода на эту платформу они не видят, а когда увидят — просто создадут дополнительную BOINC-версию своего клиента, оставив наравне с ней и первоначальную. Кстати, на данный момент BOINC-версия клиента этого проекта все-таки уже существует и проходит бета-тестирование.

Цель Folding@home — получение более точного представления о болезнях, вызываемых дефектными белками. Изучаются белки, имеющие отношение к болезни Альцгеймера, Паркинсона, диабету типа II, коровьему бешенству и рассеянному склерозу. Поняв, почему возникают дефекты в белках одного типа, ученые смогут выяснить, каким образом это происходит и с другими белками. Все желающие имеют свободный доступ к результатам. На сайте выкладываются все публикации, ролики смоделированных процессов и т. п.

Клиент состоит из двух частей — оболочки и расчетного ядра. Некоторые из расчетных ядер оптимизированы под расширенные наборы инструкций — SSE и 3DNow! либо SSE2. Но руководители проекта на этом останавливаться не собираются. Исследуется возможность создания версии клиента, использующей для расчетов графический процессор видео-

платы, и даже разрабатывается версия, задействующая специальную плату-ускоритель PCI.

На сайте <http://folding.stanford.edu> под Linux предлагается только консольная, без всякой графики, версия клиента. Однако для Folding@home существует несколько очень красивых сторонних программ для визуализации расчетов (работающих в том числе и под Linux), например fpd (<http://home.comcast.net/~wxdude2/rph/fah.html>).

| Другие проекты |

К сожалению, в рамках статьи невозможно описать и половины всех работающих на BOINC проектов, не говоря уже о гораздо более многочисленных, имеющих «традиционную» архитектуру.

Если тема распределенных вычислений вас заинтересовала, но описанного здесь показалось недостаточно, советую посетить официальный сайт BOINC (<http://boinc.berkeley.edu>), на котором перечислены все проекты, использующие в работе эту платформу. Например, стоит зайти на сайт <http://distributed.ru>, где можно найти русскоязычные описания наиболее популярных проектов, новости о них и довольно активный форум; и, конечно же, монструозный ресурс <http://distributedcomputing.info>, где есть краткие описания, наверное, всех проектов — функционирующих сейчас, когда-либо существовавших, а также еще только планируемых к запуску.

В общем, присоединяйтесь! |

Сущность Rosetta

О расчетном ядре

Ядро Rosetta — это физическая модель макромолекулярных взаимодействий и алгоритмов поиска структуры с наименьшей энергией для заданной аминокислотной последовательности (при предсказании структуры белка),

комплекса «белок-белок» (при расчете стыковки белков), а также поиска аминокислотной последовательности с наименьшей энергией для заданной белковой структуры или комплекса «белок-белок» (при дизайне белков).

Косметология и показ мод

Приятно и полезно идти в ногу с техническим прогрессом, иметь хотя бы беглое представление о том, что творится в мире хай-тека. Сегодня многие уже работают в Интернете, используя ADSL-подключение, а огромное количество семей сохраняют на память торжественные события с помощью цифрового фотоаппарата.

Ценность цифрового фотоаппарата резко возрастает, если уметь грамотно распоряжаться полученными снимками. Какие основные нужды у простых фотолюбителей?

Первым этапом всегда является собственно процесс съемки. Условия не всегда могут быть подходящими, да и не все мы фотографы от бога.

Полученные фотографии желательно сохранять на компьютере, чтобы вдоволь налюбоваться полученным материалом. Но не всегда снимки радуют глаз, поэтому очень часто приходится прибегать к инструментам, улучшающим их качество. По мере накопления изображений на жестком диске все больше возникает необходимость грамотной их сортировки с целью дальнейшего быстрого поиска любимых кадров.

Красивыми фотографиями всегда хочется поделиться с друзьями, а самые прекрасные экземпляры так и просятся напоказ всему свету.

Мы с вами работаем в Linux — системе, изначально предназначенной совсем для других, не связанных с цифровым фото, нужд. Но прелесть открытого программного обеспечения заключается в том, что разработчикам интересно писать наиболее востребованные пользователями программы. Свято место пусто не бывает, поэтому все запросы обладателя цифрового фотоаппарата реализованы и в открытой ОС.

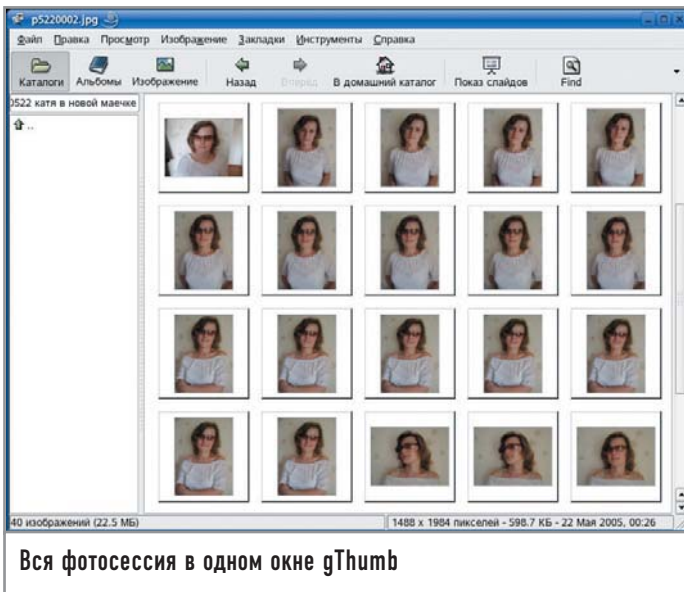
| Первая стыковка |

Ядро Linux имеет драйверы практически ко всем устройствам с USB-интерфейсом. Поддерживается стандарт USB 2.0. При

подсоединении фотоаппарата к ПК с помощью шнура с разъемом mini-USB камера автоматически монтируется в определенный каталог — свой для каждого дистрибутива. После этого фотографии можно переписать с помощью любого файлового менеджера. Исключение составляют модели фирмы Canon. Для работы с камерами в Windows требуется установка специального программного обеспечения. На дисках, поставляемых вместе с продукцией Canon, программного обеспечения для Linux нет. Однако все проблемы решает приобретение недорогого карт-ридера, который определяется в Linux как USB-устройство и автоматически монтируется.

| В розовых очках |

В популярных дистрибутивах Linux можно найти немало программ для просмотра изображений, например Kuickshow из KDE или gThumb из GNOME. Многие из них имеют в своем составе средства для каталогизации изображений на жестком диске. Программы могут создавать свои базы данных, которые расширяют возможности навигации по базе изображений и добавляют некоторые сервисные функции. Однако подобный подход к проблеме имеет отрицательные стороны. Авторы каждой программы по-своему видят пути, упрощающие нам жизнь, и, создав базу в одной программе, вы не сможете работать с ней в другой. Стандартизация — малознакомое слово для авторов баз данных. Единственным «стандартом» остается грамотное хранение файлов на жестком диске. Если не лениться и вписывать в названия папок, например, даты съемок и те-



Вся фотосессия в одном окне gThumb

мы текущих коллекций, то в дальнейшем найти нужные фотографии будет не так уж и сложно.

Цифровой мейкап

Практически ни одна утилита для просмотра не имеет даже базовых инструментов редактирования изображений. Единственная «гляделка», в которой можно выполнять простейшие манипуляции, улучшающие качество изображения, — это gThumb. Инструмент «Enhance» пытается автоматически подобрать оптимальные параметры яркости/контрастности и цветового баланса. С помощью остальных функций вы можете попробовать вручную поиграть с инструментарием, используемым на начальном этапе обработки цифровых изображений.

Широкий простор открывается для творческой мысли, если обратиться к графическому редактору GIMP. Как ни крути, а unix way

Технические детали

Мегапиксели vs фотобумага

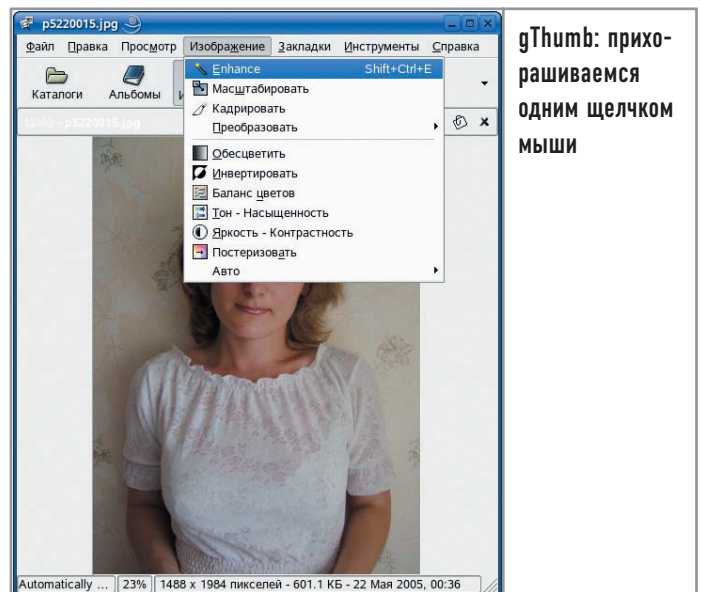
При печати бумажных фотографий используется разрешение 300 dpi. Несложные

расчеты (не менее 120 пикселей на 1 см) показывают следующую картину:

Количество мегапикселей фотоаппарата	Максимальный размер фотографии без потери качества, получаемый при печати, см
2	10x15
3	13x18
4	13x18 с небольшим запасом
5	15x20
7	Формат А4
9	Формат А4 с небольшим запасом

Как видите, рекламные заявления о том, что, например, 7-мегапиксельными камерами можно делать снимки размера А3, — не более чем пустой звук. Кро-

ме того, не стоит забывать, что наши фотоаппараты (99% — «мыльницы», 1% — «зеркалки») делают снимки с качеством ниже допустимого предела.



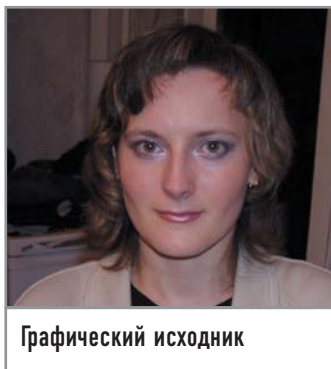
gThumb: прихорашиваемся одним щелчком мыши

и в данном вопросе дает о себе знать: лучше использовать несколько специализированных программ, чем одну универсальную.

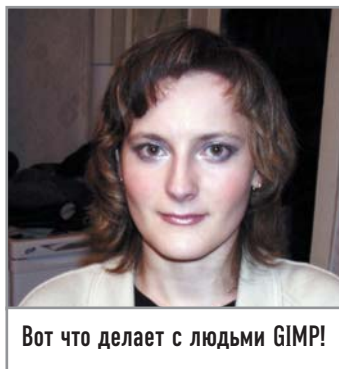
GIMP — единственный серьезный программный продукт для манипуляции с изображениями в Linux, однако в этом нет ничего страшного. GIMP — мощный и в то же время довольно простой в освоении инструмент. С другой стороны, стоит отметить, что отсутствие в GIMP некоторых инструментов «для чайников» заставляет выполнять некоторые простейшие операции через последовательность других функций. Например, если в ACDSsee и Photoshop красные зрачки глаз убираются практически одним щелчком мыши, то в GIMP необходимо вручную выделять область зрачка и выключать красный цветовой канал в этой области.

Но не будем говорить о грустном, лучше сразу перейдем к работе с нашими фотографиями. При первом взгляде на фото сразу бросаются в глаза зернистость и нарушение цветового баланса, если при съемке использовался параметр ISO 400 и выше. Для устранения шумов и зернистости используется специальный фильтр «Гауссово размывание», который вызывается из меню «Фильтры» и подменю «Размывание». При этом немного понижается общая четкость изображения, но запас в этой области для печати 10x15 см у современных фотоаппаратов просто огромен. Вы также можете сначала выделить часть изображения с помощью группы инструментов выделения (произвольное, связанных областей и по цвету) и лишь затем понизить уровень шума. При таком подходе можно «разглядеть» кожу на портрете, сохранив при этом максимальную четкость изображения на участках волос и одежды.

Частичное нарушение цветового баланса исправляется с помощью функции «Цветовой баланс», вызываемой из меню «Инструменты» и подменю «Инструменты цвета». Обратите внимание, что с помощью GIMP вы можете менять цветовую гамму отдельно в трех диапазонах яркости — тенях, полутонах и бликах. Очень часто при съемках на улице в солнечный день небо получается не голубым, а белым из-за неточной передачи яркости на предельных значениях. С помощью точной коррекции цвета вы можете добавить голубой оттенок тому интервалу яркости, в котором оказалось небо. Точно так же можно делать воду



Графический исходник



Вот что делает с людьми GIMP!

синей в грязных водоемах, убирать красноту на лицах со снимков, сделанных дешевыми аппаратами Canon со вспышкой.

Если ваш снимок получился засвеченным, в некоторых местах яркость такая, что все сливается в единый белый фон, то это уже ничем не исправить. Слишком темные снимки, напротив, поддаются исправлению. Для корректировки яркости снимков воспользуйтесь мощным инструментом «Уровни» («Инструменты → Инструменты цвета»), с помощью которого вы сможете не только установить нужные параметры яркости для снимка, но и окончательно выправить цветовой баланс.

Обратите внимание на три пипетки, расположенные справа от кнопки «Авто». Выбор черной и белой точек позволяет скорректировать изображение так, чтобы равномерно задействовать на снимке весь диапазон яркости. Выбор серой точки позволяет равномерно построить всю цветовую гамму без перекосов в сторону того или иного канала. Если вас не устраивает работа автоматики, добро пожаловать в ручную правку графиков. Вы можете плавно регулировать пороговые значения каналов и общей яркости с помощью черных и белых треугольников, а также корректировать гамму с помощью серого ползунка.

На женских портретах очень красиво смотрится эффект мягкого света, который можно создавать с помощью одноименного фильтра, вызываемого из подменю «Имитация». В реальной жизни подобный эффект можно получить, если перед вспышкой поставить листок глянцевого картона, который отразит свет на потолок. В результате модель не будет засвечена прямым ярким светом, а попадет под таинственную власть рассеянного отражения. При этом исчезнут резкие тени, зрачки глаз никогда не получатся красными, композиция приобретет объем и некий шарм.

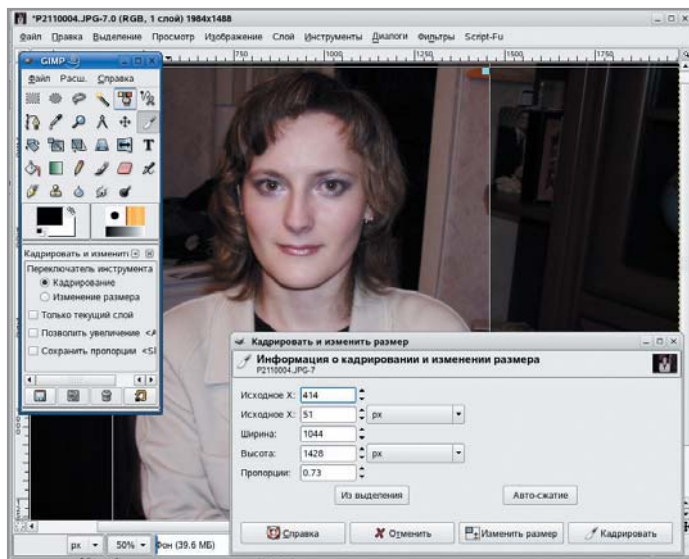
И, наконец, окончательные штрихи наносятся на снимок с помощью инструмента «Тон-насыщенность» («Инструменты →

Любопытные факты

Матрица в голове

Известно, что строение глаза человека имеет одну особенность — мы видим четкое изображение по центру, но даже небольшое отклонение взгляда резко ухудшает восприятие мира на-

шей сетчаткой. Ученые подсчитали среднее «разрешение» сетчатки глаза. Зрение здорового человека в среднем эквивалентно 4-мегапиксельной камере.



Мы простоотрежем ненужное

Инструменты цвета»). Играя насыщенностью отдельных цветов, можно подчеркивать конкретные детали. Например, уменьшение насыщенности желтого и зеленого цветов делает крону деревьев полностью обесцвеченной. В результате можно получить цветной портрет на черно-белом фоне.

Помимо общих улучшений качества снимков вы можете подчеркнуть отдельные детали. Ведь GIMP — это не просто менеджер изображений, а полноценный графический редактор.

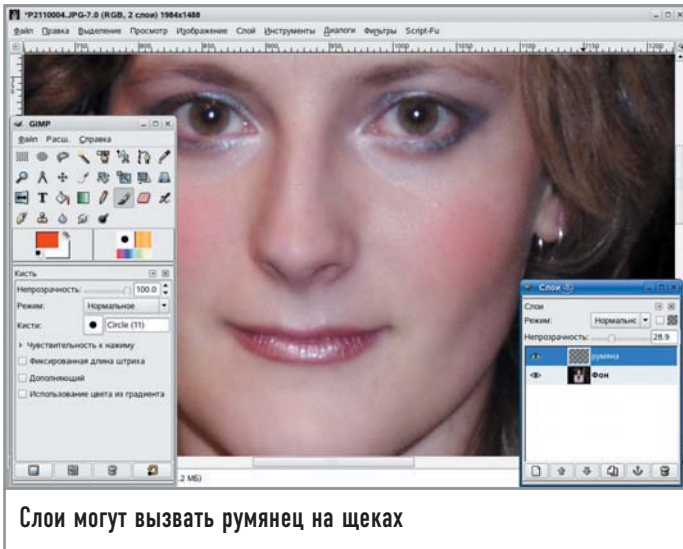
Вы можете, например, наносить произвольный макияж на женские портреты. GIMP поддерживает слои. Создайте новый слой. Нанесите на щеки румяна, вокруг глаз тени. Это делается с помощью обычного карандаша. Теперь фильтром «Гауссово размывание» очень сильно размажьте слой. Вместо резких карандашных штрихов должны получиться цветные пятна. Играя с прозрачностью слоя, добейтесь нужного эффекта. Важно, чтобы косметика не была слишком яркой, но была заметна. Брови и ресницы рисуются черным карандашом, тонкими линиями и размазываются не очень сильно. Немного сложнее нанести помаду на губы. Для этого цветным карандашом обведите контур губ, немного размажьте результат, добавьте прозрачность по вкусу. После этого выделите контур губ и внутри выделения увеличьте цветовую насыщенность. Можно также изменить тон. Даже невзрачная девушка становится писаной красавицей, если над ней поработать в GIMP.

Веб-дефиле

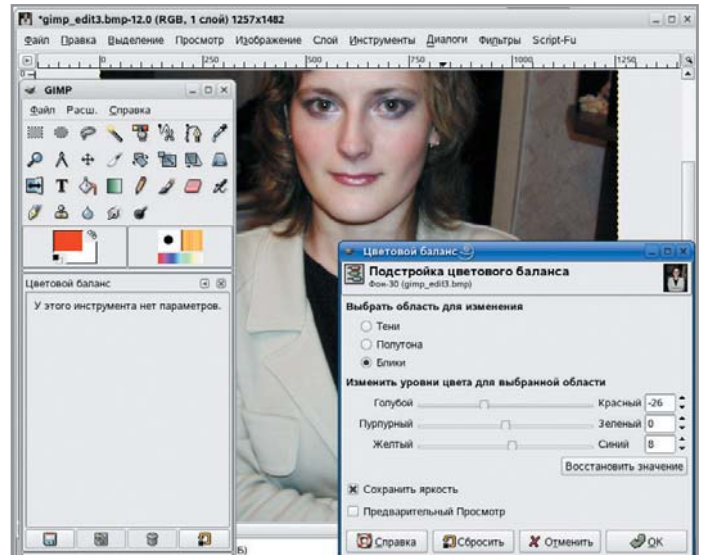
Рано или поздно возникает желание показать фотографии друзьям, а также выставить некоторые свои работы на всеобщее обозрение. Самый простой способ поделиться фотографией с другом — это послать ее по электронной почте.

Но если требуется показать снимки всему миру, а их количество неприлично велико, то электронная переписка оказывается бессильна под напором ваших требований.

Более эффективным способом обмена фотографиями является создание галерей в Интернете. Однако на этом тернистом пути поджидает немало проблем. Перед вами встает выбор: либо размещать свои фотографии, используя специальные сете-



Слои могут вызвать румянец на щеках



Цвета любят «Баланс»

вые сервисы, либо создавать галереи самостоятельно. Сайты, предлагающие услуги по размещению галерей, накладывают большие ограничения. Во-первых, максимальный размер файла с изображением не может превышать фиксированного значения, которое ставит в тупик даже плоды труда старых 2-мегапиксельных камер. Принудительное сжатие файлов также не дает желаемого результата, так как резко снижается качество изображений. Во-вторых, на многих сайтах устанавливается ограничение на размер изображения в пикселях, что ставит крест на обмене качественными фотографиями. Огромное количество рекламы на подобных сайтах подобно пороховому заряду сжигает трафик, за который вы платите свои кровные деньги. А картинки ведь не отключить (это ведь самый действенный способ снижения трафика) — вот в чем и юмор, и трагедия ситуации!

С другой стороны, многим пользователям и не нужно большего. Не требующая особых усилий возможность создания галерей из небольших фотографий устраивает многих. Тем более веб-сервисы позволяют вам создавать небольшие описания к своим фотографиям, а другие пользователи сервиса могут оставлять свои комментарии напротив ваших фото. Примеры популярных сервисов: gallery.aquareal.ru, www.photosight.ru, foto.mail.ru и другие.

Но нам, пользователям Linux, не привыкать решать вопросы самостоятельно и достигать целей своим трудом. Одна из наиболее простых программ для создания галерей — это gThumb. Выделите фотографии, из которых хотите создать галерею, а затем в меню «Инструменты» выберите пункт «Создать интернет-альбом». Вы можете произвольно сортировать изображения, менять стилевое оформление будущего альбома. Когда альбом будет создан, его нужно разместить в Сети.

Для размещения собственных галерей (и любой другой информации) в Сети нужен хостинг. Компании, предоставляющие бесплатный хостинг, накладывают ограничения на максимальный объем файла и общий объем будущего альбома. Списки наиболее популярных компаний в этой области можно найти по адресу internethosting.ru.

Но если вы собираетесь поделиться с миром очень большой коллекцией качественных фотографий, то в этом случае

не избежать финансовых затрат. Использование платного хостинга подобно таблетке аспирина снимает вечную головную боль, мучающую вас из-за необходимости постоянно идти на компромисс. Не стоит пугаться слова «платный» после воспоминаний о каком-нибудь баннере, где значится цифра порядка \$100 в месяц.

Довольно большое количество провайдеров бесплатно предоставляет собственные серверы для хостинга клиентов. На это стоит обратить внимание в первую очередь. Если такой услуги у вашего провайдера нет, наберите в поисковой строке Google запрос «платный хостинг» и убедитесь в том, что за \$2-3 в месяц можно получить практически не ограниченную по объему галерею в Интернете.

Доступ к ресурсам хостинга осуществляется по протоколу FTP. Наберите в Ksuser (или другой имеющейся у вас программе) адрес вашего хостинга и, словно у себя на локальном диске, клавишей «F5» выгрузите вашу галерею в Интернет.

Мир цифрового фото разнообразен и крайне интересен. Процесс съемки на цифровую камеру, обработка снимков на компьютере, публикация в Сети — все это становится доступным широкому кругу пользователей. Мировое сообщество разработчиков открытого программного обеспечения, к счастью, не упускает из вида этот факт, предоставляя нам с вами удобный набор инструментов в рамках популярной операционной системы Linux. |

Альтернатива

Спасательный круг

А как же быть людям, подключающимся к Интернету с помощью dialup-модемов? Если выйти за рамки передачи информации через Интернет, то можно найти и другие способы обмена фотографиями. Создайте гале-

рею в gThumb размером в несколько гигабайт, запишите ее на DVD с помощью k3b и пошлите по почте бандеролью. И это будет намного быстрее и дешевле, чем загружать фотографии, используя канал 56 кбит/с.

Радио будущего?

Даже тем, кто знаком с английским языком, сложно с первого раза понять, что же означает новый термин — podcasting. С одной стороны, слог «pod» напоминает об аудиоплеерах производства компании Apple — iPod. С другой стороны, слог «casting» наверняка имеет что-то общее с вещанием — broadcasting. Что же связывает между собой Apple iPod и вещание?

Чтобы не ломать понапрасну голову над термином podcasting, обратимся к онлайн-библиотеке Wikipedia [1]. И вот что выясняется: «Podcasting — это метод распространения аудио- и видеопрограмм через Интернет, позволяющий пользователям подписываться на определенные файлы (известные также как потоки — «feeds»), а затем прослушивать и просматривать скачанный материал в любое удобное время. Поток обычно представляет собой звуковой файл в формате MP3».

Вот так, сухо и незатейливо, написано в энциклопедии. Но что же на самом деле является подкастингом и чем это понятие качественно отличается от того, что предоставляют потоковые интернет-радиостанции?

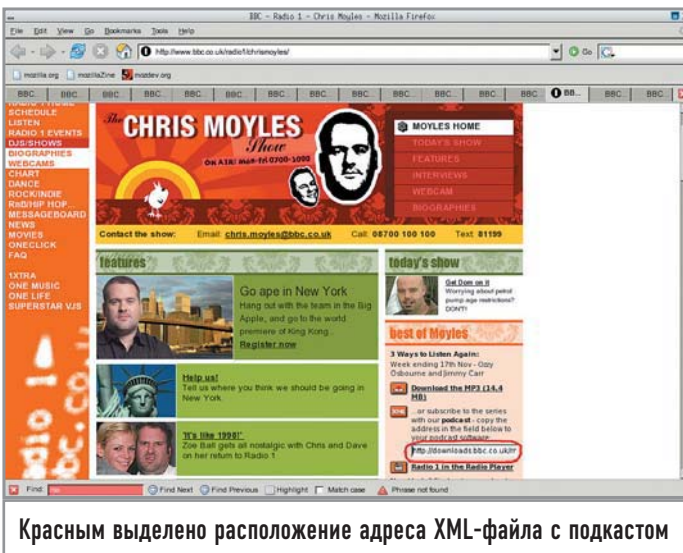
Подкастинг — это запись аудиофайлов и размещение их на каком-либо веб-сервере, с которого пользователь забирает RSS-рассылки [2]. Таким образом, как только файл попадает на сервер, он автоматически появляется в поле зрения пользователя посредством RSS-агрегации. В первую очередь это удобно самому пользователю — представим, например, что вещание опреде-

ленной передачи какой-либо интернет-радиостанции ведется в ночное время (просто расположена она на другом конце света). Можно оставить компьютер включенным на ночь и записать нужную трансляцию. А можно просто выделить эту передачу — ссылка будет передана программе загрузки, и через некоторое время эта передача окажется у вас в руках (естественно, она должна быть записана в виде отдельного звукового файла, а у вас должно быть установлено программное обеспечение, управляющее RSS-подпиской).

Что касается содержимого передач, здесь уже всю проявится фантазия тех, кто составляет звуковой ряд, а составить его может любой человек, у которого под рукой есть микрофон, программа для записи звука и соединение с Сетью. В принципе, обычного цифрового диктофона тоже вполне достаточно. Главное — передать созданный файл на какой-либо подкаст-сервер, и публикация готова. Здесь уже ни о какой фильтрации и цензуре говорить не приходится: если вам интересно слушать — вы слушаете. Если неинтересно, то никто вас не заставляет скачивать тонны мегабайт ненужных вам данных.

Кто же слушает подкасты? В общем-то, это те пользователи, которые создают свои страницы в Live Journal, различного рода блогах и ведут, скажем так, «дневники». И, конечно же, это радиостанции, точнее ведущие радиостанций. Если к блогерам все уже привыкли и ничего принципиально нового в этой сфере уже нет, то подкасты — явление сравнительно новое.

Впрочем, сам термин podcasting появился достаточно недавно, широко о нем начали говорить в 2004 году. Родоначальник этого термина, а также отец-основатель принципа подкаст-вещания — Адам Карри. Когда-то он был ведущим MTV, а сейчас ударился в сферу подкастинга. Название отчасти связано с тем, что у Карри был плеер iPod, на который можно было загружать файлы и слушать их в любое удобное время. Хотя, чтобы слушать подкаст-вещание, вовсе необязательно иметь iPod, как, впрочем, и какой-либо другой MP3-плеер. Обыкновенной ПЭВМ со звуковой платой вполне достаточно.



Красным выделено расположение адреса XML-файла с подкастом



Виртуальные радиоприемники

Для получения информации о подкаст-листах необходимо сделать так, чтобы плеер мог получать RSS-рассылки. На данный момент имеются следующие клиенты, которые могут работать с подкаст-записями:

- ▶ amaroK — приложение для воспроизведения аудиофайлов [7];
- ▶ BPCoNF — подкаст-загрузчик для KDE;
- ▶ gPodder — еще один загрузчик подкаст-рассылок, написанный на Python и GTK [8];
- ▶ Juice (ранее носивший имя iPodder) — самый первый известный подкаст-загрузчик [9];
- ▶ jPodder — также одна из самых первых программ для получения подкаст-информации (написана на Java) [10].

Я решил остановиться именно на jPodder благодаря его Java-происхождению, а также отчасти из-за того, что на момент написания этой статьи он был доступен для платформ Windows и Mac, а для GNU/Linux находился в стадии переработки кода.

Клиент jPodder-1.0RC2 достаточно крупный, около 10 Мбайт, и представляет собой инсталляционное приложение (пусть вас не смущает расширение .JAR). Следует только указать директорию, в которую его следует установить, и дело в шляпе.

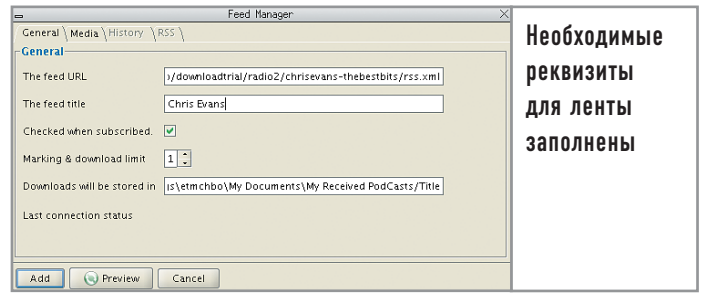
Первым делом я поспешил на сайт BBC и решил послушать подкасты двух ведущих — Криса Моэlsa и Криса Эванса.

Добавление новой ленты происходит достаточно просто. На вкладке «Personal Feeds» щелкаете правой кнопкой мышки и во всплывающем окне выбираете «Add Feed».

Далее заполняете небольшую форму с указанием адреса XML-файла подкаста и названия ленты. На этом подписка завершена, и можно приступить к загрузке подкаста.

Следует отметить, что в подкастах, помимо ID3-информации о тегах MP3-файла, может храниться и графический логотип. Например, пока загружается подкаст-интервью с Оззи Осборном, вы можете посмотреть, с каким качеством оцифрован данный поток.

Несколько слов о jPodder. Несмотря на то что клиент работает под Java, я не испытывал каких-либо проблем, связанных с быстродействием программы. Единственный недостаток — нельзя выбрать медиаплеер, так как они выполнены в качестве расширений и могут вызывать либо iTunes, либо Windows Media Player, которые под Linux пока не портированы. Впрочем, авторы уверяют, что написать расширение не представляет трудностей, было бы у Linux-программистов желание (авторы являются Win32-разработчиками). Однако объективных причин не прослушать звуковые файлы в отдельном приложении — XMMS, MPlayer или любом другом — нет.



Сам себе радиоведущий

С помощью чего получать RSS-ленты, вы теперь знаете. А что требуется для того, чтобы создать свой собственный подкаст? Будем считать, что из аппаратных решений у вас только ПЭВМ и микрофон. А также операционная система и желание что-либо записать — будь то мысли или сказка на ночь.

Понадобится пакет Audacity (для избавления от ненужных шумов и нормализации уровня звука), а также lame (для оцифровки). Как правило, Audacity уже поставляется с поддержкой библиотеки libmp3, поэтому сохранять файл MP3 можно непосредственно из программы. Следует помнить, что одна минута в формате MP3 joint-stereo имеет объем примерно 1 Мбайт.

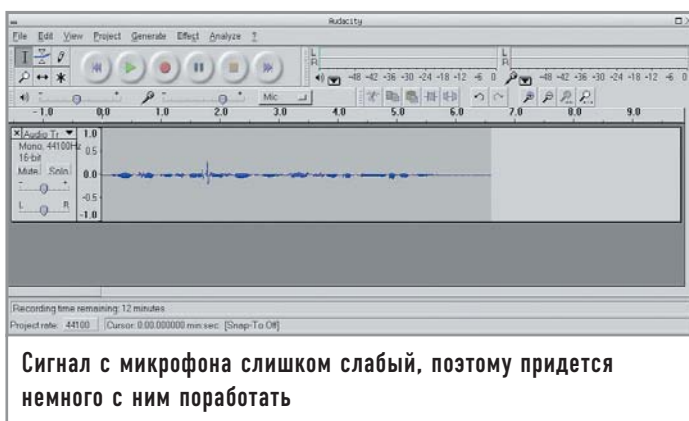
Сказать, что Audacity — сложная для понимания программа, нельзя. Несколько кнопок — «Старт», «Стоп», «Пауза», «Запись» — вот практически и весь необходимый ассортимент функций для начинающего подкастера. Способы снижения шума и удаления ненужных фрагментов постигаются почти мгновенно. Сохраняем файл в формате MP3, и половина дела сделана. Осталось оформить теги для RSS-ленты и сохранить файл на сервере, принимающем подкасты.

Информация для RSS-лент хранится в формате XML, поэтому предлагаю посмотреть, как грамотно составлять файл:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<rss version="2.0">
<channel>
<title>Site's Title</title>
<link>http://www.server.ru/podcasting/</link>
<description>All about your russian podcast</description>
<language>ru-RU</language>
```



Подкаст содержит также графический логотип

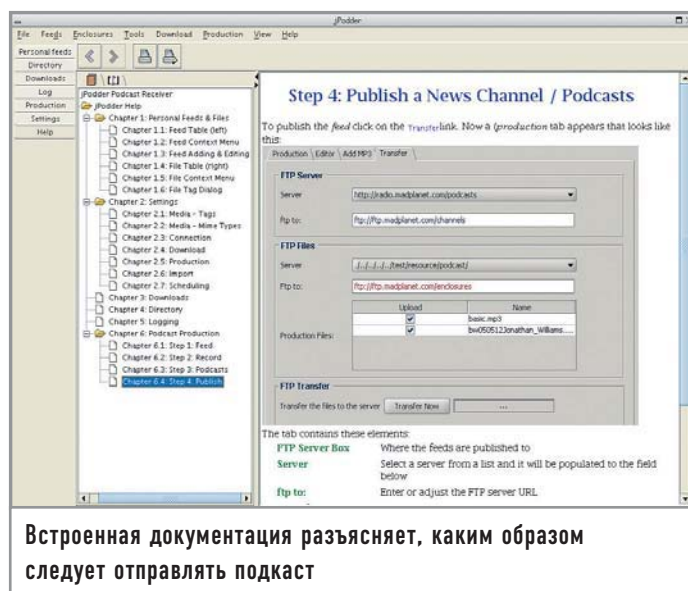


```
<lastBuildDate>Fri, 01 Jun 2005 08:00:00 +0000</lastBuildDate>
<pubDate>Fri, 01 Jun 2005 08:00:00 +0000</pubDate>
<item>
  <title>Podcast's title.</title>
  <description>Notes, ID3 tags and other information.</description>
  <enclosure url="http://www.server.ru/audiofile.mp3"
length="12345678" type="audio/mpeg"/>
</item>
</channel>
</rss>
```

Тег `<?xml version="1.0" encoding="iso-8859-1" ?>` объявляет об использовании XML-стандарта версии 1.0 в кодировке ISO-8859. Почему именно в этой, а не в русской? Мне кажется более разумным использовать на данном этапе именно транслитерацию русских слов и названий, нежели составлять описания в кодировке KOI8-R. Во-первых, нет гарантий, что абсолютно все RSS-агрегаторы поймут наши кириллические тексты. А во-вторых, описание подкаста на английском языке позволит, например, китайскому радиослушателю спокойно слушать «Kalinka Song», а не пытаться разобрать русские «иероглифы».

Тег `<rss version="2.0" ?>` начинает описание RSS-ленты. Тег `</rss>`, соответственно, заканчивает описание разметки. В общем, если вы работали с HTML-разметкой, то практически все элементы в примере будут вам понятны.

В тегах `<title>` и `<link>` описываются название ресурса (то есть сервера) и его адрес. А вот непосредственное указание подкаста происходит чуть ниже — для этого служит тег `<item>`. В нем также присутствуют теги `<title>` и `<description>`, которые чуть выше представляли описание сервера. И важное добавление — тег `<enclosure>`, в котором хранится точный ад-



рес подкаста. Информацию о подкасте также можно продублировать и в тегах самого MP3-файла [13].

На этом пока все. Далее вы отправляете ваш подкаст на радиостанцию или ресурс, принимающий подкасты, и радуетесь жизни, то есть следите за статистикой скачиваний.

Никто не обязывает вас сохранять подкаст именно в формате MP3. Вы можете сохранить его и как RealAudio (RM), Apple Audio (MOV), OGG Vorbis (OGG) или даже в Windows Audio (WMA). Только помните, для того чтобы потенциальный радиослушатель смог ваш подкаст прослушать, его оборудование (MP3-плеер, медиаплеер в ПЭВМ) должно эти форматы понимать.

Сам себе радиоведущий

Кто размещает подкасты? В первую очередь — радиостанции. В США за этим занятием были замечены Minnesota Public Radio, Air America, Boston public-radio station WGBH, Public Radio International. В Великобритании это, конечно же, BBC, Virgin Radio, Capital. В России главным источником радиопередач является, конечно же, russianpodcasting.ru. На самом деле достаточно удобно прослушивать какую-либо передачу или интервью по мере надобности, а не в прямом эфире.

Размещают подкасты и любители-энтузиасты. Впрочем, одного энтузиазма в этом деле недостаточно, потому что размещение аудиофайлов, обслуживание сервера и канал связи — тема, заслуживающая отдельного разговора. |

Источники информации

- [1] <http://en.wikipedia.org/wiki/Podcasting>
- [2] [http://en.wikipedia.org/wiki/RSS_\(file_format\).html](http://en.wikipedia.org/wiki/RSS_(file_format).html)
- [3] <http://blogs.law.harvard.edu/tech/rss>
- [4] http://en.wikipedia.org/wiki/Really_Simple_Syndication.html
- [5] http://en.wikipedia.org/wiki/List_of_news_aggregators.html
- [6] www.petefreitag.com/item/465.cfm Howto Create an RSS 2.0 Feed
- [7] <http://amarok.kde.org/content/view/58/66>
- [8] www.perli.net/projekte/gpodder
- [9] <http://ipodder.sourceforge.net/index.php>
- [10] <http://jpodder.com>

- [11] www.castpodder.net
- [12] www.linuxforums.org/Linux%20RSS%20Feeds.html
- [13] http://radio.blogware.com/blog/_archives/2005/5/24/882450.html - Proposed standard for podcast ID3 tags
- [14] <http://audacity.sourceforge.net>
- [15] www.easypodcast.com/ - a cross-platform GUI tool for easy podcast publication
- [16] www.podcastingnews.com/Podcasting_Software.html
- [17] <http://applications.linux.com/print.pl?sid=05/03/01/155208> - Podcasting from Linux

Евгений Зобнин

Ионизация консоли

Долгое время местом моего обитания была обычная Linux-консоль с запущенной программой GNU/Screen. Все бы было хорошо, не возникни потребность в графическом браузере, чтении книг в популярном ныне формате djvu и других графических приложениях. И приходилось метаться между консолью и «иксами», так как ни один из оконных менеджеров не позволял достичь той эффективности работы, какая предоставлялась той самой Screen. Так и не наступил бы покой, если бы на глаза мне не попался менеджер окон Ion.

Почему Ion?

Что же такое Ion и чем он так привлек мое внимание? Дело в том, что его разработчики отказались от привычной концепции оконного окружения, придуманной еще компанией Xerox и благополучно взятой на вооружение Apple, Microsoft и 99% менеджеров окон. Вместо нее в Ion используется уже известная, но не заслужившая должного внимания модель фреймов и табов (закладок). Фрейм в данном случае представляет собой область экрана (или весь экран), внутри которой может быть запущено несколько приложений-клиентов. Переключение между ними происходит при помощи табов (напоминающих по функциональности табы веб-браузеров). Понятие окна в данном случае отсутствует вообще, но можно сказать, что окно всегда занимает весь фрейм целиком.

Таким образом, рабочее пространство разбивается на несколько неперекрывающихся областей, а проблема с перетаскиванием и изменением размеров окон, зачастую выводя-

щая из себя, исчезает вовсе. Стоит заметить, что не все пользователи Ion предпочитают подобный стиль расположения фреймов. Многие обычно используют всего один фрейм, занимающий всю рабочую площадь.

Управление фреймами и табами происходит полностью с клавиатуры, поэтому немного освоившийся с сочетаниями клавиш пользователь сможет решать многие задачи буквально за доли секунды, не прибегая к помощи меню и иконок. Плюс ко всему Ion позволяет изменить практически все параметры, разместить запуск часто используемых приложений на клавиатурные комбинации, обладает модульной архитектурой, поддерживает dockapp (из WindowMaker и AfterStep), может быть расширен за счет простого скриптового языка lua, использует скины и содержит в себе еще множество других возможностей. Просуммировав все вышесказанное, мы получаем идеальный рабочий стол для людей, ценящих свое время, а не красоту интуитивно понятного интерфейса.

Инсталляция

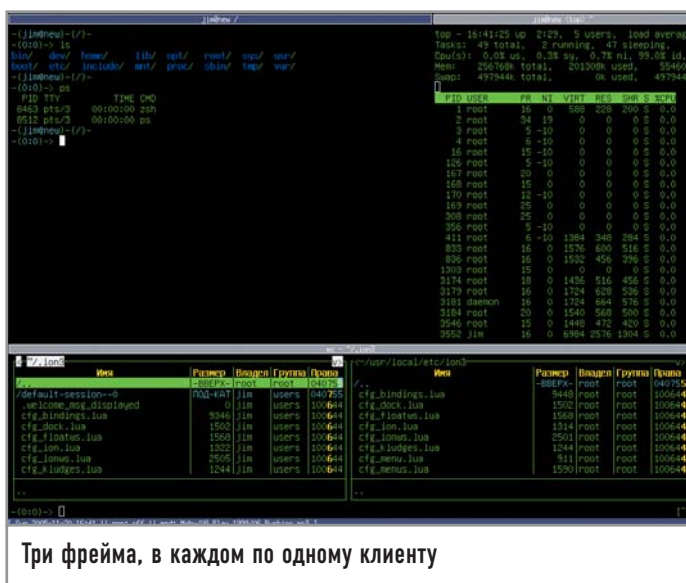
На данный момент существуют две текущие версии программы: стабильная вторая и разрабатываемая третья. Между собой они не совместимы, поэтому конфигурационные файлы от третьей версии не будут работать во второй, и наоборот. Помимо этого новая версия содержит в себе множество полезных функциональных дополнений и улучшений. По своему опыту могу сказать, что она уже достаточно стабильна и вполне подходит для повседневного использования. Поэтому дальнейшее повествование будет посвящено именно Ion 3. Взять его исходные коды можно с официальной страницы <http://modeemi.cs.tut.fi/~tuomov/ion>. Там же расположен архив с документацией, размер которого, что показательно, больше самой программы. Установка весьма тривиальна (`./configure && make && make install`), но требует наличия интерпретатора lua (www.lua.org). Для тех, кто не в курсе, lua — это простой объектно-ориентированный интерпретируемый язык программирования, созданный специально для встраивания в другие приложения. Помимо Ion он используется в текстовом веб-браузере ELinks и других менее популярных приложениях.

Первое знакомство

По окончании установки выходим из X Window, копируем файл `~/xinitrc` в укомное местечко и выполняем команду `echo "exec ion3" > ~/xinitrc`. Эти действия направлены на то, чтобы научить команду `startx` загружать Ion вместо предыдущего оконного менеджера.

После весьма непродолжительного процесса загрузки Ion встретит нас приветственным сообщением и предложит прочитать map-страницу. От чтения документации отказываться не советуем, так как стандартная настройка горячих клавиш крайне запутанна и не поддается интуитивному освоению (в следующем разделе мы это исправим).

После ознакомления с документацией можно начинать работу. Для тех, кто не привык читать мануалы или не смог разобраться с терминологией, осветим основные моменты.



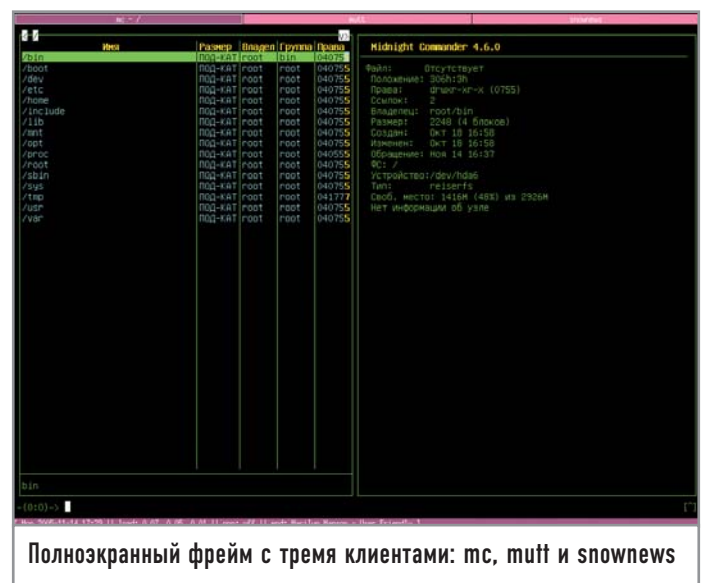
Три фрейма, в каждом по одному клиенту

По умолчанию Ion запускается с одним-единственным фреймом, растянутым во весь экран. Его верхняя строка содержит имена табов, за переключение между которыми отвечают комбинации «Alt+K N» (следующий таб), «Alt+K P» (предыдущий таб) или «Alt+K X», где X — номер таба. Внизу — строка состояния, которую отрисовывает модуль `statusbar` (мы еще займемся его настройкой). Аналог кнопки «Закрыть окно», присутствующей в заголовке окна обычного оконного менеджера, — комбинация «Alt+C». К функциональным клавишам («F1–F12») привязаны различные полезные действия, например: «F1» — просмотр map-страниц, «F2» — запуск `xterm`, «F3» — запуск выбранного приложения. По клавише «F12» доступно основное меню, из которого можно запустить программы, выбрать стиль оформления или выйти из X Window.

А как же хваленый принцип разделения рабочего пространства на области? И правда, по умолчанию у нас на экране присутствует всего один фрейм, но достаточно нажать «Alt+S», экран разделится по горизонтали на две области — и в нашем распоряжении уже два фрейма; еще одно нажатие комбинации — и три. Переключение между ними — «Alt+P» (выше) и «Alt+N» (ниже). Вертикальное разбиение экрана — «Alt+K S», переключение — «Alt+Tab» (вправо), «Alt+K Tab» (влево). Поймите, но симметричные фреймы — это же совсем неудобно. Для включения режима изменения размера фрейма нажимаем «Alt+R», а затем — клавиши направления для увеличения размера или те же самые, но в связке с «Shift» для уменьшения. Да, уничтожает фрейм комбинация «Alt+K X». Не забудьте сохранить сессию («F12 → Session → Save») перед выходом из X Window, чтобы при следующем запуске Ion положение всех фреймов не изменилось.

Вполне естественно, что Ion поддерживает популярную концепцию множества рабочих столов. Вот только эта тема не так проста, потому что здесь предлагается на выбор аж четыре вида рабочих столов:

► WIonWS — стандартный рабочий стол, исповедующий концепцию табов и фреймов. Именно он используется по умолчанию и определяет облик оконного менеджера.



Полноэкранный фрейм с тремя клиентами: mc, mutt и snownews

► WPaneWS — еще один схожий подход к организации рабочего пространства. В этом случае экран разделен на несколько областей фиксированного размера. Каждая область предназначена для запуска определенного типа приложений. Например, dockarr будет автоматически помещен в область небольших размеров, где ему и место, а веб-браузер — в основную. Причем размеры и количество областей, а также алгоритм размещения приложений полностью настраиваемы. Рекомендую поэкспериментировать, очень занятно.

► WFloatWS — специальный рабочий стол, как это ни парадоксально, со стандартным оконным интерфейсом, хотя и не лишенный табов и фреймов. Предназначен для запуска многооконных приложений, таких как GIMP и ImageMagic, управляться с которыми при помощи табов очень проблематично.

► Рабочим столом может стать одна-единственная программа! Для этого используйте комбинацию клавиш «Alt+Enter».

Чтобы получить доступ к новому рабочему столу, следует его создать, нажав «F9», а затем указать имя и тип. Переключение между столами — «Alt+X», где X — порядковый номер. Так как потребность в создании рабочего стола WIonWS возникает чаще, чем других типов, в стандартной настройке комбинация «Alt+F9» привязана к функции создания рабочего стола по умолчанию. Уничтожается рабочий стол, как и все остальное в Ion, при помощи незамысловатой комбинации «Alt+C».

Конечно же, клавиатура — не единственное средство управления. Мышь вполне корректно и предсказуемо работает и позволяет перемещаться между табами и вызывать различные контекстные меню кликом правой клавиши. Причем мышь полностью поддается процессу настройки. С другой стороны, она все-таки остается дополнительным средством управления и полезна разве что на рабочем столе типа WFloatWS.

Как уже было сказано, в нижней части рабочего стола находится строка статуса. Она очень похожа на аналогичную строку консольного Screen и предназначена для отображения различной динамически меняющейся информации. Теоретически модуль statusbar, отвечающий за отображение этой строки, можно запрограммировать на показ чего угодно.



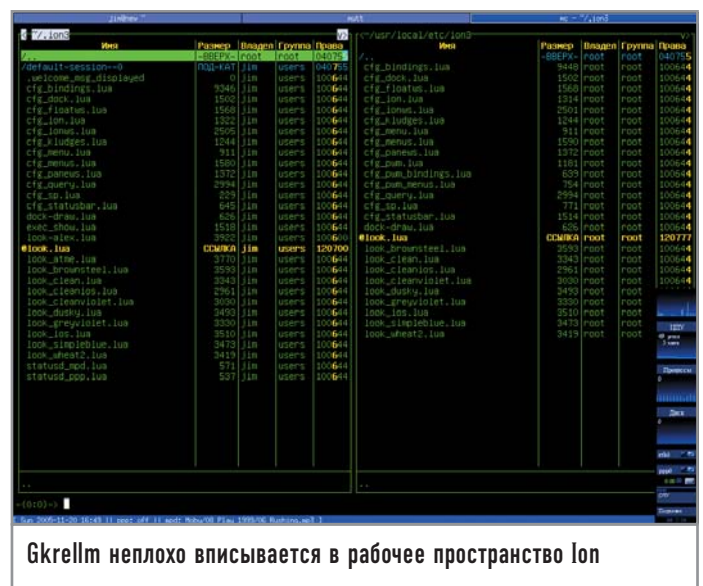
В Интернете доступны скрипты, отображающие зарядку батареи ноутбука, состояние сетевого интерфейса, название композиции, проигрываемой XMMS, и т. д. По умолчанию в строке отображаются время, общая загруженность системы и количество писем в ящике.

Так как по своей сути статусная строка — это всего лишь строка, она не может отрисовывать графики, индикаторы и прочую графическую информацию. Эту проблему можно побороть, используя так называемый dock. Он хорошо знаком пользователям WindowMaker, AfterStep и Fluxbox (правда в этом оконном менеджере он называется Slit). Dock представляет собой небольшое окошко (обычно не имеющее рамок) и позволяет разместить в своем пространстве несколько dockarr — приложений с размером окна примерно 64x64, которые как раз и занимаются отрисовкой различных графиков (загруженность процессора, количество памяти, объем дискового пространства и т. д.). Знаменитый Gkrellm тоже может стать dockarr, если в настройках поставить галочку напротив опции «Установить тип окна dock или panel». Модуль dock в Ion размещает dock в определенном углу экрана так, чтобы он перекрывал все фреймы, независимо от выбранного типа рабочего стола. Чтобы dock не мешал работать в других приложениях, его можно спрятать, нажав «Alt+D».

Настройка

Повсюду в Ion можно обнаружить следы языка lua, и даже все настроечные файлы являются lua-скриптами и содержат множество вызовов процедур, объявлений функций и комментариев. Обычному пользователю, привыкшему к графическим конфигураторам, такой способ настройки может показаться безумно сложным и неоправданным. На самом деле lua — очень простой язык, позволяющий без особого труда добавить программе уникальную функциональность.

Для начала предлагаю скопировать рабочую конфигурацию из каталога /usr/local/etc/ion3 (каталог может быть иной, если во время установки был указан другой префикс) в ~/.ion3. Главный конфигурационный файл носит имя cfg_ion.lua и включа-



```

-- Open and close CD
kpress(MOD1.."F10", "ioncore.exec_on( 'sh -c eject -t $1 mount /mnt/cdrom' )",
kpress(MOD1.."Next", "ioncore.exec_on( 'eject' )",

-- Execute a command
submap(MOD1.."E", {
  bdoc("Run a terminal emulator.",
  kpress("T", "ioncore.exec_on( 'xterm' )",

  bdoc("Run a root terminal.",
  kpress("C", "ioncore.exec_on( 'su -' )",

  bdoc("Query for command line to execute.",
  kpress("X", "wod_query.query_exec( )",

  kpress("Q", "ioncore.exec_on( 'q' )",
  kpress("H", "ioncore.exec_on( 'halt' )",
  kpress("P", "ioncore.exec_on( 'poweroff' )",
  kpress("N", "ioncore.exec_on( 'shutdown' )",
}),

-- Set functions
submap(MOD1.."W", {
  kpress("W", "ioncore.exec_on( 'sudo /etc/rc000/app-on' )",
  kpress("R", "ioncore.exec_on( 'fetchall' )",
  kpress("N", "ioncore.exec_on( 'sudo /usr/local/sbin/fetchnews -vv' )",

  bdoc("Query for host to connect to with SSH.",
  kpress("S", "wod_query.query_ssh( 'ssh' )",
}),

-- WFO control
submap(MOD1.."F", {
  kpress("F", "ioncore.exec_on( 'app play' )",
  kpress("G", "ioncore.exec_on( 'app stop' )",
  kpress("D", "ioncore.exec_on( 'app done' )",
  kpress("C", "ioncore.exec_on( 'app close' )",
  kpress("S", "ioncore.exec_on( 'app play' )",
  kpress("par10", "ioncore.exec_on( 'app next' )",
  kpress("I", "ioncore.exec_on( 'ioncore' )",
  kpress("plus", "ioncore.exec_on( 'app volume +10' )",
  kpress("minus", "ioncore.exec_on( 'app volume -10' )",
}),

```

Правим `cfg_bindings.lua` в Vim

ет в себя основные установки, а также загрузку других полезных модулей и скриптов (функция `dopath()`). Каждый модуль выполняет определенный набор функций и, соответственно, имеет собственный конфигурационный файл.

Наработки сообщества

Готовые расширения

В мире Open Source уже давно и успешно работает неписанный закон: «Перед тем как начать что-то создавать самому, узнай, не создали это до тебя». К Ion он тоже применим: существует репозиторий скриптов-расширений, в который следует заглядывать, перед тем как писать скрипты самостоятельно. Вот список самых, на мой взгляд, интересных представителей этого репозитория:

- ▶ `cfg_dock.lua` — конфигурационный файл для `dock` с расширенными возможностями управления. Теперь `dock` можно переместить в любую часть экрана при помощи клавиатурных комбинаций.
- ▶ `ctrl_statusbar.lua` — конфигурационный файл для строки состояния. Представляет собой меню, вызываемое правым кликом мыши.
- ▶ `document_menus.lua` — перемещение по файловой системе с использованием иерархии меню.

- ▶ `enumerate.lua` — пронумеровывает табы, упрощая навигацию.
- ▶ `exec_show.lua` — великолепный скрипт, позволяющий просматривать вывод `shell`-команд во всплывающем окне Ion.
- ▶ `rss_feed.lua` — встраивает в Ion настоящий просмотрщик RSS-новостей.
- ▶ `schedule.lua` — позволяет запрограммировать вывод произвольного сообщения в определенное время; эдакий будильник.
- ▶ `emacs-bindings.lua` — клавиатурные комбинации в стиле редактора Emacs.
- ▶ `statusd_netmon.lua` — монитор сетевой активности для строки состояния.
- ▶ `statusd_laptopstatus.lua` — показывает температуру, зарядку батарей и скорость процессора в строке состояния. Работает с ACPI.
- ▶ `statusd_xmms.lua` и `statusd_mpd.lua` — показывает название композиции, проигрываемой в данный момент плеером XMMS или mpd.

```

To enable bug compatibility mode:
bc

To turn keyclick off:
c

To set keyclick volume:
c [0-100] c on

To control Energy Star (DPM) features:
-dpm Energy Star features off
-wdpm Energy Star features on
-dpm [standby | suspend | off]
force standby
force suspend
force off
force on
(also implicitly enables DPM features)
s [timeout value of zero disables the mode]

To set the font path:
fp: path[,path,...]

To restore the default font path:
to default

To have the server reread font databases:
to refresh

To remove elements from font path:
fp: path[,path,...]
fp: path[,path,...]
fp: path[,path,...]
fp: path[,path,...]

look_size s off or on
look_size s on
look_brightness led on
look_brightness led off
look_clean ejection and threshold:
[acc_div] [thr] m default
acc:
look_cleanstate r on
look_color_name s color_name
look_heartbeat led off or on
r off
look_busy r on
look_refresh_rate [rate]
control:
look_in [cycle] s default s on
s make s off
look_timeable s noescape s reset
look_wheat [matron: q
Refresh list

```

Меню Ion

Начнем, как и было обещано, с настройки горячих клавиш. Для этого предусмотрен скрипт `cfg_bindings.lua`. Большинство привязок клавиш задается именно в нем, хотя никто не запрещает делать это в любом другом скрипте, главное — загрузить его из `cfg_ion.lua`. Процесс конфигурации привязок осуществляется посредством вызова функции `defbindings()` (на самом деле это метод класса `ioncore`, но в этом и всех последующих случаях мы опустим ненужные подробности) с двумя аргументами: контекст и блок вызова функции `kpress()`. Предусмотрено четыре контекста.

- ▶ `WScreen` — комбинации работают везде.
- ▶ `WMplex` — комбинации работают во всех фреймах и рабочих столах.
- ▶ `WFrame` — комбинации работают во всех фреймах. То есть везде кроме «голого» рабочего стола типа `WFloatWS` и рабочего стола в виде одного приложения.
- ▶ `WMoveresMode` — комбинации доступны в режиме изменения размера и перемещения.

Типичный вызов `defbindings()` выглядит примерно так:

```

defbindings("WScreen", {
  kpress(MOD1.."F9", "ioncore.create_ws( )",
})

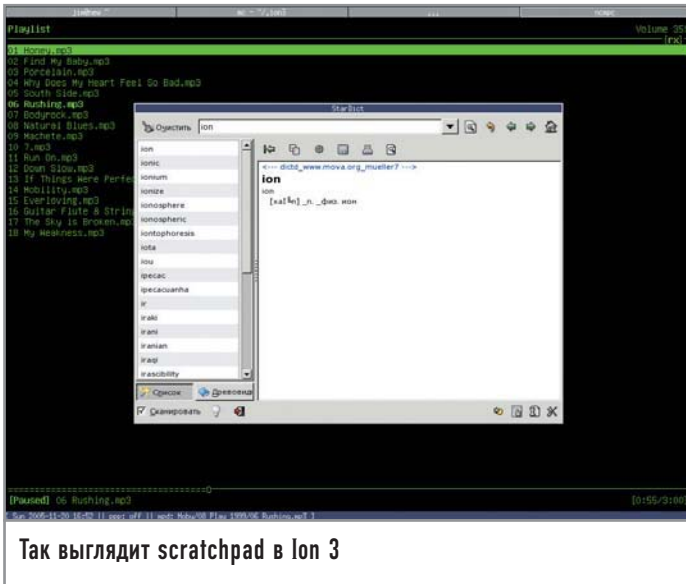
```

Подобная запись означает, что нужно привязать комбинацию «Alt+F9» к функции создания нового рабочего стола. `MOD1` содержит строку «Alt+», а оператор «..» складывает две строки (в данном случае — «Alt+» и «F9»). Не пугайтесь, если ничего не поняли, скрипт `cfg_bindings.lua` уже содержит все необходимые для вызова функции, остается только изменить стандартные комбинации на свои собственные. Хорошей идеей будет изменить комбинации «Alt+X» (переключение между столами) на «Alt+FX», предварительно закоментировав (в lua комментарий — «--») все строки, привязывающие «Alt+FX» к другим функциям. Освободившиеся «Alt+X» можно запрограммировать на переключение между табами, что гораздо удобнее, чем «Alt+K X». Также можно прописать следующие строки:

```

-- Execute a command
submap(MOD1.."E", {

```



```
kpress("T", "ioncore.exec_on(_, 'xterm)'),
kpress("M", "ioncore.exec_on(_, 'mc)'),
kpress("E", "ioncore.exec_on(_, ':elinks)'),
})
```

Теперь по комбинации «Alt+E T» будет запускаться xterm, по «Alt+E M» — mc и т. д., можете добавить строки для запуска нужных вам программ. Кстати, знак «:» перед командой указывает на то, что ее следует запускать в новом терминале. Как видите, функция submap позволяет создавать многоуровневые комбинации, благодаря чему мы получаем огромное разнообразие оных. Например, клавиатурные комбинации можно использовать для управления аудиоплеером, запуска сетевых функций (подключение, отключение, SSH и т. д.), выдвижения лотка CD-ROM, монтирования файловых систем и многого другого.

Теперь создадим свое меню (то, которое вызывается по «F12»). Для этого опять же предусмотрен особый скрипт cfg_menus.lua, в него мы и впишем нужные нам строки. Чтобы понять, как выглядит меню в исходном (на языке lua) виде, рассмотрим следующий пример:

```
defmenu("mainmenu", {
    submenu("programs", "appmenu"),
})
```

```
defmenu("appmenu", {
    menuentry("xterm", "ioncore.exec_on(_, 'xterm)'),
})
```

Здесь мы при помощи функции defmenu() создаем основное меню (оно всегда носит имя mainmenu) и добавляем в него подменю appmenu с именем programs. Далее по описанной выше схеме создается подменю appmenu, которое будет содержать пункт xterm, запускающий одноименную программу. Вот и все, за подсказками обращайтесь к содержимому оригинального скрипта.

Статусная строка конфигурируется в файле cfg_statusbar.lua, и только в нем. Содержимое строки задается в переменной template в виде текста и специальных слов.

Рассмотрим пример:

```
template="[ %date || load: %load || mail: %mail_new/%mail_total ]"
```

Специальные слова %date, %load, %mail_new и %mail_total при выводе будут заменены на показания соответствующего счетчика, все остальное пойдет на экран в неизменном виде. Все четыре перечисленных счетчика являются встроенными. Чтобы создать свой счетчик, нужно написать специальный скрипт с именем «statusd_имя.lua», поместить в каталог ~/.ion3 и вставить его название в переменную template (не забыв указать знак %). Написание такого скрипта я рассматривать не буду, так как это требует описания синтаксиса lua.

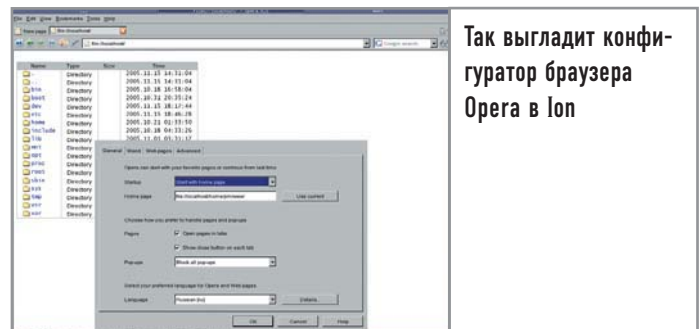
Ну и напоследок внесем несколько косметических изменений. Заглянем в файл cfg_dock.lua и изменим позицию dock. Для этого запишем в переменную pos одно из четырех значений: tl (верхний левый угол), tr (верхний правый угол), bl (нижний левый угол) и br (нижний правый угол). Также полезно изменить направление роста (с какой стороны будет добавляться новый dockapp), присвоив переменной grow значение left, right, up или down. В Ion 3 есть интересный модуль scratchpad, отключенный по умолчанию. Scratchpad — это не зависимый от рабочих столов фрейм размером 640x480, появляющийся в центре экрана после нажатия комбинации клавиш «Alt+Пробел». Его удобно использовать для запуска аудиоплеера, словаря и других приложений, которые не должны присутствовать на экране постоянно. Чтобы активировать этот модуль, раскомментируйте строку dopath("mod_sp") в скрипте cfg_ion.lua. |

Дополнительная информация

Работа с диалоговыми и уведомляющими окнами

Может быть, у кого-то из читателей возник вопрос: как Ion поступает с диалоговыми и уведомляющими окошками? Ведь если такие окна будут открываться в новом табе, то идея «каждой программе по табу» пойдет коту под хвост, а работать в таком оконном менеджере станет невыносимо. Но и с этой про-

блемой программисты справились, придумав интересное решение. По определенным признакам Ion распознает окна такого типа, уменьшает их размеры до одной трети экрана и отрисовывает прямо поверх существующей картинке. Таким образом, диалоговое окно как бы вливается в основное.



ТВ-приставка на мониторе

Этот обзор посвящен эмуляторам устаревших игровых консолей, классическим культовым ТВ-приставкам NES, SEGA Genesis/Mega Drive и Super Nintendo. Впрочем, называть их раритетами нельзя — они до сих пор присутствуют на отечественном рынке и пользуются спросом, будучи дешевле современных и невероятно мощных PlayStation 2 и Xbox, не конкурируя с чем-либо, а просто заполняя нишу недорогих игровых консолей.

Но вот беда — картриджи с играми к таким приставкам стоят довольно дорого. Да и не купишь все, не найдешь. Но если пойти в Сеть и поискать, то можно найти так называемые ROM — файлы с образами, снятыми с картриджей. То есть сами игры либо их сборники. Разумеется, чтобы запустить ROM и играть в игры, нужен эмулятор соответствующей приставки. Сразу отмечу, что существуют законные и незаконные ROM, а также созданные энтузиастами. Кроме того, есть ROM коммерческих игр, в том числе и очень старые, которые компании-издатели до сих пор не отдали народу задаром. Недавно появилось понятие «warogware», когда старые игры списывают, переводят в статус бесплатных, свободно распространяемых.

Обычно сайты с ROM коммерческих игр держатся на плаву от силы несколько недель. В России, впрочем, есть и рекордсмены. Я не буду рассказывать, где можно взять образы. Ищущий да обрящет. Ведь эмулятор — это как компьютер. На нем можно запускать пиратский софт и легальный. С помощью программ для создания ROM вы можете создать копии собственных игр (при наличии необходимого для этого железа), переведя их с картриджей в ROM, и играть уже на эмуляторе. Существуют ROM с разными графическими и звуковыми демо-версиями, а также прикладные программы для консоли NES. Тут уже все совершенно легально. Подробную информацию можно получить, например, на сайте <http://nesdev.parodius.com>.

Эмуляторов в Linux достаточно много. Нас интересуют эмуляторы приставок NES, SEGA Genesis и Super Nintendo (SNES).

В этом обзоре основное внимание уделено тем из них, которые я использую сам, потому что считаю их лучшими по следующим параметрам: простота установки, поддержка максимального количества игр и возможность играть с помощью геймпада или джойстика. Ну и, конечно же, немаловажное значение играет наличие дополнительных функций, таких как сохранение/загрузка в несколько слотов и скорость рендеринга. Об остальных эмуляторах я тоже расскажу, однако мимоходом, подчеркнув лишь их слабые и сильные стороны.

| NES |

Начнем с эмуляторов консоли NES. Некоторые читатели, возможно, недоумевают — что это за NES такая? Сокращение NES расшифровывается как Nintendo Entertainment System. У нас она известна благодаря клону — Dendy. Возможно, подобные устройства до сих пор пылятся у многих на антресолях. Кроме Dendy на рынке были и остаются китайские клоны NES вроде Zhilitong. В народе NES называют «Денди» либо просто «восьмибиткой».

Самый популярный эмулятор NES для Linux — это, конечно же, TuxNES (<http://tuxnes.sourceforge.net>). Последняя его версия вышла в 2001 году. Впрочем, это не должно огорчать: бытует мнение, что эмуляторы, как хорошее вино, — чем старше, тем лучше.

На первый взгляд TuxNES не впечатляет. Маленькое окошко, геймпада не видно, разве что ресурсы процессора поедаются довольно умеренно. Но оказывается, у эмулятора очень много параметров, передаваемых через командную



строку. Не так много, как в MPlayer, но все-таки достаточно. Например, не видит TuxNES джойпада. А мы зададим ему опцию `--jsl=/dev/js`, и все заработает.

Игровое окно маленькое? Его можно увеличить командой: `tuxnes -E <имя файла с игрой>`. Можно выбрать и рендерер по вкусу, запустив эмулятор следующим образом:

`tuxnes -r=<значение>`

Здесь значение может быть одним из следующих: `x11`, `diff` или `old` (старый рендерер).

TuxNES умеет делать скриншоты (клавиши «S», «F7», «PrintScreen»). Цифровые клавиши от 0 до 8 управляют скоростью эмуляции, а не, как это иногда бывает, слотами сохранения? Увы. В TuxNES сохранения нет.

По поводу совместимости с играми замечу, что удивительная Battle Toads работает. А раз нормально можно играть в такую продвинутую игру, то остальные тоже наверняка будут работать. Даже King's Quest V. Не пойдут только самые передовые игры, вышедшие в середине 90-х годов вроде Aladdin, или «переформатированные» (Lion King, портированная с 16 бит). И придется обратить взор на другой эмулятор. Им будет, вероятно, Fceu (www.icewalkers.com/Linux/Software/58270/FCE-Ultra.html).

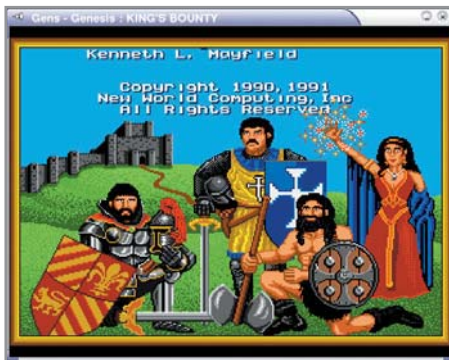
Легендарный Fceu... Если бы не его нелюбовь к геймпаду (хотя, по идее, он должен работать), я бы все время использовал для эмуляции NES именно его. Чем же именно он так хорош? А рендеринг через OpenGL не хотите ли? А поддержку всех игр, какие только можно себе представить? Все это есть. Полноэкранный режим? Да, в отличие от TuxNES. Сохранение, загрузка состояния игры — к вашим услугам.

Если у вас нет джойстика или его поддержка не критична, то Fceu полностью удовлетворит ваши потребности в эмуляции NES. А как насчет SEGA?

| SEGA |

Под именем SEGA у нас подразумевают обычно 16-битное детище корпорации SEGA, которое называлось Mega Drive (в Японии и Европе) и Genesis (в США). К концу девяностых годов прошлого столетия на игровом рынке более чем 90% принадлежало NES. Восьмибитная приставка от SEGA — Master System, — будучи технически более развитой, чем NES, не смогла победить конкурента. Превзойти NES можно было лишь благодаря техническому прорыву. Отмечу, что годом ранее в продаже уже появилась 16-битная приставка компании NEC — PC Engine, но темпы ее продаж были довольно низкими. И вот выходит SEGA Mega Drive. Тогда она стоила \$200. Мало кто знает, что в нее встроен эмулятор Master System, а некоторые картриджи к Mega Drive — на самом деле игры Master System, адаптированные под новый разъем картриджа. Центральный процессор в Mega Drive — 16-битный Motorola 68000 (около 8 МГц, частота варьируется в зависимости от версии консоли для PAL или NTSC), а для обработки звука использовался старый добрый восьмибитный Zilog Z80, работавший на частоте около 4 МГц. Этот же Z80 работал как основной процессор в режиме эмуляции Master System. Хотя какая тут эмуляция — скорее это натуральный режим работы. В конце девяностых и начале нынешнего века Mega Drive прочно делила отечественный рынок с NES. SNES (Super Nintendo), о которой еще пойдет речь ниже, не снискала в





России такой народной любви (одна из причин — пираты создавали в основном клоны NES и Mega Drive).

Для эмуляции Mega Drive первым делом советую попробовать Gens. На официальном сайте программы (www.gens.ws) вы не обнаружите версию для Linux. Исходник многоплатформенной версии можно отыскать на <http://sourceforge.net/projects/gens>. В нем содержится версия Gens 2.12-rc3. Справедливости ради отмечу, что стоит поискать исходник этой версии отдельно, поскольку скомпилировать его будет проще, нежели аналогичную версию из «многоплатформенного» дистрибутива. В последнем случае могут возникнуть проблемы с версией Autotools (это программы-скрипты automake, autosconf и другие). И собрать исходник можно будет только заменив используемые в нем файлы вроде dercomp, config, config.sub и им подобные на эти же файлы из вашего дистрибутива Linux (достаточно сделать на них ссылки). Возиться с этим захочет не каждый, однако я собрал свою версию именно так. При наличии навыков это займет лишь три минуты на установку.

Gens считается одним из лучших эмуляторов Mega Drive под Windows. Справедливо это мнение и применительно к версии для Linux. Графический движок Gens завязан на библиотеке SDL, а интерфейс создан на основе GTK+ 2.4 (или выше). При запуске на рабочем столе отдельно отображается окно с игрой и отдельно — узкое GTK-окошко с полоской меню. Эмулятор может работать и в полноэкранном режиме (переключается комбинацией клавиш «Alt+Enter»).

Графика в эмуляторе на высшем уровне. Есть окно подстройки яркости и контраста. Там же можно включить режим, при котором игры работают в режиме градаций серого цвета — вероятно, для любителей вспомнить молодость, когда не все телевизоры поддерживали видеостандарт PAL и приходилось играть на цветном телевизоре в черно-белые игры. Присутствует функция вертикальной синхронизации, а также некая «Stretch» (растягивание). Как действует последняя, осталось для меня тайной. Можно выбрать один из целой дюжины рендеров, что, впрочем, повлияет только на оконный режим. В полноэкранном используется только один рендерер — Double. С помощью комбинации клавиш «Shift+Backspace» создается скриншот текущего кадра.

Gens прекрасно эмулирует и звук, и музыку. Кстати, для музыки эмуляторам Mega Drive приходится эмулировать два чипсета — YM2612 от Yamaha и SN76489 от Texas Instruments. Последний используется в озвучке игр к Master System.

В меню «Sound» я нашел пункт «Start WAV dump» — для вывода звука во внешний WAV-файл, однако эта опция за-

блокирована. Насколько я помню, в Windows-версии аналогичный пункт меню работает.

В эмуляторе очень хорошо реализована система сохранения — по десять слотов сохранения/загрузки на каждую игру. За всеми этими действиями закреплены горячие клавиши. Более того, формат сохранения совместим с такими эмуляторами Mega Drive как DGen и Genecyst.

Одной из изюминок Gens является возможность настраивать клавиши управления — делается это в окне «Controller Setting», которое вызывается через пункт меню «Option → Joypads». Кроме обычной клавиатуры поддерживается и геймпад. В меню «Option» находятся еще две полезные функции «Load config» и «Save config As» — для загрузки и сохранения альтернативных файлов конфигурации, разные варианты которых зачастую бывает удобно хранить для различных игр. В файле конфигурации сохраняются и настройки клавиш/геймпада.

Прочие замечания. В плане совместимости с играми — я не встречал еще такие ROM, которые бы не запускались в Gens. В играх «с батарейками» работают обыкновенные, предусмотренные в таких играх, сохранение и загрузка.

Gens имеет обыкновение потреблять в полноэкранном режиме почти все ресурсы процессора, какой бы он ни был частоты. При установленной Karamba/SuperKaramba играть в Gens практически невозможно — прерывается звук. Из-за этого я удалил со своего компьютера SuperKaramba.

Мог ли я заменить Gens на какой-нибудь другой эмулятор? Давайте посмотрим.





Есть такая программа DGen (www.pknet.com/~joe/dgensdl.html). Разработка ее остановлена. Графического интерфейса нет. Запускается из командной строки. Настраивается через нее же, а еще посредством файла конфигурации. В руководстве к нему очень смутно описано, как настраивать геймпад. А без ручной настройки работают только крестовина и основные клавиши, но не кнопка «Start». Вместо нее приходится нажимать «Enter».

DGen оснащен механизмом сохранения/загрузки (клавиши «F2» и «F3») опять-таки в слоты, а также возможностью записи и воспроизведения демороликов в свой формат.

Звук в DGen, на мой взгляд, более качественный, чем в Gens, потому что 16-битный. А еще DGen поддерживает рендеринг через OpenGL.

Теперь о недостатках DGen. На моей машине (Athlon XP 2600) DGen независимо от режима рендеринга потреблял около 99% ресурсов процессора. Лично меня в DGen не устраивает разве что сложность в настройке управления.

И последний в обзоре эмулятор Mega Drive — Generator (www.squish.net/generator/). Я скачал версию Generator 0.35.gtk. Для пользовательского интерфейса в ней применен старый GTK+ версии 1, а для графики — вездесущая SDL. Кроме того, Generator можно компилировать под SVGAlib и Allegro.

Приятный эмулятор — правильно срабатывает кнопка «Start» на моем геймпаде, ресурсы процессора потребляются умеренно. Работает даже функция записи видео и звука в AVI! Чтобы полу-

ченный файл воспроизводился на большинстве плееров, надо зайти в меню «Emulation → Options» и на вкладке «Logging» в секции «AVI output settings» в списке форматов выбрать значение «RGB» (с MJPEG-сжатием наблюдаются проблемы).

Всем этот эмулятор хорош, да нет у него в SDL-версии полноэкранного режима. Последнее обновление Generator было проведено в мае 2003 года. В то же время Gens является самым свежим работающим эмулятором Mega Drive для Linux — его последний релиз вышел зимой 2005 года.

Super Nintendo

SNES — Super Nintendo Entertainment System. Вживую я никогда не играл на SNES. Этот 16-битный конкурент SEGA Mega Drive превосходил последнюю по техническим характеристикам, однако обогнал ее в популярности только на островах Японии. Остальной мир предпочел Mega Drive.

Лучший эмулятор SNES всех времен и народов, а также всех платформ — ZSNES (www.zsnes.com). Программа активно разрабатывается, причем и Linux-порт никто не пускает на самотек.

Графический интерфейс в ZSNES реализован на оригинальном движке и работает внутри окна, в котором запущен эмулятор. Доступен этот интерфейс и в полноэкранном режиме.

В ZSNES можно выбрать один из нескольких рендереров, что позволяет использовать эмулятор как на относительно маломощных компьютерах, так и на «навороченных» машинах. Каждому по потребностям. Можно настроить видеорежим, причем кроме обычных доступны также и OpenGL-варианты — оконные и полноэкранные. Богат ZSNES и звуковыми настройками — в меню можно найти разные виды интерполяции, фильтры и включение режима «Surround».

Поддерживаются сохранение и загрузка состояния игры. В процессе работы эмулятор загружает процессор примерно на 99%, снижая нагрузку только в режиме паузы. ZSNES довольно стабилен — только однажды он у меня «вылетел», на самом ответственном моменте в Donkey Kong. Хотя, возможно, дело было в проблемах с OpenGL на моем компьютере.

Вступая в игру

Как видим, в Linux можно подобрать эмулятор практически для любой консоли — хотя для Windows эмуляторов все-таки больше. Но играть есть на чем и, главное, во что. А некоммерческие ROM для всех игровых консолей можно найти на сайте www.pdroms.de/index.php.

Любопытные факты

Из истории NES

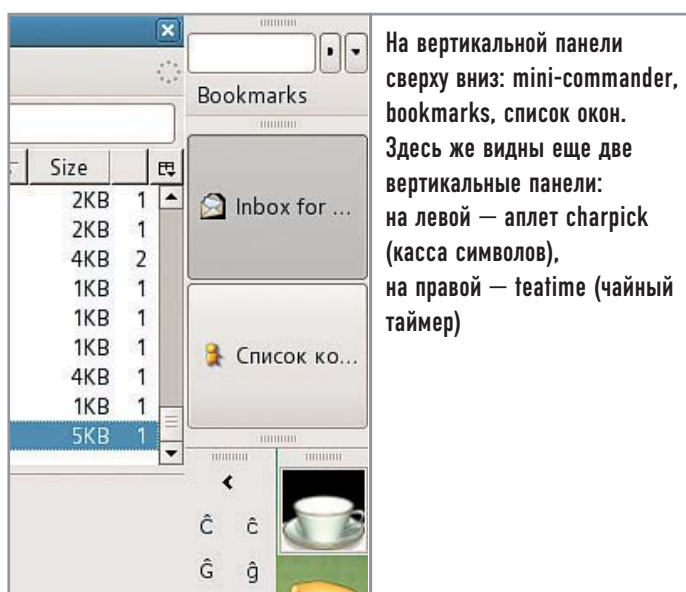
На японском рынке NES появилась в 1983 году под названием Famicom (Nintendo Family Computer). Два года спустя компания Nintendo выпустила американскую версию консоли, уже под именем NES. В течение почти десятилетия она была лидером продаж среди игровых консолей, и лишь в 1988 году эре NES был положен конец с приходом 16-битной

SEGA Genesis/Mega Drive, ставшей королевой консолей к началу 90-х годов. Между тем японское подразделение Nintendo продолжало выпускать NES вплоть до 2003 года, а производство было свернуто по причине нехватки нужных электронных компонентов. Впрочем, это не мешает другим компаниям выпускать клоны NES по сей день.

Панель и ее обитатели

В этой статье вы узнаете о нескольких представителях сословия апплетов. Это скромные по размеру программы с узкой функциональностью, место обитания которых — панель оконного менеджера.

Многие дистрибутивы Linux вносят лишь незначительные изменения в исходное расположение панелей GNOME. Стандартная для GNOME конфигурация — пара панелей, которая появляется при первом запуске GNOME. Сверху — меню и несколько кнопок запуска приложений, снизу — панель задач и переключатель рабочих столов, а где-нибудь — часы, индикатор раскладок, область уведомлений. Многие любители «приборных панелей» с кучей индикаторов сразу устанавливают Gkrellm как стандарт де-факто для GTK-монитора ресурсов и прочего. Между тем панель GNOME можно расположить как горизонтально, так и вертикально, и есть полезные апплеты, позволяющие ей составить конкуренцию Gkrellm, если у него не слишком сложная конфигурация (конечно, возможности этого монитора оставляют далеко позади апплеты GNOME, однако не всем это изобилие нужно).



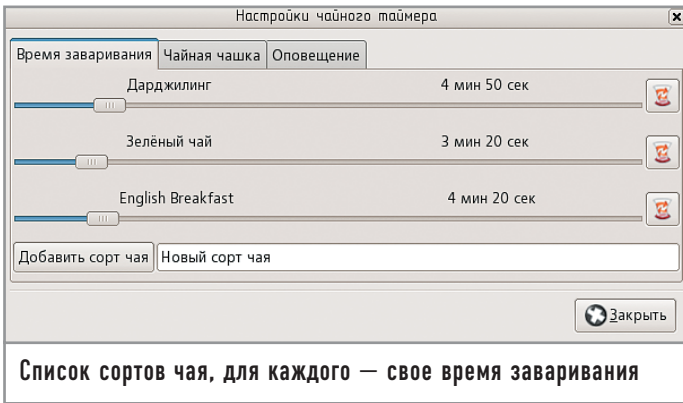
| Кручу-верчу... |

Горизонтальное пространство на современном (с разрешением 1024x768 и более) экране дешевле, чем вертикальное. Другими словами, добавить хотя бы строчку к экрану хочется чаще и сильнее, нежели вертикальную полосу. Согласитесь, читать длинные строки не так уж удобно, зато от дополнительной строчки на экране хуже явно не станет. В этом разделе я расскажу о том, какие элементы можно разместить на вертикальной панели и что с ней можно сделать для общей пользы.

Начнем со строки меню. Если вы наберетесь смелости и перетащите ее на вертикальную панель, то обнаружите, что строка меню разворачивается вертикально. В GNOME 2.12.1 тот же фокус можно проделать и с часами. Правда, на одной и той же вертикальной панели часы и строка меню будут направлены строго в противоположные стороны. Хочется верить, что в следующих версиях это будет исправлено. Так или иначе, мы видим, что есть возможность освободить место, занимаемое верхней панелью, перенеся все ее содержимое на боковую панель.

Вместо верхней панели можно попробовать перенести вбок нижнюю. Здесь нас больше всего интересует апплет списка окон. Направление текста в этом апплете на вертикальной панели остается горизонтальным. Но если сделать панель шире (максимально допустимая ширина панели — 120 точек, для наших целей хватит и 100), этого будет достаточно, чтобы идентифицировать окна. В результате мы получаем большие кнопки окон (больше, чем мы могли позволить себе, когда панель была внизу), по которым гораздо проще попасть. Очевидно, это не бесплатно, все-таки мы отрезали довольно широкую вертикальную полосу от рабочего стола. В данном случае интересна идея, а дальше каждый пусть выберет то, что ему удобнее.

А теперь — краткий рассказ о нескольких полезных апплетах для панели GNOME.



| Таймеры |

В KDE с давних пор имеется программа KTea, помогающая заваривать чай. Программа простая: лежит себе значок в области уведомления (известной в народе как системный трей) и по клику мыши начинает отмерять время, необходимое для заваривания чая. Для GNOME 2 подобные программы появились сравнительно недавно. На сегодняшний день для заваривания чая (и вообще для замера промежутков времени) можно воспользоваться одним из двух апплетов — *teatime* и *timer*.

Апплет *teatime*, как следует из названия, является прямым конкурентом KTea. На панели он выглядит как чашка или кружка, из контекстного меню можно выбрать нужный сорт чая, после чего таймер будет автоматически запускаться. Последний выбранный чай можно заварить еще раз кликом левой кнопки мыши по апплету. Когда время заваривания истечет, на экране появится чашка с надписью «Ваш чай готов». Все просто и понятно.

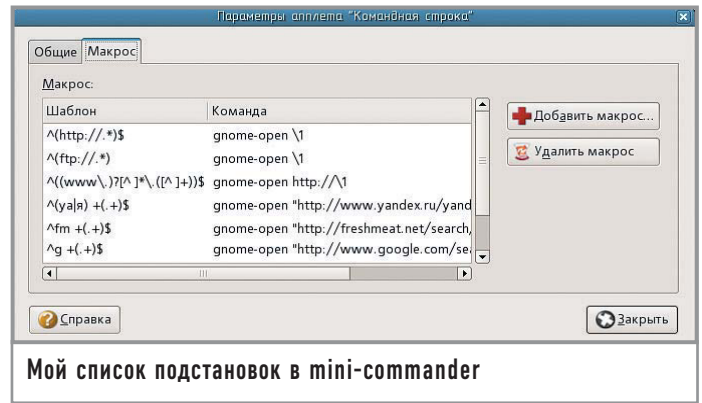
Второй апплет называется *timer* и тоже полностью соответствует своему названию, претендуя на некоторую универсальность по сравнению с *teatime*. Оформлен он более строго и показывает оставшееся время не в виде графики, а цифрами, что бывает очень полезно. К сожалению, окно с предупреждением об истечении времени отображается только на том рабочем столе, который был текущим на момент окончания; об этом окошке легко забыть, если в это время вы переключились на другой рабочий стол.

В Глобальной сети апплет *teatime* можно найти по адресу <http://teatime.sourceforge.net>; апплет *timer* располагается на сайте <http://timerapplet.sourceforge.net>.

| mini-commander |

Хотя этот апплет входит в стандартный набор апплетов GNOME, я бы хотел на нем остановиться подробнее, его возможности заслуживают внимания.

Диапазон средств для выполнения команд, набранных от руки, простирается от спартанского диалогового окна «Запустить программу» до «навороченной» оболочки вроде *zsh*, работающей в терминальном окне. Апплет *mini-commander* находится где-то посередине. На первый взгляд он мало чем отличается от диалогового «Запустить программу», который вписан в панель. Но если, например, набрать в строке апплета что-нибудь начинающееся с «<http://>», запустится браузер с адресом, который вы набрали. А если набрать «*g*:», а следом... угадали — все, что после двоеточия, будет использовано для поиска в Google.



Мой список подстановок в mini-commander

Самое приятное состоит в том, что такие магические строчки можно создавать самостоятельно. Правда, для этого понадобятся знания, которым в средних школах пока не учат. В параметрах апплета есть отдельная вкладка под названием «Макрос» (макрос там на самом деле не один — это огрех перевода) со списком выполняющихся подстановок. Так вот, в первой колонке находится не просто строка, а регулярное выражение. Благодаря регулярным выражениям подстановки в *mini-commander* обладают потрясающей гибкостью и мощностью, но, как и в случае с любым солидным инструментом, требуют умелого обращения. Если вы еще не знаете регулярных выражений, потратьте пару часов на освоение хотя бы азов. Вы никогда не пожалеете об этом. Очень рекомендую прочитать хотя бы первую главу книги Фридла «Регулярные выражения».

| bookmarks |

Это очень простой и удобный апплет, который предоставляет доступ к закладкам основного браузера прямо из панели GNOME. Поддерживаются Galeon, Epiphany, Mozilla и Firefox. Ничего особенного, просто меню закладок. Вызывать редактор закладок пока нельзя, пользоваться «умными» закладками вы тоже не сможете. Но основную свою функцию этот апплет выполняет исправно. В Интернете он находится по адресу: <http://browserbookapp.sourceforge.net>; там же можно посмотреть на еще один очень интересный апплет, который, возможно, будет включен в GNOME 2.14, — *desktop*, строка поиска, вынесенная на панель GNOME.

| hardware-monitor |

Другое название этого же апплета — «Сенсоры». Апплет *Sensors* в природе тоже есть, но не имеет ничего общего с *hardware-monitor*. Последний гораздо более функционален.

С его помощью вы можете следить за данными сенсоров (как температуры, так и скорости вентиляторов), загрузкой процессора, использованием диска и памяти, сетевыми интерфейсами. Все это можно отображать как в виде текста, так и графиков, причем их стиль может быть разным, и на одном графике можно показывать несколько параметров, сливая цвета (поддерживается прозрачность). Дополнительным достоинством *hardware-monitor* является то, что, сколько бы апплетов вы ни положили на панель, обслуживать их будет всего один процесс, а он потребляет крайне мало процессорного времени, как и полагается монитору. Страница апплета: www.cs.aau.dk/~olau/hardware-monitor.



Содержание

CHIP SPECIAL

LINUX

ЗИМА/2005



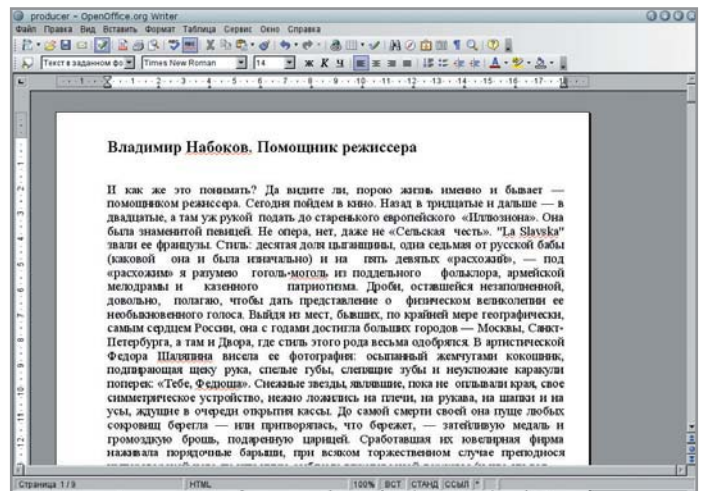
OpenOffice.org 2.0

На сегодняшний день для Linux существует только один офисный пакет, способный составить конкуренцию продукту Microsoft Office. Этим пакетом является OpenOffice.org от компании Sun Microsystems. Распространяется он бесплатно по лицензии LGPL. В состав этого продукта входят следующие программы:

- Writer — инструмент для создания и редактирования текстовых документов, аналог MS Word;
- Calc — программа для работы с электронными таблицами, аналог MS Excel;
- Impress — утилита для создания презентаций, аналог MS PowerPoint;
- Math — неплохое средство для построения всевозможных формул;
- Draw — простой инструмент для рисования примитивной графики;

- Base — программа для управления базами данных, аналог MS Access. OpenOffice.org полностью поддерживает форматы документов MS Office (DOC, XLS и т. д.). В версии 2.0 появилась поддержка формата OASIS Open Document, принятого в качестве стандарта электронных документов уже во многих странах мира. На диске, прилагающемся к этому номеру журнала, вы найдете дистрибутив OpenOffice

2.0, полностью русифицированный специалистами фирмы «ИнфраРесурс» при участии компании ALT Linux.
<http://openoffice.org>



WINE 0.9.1 Beta

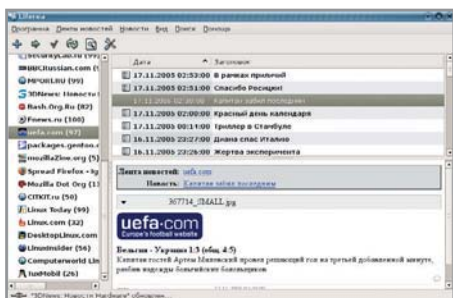
WINE (WINE Is Not an Emulator — «WINE не является эмулятором») — это проект по созданию альтернативной реализации Win32 API для Unix-систем. WINE позволяет запускать и с комфортом работать во многих Windows-приложениях, включая MS Office, Adobe Photoshop, Dreamweaver MX и IE 6. Любители игр тоже не остались обделен-

ными вниманием. Под WINE можно запустить такие популярные игры как Diablo 2 и World of Warcraft. Входящий в пакет набор утилит WineTools поможет легко настроить программу. Совсем недавно, после 12 лет разработки, проект получил статус beta. Чем не повод включить WINE в состав нашего диска?
<http://winehq.com>

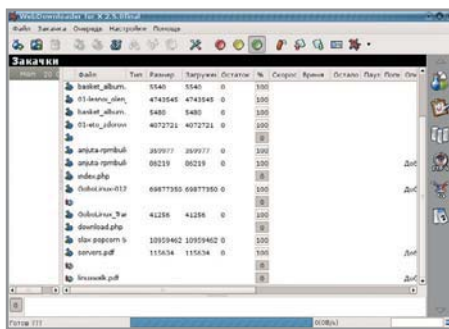
Liferea 1.0 rc3

Каждый день мы просматриваем десятки сайтов, чтобы прочитать свежие новости. Обычно это занимает довольно много времени, и не всегда мы находим нечто интересное для себя. С появлением новостных лент этот процесс стал гораздо проще и удобнее, а как следствие — и быстрее. Liferea — программа для чтения новостных лент. Она совместима практически со всеми существующими на сегодняшний день стандартами трансляции, поддерживает фильтрацию, поиск, сжатие лент и подкастинг. Радует и то, что Liferea умеет прятаться в системный трей и напоминает о появлении свежих новостей с помощью всплывающего окошка.

<http://liferea.sourceforge.net>



Downloader for X 2.5.6



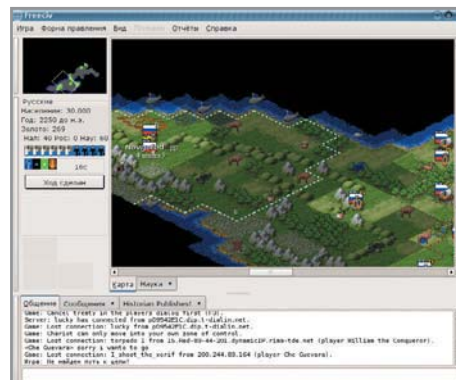
Менеджер загрузок для Linux существует не так много, в основном это графические реализации консольных fetch, wget и curl. Среди них внимания, безусловно, заслуживает программа Максима Кошелева — Downloader for X. Этот менеджер поддерживает докачку данных, загрузку по расписанию, лимитирование скорости и очереди файлов, а также их многопоточную загрузку. Присутствует поиск по крупным российским и международным FTP-серверам. Но все же программа имеет недостаток — неумение сворачиваться в системный трей. Надеемся, в следующих версиях этот недочет обязательно будет устранен.

www.krasu.ru/soft/chuchelo

Freeciv 2.0.7

Эту пошаговую стратегическую игру можно смело назвать копией знаменитой Colonization от Microprose. Пусть у Freeciv не столь выдающаяся графика, зато качество игрового процесса на высоте. Вам отводится роль правителя одной из ста цивилизаций. Цель — получить господство над миром. В вашем подчинении будут находиться около 50 разновидностей войск — от обычной пехоты до ракет с ядерными боеголовками. Играть можно либо по Сети на официальном сервере pubserver.freeciv.org, либо создать свой собственный.

www.freeciv.org



Beep Media Player 0.9.7



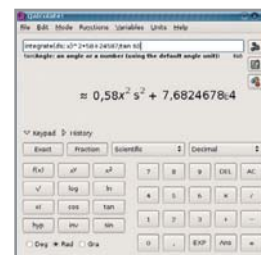
Все знают музыкальный проигрыватель XMMS. Отличная программа, но есть в ней и недостатки: неудобные настройки, проблемы с русификацией, к тому же интерфейс написан на GTK первой версии. Сообщество Open Source ответило на это появлением программы Beep Media Player. BMP — это тот же XMMS, но написанный с нуля и уже с использованием GTK2, что решило многие проблемы — в частности, с русификацией. Более удобными стали и настройки. Beep Media Player полностью совместим со скинами и плагинами для XMMS, а также со скинами и настройками эквалайзера плеера Winamp. Программа воспроизводит все популярные на сегодняшний день форматы (MP3, WAV, OGG) — естественно, при наличии соответствующих плагинов — и имеет эквалайзер с возможностью как ручной, так и автоматической подстройки звучания.

www.sosdg.org/~larne/w/BMP_Homepage

Qalculate! 0.9.0

Программа, необходимая любому студенту, уважающему себя и свое время. Qalculate! — это мощный калькулятор, который поддерживает GNOME и KDE, имеет богатый набор различных функций из алгебры, тригонометрии, логики, статистики, теории чисел. Поддерживаются операции с векторами, логарифмирование. Рассчитываются скорость, площадь, энергия и т. д. Калькулятор ведет историю вычислений, строит графики функций; к вашим услугам даже периодическая таблица химических элементов Менделеева. Кроме того, в состав Qalculate! входит конвертер валют с возможностью обновления курса по Интернету, а также конвертер величин (размерностей).

<http://qalculate.sourceforge.net>



ZoneMinder 1.21.4



Пакет программ для создания системы видеонаблюдения на базе операционной системы Linux. Программа реагирует на движение, тут же заносит событие в отчет (отсылает на FTP или отправляет e-mail). ZoneMinder работает при помощи интерфейса Video4Linux и поддерживает веб-камеры с интерфейсом USB, сетевые IP-камеры, а также камеры, подключенные через CCTV-карты. Программа обладает простым HTML-интерфейсом, благодаря которому можно осуществлять мониторинг с экрана обычного смартфона или КПК. Для работы ZoneMinder требуется MySQL и PHP, а также веб-сервер, например Apache. Способы применения данной программы зависят только от вашего воображения. Наблюдайте за родственниками и соседями, за дачей и офисом, но не забывайте об уважении к праву других на личную жизнь.

www.zoneminder.com

О диске

Установка программ

Программы на диске представлены в двух вариантах — исходные коды для последующего самостоятельного их компилирования и RPM-пакеты для дистрибутива Fedora Core 4 Linux.

RPM-пакеты

Мы выкладываем пакеты именно для Fedora Core, так как они являются более унифицированными и подходят для всех дистрибутивов, базирующихся на RPM. Установить пакет очень просто, достаточно в консоли с правами суперпользователя (root) набрать:

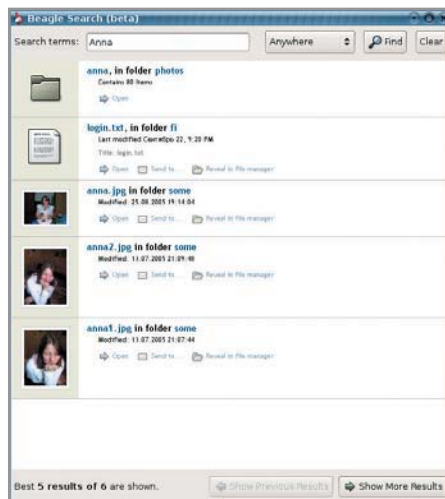
```
rpm -ivh <имя_пакета>
```

Исходные коды

Предназначены они для тех, кто пользуется такими дистрибутивами как Gentoo и Slackware. Инсталляция из исходников

Beagle 0.1.1

Пользоваться консольными командами locate и find не всегда удобно. Но теперь решение есть! Под крылом GNOME Foundation вырос проект Beagle — программа поиска по локальной файловой системе. Beagle индексирует картинки, музыку, веб-страницы, историю посещения браузеров, почту, тексты, изображения, PDF, DOC, аудиофайлы, логи мессенджеров и другое. Поисковик обрабатывает базу и выводит результаты поиска в виде списка со всеми подробностями о найденных документах. Можно установить дополнительный плагин для Firefox. <http://beagle-project.org>



distcc 2.18.3

Distcc — это программа, организующая параллельную распределенную компиляцию исходных кодов языков C, C++, Objective C и Objective C++ между компьютерами сети.

Использование distcc значительно сокращает время компиляции. Установка же программы по-детски проста; distcc не требует ни каких-либо предустановленных библиотек, ни синхронизации системных часов и даже не зависит от типа установленной операционной системы (поддерживаются: GNU/Linux, FreeBSD, NetBSD, Darwin, Solaris, HP-UX, IRIX, Cygwin и BSD/OS) и модели процессора. В комплекте с distcc поставляется небольшая программа distccmon, позволяющая наблюдать за процессом компиляции вашего кода на всех подключенных компьютерах. Distccmon работает как в графическом режиме, так и в текстовом. Программа особенно удобна при работе в дистрибутивах, требующих частой компиляции исходных кодов (Gentoo, Slackware, Linux From Scratch).

<http://distcc.samba.org>

Список программ

Desktop

Beagle 0.1.1, Dia 0.95, Ion 3-20051029, OpenOffice.org 2.0, WindowMaker 0.92.0

Мультимедиа

Beep Media Player 0.9.7, GQRadio 1.9.2, Graveman 0.3.12-4, Streamtuner 0.99.99, ZoneMinder 1.21.4

Интернет

Downloader for X 2.5.6, Firestarter 1.0.3, Licq 1.3.2, Liferea 1.0 rc3, Links 2.1 pre18, X-Chat 2.6.0

Разработка

Advanced Bash-Scripting HOWTO, distcc 2.18.3, NVU 1.0, RPM Builder plugin for Anjuta 0.1.3

Система

ATI Proprietary Linux Driver 8.18.8, Linux Kernel 2.6.14, Nmap 3.93, NVIDIA nForce Driver 1.0-0306, NVIDIA Display Driver 1.0-7676, Torsmo 0.18, WINE 0.9.1 beta

Игры

BilliardGL 1.75, Freeciv 2.0.7, Gens 2.12, SuperTux 0.1.3, ZSNES 1.42

Бонус

Aewan 1.0.01, Qalculate! 0.9.0

Сергей Яремчук

Вектор скорости

По быстрдействию с этой ракетой можно сравнить разве что Gentoo stage3. На ее фоне большинство дистрибутивов, ориентированных на конечного пользователя, выглядит медленно ползущими черепахами. Итак, знакомьтесь — VectorLinux!

В последнее время наметилась устойчивая тенденция — появляются версии дистрибутивов, направленные на конкретную целевую аудиторию. Многодисковые программные продукты в некоторых случаях оказываются лишними. Например, если пользователь просто желает познакомиться с новой системой, достаточно и LiveCD. Аналогично, если необходима система для использования на флеш-драйве, то, скорее всего, он предпочтет специализированный дистрибутив, нежели будет пытаться вставить в него Mandriva. С другой стороны, на сервере совсем не нужны графические оболочки и различные проигрыватели мультимедиа. К тому же очень жесткие требования к защите серверов на домашнем компьютере будут скорее мешать, так как пользователю в первую очередь важны удобство и простота. Поэтому появление направленных на конечного пользователя дистрибутивов — вполне закономерное развитие событий.

Популярные производители настольных версий Linux — SUSE, Mandriva, Fedora — пытаются разместить все приложения на трех-пяти дисках, и пользователю при установке приходится долго выбирать из длинного списка то, что ему действительно нужно. Создатели канадского дистрибутива VectorLinux (www.vectorlinux.com) хотели сделать его простым, небольшим и позволить конечному пользователю решать самому, что будет входить в его операционную систему. По их мнению, установка должна быть максимально простой, поэтому они решили ограничиться одним-единственным диском, на котором поместилась полноценная рабочая среда со всеми необходимыми библиотеками и приложениями.

VectorLinux основан на популярном дистрибутиве Slackware, который всегда отличался устойчивостью в работе, но пользователи, особенно начинающие, не очень его любят из-за сложности. А в VectorLinux дружелюбность как раз и стала основной задачей разработчиков. Кстати, дистрибутив способен нормально работать на компьютере с объемом оперативной памяти всего 32 Мбайт. Он лоялен к старому оборудованию, поддержка которого по умолчанию уже не включается в некоторые современные дистрибутивы, и отлично себя

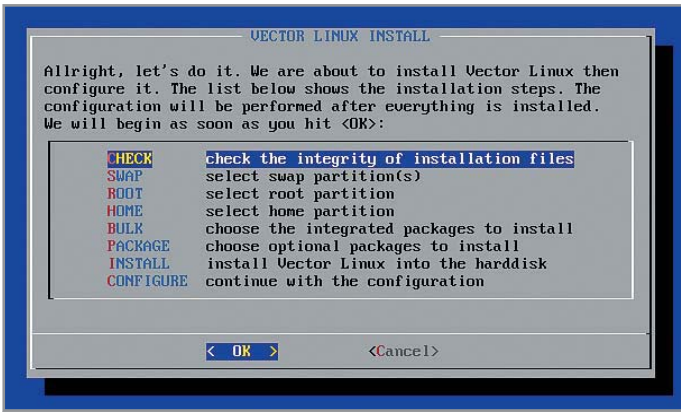
чувствует на самом современном ПК. Во второй половине лета 2005 года был анонсирован релиз 5.1 этого дистрибутива. О нем мы и расскажем вам более подробно.

Семейство VectorLinux

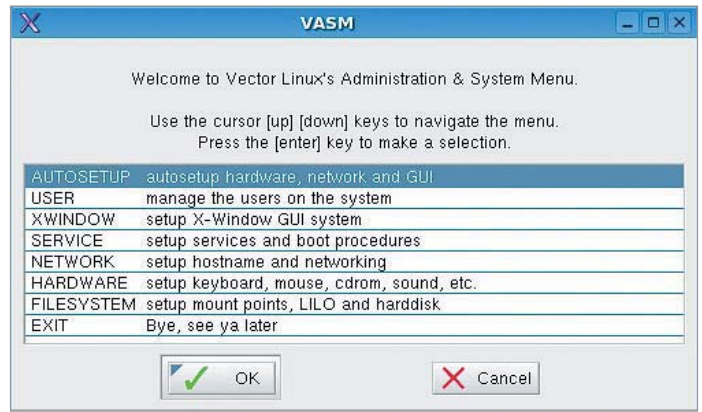
В настоящее время VectorLinux доступен в нескольких вариантах, каждый из которых предназначен для специфических целей. Базовым является VectorLinux Download (или Standard Edition). Его размер составляет всего 350 Мбайт, при этом он содержит минимально необходимую среду, включая легкие оконные менеджеры, мультимедиа, почтовые приложения и прочие необходимые программы. Главный девиз — скорость и легкость! Deluxe Edition включает также диск объемом 350 Мбайт с дополнительным софтом (в том числе KDE, GNOME, OpenOffice) и брошюру, в которой содержится информация об установке. Эта версия предназначена больше для опытных пользователей. Самый популярный вариант — VectorLinux SOHO (Small Office/Home Office), — как видно из названия, создан для использования в настольных компьютерах. И поэтому он включает все необходимые приложения — KDE, OpenOffice, Java, GIMP, Xsane, CUPS, финансовые утилиты и прочее. Главная задача версии SOHO — создать удобную рабочую среду с хорошей интеграцией приложений, работающую быстро, простую в установке и не занимающую много места. Последнее реализуется за счет тщательного отбора утилит, в которых наибольшая функциональность сочетается с удобством и уменьшением количества дублирующих приложений.

Плюс ко всему есть два неофициальных дистрибутива. Нестабильной веткой Dynamite Edition пользуются в основном разработчики — здесь отрабатываются нововведения, которые, возможно, появятся в следующих релизах VectorLinux. И вполне вероятно, он является самым дружелюбным из всех Slackware-дистрибутивов на сегодняшний день.

Еще существует VectorLinux GX, предназначенный для школ и поэтому содержащий специфические приложения для обучения, которые обычно не включаются в состав дистрибутивов.



Процесс установки консоли не менее удобен, чем графический интерфейс других дистрибутивов



VASM — мощное средство настройки системы с графическим и текстовым интерфейсом

Не так давно проект обзавелся более-менее понятной документацией (доступной в том числе и на диске, правда, на английском языке), позволяющей разобраться с особенностями работы. Единственный недостаток VectorLinux, который в наших условиях является критичным, — полное отсутствие локализации: в системе доступен только английский интерфейс. Русификацией придется заниматься самостоятельно, устанавливая необходимые пакеты Slackware.

Инсталляция

Устанавливать VectorLinux можно несколькими способами, в том числе и используя скачанный ISO-образ, без записи его на диск. Начинающие пользователи не очень любят Slackware из-за отсутствия графической программы-инсталлятора. В VectorLinux процесс установки также проходит в консоли, но создатели проделали хорошую работу и упростили то, что упростить, казалось, уже невозможно. Единственный момент, который может напугать неопытного пользователя, — подготовка разделов для установки, но, немного почитав документацию, можно справиться и с этим. Для создания разделов используются GNUParted (в пункте меню «Resize») и cfdisk, но теперь в ходе установки позволено выбрать две точки монтирования (корневой каталог и /home), которые можно отформатировать под reiserfs, ext2 или ext3. Все входящие в состав дистрибутива приложения после установки занимают 950 Мбайт. Кстати, в более ранних версиях установка заключалась в распаковке архива на раздел диска, и единственное действие, которое необходимо было совершить пользователю, — это создать его при помощи cfdisk, отформатировать под необходимую файловую систему и указать на него программе. Если что-то не выходит, то, выбрав соответствующий пункт меню, можно выйти в оболочку (причем это будет полноценная оболочка с tc и прочими удобствами), где будут доступны fdisk и другие утилиты форматирования и проверки указанных выше файловых систем. Из меню можно легко восстановить и загрузчик, для чего достаточно просто выбрать нужный пункт.

После копирования файлов установка, собственно, заканчивается, и система предлагает извлечь диск и перезагрузиться. Весь процесс на компьютере с процессором 1,1 Гц занимает в общей сложности 25 минут.

Первая загрузка

Система грузится достаточно быстро по сравнению с Mandriva или SUSE. Не зря VectorLinux называют Slackware на ракетном топливе, он по праву считается самым быстрым дистрибутивом. Теперь наступает этап постинсталляционной настройки, в процессе которой необходимо указать, куда устанавливать загрузчик, а также настроить X.Org-6.8.2, временной пояс, сеть, звук, задать пароль root и создать дополнительных пользователей. При выборе графического режима сразу же загружается X Window. Можно все настроить сразу, но лучше это сделать позднее — загрузившись и воспользовавшись утилитой VASM (Vector Linux System Menu).

Оборудование определилось нормально. Это в принципе можно считать визитной карточкой дистрибутива, корректно работающего в том числе и на старых машинах. Автору постоянно не везло в VectorLinux с мышью. В версии 3.2, которая стояла на ноутбуке почти год, внешняя USB-мышь исправно определилась, а тачпад пришлось настраивать, вручную изменяя параметр «Protocol» в конфигурационном файле «иксов» с IMPS/2 на PS/2. В версии 5.0 после загрузки не работали уже оба устройства позиционирования курсора. В этот раз все обошлось без проблем. Не советуем сейчас выбирать русскую раскладку клавиатуры для X Window, иначе после перезагрузки, которая последует по окончании настройки, вы не попадете в систему, потому что не сможете изменить раскладку. Все это лучше будет сделать потом, вручную подправив файл /etc/X11/xorg.conf. Новые USB-устройства монтируются автоматически, по крайней мере цифровая камера, флеш-накопитель и Bluetooth-донгл были распознаны без проблем. Для владельцев ноутбуков нелишней окажется поддержка PCMCIA и APM в ядре 2.6.12, будут работать также SATA и SCSI.

Работа в VectorLinux

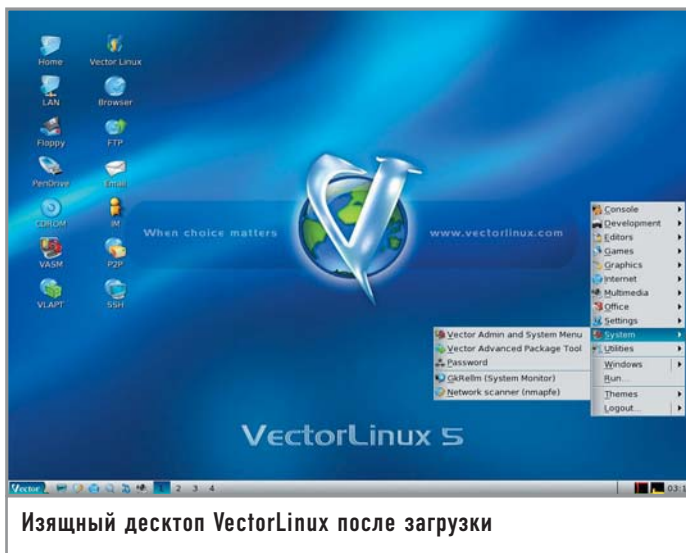
После перезагрузки пользователя встретит несколько видоизмененный xdm с пляшущими пингвинами. Для работы доступны Xfce, Fluxbox и IceWM. В версии Deluxe вы найдете KDE 3.4.1 и Freerock GNOME 2.10.2. Интересно, что IceWM можно загрузить, так сказать, в чистом виде, а можно и с включенным ROX-Fileer (<http://rox.sourceforge.net>), который существенно расширит возможности IceWM, добавляя, например,

иконки на рабочий стол. Обычно у пользователей возникает проблема при монтировании/размонтировании сменных носителей. В VectorLinux работа с такими устройствами реализована на должном уровне, а при помощи контекстного меню, вызываемого правой кнопкой мыши, можно всегда отмонтировать устройство даже в том случае, когда оно занято. Успешно получается проделать это и с CD-ROM.

Кроме стандартных утилит для настройки имеется и свой конек — VASM, который появился в VectorLinux еще в версии 2.0 и постоянно модернизировался с учетом предложений пользователей. А начиная с версии 3.2 SOHO, VASM получил графический интерфейс (ранее он был доступен только из консоли), позже некоторые пункты стали доступны в виде отдельных команд. При вызове проверяется значение переменной \$DISPLAY, и в зависимости от того, в каком режиме (консоль или X Window) его запускают, используется соответствующий интерфейс. VASM позволяет отключить, определить и автоматически настроить новое оборудование, берет на себя менеджмент пользователей (добавление, удаление, изменение пароля, быстрые команды /sbin/vuser, /sbin/vuseradd, /sbin/vuserdel и /sbin/vpasswd), конфигурацию X Window, уровень запуска системы, задает сервисы, запускаемые при загрузке, настраивает сеть, принтеры, межсетевой экран, устанавливает загрузчик, производит резервное копирование файлов и т. д. Некоторые функции настройки, предоставляемые скриптом, доступны только пользователю root, а часть параметров (настройка X-сервера, обнаружение нового оборудования) работает только в консоли. Как видите, разработчики собрали в одной простой утилите все возможные настройки, с которыми обычно приходится иметь дело пользователю. Хотя работа с ним не всегда интуитивна и иногда приходится экспериментировать, все же VASM неплохо справляется с задачей базовой настройки системы.

Естественно, никуда не делись fdisk, rppsetup, wvdial, gkdi-al, GuardDog (настройка межсетевого экрана) и прочие утилиты, предназначенные для настройки отдельных параметров и сервисов системы.

Установку приложений можно производить и при помощи традиционных утилит от Slackware вроде installpkg и upgradepkg.



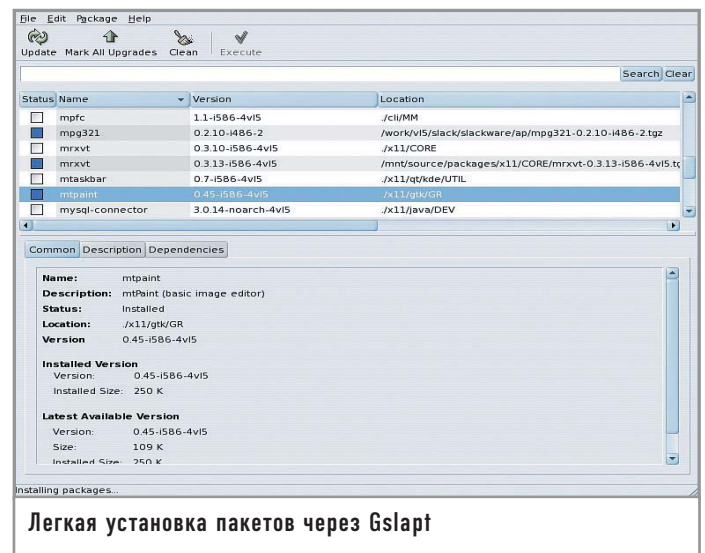
Изящный десктоп VectorLinux после загрузки

Но теперь, по задумке разработчиков, задача по управлению пакетами возложена на утилиту Gslapt, которая базируется на slapt-get. Ранее Gslapt вызывала некоторые сомнения, так как не очень дружила с зависимостями и, не признавая sudo, запускалась, только когда пользователь регистрировался в системе как root. С VLAPT, который был в версии дистрибутива 5.0, таких проблем не случалось. Также в меню файлового менеджера Midnight Commander имеются дополнительные пункты, позволяющие установить, распаковать и деинсталлировать пакеты. Причем работа с RPM-, DEB- и TGZ-пакетами реализована так, чтобы пользователь вообще не задумывался о какой-либо разнице между ними.

Поставляемые по умолчанию в дистрибутиве утилиты действительно охватывают весь спектр потребностей среднестатистического пользователя. Браузером по умолчанию является Firefox 1.0.4, но в наличии есть и Dillo, как альтернативу можно использовать Konqueror. С Firefox все-таки возникли проблемы, так как не все плагины устанавливались, и его пришлось сразу же обновлять. Для работы (если, конечно, можно назвать это работой) в IRC в состав включены Gaim 1.3.0 и Xchat. Документы Word открываются в Abiword 2.2.5, музыка и фильмы — в XMMS 1.2.10, MPlayer 1.0preba. Ну и, конечно же, не обошлось без GIMP 2.2.1, Xview и Gqview. А OpenOffice, а также Flash, Acrobat Reader, vim, Apache, MySQL, Emacs, Samba, Screen и многие другие полезные приложения присутствуют в версии Deluxe. Серьезное внимание уделяется защите. В состав дистрибутива входит система обнаружения атак portsentry, openssl и openssh.

Не имеет смысла перечислять все пакеты — заметим только, что в VectorLinux нет десятка текстовых редакторов, проигрывателей и почтовых клиентов. Пользователю предоставлен необходимый, но удобный в работе минимум.

Пора подвести итоги. Дистрибутив создан с любовью. Несомненно, VectorLinux придется по вкусу пользователю с некоторым уровнем подготовки и, возможно, новичку, которого не испугает отсутствие графической программы установки, а также постинсталляционная возня с локализацией. К тому же дополнительным преимуществом VectorLinux можно считать возможность работы на устаревшем оборудовании. |



Легкая установка пакетов через Gslapt

InitNG — начни сначала

С InitNG мне довелось познакомиться еще в версии 0.06 — одной из первых публично доступных. Тогда новость о нем облетела все ленты. И я не удержался от того, чтобы не поэкспериментировать. И не зря: тогда я впервые увидел, что замученный жизнью Celeron 480 со 128 Мбайт памяти, оснащенный к тому же множеством сервисов, может запускать пингвина в полет быстрее, чем это делает стоящий рядышком Athlon 1700+ с 256 Мбайт.

Правда, с той версией пришлось изрядно повозиться, поскольку стабильно она работать никак не хотела и периодически норовила отказаться то от загрузки kdm, а то и других сервисов. Тем не менее старый добрый шаманский бубен меня выручил, и эта версия простояла на том самом компьютере очень долго, периодически выдавая сбои, но в целом обеспечивая достаточный уровень надежности работы.

Да, InitNG все еще является экспериментальным софтом, поэтому накладки в процессе его использования весьма вероятны. Позже я заменил старую версию на InitNG 0.3.x, поборолся с новыми глюками и отметил чуть более стабильное поведение системы. Однако по-настоящему беспроблемной стала только лишь последняя ветка InitNG версий 0.4.x.

| Init |

Что же такое InitNG и что он призван заменить? InitNG является альтернативой Init, который однозначно наличествует в вашем дистрибутиве, каким бы он ни был. Init — процесс номер один в системе, именно ему ядро передает управление после инициализации, и именно он запускает все, что вы от него хотите на этапе загрузки. К тому же Init является предком любого процесса, запущенного в вашей системе.

Так сложилось, что эта область свободных систем практически не развивалась. В GNU использовался аналог — SysVinit, и всем жилось, в общем-то, хорошо. Ровно до тех пор, пока не обнаружилось, что Windows XP загружается быстрее мно-

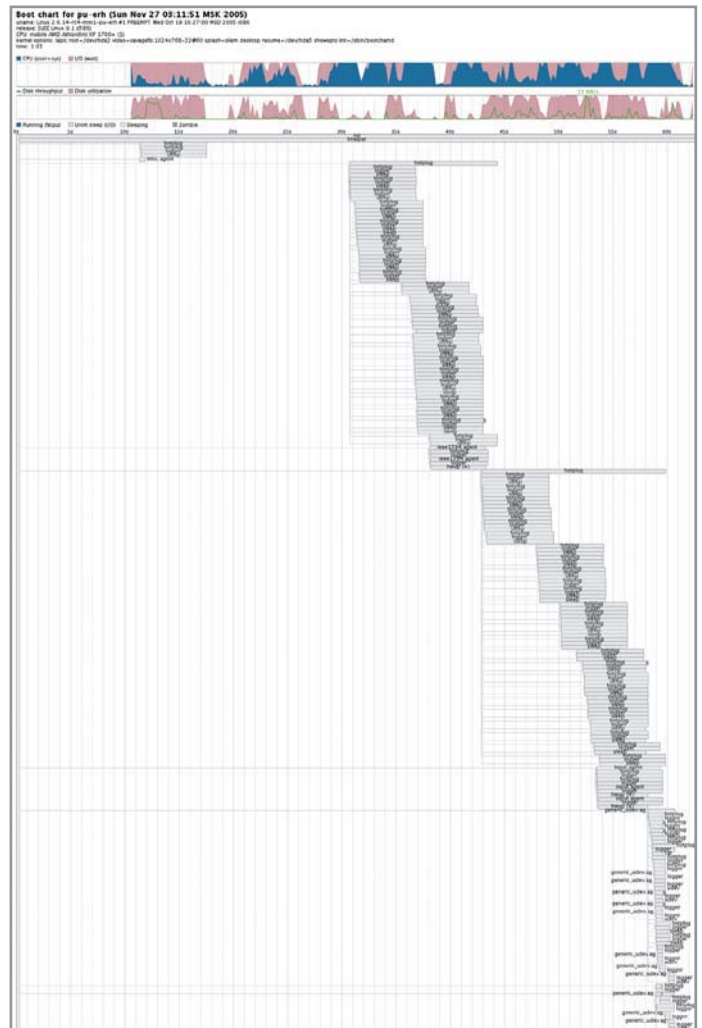
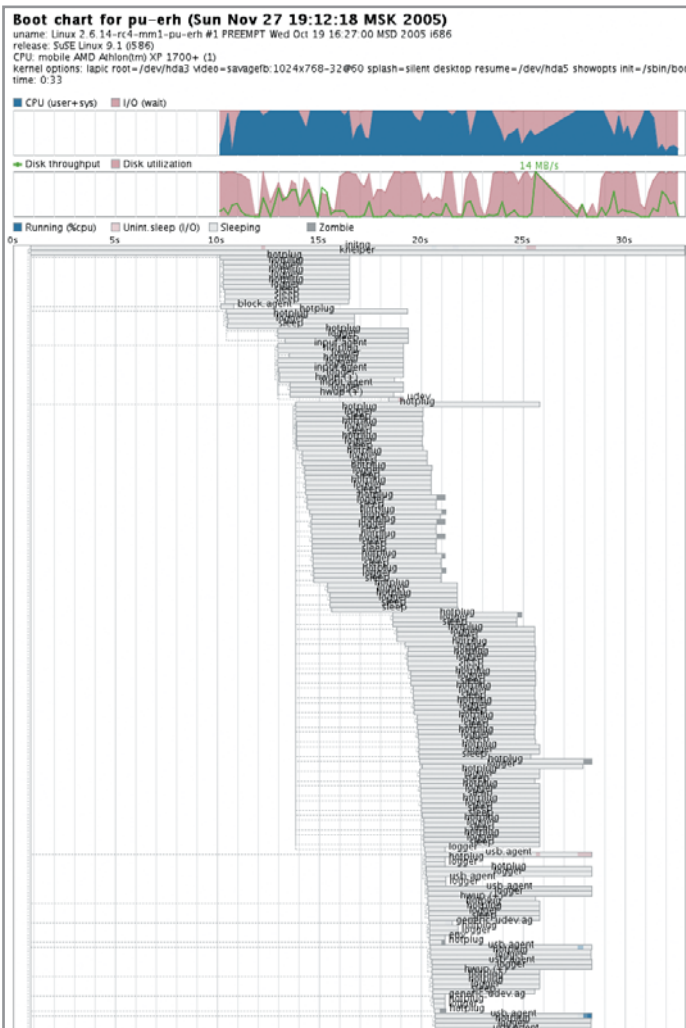
гочисленных пользовательских дистрибутивов GNU/Linux. Время загрузки явилось основным катализатором исследований в этой области, а InitNG стал одним из результатов этого трудоемкого процесса.

Естественно, проект не представляет интереса для серверов, поскольку в этой области машины не должны выключаться и перезагружаться вообще, а уж если и придется, то минута-другая погоды не делает. InitNG, равно как и многие из последних довольно значительных изменений в GNU/Linux, предназначен для настольных машин простых пользователей, которые не любят долго сидеть перед монитором в ожидании окончания процесса загрузки. Правда, пока он не используется как стандарт ни в одном дистрибутиве. Пользователям придется разминать пальчики в консоли и познавать таинства конфигурации своих систем.

| Проблемы Init |

В чем же проблема старого Init, почему ему потребовалась замена? Он плохо дружит с параллельным запуском сервисов, это факт. Весь процесс загрузки должен быть расписан в скриптах строго последовательно. Это плохо, так как большинство сервисов на самом деле не зависят друг от друга, но при этом они не могут запускаться параллельно из-за ограничений Init.

Как правило, все запускаемые сервисы что-то считывают с диска, а это всегда происходит очень медленно. Во время та-



Графики загрузки, полученные в bootchartd. InitNG обогнал Init почти в два раза

кого ввода/вывода процессор мог бы уделить внимание другому сервису, но старый Init этому не способствует. Аналогично происходит и с coldplug, который задает работу ядру по общению с устройствами, а это тоже весьма длительная операция, и хорошо бы в эти миллисекунды просто занять процессор чем-нибудь другим, не менее полезным.

Новации InitNG

InitNG решает описанную проблему путем параллельного запуска всех не зависящих друг от друга сервисов, что способствует равномерной загрузке процессора и периферии, в результате все сервисы вместе успевают загрузиться значительно быстрее.

InitNG ладно скроен и имеет модульную архитектуру, хотя это больше интересно программистам, которые захотят его расширить. Параллельность же запуска обеспечивается правильной системой скриптов, в которой порядок их выполнения рассчитывается на основе реальных зависимостей сервисов.

Надо отметить, что существующая сегодня система загрузки в Gentoo имеет схожую структуру (основывается на зависимостях) и даже позволяет параллельно запускать сервисы, однако старый Init ограничивает эти возможности, и реальный прирост скорости загрузки от этого весьма невелик. Новый InitNG полностью лишен подобных недостатков.

Установка и дистрибутивы

Надо признать, что наиболее комфортно InitNG чувствует себя в той среде, где он и вырос, а именно в Gentoo. Еще в очень ранних версиях для него был доступен ebuild, который теперь также входит в основной репозиторий Gentoo. Поэтому все, что от вас здесь потребуется, — это ввести emerge initng, а portage сделает все самостоятельно. Хотя минимальная конфигурация, наверное, все-таки потребуется.

Обладателям других дистрибутивов придется засучить рукава. Пользователям Debian и его многочисленных отпрысков (в том числе и Ubuntu) здесь повезло несколько больше, поскольку для этих систем существует специальный пакет Debian (<http://alioth.debian.org/projects/pkg-initng>), который упрощает процесс установки.

Остальным же придется производить сборку своими руками из исходников (<http://initng.thinktux.net>), что на самом деле не так уж и плохо. InitNG имеет достаточно умный Makefile, который пытается догадаться, какой дистрибутив у вас установлен (соответственно подстраивая скрипты запуска сервисов) и что за сервисы стартуют по умолчанию (соответственно составляя список запуска у себя).

Отмечу, что все тесты и примеры конфигураций основаны на InitNG версии 0.4.4. Проект очень бурно развивается, по-

этому, возможно, к моменту выхода журнала некоторая информация будет уже не совсем точной.

После установки необходимо добавить еще один пункт в меню загрузки вашей системы и в нем прописать ядру параметр `init=/sbin/initng`. В GRUB он напрямую помещается в конец строки «kernel (...)», а в LILO это означает добавление строчки `'append = «init=/sbin/initng»'` в желаемый пункт. Не стоит пока что делать эту конфигурацию загружаемой по умолчанию.

Далее необходимо попробовать перезагрузиться с InitNG и приготовиться к работе в чистом консольном режиме — ставлю пять к одному, что с первого раза X и [kx]dm у вас не запустятся. Равно как и многие другие сервисы. Не нужно бояться этих проблем — хотя вывод об ошибках может быть и пугающим, навредить системе вы вряд ли сможете, незапускающийся сервис не означает разрушенной конфигурации, достаточно перезагрузиться со старым Init — и все будет в порядке.

Притирка и доводка

К огромному сожалению, каждый дистрибутив считает чуть ли ни своим долгом использовать собственные загрузочные скрипты, а также расположение и формат конфигурационных файлов, скрипты-помощники и прочее. Но к этому можно относиться с большим энтузиазмом — настраивая InitNG, вы лучше узнаете свою систему, а любители LFS (Linux from Scratch) уже могут примерять InitNG к своим системам в качестве единственного Init. В конечном счете на это вряд ли уйдет так много времени (все зависит от ваших навыков работы с консолью).

Итак, пробуем перезагрузиться с InitNG. Если у вас все заработало, знайте — вы редкий везунчик. Скорее же всего, вы увидите, что система загрузилась не полностью, и вывод InitNG запнулся на уровне 70–80% загрузки, при этом первая консоль совершенно непригодна к использованию, а «иксы»... о них на время вообще лучше забыть.

Вспоминаем о существовании других терминалов и смело нажимаем «Alt+F2». По умолчанию InitNG запускает `agetty` на шести терминалах, но первый из них запускается только в случае абсолютно успешной загрузки стандартного уровня исполнения. Заходите сразу как `root` и не стесняйтесь.

Основная задача на данном этапе — выявить сервис или демон, который помешал полноценной загрузке. Для этого

Необходимая настройка

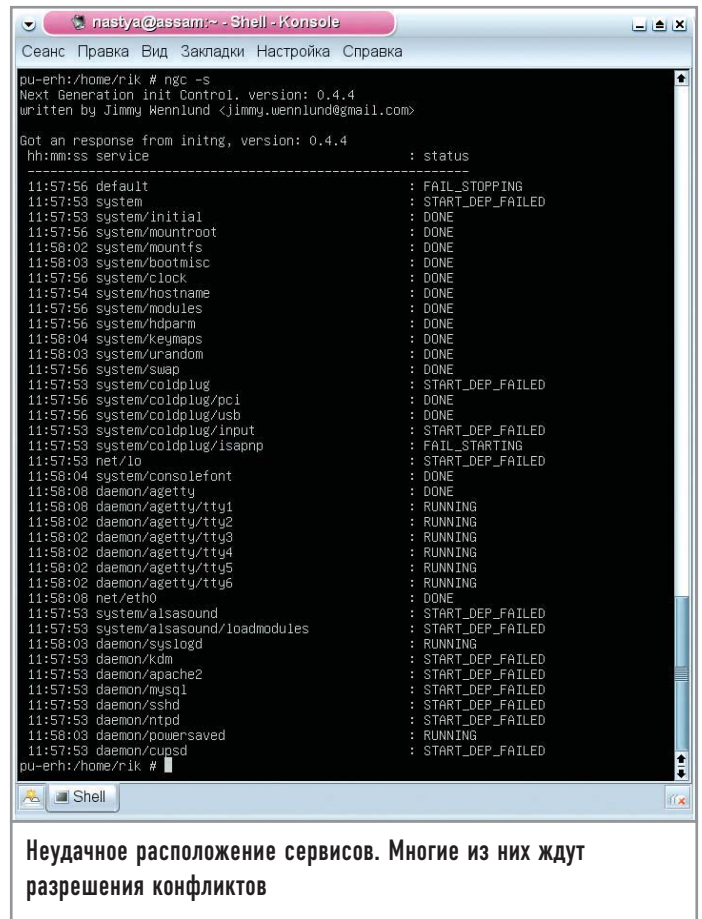
Новые демоны

К сожалению или к счастью, но в составе InitNG может и не оказаться скрипта запуска используемого вами демона. Однако теперь вы уже гуру в области его конфигурации, поэтому, думаю, у вас не возникнет проблем с добавлением новых скриптов. Вот в качестве

```

ве примера самописный скрипт запуска pure-ftpd:
daemon daemon/pure-ftpd {
    need = system/initial system/mountfs system/hostname net/lo
    require_network
    daemon = /usr/sbin/pure-ftpd
}

```



Неудачное расположение сервисов. Многие из них ждут разрешения конфликтов

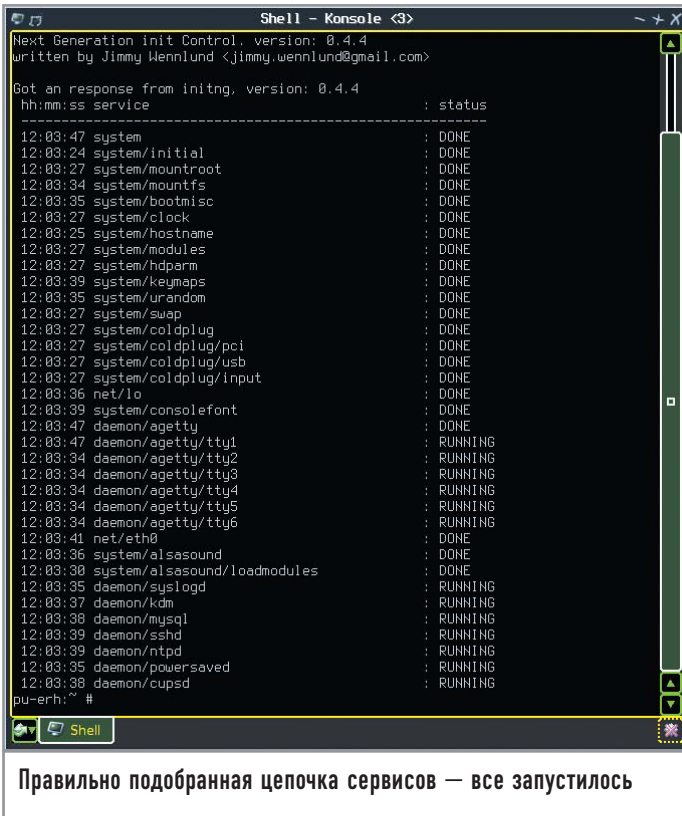
придется познакомиться с инструментом `ngc` (New Generation Init Control), который входит в состав InitNG. Он позволяет полностью управлять процессом загрузки с InitNG. В нашем случае потребуется `ngc -s`, который покажет статус загруженных сервисов/демонов.

Скорее всего, в выводе будет обнаружено несколько сервисов, ожидающих разрешения зависимостей (`START_DEP_FAILED`), а также один или два, которые не сумели запуститься (`FAIL_STARTING`). Ваша задача — избавиться от «плохих» сервисов и демонов.

Ненужные сервисы можно просто удалить из рассмотрения с помощью `ng-update del service/badservice`, а проще, на мой взгляд, сразу перейти к прямой правке конфигурационных файлов InitNG. Загрузочные скрипты и настройки InitNG располагаются в `/etc/initng`. Там же вы обнаружите несколько файлов `*.runlevel`, `*.i` и каталогов с файлами `*.i`.

Файлы `*.runlevel` описывают уровни исполнения InitNG, вы можете загрузить любой из них, передав ядру в качестве параметра необходимый уровень в виде `runlevel=level`. Файлы `*.i` описывают конкретные сервисы и демоны, запускаемые InitNG, а в каталоги они просто логически группируются для удобства.

Таким образом, вы сразу можете перейти к правке файлов `system.runlevel` или `default.runlevel` (нетрудно догадаться, что именно этот уровень запускается по умолчанию, однако он зависит от параметра `system`) с целью удаления сервисов и демонов, которые дали сбой. В этих файлах используется простейший список, одна строка — один сервис.



Однако какая же польза от InitNG, если мы вырежем все запускаемые сервисы и демоны? Они нам все-таки нужны, поэтому пора познакомиться с файлами *.i, описывающими все, что стартует при помощи InitNG. Стоит отметить, что сегодня также имеется возможность использовать XML для описания сервисов, но это сложнее и не так наглядно, как использование формата *.i. Оставим XML заботливым дистрибьюторам, которые добавят к ним красивые графические настройки; кто-нибудь из них, возможно, решится в следующем году сделать InitNG основным для своего дистрибутива.

| Сервисы |

Чем отличаются сервисы от демонов? Все очень просто, демоны — стандартные и привычные всем sshd или mysqld — запускаются и функционируют в течение всего времени работы системы. Сервисы же, в терминологии InitNG, являются сущностью, запускаемой единожды на этапе загрузки или выгрузки системы. Примеры — coldplug или настройка часов; все это выполняется один раз при загрузке, настраивает то, что надо, и завершается.

Примеры очень простых сервисов можно также увидеть в system/local.i, system/ntpdate.i и system/swap.i. Рассмотрим синтаксис на примере system/swap.i:

```

service system/swap {
    need = system/initial system/mountroot
    start = /sbin/swapon
    start_args = -a
#   start {
#   #!/bin/bash
#   /sbin/swapon -a
#   exit $?
    
```

```

#   }
}
    
```

Каждый сервис объявляется как «service имя_сервиса», и все его описание заключается в фигурные скобки. Оно состоит из ключевых слов и их параметров. Самое главное, что должно быть у сервиса, — это start и/или stop.

Посмотрите на различия между закомментированным форматом (а комментарии по традиции начинаются с «#») и рабочим (аналогично отличаются system/local.i и system/ntpdate.i): в них представлены два разных способа задания действий.

Первый использует ключевые слова start и start_args. Команда, выполняемая InitNG при запуске такого сервиса, формируется напрямую из них и тут же вызывается. В данном примере это /sbin/swapon -a.

Второй способ задания действий значительно более гибкий, поскольку все, что заключается в блоке start{} (или stop{}), является обычным shell-скриптом — со всеми вытекающими последствиями. Но такой метод работает медленнее (и именно поэтому в system/swap был закомментирован), так как для его запуска InitNG приходится разворачивать отдельный процесс интерпретатора командной строки и отдавать ему скрипт. Для простых сервисов, потребности которых можно описать в первом формате, он явно менее предпочтителен.

Простейший сервис может состоять только из одного блока start и будет успешно работать, однако сервисы могут зависеть от других сервисов, и мы переходим к наиболее интересной части InitNG — работе с зависимостями.

| Зависимости |

В простейшем виде используется ключевое слово need, в котором через пробелы вписывается все, что требуется запустить, до того как будет запущен описываемый сервис. В примере с system/swap это system/initial и system/mountroot; действительно, все сервисы зависят от успешного завершения работы скрипта начальной инициализации системы, и

Дополнительная информация

Полезные возможности ngs

После того как вы, надеюсь, разрешите все проблемы, связанные с настройкой InitNG, стоит поближе познакомиться с инструментом ngs. Продолжив это знакомство, например, с ngs -O. Эта нехитрая команда расскажет вам все об опциях, которые можно использовать в описаниях сервисов и демонов, поскольку на самом деле их несколько больше, чем было упомянуто ранее. Более того, они постоянно разви-

ваются, расширяются и изменяются, поскольку InitNG — все еще очень молодой продукт. Также сразу же стоит запомнить ngs -u и ngs -d, которые соответственно запускают и останавливают сервисы и демоны, передаваемые следующим параметром. Полезен и ngs -r, позволяющий сервисы перезапускать. Остальное уже не так интересно, но при желании доступно для изучения по ngs -h.

почти всем необходимо иметь возможность записывать информацию в корневой раздел.

Важно, что описываемые в `need` зависимости являются критическими для сервиса, и без удовлетворения всех зависимостей `InitNG` даже не будет пытаться запускать сервис.

Несколько другой подход представляет использование ключевого слова `use`. Описанные с его помощью зависимости позволяют задавать порядок запуска сервисов/демонов, если они описаны в заданном уровне запуска. То есть, в отличие от `need`, если вы не собираетесь запускать нечто из списка `use`, все будет в порядке.

Например, в `daemon/apache2` среди `use` числятся `daemon/sshd` и `daemon/mysql`. Это значит, что если вы запускаете на своей машине `sshd`, то `apache2` будет запущен после него. Однако если вы не запускаете `sshd`, то `apache2` будет успешно запущен и без него. Аналогично и с демоном `mysql`.

Дополнительные возможности

Одной из интересных зависимостей, вынесенной даже в отдельное ключевое слово, является также `require_network`. Как можно догадаться из названия, она используется теми сервисами и демонами, которые нуждаются в сетевом подключении. Локальное сетевое подключение `lo` эту зависимость не удовлетворяет.

Помимо этого также доступен `env`, в формате `env = VAR=something` можно передавать переменные окружения вызываемым сервисам и демонам. Таких определений может быть несколько для одного сервиса (соответственно, будут заданы несколько переменных окружения). Это и отличает `env` от других описателей.

Также можно задавать `nice`, `suid` и `sgid` — соответственно приоритет, пользователь и группа, от имени которых будет запущен процесс. На закуску имеются `stdout` и `stderr`, с помощью которых можно перенаправлять стандартные потоки ввода и вывода ошибок.

Демоны

Что касается различий сервисов и демонов, последние имеют дополнительные возможности и несколько иной формат задания процессов. С одной стороны, их можно описывать с помощью `start{}` и `stop{}` (в качестве примера смотрите `daemon/openvpn`), но `InitNG` имеет гораздо более интересные возможности.

Существует ключевое слово `daemon` — опять-таки в разных форматах. Пример использования короткого формата можно посмотреть в `daemon/apache2`: как видите, существует и `daemon_args` для передачи параметров демону. Пример формата `daemon{}` — в `daemon/acpid`.

Необходимо пояснить, что `InitNG` не просто запускает демоны, он за ними еще и приглядывает. По умолчанию считается, что если демон возвращает управление (стандартное поведение, например, `sshd`), значит он так или иначе был убит, а стало быть, `InitNG` попытается перезапустить его. Именно поэтому в скриптах для запуска демонов стоит использовать `exec` (как в примере с `acpid`).

Но можно поступить иначе. Для `sshd`, например, можно использовать параметр `-D`, и тогда все будет в порядке.

А можно найти и альтернативные способы. Пускай демон возвращает управление `InitNG`, но мы можем отслеживать его состояние через файл `pid`. Для этого и предназначена опция `pid_file`. Пример ее использования есть в `daemon/vixie-cron`; он возвращает управление в любом случае, но благодаря `pid`-файлу `InitNG` может убедиться в том, что он жив-здоров.

Все это позволяет обеспечить надежность функционирования демонов. В случае их внезапной смерти по тем или иным причинам они будут перезапущены (без необходимости применения дополнительных решений вроде `daemontools`). Однако, чтобы обеспечить подобную функциональность, стоит прописать в параметрах демона «`respawn=yes`», так как стандартное поведение от версии к версии здесь менялось (в версии 0.4.4 по умолчанию перезапуск отключен).

Прочее

Осталось только упомянуть возможность использования шаблонов имен, а также то, что имена файлов не обязаны совпадать с именами сервисов (хотя это удобно). Прекрасным примером последнего является файл `net/net.i`, где описаны `net/lo` и `net/*`, имеющие разные зависимости и последствия (`network_provider` в `net/*` как раз и удовлетворяет зависимость `require_network`). Шаблоны здесь позволяют единообразно работать с сетевыми подключениями любого вида; запускаемое в данный момент подключение передается в переменной `${NAME}`.

Настройка

Всей этой информации должно с лихвой хватить для осознанного изменения настроечных файлов `InitNG`, что в конечном счете приведет вас к желаемому результату.

В деле исправления настроек в первую очередь стоит обратить внимание на параметры, с которыми запускаются те или иные демоны/сервисы — убедиться в том, что запускаются именно те бинарники и что они лежат там, где предполагает это `InitNG`. Он действительно здорово подстраивается под окружающую обстановку (если не верите, сравните содержимое файлов `*.ii` и `*.i` каталога `initfiles` в дереве исходников `InitNG` после сборки), но все же есть свои пределы, и иногда он может ошибаться.

Например, созданный на `SUSE 9.1`, использованной в качестве тестовой площадки, скрипт `daemon/apache2` планировал вызвать `/usr/sbin/apache2 -D PHP5 -k start`, в то время как `PHP5` на этой системе не стоит, а файл `/usr/sbin/apache2` вообще отсутствует. Вместо этого в `SuSE 9.1` есть `/usr/sbin/apache2ctl`, которому необходимо передать единственный параметр `start`. Однако и этого маловато — `pid`-файл для `Apache 2` в `SUSE 9.1` виден как `/var/run/httpd2.pid`, а не `/var/run/apache2.pid`. После этих тривиальных изменений `Apache` стал запускаться без каких-либо проблем.

А вот на примере `mysqld` уже можно рассмотреть твичинг скриптов `InitNG`. Скрипт по умолчанию запускал `/usr/bin/mysqld_safe` и зависел от сетевого подключения (`require_network`), однако `mysqld_safe` всего лишь обеспечивает поддержку `mysqld` в запущенном состоянии и логирует

вание некоторой информации. Последнее вам может быть просто не нужно, а первое вполне может сделать сам InitNG. Поэтому можно смело вписать «daemon=/usr/sbin/mysqld», а в daemon_args прописать все желаемые параметры, после чего добавить respawn=yes и убрать require_network, если вы используете этот сервер только с локальной машины. MySQL после этого прекрасно запускается и работает, а killall mysqld продемонстрирует живучесть сервера — InitNG его моментально перезапустит. Замечу, что в отличие от mysqld_safe эта связка значительно надежнее, поскольку первый можно просто убить, а вот если убить Init, будет слишком заметно.

По поводу «иксов» надо заметить, что основные проблемы с ними сводятся к загрузке модулей. Необходимо убедиться в том, что к моменту запуска [gkx]dm у вас будет загружено все, что касается устройств ввода/вывода. Например, драйвер графического планшета вместе с evdev или, не дай бог, проприетарный драйвер NVIDIA. Если с этим возникают какие-то проблемы, можно прописать принудительную последовательную загрузку необходимых модулей в отдельный (или существующий, хотя это не лучший вариант) сервис, от которого будет зависеть запуск [gkx]dm.

Все это потребует от вас определенных знаний о своей системе и ее настройках, используемых демонах, утилитах, скриптах и, соответственно, их настройках. Отличный подсказчик — /etc/init.d, где расположены «эталонные» скрипты

запуска всего, что нужно вашей системе (вы же работали на ней до этого, не так ли?). Пользуйтесь ими как ценными источниками информации.

Хочется отметить, что на сегодня у InitNG есть еще одно огромное преимущество перед стандартным init с любым стилем загрузки — он очень прост, и у вас есть прекрасная возможность проделать персональную конфигурацию с минимумом усилий, которая будет очень быстро запускаться и поддерживать работу необходимых вам демонов.

Как видите, ничего сверхсложного.

И о результатах

Так ради чего же боролись? Предлагаю взглянуть на красивые графики загрузки, полученные с помощью bootchartd (www.bootchartd.org). Результат очевиден — время загрузки уменьшилось практически вдвое. При этом еще остался резерв для оптимизации, поскольку из скриптов можно убрать как минимум все специфичные для Gentoo проверки и настройки (если, конечно, вы не используете именно Gentoo), а также вообще обойтись без использования некоторых конфигурационных файлов, вписав желаемое прямо в скрипты.

Полезно будет прогнать на своей машине bootchartd и выявить узкие места, которые тормозят загрузку системы, после чего заняться их устранением. Как говаривал Ричард Столлмен, «счастливого хака!»

Самостоятельная работа

Еще быстрее?

Чтобы добиться еще более высокой скорости загрузки, можно переписать скрипты InitNG для своей конфигурации. Если это дополнить перекомпиляцией ядра и отказом от coldplug (все ведь уже в ядре?), то можно сократить время загрузки еще в полтора-два раза — взгляните на еще одну картину загрузки системы. Машина по-прежнему та же, сервисы по-прежнему те же, однако время — уже всего лишь 18 секунд! Надо отметить, что за чертой двадцати секунд результаты загрузок довольно сильно варьируются (минимум, которого удавалось достичь на тестовой машине — 17 секунд). Например, задержавшийся ответ сервера на DHCP-запрос может запросто изменить время загрузки на несколько секунд, причём еще вопрос — в какую сторо-

ну, поскольку bootchart считает окончанием загрузки полноценный графический логин. Кстати, bootchart в такой конфигурации уже тоже начинает мешать сервисам, и сам успевает запустить мониторинг процессора/диска только тогда, когда половина дела уже сделана. Также становится очевидным то, что дальнейшее сокращение времени зависит в первую очередь от винчестера и файловой системы, поскольку узким местом становится ввод/вывод — он нужен всем стартовым демонам, и параллельный запуск здесь может даже мешать. С чего начать хакинг скриптов InitNG? В первую очередь об этом расскажет bootchartd, но вот некоторые общие рекомендации:

- ▶ Начните с оптимизации самых первых скриптов, запускающихся при старте, их завершения ждут все.
- ▶ Трезво оценивайте, действительно ли сервис не может жить без сети.
- ▶ Разберитесь, нужно ли ему дожидаться, когда смонтируются все файловые системы кроме корневой.
- ▶ Перейдите на журналируемые ФС и избавьте себя от mountroot.
- ▶ Не стесняйтесь изучать скрипты-обертки для запуска разных сервисов и скрипты-настройки сети вашего дистрибутива. Затем подумайте, нужно ли вам все это, и не проще ли написать их самим для своей системы и конфигурации?

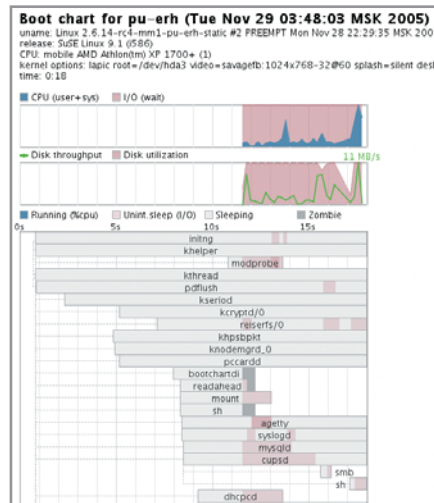


График оптимизированной загрузки

На страже системы

В прошлом выпуске журнала рассказывалось о системах управления доступом в общем, а кроме того, была подробно рассмотрена одна из самых популярных систем — Grsecurity. В этот раз мы поговорим о не менее интересном программном продукте, который называется LIDS (Linux Intrusion Detection System).

Изначально LIDS задумывалась как простая система обнаружения вторжения, но с годами благодаря усилиям разработчиков ей удалось вырасти в комплексный механизм обеспечения безопасности Linux-машины. Сейчас можно с уверенностью заявить, что LIDS уже достигла уровня Grsecurity, а в кое в чем даже превзошла ее. Преимуществом второй версии LIDS является поддержка LSM (загружаемых модулей безопасности), которой не было в предыдущих версиях, и это обстоятельство делало установку системы и работу с ней весьма затруднительным мероприятием.

Установка

LIDS реализована как патч ядра и набор пользовательских утилит, загрузить которые можно с сайта, расположенного по адресу www.lids.org/download.html. В отличие от других проектов, патчи которых подходят только к какой-либо конкретной версии ядра, патчи LIDS подходят для всех версий 2.4.x и 2.6.x, поэтому данная система способна работать с любым ядром.

Конфигурирование ядра

Распакуйте архив с LIDS, перейдите в каталог ядра и примените следующий патч:

```
# cd /usr/src
# wget http://www.lids.org/download/v2.6/2.6.7/lids-2.2.0rc3-
```

2.6.7.tar.gz

```
# tar -zxvf lids-2.2.0rc3-2.6.7.tar.gz
```

```
# cd linux-2.6.7
```

```
# patch -p1 < /usr/src/lids-2.2.0rc3-2.6.7/lids-2.2.0rc3-2.6.7.patch
```

После этого можно приступать к конфигурации ядра. LIDS требует включить поддержку алгоритма SHA256. Соответствующую опцию можно найти в меню «Cryptography Options/Cryptography API». Опции конфигурации LIDS находятся в разделе «Security Options». Желательно отключить SELinux и Capabilities.

Рассмотрим опции LIDS:

Attempt Not to Flood Logs (CONFIG_LIDS_NO_LOOD_LOG).

Данная опция ограничивает частоту протоколирования идентичных сообщений.

Allow Switching the LFS and States (CONFIG_LIDS_ALLOW_SWITCH). LFS (LIDS-free session, о ней мы поговорим чуть позже) позволяет администратору выполнять команды без каких-либо ограничений со стороны LIDS. Это довольно полезно, но может стать источником для атаки вашей системы. Рекомендуем активировать эту опцию на первые несколько дней — пока вы экспериментируете с LIDS, а потом, когда все будет настроено, можно ее спокойно отключить.

Allow Switch Off the Linux Free Session (CONFIG_LIDS_ALLOW_LFS). Позволяет отключить LIDS во время выполнения системы. Так безопаснее.

Restrict Mode Switching to Special Terminals (CONFIG_LIDS_RESTRICT_MODE_SWITCH). Позволяет задавать терминалы, с которых разрешается LFS. Они делятся на три класса: консоль (console), последовательная консоль (serial console) и PTY. Третий — наиболее опасный, поскольку дает возможность злоумышленнику удаленно запустить LFS. Выберите только первый класс, позволяющий запускать LFS лишь пользователям, физически работающим с машиной.

| lidstools |

После компиляции ядра можно установить пакет lidstools. Сценарию ./configure необходимо явно передать каталог ядра:

```
$ tar -zxvf lidstools-2.2.5
$ cd lidstools-2.2.5rc1.tar.gz
$ ./configure KERNEL_DIR=/usr/src/linux-2.6.7
$ make
$ su
# make install
```

При установке программы (make install) вас попросят ввести пароль для администрирования LIDS — он не должен совпадать с паролем root!

Чтобы использовать новое ядро, нужно перезагрузить систему. Если необходимо отключить LIDS, перед загрузкой системы передайте ядру параметры security=0.

| Администрирование LIDS |

Поскольку конфигурирование ядра мы уже рассмотрели, займемся конфигурированием пользовательского уровня. Как и Grsecurity, LIDS позволяет определить, каким образом файлы и процессы будут взаимодействовать между собой в системе. Кроме этого LIDS имеет две очень полезные функции — LFS и «опечатывание» ядра (sealing the kernel).

Загружаемые модули очень полезны, так как позволяют добавлять код во время выполнения ядра без его перекомпиляции. С другой стороны, злоумышленник может добавить в ядро свои собственные модули, а для нас это очень нежелательно. Конечно, самое лучшее — включить весь необходимый код в состав ядра и вообще отключить поддержку загружаемых модулей, но вряд ли это решение окажется для вас удобным.

LIDS предлагает концепцию «опечатывания» ядра. Если ядро «опечатано», никто не может загрузить или выгрузить модуль. Сделать это можно с помощью команды lidsadm -I. Ее необходимо поместить в сценарии загрузки системы, только при этом убедитесь, чтобы еще до начала выполнения этой команды все необходимые модули были уже загружены. «Опечатывание» ядра также предусматривает некоторые ограничения набора возможностей, но об этом мы поговорим чуть позже.

| LFS (LIDS-free Sessions) — сессии без LIDS |

LFS — это всего лишь оболочка, на выполнение команд которой не накладываются ограничения LIDS, что позволяет адми-

нистратору работать в системе как обычно, без выключения основной системы безопасности, ведь когда LIDS включена, ограничения накладываются даже на пользователя root. Однако LFS потенциально опасна: если злоумышленнику удастся завладеть доступом к ней, то он получит полный контроль над системой — вплоть до отключения LIDS.

Доступ к LFS регулируется установленным ранее паролем. Дополнительно при конфигурации ядра можно указать терминалы, с которых доступ к LFS будет открыт (очень хорошее решение).

Главное назначение LFS — разрешить администратору редактировать файлы в каталоге /etc/lids, который недоступен во время работы LIDS даже пользователю root. В этом каталоге находятся следующие файлы:

- ▶ lids.cap — ограниченный набор возможностей;
- ▶ lids.conf — ACL (будет рассмотрен позже);
- ▶ lids.pw — пароль администратора LIDS;
- ▶ lids.ini — начальные конфигурационные значения.

| lidsadm |

Администрирование LIDS выполняется программой lidsadm. Рассмотрим подробнее ее опции:

- ▶ -P — зашифровать пароль LIDS, например lidsadm -P mypassword;
- ▶ -S — изменить аспект защиты LIDS;
- ▶ -I — «опечатать» ядро; для этой опции не нужен пароль;
- ▶ -V — просмотр состояния системы;
- ▶ -h — вывести краткую справку;
- ▶ -v — вывести версию lidsadm.

Опция -S используется вместе с одним из следующих флагов, предваряемых либо знаком «+» — включить, либо «-» — выключить:

- ▶ LIDS_GLOBAL — включить/выключить LIDS глобально;
- ▶ RELOAD_CONF — перезагрузить файл lids.conf и обновить список защищенных инодов (об этом мы поговорим чуть позже);
- ▶ LIDS — включить/выключить LIDS локально, то есть создать LFS, которая будет применена только к текущей оболочке;
- ▶ ACL_DISCOVERY — используется для отладки; когда включена, нарушения правил не запрещаются;
- ▶ SHUTDOWN — переключается в состояние shutwodn.

Например, чтобы войти в LFS, выполните нижеприведенную команду:

```
# lidsadm -S — -LIDS
```

Чтобы выключить защиту LIDS, сделайте следующее:

```
# lidsadm -S — -LIDS_GLOBAL
```

| ACL файлов и возможностей |

ACL используется для управления доступом к различным объектам, например файлам или ресурсам (но в отличие от Grsecurity в LIDS нет ролей). В системе LIDS есть ACL двух типов: ACL файлов (контролирует доступ к файлам и каталогам) и ACL возможностей (регулирует возможности исполнимых файлов).

| ACL файлов |

LIDS определяет четыре режима для объектов:

- ▶ DENY. Доступ к файлу запрещен. При обращении к нему

приложения (например, ls или cat) будет получено сообщение об ошибке «No such file or directory (ENOET)» — «Нет такого файла или каталога». Внешне это будет выглядеть так, как будто файла вообще не существует.

► **READ.** Объект может быть открыт в режиме «только для чтения», запись запрещена.

► **APPEND.** Объект может быть открыт для чтения или добавления информации. Данный режим удобно использовать для файлов журналов.

► **WRITE.** Чтение и запись не ограничены. LIDS не защищает этот файл.

Данные режимы могут быть заданы как для файла, так и для каталога. Если режим используется для каталога, то он будет применен ко всем файлам, входящим в этот каталог.

В LIDS ACL описывается следующим образом:

<Тип ACL> <субъект> <объект> <доступ> <наследование>

Тип ACL определяет, на какой стадии работы будет контролироваться доступ к системе. Доступно четыре типа:

- **BOOT.** Доступ будет контролироваться на стадии загрузки.
- **POSTBOOT.** После загрузки.
- **SHUTDOWN.** При разгрузке системы.
- **null.** Контроль доступа вне зависимости от стадии работы.

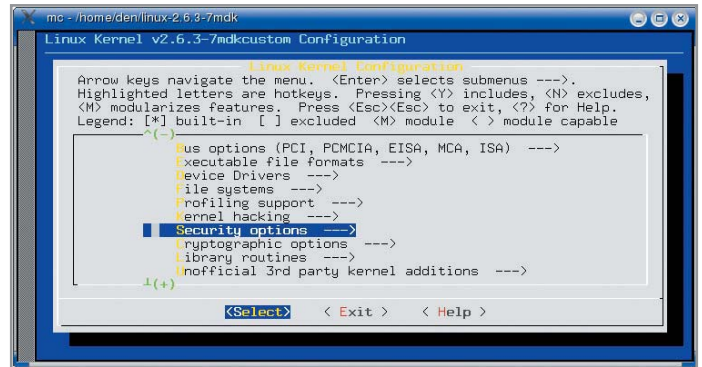
Обычно тип ACL просто не указывается (null) — это делается для постоянного контроля, а первые три типа используются для ослабления определенных ограничений.

Субъект — это приложение, которому предоставляется доступ (режимы доступа описаны выше) к объекту — файлу или каталогу. Последнее поле, <наследование>, определяет, будет ли ACL наследоваться дочерними процессами или нет, и может принимать значения 0, 1 и -1.

Объект и субъект тесно связаны друг с другом. Например, субъекту /usr/bin/sshd (демону SSH) требуется доступ к объекту /var/run/sshd.pid. Вы определяете доступ, например WRITE. Но вы должны понимать, что данное правило определяет доступ только данного субъекта к конкретному объекту, оно никак не относится к другим субъектам. Если же поле <субъект> пусто (то есть субъект не указан), то данный режим доступа применяется к объекту для любого субъекта (то есть любого процесса).

Многие приложения (особенно shell-сценарии) во время выполнения вызывают другие программы, и это обязательно нужно учитывать при разработке ACL. По умолчанию наследование выключено, и дочерние процессы не наследуют режимы доступа, определенные для родительского процесса. Такой режим работы может стать настоящей головной болью для администратора — ему придется устанавливать режим доступа для каждого процесса, вызываемого сценарием. Наследование можно включить, установив значение 1 в поле <наследование>: это так называемый рекурсивный режим наследования, то есть режим доступа будет унаследован только дочерним процессом. Если же вы хотите, чтобы режим доступа был унаследован не только дочерним процессом, но и всеми его потомками, установите рекурсивный режим — значение -1 для поля <наследование>.

ACL хранятся в файле /etc/lids/lid.conf. Открыв его, вы увидите, что правила записаны в непонятном для человека виде. Но вам не нужно редактировать этот файл вручную — для



этого существует утилита lidsconf. Но прежде чем мы перейдем к ее рассмотрению, все-таки необходимо понять, что означают все эти числа:

0:0::1:0:1114128:834:/sbin:0-0

0:0::1:0:1933326:834:/bin:0-0

...

ACL системы LIDS содержит номера инодов вместо имен файлов — например, если мы установим режим APPEND для /var/log/messages, LIDS сохранит в ACL номер инода этого файла. Это позволит более жестко контролировать доступ к файлу, ведь имя ничего не означает, главное — инод: можно удалить файл с именем /var/log/messages и заново создать файл с тем же именем. При этом имя не изменится, а инод — да.

lidsconf

Начиная со второй версии для администрирования и конфигурирования LIDS используются две разные утилиты. Утилите администрирования — lidsadm — мы уже рассмотрели. Осталось исследовать утилиту конфигурирования — lidsconf. Ниже приведены ее наиболее важные опции:

- **-A, --add.** Добавить запись.
- **-C, --check.** Проверить существующие записи.
- **-D, --delete.** Удалить запись.
- **-Z, --zero.** Удалить все записи.
- **-U, --update.** Обновить /dev и номера инодов.
- **-L, --list.** Вывести все записи.

Следующий пример показывает ACL по умолчанию, который поставляется вместе lidstools-2.2.5rc1 (для удобства мы добавили в листинг номера строк):

```
# lidsconf -L
```

	Subject	ACCESS	inherit	Object
1)	Any file	READONLY:	0	/sbin
2)	Any file	READONLY:	0	/bin
3)	Any file	READONLY:	0	/boot
4)	Any file	READONLY:	0	/lib
5)	Any file	READONLY:	0	/usr
6)	Any file	READONLY:	0	/etc
7)	Any file	DENY:	0	/etc/lids
8)	Any file	DENY:	0	/etc/shadow
9)	Any file	APPEND:	0	/var/log
10)	Any file	WRITE:	0	/var/log/wtmp
11)	/bin/login	READONLY:	0	/etc/shadow
12)	/bin/login	GRANT:	0	CAP_SETUID

```
13) /usr/sbin/sshd GRANT: 0 CAP_NET_ADMIN
14) /bin/login GRANT: 0 CAP_GETID
```

Если несколько правил применимы к одному объекту, будет использовано самое последнее из них. Лучше всего это можно продемонстрировать на примере каталога /etc, который указан в правилах 6, 7, 8 и 11. Сначала /etc объявляется как READONLY (только чтение), затем объекты /etc/lids и /etc/shadow делаются невидимыми (DENY). Поскольку субъект не указан (Any file — «любой файл»), то ограничение применяется к любому процессу. А в строке 11 READONLY доступ к файлу /etc/shadow предоставляется только субъекту /bin/login.

Для добавления новых записей используется опция -A программы lidsconf:

```
lidsconf -A [тип ACL] [-s субъект] [-o объект [-t с-по]
[-i уровень] -j действие
```

Как видите, обязательными в этом случае являются только опции A, o и j:

```
lidsconf -A -o /etc/hosts.conf -j READ
```

Все опции нам понятны, кроме разве что [-t с-по]. Она позволяет установить время действия правила, которое обычно указывается в формате ЧЧММ-ЧЧММ. Например, чтобы правило действовало с 8:00 по 19:35, используйте следующую опцию:

```
-t 0800-1935.
```

Для удаления записи используется синтаксис:

```
lidsconf -D [тип ACL] [-s субъект] [-o объект]
```

Мы можем указать субъект и/или объект — будут удалены все совпадающие правила. Можно также обозначить и тип ACL, например POSTBOOT.

| Возможности |

В предыдущем ACL, наверное, вы заметили действие GRANT, предоставляющее возможность CAP_SETUID, а именно их мы до этого и не рассматривали, поэтому самое время приступить.

LIDS предоставляет расширенное использование возможностей (модуль возможностей capability security module должен быть отключен в ядре). Кроме стандартных возможностей Linux система LIDS предлагает две собственные:

- ▶ **CAP_HIDDEN.** Процессы с данной установленной возможностью не будут отображаться в /proc, что позволит скрыть процесс от программ ps, lsof и top;
- ▶ **CAP_INIT_KILL.** Если эта возможность выключена для демона, то он не будет получать KILL-сигналы.

CAP_HIDDEN не гарантирует, что процесс будет полностью невидимым: например, сетевой демон можно обнаружить с помощью netstat или сканера портов, а также по наличию файла /var/run/название.pid.

А теперь перейдем к CAP_INIT_KILL. Рассмотрим дерево процессов с помощью pstree:

```
$ pstree -a
init)
|-atd)
|-(bdf flush)
|-crond)
|-httpd)
| |-httpd)
```

```
| |-httpd)
| |-httpd)
| |-httpd)
|-(keventd)
|-(khubd)
|-(kjournald)
|-klogd) -x
```

В случае с CAP_INIT_KILL демон определяется как процесс, исходящий непосредственно от Init (PID Init всегда равен 1). К сожалению, это недостаток. Поскольку демон не может получать сигналы, администратор не сумеет ни остановить (SIGKILL), ни перезагрузить процесс (SIGHUP). Ко всему прочему, некоторые процессы, например Apache, которые «общаются» со своими «родственниками» с помощью сигналов, не смогут нормально работать. Если же вы хотите использовать CAP_INIT_KILL, первая проблема может быть решена с помощью LFS — отсюда разрешено отправлять процессам сигналы. А вот для решения второй проблемы вы должны включить CAP_INIT_KILL для определенного процесса, например Apache. По-другому тут никак нельзя.

LIDS немного модифицирует возможность CAP_BIND_NET_SERVICE. Обычно она включается для процесса, которому нужно «привязаться» к привилегированному порту (с номером 0-1024). Но LIDS расширяет ее синтаксис, позволяя указывать порт или диапазон портов, к которым разрешена привязка процесса. Например, для Apache раньше разрешалась привязка к любому привилегированному порту, а теперь мы можем четко указать, к каким именно портам разрешается привязываться этому сервису — 80 и 443.

| Ограниченный набор возможностей |

Это список возможностей, доступных (но необязательно установленных) процессу в системе. Если какая-то из них не значится в данном списке, ее нельзя назначать процессу. Соответственно, для каждого процесса можно определить свои списки возможностей.

Конфигурация LIDS по умолчанию (/etc/lids.cap) разрешает все возможности, за исключением следующих:

- ▶ **CAP_SETPCAP.** Возможность устанавливать возможности другого процесса.
- ▶ **CAP_SYS_MODULE.** Возможность загружать и выгружать модули ядра.
- ▶ **CAP_SYS_RAWIO.** Возможность прямого ввода/вывода, то есть доступа к файлам /dev/port, /dev/mem, /dev/kmem, а также прямого доступа к дискам (например, /dev/hda).
- ▶ **CAP_KILL_PROTECTED.** Возможность «убивать» защищенные процессы.

Помните, что X Window требует возможности CAP_SYS_RAWIO. Если вам нужна эта система, установите данную возможность для исполнимого файла X.

| Установка и модификация возможностей |

Установка возможностей производится с помощью все той же утилиты lidsconf. Синтаксис добавления возможности — тот

же, что и в случае добавления правила для файла, только для разрешения этой возможности используется действие GRANT. Добавим возможность запуска X Window:

```
lidsconf -A -s /usr/X11/bin/X -o CAP_SYS_RAWIO -j GRANT
```

В этом случае нужно конкретно указывать субъект правила — процесс, которому разрешается та или иная возможность. В вышеприведенном примере мы предоставляем возможность прямого ввода/вывода CAP_SYS_RAWIO процессу /usr/X11/bin/X.

А теперь разрешим веб-серверу Apache привязываться к непривилегированному порту:

```
lidsconf -A -s /usr/sbin/httpd -o CAP_BIND_NET_SERVICE -j GRANT
```

Более безопасным будет вариант разрешения привязки Apache к портам 80 и 443. Синтаксис возможности не предусматривает задание единичных портов, а только диапазонов, поэтому мы будем использовать нулевые диапазоны 80-80 и 443-443:

```
lidsconf -A -s /usr/sbin/httpd -o CAP_BIND_NET_SERVICE 80-80,443-443 -j GRANT
```

Также рекомендуем установить для Apache возможность CAP_INIT_KILL, чтобы он мог «общаться» со своими потомками (но Apache — это только пример, не забывайте и о других сервисах, которые используют подобную форму IPC!):

```
lidsconf -A -s /usr/sbin/apache -o CAP_INIT_KILL -j GRANT
```

Реализация LIDS

Теперь, когда мы знаем, как работает LIDS, и знакомы с ее основными опциями, можно приступить к ее практической реализации в нашей системе.

Помните, что разработка ACL для всей системы — очень трудная и объемная задача, поэтому мы рекомендуем вам создать shell-сценарий, содержащий вызовы lidsconf. Первой командой будет являться lidsconf -Z — эта директива удаляет все ранее существующие ACL. Данный сценарий нужно поместить в /etc/lids — это сделает его не видимым за пределами LFS-сессии.

Защита системных программ

Какие же системные файлы и каталоги требуют защиты LIDS? Защищать нужно содержимое /bin, /sbin, /lib, /usr/bin, /usr/lib и /usr/sbin. В эти каталоги производится установка программного обеспечения, и, если мы сделаем их доступными только для чтения (READONLY), злоумышленник не сможет записывать троянские версии системных программ. Также не забудьте защитить каталоги /usr/local/bin и /usr/local/lib, если придерживаетесь стратегии установки программ в /usr/local. Конечно, при установке новой программы защита этих каталогов может создать некоторые неудобства для администратора, но поверьте, это стоит того.

/etc и /etc/shadow

Конфигурационные файлы — это другая область защиты. Подавляющее большинство файлов в этом каталоге требует доступ «только для чтения», поэтому можно сначала установить для всего каталога режим READONLY, а затем разрешить доступ в режиме WRITE к каким-то отдельным файлам — глобально или для определенных субъектов.

Наиболее важными файлами в каталоге /etc являются passwd/passwd- и shadow/shadow-: вы должны запретить всем субъектам доступ к shadow/shadow- (DENY), за исключением субъектов /bin/login, su и /usr/sbin/sshd — им полагается доступ READONLY.

Но мы еще не учли утилиту /usr/bin/passwd. Ей нужен WRITE-доступ к файлу /etc/shadow, чтобы пользователь имел возможность изменить свой пароль. Когда он сделает это, файл /etc/shadow будет создан заново, а не просто модифицируется. В результате изменится инод файла, а поскольку LIDS привязывается к инодам, то после его изменения она уже не будет защищать файл — ведь нового инода в «базе данных» LIDS не существует. Кроме того, нужно предоставить программе passwd WRITE-доступ ко всему каталогу /etc, поскольку запись файла — это изменение каталога. Если на месте passwd будет эксплоит злоумышленника, он сможет получить доступ ко всем файлам в каталоге /etc.

К сожалению, не существует простого способа решения этой проблемы: вы или должны использовать альтернативную схему аутентификации, например LDAP, чтобы запретить пользователям изменять свои пароли, или открыть WRITE-доступ к /etc. Существует, правда, еще один способ, самый безопасный, но очень неудобный для администратора. Вы запрещаете WRITE-доступ к /etc, а для того чтобы изменить свой пароль, пользователь должен будет обратиться непосредственно к вам. Вы сможете изменить его пароль в LFS-сессии, а заодно и проверите его на «стойкость». Такой вариант приемлем, если пользователей у вас немного — до 10 человек. Учитывая то, что пароли они меняют не слишком часто, раздражать это вас особо не будет. А вот если пользователей 200...

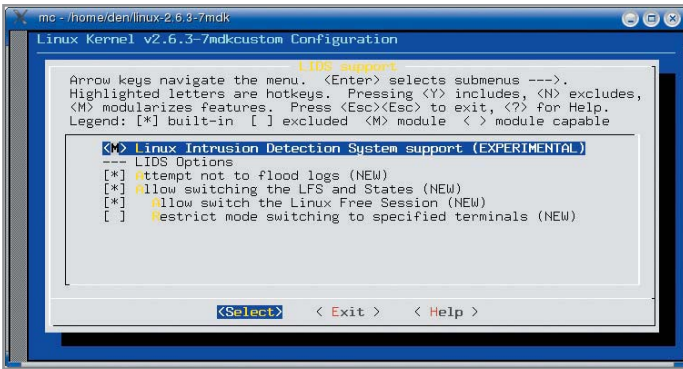
Если обеспечение полной защиты каталога /etc для вас слишком сложно, нужно обеспечить хотя бы защиту ключе-

Предыстория технологии

Модули защиты (LSM)

Все начиналось с разработки различных систем управления доступом, которые реализовывались в виде патчей ядра Linux. Но в результате из-за отсутствия централизованной координации получилось, что каждый проект защиты имел свой патч для ядра, часто несовместимый с другими патчами. В прошлом Линус Торвальдс категорически отвергал все эти патчи, но в 2001 году в ответ на технологию SELinux, представленную NSA (Агентство национальной безопасности) на Linux Kernel Summit, он заявил, что для включения в ядро

будет рассматриваться более обширная система безопасности. Так появился проект LSM (Linux Security Modules). Его цель — предоставить разработчикам систем защиты общий (единый) интерфейс для реализации проектов, основанных на ядре. Это позволит уменьшить зависимость от ядра, кроме того, не нужно будет его перекомпилировать, как в случае с патчами. Все LSM-модули используют стандартный интерфейс для взаимодействия с ядром, который не будет изменяться с выходом следующего релиза (не версии) ядра.



вых файлов. Наиболее важными являются каталоги /etc/rc.d и /etc/rc.d/init.d.

Теперь перейдем к файлу /etc/lids/lids.cap. Как уже было отмечено, следующие возможности должны быть запрещены:

- ▶ **CAP_SYS_RAWIO.** Возможность прямого ввода/вывода, то есть доступа к файлам /dev/port, /dev/mem, /dev/kmem, а также прямого доступа к дискам (например, /dev/hda).
- ▶ **CAP_SYS_PTRACE.** Возможность трассировки системных вызовов, производимых процессом.
- ▶ **CAP_SETPCAP.** Возможность устанавливать возможности другого процесса.
- ▶ **CAP_KILL_PROTECTED.** Возможность «убивать» защищенные процессы.
- ▶ **CAP_SYS_MODULE.** Возможность загружать и выгружать модули ядра.

Единственное приложение, которое требует первую возможность, — это X11. Для всех остальных приложений все эти возможности должны быть отключены.

Определение требуемого доступа

Как определить, к каким файлам и каталогам нужно обращаться тому или иному приложению? Отслеживать системные вызовы open(), chdir(), mkdir() и другие — дело неблагодарное, но вам все же нужно будет через это пройти. Немного облегчить задачу позволяет LIDS-FAQ, расположенный по адресу www.lids.org/fids-faaq/lids-faq.html. Там вы найдете ACL для таких приложений как login, su, MySQL, BIND, OpenSSH, Apache и других.

При самостоятельной разработке ACL для приложения, которое вы не встретите в LIDS-FAQ, последовательность действий приблизительно следующая:

- ▶ Проверьте, к каким файлам и каталогам необходим доступ для этого приложения. Это можно сделать с помощью ptrace. Какие файлы конфигурации использует программа? Будет ли она записывать что-то в /var/run?
- ▶ Нужно ли приложению привязываться к привилегированному порту? Если да, то для него нужно включить возможность CAP_BIND_NET_SERVICE, указав необходимые порты.

Проверить ACL просто: запустите приложение при включенной защите LIDS. Если что-то пойдет не так, в /var/log/messages появится соответствующее сообщение LIDS.

Пример ACL для DNS-сервера

Давайте рассмотрим пример ACL, позволяющий DNS-серверу работать на нашей машине. Этот ACL будет состоять из двух

частей. Первая часть содержит общий набор правил, предоставляющих базовый доступ, — она подойдет не только для DNS-сервера, но и других приложений, запущенных в системе Linux. Во второй части будут описаны специфические для DNS-сервера правила, ограничивающие доступ к файлам, а также определяющие его возможности. Как было отмечено ранее, набор правил мы представим в виде shell-сценария, содержащего серию команд lidsconf.

Сперва нужно сделать системные исполнимые файлы и библиотеки доступными только для чтения. Обойти это ограничение можно только в LFS-оболочке. Итак, нам нужно защитить каталоги /bin, /sbin, /usr и /opt:

```
/sbin/lidsconf -A -o /bin -j READONLY
/sbin/lidsconf -A -o /sbin -j READONLY
/sbin/lidsconf -A -o /usr -j READONLY
/sbin/lidsconf -A -o /opt -j READONLY
```

Мы не определили субъект, поэтому указанные объекты будут доступны только для чтения всем процессам в системе. Не забывайте, что ACL наследуется, то есть доступ READONLY получают также и все подкаталоги данных каталогов. Также следует помнить, что наследование не распространяется на подмонтированные файловые системы. Например, если к /usr/local подмонтирован другой раздел, то правила, примененные к /usr, не будут распространяться на файлы и каталоги, находящиеся на другом разделе.

Кроме этих каталогов нам нужно защитить также /etc и /boot. Однако, как было отмечено ранее, предоставление доступа READONLY к каталогу /etc довольно проблематично, поэтому мы сконцентрируемся на защите его ключевых файлов, ведь наибольший интерес для злоумышленника представляют именно они. Например, /etc/exports определяет экспортируемые файловые системы, а изменив файл /etc/resolv.conf, он может перенаправить запросы нашего резолвера на свой DNS-сервер, который будет выдавать неправильную информацию.

```
/sbin/lidsconf -A -o /boot -j READONLY
/sbin/lidsconf -A -o /etc/HOSTNAME -j READONLY
/sbin/lidsconf -A -o /etc/apache -j READONLY
/sbin/lidsconf -A -o /etc/cron.daily -j READONLY
/sbin/lidsconf -A -o /etc/cron.hourly -j READONLY
/sbin/lidsconf -A -o /etc/cron.weekly -j READONLY
/sbin/lidsconf -A -o /etc/exports -j READONLY
/sbin/lidsconf -A -o /etc/hosts -j READONLY
/sbin/lidsconf -A -o /etc/hosts.allow -j READONLY
/sbin/lidsconf -A -o /etc/hosts.deny -j READONLY
/sbin/lidsconf -A -o /etc/hosts.equiv -j READONLY
/sbin/lidsconf -A -o /etc/identd.conf -j READONLY
/sbin/lidsconf -A -o /etc/ld.so.conf -j READONLY
/sbin/lidsconf -A -o /etc/login.access -j READONLY
/sbin/lidsconf -A -o /etc/login.defs -j READONLY
/sbin/lidsconf -A -o /etc/logrotate.conf -j READONLY
/sbin/lidsconf -A -o /etc/mail -j READONLY
/sbin/lidsconf -A -o /etc/modules.conf -j READONLY
/sbin/lidsconf -A -o /etc/named.conf -j READONLY
/sbin/lidsconf -A -o /etc/networks -j READONLY
/sbin/lidsconf -A -o /etc/ntp.conf -j READONLY
```



```

/sbin/lidsconf -A -o /resolv.conf      -j READONLY
/sbin/lidsconf -A -o /rc.d            -j READONLY
/sbin/lidsconf -A -o /services        -j READONLY
/sbin/lidsconf -A -o /shells          -j READONLY
/sbin/lidsconf -A -o /ssh              -j READONLY
/sbin/lidsconf -A -o /sudoers         -j READONLY
/sbin/lidsconf -A -o /sudoers.conf    -j READONLY
/sbin/lidsconf -A -o /etc/           -j READONLY

```

В зависимости от установленных в системе пакетов, возможно, придется добавить и другие файлы в этот список. Мы же описали наиболее критичные из них.

Также не нужно забывать про файлы журналов, которые находятся в каталоге `/var/log`. Для большинства из них можно установить режим APPEND, для некоторых — WRITE, но только для субъектов login, init и halt возможны лишь следующие режимы:

```

/sbin/lidsconf -A -o /var/log          -j APPEND
/sbin/lidsconf -A -s /bin/login -o /var/log/wtmp -j WRITE
/sbin/lidsconf -A -s /bin/login -o /var/log/lastlog -j WRITE
/sbin/lidsconf -A -s /sbin/init -o /var/log/wtmp -j WRITE
/sbin/lidsconf -A -s /sbin/init -o /var/log/lastlog -j WRITE
/sbin/lidsconf -A -s /sbin/halt -o /var/log/wtmp -j WRITE
/sbin/lidsconf -A -s /sbin/halt -o /var/log/lastlog -j WRITE

```

Благодаря этим ограничениям злоумышленник, получивший root-доступ, не будет способен отредактировать эти файлы, чтобы скрыть свое присутствие. Недостаток этого метода заключается в том, что утилита logrotate не сможет больше функционировать, но все же есть своя цена. Теперь за «уборку» журналов отвечает лично вы — администратор. Чтобы logrotate работала, ей нужно предоставить WRITE-доступ ко всему каталогу `/var/log`, но мы не рекомендуем этого делать, поскольку злоумышленник может начать запускать logrotate много раз подряд — до тех пор, пока из журнала не будут удалены следы его присутствия. Отключите logrotate и «почистите» журналы вручную. Во многих системах logrotate вызывается демоном cron — чтобы этого не происходило, нужно удалить файл `/etc/cron.daily/logrotate` или закомментировать его содержимое.

Теперь вернемся к правилам. Нам нужно определить, как named взаимодействует с системой; мы должны предугадать все файлы, доступ к которым понадобится приложению, а также определить возможности этого самого приложения. Сперва запретим доступ ко всей файловой системе:

```

/sbin/lidsconf -A -s /usr/sbin/named -o /      -j DENY

```

BIND должен получить доступ к файлу конфигурации (`/etc/named.conf`), а также файлам зоны (каталог `/var/named`):

```

/sbin/lidsconf -A -s /usr/sbin/named -o /etc/named.conf -j READ
/sbin/lidsconf -A -s /usr/sbin/named -o /var/named      -j READ

```

Как и любой другой демон, named записывает свой PID в файл, расположенный в каталоге `/var/run`. Обычно он называется `/var/run/named.pid`. Мы должны разрешить приложению создавать файл с таким именем:

```

/sbin/lidsconf -A -s /usr/sbin/named -o /var/run/named.pid -j WRITE

```

На данный момент мы позаботились обо всех файлах, которые нужны демону named. Теперь нужно определить, какие ему понадобятся библиотеки. Для этого мы будем использовать

программу strace, которая отобразит все системные вызовы, а вместе с ними и внешние файлы. Опция `-f` не позволяет приложению перейти в фон:

```
# strace -f -o named_trace named
```

Вывод программы named будет записан в файл `named_trace`. Демон named должен поработать несколько часов, после этого завершите процесс (named, а не strace!) и проанализируйте файл `named_trace`:

```
# cat named_trace | grep open
```

Данная команда выведет все вызовы `open()` — вы увидите не только файлы, задействованные приложением, но и режимы, в которых они используются. На основании этого списка мы можем составить следующий список правил LIDS:

```

/sbin/lidsconf -A -s /usr/sbin/named -o /      -j DENY
/sbin/lidsconf -A -s /usr/sbin/named -o /usr/lib -j READ
/sbin/lidsconf -A -s /usr/sbin/named -o /lib    -j READ
/sbin/lidsconf -A -s /usr/sbin/named -o /usr/share/locale -j READ
/sbin/lidsconf -A -s /usr/sbin/named -o /etc/ld.so.preload -j READ
/sbin/lidsconf -A -s /usr/sbin/named -o /etc/ld.so.cache -j READ
/sbin/lidsconf -A -s /usr/sbin/named -o /etc/localtime -j READ
/sbin/lidsconf -A -s /usr/sbin/named -o /etc/rndc.key -j READ
/sbin/lidsconf -A -s /usr/sbin/named -o /var/log  -j APPEND
/sbin/lidsconf -A -s /usr/sbin/named -o /dev/random -j READ

```

Данный перечень несколько упрощен, поскольку named использует много библиотек в `/usr/lib` и `/lib`, но проще предоставить READ-доступ к этим каталогам, чем прописывать отдельно каждую библиотеку.

Теперь пришла очередь установки возможностей. Прежде всего, разрешим named привязываться к порту 53:

```

/sbin/lidsconf -s /usr/sbin/named -o CAP_NET_BIND_SERVICE 53-53 -j GRANT

```

При запуске named с опцией `-u <имя пользователя>` он снижает свои привилегии до уровня обычного пользователя, указанного с помощью опции `u`. Поэтому нужно разрешить ему производить вызовы SUID и SGID:

```

/sbin/lidsconf -s /usr/sbin/named -o CAP_SETUID 53-53 -j GRANT

```

```

/sbin/lidsconf -s /usr/sbin/named -o CAP_SETGID 53-53 -j GRANT

```

Если BIND запускается в chroot-окружении, нужно установить возможность CAP_SYS_CHROOT:

```

/sbin/lidsconf -s /usr/sbin/named -o CAP_SYS_CHROOT -j GRANT

```

Как и Apache, BIND может иметь несколько потомков, которым понадобится доступ ко всем файлам и возможностям, описанным ранее, поэтому для переноса возможностей другим процессам нужно разрешить CAP_SYETPCAP:

```

/sbin/lidsconf -s /usr/sbin/named -o CAP_SYETPCAP -j GRANT

```

Настало время протестировать созданный ACL. Запустите named и следите за системными журналами — в них вы найдете сообщения об ошибках, если что-то вдруг пойдет не так.

В завершение нужно сказать, что если у вас возникли проблемы с тем или иным сервисом, посетите сайт www.lids.org — там вы найдете множество готовых ACL для разных сервисов. |

Сергей Яремчук

Агент системной безопасности

По данным институтов, занимающихся безопасностью (например, CERT, www.cert.org), число инцидентов в Интернете постоянно растет. Действительно, данные с описаниями взломов, а также готовые программы и эксплоиты сегодня найти довольно просто. Поэтому даже неопытный пользователь может вообразить себя хакером и попробовать в действии весь доступный арсенал.

Для обнаружения атак в состав Unix-систем включается целый ряд полезных приложений, имеющих определенные задачи и нацеленных на конкретный класс атак. Здесь и межсетевые экраны, закрывающие ненужные порты, и антивирусные программы, осуществляющие поиск вирусов, и сетевые сканеры, и сканеры безопасности, определяющие уязвимые участки сети, которые могут быть использованы злоумышленником для атаки. Пассивные атаки, цель которых — сбор информации без воздействия на работающие службы вроде перехвата трафика, определяются программами-антиснифферами (AntiSniff, Sentinel). Средства контроля целостности файловых систем (Tripwire, AIDE) и обнаружения закладок (chkrootkit, rkdetect) позволяют безошибочно определить несанкционированное изменение важных системных файлов или установку suid/guid на пользовательские приложения. Всех программ не перечислить. Особое место в этом списке занимают IDS (Intrusion Detection Systems) — системы обнаружения атак и вторжений.

Системы IDS, в свою очередь, также подразделяются на классы. Например, сетевые системы обнаружения атак (Network

Intrusion Detection System — NIDS) и «индивидуальные» системы host-based. Первые контролируют проходящий сетевой трафик, а вторые, анализируя данные на конкретном узле, пытаются обнаружить злонамеренные действия. В последнее время начали быстро развиваться так называемые гибридные IDS, сочетающие в себе возможности обоих типов — сетевых и узловых. Есть еще и Application Based IDS, выявляющие атаки, направленные на конкретные приложения. По типу определения атак системы IDS делятся на сигнатурные, работа которых подобно антивирусам заключается в поиске заранее известных признаков атаки, и системы, реагирующие на аномалии в контролируемой системе или сети. Имеются и подклассификации, например статистические и адаптивные системы, строящие профиль защищаемой среды по разным правилам. Кроме того, сегодня быстрыми темпами развиваются Intrusion Prevention System, способные не только обнаруживать, но и останавливать атаки. Каждый тип подобных систем имеет свои достоинства и недостатки, о них мы говорить не будем, классификация же приведена для того, чтобы была понятна сложность этой задачи,

а также показано конкретное место в общей системе обнаружения телекоммуникационных атак, которое занимает NIDS Snort (www.snort.org). Еще одно отступление, которое необходимо сделать: дословно IDS переводится как «система обнаружения вторжений», поэтому в литературе чаще всего используется именно этот термин, что, в принципе, не всегда правильно. Например, система IDS может обнаружить неудачную атаку, которая не привела к вторжению, поэтому более логичным будет использование термина «система обнаружения атак» — COA.

Возможности Snort

Первые упоминания о COA относятся к 1980 году, а именно к публикации Джона Андерсона «Computer Security Threat Monitoring and Surveillance». Но активно развиваться это направление стало гораздо позже, в 90-х годах прошлого столетия.

О Snort впервые заговорили в 1998 году, когда ее основатель Мартин Роеш решил создать некие правила, которые могли бы логично описывать информацию перехваченных сетевых пакетов в созданном им пакетном сниффере. Snort очень быстро учился обнаружению атак, а Мартин постоянно совершенствовал свой продукт, привлекая все большее количество сторонников. Постепенно Snort обретал способность работать с фрагментированными пакетами, учился обрабатывать TCP-заголовки, SLIP- и PPP-пакеты, правила становились несколько неудобными в написании, но зато выигрывали по части читаемости, росло количество операционных систем, в которых можно было запустить Snort, появлялись препроцессоры.

Итак, Snort является сетевой системой обнаружения атак с открытым исходным кодом, способной выполнить в реальном времени анализ IP-пакетов, передаваемых на контролируемых интерфейсах, с целью обнаружения атак или попыток поиска уязвимостей. Snort обнаруживает атаки, комбинируя два метода — сигнатурный и анализ протоколов. Для описания событий, которые могут считаться злонамеренными или аномальными, используются гибкий язык правил и модульная система анализа. Кстати, сегодня существует два типа правил: официальные сертифицированные и строго протестированные Sourcefire VRT Certified Rules, распространяющиеся по лицензии VRT Certified Rules License Agreement, ограничивающей их коммерческое использование. Эти правила доступны в двух вариантах — для зарегистрированных и незарегистрированных пользователей. Регистрация абсолютно бесплатна. Все изменения в первую очередь распространяются по подписчикам (subscription release с буквой s в названии пакета), затем становятся доступными для зарегистрированных пользователей (без буквы s). Те же, кто не регистрировался, довольствуются статистическими правилами, обновляемыми только к выходу очередного релиза Snort и, естественно, отстающими от жизни (они имеют префикс rg в названии). Например, на момент написания статьи (середина ноября) эти правила датировались июлем 2005 года. Второй тип правил называется Community Rules. Они созданы добровольцами, но еще не прошли проверку и распространяются под лицензией GPL. Решать, использовать их или нет, вам.

Но это еще не все. Начиная с версии 2.3.0 RC1, в Snort включен код проекта Snort-inline, тем самым он получил возмож-



ность не только выявлять, но и останавливать предпринятую атаку, перестраивая правила iptables. И теперь Snort можно по праву назвать полноценной системой предотвращения атак.

Сегодня Snort может работать в четырех режимах:

- ▶ как пакетный сниффер, который подобно tcpdump, также используя libpcap, отлавливает пакеты в сети и выводит на экран информацию о них;
- ▶ как регистратор пакетов, записывающий данные на диск;
- ▶ как комплексная перестраиваемая система обнаружения атак, анализирующая сетевой трафик, следуя правилам, и выполняющая на их основе определенные действия;
- ▶ как система предотвращения атак, получающая вместо libpcap пакеты из iptables, способная отвергать или пропускать пакеты, основываясь на специфических правилах.

Всю собранную информацию детектор Snort позволяет сохранить в файлах журналов различных форматов (обычные ASCII, текстовые файлы или так называемые бинарные, совместимые с tcpdump). Кроме того, для удобства анализа всю собранную информацию можно занести в базу данных: PostgreSQL, MySQL, MS SQL Server, Oracle или unixODBC. Хотя, в принципе, можно использовать любую другую базу данных, не представленную в этом списке, но тогда таблицы придется формировать вручную. Часто в литературе Snort называют легкой IDS, подразумевая, что она предназначена для работы в сетях с небольшой нагрузкой. Это утверждение несколько устарело, учитывая хотя бы то, что сегодня даже домашние сети могут быть построены на гигабитных адаптерах, которые называют легкими язык уже не поворачивается. Система, построенная на датчиках Snort, способна собирать и обрабатывать информацию с нескольких сетевых локаций. Все в дело в производительности компьютеров, используемых в качестве сенсоров. Для того чтобы улучшить производительность, разделяя быструю работу IDS по захвату пакетов и относительно медленную по занесению информации, необходимо использовать Barnyard, который доступен на странице загрузки проекта Snort. В этом случае Snort создает двоичный выходной, так называемый «унифицированный» формат, с которым в дальнейшем и работает Barnyard.

Установка Snort

На момент написания статьи актуальной была версия 2.4.3. В качестве варианта можно использовать последний снимок snort-snapshots-CURRENT.tar.gz. Хотелось бы отметить, что подкаталог contrib, содержащий различные дополнения к Snort, начиная с версии 2.2.0 пустует. Скрипты для создания баз данных переместились в подкаталог schemas, а скрипты для создания RPM-

пакетов — в одноименный подкаталог. Остальные же расширения можно найти на странице www.snort.org/dl/contrib. Все самое важное сказано, теперь можно начинать установку. Для выполнения большинства операций потребуются права root.

Распаковываем архив:

```
$ tar -xzf snort-snapshots-CURRENT.tar.gz
$ cd HEAD
```

Внутри вы не найдете привычного конфигурационного скрипта, его необходимо создать:

```
$ ./autojunk.sh
```

Теперь конфигурируем. В самом простом случае скрипту никаких параметров передавать не надо. Если же необходимо использовать базу данных, то, например, для MySQL добавляем опцию `--with-mysql`, для включения режима остановки атак добавляем `--enable-inline`. Остальные опции можно посмотреть с помощью `-help`:

```
$ ./configure --with-mysql
```

Так как Snort является, по своей сути, сниффером, перехватывающим пакеты, то для компиляции требуется библиотека `libpcap`, которая обычно идет вместе с `tcpdump`. Поэтому, если во время конфигурирования вы получили такую ошибку:

```
ERROR! Libpcap header not found, go get it from
www.tcpdump.org/ or use the --with-libpcap-*
options, if you have it installed in an unusual place
```

Необходимо зайти на сайт www.tcpdump.org, скачать и затем установить библиотеку `libpcap`. Или другой пример. При конфигурировании с опцией `--enable-inline` я получил следующую ошибку:

```
checking "for libnet.h version 1.0.x"... /sw/include
./configure: line 1: dnet-config: command not found
./configure: line 1: dnet-config: command not found
checking dnet.h usability... no
checking dnet.h presence... no
checking for dnet.h... no
```

```
ERROR! Libdnet header not found, go get it from
http://libdnet.sourceforge.net or use the --with-dnet-*
options, if you have it installed in an unusual place
```

В принципе, в подсказке написано, что необходима библиотека `libnet`, но в этом случае я знал, что она у меня есть. Поэтому я просто нашел, где лежит нужный заголовочный файл:

```
# find /usr -name libnet.h
```

И добавил при конфигурировании параметр `--with-dnet-includes=/usr/include/`.

Также при установке в режиме `inline` вам обязательно потребуются наличие заголовочных файлов `iptables`. Если конфигуратор не может их найти, то, для того чтобы преодолеть эту проблему (если не помогают вышеописанные методы), необходимо скачать и распаковать исходные тексты `iptables` (www.iptables.org) и в образовавшемся каталоге ввести команду `make install-devel`. Когда конфигурация завершится без ошибок, нужно ввести традиционные `make` и `make install` и приступить к настройке.

| Настройка Snort |

Не знаю, с чем это связано, но все каталоги, необходимые для работы Snort, до сих пор приходится создавать вручную.

```
# mkdir /etc/snort
```

Сюда мы будем складывать все конфигурационные файлы и правила.

```
# mkdir /var/log/snort
```

А здесь будет вестись журнал работы.

Теперь в каталог `/etc/snort` копируем все, что лежит в подкаталоге `etc` дистрибутива.

```
# cp -R /home/sergej/work/HEAD/etc/* /etc/snort/
```

Далее распаковываем файл правил и помещаем их в `/etc/snort/rules`. В принципе, место для них можно выбрать любое, но так удобнее, к тому же это считается традиционным:

```
# tar -xzf snortrules-snapshot-CURRENT.tar.gz
# mv rules /etc/snort
```

Рассмотрим одно правило, для того чтобы стало ясно, как они пишутся. Например, одно-единственное правило в файле `virus.rules` содержит такую запись:

```
alert tcp $HOME_NET any -> $EXTERNAL_NET 25
(msg:"VIRUS OUTBOUND bad file attachment"; flow:to_server,established; content:"Content-Disposition[3A]"; nocase;
pcre:"/filename\s*=\s*.*\.(?=[abcdehijlmnoprsvwxyz])(a(d[ep])[s]
dfx)|c([ho]m[il]md[pp])d(iz|ll|ot)e(m[fl]|xe)h(lp[sql]ta)]se?|m(d[abe]
w)[s[ip])|p(p[st]|if[lm]|ot)|r(eg[tf])s(cr[hy]s[wf])v(b[es])?cf[xd]w(m
[dfsz])p[dmsz])s[cfh])xl[tw]|bat[ini]lnk[nws]ocx)[\x27\x22\n\r\s]/
iR"; classtype:suspicious-filename-detect; sid:721; rev:8;)
```

Несмотря на довольно внушительный вид, правило очень простое, и если разобрать его по частям, то все становится на свои места. Первая строка говорит о том, что все сообщения по протоколу TCP, направленные из домашней сети с любого порта во внешнюю сеть на порт 25 (что говорит о почтовых сообщениях), имеющие в своем составе присоединенный файл с определенными расширениями, определяются как подозрительные. Директива `alert` указывает на действия, которые должен производить Snort при обнаружении пакета, попадающего под это правило. По умолчанию имеется пять действий: `alert`, `log`, `pass`, `activate` и `dynamic`. Кроме того, в режиме `inline` доступны еще три — `drop`, `reject` и `sdrop`. В некоторых случаях в состав правила включены комментарии (`reference`), позволяющие найти более подробную информацию об уязвимости на специальных ресурсах. В нашем же случае таких указателей нет, так как это правило можно отнести к общим, но в самом файле содержится подробное объяснение. Правило может быть односторонним (`->`) и двусторонним (`< >`), когда направление движения пакета роли не играет. Также в правиле может использоваться директива `priority`, указывающая на приоритет. При появлении событий с определенным приоритетом они могут быть обработаны сторонними утилитами, такими как `swatch` (Simple Watcher) или `syslog-ng` (`syslog-next generation`), и выполнять какие-либо действия, например отправлять e-mail.

| Файл конфигурации snort.conf |

И последний шаг, который осталось сделать, — отредактировать конфигурационный файл `/etc/snort/snort.conf`. В дистрибутиве уже имеется готовый шаблон, поэтому с нуля его писать не придется. В файле используются переменные, в том числе встречающиеся и в правилах. Это довольно удобно — при смене какого-либо параметра не придется его переписывать несколько раз. Кроме того, некоторые опции вынесены во внеш-

ние файлы, которые подключаются директивой `include` с именем файла. Все параметры щедро снабжены комментариями, которые традиционно начинаются со знака решетки. Для удобства восприятия файл разбит на пять частей:

- ▶ установка переменных сети;
- ▶ настройка препроцессоров;
- ▶ настройка вывода информации;
- ▶ установка дополнительных директив;
- ▶ модификация правил.

Для нормальной работы достаточно настроить первые три пункта, остальные можно пока не трогать.

Переменная `HOME_NET` определяет IP-адреса, которые Snort будет считать адресами домашней сети. Возможно задание отдельного адреса или диапазона. Если требуется указать несколько адресов, они перечисляются через запятую. Ключевое слово `any` означает любой адрес. Например:

```
var HOME_NET 10.1.1.0/24
var HOME_NET [10.1.1.0/24,192.168.1.0/24]
```

Переменная `EXTERNAL_NET` указывает на внешние узлы. По умолчанию выставлено значение `any`. Его можно оставить как есть, а более логичным будет указать, что все, не являющееся домом, будет внешним:

```
var EXTERNAL_NET !$HOME_NET
```

Ниже в файле идет список серверов (DNS, SMTP, web, SQL, telnet и SNMP), используемых в сети. Можно оставить как есть, то есть `$HOME_NET`, или указать конкретный IP-адрес, но, с другой стороны, если у вас нет веб-сервера, то зачем отслеживать специфические для него атаки? Поэтому лишнее можно смело отключить. Далее задаются номера портов, используемых серверами. Это позволяет Snort не распылять ресурсы, а искать атаку более конкретно. Принцип тот же: если нет Oracle, то соответствующую строку лучше закомментировать. Обратите внимание, что номер порта может быть задан как единичный [80] и как непрерывный [80:8080]. Перечисление портов через запятую работать не будет (это обещается исправить в будущем). Поэтому, если веб-сервер использует два порта, необходимо написать следующее:

```
var HTTP_PORTS 80
var HTTP_PORTS 8080
```

Также обратите внимание, что переменную `RULE_PATH` нужно определить как

```
var RULE_PATH rules
```

А не как

```
var RULE_PATH ../rules
```

Препроцессоры, подключаемые во второй секции «Configure preprocessors», — штука довольно серьезная и полезная, но требующая некоторого времени, для того чтобы разобраться с назначением и особенностями работы. Обратите внимание, что некоторые препроцессоры дублируют друг друга, поэтому включать все сразу также не имеет смысла. Так, вместо Portscan и Flow-Portscan разработчики рекомендуют использовать sfPortscan, разработанный в Sourcefire и предназначенный для тех же целей, то есть для определения сканирования портов. Более быстрый в работе модуль Frag3, предназначенный для дефрагментации IP-пакетов, пришел на замену устаревшему Frag2. Кроме того, некоторые препро-

цессоры направлены на определение аномалий в работе определенных сервисов. Так, X-Link2State предназначен для определения уязвимости в Microsoft Exchange Server, HTTPInspect изучает аномалии в HTTP-трафике.

В третьей секции «Configure output plugins», как уже говорилось, настраиваются выходные параметры. В общем случае строка параметров имеет такой вид:

```
output <name_of_plugin>: <configuration_options>
```

В настоящее время Snort может использовать десять плагинов для вывода информации (каждый из которых имеет дополнительные опции):

- ▶ `alert_syslog` — для вывода информации используется демон `syslog`; модуль настраивает приоритеты сообщений и уровень;
- ▶ `alert_fast` — информация о возможной атаке выводится в указанный в качестве дополнительного параметра файл в сокращенном формате, без подробностей;
- ▶ `alert_full` — модуль, подходящий для небольших сетей, так как сильно затормаживает работу Snort; заголовок пакета выводится полностью, в лог-каталоге будет создан подкаталог, в который по каждому IP будут записываться пакеты, вызвавшие предупреждение;
- ▶ `alert_unixsock` — похож на предыдущий, только информация в реальном времени передается в Unix-сокеты, откуда может быть считана любой другой программой;
- ▶ `log_tcpdump` — записывает в указанный файл (к его имени будет добавляться метка времени, поэтому затереть его при перезапуске не получится) перехваченные пакеты в формате утилиты `tcpdump`;
- ▶ `database` — модуль, позволяющий заносить информацию в базу данных;
- ▶ `csv` — вывод в файл формата `csv`, который может быть использован для занесения информации в базу данных; кроме имени файла необходимо перечислить параметры, которые в него заносятся;
- ▶ `unified` — выводит данные в специальном формате, оптимизированном для обработки внешними утилитами, которые затем будут заниматься регистрацией события; в универсальном формате используются два файла — `alert_unified` и `log_unified`, первый содержит IP-адрес, порт, протокол, ID сообщения, во второй записывается дамп пакета;
- ▶ `alert_prelude` — доступен при конфигурировании с опцией `-enable-prelude`, в этом случае Snort используется как датчик гибридной IDS Prelude (www.prelude-ids.org);
- ▶ `log_null` — в этом случае Snort способен реагировать на указанные предупреждения без регистрации пакетов.

Готовые примеры имеются в файле. Для работы их достаточно раскомментировать, поэтому останавливаться на них мы не будем. Можно создать и собственные правила и привязать к ним определенный тип действий. Например, изменив `alert` на `suspicious` в приведенном выше правиле и записав следующее действие, подозрительные пакеты можно захватывать в формат `tcpdump`:

```
ruletype suspicious
{
  type log
  output log_tcpdump: suspicious.log
}
```

В конце секции с помощью директивы `include` подключаем файл `classification.config`, содержащий описание классификаций и приоритетов атак, а также файл `reference.config`, содержащий URL для сообщений об обнаруженных уязвимостях.

И, наконец, в конце файла вы найдете секцию «Customize your rule set», в которой необходимо убрать комментарии, указывающие на файлы с правилами:

```
include $RULE_PATH/local.rules
include $RULE_PATH/bad-traffic.rules
include $RULE_PATH/exploit.rules
include $RULE_PATH/scan.rules
include $RULE_PATH/finger.rules
....
# include $RULE_PATH/multimedia.rules
# include $RULE_PATH/p2p.rules
include $RULE_PATH/experimental.rules
```

Названия правил говорят сами за себя. Оставьте то, что действительно нужно (если сомневаетесь, лучше включите все). По умолчанию файл `local.rules` пуст, в него заносит свои правила сам пользователь. В качестве варианта можно создать отдельный файл, вынести в него необходимые параметры и подключить с помощью директивы `include`.

| Запуск Snort |

После того как все готово, можно запускать Snort. Для работы в режиме sniffера Snort запускается с флагом `-v`. При этом на экран будут выводиться заголовки пакетов. Если же вы хотите видеть и данные, используйте команду:

```
# snort -vd
*** interface device lookup found: eth0
Running in packet dump mode
Initializing Network Interface eth0
--== Initializing Snort ==--
Initializing Output Plugins!
Decoding Ethernet on interface eth0
--== Initialization Complete ==--
,,_  -> Snort! <*-
o" )~  Version current (Build 29)
""  By Martin Roesch & The Snort Team:
http://www.snort.org/team.html
(C) Copyright 1998-2005 Sourcefire Inc., et al.
11/14-18:06:25.638963 192.168.0.1:2291 -> 192.168.0.20:114
TCP TTL:128 TOS:0x0 ID:17112 IpLen:20 DgmLen:48 DF
***S* Seq: 0x3AB82ED6 Ack: 0x0 Win: 0x4000 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
11/14-18:06:25.639014 192.168.0.20:114 -> 192.168.0.1:2291
TCP TTL:64 TOS:0x0 ID:2523 IpLen:20 DgmLen:40 DF
***A*R** Seq: 0x0 Ack: 0x3AB82E
Snort received 1714 packets
  Analyzed: 1157(67.503%)
  Dropped: 557(32.497%)
Breakdown by protocol:
  TCP: 451 (26.313%)
  ICMP: 71 (4.142%)
  ETHLOOP: 0 (0.000%)
```

```
DISCARD: 0 (0.000%)
Action Stats:
ALERTS: 0
LOGGED: 0
PASSED: 0
```

Если в системе один интерфейс, то программа сама разберется, с чем ей работать. В противном случае его требуется указать при помощи `-i`:

```
# snort -vd -i eth0
```

Можно указать на конкретную информацию, которую требуется захватить. Например, устанавливаем в качестве домашней сети `192.168.1.0` и захватываем пакеты с узла `192.168.1.1`:

```
# snort -h 192.168.1.0/24 -d -v host 192.168.1.1
```

Для регистрации пакетов в общем случае указываем каталог, в который надо записывать информацию:

```
# snort -l ./log
```

Если на выходе требуется файл формата `tcpdump`, то добавляем параметр `-b`.

И, наконец, работа в режиме обнаружения атак. Так как файл `snort.conf` уже создан, то поступаем просто:

```
# snort -c /etc/snort/snort.conf
```

Для тестирования набираем `ping -s 65507`. После чего, если выбран соответствующий режим ведения журнала, в каталоге `/var/log/snort` появится файл с предупреждением о потенциально зловредном пакете:

```
[**] [1:499:3] ICMP Large ICMP Packet [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
15/11-18:21:2.1131991802 192.168.0.1 -> 192.168.0.20
ICMP TTL:255 TOS:0x0 ID:18479 IpLen:20 DgmLen:63028
Type:0 Code:0 ID:512 Seq:19456 ECHO REPLY
[Xref => arachnids 246]
```

При более всестороннем тестировании COA следует использовать специальные утилиты вроде `IDSwakeUp` (www.hsc.fr/ressources/outils/idswakeup/download).

Для автоматического запуска Snort при загрузке системы необходимо использовать скрипт `snortd`, который находится в подкаталоге RPM-дистрибутива. Копируем его в `/etc/rc.d/init.d/` и даем команду:

```
# chkconfig snortd on
```

Snort создан для того, чтобы выполнять одну задачу — определение атак, и выполняет он ее хорошо. Анализ файлов журналов отдан на откуп сторонним разработчикам. Некоторые утилиты, предназначенные для этих целей, вы найдете на сайте проекта. Например, при помощи Perl-скрипта `SnortALog` (<http://jeremy.chartier.free.fr/snortalog>) можно отобразить необходимую информацию и вывести ее в удобочитаемом виде. Вот так можно вывести топ-лист атак, сгруппированный по времени, и отослать его по почте:

```
# cat /var/log/snort/snort.log.1131990681 | ./snortalog.pl
-hour_attack | /usr/sbin/sendmail -f admin@domain.com
```

Несмотря на то что описано уже многое, за кадром осталось еще немало вопросов, таких как работа с базой данных и автоматическое обновление правил. Почти ничего не сказано о средствах анализа собранной информации. Обо всем этом мы поговорим в следующий раз. |

Полный realtime

В наше время большинство домашних компьютеров применяется совсем не в бытовых целях — уже стало обычным делом производить дома оцифровку видео или музыки в реальном масштабе времени (realtime), а не идти для этого в специализированную студию.

Это стало возможным в связи с увеличением вычислительных мощностей современных компьютеров и возможностей новых версий ОС. Например, ядро Linux версии 2.6 уже имеет поддержку так называемого soft-realtime, то есть работы в режиме вытесняющего, или приоритетного (preemptive), планирования, что помогает достичь хороших результатов при обслуживании большого количества процессов или во время выполнения активных дисковых операций. Под результатами подразумеваются такие характеристики как отклик (response) ядра на какой-либо системный вызов или задержка (latency) в его работе, связанная с нагрузкой или реализацией алгоритма. Более того, существует ряд патчей, которые позволяют ядру работать в режиме так называемого hard-realtime, то есть с гарантированным откликом системы при любой загрузке. Для простоты восприятия будем называть soft-realtime «легким» realtime, а hard — «тяжелым». И пусть вас не пугает частота употребления слова «патч» — ведь на них держится вся POSIX/Unix-система, а мы говорим здесь о его сердце, то есть ядре.

«Легкий» realtime

| Патчи от Con Kolivas |

Это наиболее популярные патчи, которые используются большинством дистрибутивов (как минимум это Debian, Gentoo, SUSE, Fedora, ALT Linux) в своих как бы realtime-ядрах. Еще их

очень любят всевозможные «ядроделатели», например nitro sources или СКО patchset. Патчи -ck позволяют добиться большей «отзывчивости» ядра при работе со значительными объемами вычислений, передаче данных или активной деятельности системы ввода/вывода.

Их отличительные особенности:

- ▶ Полностью изменена работа планировщика ядра, теперь можно более гибко регулировать, что и как планировать и с каким приоритетом.
- ▶ Добавлен новый вид планирования — Batch (idle) scheduling, позволяющий перепланировать процессы таким образом, что они выполняются только во время простоя (idle) системы. Это очень удобно для таких задач как компиляция или длительных операций наподобие apt-get dist-upgrade. Грубо говоря, процесс выполняется «в фоне», не мешая другим.
- ▶ Добавлен новый вид планировщика — SCHED_ISO, позволяющий добиться soft-realtime, то есть возможностей работы приложения, приближенных к realtime (на самом деле там введено ограничение на загрузку процессора в режиме realtime). Этот планировщик доступен не только привилегированным процессам, но и обычным (non-root) приложениям.
- ▶ Интересная вещь — swp prefetch. Вместо бездумного сбрасывания страниц памяти в swp создает связанный список того, что уходит в файл подкачки, и, если система находится в состо-

янии проста, возвращает эти страницы в память в обратном порядке. Главная идея такого поведения — если программа не использовалась долгое время и была переведена в swar, то вам не придется ждать ее долгого пробуждения с диска.

| Патчи от MontaVista |

Один из первых патчей, превращающих ядро действительно в realtime-preemptive (то есть с вытесняющей многозадачностью, приближенной к realtime). Он разработан Sven-Thorsten Dietrich на базе патча, написанного Инго Молнером (о нем мы расскажем чуть позже). Главная идея — запросы на прерывание помещаются в отдельные нити исполнения (threads), которые борются за процессор наравне с основными процессами. Как только такое прерывание добивается своей цели, его нить исполнения становится вытесняющей, то есть может быть вытеснена другим процессом, с большим приоритетом, или по истечении времени занятия ресурса, что позволяет обслужить больше прерываний, чем при обычном монопольном блокировании/разблокировании ядра.

Основная причина всех задержек в ядре — это критические секции в коде, защищенные механизмами блокирования (spinlocks). Поэтому рождается резонный вопрос: а почему не сделать и эти секции вытесняющими? Для ядер 2.4 уже есть реализация — это PMutex, разработанный группой исследователей из Мюнхенского университета, который и был взят за основу для ядра 2.6. Конечно, сразу же появились проблемы реализации — в версии PMutex был применен простейший механизм наследования приоритетов: как только процесс устанавливал флаг блокировки (mutex), его приоритет увеличивался, и он мог быстрее выполнить свою задачу и освободить mutex. Вместе с другими улучшениями это позволяло избежать многих проблем с искажением приоритетов при вытесняющем планировании. В версии от MontaVista был применен другой подход — все существующие механизмы блокирования были заменены на новые флаги блокировки (этот механизм был похож на существующую реализацию семафоров в ядре). То есть все критические секции в ядре стали вытесняющими.

К тому же, перенеся все запросы на прерывание в отдельные нити исполнения, решили еще одну проблему: теперь не возникает тупиковых ситуаций, связанных с прерываниями.

К сожалению, не обошлось без ложки дегтя. Теперь надо пересмотреть весь код ядра, где используются блокировки, и проверить его работоспособность (если что-то не работает, вернуть семафоры и spinlock на место). К тому же не все компоненты ядра должны быть вытесняющими — например, сложно представить себе вытесняемый планировщик ядра. Но разработчики MontaVista дали хороший старт всем проектам soft-realtime, доказав, что ядро Linux вполне можно сделать таким, каким нужно.

| Патчи от Инго Молнера |

Успех MontaVista заинтересовал Инго Молнера, одного из известных «ядерных хакеров». История гласит, что он пропал на два дня и потом выложил свою реализацию, названную впоследствии voluntary preemption, а затем realtime preemption. Инго согласился с основной идеей патча от MontaVista, но с некоторыми изменениями:

► PMutex не применялся. Вместо него Инго использовал существующий механизм семафоров в ядре. Как он сам говорил впоследствии, объясняя свое решение, семафоры работают на всех архитектурах, а PMutex — только на x86. Было бы здорово реализовать наследование приоритетов через семафоры, тогда это будет доступно и решениям на базе семафоров, и решениям, переписанным из spinlock. Однако в реализации voluntary preemption этого так и не было сделано.

► Вместо переписывания кода на новый механизм блокировки был предложен интересный вариант — функции блокировки не изменялись, менялась только модель их поведения: теперь они могли быть обычными блокировками, или «хитрыми» семафорами.

► Инго выяснил, что довольно много блокировок должны использовать старый механизм, а не новые «полиморфные» блокировки. В его патче сохранилось около 90 «старых» блокировок (в версии MontaVista — 30), но благодаря «полимор-

Проект RTnetwork

Сетевой realtime

RTnetwork (www.rtnet.org) — это попытка решить проблемы задержек и коллизий в сети Ethernet при передаче данных с помощью сетевого стека, специально разработанного для условий строгого реального времени. Отличительные особенности:

- реализация Open Source, основанная на открытых протоколах и стандартах;
- используется свой сетевой стек, отличный от стандартной реализации TCP/IP, интегрирован в инфраструктуру ядра Linux;
- используется Ethernet как транспортная технология;
- используется TDMA (Time Division Multiple Access) — улучшенный протокол для контроля передачи данных;
- используется специализированный заголовок для TDMA-передачи — RTmac;
- реализовано как расширение для RTAI (Real-Time Application Interface).

Интеграция RTnetwork с FireWire

FireWire, также известная как IEEE-1397, — это высокопроизводительная последовательная шина для соединения разобщенных (heterogeneous) устройств. Предназначенная в основном для передачи данных в бытовой электронике (например, высокоскоростная передача видеоданных), она может быть с успехом применена в промышленных или научных целях. Поддержка RTnetwork для FireWire под ОС Linux во многом повторяет реализацию RTnetwork для сетевого стека. Общение FireWire стека и ядра RTnetwork реализовано через Ethernet-эмуляцию, то есть передаваемые пакеты преобразуются в Ethernet-пакеты для RTnetwork и FireWire-пакеты для стека FireWire.

Таким образом, FireWire работает так же, как и все остальные Ethernet-устройства в RTnetwork.

фичности» нового блокирования его патч получился гораздо меньше, чем у MontaVista.

Конечно, работа над патчем продолжается (и очень бурно — Инго выпускает по одной новой версии патча в неделю). Не все пока гладко, но надеемся, что все у него получится, а патч все-таки войдет в основное ядро Linux.

«Тяжелый» realtime

Теперь перейдем к проектам, цель которых — сделать из Linux настоящее realtime-ядро. «Интересно, чем их не устраивают, например, RT-патчи?» — спросите вы. А вот чем: все реализации soft-realtime не дают заданных (или детерминированных) параметров задержек, они лишь помогают их снизить. Realtime-система с негарантированными задержками, конечно, может сводить звуковые дорожки, но вот управлять атомной электростанцией, быть бортовой системой истребителя-перехватчика или просто супербыстрым IRC-спамботом — вряд ли. И вообще, настоящий realtime должен начинаться с отклика в несколько микросекунд. Существует пара проектов, которые реализуют hard-realtime на ядре Linux. Это RTLinux и RTAI/fusion.

RTLinux

Можно назвать этот проект паршивой овцой среди стада Open Source — реализация закрыта патентом на ПО. Основная идея RTLinux (и суть патента) — все аппаратное обеспечение функционирует под управлением маленького realtime-ядра, а само ядро Linux вместе с его процессами и задачами является лишь одним из низкоприоритетных процессов realtime-ядра. С микроядром можно общаться через специальный модуль, имеющий ограниченный набор команд. Для общения микроядра с прикладным ПО существуют так называемые каналы связи микроядра и ядра Linux. Таким образом, realtime-микроядро контролирует все аппаратное обеспечение, и можно гарантировать работу в реальном времени всей системы в целом.

RTAI/fusion

Разработчики из RTAI сначала хотели пойти по пути RTLinux — то есть изолировать основное ядро и ядро Linux друг от друга, но потом решение изменилось в сторону ядра Adeos. Это некое «суперядро», в котором realtime-ядро и классическая Linux выполняются как обычные процессы, что позволяем им свободно общаться между собой. То есть, мы используем множество вариантов связи между системами, чтобы realtime-компонент ответил первым на запросы оборудования, что является более гибким решением, нежели подход RTLinux. К сожалению, ядерный модуль для доступа к realtime-компонентам ядра все равно необходим, что усложняет реализацию пользовательского ПО.

Для решения этих проблем Филипп Джером, один из разработчиков ядра Adeos, предложил свою реализацию, названную RTAI/Fusion (впоследствии Xenomai). Она позволила применять обычные приложения Linux, использующие стандартные вызовы ядра, но при этом время отклика (response time) этих приложений было гарантированным, то есть при-

ближенным к требованиям hard-realtime. RTAI/fusion может работать в двух режимах:

- ▶ «Усиленный» — гарантирует более низкие задержки, но сама программа должна зависеть только от возможностей, предоставляемых API RTAI. Некоторое количество системных вызовов ядра Linux реализуется RTAI, но не все.
- ▶ Когда выполняемый процесс запрашивает системный вызов, который не может быть реализован в «усиленном» режиме, то система переводится в следующий режим, называемый «защищенным». Он похож на реализации от MontaVista или Инго Молнера, то есть доступны все вызовы ядра, но задержки тоже выше. «Защищенный» режим запрещает все запросы на прерывание во время выполнения realtime-задачи, что может также увеличивать задержки.

Процессы могут выполняться как в «усиленном», так и в «защищенном» режиме — все зависит от требований этих процессов и реализации данных требований в RTAI.

Перспективы

Как правило, в заключение подобных обзоров пишут о том, когда тот или иной патч появится в основном ядре Linux, но пока ни один из описанных патчей (за исключением, может быть, модификаций -ск) не был принят большинством разработчиков ядра. Основные проблемы — это стабильность, которая может пострадать, и объем кода, который надо переписать. Возможно, все эти инициативы станут хорошим катализатором для открытия ветки 2.7, а может, и нет, так как на проверку этих патчей и их дальнейшее улучшение и развитие необходимо время. |

Дополнительная информация

Ссылки по теме

Цикл заметок о realtime-возможностях Linux на LWN:

- ▶ <http://lwn.net/Articles/106016>;
- ▶ <http://lwn.net/Articles/106010>;
- ▶ <http://lwn.net/Articles/129511>;
- ▶ <http://lwn.net/Articles/108216>;
- ▶ <http://lwn.net/Articles/96494>;
- ▶ <http://lwn.net/Articles/105866>;
- ▶ <http://lwn.net/Articles/129211>.

Подборка патчей от Con Kolivas для ядра 2.6:

- ▶ <http://members.optusnet.com.au/ckolivas/kernel>.

Подборка патчей от Con Kolivas для ядра 2.4:

- ▶ www.plumlucosoft.com/kernel.

Linux + PMutex для ядра 2.4:

- ▶ <http://inf3-www.informatik.unibw-muenchen.de/research/linux/mutex>;
- ▶ http://inf3-www.informatik.unibw-muenchen.de/research/linux/hannover/automation_conf04.pdf.

RTAI/fusion/Xenomai:

- ▶ www.rtai.org;
- ▶ <http://snail.fsffrance.org/www.xenomai.org>;
- ▶ www.rtnet.org.

Степан Заступов

Завидный конкурент bash

Наверное, каждый пользователь под понятием shell подразумевает именно bash. Не спору, эта командная оболочка действительно не такая плохая, иначе ее не стали бы использовать по умолчанию в большинстве Linux-дистрибутивов. Однако, будь она действительно идеальна, не существовало бы альтернативных программных продуктов. Я не говорю про csh/tcsh, так как у него совершенно другой путь развития, и он уже стал привычной оболочкой для многих BSD-администраторов. Но зачем использовать то, что установлено по умолчанию, если есть альтернатива? Имя ей — zsh!

Zsh — пожалуй, самая «навороченная» и динамично развивающаяся оболочка. От аналогичных продуктов ее отличают гибкая конфигурация, удобные алиасы, модульная структура и самая лучшая система автоматического дополнения. Вы можете сказать, что у bash есть и алиасы, и автодополнение,

```
[~] % cd articles/му/песни\ и\ стихи [17:45]
[~/articles/му/песни и стихи] % vim песочная\ кровь [17:46]
20\ \шверта\ двадцати\ стел\
бег\ от\ судьбы
выгляд\ на\ сердце
зачем\ жить?
зов\ Ктулху
красные\ слёзы\ горгульи
ледяная\ могила
летаргия
лики\ смерти
мясорубка
осколки\ ЗАВТРАКА
пассивный\ каннибализм\ \ (теория\ самоуничтожения\
песочная\ кровь
последовательность
пиль
ржавое\ небо
сколько\ стоит\ твоя\ душа?
спасибо\ тебе\ за\ повод\ умереть
теледильдоника
это\ часть\ меня
я\ иду...
```

Перемещаемся по вариантам дополнений

его можно улучшить с помощью bash-complition, но, дочитав статью до конца, вы убедитесь, что zsh оставляет всех далеко позади. Стоит ли говорить, что zsh заменил мне все файловые менеджеры — как графические (вроде Krusader), так и пресловутый mc? Думаю, начинать лучше постепенно, с первых строчек конфигурации.

Приглашение

Главный конфигурационный файл .zshrc должен храниться в вашем домашнем каталоге. Есть также файлы .zlogin и .zlogout, которые выполняются во время входа и выхода из системы соответственно.

Так как многие системные переменные, установленные поставщиком дистрибутива, находятся в файле /etc/profile, его можно запустить с помощью команды source /etc/profile. Как уже замечалось, zsh имеет модульную структуру, и нужные модули также следует загрузить:

```
autoload -U compinit
compinit
autoload -U incremental-complete-word
```

```
[~/video/Cradle of filth] x
```

```
[16:18]
```

```
bash и не снились такие PS1
```

```
zle -N incremental-complete-word
autoload -U insert-files
zle -N insert-files
autoload -U predict-on
zle -N predict-on
```

Теперь можно приступить к настройке PS1. Вообще, в zsh традиционно принято использовать переменную PROMPT, но PS1 тоже будет работать.

Вот основные переменные, которые могут быть использованы в PROMPT:

- ▶ `%{\e[1;32m%}` — цветовой блок; код цвета — как и у bash, в данном случае зеленый;
- ▶ `%M` — полное имя хоста машины;
- ▶ `%m` — имя хоста без домена;
- ▶ `%n` — имя пользователя;
- ▶ `%~` — текущая директория;
- ▶ `%T` — текущее время в 24-часовом формате.

Если вам этого мало (а так, скорее всего, и есть), посмотрите документацию. Стоит обратить внимание на то, что у zsh может быть так называемый «двусторонний» PROMPT! У меня, к примеру, с левой стороны отображается текущая директория, а с правой — время:

```
PROMPT='${\e[1;32m%}[%{\e[1;34m%}~%{\e[1;32m%
}]{\e[1;31m%}##%{\e[0m%} '
RPROMPT='${\e[1;32m%}[%{\e[1;33m%}%T%{\e[1;32m%
}]{\e[0m%}'
```

Все это, конечно, замечательно, но большинство пользователей почти всегда работают в графической среде, и было бы не плохо отображать некоторую информацию в заголовке окна терминала. К примеру, помимо пути вы можете захотеть наблюдать текущую команду, выполняемую в терминале. Нет проблем! В zsh присутствуют две специальные функции: выполняющаяся перед каждой командой `precmd`, а также `preexec`, которая будет вызвана после выполнения команды. Ими мы как раз и воспользуемся:

```
precmd()
{
    [[ -t 1 ]] || return
    case $TERM in
        *xterm*|rxvt|(dt|k|E)term*) print -Pn
        "\e]2;[%~] :: %l\a"
        ;;
    esac
}
preexec() {
    [[ -t 1 ]] || return
    case $TERM in
        *xterm*|rxvt|(dt|k|E)term*) print -Pn
        "\e]2;<$1> [%~] :: %l\a"
        ;;
    esac
}
```

История

Нет, в этой рубрике вы не найдете рассказ о том, как создавался zsh. Речь пойдет о настройке истории команд.

Определить файл истории можно следующим образом:

```
HISTFILE=~/.zhistory
```

Число команд, сохраняемых в HISTFILE:

```
SAVEHIST=5000
```

Число команд сохраняемых в сеансе:

```
HISTSIZE=5000
```

Лично мне этих значений хватает, а вот историю нужно еще включить:

```
setopt APPEND_HISTORY
```

Меня не удовлетворяют повторяющиеся команды, лишние пробелы и пустые строки в истории. Эта проблема устраняется следующим образом:

```
setopt HIST_IGNORE_ALL_DUPS
```

```
setopt HIST_IGNORE_SPACE
```

```
setopt HIST_REDUCE_BLANKS
```

Алиасы

Обычными алиасами никого не удивишь, тем не менее приведу пару примеров:

```
alias mkisofs='mkisofs -J -joliot-long -l -max-iso9660-file-
names'
```

```
alias cdrecord='cdrecord -v -dev=1,0,0'
```

Теперь самое интересное. В любом файловом менеджере можно привязать программу к расширению или MIME-типу. Из-за этого многие пользуются слабеньким `mc` или же очень «тяжелыми» (в плане используемых ресурсов) `Konqueror` и `Nautilus` и не хотят попробовать что-то другое. Но zsh тоже умеет назначать алиасы по расширению:

```
alias -s {avi,mpeg,mpg,mov,m2v}=mplayer
```

```
alias -s {odt,doc,sxw,rtf}=openoffice.org
```

```
alias -s {ogg,mp3,wav,wma}=beep-media-player
```

```
alias -s pdf=xpdf
```

Все, теперь bash точно начинает сдавать позиции. А вот для браузеров разработчики написали отдельный модуль:

```
autoload -U pick-web-browser
```

```
alias -s {html,htm}=pick-web-browser
```

Скрипт поддерживают такие браузеры как Opera, Konqueror, Lynx, Mozilla и другие. Думаю, вы сами при желании сможете найти этот скрипт и заменить Mozilla на любимый Firefox.

Дополнение

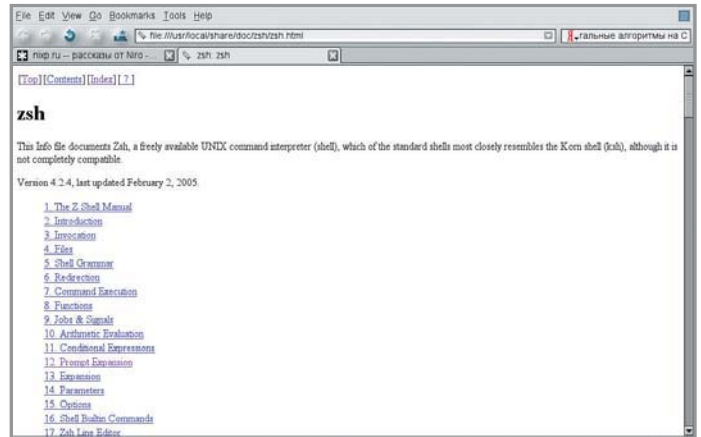
Теперь перейдем к самому интересному. Ни один shell не имеет такой сильной поддержки автодополнений, как zsh. Стандартные функции, заключающиеся в дополнении имен директорий и команд, есть у всех, а как насчет опций определенной программы?

Bash имеет такую возможность, но только за счет установки пакета `bash-complition`, который будет достаточно сильно

```

<sudo make kernel> [/usr/src] :: pd
awk -f @/tools/makeobj.jobs.awk @/kern/device_if.m -h
awk -f @/tools/makeobj.jobs.awk @/kern/bus_if.m -h
awk -f @/tools/makeobj.jobs.awk @/dev/pci/pci_if.m -h
rm -f .depend
CC="/usr/local/libexec/ccache/cc" makedep -f .depend -a -nostdinc -D_KERNEL -DXL
D_MODULE -I -DHAUE_KERNEL_OPTION_HEADERS -I -Ie -Ie/contrib/altq -Ie/./includ
e -I/usr/include -I/usr/obj/usr/src/sys/REDCHROMPC /usr/src/sys/modules/amr/./
./dev/amr/and.c
==> amr (depend)
@ -> /usr/src/sys
machine -> /usr/src/sys/i386/include
awk -f @/tools/makeobj.jobs.awk @/kern/device_if.m -h
awk -f @/tools/makeobj.jobs.awk @/kern/bus_if.m -h
awk -f @/tools/makeobj.jobs.awk @/dev/pci/pci_if.m -h
ln -s /usr/obj/usr/src/sys/REDCHROMPC/opt_cam.h opt_cam.h
ln -s /usr/obj/usr/src/sys/REDCHROMPC/opt_scsi.h opt_scsi.h
rm -f .depend
CC="/usr/local/libexec/ccache/cc" makedep -f .depend -a -nostdinc -D_KERNEL -DXL
D_MODULE -I -DHAUE_KERNEL_OPTION_HEADERS -I -Ie -Ie/contrib/altq -Ie/./includ
e -I/usr/include -I/usr/obj/usr/src/sys/REDCHROMPC /usr/src/sys/modules/amr/./
./dev/amr/amr.c /usr/src/sys/modules/amr/././dev/amr/amr_pci.c /usr/src/sys/mo
dules/amr/././dev/amr/amr_disk.c /usr/src/sys/modules/amr/./././dev/amr/amr_ca
n.c
    
```

Сигнал с микрофона слишком слабый, поэтому придется немного с ним поработать



Как ни крути, чтобы воспользоваться всей мощью zsh, придется читать документацию

тормозить саму оболочку. Zsh уже «из коробки» может дополнять команды и директории, опции огромного количества программ и сразу представлять все это в виде меню! Я уже не говорю про гибкость настройки системы дополнений и скорость, с которой работает сама оболочка, несмотря на потрясающую функциональность. Как всегда не забываем подгрузить некоторые необходимые модули:

```

zmodload -a zsh/stat stat
zmodload -a zsh/zpty zpty
zmodload -a zsh/zprof zprof
zmodload -ap zsh/mapfile mapfile
    
```

А теперь можно смело переходить к непосредственной настройке системы дополнений. Чтобы определить опции нужной функциональности, необходимо вызвать функцию `zstyle` с указанием подсистемы и ее параметрами.

Скучно смотреть на монохромную консоль, в то время как команда `ls`, например, может подсвечивать все объекты файловой системы разными цветами. Этого можно добиться и с помощью `zsh` — благодаря опции `list-colors`, которая принимает тот же синтаксис указания цветов, что и GNU `ls`. Следовательно, можно использовать переменную `$LS_COLORS`:

```
zstyle ':completion:*:default' list-colors '${LS_COLORS}'
```

Но так как я работаю во FreeBSD, а установленная там утилита `ls` использует другой синтаксис, пришлось непосредственно указать цвета:

```

zstyle ':completion:*:default' list-colors
'no=00:fi=00:di=01;34:ln=01;36:pi=40;33:so=01;35:do=01;3
5:bd=40;33;01:cd=40;33;01:or=40;31;01:ex=01;31:'
    
```

Далее выберем некоторые опции, которые будут использоваться по умолчанию:

```

zstyle ':completion:*' completer _complete _list _oldlist _
expand _ignored _match _correct _approximate _prefix
zstyle ':completion:*' insert-unambiguous true
zstyle ':completion:*' add-space true
    
```

Дополнения есть и к командам `kill` и `killall`. Они идут под общей подсистемой `processes`. К примеру, можно указать команду для получения списка процессов:

```

zstyle ':completion:*:processes' command 'ps -xuf'
zstyle ':completion:*:processes-names' command 'ps xho command'
    
```

Или же отключить их сортировку:

```
zstyle ':completion:*:processes' sort false
```

Если хочется перейти в другую директорию, находящуюся, к примеру, в предыдущей, то наличие в вариантах текущей директории нежелательно:

```
zstyle ':completion:*:cd:*' ignore-parents parent pwd
```

А вот таким образом можно ограничить число ошибок на каждый символ дерева дополнений до одной:

```
zstyle -e ':completion:*:approximate:*' max-errors 'reply=( $((
($#PREFIX+$#SUFFIX)/3 )) numeric )'
```

Для повышения скорости работы пользователя лучше отключить различие регистров:

```
zstyle ':completion:*' matcher-list 'm:{a-z}={A-Z}'
```

Можно и игнорировать внутренние функции дополнений:

```
zstyle ':completion:*:functions' ignored-patterns ' _*'
```

Как я уже говорил, можно включить меню выбора вариантов дополнения. К примеру, вы хотите открыть один из нескольких файлов, а они на русском языке — лень переключать раскладку. Просто пишете команду (к примеру, `vim`), затем пробел, два раза нажимаете кнопку «Tab» — и можете перемещаться по вариантам (в данном случае — файлам) клавишами стрелок или с помощью той же «Tab» и выбирать их! Вот несколько строк, которые позволяют все это проделать:

```

zstyle ':completion:*' menu select=long-list select=0
zstyle ':completion:*' old-menu false
zstyle ':completion:*' original true
zstyle ':completion:*' substitute 1
zstyle ':completion:*' use-compact true
zstyle ':completion:*' verbose true
zstyle ':completion:*' word true
    
```

Напоследок

Возможно, у вас возникнут проблемы с распознаванием клавиш. Это решается простым переназначением клавиш в `zsh`. Почитайте по этому поводу документацию.

Надеюсь, теперь у вас не осталось сомнений в том, что лучше оболочки, чем `zsh`, попросту не существует, что не обязательно использовать файловые менеджеры для удобства работы, а стремление к аскетизму может привести к удобству. |

Apache & Co

Сегодня разработчику приложений, использующих веб-технологии, необходимо разбираться в настройках Apache, MySQL, PHP, Perl и прочих продуктов ничуть не хуже системного администратора. При этом время, которое обычно тратится на поиск, загрузку и установку, могло бы быть использовано более рационально.

Вообще-то, сегодня практически во всех дистрибутивах GNU/Linux уже имеются необходимые для работы компоненты. В большинстве случаев они готовы к использованию сразу после установки. Разработчику остается только доустановить недостающие приложения или модули. То есть, фактически мы возвращаемся к самому началу, не говоря уже о необходимости периодического обновления всех компонентов; да и некоторые модули требуют перекомпиляции Apache. Но если разработчик использует Windows, то ему вообще придется собирать всю систему с нуля, попутно разбираясь в дебрях конфигурационных файлов и опций настроек. К тому же эти файлы придется редактировать вручную, что очень неприятно пользователям данной системы.

Естественно, эта проблема не осталась незамеченной, и в настоящее время известно несколько проектов, задачей которых является упаковка всех необходимых элементов в единый модуль, который может быть легко установлен, обновлен и также без проблем удален, если в нем уже не будет необходимости. Среди таких проектов наибольшей популярностью пользуются «Денвер» (www.dklab.ru), AppServ (www.appserv-network.com) и XAMPP (www.apachefriends.org). Каждый из них, казалось бы, похож на остальные, и в первую очередь — основой, которой во всех этих проектах является связка Apache+PHP+MySQL. Но дополнительно кроме базового набора приложений каждый из этих проектов предлагает расширения, позволяющие нарастить возможности и настроить систему под индивидуальные потребности веб-мастера. Если приглядеться повнимательнее, можно увидеть и отличия. Так, «Денвер» предназначен для разработки сайтов в Windows, но с расчетом на то, что в дальнейшем в качестве системы будет использоваться Unix. Интересно, что базовый набор компонентов в «Денвере», имеющий размер всего 3 Мбайт, включает в себя около 30 Мбайт приложений, которые пользователю пришлось бы самостоятельно качать

из Интернета в случае индивидуальной установки. Этого удалось достичь за счет изъятия некоторых редко используемых компонентов (все они вынесены в модули расширений), а также компрессии.

Еще одной особенностью является эмулятор sendmail, позволяющий «отправлять» письма. Кроме того, существует дискетная версия «Денвера». Для разработчиков, использующих Windows, предназначен и AppServ. В отличие от остальных он не содержит Perl, но интересен дополнительными модулями, среди которых phpBB, Zend Optimizer, Mambo, Drupal и другие. А вот XAMPP выделяется среди них не только большим количеством компонентов, входящих в базовый состав (правда, и размер поэтому достигает 40 Мбайт). Здесь присутствуют Apache, MySQL, PHP, Perl, ProFTPD, phpMyAdmin, OpenSSL, GD, Freetype2, libjpeg, libpng, gdbm, zlib, expat, Sablotron, libxml, Ming, Webalizer, pdf class, ncurses, mod_perl, FreeTDS, gettext, mcrypt, mhash, Turck MMCache, SQLite и IMAP C-Client. В будущем планируется добавить Java 2 SDK и Tomcat. Представьте, что бы было, если бы все это пришлось устанавливать самостоятельно.

XAMPP очень просто обновлять, и самое главное — имеются версии для различных систем. На сегодняшний день их четыре:

- ▶ для Linux (разработчиками тестировались версии под SUSE, Red Hat, Mandrake и Debian, автор делал это также под ASPLinux, ALT Linux, VectorLinux);
- ▶ для Windows 98, NT, 2000, 2003 и XP;
- ▶ бета-версия для Solaris SPARC;
- ▶ бета-версия для Mac OS X.

Распространяется XAMPP по лицензии GNU GPL, хотя при коммерческом использовании стоит обратить внимание на лицензии отдельных продуктов, входящих в комплект.

Как уже говорилось, установка XAMPP довольно проста. Для Windows она традиционна и заключается в распаковке архива и запуске исполняемого файла. В случае установки в Linux также особых трудностей не возникает.

Скачиваем архив, распаковываем его:

```
# tar xvfz xampp-linux-1.4.13.tar.gz -C /opt
```

Вот и все. Теперь можно запускать (требуется привилегии root).

```
# /opt/lampp/lampp start
```

Starting XAMPP for Linux 1.4.13...

XAMPP: Starting Apache with SSL (and PHP4)...

XAMPP: Starting MySQL...

XAMPP: Starting ProFTPD...

XAMPP for Linux started.

Интересно, что в составе XAMPP имеется PHP версий 4 и 5.

Та, что нужна, активируется командой `php` с указанием номера.

```
# /opt/lampp/lampp php5
```

XAMPP: Activate PHP 5.0.4...

XAMPP: XAMPP-Apache is not running.

XAMPP: Starting Apache with SSL (and PHP5)...

XAMPP: Starting MySQL...

XAMPP: Starting ProFTPD...

XAMPP for Linux started.

Какая версия сейчас активирована, можно узнать при помощи `phpinfo()`, либо введя следующую команду:

```
# /opt/lampp/lampp phpstatus
```

Если нет необходимости в старте всех сервисов, можно запускать их поодиночке:

```
# /opt/lampp/lampp startapache
```

Для запуска с поддержкой SSL это будет выглядеть так:

```
# /opt/lampp/lampp startssl
```

Набираем в браузере `http://localhost` (или `https://localhost` — при запуске с поддержкой SSL) и смотрим на результат работы. Здесь можно ознакомиться со статистикой работы, компонентами, входящими в состав XAMPP, получить доступ к инструментам управления и сбора статистики PhpMyAdmin, webalizer и phpSQLiteAdmin, увидеть результат работы демонстрационных примеров. По умолчанию для операций с базами данных, FTP и прочей информацией XAMPP не использует пароли, а это небезопасно. Администратор MySQL не имеет пароля, и к демону можно получить доступ через сеть. Также доступен из сети PhpMyAdmin, а ProFTPD для пользователя nobody требует пароль lampp. Если вам предстоит работать через Интернет, обязательно используйте команду `security` и установите пароли, иначе ваш компьютер рискует стать легкой мишенью для сетевых вандалов. Установленный таким образом пароль применяется для доступа к тестовой странице, имя пользователя — `lampp`.

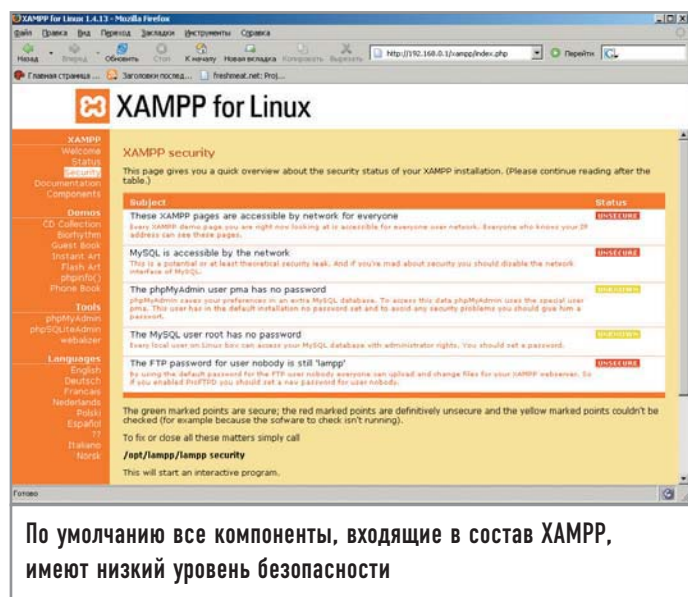
Для сохранения настроек, файлов данных и журналов используется команда `backup`. Среди дополнений на странице проекта имеется только Development package, содержащий инструменты, позволяющие скомпилировать недостающие компоненты самостоятельно. Готовые дополнения можно найти на сайте `http://addons.xampp.org`. Там хорошо организован поиск по группам (Apache, MySQL, PHP, Perl Mailserver, Misc) и версиям XAMPP.

Если необходим автоматический запуск XAMPP при загрузке системы, создайте символическую ссылку на используемый уровень запуска:

```
# egrep initdefault /etc/inittab
```

```
id:5:inittdefault:
```

```
# ln -s /opt/lampp/lampp /etc/rc5.d/S99lampp
```



Как уже говорилось, XAMPP можно обновлять. Это позволяет не скачивать заново весь пакет, к тому же все настройки и рабочие файлы будут сохранены. Для обновления необходимо скачать пакет, который называется `xampp-linux-upgrade`, и распаковать его (место распаковки значения не имеет):

```
$ tar xvfz xampp-linux-upgrade-1.4.13.tar.gz
```

Теперь дадим команду на обновление:

```
$ su
```

```
# xampp-upgrade/start
```

Вот и все. Начиная с версии 1.4.2 XAMPP позволяет создавать резервные копии. Для этого воспользуйтесь командой `backup` (или `backup mysql_password`):

```
# /opt/lampp/lampp backup
```

Backing up databases...

Backing up configuration, log and htdocs files...

Calculating checksums...

Building final backup file...

Backup finished.

Take care of /opt/lampp/backup/xampp-backup-22-09-05.sh

Файл `/opt/lampp/backup/xampp-backup-22-09-05.sh` содержит теперь резервные данные. Запустив следующий скрипт на компьютере с той же версией XAMPP, вы получите точную копию сохраненной рабочей среды:

```
# sh xampp-backup-22-09-05.sh
```

Checking integrity of files...

Restoring configuration, log and htdocs files...

Checking versions...

Installed: XAMPP 1.4.13

Backup from: XAMPP 1.4.13

Restoring MySQL databases...

Restoring MySQL user databases...

Backup complete. Have fun!

You may need to restart XAMPP to complete the restore.

После восстановления требуется перезапустить XAMPP. Кстати, для его удаления достаточно избавиться от каталога:

```
# rm -rf /opt/lampp
```

Теперь, когда имеется готовая к работе среда, можно смело приступать к созданию приложений. |

Одной левой. Кнопкой

Комфорт — дело техники. В том числе и той, с которой вы работаете каждый день. И если в Windows приходится пользоваться тем, что есть, то в Linux никто не мешает нам настроить систему под себя.

Не стоит даже говорить о том, что работа с клавиатурой для настоящего юниксоида — это как «Отче наш» для выпускника воскресной школы.

А уж если на этой клавиатуре есть дополнительные кнопки, то просто грех ими не воспользоваться.

Рабочий кабинет заполняли звуки «Лунной сонаты». Неожиданно раздался звонок телефона, и я потянулся к трубке, нажав на клавиатуре компьютера кнопку «Mute». Звонил начальник. Минут через двадцать, утерев пот со лба и отпив из кружки остывший к тому моменту чай, я облокотился на спинку кресла, поиграл в воздухе ручкой с фирменной символикой корпорации и, еще раз нажав на клавиатуре «Mute», а потом и «Volume up», клавишами «Prev», «Next» выбрал саундтрек к «Матрице» и принялся выполнять скромную работу системного администратора — вершить судьбы пользователей и править IT-миром.

Мультимедийные клавиатуры для персональных компьютеров и мультимедиакнопки на них получили широкое распространение относительно недавно. Но если под Windows они, как правило, выполняют свою роль довольно исправно, то на Linux-десктопах часто смотрят на нас с немым укором. А между тем им можно назначить массу полезных функций: перелистывать терминалы в консоли кнопками «Windows», управлять, вызывать различные приложения (браузеры, почтовые клиенты). Достаточно лишь произвести необходимые настройки.

Итак, для сегодняшнего эксперимента нам понадобится следующее: любая клавиатура с мультимедиаклавишами; программы `showkey`, `setkeycodes`, `loadkeys` (они входят в пакет `console-tools` большинства дистрибутивов), `xev`, а также `ermixer`

и `xmmsctrl` — если мы хотим задействовать управление плеером XMMS. Все описанные манипуляции производились на машине с дистрибутивом Mandrake Linux и установленным KDE, хотя могут применяться к любому дистрибутиву с другим оконным менеджером.

Настройка, часть первая. X Window

Для начала запускаем в console от имени root программу `xev`. Перед нами появляется окошко «Event Tester» с квадратом внутри. Помещаем курсор мыши внутрь него и последовательно нажимаем мультимедиаклавиши. В результате в окне терминала отображаются следующие события, соответствующие нажатию на определенную кнопку:

```
KeyPress event, serial 30, synthetic NO, window 0x3e00001,
root 0x48, subw 0x0, time 11616554, (170,-16), root:(173,34),
state 0x0, keycode 160 (keysym 0xffd0, F19), same_screen YES,
XLookupString gives 0 bytes:
```

```
XmbLookupString gives 0 bytes:
```

```
XFilterEvent returns: False
```

```
KeyRelease event, serial 30, synthetic NO, window 0x3e00001,
root 0x48, subw 0x0, time 11616673, (170,-16), root:(173,34),
state 0x0, keycode 160 (keysym 0xffd0, F19), same_screen YES,
XLookupString gives 0 bytes:
```

Из всей полезной информации, которая тут имеется, нас интересует значение keycode. Как видим, для данной мультимедиа-кнопки это 160. Для активации этой клавиши необходимо в скрипт автозапуска KDE добавить ее маппинг на некую виртуальную клавишу. Помещаем в `~/kde/Autostart/скрипт keys.sh`:

```
xmodmap -e 'keycode 160 = F19'
```

Таким образом, «медийную» кнопку мы превратили в виртуальную клавишу «F19». Аналогичным образом активируем все оставшиеся кнопки. Теперь для реализации какого-либо действия с их помощью достаточно запустить Центр управления KDE и настроить в нем привязку действий. Для настройки громкости мы используем утилиту `ermixer`, запуская ее со следующими параметрами для уменьшения и увеличения звука соответственно:

```
ermixer vol -1
```

```
ermixer vol +1
```

Плеером XMMS можно управлять с помощью утилиты `xmmsctrl`, настроив мультимедиа-клавиши на переключение треков и управление воспроизведением:

```
xmmsctrl next
```

```
xmmsctrl prev
```

```
xmmsctrl play
```

```
xmmsctrl stop
```

| Настройка, часть вторая. Консоль |

Как известно, при работе в консоли Linux в текстовом режиме пользователю доступны шесть терминальных сеансов, переключаемых кнопками «Alt+F1... F6». Для большего удобства я предлагаю назначить их переключение на не используемые в консоли клавиши. Для клавиатур с кнопками вызова стартового и контекстного меню Windows такая возможность изначально предусмотрена в любом Linux-дистрибутиве. Но, скажем, на ноутбуках от IBM кнопки Windows отсутствуют, однако их роль с успехом могут сыграть другие клавиши.

Для их подключения запускаем от имени `root` `showkeys -s` и последовательно нажимаем клавиши, которые будут выполнять перелистывание консолей. Сначала левую, потом правую:

```
# showkey -s
```

```
0xe0 0x20
```

```
0xe0 0xa0
```

```
0xe0 0x6f
```

```
0xe0 0xef
```

Основная сложность состоит в том, что рассматриваемые дополнительные клавиши при нажатии генерируют не сканкоды, как обычно, а последовательность сканкодов, начинающуюся с `e0`. Первая последовательность кодов, так называемый Make Code, отвечает за нажатие клавиши. Вторая последовательность, Break Code, — за ее отпускание.

Для того чтобы внести в раскладку клавиатуры, загружаемую программой `loadkeys`, какую-либо клавишу, надо знать ее `keycode` — число от 0 до 127, характеризующее только эту клавишу. Утилита `setkeycodes` как раз связывает некоторую заданную последовательность сканкодов с указанным `keycode`. Итак, в скрипте инициализации клавиатуры (в моей системе это `/etc/init.d/keytable`) я написал две строчки:

```
setkeycodes e020 113
```

```
setkeycodes e06f 112
```

В данном случае `e020` — это последовательность, сгенерированная при нажатии на левую клавишу переключения консоли, `e06f` — на правую, как видно из листинга `showkey`.

Затем необходимо добавить в раскладку клавиатуры для консоли (обычно она находится в `/usr/share/keymaps`) две строчки, которые активируют возможность перемещаться по консолям вправо-влево:

```
keycode 112 = Incr_Console
```

```
keycode 113 = Decr_Console
```

Как видим, нет ничего особо сложного в том, чтобы заставить «лишние» кнопки на клавиатуре работать на вас. И единожды настроив их, вы сэкономите массу времени в дальнейшем. |



«Привяжите» к мультимедийным клавишам самые любимые программы и команды, и вы сэкономите массу времени

Невидимые связи

Под Linux работает практически любой Wi-Fi-адаптер. Так что можно выбирать самое доступное по цене решение (конечно же, из устройств класса 802.11g, потому что во многих магазинах еще продаются адаптеры устаревшего стандарта 802.11b). Но готовьтесь к тому, что не все они заработают автоматически — возможно, придется приложить некоторые дополнительные усилия.

Вообще-то, дела с поддержкой Wi-Fi-оборудования в Linux обстоят довольно странно. Даже в самом свежем ядре поддержка беспроводных сетевых карт довольно слабая: хорошо работают лишь адаптеры на чипах Aviator/Raytheon, Intersil HFA384, Intersil Prism GT, Amtel at76c50x, Cisco Aironet 34x/35x/4500/4800 и Planet WL3501. Причем, как показала практика, при покупке узнать чип, на котором создан адаптер, практически невозможно. Тем более что производители беспроводного оборудования имеют дурную привычку выпускать одну и ту же модель на разных чипах. Так, например, популярный PCMCIA Wi-Fi-адаптер D-Link DWL-G650+ выпускался в трех различных вариантах: на чипах ACX100, ACX111 (Rev. A1) и TI1130 (Rev. B). Естественно, никаких упоминаний о том, к какой ревизии принадлежит ваша карта, на коробке вы не обнаружите.

Заставить нормально работать свободный драйвер, как оказалось, не так уж и просто, однако в большинстве случаев этого и не нужно.

Существует одно универсальное решение, о котором я и хотел бы рассказать, — это ndiswrapper.

| Приручаем Wi-Fi-адаптеры |

Ndiswrapper — это модуль к ядру Linux, позволяющий использовать NDIS-драйвер для беспроводных сетевых карт, написанный для Windows 2000/XP/2003. Да, с ndiswrapper можно заставить работать практически любой Wi-Fi-адаптер и без проблем использовать WEP-шифрование.

Со списком протестированных (и работающих) адаптеров можно ознакомиться на сайте <http://ndiswrapper.sourceforge.net/mediawiki/index.php/List>. Если нужной модели в списке не оказалось, не отчаивайтесь — скорее всего, она работает, и вы сами сможете добавить ее в этот перечень.

| Установка |

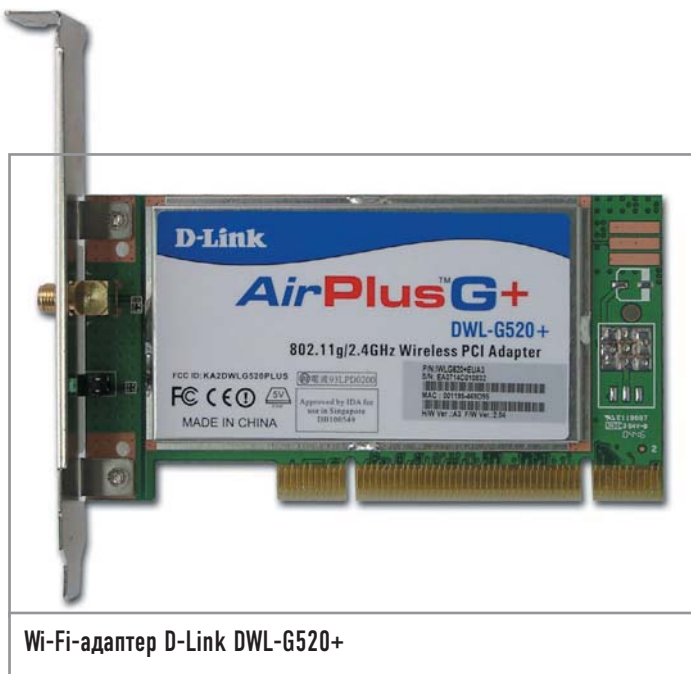
Прежде чем бросаться в бой, выпейте чашечку кофе и заодно посмотрите — быть может, ndiswrapper уже есть в вашем дистрибутиве? Если так, то вам повезло: переходите сразу к следующему разделу.

Для установки ndiswrapper потребуются компилятор GCC и утилита GNU make. Обратите внимание, что ndiswrapper может работать только на новых ядрах (версий выше 2.4.26 и 2.6.6). Если у вас более старое ядро, рекомендую обновить систему.

Также потребуются исходные тексты вашего ядра, причем их желательно брать из дистрибутива, так как практически каждый дистрибутив обладает своим ядром с собственным набором патчей. Версию ядра можно узнать с помощью команды `uname -r`.

После установки исходных текстов ядра убедитесь, что каталог `/lib/modules/<версия_вашего_ядра>/build` является ссылкой на директорию с исходными текстами:

```
# ls -l /lib
lrwxrwxrwx 1 root root 21 2005-11-14 16:56 /lib/modules/2.6.12/build -> /usr/src/linux-2.6.12
```



Wi-Fi-адаптер D-Link DWL-G520+

Если нет, то следует поправить ссылку, иначе ndiswrapper просто не скомпилируется:

```
# ln -sf /usr/src/linux-<версия_ядра>/lib/modules/  
<версия_ядра>/build
```

Теперь загрузите архив со свежей версией исходных текстов ndiswrapper (<http://sourceforge.net/projects/ndiswrapper>) и распакуйте его куда-нибудь, например в каталог /usr/src:

```
# cd /usr/src/  
# tar zxvf ndiswrapper-1.5.tar.gz
```

Если вы используете самое свежее ядро, имеет смысл загрузить последнюю версию ndiswrapper с cvs:

```
# cd /usr/src  
# cvs -z3 -
```

Дополнительная информация

Графические удобства

Конечно, пользоваться консольными wireless-tools не очень удобно, особенно если вы привыкли работать в удобном графическом окружении и в консоль заходите редко. В этом случае можно порекомендовать отличную альтернативу — утилиту KwifiManager (www.kde.org), с недавних пор входящую в состав KDE (пакет kdenetwork). Эта программа умеет отображать уровень сигнала в системном трее KDE/GNOME, что очень удобно. Есть опция звукового оповещения (при нахождении сети программа запищит) — полезная функция, пригодится для любителей погулять с ноутбуком возле какого-нибудь посольства.

Если вас по каким-то причинам не устраивает KwifiManager, взгляните на WiFi Scanner — ее интерфейс выполнен средствами псевдографики. Однако самой функциональной программой для Wi-Fi является популярная Kismet (www.kismetwireless.net). Кроме поиска и сканирования сетей она умеет перехватывать незашифрованный трафик и имеет механизм обнаружения атак. Вся эта мощь скрыта в неприметном консольном интерфейсе, однако для Kismet написано множество графических фронтендов — как для GTK (<http://gkismet.sourceforge.net>), так и для Qt (<http://sourceforge.net/projects/kismet-qte>).

```
d:pserver:anonymous@cvs.sourceforge.net:/cvsroot/ndiswrapper  
co ndiswrapper
```

Теперь попробуйте скомпилировать его:

```
# cd ndiswrapper-1.5  
# make distclean  
# make && make install
```

Если компиляция не удастся и выскочит какая-нибудь ошибка, проверьте, включена ли в ядре поддержка PCI, USB и PCMCIA. Если нет, вам придется пересобрать ядро.

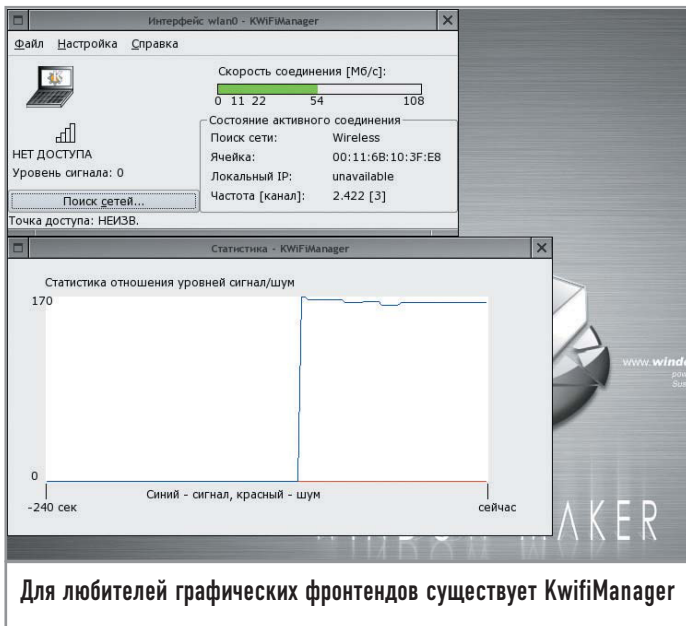
Установка Windows-драйвера

Во-первых, давайте удостоверимся, что система видит ваш Wi-Fi-адаптер. Для этого необходимо изучить вывод команды lspci (если у вас PCI- или PCMCIA-адаптер) или lsusb (при наличии USB-карты):

```
# lspci  
00:00.0 Host bridge: Intel Corporation 440BX/ZX/DX —  
82443BX/ZX/DX Host bridge (rev 03)  
00:01.0 PCI bridge: Intel Corporation 440BX/ZX/DX —  
82443BX/ZX/DX AGP bridge (rev 03)  
00:04.0 CardBus bridge: Texas Instruments PCI1450 (rev 03)  
00:04.1 CardBus bridge: Texas Instruments PCI1450 (rev 03)  
00:07.0 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ISA  
(rev 02)  
00:07.1 IDE interface: Intel Corporation 82371AB/EB/MB  
PIIX4 IDE (rev 01)  
00:07.2 USB Controller: Intel Corporation 82371AB/EB/MB  
PIIX4 USB (rev 01)  
00:07.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4  
ACPI (rev 03)  
00:08.0 Multimedia audio controller: ESS Technology ES1978  
Maestro 2E (rev 10)  
00:09.0 Ethernet controller: Intel Corporation 82557/8/9  
[Ethernet Pro 100] (rev 09)  
00:09.1 Serial controller: Agere Systems LT WinModem  
01:00.0 VGA compatible controller: ATI Technologies Inc Rage  
Mobility P/M AGP 2x (rev 64)  
02:00.0 Ethernet controller: Atheros Communications, Inc.:  
Unknown device 001a (rev 01)
```

Обратите внимание на последнюю строку — Linux догадалась, что это сетевая карта, а вот драйвера к ней подобрать не смогла, о чем говорит сообщение «Unknown device». Запомним номер устройства (02:00.0). Теперь давайте определим его PCI-идентификатор. Для этого выполните команду lspci -n:

```
# lspci -n  
00:00.0 Class 0600: 8086:7190 (rev 03)  
00:01.0 Class 0604: 8086:7191 (rev 03)  
00:04.0 Class 0607: 104c:ac1b (rev 03)  
00:04.1 Class 0607: 104c:ac1b (rev 03)  
00:07.0 Class 0680: 8086:7110 (rev 02)  
00:07.1 Class 0101: 8086:7111 (rev 01)  
00:07.2 Class 0c03: 8086:7112 (rev 01)  
00:07.3 Class 0680: 8086:7113 (rev 03)  
00:08.0 Class 0401: 125d:1978 (rev 10)  
00:09.0 Class 0200: 8086:1229 (rev 09)
```



00:09.1 Class 0700: 11c1:0445

01:00.0 Class 0300: 1002:4c4d (rev 64)

02:00.0 Class 0200: 168c:001a (rev 01)

Идентификатор находится в третьей колонке, для нашей карты (02:00.0) это 168c:001a. Зачем это нужно? Как я уже говорил, очень часто под одним и тем же именем производитель выпускает совершенно разные адаптеры на разных чипах (и, конечно же, они работают с различными драйверами). Безошибочно определить модель можно только с помощью PCI-идентификатора.

Теперь попробуйте поискать идентификатор вашего устройства в списке поддерживаемых адаптеров на сайте <http://ndiswrapper.sourceforge.net/mediawiki/index.php/List>. В нашем случае можно утверждать, что 168c:001a — это D-Link DWL-G630 rev. C на чипе Atheros. Там же можно получить ссылку на рабочий Windows-драйвер, который был протестирован в связке с ndiswrapper.

Перепишите все файлы Windows-драйвера (inf, sys, cat и dll) на жесткий диск в отдельную директорию. И попробуйте установить:

```
# cd win2kxp-driver
# ndiswrapper -i driver.inf
```

После этого ndiswrapper создаст конфигурационные файлы для вашей карты в каталоге /etc/ndiswrapper. Убедимся, что драйвер установлен успешно:

```
# ndiswrapper -l
Installed ndis drivers:
neta3ab driver present, hardware present
```

Как видите, все в порядке. Строка «neta3ab driver present» говорит о том, что драйвер net3ab установлен, а «hardware present» — что соответствующее устройство найдено. Если вместо этого вы увидите сообщение вроде «Cannot locate lspci. Unable to see if hardware is present» — значит у вас не установлен пакет pciutils.

Испытание

Ну что ж, теперь можно проверить работу драйвера, но перед этим обновите конфигурацию модулей ядра:

```
# depmod -a
# modprobe ndiswrapper
```

На всякий случай проверьте сообщения ядра:

```
# dmesg
wlan0: ndiswrapper ethernet device 00:11:95:bf:21:cb using driver
neta3ab, 168C:001A.5.conf
wlan0: encryption modes supported: WEP; TKIP with WPA,
WPA2, WPA2PSK; AES/CCMP with WPA, WPA2, WPA2PSK
```

Все в порядке, появился новый сетевой интерфейс (в это время на самой карте должен замигать индикатор «Link»). Для работы с Wi-Fi-адаптерами написан пакет wireless-tools, который на данный момент есть в каждом дистрибутиве. В него входят утилиты iwconfig, iwlist, iwconfig, iwspy. Для просмотра подробной информации о сетевом интерфейсе выполните следующую команду:

```
# iwconfig wlan0
wlan0 IEEE 802.11b ESSID:"" Nickname:"random"
Mode:Managed Frequency:2.412 GHz Access Point:
00:00:00:00:00:00
Bit Rate=54 Mb/s
Encryption key:off
Power Management:off
Link Quality:0 Signal level:0 Noise level:0
Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
Tx excessive retries:0 Invalid misc:0 Missed beacon:0
```

Для поиска точек доступа воспользуйтесь утилитой iwlist:

```
# iwlist wlan0 scan
wlan0 Scan Completed:
Cell 01 —Address: 00:02:71:50:EF:DD
ESSID: «Yandex.wifi»
Mode: Master
Encryption key: Off
Channel: 1
Quality: 81/92 Signal Level:
150/154
```

Видно, что iwlist нашла одну сеть со свободным доступом (Encryption key: Off). Давайте попробуем подключиться к ней:

```
# iwconfig wlan0 essid 'Yandex.wifi'
# dhclient wlan0
```

Все, теперь можно спокойно сидеть в кафе и не спеша, потягивая кофе, запускать любимый Firefox (а заодно и Wget). |

Дополнительная информация

Полезные ссылки

- ▶ www.linux-wlan.com — открытые драйверы для карт на чипе Prism/2.
- ▶ <http://sourceforge.net/projects/madwifi> — открытые драйверы для карт на чипе Atheros.
- ▶ <http://prism54.org> — открытые драйверы для карт на чипе Prism54.
- ▶ <http://acx100.sourceforge.net> — открытые драйверы для карт на основе чипов acx100.
- ▶ <http://ipw2100.sourceforge.net> — открытые драйверы для карт, встроенных в ноутбуки на платформе Intel Centrino.
- ▶ www.linux-wlan.org/docs/wlan_adapters.html.gz — постоянно обновляемый список моделей Wi-Fi-адаптеров и чипов.

Free Software и Open Source: разные слова — разные идеи

Свободное программное обеспечение, как его ни назови, дает нам одинаковую свободу. Но нельзя забывать о том, что разные слова выражают

разные идеи [1]. Задачей данной статьи является анализ различий между движениями Free Software и Open Source.

Официальные определения терминов Free Software и Open Source достаточно схожи. Я подчеркиваю слово «официальные», поскольку люди часто склонны выводить свои собственные определения этих понятий, из-за чего возникает путаница. Рекомендую читателям ознакомиться с официальными определениями [2][3][4].

Термин Free Software может быть воспринят двояко: как «программное обеспечение, которое распространяется бесплатно» или «которое дает пользователю определенные свободы». Именно такие значения вкладывает в этот термин Ричард Столлмен [5], начинатель Free Software movement, движения за свободу программного обеспечения. Несмотря на то что определение термина Open Source, опубликованное Open Source Initiative (OSI) [6], схоже с определением Free Software от Free Software Foundation [7], они несут в себе разные идеи. Open Source — это методология разработки, Free Software — общественное движение. Для Open Source movement проприетарные программы — неудачное техническое решение, для Free Software movement — социальная проблема, решением которой является использование свободного программного обеспечения.

Разговоры о свободе, этических принципах, ответственности наряду с удобством использования заставляют людей задумываться о тех вещах, которые им проще игнорировать. Термин Open Source (в нем нет ни слова о свободе и этике) гораздо больше нравится бизнесменам. Но это не значит, что от этого «молчания» выигрывает общество. Многие люди сегодня переходят на использование свободного программного обеспечения из чисто прагматических соображений. Но привлечение пользователей — это лишь часть задачи, которая должна стоять перед разработчиками свободных программ. Рано или поздно человеку предложат снова начать использовать проприетарные программы, которые будут более совершенными с технической точки зрения. И почему он должен отвергать это предложение? Только в том случае, если человек ценит свободу, которую дает ему свободное программное обеспечение, он сможет сказать: «Я отказываюсь использовать проприетарные программы!» Именно по этой причине мы должны говорить об этических аспектах использования Free Software.

Термин Open Source не только не может быть полноценной заменой термина Free Software [8], он еще и создает дополнительную путаницу вокруг своего использования (его гораздо сложнее объяснить): пример про разницу между свободой слова и бесплатным пивом уже не подходит. В статье «Мода на Open Source» [9] Юрий Лушня утверждает, что Ричард Столлмен является «пионером движения Open Source», а также приписывает хакерам из MIT авторство идеи Open Source. Оба эти утверждения ошибочны. Ричард Столлмен не раз высказывался о том, что именно этический аспект должен быть решающим при выборе той или иной программы для использования [1][2][10][11], а Стивен Леви посвятил целую книгу вопросам хакерской этики [12]. Представители Open Source Initiative [6], в свою очередь, открыто говорят о том, что ими движет исключительно прагматизм [13]. Следует отметить и тот факт, что термин Open Source появился 3 февраля 1998 года [13], и по этой причине его недопустимо использовать при описании событий, предшествующих указанной дате.

Движения Free Software и Open Source не являются врагами друг друга, они вполне могут сотрудничать. Благодаря действиям Open Source Initiative количество людей, использующих свободное программное обеспечение, стремительно растет. Такие люди получают свободу, которую они должны ценить, чтобы не потерять. Именно поэтому нужно больше и громче говорить об этических аспектах использования свободных программ [1][10][11].

- [1] www.gnu.org/philosophy/free-software-for-freedom.html
- [2] www.gnu.org/philosophy/free-sw.html
- [3] www.opensource.org/docs/osd.pdf
- [4] www.opensource.org/docs/def_print.php
- [5] www.stallman.org
- [6] www.opensource.org
- [7] www.fsf.org
- [8] www.gnu.org/philosophy/words-to-avoid.html
- [9] www.linuxrsp.ru/artic/opensource.html
- [10] www.audio-video.gnu.org/audio/rms-speech-curitiba.ogg
- [11] www.gnu.org/philosophy/pragmatic.html
- [12] Steven Levy, *Hackers: Heroes of the Computer Revolution*, Doubleday: New York, 1984
- [13] www.opensource.org/docs/history.php

Секреты системы

Очередная десятка советов от Дениса Колисниченко охватывает широкий диапазон тем — от настройки оборудования (ОЗУ и модемов) до тонкостей тюнинга системы.

Горячие клавиши

«SysRq», или Что делать во время аварии?

Linux считается одной из самых надежных операционных систем, но иногда и она дает сбой. Все мы знаем о двух волшебных комбинациях клавиш — «Ctrl+Alt+Del» и «Ctrl+Alt+Backspace». Первая используется для перезагрузки системы, а вторая — для перезапуска X Window.

Что же делать, если система все-таки зависла? Сразу нужно оговориться, что тут все зависит от «степени зависания»: может случиться так, что она не будет реагировать на внешние сигналы (в том числе и нажатия клавиш) — в этом случае уже ничем кроме «Reset» не поможешь. Но шанс сохранить данные в случае сбоя все же существует. Об этом сейчас мы и поговорим.

Обратите внимание на клавишу «Print-Screen». Сверху написано «PrtSc», а внизу — «SysRq». Нажав «Alt+PrtSc» и одну из буквенных клавиш, вы произведете некоторые действия, которые могут оказаться спасительными в случае аварийных ситуаций. Возможные комбинации с клавишей «SysRq» вы найдете в текстовом файле sysrq.txt (обычно он находится в каталоге /usr/src/linux/Documentation/systq.txt). Мы рассмотрим лишь самые полезные из них.

Alt+SysRq+K

Нажав эту комбинацию клавиш, вы сможете «убить» все зависшие процессы (точнее, процессы, запущенные в теку-

щей виртуальной консоли), которые не отвечают на «Ctrl+C», и их нельзя завершить обычным образом. Та же клавиатурная комбинация помогает и в тех случаях, когда система X Window зависла и не реагирует даже на нажатие «Ctrl+Alt+Backspace». Конечно, можно воспользоваться сочетанием «Ctrl+Alt+Del», но далеко не всегда хочется полностью перезагружать систему. Данная комбинация клавиш полезна не только для снятия зависших процессов. Ее можно использовать, если какой-то злоумышленник установил в вашей системе программу, эмулирующую работу процесса login. Она выводит обычное приглашение (вы его не отличите от приглашения стандартного login), получает от вас пароль, записывает его в специальный файл, а потом выдает сообщения о том, что вы ввели неверный пароль, и передает управление оригинальной программе login. Вы даже ничего можете и не заподозрить — подумаешь, наверное, на самом деле неправильный пароль. После нажатия этой комбинации клавиш происходит завершение всех процессов, кроме оригинального login. Все программы, которые маскируются под login, будут также завершены. Данную комбинацию клавиш еще называют SAK (Secure Access Key). Подробнее о ней вы сможете прочитать в файле /usr/src/linux/documentation/SAK.txt

Alt+SysRq+E (tErm)

Данная комбинация посылает всем процессам (кроме Init) сигнал SIGTERM.

После этого в системе остаются только ядро, Init и текущая консоль. После этого можно запустить заново все сервисы (Init 3 или Init 5).

Alt+SysRq+I (kIll)

Эта комбинация аналогична «Alt+SysRq+E», но посылает всем процессам (кроме Init) сигнал SIGKILL. Сигналы SIGTERM и SIGKILL различаются следующим образом: получив SIGTERM, программа должна сохранить данные (если, конечно, программист предусмотрел реакцию на этот сигнал) и завершить работу; сигнал SIGKILL моментально «убивает» программу — сохранить данные она уже не сможет.

Alt+SysRq+S (Sync)

Это сочетание клавиш заставляет ядро выполнить синхронизацию буферов ввода/вывода, то есть сбросить содержимое дисковых буферов на диск. Очень полезная комбинация клавиш, помогающая сохранить ваши данные. Ведь, если вы сохранили данные в своей программе, это еще не означает, что они были физически записаны на диск. Синхронизация буферов — это не мгновенная процедура. После нажатия этих клавиш вам нужно немного подождать, пока на консоли не появится сообщение «Emergency Sync.... ОК».

Если же вывод на консоль невозможен, просто подождите около пяти секунд. Будем надеяться, что система выполнила синхронизацию буферов (хотя, повторюсь, все зависит от «степени зависания»).

Alt+SysRq+U (Umount)

Данная комбинация используется для размонтирования всех смонтированных файловых систем, а это занимает больше времени, чем синхронизация, поэтому ждать нужно минимум пять секунд (а то и больше) — до появления сообщения «Emergency Umounting... OK».

Но если оно так и не появилось, возможны два варианта:

- ▶ файловые системы все-таки размонтированы, вывод на консоль невозможен;
- ▶ файловые системы не размонтированы, поскольку система вообще ни на что не реагирует.

Перед тем как воспользоваться этой комбинацией, нужно нажать предыдущую («Alt+SysRq+S») с целью синхронизации буферов ввода/вывода и только после этого размонтировать файловые системы.

В общем, при зависании системы наиболее верной будет следующая последовательность действий.

- ▶ Нажать «Alt+SysRq+E (или K)». Если ничего не помогло и система по-прежнему висит, тогда переходим к следующему пункту.
- ▶ Нажать «Alt+SysRq+S» и подождать пять секунд.
- ▶ Нажать «Alt+SysRq+U». Подождать 5–10 секунд (в зависимости от количества смонтированных файловых систем).
- ▶ Нажать «Reset».

Alt+SysRq+B (reBoot)

Используется для мгновенной перезагрузки. До применения этой комбинации желательно нажать «Alt+SysRq+S» и «Alt+SysRq+U» (соответственно, подождать 2-5 и 5-10 секунд). Стоит отметить, что эффект от комбинации «Alt+SysRq+B» практически эквивалентен «Reset».

Alt+SysRq+O (power off)

Мгновенно выключает питание, не размонтируя файловые системы. Ясно, что до этого нужно воспользоваться комбинациями «Alt+SysRq+S» и «Alt+SysRq+U». |

Кодировки

Win-1251 и KOI8-R

Скорее всего, Linux на вашей домашней машине — не единственная операционная система. Как минимум у вас установ-

лена еще и Windows. Какая именно — не важно. Ведь все версии этой системы используют кодировку Win-1251.

А что делать, если нужно отредактировать текстовый файл, созданный в Windows? С документами MS Office все ясно: их полностью (ну или почти полностью) поддерживает пакет OpenOffice. Сейчас мы поговорим о просмотре и редактировании «инопланетных» файлов.

Начнем с просмотра. Предположим, что у нас есть файл в кодировке Win-1251. Хотя не так уж и важно, cp1251 это или cp866. Главное, что мы не можем посмотреть его обычным способом, требуется перекодировка. Для этого используется команда iconv:

```
iconv -f <исходная кодировка> -t <результатирующая кодировка> <исходный файл>
```

Вот примеры использования этой команды (наш файл называется report.txt):

```
iconv -f cp1251 -t koi8-r report.txt | less
iconv -f cp1251 -t koi8-r report.txt > report-koi8.txt
```

В первом случае мы перекодируем файл и передаем программе less, которая выводит его на консоль. Можно обойтись и без нее, но если файл большой, то вряд ли мы успеем его прочитать.

Во втором случае мы перекодируем файл и результат перекодирования записываем в report-koi8.txt, который можно будет использовать обычным образом.

Кроме кодировки cp1251 вы также можете использовать KOI8-R, cp866 и UTF-8. Теперь поговорим о редактировании. Вообще-то, лучше всего перекодировать файл в KOI8-R и редактировать его как обычно, но можно сразу делать это с помощью редакторов, поддерживающих несколько кодировок, например kwrite, kate, mc и vim. С первыми двумя все очень просто — из меню выбираем нужную кодировку, и все. В mc нужно зайти в меню «Настройки → Биты символов» и в появившемся окне выбрать системную кодировку. В режиме просмотра/редактирования файла для выбора кодировки нажмите «Ctrl+T». Заданная кодировка сохранится и будет использоваться при последующем редактировании — до тех пор, пока вы не выберете другую.

В vim для перекодирования файла введите команду:

```
:e ++enc=cp1251
```

Вместо cp1251 можно указать другую кодировку, например cp866 или UTF-8. Для сохранения файла в нужной кодировке введите команду:

```
:w ++enc=<кодировка>
```

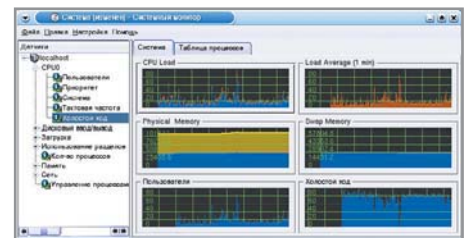
Тюнинг системы

Мониторинг

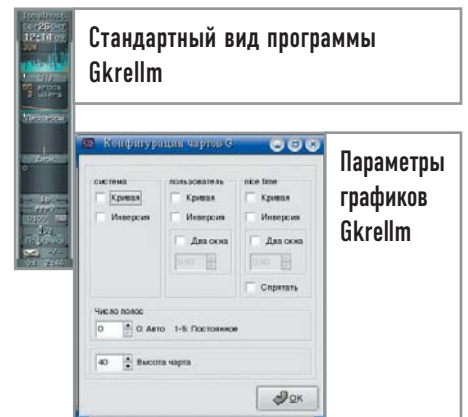
В составе KDE есть свой собственный системный монитор, но использовать его не хочется по двум причинам:

- ▶ он довольно громоздкий и занимает много места на экране;
- ▶ не хочется, чтобы было «как у всех».

Если вам важно, чтобы системный монитор был компактным и гибким в настройке и к тому же не похожим на тот, что есть у соседа, установите пакет gkrellm. Он отвечает всем вышеперечисленным требованиям, к тому же поддерживает различные графические схемы, что немаловажно для домашнего пользователя. В стандартной редакции Gkrellm умеет показывать часы, загрузку процессора, сети, обращения к диску, сенсоры (lm_sensors) и многое другое. Кроме того, в Интернете можно найти огромное количество плагинов, например аналоговые часы, индикатор погоды и т. д. Согласитесь, интересно из всего этого множества собрать свой собственный

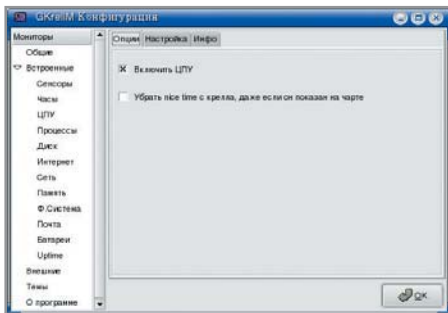


Системный монитор KDE

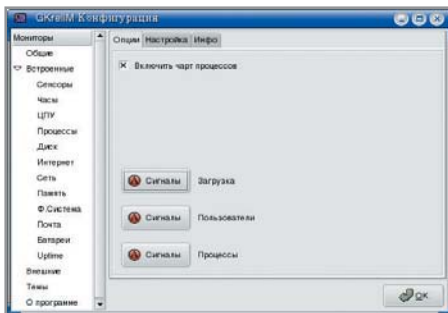


Стандартный вид программы Gkrellm

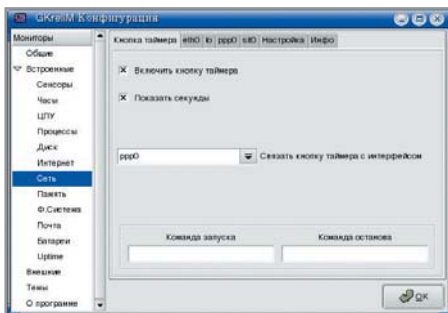
Параметры графиков Gkrellm



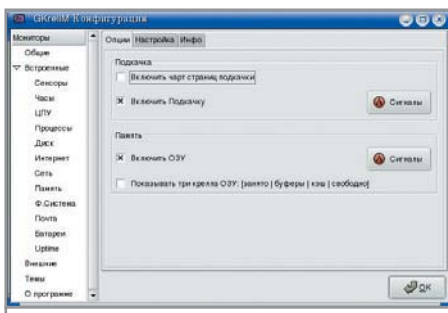
Настройка параметров CPU



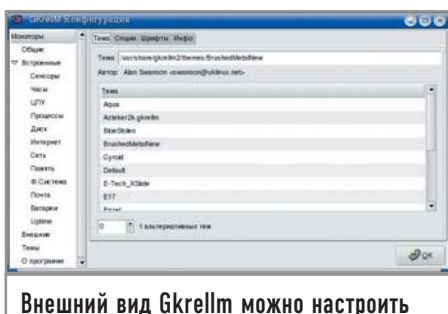
Настройка отображения процессов



Для какого сетевого интерфейса построить график загрузки?



Состояние ОЗУ и файла подкачки



Внешний вид Gkrellm можно настроить

системный монитор, подобного которому не будет ни у кого.

Скорее всего, пакет gkrellm будет присутствовать в вашем дистрибутиве. Обычно этот системный монитор не значится по умолчанию, поэтому его нужно установить самостоятельно. Рекомендую также пакет gkrellm-devel — он понадобится при сборке дополнительных плагинов.

Вы еще не установили Gkrellm? Значит, сейчас самое время это сделать.

В данном варианте программа информирует о следующих параметрах системы:

- ▶ загрузка процессора (ЦПУ);
- ▶ количество запущенных процессов и зарегистрированных пользователей;
- ▶ нагрузка на дисковую систему ввода/вывода;
- ▶ состояние сетевых интерфейсов lo и ppp0;
- ▶ состояние физической памяти и свопинга;
- ▶ количество писем в почтовом ящике.

Нажав клавишу «F1», вы сможете настроить программу и ее датчики на свой вкус. Утилита использует темы интерфейса, благодаря которым вы можете изменить ее внешний вид. Если в вашей системе не установлено ни одной темы кроме стандартной, загляните на сайт www.muhi.net. Загрузив тему, распакуйте ее в каталог `~/gkrellm2/themes` (~ — ваш домашний каталог).

Также в Интернете можно найти огромное количество плагинов.

- ▶ aLLtraXClock и GkrellAClock — отображают аналоговые часы.
- ▶ GkrellWeather — показывает погоду в вашем регионе.
- ▶ GkrellShoot — позволяет сделать снимок экрана одним нажатием кнопки.
- ▶ GkrellMLaunch — добавляет в окно монитора кнопки для вызова пользовательских программ.
- ▶ gkrellm xkb — отображает раскладку клавиатуры.
- ▶ GkrellMMS — плагин управления плеером XMMS.
- ▶ Volume — регулятор громкости.

После сборки плагина поместите его (файл .so) в каталог `~/gkrellm2/plugins`, а затем в настройках Gkrellm в разделе «Внешние», включите плагин. После этого там должен появиться отдельный пункт для настроек плагина.

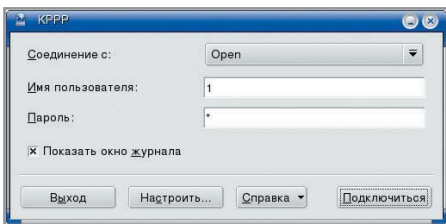
Если Gkrellm вам не понравился, можете использовать программу KSim (команда и пакет ksim), которая обычно входит в состав KDE. |

Оборудование

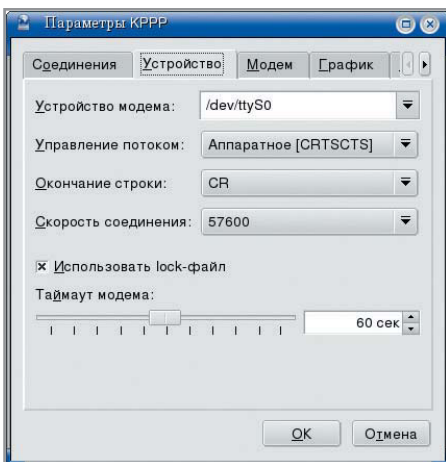
Модемы и Linux

Теперь мы поговорим о том, как выбрать модем для своего компьютера и как заставить работать win-модем под Linux. Модемы бывают внутренними и внешними. Чем они отличаются, думаю, говорить не нужно. Большинство внутренних устройств — это так называемые win-модемы. Для тех, кто не в курсе, отмечу, что win-модемы (их еще называют программными, или «софтовыми»), в отличие аппаратных, не могут работать без соответствующего программного обеспечения. Так уж сложилось, что поставщики этих устройств выпускают драйверы только для Windows. Аппаратный модем может работать в любой операционной системе без драйвера в качестве AT-совместимого устройства (AT — это набор стандартных команд модема, а не компьютер от IBM). Никогда не покупайте win-модем. Да, вы сэкономите немного денег, но в итоге намучаетесь и все-таки приобретете обычный внешний модем, к тому же стоит он сейчас совсем недорого — около \$30–40, а меньше чем за \$80 можно купить ZyXEL OMNI. Модемы ZyXEL необходимы в том случае, если качество вашей линии связи оставляет желать лучшего. Если же линия более или менее нормальная, то и 30-долларовый GVC будет работать отлично. Не смейтесь, я использую дома GVC F2000 и не жалею. При покупке внешнего модема старайтесь выбирать классику — обычный COM- (RS-232C), а не USB-модем. Дело в том, что первый подключается к компьютеру через последовательный порт, а второй — через шину USB. Если вы планируете использовать модем в Linux, то первый вариант предпочтительнее. Я не говорю, что у Linux особые проблемы с USB-модемами, просто COM-модем настроить легче, а работать он будет одинаково в любом дистрибутиве — что в старом Red Hat 5, что в новом Mandriva 2005. В общем, с модемом определились. Теперь подключаем его к компьютеру (если

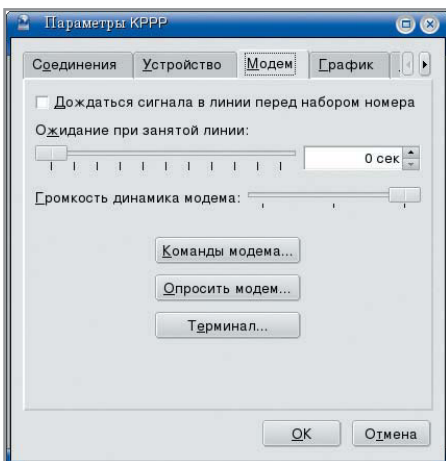
у вас COM-модем, то сначала желательно выключить компьютер) и включаем компьютер. Загружаем Linux. Система сама определит устройство, хотя для нас это не столь важно, если мы используем аппаратный внешний модем. Для его нормальной работы нам нужно только знать, к какому порту он подключен — COM1 (/dev/ttyS0) или COM2 (/dev/ttyS1). Если у вас только один последовательный порт, то это будет COM1 (/dev/ttyS0), другого не дано.



Вход в программу kppp



kppp поможет настроить все параметры модема в удобном графическом виде



Некоторые AT-команды можно задать в графическом режиме

Скорее всего, в вашем дистрибутиве есть какая-либо программа — конфигураатор модема. Если вы обладатель аппаратного COM-модема, можете ее не использовать: все, что она сделает, это создаст ссылку /dev/modem на один из файлов устройств — /dev/ttyS0 или /dev/ttyS1.

Для настройки соединения с Интернетом проще всего сразу использовать программу kppp. Описывать ее полностью я не стану — работать с ней не сложнее, чем со стандартным дозвонщиком Windows. Остановимся только на некоторых интересных моментах. Запустите kppp и нажмите кнопку «Настроить».

На закладке «Устройство» необходимо ввести имя устройства /dev/ttyS0 (или /dev/ttyS1) и его скорость. Помните, если вы установите значение, превышающее максимальную скорость вашего модема, то быстрее работать от этого он не будет! Теперь перейдите на страничку «Модем» в окне «Параметры KPPP» и установите следующее:

- ▶ «Дождаться сигнала в линии перед набором номера» — обычно данный параметр включен по умолчанию, но если у вас старенькая АТС и сигнал очень слаб, модем даже не будет пытаться набирать номер. Поэтому в случае слабого сигнала можно отключить этот параметр — модем наберет хотя бы номер.
 - ▶ «Ожидание при занятой линии» — 0 секунд, нет смысла ждать.
 - ▶ «Громкость динамика» — это, как уже говорилось, на любителя.
- Теперь нажимаем кнопку «Команды модема» и в появившемся окне ищем пункт «Строка набора номера». Его нужно изменить следующим образом:

- ▶ ATDP — если у вас импульсная АТС;
- ▶ ATDT — если у вас тоновая АТС.

Все остальные параметры можно оставить без изменения (во всяком случае пока). Теперь самое время создать соединение и опробовать новый модем.

Оборудование

Win-модемы

Если вы все-таки не вняли предыдущему совету и приобрели win-модем (или он уже давно был установлен в вашем компьютере), сейчас мы попытаемся его настроить. Но только в том случае, если он

с чипом Lucent. Во всех остальных ситуациях этот совет не поможет.

Итак, заходим на сайт <http://heby.de/ltmodem>. Там есть уже откомпилированные модули для Lucent Apollo и Mars. ARM-модемы этим драйвером не поддерживаются. Скачиваем откомпилированный модуль. Кстати, модули доступны для следующих дистрибутивов:

- ▶ Debian GNU/Linux;
- ▶ Gentoo;
- ▶ Red Hat Linux;
- ▶ Fedora Core;
- ▶ Mandrake Linux;
- ▶ SUSE Linux;
- ▶ Conectiva Linux.

Если вашего дистрибутива нет в списке (или же откомпилированный модуль почему-то не устанавливается в системе), нужно скачать пакет с исходным кодом. Последняя версия модуля (драйвера для модема) на момент написания этих строк — 8.30a3. Пока она находится на стадии тестирования, но у нас нет выбора: стабильная версия 8.2x не поддерживает ядро 2.6.

Если вы скачали архив с исходным кодом, установить модуль очень просто, но перед этим убедитесь, что у вас установлены заголовочные файлы ядра и компилятор GCC. Для установки модуля распакуйте архив с исходным кодом в /usr/src/ltmod и запустите сценарий build_module. Затем, если модуль, конечно, успешно откомпилировался, запустите сценарий ltinst2 для установки модуля (копирования его в каталог с модулями), а после этого — сценарий autoload, который «пропишет» только что откомпилированный модуль в файл /etc/modules.conf. В результате выполнения этих команд модуль скопируется в каталог /lib/modules/`uname -r`/ltmodem/. В документ etc/modules.conf нужно добавить следующие строки:

```
alias char-major-62 lt_serial
alias /dev/ttyLT0 lt_serial
alias /dev/modem lt_serial
```

Кроме того, необходимо обновить список зависимостей (команда depmod -a) и создать устройство /dev/ttyLT0:

```
# mknod /dev/ttyLT0 c 62 64
# ln -s /dev/ttyLT0 /dev/modem
```

В kppp в качестве устройства нужно указать /dev/modem (можно напрямую —

/dev/ttyLT0), а дальше все как обычно. Если ваш модем будет плохо распознавать сигнал «занято», то установите следующую строку инициализации модема — «AT+GCI=B8».

Настройка системы

Сверим часы

Как вы уже догадались, ниже речь пойдет о синхронизации часов компьютера с сервером времени. Для этого мы будем использовать протокол NTP (Network Time Protocol). Принцип синхронизации заключается в следующем: NTP-клиент (программа ntpdate) подключается к серверу времени, получает от него точное значение часов, минут и секунд и устанавливает системное время в соответствии с полученными данными. Программу ntpdate можно скачать с сайта www.ntp.org.

Для синхронизации времени выполните команду (от имени пользователя root):

```
# ntpdate <сервер_имен>
```

Список общедоступных серверов имен находится на сайте www.eecis.udel.edu/~mills/ntp/servers.html.

Ясно, что вам не хочется каждый день вводить эту команду вручную. Поэтому для автоматической синхронизации времени поместите команду синхронизации в расписания. Например, для синхронизации времени в 7 часов утра каждого дня кроме воскресенья добавьте в /etc/crontab следующую запись:

```
0 7 * * 1-6 ntpdate <сервер_имен>
```

Думаю, не нужно напоминать о том, что еще до выполнения этой команды у вас должно быть установлено соединение с Интернетом. Если это коммутируемое соединение (обычное модемное), то в дозвонщике (программе, которую вы обычно используете для установки соединения с Интернетом) укажите, что команду ntpdate нужно запускать сразу после установки соединения.

Можно использовать более универсальный способ — создать сценарий, вызывающий команду ntpdate, и поместить его в каталог /etc/ppp/ipup.d, все файлы которого будут выполняться непосредственно при установке соединения с Интернетом. Код сценария следующий:

```
#!/bin/sh
```

```
ntpdate сервер_времени > /dev/null
2>&1
```

Не забудьте сразу сделать сценарий исполнимым:

```
# chmod +x time-sync (мы предполагаем, что этот сценарий называется именно так)
```

Настройка системы

Нестандартные видеорежимы

Преимущество подобных видеорежимов в том, что они позволяют задать нестандартное разрешение, увеличить частоту обновления монитора и провести прочие эксперименты с видеоадаптером и монитором. Сразу хочу отметить, что для некоторых моделей мониторов и видеоадаптеров делать это весьма нежелательно, поэтому все операции вы будете проводить на свой страх и риск.

Итак, начнем. В файле XF86Config (или XF86Config-4) прописывается вся информация о видеоподсистеме, в том числе и режим монитора. Он задается специальной строкой (в англоязычной литературе строка, описывающая режим монитора в файле XF86Config, называется Modeline) следующего формата:

```
Modeline <name> <clock> <horiz-values> <vert-values>
```

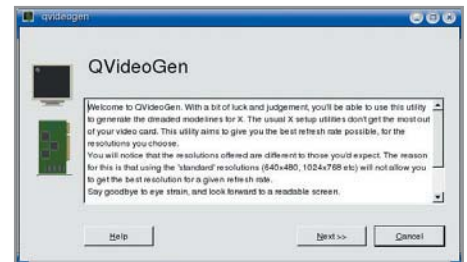
Здесь:

- ▶ name — имя режима, обычно используется для указания разрешения, например 1024x768;

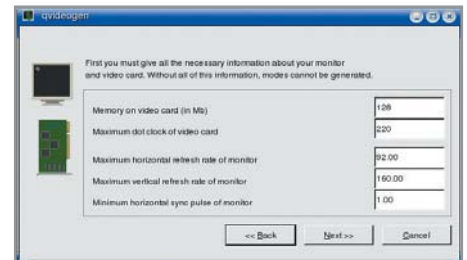
- ▶ <clock> — частота генератора для указанного в <name> разрешения; прописывается в мегагерцах и определяет скорость, с которой видеоплата должна посылать значения точек на монитор;

- ▶ <horiz-values> и <vert-values> — задают время, когда ЭЛТ монитора должна включиться и когда должны прийти импульсы горизонтальной и вертикальной синхронизации при развертке луча. Значения <horiz-values> и <vert-values> состоят из четырех дополнительных значений: причем четыре значения относятся к горизонтальной развертке и еще четыре — к вертикальной.

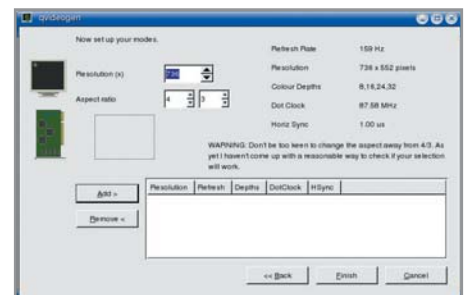
Задавать режимы вручную не очень хочется, особенно если не совсем пони-



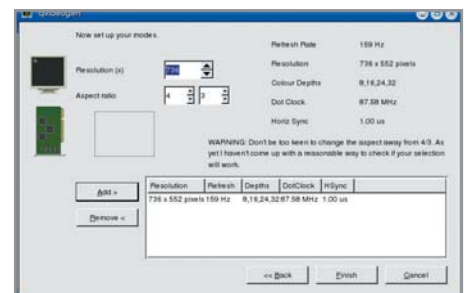
Окно приветствия QVideoGen



Задаем частоту развертки монитора и объем ОЗУ видеоадаптера



Введите разрешение и соотношение сторон для вашего монитора



Для ноутбуков будет полезно создать несколько профилей видеорежимов

маешь, что здесь и к чему. Облегчить задачу сможет программа QVideoGen — генератор видеорежимов. Скачать ее можно с сайта, расположенного по адресу <ftp://ftp.altlinux.ru/pub/distributions/ALTLinux/Compact/2.3/ALTLinux/RPMS.contrib/qvideogen-1.3-alt1.i586.rpm>.

Это далеко не единственное зеркало, просто именно оттуда ее скачал я. После запуска программа предлагает установить параметры видеоадаптера и мо-

нитора, а именно максимальные размер памяти и частоту для видеоадаптера и максимальные частоты развертки для монитора. После этого введите нужное разрешение и нажмите «Add» для добавления режима в список. При нажатии кнопки «Finish» будет создан текстовый файл, содержащий «Modelines» для всех видеорежимов, которые вы добавили в список. Вам осталось только скопировать эти строки в ваш файл XF86Config (или XF86Config-4).

После редактирования XF86Config нужно перезагрузить графическую систему X Window. Для этого нажмите комбинацию клавиш «Ctrl+Alt+Backspace». Не исключено, что после перезагрузки результат будет несколько другим, чем вы ожидали. Экспериментируйте! Кстати, предварительно, пока вы еще не начали это делать, обязательно сохраните копию XF86Config (XF86Config-4) — на всякий случай. |

Настройка системы

Не хватает оперативной памяти

Лучший совет — это пойти в магазин и купить дополнительный модуль памяти. А пока вы выискиваете на это деньги в своем бюджете, можно создать дополнительный файл подкачки — хоть чем-то он сможет помочь. Да, мы создадим именно файл подкачки, а не раздел. Для создания раздела вам нужно будет переразбить заново жесткий диск, а это займет много времени, учитывая, что придется создавать резервные копии всех данных, а до этого еще и произвести дефрагментацию всего жесткого диска. Поэтому мы создадим файл подкачки размером 128 Мбайт (потом его можно будет деактивировать и безболезненно удалить):

```
# dd if=/dev/zero of=/swap/sw-file
bs=1k count=131072
```

Данная команда создала файл, состоящий из ASCII-нулей, размером 128 Мбайт (131 072 байта). Теперь форматируем этот файл под swap:

```
# mkswap /swap/sw-file 131072
```

Все, что нам осталось сделать, — это активизировать файл подкачки:

```
# swapon /swap/sw-file
```

Данную команду желательно добавить в сценарий загрузки, иначе вам придется вводить ее каждый раз при загрузке и перезагрузке системы. |

Оборудование

Установка еще одного модуля памяти

Вы установили дополнительный модуль памяти, но система не видит его, например, вместо 512 Мбайт по-прежнему используются только 256. Причина в том, что некоторые дистрибутивы при установке в конфигурацию загрузчика добавляют параметр ядра MEM=<NNNM>, где NNN — объем оперативной памяти в мегабайтах на момент установки.

```
# append="mem=256M"
```

```
# объявление параметров ядра
```

Все, что теперь нужно сделать, — это отредактировать параметр mem, например mem=512M, или вообще его удалить. Только не перестарайтесь и не удаляйте всю строку параметров (append) — возможно, у вас установлены какие-то специфические параметры ядра. Если же параметр mem у вас единственный, можно удалить всю строку. |

Администрирование

Альтернативный кеширующий DNS-сервер pdnsd

Для начала давайте разберемся, что такое кеширующий DNS-сервер и зачем он нужен. Когда вы обращаетесь к какому-нибудь веб-узлу по его имени, а не IP-адресу (в большинстве случаев так оно и есть), для установки соединения с этим узлом нужно разрешить его символьное имя в IP-адрес. Программа-резолвер ищет его в своем кеше. Если соответствующего имени там нет, она обращается к DNS-серверу провайдера. Ясно, что на это уходит определенное время. Установив кеширующий сервер, мы можем существенно сократить число запросов к DNS-серверу провайдера, поскольку все необходимые записи узлов, в которых вы часто обращаетесь, хранятся в кеше вашего собственного DNS-сервера. Особо это позволяет ускорить запросы на медленных модемных соединениях.

Если вы следили за мыслью, то у вас возник запрос: «А зачем настраивать кеширующий DNS-сервер, если резолвер также кеширует DNS-ответы?» А затем, что кеширующий DNS-сервер сохраняет DNS-ответы при перезагрузке системы, а резолвер — нет.

Традиционно в качестве кеширующего DNS-сервера даже на домашней машине использовался BIND (named). Но сейчас мы поговорим о более легком в настройке кеширующем сервере pdnsd. Да, у него нет и половины функций BIND, но ведь они вам и не нужны. Ваша задача — настроить кеширующий сервер для своего локального компьютера. А использовать BIND для этих целей, мягко говоря, совсем не рационально.

Загрузить сервер pdnsd можно по адресу www.phys.uu.nl/~rombouts/pdnsd/. После установки отредактируйте файл конфигурации /etc/pdnsd.conf:

```
# local usage only
server_ip="127.0.0.1";
server_port=53;
```

В этом случае сервер будет доступен локальным приложениям (он будет использовать традиционный 53-й порт). После этого нужно определить два любых корневых сервера. Лучше всего использовать DNS-серверы провайдера, поскольку они ближе всего к вам:

```
server {
ip="82.207.9.115";
changeable_ip=on;
label="pppdns1";
timeout=30;
purge_cache=off;
}
server {
ip="82.207.9.116";
changeable_ip=on;
label="pppdns2";
timeout=30;
purge_cache=off;
}
```

В файл конфигурации резолвера добавьте всего одну строку — она должна быть первой:

```
nameserver 127.0.0.1
```

Все, запускаете сервер и наслаждаетесь его работой (не забудьте только подключиться к Интернету!):

```
service pdnsd start
```

CinePaint

Есть такие приложения, которые еще до выхода версии 1.0 снискали и уважительное отношение, и повышенное внимание, демонстрирующие своим существованием преимущества открытой модели разработки ПО. В их числе, без сомнения, и CinePaint — проект, объединяющий открытые ресурсы киноиндустрии, университетских кругов и сообщества Open Source. CinePaint находит себя главным образом в тщательной обработке изображений, в особенностях цвета, поддерживая работу с

глубиной 32 бита на канал. Успеху CinePaint можно позавидовать: такие студии как Rhythm & Hues, Computer Cafe, Amalgamated Pixels, Sony Pictures Imageworks, DreamWorks и многие другие не обошлись без него при создании «Гарри Поттера», обеих частей «Маленького Стюарта», «Планеты обезьян», «Лиги выдающихся джентльменов», «Последнего самурая» и других фильмов. Кого после этого остановит скромный номер 0.20 текущей версии? Проект CinePaint не ограничивается созданием одного редактора — объявлено о готовящемся к 31 декабря выходе архитектуры Glasgow, во многом

схожей с Film Gimp, но отличающейся по исполнению и масштабам.

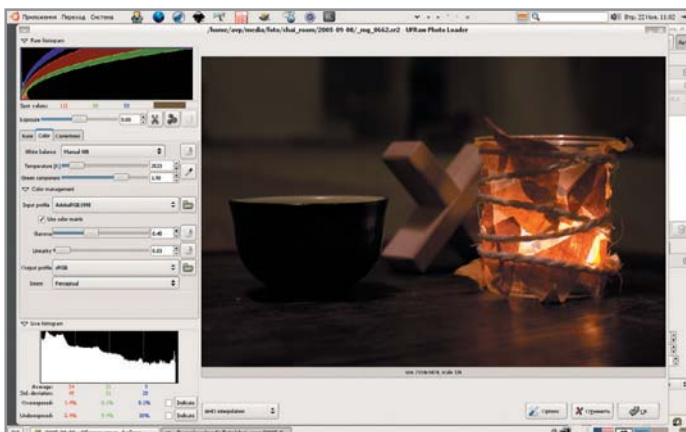
Язык интерфейса: русский

Домашняя страница: cinpaint.sourceforge.net



Вышла новая версия UFRaw — программы для обработки RAW-снимков из цифровых камер. Релиз основан на DCRaw v7.84. При помощи

программы снимки, сделанные в формате RAW, можно подвергнуть предварительной обработке — скорректировать баланс белого, экспозицию, применить готовые



кривые коррекции тона и экспортировать результат в TIFF или JPEG со встраиванием EXIF-данных. Программа умеет работать как модуль GIMP, но при этом расчеты происходят в 8-битном режиме. В новой версии программы:

- ▶ включена AHD-интерполяция (Adaptive Homogeneity-Directed Interpolation);
- ▶ добавлена основная кривая, эмулирующая поведение кривой коррекции тона Nikon;

▶ автоматически применяется встроенная пользовательская кривая, если камера настроена для ее использования;

- ▶ поддерживаются прямоугольные пиксели D1X;
- ▶ добавлена поддержка EXIF для libtiff 3.7.4;
- ▶ исправлены некоторые ошибки.

Язык интерфейса:

английский

Домашняя страница:

<http://ufraw.sourceforge.net>

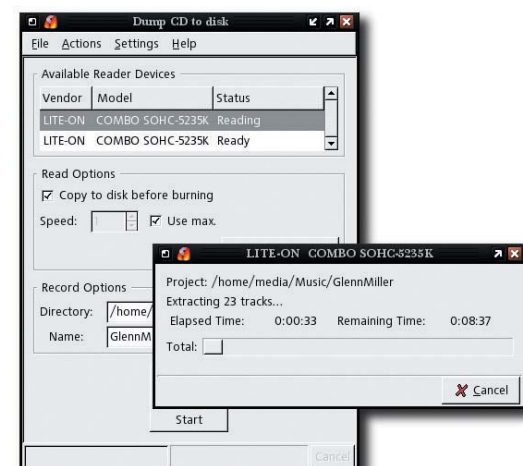
cdrdao и GCDMaster

cdrdao — незаменимый помощник в тех случаях, когда возникает необходимость записи дисков в режиме DAO, или «диска за раз», основанном на описании записываемого содержания в текстовом файле (так называемый TOC-файл). С помощью приложения пользователь получает контроль за продолжительностью пауз между треками (pre-gaps) вплоть до их полного исключения, а также такими данными субканалов как нумерация каталогов, флаги 2/4-каналов,

код ISRC, отметки индексации и т. д. Стало хорошей традицией вместе с выходом очередной версии cdrdao выпускать и обновленный GNOME CD Master — графический интерфейс, упрощающий процесс импорта файлов MP3 и WAV, обеспечивающий удобство при создании собственного аудиопрокта и последующей его записи на CD, копировании с CD на CD или раздел жесткого диска.

Язык интерфейса: английский

Домашняя страница: cdrdao.sourceforge.net



Audacity 1.2.4 и 1.3.0

Вышли две новые версии свободного звукового редактора Audacity. Версия 1.2.4 относится к старой ветке и содержит в основном исправления найденных ошибок. Добавлена обычная для других приложений возможность открыть недавно открывавшиеся файлы. Кроме того, программа теперь использует Vorbis 1.1.

В 1.3.0, напротив, масса новшеств. Помимо мелких удобств вроде быстрого сворачивания и разворачивания дорожек по вертикали появились очень полезные функции, которые, впрочем, еще будут совершенствоваться. Во-первых, это долгожданные множественные области (clips) внутри дорожек и сопутствующие функции разделения и объединения областей.

Во-вторых, новая панель внизу для точного выбора области по хронометражу, при помощи которой также легче определяется текущая позиция во времени. В-третьих, пересмотрен интерфейс дорожки для заметок. Теперь гораздо легче менять границы выделения для каждой заметки. Четвертое немаловажное новшество — возможность пакетной обработки файлов, реализованная как часть проекта

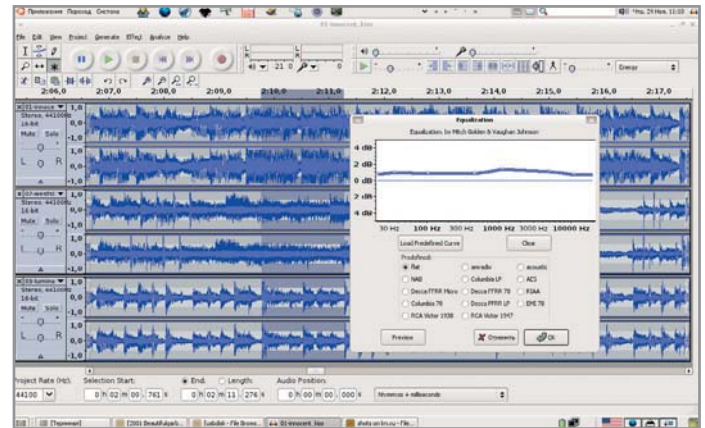
CleanSpeech — расширение возможностей Audacity для автоматической обработки голосовых записей.

И пятая новая функция — загрузка файлов на FTP-сервер прямо из Audacity.

Язык интерфейса: русский

Домашняя страница:

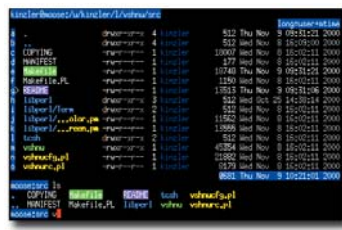
<http://audacity.sf.net>



Всерьез или в шутку пишет Стив Кинцлер о своей работе, что «это название вызывает к индийскому богу Вишне, хранителю вселенной и восстановителю дхармы», но сам vshnu берет начало от известного еще в 80-е годы проекта vsh, а nu — не что иное, как игра слов «nu — new».

В отличие от GNU Midnight Commander, или просто mc, эта графическая оболочка полностью исполнена на Perl, но для

работы с ней пользователю не нужно владеть этим языком. Установка vshnu не займет много времени, она не требует компиляции, и вместе со скачанным архивом к вам попадут подробные инструкции по ее настройке и использованию, а



vshnu

также обзор основных ее достоинств. Для поддержки подсвечивания различных типов файлов графической оболочке достаточно, чтобы это делала ваша командная оболочка. Перед первым запуском vshnu настоятельно рекомендуется проверить окружение оболочки (с помощью env или printenv). В случае некорректной установки TERM и LS_COLORS могут помочь следующие ре-

шения для bash:
`export TERM=xterm-color`
`eval $(dircolors)`
 или для tcsh:
`setenv TERM xterm-color`
`eval `dircolors``
 Итак, пользователь получает дополнительный графический режим в строке терминала и интерпретатор Perl.
Язык интерфейса: английский
Домашняя страница: cs.indiana.edu/~kinzler/vshnu

GIMP 2.3.5

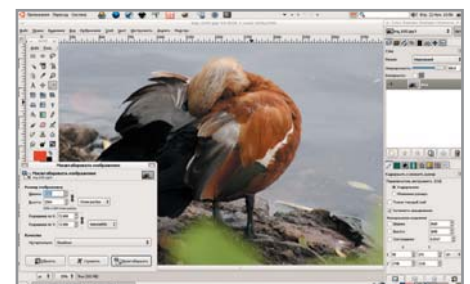
По очередной нестабильной версии GIMP видно, что работа ведется практически во всех возможных направлениях. В категорию «Упростили жизни» можно отнести прилипание к контуру и быстрый доступ к редактору горячих клавиш через меню «Правка». Категория «Хранители нервов» пополнилась оптимизацией отмены/повтора действий, «ускоренной загрузкой про-

граммы и повторной загрузкой данных, оптимизацией SIOX (алгоритм выделения объекта на переднем плане). В категории «На страже сохранности цвета» одно небольшое, но приятное новшество — чтение/сохранение ICC-профиля в файлах формата PNG. Для пишущих собственные модули эффектов добавлен доступ к еще большему количеству внутренних функций. Как

всегда, исправлено много ошибок, заботливо добавленных в код ранее.

Язык интерфейса: русский

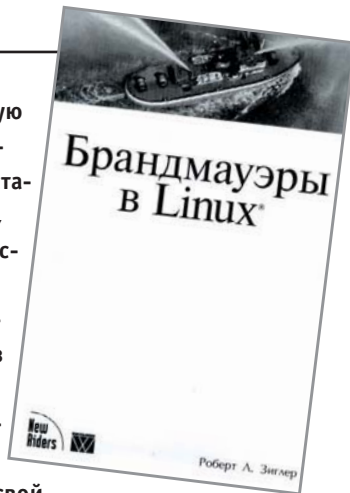
Домашняя страница: www.gimp.org



Сергей Мороз, Александр Прокудин

Когда ваша деятельность напрямую зависит от исправной работы Глобальной сети, какие проблемы встают наиболее остро? Скорее всего, это ненадежность сетевой безопасности. В последнее время темпы развития компьютерных технологий сделали этот вопрос одним из основных в сетевой индустрии, и от его качественного решения зависит комфортная работа огромного числа людей. Как защитить свой компьютер? Как создать безопасный сервер? Как лучше контролировать сетевое подключение? На эти и многие другие вопросы призвана ответить эта книга.

Одним из наиболее современных способов защиты компьютера является использование брандмауэра. Для того чтобы максимально полно раскрыть вопросы организации качественного брандмауэра под Linux, разработчикам пришлось рассмотреть практически все аспекты работы Сети. Дополнительная информация позволяет читателям лучше понять принципы создания и функционирования систем электронной безопасности. На суд читателей авторы предлагают четыре главы. Первая освещает общие понятия функционирования брандмауэра. Во второй подробно разбирается работа с пакетами данных и даются основные рекомендации по усилению безопасности системы. Третья посвящена продвинутым способам защиты. Четвертая часть является фактически блоком дополнительной информации, которая поможет закрепить усвоенные знания после прочтения этой книги. В этой главе вы найдете примеры брандмауэров и ссылки на популярные интернет-ресурсы.



Защита на высоком уровне

Сразу хочется предупредить, что это не банальное пособие по настройке готового брандмауэра, здесь представлена важнейшая информация для создания и оптимизации своего собственного программного продукта.

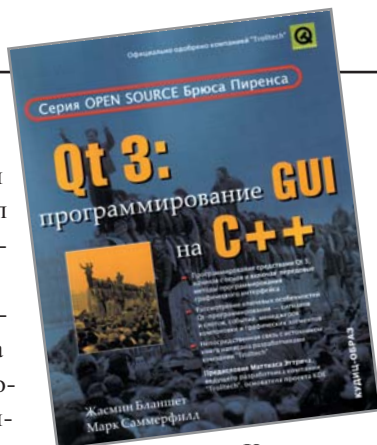
Издание интересно тем, что отвечает на многие вопросы, выходящие за пределы программирования и управления системой безопасности. Можно привести в пример рекомендации по решению проблем с взаимодействием брандмауэра и локальной сетью или основы создания архитектуры с двумя брандмауэрами. Эти «хитрости» позволяют преодолеть множество трудностей, которые могут возникнуть при внедрении в работу firewall. Материалы такого плана встречаются в подобной литературе не так часто, и это выгодно отличает данное издание от конкурентов.

Несмотря на все достоинства книги, нельзя сказать, что она идеальна во всех отношениях. Некоторых может оттолкнуть сложность преподнесенного материала. Для полноценного понимания текста необходимо обладать довольно большим багажом знаний. Одним из недостатков является отсутствие твердого переплета, ведь книга явно не предназначена для одноразового чтения. Шрифт небольшого кегля не очень удобен для восприятия (это издержка небольших габаритов книги).

Если не смотреть на мелкие недочеты, изданию можно поставить положительную оценку. Большой объем материала, расширение стандартной базовой информации о системе защиты, невысокая цена — все эти факторы должны стать определяющими при покупке.

Митчелл Марк, Оулдем Джеффри, Самьюэл Алекс. Брандмауэры в Linux. — М.: Вильямс/New Riders Publishing, 2004. — 288 с.

Пожалуй, в мире разработчиков прикладных программ под Unix мало кто не слышал о кросс-платформенной библиотеке от норвежской компании Trolltech для разработки графического интерфейса Qt. С ее помощью можно создавать интерфейсы практически любой сложности на платформах Linux/Unix, Mac OS X и Windows. На базе Qt было создано множество популярных приложений, в том числе и популярная среда KDE. Теперь, с появлением этой книги пользователи получили отличное руководство по созданию приложений с использованием Qt. Ее авторы являются сотрудниками Trolltech, и кто, как ни они, может до-



Книга состоит из двух частей.

Первая рассчитана на новичков и занимает приблизительно четверть книги. В ней, следуя от простого к сложному, авторы на реальном примере создания приложения электронной таблицы знакомят читателя с технологией сигналов и слотов, разработкой интерфейса с помощью визуальных средств и вручную, а также с использованием виджетов.

Работа с QT в примерах

ходчиво и со знанием дела рассказать об эффективных приемах работы с библиотекой и средствами разработки Qt?

Вторая часть описывает важные аспекты применения Qt на практике — компоновку и создание собственных виджетов, работу с графикой, потоками и сетью, использование баз данных и документов XML, а также рассматривает вопросы интернационализации приложений.

Все описываемые в книге технологии и приемы сопровождаются примерами. Книга настоятельно рекомендуется как начинающим, так и опытным разработчикам графических интерфейсов.

Блانشет Жасмин, Саммерфилд Марк. Qt3: программирование GUI на C++. — М.: КУДИЦ-ОБРАЗ, 2005. — 448 с.

Сольфеджио для Linux

Эту книгу стоит рассматривать как своего рода «пробный шар». До сих пор изданий по мультимедийному свободному софту в России и сопредельных государствах не выходило. Так что на это издание разом легла большая ответственность за то, воспримет ли массовый читатель GNU/Linux как достойную платформу для звукоизвлечения или же решит подождать до лучших времен.

Автор очень удачно подошел к изложению материала. Описание каждой программы начинается с краткого резюме, вслед за которым идет перечисление основных принципов работы программы, ее концепции, если хотите. Только после этого рассматриваются практические аспекты использования программы: настройка, примеры и т. д. Таким образом, читатель не чувствует себя внезапно выброшенным в море консольных команд без спасательного круга.

В оригинале книга носит название «Linux Music & Sound, How To install, Configure and Use Linux Audio Software» («Музыка и звук в Linux. Как установить, настроить и использовать звуковой софт для Linux»), что гораздо точнее русской версии отражает ее содержание. Могу только предполагать, какими соображениями руководствовались переводчик и издательство, но читатель может оказаться в недоумении от несоответствия названия содержанию. Дело в том, что пять лет назад, когда была написана и издана первая версия «Звуковой студии», программы работали по большей части сами по себе и друг с другом никак не стыковались. Это было достаточно «дикое» время, когда трудно было всерьез говорить об удобном и функционально богатом софте для работы со звуком в GNU/Linux. Программ для серьезной студийной работы тогда, по сути, не было.

Обладает ли в таком случае книга хоть какой-то практической ценностью? Ответ, как ни странно, положительный. Разумеется, вместо звуковой подсистемы OSS сейчас однозначно рекомендуется использовать ALSA. Бесспорно,

часть описанных в книге приложений в современных дистрибутивах заработает только после применения большого шершавого напильника. Да, на смену описанным программам пришли новые, куда более совершенные продукты.

И все же достаточно актуальными можно считать несколько глав. Во-первых, информация о трекерных форматах и программах для создания трекерной музыки интересна и сегодня. Просто в то время этот способ сочинительства был в моде наравне с другими, а сейчас для одних он моден как своего рода «винтаж», а для других, даже спустя некоторое время, он остался единственно правильным и любимым.

По-прежнему ценна информация о мощном звуковом редакторе Snd, который с тех пор практически не изменился. Вводная информация по версии Csound для Linux тоже весьма актуальна, поскольку новая, пятая, версия этого знаменитого синтезатора, в котором произошла масса корректировок, официально пока не вышла и все еще не очень стабильна. Не помешает почитать главу о программах для работы с нотными записями. При этом стоит помнить, что интерфейс нотного редактора и MIDI-секвенсора Rosegarden (www.rosegardenmusic.com) заметно похорошел, а вместо мур обычно используется Lilypond. Словом, мысленно делайте поправку на пять лет вперед, и все будет нормально.

Издательство приложило все усилия к тому, чтобы не дать изданию морально устареть, добавив 16-ю главу. Она построена по тому же принципу, что и оригинальный текст, и старания переводчика, написавшего ее, несомненно, заслуживают похвалы. В ней раскрыто несколько тем.

Начинается она с рассказа о звуковом сервере JACK с малой задержкой отклика (low latency), а позднее в ней повествуется об использующем JACK совре-

менном ПО для работы со звуком. Кроме того, приводится фрагмент вступительного урока по Lilypond — средству нотной разметки, при помощи которого можно получать качественно сверстанные нотные партитуры. Рассказано об основных редакторах аудиофайлов (Sweep, Audacity, ReZound), синтезаторе ZynAddsubFX, а также Pure Data — единственной пока визуальной среде обработки звука и графики, имеющей собственное русскоязычное сообщество, пишущее к ней документацию на родном языке. Достаточно обстоятельно в дополнительной главе рассматривается общепризнанный флагман свободного звукового софта — программа для многоканальной записи и сведения под названием Ardour (<http://ardour.org>). К сожалению, при снятии скриншотов с современного ПО переводчик не напомнил читателям, что многие программы уже переведены на русский язык. Очень порадовало качество перевода. Так получилось, что я хорошо знаком с манерой изложения автора. Дейв — человек, несомненно, увлеченный, с очень живым характером и редким ныне умением писать не только интересно, но еще и по делу, без фактических ошибок. Очевидно, что переводчику местами приходилось очень несладко с образной письменной речью Филлипса, но свою задачу он выполнил: скучать при чтении книги уж точно не придется.

В справочную часть издания вошли глоссарий, список ПО на приложенном диске, весьма толковая библиография и, что немаловажно, предметный указатель. Очень надеюсь, что новое, серьезно переработанное издание, ожи-

даемое к выходу на английском языке весной, будет так же качественно переведено на русский язык и выпущено в стенах того же издательства.

Филлипс Дейв. Звуковая студия в Linux. Руководство профессионала. — М.: DiaSoft, 2005. — 448 с.

