

Open Source на корпоративном уровне

Не первый год IBM поддерживает Linux, в свое время этот смелый шаг компании благоприятно сказался на продвижении открытой ОС (отсылаем читате-



лей к восьмой главе «Just for Fun» Линуса Торвальдса). История продолжается, но времена другие: тогда IBM начинала с одного Apache, а совсем недавно Dell объявила о том, что серверные решения компании отныне оснащаются LAMP — Linux, Apache, MySQL и PHP. Дело отнюдь не ограничивается рынком серверов. Dell идет дальше и совместно с Mandriva объявляет о выпуске новых ноутбуков с предустановленным Mandriva Linux Limited Edition 2005. Модель Latitude 110L доступна в базовой комплектации по цене €759.

www.globetechnology.com

Свобода творить

Когда более полугода назад фирма Blender Foundation, в ведении которой находится известная среда для пространственного моделирования Blender, объявила о запуске проекта по созданию свободного короткометражного трехмерного фильма, мало кто воспринял эту новость всерьез. Между тем время шло, работа продвигалась, в стороне от суеты больших событий, незаметно для чужих глаз... Что же происходит в недрах The Open Movie Project, или попросту Orange?

Напомним, проект начинался с совместной работы Blender Foundation и Института медиаискусств Нидерландов над сценарием анимационного фильма, а затем и воплощения его на



экране силами программистов, художников и музыкантов с использованием исключительно открытого ПО. Костяк команды ныне уютно обосновался в небольшой студии в Амстердаме и имеет инфраструктуру, состоящую из шести рабочих станций и файлового сервера, объединенных гигабитной сетью. Сама лента должна увидеть свет в марте будущего года и обещает стать большим событием. Демонстрация фильма планируется даже

на кинофестивале в Каннах и выставке SIGGRAPH. Страница проекта несет в себе и координирующую функцию — команде в Амстердаме оказывают поддержку добровольцы-профессионалы со всего мира. Результат совместного труда выйдет на DVD, в расширенную версию которого под лицензией Creative Commons попадут трехмерные модели, текстуры, наброски и анимация — все, на основе чего и создается фильм. Естественно, помимо DVD фильм

будет распространяться такими же способами, что и свободное ПО. Сделать предварительный заказ фильма можно уже сегодня на сайте blender3d.org — и тогда вы не только гарантированно получите оригинальный DVD, но и сможете

поучаствовать в его создании. На фоне пропаганды открытого ПО явно проступает и другая цель проекта, не афишируемая столь открыто, как прочие. Orange может стать первым в мире результатом творческой работы, за право на просмотр которого с вас не станут требовать каких-либо отчислений и не будут преследовать, если вы надумаете поделиться впечатлениями с кем бы то ни было.

<http://orange.blender.org>

Самый ожидаемый релиз года

Не надо быть пророком, чтобы предположить, что именно OpenOffice.org 2.0 получит в этом году звание самого ожидаемого релиза. Время идет, наиболее смелые разработчики открытых дистрибутивов продолжают включать в них 1.9.x, а между тем поддержка Open Document, с которой связывали будущий офисный пакет, появилась в текущей стабильной, но уже уходящей ветке версии 1.1.5.

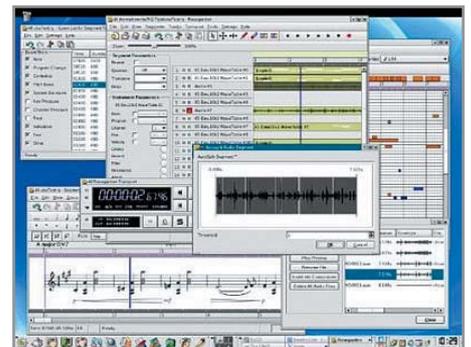
www.openoffice.org



Фабрика звука и видео

AGNULA — проект, начало которому было положено в апреле прошлого года Еврокомиссией, финансировавшей его до недавнего времени. Теперь он стал международным, и его цель — распространение свободного ПО на арене профессионального звука и видео. Сегодня результат большой работы воплотился в AGNULA/DeMuDi — дистрибутиве GNU/Linux, основанном на Debian, который предназначен для профессиональной работы с мультимедиа.

www.ittoolbox.com



Открытая ОС реального времени



В течение последних лет Linux остается в центре внимания разработчиков, нацеленных на завоевание рынка встраиваемых систем. Но до сих пор эту ОС обходили стороной те, кому требовалась производительность в жестком режиме реального времени. Сегодня ситуация меняется и здесь. Работа ведется сразу в нескольких направлениях. С одной стороны, это совместный труд сообщества над соответствующими возможностями ядра 2.6. Очень близко к этому направлению работает TimeSys, строящая свою ОС, отталкиваясь во многом от этой версии ядра. Реальное воплощение детище TimeSys получило в системе управления двигателем Jet Propulsion Laboratory, создаваемой в рамках программы по исследованию Марса.

С другой стороны, есть компании, считающие работу ядра Linux в реальном вре-

мени ему несвойственной и тупиковой; они высказывают собственное мнение о том, каким должно быть сердце ОС, работающей в режиме реального времени. В первую очередь здесь выделяется LynxOS, использующая открытый API, следующая Linux-стандартам и уже прижившаяся в военной сфере. Наконец, FSMLabs предлагает двухядерное решение RTLinuxPro 2.2. Инженеры компании полагают, что именно так можно с успехом сочетать все достоинства архитектуры Linux, оставляя полный доступ к приложениям, не работающим в столь жестком режиме, с производительностью системы в реальном времени. Нарботки FSMLabs с успехом используются в различных сферах. В частности, клиентом RTLinux является компания Pratt & Whitney, работающая с ее помощью над новым двигателем F135.

www.ittoolbox.com



«Мятежный» штат Массачусетс vs Microsoft

Массачусетс принял окончательное решение о всесторонней поддержке формата Open Document. В ответ на это Microsoft лишь отмахнулась колкостью о нелегальности якобы общего кода в StarOffice, OpenOffice, KOffice и IBM Workplace, обеспечивающего работу с Open Document. Это сообщение вызвало возмущение и недоумение у разработчиков, поддержавших данный формат. Страсти не утихают до сих пор: команда KOffice в открытом письме напомнила Microsoft о некоторых принципах независимости разработки, принятых в мире Open Source, соответ-

ствующие заявления сделали и другие адресаты послания от Microsoft. СЮ штата Питер Квин ранее высказал, что «и сейчас, и в будущем все документы, которые мы сохраняем, должны быть доступны каждому без обязательного использования закрытого ПО». Шаг штата в сторону открытого стандарта электронного документооборота затрагивает порядка 50 тыс. пользовательских машин и будет завершён к 1 января 2007 года. Помимо Open Document власти Массачусетса в тех же рамках одобрили формат PDF.

www.mass.gov

Короткой строкой

Выходу нового дистрибутива от Unix-AG не предшествовала рекламная кампания. Тем не менее первые оценки, полученные KNOPPIX 4.0.2, весьма достойные. CD-версия содержит средства разработки, включая GCC. «Пользователь найдет здесь и Emacs; старый добрый KNOPPIX, который мы научились любить и ценить, все еще здесь, в своем оригинальном формате», — сообщает DistroWatch.com.

www.knopper.net/knoppix



Выход Mandriva Linux 2006 — дело ближайших дней. Пока этот номер Chip Linux поступит в продажу, дистрибутив уже найдет своего пользователя, которому предстоит осваивать обещанные нововведения, в числе которых и локальная интегрированная поисковая система Kat. А пока Mandriva тестирует новую страницу своего проекта, которая заменит mandrivalinux.com и mandriva.com.

<http://beta.mandriva.com>

Свежей страницей может похвастать и отечественный дистрибутив ALT Linux. «Новый сайт ALT Linux будет очень хорошим и красивым. Здесь мы будем выкладывать разные тексты — смешные и грустные, но никогда не скучные. Наши пользователи будут довольны и поймут, так же как и наши заказчики. Потому что они все умные. И хорошие. И добрые. За это мы их и любим. Мы рады работать с честными, открытыми и прогрессивными партнерами — свободными людьми и независимыми компаниями. Свободные программы — для свободных людей!»

www.altlinux.ru

Вклад в будущее

Лэптоп за \$100? Процессор 500 МГц от Advanced Micro Devices Inc. и флеш-накопитель вместо жесткого диска с предустановленной ОС Linux предлагают детям специалисты Массачусетского института технологических исследований, находящиеся под сильным впечатлением от посещения одной из камбоджийских деревень. Проект One Laptop Per Child ведется Николасом Нигропонтом и его супругой. На сегодняшний день в планах сбор машин в количестве от 100 до



150 млн штук, что значительно превышает общее число ежегодно выпускаемых компьютеров этого класса. Прототип будет представлен в ноябре; обещается, что он будет оснащен четырьмя портами USB и возможностью работы в беспроводных Wi-Fi-сетях. Потенциальные обладатели новинки живут в Бразилии, Китае, Египте, Таиланде и Южной Африке. Несмотря на все достоинства модели, сам Нигропонт считает, что для его целей даже \$100 — «это все еще слишком дорого».

<http://msnbc.msn.com>

Ложь и правда Symantec

Symantec опубликовала отчет Internet Threat Report за первое полугодие 2005 года, согласно которому Mozilla Firefox содержит 25 уязвимостей, подтвержденных производителем, 18 из которых получили статус крайне опасных, что значительно превосходит аналогичные показатели Internet Explorer — 13 и 8 соответственно. Так ли все серьезно на самом деле? Любому здравомыслящему человеку понятно, что показатель безопасности заключается не столько в ко-

личестве «дыр», сколько в своевременном их устранении. Здесь ситуация диаметрально противоположная. На обнаружение и решение проблем Mozilla тратит несколько дней, в то время как Microsoft может только признание об обнаруженной бреши делать от месяца до двух. Более того, в интервью eWEEK.com Элиас Леви, бывший модератор Bugtrac, ныне работающий в Symantec, упоминает сегодняшнюю, пересмотренную, цифру — 31 уязвимость, официально

подтверждаемую самой Microsoft. Конечно, Internet Explorer тесно интегрирован с другими подсистемами Windows, что дает злоумышленникам дополнительные поводы для «творческих изысканий». Однако это не оправдывает затягивание времени как с признанием наличия уязвимости, так и с выпуском заплаток — например, у IE до сих пор не решена проблема, обнаруженная eEye Digital Security еще шесть месяцев назад...

www.eweek.com

Благотворительность

В сентябре традиционное ежемесячное пожертвование от DistroWatch.com получили разработчики мультимедийного проигрывателя MPlayer. На этот раз присуждение гранта состоялось «под давлением» — на основании многочисленных обращений пользователей к руководству портала. После выполнения ряда формальностей с разработчиками проекта был объявлен и размер суммы — \$400. Всего с момента запуска программы поддержки в марте 2004 года DistroWatch.com пожертвовала \$5305 на развитие свободных проектов:

03/2004:	GnuCash (\$250)
04/2004:	Quanta Plus (\$200)
05/2004:	PCLinuxOS (\$300)
06/2004:	The GIMP (\$300)
07/2004:	Vidlinux (\$200)
08/2004:	Fluxbox (\$200)
09/2004:	K3b (\$350)
10/2004:	Arch Linux (\$300)
11/2004:	Kile KDE LaTeX Editor (\$100)
12/2004:	UNICEF — Tsunami Relief Operation (\$340)
01/2005:	Vim (\$250)
02/2005:	AbiWord (\$220)
03/2005:	BitTorrent (\$300)
04/2005:	NdisWrapper (\$250)
05/2005:	Audacity (\$250)
06/2005:	Debian GNU/Linux (\$420)
07/2005:	GNOME (\$425)
08/2005:	Enlightenment (\$250)
09/2005:	MPlayer (\$400)

<http://distrowatch.com>

План выхода дистрибутивов

Релиз-календарь

октябрь	NetBSD 3.0 Mandriva Linux 2006
6 октября	SUSE Linux 10.0 Ubuntu Linux 5.10 Release Candidate
7 октября	ALT Linux 3.0 Compact (на CD, DVD-9 и LiveCD)
13 октября	Ubuntu Linux 5.10
15 октября	Symphony OS Beta
1 ноября	OpenBSD 3.8
7 ноября	Fedora Core 5 Test1
24 ноября	SUSE Linux 10.1 alpha3



География Linux

Вооруженные силы Франции выбирают высокопроизводительные кластерные решения на базе Linux для проведения технической и научной работы. В настоящее время проводится тендер на установку 64-битного кластера на основе Linux. В последнее время повышенное внимание к этой ОС во Франции становится нормой. Чуть раньше с Windows на Mandriva Linux мигрировали несколько правительственных министерств и ведомств.

<http://news.zdnet.co.uk>

Законодатели Перу после бурного обсуждения практически единогласно (всего при пяти воздержавшихся голосах) одобрили законопроект, официально устанавливающий приоритет свободного ПО. В то же время на сайте TheRegister.co.uk вспоминают, что в 2002 году Билл Гейтс сделал пожертвование перуанским школам в размере \$500 тысяч.

www.theregister.co.uk

Компания HP выбрала Австралию для начала внедрения новых корпоративных решений на базе Linux. В качестве партнеров компании выступают Red Hat и Novell. Заказчики, подписавшиеся на пакеты Mission Critical Services в Linux-окружении от HP, теперь получают тех-

ническую поддержку гиганта вдобавок к поддержке Red Hat и/или Novell.

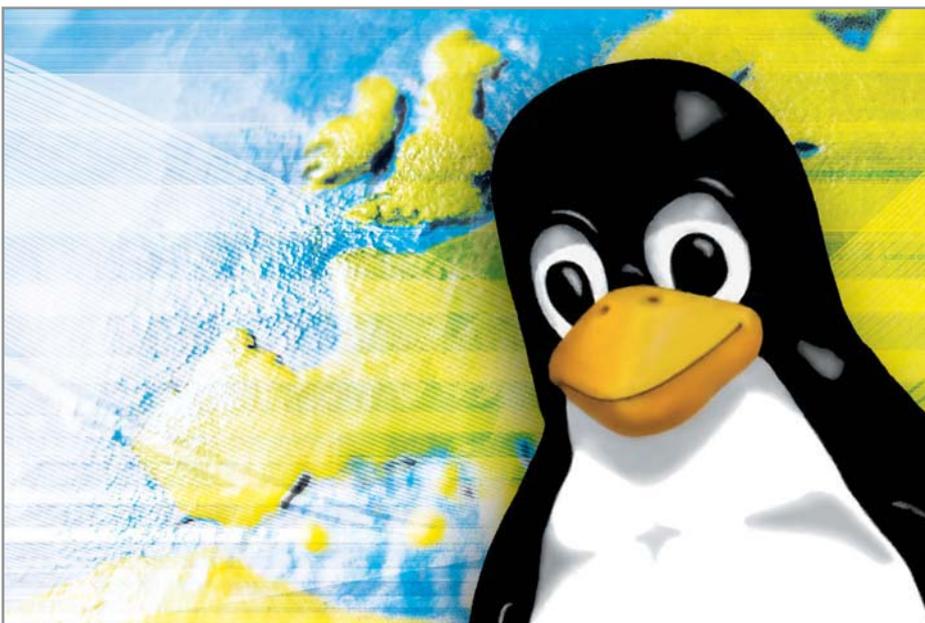
www.linuxworld.com.au

Успех дистрибутива Ubuntu у пользователей приносит и финансовые дивиденды. Можно отметить в некоторой степени рискованное решение компании Ubuntu, позволившей себе вложить часть капитала в южноафриканский ImpiLinux, что говорит об оживлении на африканском рынке, который до сих пор подобным вниманием не баловали. Кроме того, объявлено о серьезных изменениях в планах поддержки Ubuntu своих продуктов начиная с релиза 6.04, планируемого к выходу в апреле будущего года: решения для рабочих столов будут поддерживаться в течение трех, а серверные — пяти лет.

<http://beta.news.com.com>

На азиатский рынок вышел долгожданный Asianux 2.0 — результат совместной работы компаний Red Flag Software, Miracle Linux и HaanSoft. Каждая из них собирает и распространяет дистрибутив под собственным брендом. Едва оказавшись на прилавках, Asianux 2.0 уже получил контракт на работу в создаваемой в Южной Корее Национальной образовательной информационной системе, которая охватит более 10 тысяч школ.

www.linux.com



Короткой строкой

Пользователи Linux увидят на своих машинах Quake 4. Официальное подтверждение этому получили linux-gamers.net от id Software. Имеются и некоторые технические подробности: Quake 4 будет использовать SDL и OpenGL.

www.linux-gamers.net

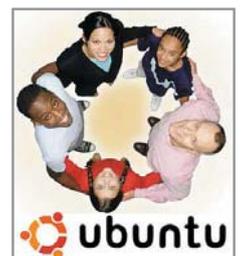


Не успел появиться SUSE Linux 10.0, уже анонсирован выход альфа-версии 10.1, которую сегодня можно скачать с интернет-страницы проекта. Работа над ним, по предварительным оценкам, будет продолжаться до конца февраля будущего года.

www.opensuse.org

Растет популярность дистрибутива Ubuntu — в статистическом рейтинге пользователей на DistroWatch.com он продолжает прочно удерживать первую позицию (2758), оставив далеко позади себя Mandriva (1720), SUSE (1384) и Fedora (1273).

www.distrowatch.com



Мы уже стали привыкать к тому, что обновления популярных браузеров Mozilla Firefox и Thunderbird выходят практически синхронно. Исключения нет и сейчас: появилась версия 1.0.7 — результат усилий разработчиков по достижению большей стабильности и безопасности.

www.mozilla.ru

Как справиться с канцлером

KDE Project представил первую бета-версию KDE 3.5. Чтобы одолеть «Канцлера» (именно такое имя получил релиз), в первую очередь необходимо внимательно ознакомиться со всей документацией. Разработчики обращают особенное внимание на то, что версия не поддерживает Qt 3.3.5. В остальном все как обычно: обратную связь налажена через систему отслеживания ошибок bugs.kde.org, а как подспорье для скачивания и установки пакетов используется Konstruct. Как бы между прочим приоткрыта завеса тайны над KDE 4. Сегодня известно, что на ранней стадии разработки KDE образовывается Arreal group, в которой объединяются усилия художников, экспертов, способных оценить степень удобства работы пользователя с продуктом, и самих разработчиков — в известном духе Vazaar. Также можно говорить о включении в проект платформы контекстной связи Topog и об идеях Plasma для рабочего стола. Удобным решением для сторонних разработчиков, намеренных интегрировать свои приложения с KDE, станет принципиально новая библиотека RuDI. Своеобразным отчетным моментом для KDE Project будет являться запланированный на следующий год aKademy, где будут подведены первые итоги работы над KDE 4.

www.kde.org

Nokia идет в народ



Финская компания привлекает разработчиков на свою платформу в открытый проект Eclipse, призванный стать основой для различных приложений. В частности, планируется создание интегрированных сред разработки на Java для обеспечения полной поддержки J2ME (Java 2 Platform, Micro Edition).

www.zdnetasia.com

Главный критерий — безопасность

Red Hat Enterprise Linux отвечает всем стандартам безопасности в области обеспечения информационного обмена между различными правительственными службами и теперь становится общим критерием соответствия внутри широкого

круга систем IBM eServer. Таковы первые результаты совместных усилий Trusted Computer Solutions, Inc. (TCS), Red Hat и IBM. Отметим, что TCS тесно сотрудничает с департаментом обороны США.

www.eweek.com



Новый офисный стандарт

Red Hat и IBM, высоко оценивая перспективы растущего российского рынка, предложили совместную инициативу по оказанию разработчикам услуг по обучению и технической поддержке в переносе кода на ОС Red Hat Enterprise Linux, а также оборудование от IBM. В глобальном аспекте это означает расширение масштабов разработки и использования приложений Linux. При этом данное решение затрагивает не только Россию, но и азиатские страны — Китай, Южную Корею, Индию. В конкретном приложении эта инициатива реализуется в виде инновационных центров от IBM, где разработчики могут получать необходимую помощь. Что касается роли Red Hat, то компания берет на себя образовательную часть программы. Стало известно о проведении совместной акции «Стань RHCE в 2005 году!» Red Hat

и «Инвента». Программа сертификации специалистов проводится с 15 октября по 31 декабря 2005 года, в ее рамках в Учебном центре «Инвента» всем слушателям предоставляются специальные условия для подготовки и сдачи сертификационных экзаменов Red Hat. При этом любому кандидату, подавшему заявку на экзамен RHCE в период действия программы, будет предоставлена возможность бесплатной пересдачи (в срок до 28 февраля 2006 года) — конечно, в том случае, если первая попытка окажется неудачной. Кроме того, объявлено о специальных предложениях по обучению на авторизованных курсах подготовки к сертификации (стоимость регистрации — от 18 500 до 25 500 рублей, в зависимости от курса).

www.rhd.ru

www.oasis-open.org

Линус о спецификациях, научных теориях и религии

Отвечая в дискуссии на вопрос о включении SAS Transport Layer в главную линию разработки ядра, Торвальдс отметил, что закрывает тему о спес за ненадобностью: «Я никогда не встречал спес, который был бы одинаково хорош для того, чтобы оказаться и полезным, и аккуратным... Это самый худший путь для создания программы, потому что он по определению означает, что программа была написана в соответствии с теорией, а не реальностью... Спецификации опасно кривы. Реальность выглядит иначе, и любой, кто ставит спес выше реальности, должен уйти из работы над ядром прямо сейчас. Пока объективная данность и спес не гармонируют, спес имеет нулевое значение... Это похоже на науку: если у вас есть теория, которая не соответствует опыту, то не имеет никакого значения, насколько она вам нравится. Вы можете использовать ее как приблизительную, но обязаны помнить это всегда.



Дискуссия продолжается. Любен Туйков: «Но это и есть определение теории. Никто не спорит о том, что теория неточна по отношению к наблюдаемому поведению». Линус Торвальдс: «Научная теория приближена к наблюдаемому поведению без известных пробелов. Однажды обнаружив пробелы, научная теория перестает быть таковой... Спецификации не просто едва ли не плохо написаны, они к тому же фактически никогда не соответствовали действительности... В худшем случае это станет для людей причиной пренебрежения реальностью, а затем уведет их к религии». После этого любой комментарий о том, что «наличие большого числа плохих спецификаций не лишает представлений о них законной силы», становится излишним. И так ясно, к чему призывает Торвальдс и от чего он предостерегает.

<http://kerneltrap.org>
<http://slashdot.org>

Brought to you by:
 amaroK & Magnatune
www.magnatune.com

Дерзость берет высоты. Анонсируя новую версию amaroK LiveCD, команда разработчиков выделила десять операций, которые удобно производить с его помощью.

Размах крыльев — 1.3

Например, можно оснастить дисками с amaroK компьютеры на выставке MS для демонстрации того, на что должен быть похож компьютер, хорошо проигрывающий музыку. Также можно отослать его в правительство или лоббистам, показав, что хорошие вещи могут быть свободными, как и музыка. Можно разослать его по почтовым ящикам в своем сетевом окружении, напечатав на диске «Свободное обновление AOL». А еще — заточить края дисков с amaroK LiveCD и использовать их в качестве сюрикенов; собрать из трех дисков бикини для подружки. Наконец, если какое-либо из этих действий является незаконным в вашей стране/коммуне/тюрьме/графстве/империи,

воспользуйтесь здравым смыслом и держитесь подальше от проблем.

Технически все осталось прежним. Основа LiveCD — KDE-ориентированная PCLinuxOS, собранная с целью демонстрации широкого спектра возможностей свежей версии аудиопроигрывателя — amaroK 1.3.2. Как и раньше, на диске выложены новые музыкальные композиции: в этот раз это треки немецких исполнителей Paniq и Snooze, а также норвежцев Ugress. Одновременно запущена кампания по сбору средств, необходимых для обеспечения дальнейшей работы. Поставлена задача получить \$5000 к началу ноября этого года.

<http://amaroklive.com>
<http://amarok.kde.org>

Сергей Мороз

Музыкальная шкатулка

Хороший IT-работник должен быть ленивым, а точнее — по-хорошему ленивым: один раз настроил систему и забыл. Эта идея правомерна не только для рабочих станций или серверов, но и домашних компьютеров, которые обычно используются для того, чтобы побродить по Интернету, послушать музыку или поиграть в игры.

Обычно люди, которые разносторонне используют свой компьютер, часто не хотят тратить время на установку системы, настройку и инсталляцию драйверов, антивирусов, firewall и прочего. Почему-то хочется просто установить систему — и чтобы все заработало сразу и без настройки. Возможно, у меня какой-то особенный компьютер, или мне просто повезло, однако сначала Fedora Core, а потом и Ubuntu без проблем устанавливались, работали и продолжают это делать вполне исправно по сей день.

Так вот именно эта естественная лень и сподвигла меня на подвиг (впрочем, не очень-то и трудный) по настройке — фактически созданию мультимедийного компьютера. У меня возникло желание слушать музыку и управлять ею, не приближаясь к компьютеру и не включая экран. Причем слушать хотелось в хорошем качестве и при этом не загромождать все вокруг множеством аналоговых проводов, идущих от компьютера к усилителю.

| Начальные условия |

Итак, что нам необходимо? В качестве пульта дистанционного управления возьмем карманный компьютер с картой Wi-Fi. Я использовал Fujitsu Siemens Pocket LOOX 730. Сердцем музыкальной шкатулки будет компьютер MSI MegaPC 865 с предустановленным Linux-дистрибутивом Fedora Core 3, оснащенный оптическим цифровым выходом и дополнительной Wi-Fi-картой Orinoco Gold (к слову, заставить работать встроенную по умолчанию Wi-Fi-карту так и не удалось). Обеспе-

чить достойный звук призван ресивер Yamaha RX-V450 с оптическим цифровым входом.

Сценарий действий выглядит следующим образом:

- ▶ настройка проигрывателя для работы с файлами домашней музыкальной коллекции;
- ▶ настройка оптического цифрового выхода звука с компьютера на ресивер;
- ▶ настройка параметров Wi-Fi-карты компьютера для организации соединения;
- ▶ настройка КПК для работы с сетью и управления плеером.

| Настройка плеера |

В качестве плеера, работающего по клиент-серверной модели, я выбрал mpd. Имея опыт работы с двумя ОС (Windows и Linux), могу с уверенностью сказать, что это самый удобный плеер для обеих платформ. Его настройка заняла совсем немного времени. Исходные тексты Music Player Daemon были взяты с интернет-сайта <http://musicpd.org>.

При компиляции плеера обратите внимание на наличие библиотек flac-develop и faad-develop (хотя я компилировал без него), иначе никаких Flac или MPEG-4 он поддерживать не будет. Также понадобится библиотека libao для работы с разными звуковыми модулями. Если в вашей системе ее нет, необходимо будет ее скомпилировать и установить, причем компилировать придется с включенной опцией alsa09.

Для корректной работы с вашей домашней музыкальной коллекцией необходимо произвести настройку следующих параметров файла `/etc/mpd.conf`:

- ▶ **music_directory** — путь к каталогу музыкальной коллекции;
- ▶ **playlist_directory** — каталог для хранения плейлистов;
- ▶ **log_file** — путь к файлу с логами;
- ▶ **error_file** — путь к файлу с ошибками;
- ▶ **db_file** — путь к файлу с музыкальной базой;
- ▶ **state_file** — путь к файлу с текущим состоянием плеера;

Первоначальное создание музыкальной базы производится с помощью команды `mpd -create-db`.

Неплохо было бы установить графический клиент к `mpd`. Я использовал `gmrc`, но ничто не мешает вам установить что-то другое, если вы используете KDE или работаете в консоли. Более десятка подходящих программ можно найти на сайте <http://musicpd.org/clients.shtml>.

Попробуем запустить `mpd`, соединиться с клиентом и послушать качество звука. Для большего удобства не лишним будет проверить, возможна ли работа через обычный аудиовыход. В настройках `gmrc` можно сразу выставить `autoconnect`. После тестирования выключите `mpd` командой `killall mpd`.

Чтобы настроить вывод звука через оптический цифровой выход, отредактируем конфигурационный файл `mpd /etc/mpd.conf`. Для этого посмотрим файлы `/proc/asound/card0/pcmXXX/info` и запомним номера устройств (`device:`) для тех из них, где упоминается IEC958. После этого в `/etc/mpd.conf` в разделе «AUDIO OUTPUT» пропишем следующее:

```
ao_driver "alsa09"
ao_driver_options "dev=hw:0,X" # Здесь X — номер запоминаемого устройства
```

Теперь снова запускаем `mpd`, слушаем звук через ресивер и получаем удовольствие от проделанной работы.

Наконец, вносим последние штрихи в конфигурацию плеера. Для его автоматического запуска при старте системы нужно прописать строку `/usr/bin/mpd` в файл `/etc/rc.d/rc.local` (или в `ubuntu /etc/init.d/bootmisc.sh`). Дополнительно можно внести в `cron` периодическое сканирование каталогов на предмет обновления музыкальной базы:

```
mpd -update-db
```

| Настройка беспроводной сети |

Музыка заиграла, теперь необходимо создать беспроводную сеть (если ее до сих пор не существовало). Тонкости ее настройки в Linux описывались не один раз. От себя могу сказать только то, что неплохо использовать несколько старую, но тем не менее отлично работающую связку — переходник PCMCIA (чипсет TexasInstruments) в сочетании с картой Lucent Tech Origin Gold. Хотя данный чипсет обеспечивает скорость «всего» 11 Мбайт, однако драйверы для него в Fedora Core 3 уже имеются. У меня все прошло без проблем, «магического» запуска потребовала, пожалуй, только лишь `/sbin/iwconfig`. Если утилиты в системе нет, ее можно установить вместе с пакетом `wireless-tools`. Данная связка также недорога по причине возраста, тем не менее обеспечивает отличное качество связи и стабильность в работе.

Естественно, необходимо указать имя сети, адрес адаптера, маску подсети и тип соединения Ad-Нос (если у вас нет Wi-Fi-концентратора) или Managed (если он наличествует).

Наконец, если у вас настроен `firewall`, необходимо разрешить с данного сетевого интерфейса работу с портом 6600.

Delux-версию моей `barebone`-системы MSI MegaPC 865 так и не удалось заставить работать ни со встроенной (и естественно, стоящей дополнительных денег) беспроводной картой, ни со встроенным же и также стоящим дополнительных денег FM-тюнером. Причем, если второй просто не работал (ПО существует только для MS Windows), то с первой происходили странные вещи: при установке драйверов и попытке настроить сетевой интерфейс компьютер наглухо зависал, и спасала только команда `reset`.

| Настройка ПДУ |

Наконец, самое интересное, ради чего все и затевалось, — пульт дистанционного управления.

Создадим и настроим беспроводное сетевое соединение на КПК. В процессе нужно указать имя сети, IP-адрес наладонника и тип соединения Ad-Нос (аналогично настройке «старшего» компьютера можно указать и тип Managed и, соответственно, вписать КПК в домашнюю беспроводную инфраструктуру).

Хотя на самом сайте проекта `mpd` упоминается довольно много клиентов, однако, как мне показалось, самого интересного там нет. К счастью, документацию авторы проекта ведут на Wiki, и вот там как раз можно найти замечательную ссылку на проект `mpd4ppc.mijnflat.net`, посвященный разработке `mpc4mpd` — клиента для `mpd`, запускаемого под PocketPC.

Устанавливаем его на КПК, запускаем, заходим в настройки, вводим IP-адрес «старшего» компьютера, порт (порт в настройках — `mpd`), применяем настройки — и шикарный пульт дистанционного управления готов!

| Идеи для реализации |

А теперь предположим, что можно взять достаточно мощный, но уже старый и недорогой Pentium II (или даже Pentium с поддержкой инструкций MMX), оснастить его жестким диском, оптическим цифровым выходом и картой беспроводной сети. В результате получится готовый музыкальный центр, спрятанный где-нибудь на антресолях, от которого тянется одинокий провод до домашнего ресивера или даже ведется потоковое вещание по той же беспроводной сети (да-да, с помощью `mpd` возможна и такая настройка). Можно также отключить или сильно снизить скорость вентиляторов, так как для декодирования MP3/OGG/Flac не требуется много ресурсов.

Стоимость такой «шкатулки» составит около \$200 — и это в том случае, если у вас случайно не завалюсь где-нибудь необходимое железо, а возможно, и еще дешевле.

Если учесть, что к компьютеру можно подключить районную сеть, многие из которых сейчас предлагают громадные музыкальные архивы, и подмонтировать ресурсы с музыкой или сетевое радио (эти опции тоже поддерживаются `mpd`), то может получиться почти неисчерпаемый источник легко архивируемой музыки, удобно управляемый прямо с дивана. |

Давай с тобой поговорим...

Linux — стабильная операционная система, вольготно чувствующая себя на мелких и крупных серверах. Но разве этим можно убажить пользователей, для которых Apache — это племя индейцев, а не HTTP-сервер; Squid — кальмар, а не прокси?

Запросы наши намного скромнее. Нам чаще всего хочется путешествовать по просторам Глобальной сети, отправлять и получать письма от друзей, слушать музыку, смотреть фильмы, распечатывать фотографии, снятые цифровым фотоаппаратом. А еще мы любим виртуально общаться. В процессе общения мы получаем информацию, делимся опытом и даже влюбляемся, хотя потом порою нервно покусываем свои локти. И все эти радости жизни в Интернете должны быть доступными и способными работать без какого-либо особенного напряжения серых участков головного мозга. Сможет ли Linux обеспечить нам такое непосредственное общение?

| Протокол номер один |

ICQ, или, как ее ласково называют соотечественники, «Ася», является основным средством обмена мгновенными сообщениями во Всемирной сети. На практике ICQ является вторым рубежом виртуальной близости после электронной почты. Сетевое общение в режиме реального времени раскрепощает и ни к чему не обязывает, ведь разговор всегда можно прервать, закрыв окно программы, а расстаться с неприятным человеком навечно можно с помощью его удаления из списка контактов и занесения в список игнора. В отличие от веб-чатов и IRC, куда люди чаще всего приходят, чтобы просто потренироваться в красноречии, общение в ICQ всегда имеет осмысленный оттенок.

Но не только праздное общение — удел ICQ. Хлеб технического специалиста — это знания, а ближайший способ что-либо узнать — это спросить у друга. ICQ, являясь инструментом общения в реальном времени, становится для нас оперативной энциклопедией, откуда можно бесконечно черпать новую информацию. «Не имей сто рублей, а имей сто друзей» — гласит известная поговорка, открывая вечную истину: чем больше у вас умных друзей в списке контактов, тем богаче вы сами — новыми полезными знаниями и оперативной информацией.

Так почему же именно зеленый цветочек прочно обосновался в системном лотке пользователя Всемирной сети? Давайте рассмотрим возможности ICQ как протокола:

- ▶ ICQ позволяет одновременно общаться в режиме онлайн любому количеству пользователей с одинаковыми именами;
- ▶ вы в любой момент времени можете видеть, кто из ваших друзей сейчас находится в Сети;
- ▶ протокол дает возможность обмениваться сообщениями между компьютерами собеседников напрямую, не задействуя при этом промежуточные серверы, что позволяет передавать файлы неограниченного размера, отправлять гигантские сообщения и многое-многое другое;
- ▶ сервер ICQ предоставляет возможность поиска пользователей по любому из основных признаков, а также по совокупности дополнительных данных;

- ▶ протокол позволяет отправлять сообщение неактивному пользователю, который обязательно его получит, как только войдет в Сеть с помощью ICQ-клиента;
- ▶ протокол поддерживает несколько уровней приватности, предоставляя возможность общаться только с избранными пользователями или даже сохранять полную конфиденциальность в Сети;
- ▶ с помощью ICQ вы можете организовать небольшую беседу с несколькими пользователями одновременно;
- ▶ протокол позволяет заносить нежелательных собеседников в черный список и, таким образом, навсегда забыть об их неприятных сообщениях.

Как показывает опрос, проведенный компанией Infowatch, 78% процентов пользователей интернет-пейджеров предпочитают именно ICQ-клиенты. Получается, что для полноценного обмена мгновенными сообщениями сегодня альтернативы фактически нет. Но поддержка других протоколов — это все равно большой плюс для программ, авторы которых вынашивают амбициозные идеи покорения рынка мессенджеров.

Кардинальные изменения протокола остались на стыке тысячелетий, и сейчас поддержку ICQ осуществляют практически все программы. Но даже редкие исключения из правил, такие как программа PSI, позволяют входить в ICQ-сеть через протокол Jabber или каким-либо другим способом.

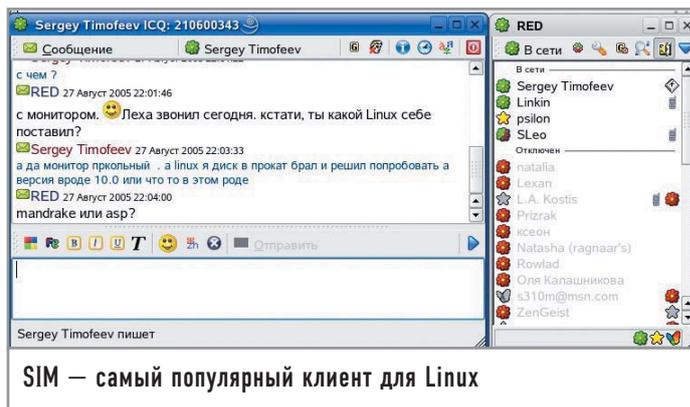
Растущая конкуренция заставляет разработчиков придумывать все новые способы привлечения потенциальных пользователей, клиенты ICQ не похожи друг на друга, в их разнообразии нетрудно запутаться. Попробуем распутать гордиев узел рынка ICQ-клиентов.

SIM 0.9.3

Аббревиатура SIM (<http://sim-icq.sourceforge.net>) расшифровывается как Simple Instant Messenger, что в переводе с английского означает «простой клиент для обмена мгновенными сообщениями». Но простой программу называть не совсем корректно, так как она умеет работать не только с ICQ, но и с протоколами AIM, Jabber, Yahoo! и MSN, а также имеет множество полезных функций.

Для установки мессенджера вам понадобятся культовая библиотека QT и еще несколько модулей. Как следствие — программа вольготно чувствует себя в среде KDE, не засоряя оперативную память лишними библиотеками, выходящими за пределы рабочей среды. SIM — кросс-платформенное приложение, умеющее работать не только в Linux, но и в Windows, а также во всех Unix-совместимых системах.

На вкус и цвет товарищей нет, SIM субъективно является самым красивым ICQ-клиентом. Красота программы — это не только нестандартные темы, репродукции картин Пикассо в виде подложек и сотня великолепных иконок, разбросанных по всей программе. Под красотой можно понимать еще и продуманный интерфейс, лаконичные формы меню, удачную компоновку многочисленных элементов управления. По этим параметрам у SIM точно нет конкурентов. Взяв за основу интерфейс фирменного мессенджера ICQ 1999 года, автор SIM довел его до совершенства.



Вы можете гибко настраивать шрифты в списке контактов и истории сообщений, подключать иконки от Miranda или ICQ, менять темы оформления. Причем смена тем не равноценна простой смене набора текстур. SIM по-настоящему меняет свой облик, то становясь типичным приложением Windows XP, то ударяясь в ностальгию по Windows 9x, а может применять элементы многих графических оболочек из мира Unix.

Вы можете общаться со множеством собеседников внутри одного окна, переключаясь между ними с помощью вкладок. SIM очень удобно организует чат между группой пользователей. Вы просто выделяете в списке тех, кому хотите отправить сообщение. И если все эти пользователи желают общаться, получается небольшая групповая беседа. SIM — единствен-

Дополнительная информация

История создания ICQ

ICQ — это сокращение от фразы «I seek you», что в переводе на русский язык означает «Я ищу тебя». До появления этого протокола единственным способом общения в реальном времени были каналы IRC. Там разговаривали и договаривались о встречах, ведь в 90-е годы встречи пользователей Интернета больше походили на сегодняшние слезы исследователей далеких галактик. Это были действительно события, встречи фанатично преданных делу людей. Науку и культуру двигают молодые. Пушкин начал писать «Онегина» в 23 года и стал легендой поэзии, а четыре молодых студента-выпускника из северной части Тель-Авива решили написать программу для обмена сообщениями в реальном времени. Запомним их имена: Сефи Вигисер, Арик Варди, Яир Гольдфингер и Амнон Амир.

Бакалавры по математике Яир и Амнон и окончившие десятилетку Сефи и Арик в ноябре 1996 года выпустили бесплатную программу с непонятным названием ICQ, которая привнесла в мир виртуального общения совершенно новые идеи. Молодые программисты решили организовать свою фирму. А как еще ее можно назвать, если не женским именем? Но имя выбирали родители одного их программистов, детям такое было доверить, видимо, нельзя. В конце концов, остановились на Мирабилис. Спустя год весь мир уже рукоплескал таланту, мужчины-технари и девушки-гуманитарии гурьбой валили за зелеными цветочками, стучали друг другу, мол, я в сети, и слали свои первые сообщения. Еще через год количество пользователей ICQ достигло шести миллионов.

ный ICQ-клиент, который имеет возможность писать сообщения в Live Journal и может общаться со всеми другими клиентами с помощью кириллицы.

Список уникальных возможностей SIM на этом не заканчивается. Он может сканировать список контактов и проверять, кто находится вне сети, а кто — в невидимом режиме. Только SIM хватает смелости добавлять пользователей в список контактов независимо от их разрешения, то есть без авторизации. SIM «видит» не просто IP собеседника, но еще и его внутренний IP-адрес! Имеется также кнопка «Цитирование» в окне чата, позволяющая при ответе процитировать последнее сообщение собеседника.

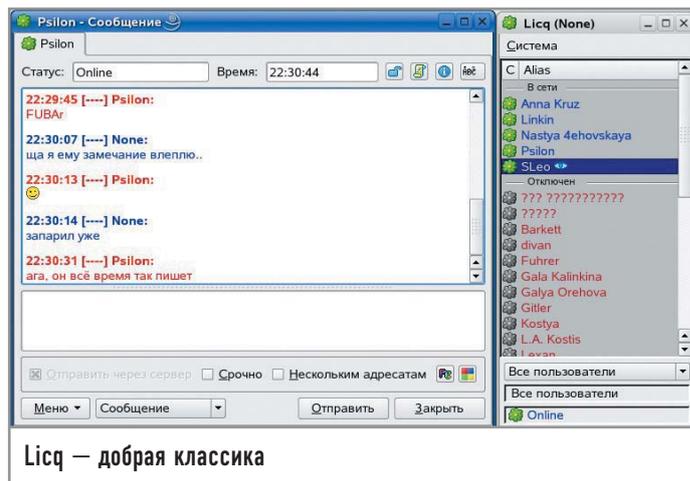
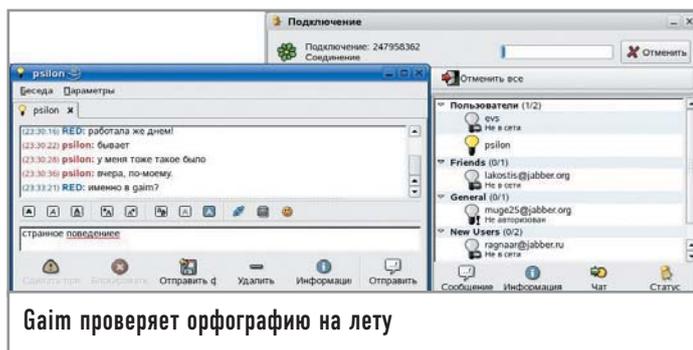
SIM практически единственный клиент, который может проверять орфографию на ходу, сразу выделяя слова с ошибками красным цветом. Есть такая поговорка: «владение русской орфографией — это как владение кунг-фу, настоящие мастера не применяют его без необходимости». Надо признаться, в ICQ мы, «мастера», пишем текст очень быстро, но с ошибками, порой даже не читая собственные слова. С SIM же сообщения обретают небывалую до этого грамотность. Огорчает лишь то обстоятельство, что модуль проверки орфографии надо загружать отдельно, а занимает он почти 12 Мбайт. Но если в вашем дистрибутиве Linux имеется русский словарь aspell, то подобная трата интернет-трафика у вас не предвидится.

Небо над SIM отнюдь не безоблачно. Программа практически не развивается, разработчики постепенно уходят из проекта. Последняя стабильная версия вышла в 2004 году. При старте мессенджер иногда надолго задумывается и со скоростью профессионального шахматиста начинает неторопливо соединяться с протоколами, отличными от ICQ. Любопытно, что это происходит не каждый раз, а лишь изредка. SIM не имеет документированной системы подключения модулей, то есть только сами авторы этой программы могут создавать для нее расширения. Также не хватает хорошей защиты от спама, и иногда SIM очень медленно доставляет сообщения адресатам.

SIM понравится тем, кто ценит универсальность и простоту настройки и предпочитает эстетику в программном мире.

| Gaim |

Необычный ICQ-клиент, разработанный как для Linux, так и для Windows. В Linux программа является стандартным средством обмена мгновенными сообщениями в среде GNOME.



Gaim (<http://gaim.sourceforge.net>) поддерживает ICQ, AIM, MSN, Yahoo!, IRC, Jabber, Gadu-Gadu и Zephyr. Вы слышали когда-нибудь, например, о Gadu-Gadu? Риторический вопрос. Так же как и SIM, этот клиент может на лету проверять орфографию русского языка. Слова с ошибками тут же подчеркиваются красной волнистой чертой.

В отличие от многострадальной версии для Windows, где до сих пор еще не решены проблемы с кириллицей, в родной системе Gaim чувствует себя отлично и позволяет общаться с друзьями на русском языке. Сторонники монолитных продуктов будут обрадованы тесной интеграцией Gaim с почтовым клиентом Evolution.

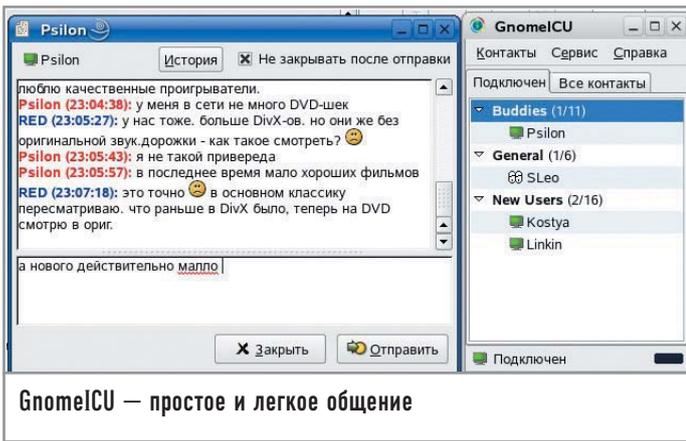
Gaim может работать за стеной прокси-сервера, в настройках программы вы сможете даже вручную указывать прослушиваемые порты. В пункте меню «Конфиденциальность» можно гибко отрегулировать правила приема сообщений во избежание потока спама.

Однако интерфейс программы пугающе непривычен. В списке контактов значки пользователей неприлично крупного размера, кроме того, отсутствуют какие-либо средства настройки внешнего вида списка контактов. Сортировка пользователей по алфавиту — неужели этого достаточно для нас, избалованных обилием альтернативных программных продуктов пользователей?

Информация о пользователе фактически не работает, так как не поддерживается кириллица. Изменение статуса нигде тоже визуально не отображается. Значок мессенджера в системном лотке указывает всего два состояния пользователя — «подключен» или «не подключен».

Справка к программе не прилагается, поэтому ее изучение ложится на плечи самого пользователя. Справка в Глобальной сети, вызываемая из меню Gaim, существует только на английском языке, что несколько затрудняет ее понимание и делает ее неполной: как вам кажется, имеются ли в англоязычной справочной статье пункты, описывающие работу клиента с кириллическими шрифтами?

Gaim подходит любителям оболочки GNOME, а также тем, чей мозг не заражен привычкой работать со стандартными ICQ-клиентами. Программа рассчитана на тех, кто хочет иметь один инструмент для всех протоколов сразу и не возиться с настройкой множества разноликих программ.



| Licq |

В мире Windows есть группа программ, намертво вшитых в операционную систему. Их ругают, стонут от неизбежности их присутствия, но все равно используют. В Linux, к счастью, такого явления не существует, мы всегда можем удалить одну программу или среду целиком и полностью посвятить себя работе с другим программным пакетом.

Licq (www.licq.org) производит впечатление неотъемлемой части KDE — настолько органично вписывается этот ICQ-клиент в популярную рабочую среду. На самом же деле Licq является независимым программным продуктом.

Программа без дополнительных модулей работает только с протоколом ICQ, что отпугивает любителей альтернативных протоколов. Licq не подходит почитателям Jabber, которых немало среди пользователей Linux.

С помощью внешнего модуля возможно подключение протокола MSN. Процедура эта довольно нелогична и вызовет затруднение у пользователя, впервые осваивающего программы обмена мгновенными сообщениями. Для того чтобы воспользоваться этим протоколом, в менеджере модулей загрузите поддержку MSN, затем в меню «Системные операции» выберите пункт «Owner Manager». И уже оттуда вы можете добавить свою учетную запись в сети Microsoft Network.

Licq прекрасно справляется с кириллическими шрифтами, причем поддержка не ограничивается стандартной кодировкой cp1251. Мессенджер может менять темы оформления, наборы иконок, звуки. Так же, как и в SIM, вы можете посылать сообщения сразу нескольким адресатам.

Программа имеет удобную «шпаргалку» в разделе справки, где на русском языке дается множество советов по работе с программой, а также описываются все горячие клавиши.

Licq придется по душе старожилам, ведь ее интерфейс до боли напоминает Mirabilis ICQ 1998. В настройках удобно разместились все, что может понадобиться непритязательному пользователю. Родство становится еще более очевидным, когда всплывает тот факт, что Licq понимает файл списка контактов от Mirabilis ICQ.

Говорят, что в каждой бочке меда есть ложка дегтя, но в некоторые бочки деготь, похоже, наливают кастрюлями. Первая порция отвратительного зелья — это отсутствие кириллицы в списке контактов (однако в сборках этой программы, к примеру в ALT Linux, такой проблемы нет). Вторая — невозможность задания одного имени пользователя для тех, кто работает с несколькими протоколами. Впрочем, отнесем это к особенностям программы, ведь она не оптимизирована для работы с протоколами, отличными от OSCAR (ICQ). Отправка сообщений происходит только с помощью сочетания клавиш «Ctrl+Enter», перенастроить это безобразие на простое нажатие «Enter» не удастся. Внешний вид истории сохраненных сообщений выглядит довольно убого в плане эргономики, на экране остается много пустого места. И если, например, в SIM это легко исправляется в настройках (по умолчанию история переписки в SIM имеет схожий вид), то Licq заставляет смириться с задумкой авторов.

Очередная порча «меда» происходит при просмотре детальной информации о пользователе. Licq, как и Gaim, не имеет поддержки кириллицы (что также решено в российских сборках). Любителям красивых значков и удобных панелей инструментов делать здесь нечего, так как все управ-

Дополнительная информация

«Jabber — протокол настоящих перцев и джедаев» ©Константин Лепихов, технический администратор Mozilla.ru

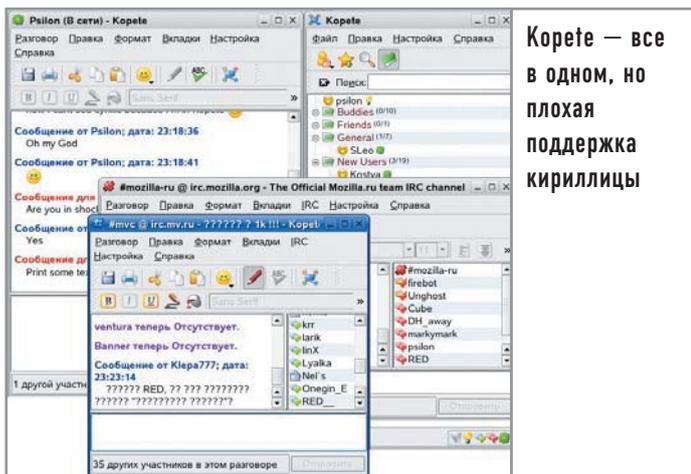
Jabber — это открытый протокол, использующий XML, для быстрого обмена сообщениями и информацией о присутствии между любыми двумя абонентами в Интернете. Этот протокол похож на другие, используемые в Сети, но имеет ряд преимуществ по сравнению с коммерческими системами обмена сообщениями:

- ▶ протокол Jabber является свободным от лицензирования, открытым и общедоступным;

- ▶ с помощью пространств имен в XML можно расширить протокол Jabber для выполнения требуемых задач и обеспечения поддержки взаимодействия между различными системами;
- ▶ кто угодно может запустить свой собственный сервер Jabber, это позволит организациям и частным лицам заниматься любыми экспериментами с обменом сообщениями;
- ▶ любой сервер Jabber может быть изолирован от об-

щедоступной сети Jabber, многие из вариантов реализации сервера используют SSL при обмене между клиентом и сервером, и немало клиентов поддерживают шифрование с помощью PGP/GPG внутри протокола. Но не все так гладко на пути этого молодого и перспективного протокола. Кроме того, что он не популярен за пределами узкого круга системных администраторов, у него есть еще немало недостатков и нерешенных проблем:

- ▶ взаимодействие между Jabber и другими системами до сих пор осуществляют шлюзы на стороне сервера;
- ▶ некоторые серверы Jabber предоставляют возможность взаимодействия с другими системами обмена сообщениями через шлюзы, но Jabber не был ориентирован на обеспечение взаимодействия с другими системами;
- ▶ сообщество в Jabber не монолитно; существует большой набор общедоступных и закрытых серверов Jabber.



Kopete — все в одном, но плохая поддержка кириллицы

ление программой сосредоточено в меню «Система», освоение которого требует усилий и времени.

Licq полюбят те, кому хочется иметь простой инструмент для работы с протоколом ICQ. Сторонники KDE будут рады иметь на рабочем столе ICQ-клиент, гармонично вписывающийся в общий стиль среды.

Между прочим, Licq можно запустить и в консоли. Для этого введите команду:

```
licq -p console
```

И вот перед вашими глазами ICQ-клиент, для работы которого не требуется даже X-сервер. Достаньте свой старенький пыльный компьютер из кладовки, поставьте Linux без графики, и вперед — в Интернет с links, за почтой посредством Mutt и в ICQ с помощью Licq!

| GnomeICU |

GnomeICU (<http://gnomeicu.sourceforge.net>) очень простой ICQ-клиент для среды GNOME, лишенный множества вспомогательных функций. В его простоте кроются и достоинства. Программа работает очень стабильно, в ней мало ошибок.

Дополнительная информация

Как бороться со спамом в ICQ

Довольно быстро, после того как вы подключились к сети ICQ, вам начинают приходить подозрительные сообщения от незнакомых людей. Обычно это начинается так: «Привет, как поживаешь, солнышко?» Вы, ничего не подозревая, пишете в ответ «Все хорошо, работаю вот». И после этого внезапно сваливается «А ты купи слона у нас! Он такой большой и красивый. \$7000. Адрес, телефон». И настроение сразу падает. Что сделать, чтобы сэкономить свои нервы и сердечное тепло для тех, кто этого достоин?

- из информации о себе на сервере уберите страну, в которой вы живете, и не заполняйте графу «Языки»; спамеры чаще всего ищут вас именно по этим параметрам;
- требуйте авторизации для принятия сообщений, в ином случае поставьте блокировку пользователя;
- можно вообще указать клиенту — принимать сообщения только от пользователей из списка контактов;
- работайте в невидимом режиме; ваше обнаружение остается теоретически доступным, но более сложным.

Поддержка кириллицы выполнена на должном уровне. Вы можете общаться со всеми собеседниками по-русски, а также видеть символы кириллицы в списке контактов. Удивительно, но кириллица видна даже во время просмотра информации о пользователе! Если вы работаете в среде GNOME, вам доступны звуки и всплывающие окна. GnomeICU поддерживает смену тем и значков.

Что же плохого в ней? Программа имеет очень ограниченный спектр возможностей. Вы можете работать только через протокол OSCAR, к тому же GnomeICU не поддерживает передачу сообщений в формате RTF (только обычный текст), отсутствует и настройка горячих клавиш. К GnomeICU сложно придаться в плане некорректности чего-либо, но упрекнуть разработчиков в отсутствии множества сервисных функций довольно легко. Наверное, им надо пообщаться с программистами из компании Mirabilis, те их точно научат нашпиговывать программные продукты обилием сервисов.

GnomeICU создан для тех, кто ценит стабильность и для кого ICQ — не предмет культа, а просто рабочая лошадка.

| Kopete |

Стандартная программа, входящая в состав KDE, построенная по принципу «все в одном». Kopete (<http://kopete.kde.org>) может работать с протоколами AIM, Gadu-Gadu, GroupWise, ICQ, IRC, Jabber, Meanwhile, MSN, SMS и Yahoo! Программа оптимизирована для одновременной работы со множеством протоколов благодаря их тесной интеграции. Вы можете создавать «метаконтакты», в которых у пользователя прописывается сразу несколько протоколов. Аналогичная функция есть только в рассмотренном выше SIM. Интерфейс программы выдержан строго в стиле KDE, поэтому не вызывает особых трудностей в освоении. Программу можно заставить загружаться вместе со средой KDE штатными средствами. При появлении новых сообщений или иных сетевых событий всплывает симпатичное окошко, которое невозможно не заметить, однако в то же время навязчивым его назвать нельзя. Все сообщения можно просматривать в пределах одного окна во вкладках.

Уникальной особенностью Kopete является возможность выделения разными цветами отдельных контактов. Например, вы одновременно общаетесь с десятком друзей и одной-единственной любимой девушкой, каждое сообщение от которой ускоряет биение сердца на 20%. Выделите значок контакта с ее именем другим цветом, позвольте своему сердцу раскрутиться до запредельных оборотов. А если вы находитесь на нескольких IRC-каналах, то выделение различными цветами позволяет осуществлять оперативную навигацию по ним.

Без дополнительных модулей Kopete не может похвастаться богатством возможностей. Но, вооружившись дополнительными программами, можно смело идти в функциональный бой с конкурентами. Дополнительные модули позволяют Kopete производить автозамену в сообщениях (подобие макросов), вести их шифрование, настраивать фильтры на входящие сообщения. Очень странно, что история сообщений выполнена также в виде отдельного модуля, ведь это базовая

функция всех мессенджеров. Korpete может осуществлять перевод сообщений на другие языки мира с помощью службы переводов. У программы есть подробная справка, которая позволяет решить многие вопросы.

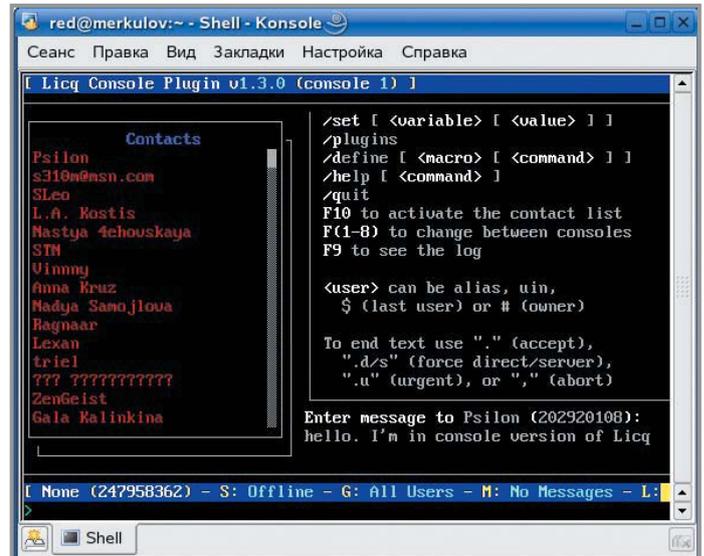
А теперь поговорим о самой многозначительной функции Korpete. Если у вас установлена веб-камера, то программа автоматически определяет, находитесь ли вы возле компьютера. Можно предположить, что к следующей версии программа научится предупреждать собеседников о том, что вы, например, ковыряете в носу.

Но это только начало. Программа совершенно не понимает кириллицу, несмотря на то что в настройках явно присутствует указание кодировки текста. Список контактов также лишен поддержки кириллицы. Korpete не умеет работать через прокси, регулярно вылетает во время путешествия по меню настроек. Несмотря на то что в контекстном меню программы есть пункт, связанный с информацией о пользователе, там ничего не показывается. Все эти недостатки ставят крест на Korpete для большого количества потенциальных пользователей.

Она придется по душе лишь тем, кто ценит интеграцию, множество нестандартных функций, а также ведет активное общение при помощи различных протоколов. Korpete также вполне устроит знатоков иностранных языков, основанных на латинице, и тех, кому не лень набирать сообщения на русском языке транслитом.

Выводы

Что же выбрать простому любителю путешествовать по просторам Всемирной сети? Или вы уже опытный пользователь пейджеров, и по вашему списку контактов можно составлять перепись населения? Если у вас слабый компьютер или ваша душа не воспринимает тяжелые рабочие среды KDE и GNOME, то консольный вариант Licq станет для вас прекрасным выбором.



Консольный Licq — строгость интерфейса и экономия ресурсов

Мир KDE любезно принимает программы, использующие библиотеку QT. Это красивый и мощный SIM, привычный и лаконичный Licq, а также навороченный Korpete. А сторонники GNOME одобряют либо универсальный комбайн Gaim, либо простой и стабильный GnomeICU.

Впрочем, однозначного ответа на вопрос, что же лучше, нет, да и быть не может, иначе не было бы всех этих жарких споров, не существовало бы до сих пор некоторых проблем с совместимостью. Однако одно можно сказать точно: протокол, которому через год исполнится 10 лет, даже и не собирается морально устаревать. Количество пользователей ICQ по всему миру и по сей день неуклонно растет. И все большее число любителей пообщаться в Интернете каждый раз с замиранием сердца открывают окно сообщения, едва услышав ироничное и знакомое приветствие «o-ou!». |

Комментарий специалиста

Работа русскоязычных пользователей в Korpete

Комментирует А. Черепанов, ведущий координатор локализации KDE на русский язык. В последнем стабильном релизе Korpete 0.10.3 из KDE 3.4.2 вы можете без проблем работать со следующими службами, отправляя и принимая сообщения с кириллицей:

- ▶ AIM;
- ▶ ICQ;
- ▶ IRC (через серверы, поддерживающие Unicode, например utf8.rusnet.org.ru);
- ▶ Jabber;
- ▶ MSN;
- ▶ Yahoo!

Имея только одну учетную запись Jabber (например, на сер-

вере <http://jabber.krasu.ru>), вы можете обмениваться сообщениями с пользователями Jabber, AIM, MSN, Yahoo! и ICQ (если, конечно, имеете зарегистрированные учетные записи в этих службах). Пользователь ICQ через учетную запись Jabber будет иметь следующий вид: 1111111@icq.jabber.krasu.ru. Список поддерживаемых Jabber-сервером служб вы можете посмотреть, выбрав в контекстном меню учетной записи Jabber (иконка с изображением лампочки внизу главного окна Korpete) пункт «Службы...».



Цифровая мастерская

В области обработки изображений GIMP практически ничем (разве что кроме работы с цветом) не уступает Photoshop. Главное — разобраться в обилии его инструментов и средств и применять их в меру и с пользой.

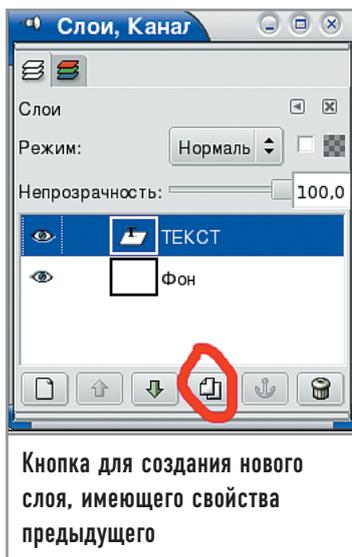
В этой статье вы познакомитесь с некоторыми эффективными приемами работы с GIMP. Некоторые из них подходят и для других программ создания и обработки растровой графики, поскольку универсальны (например, создание тени). Примеры даны исходя из названий меню в установленной у меня версии GIMP 2.2.4 из состава дистрибутива Mandriva 2005 Limited Edition. Не исключено, что в следующих версиях некоторые пункты меню будут переименованы — нет постоянства в этом мире. Но приемы работы наверняка останутся прежними. Перейдем же скорее к ним.

| Эффекты | | Создание текста с тенью |

С помощью описанного далее способа можно сделать тень не только для текстового блока, но и вообще для любого графического объекта. Просто с текстом этот эффект наиболее очевиден, да и используется он гораздо чаще. К сожалению, в GIMP нет специального средства для создания тени, подобного встроенному в Photoshop, где вы можете открыть окно эффектов слоя, включить тень и настроить ее многочисленные параметры. В GIMP можно добиться примерно таких же выдающихся результатов, но только вручную.

Разберем на примере, как же делать тень. Создадим новый пустой файл, затем сделаем на нем текстовый слой (с помощью инструмента «Текст» из палитры инструментов). Пусть для наглядности демонстрации тени цвет букв у нас будет не черный, а какой-нибудь другой — например, красный. В палитре слоев у нас теперь два слоя — фон и один текстовый слой. Нажимаем кнопку дублирования слоя, выбрав предварительно текстовый слой текущим.

Теперь у нас два текстовых слоя идут один над другим. Делаем активным верхний слой, затем снова выбираем инструмент «Текст» и щелкаем им по текстовому слою. В окне «Параметры инструмента» (оно вызывается, кроме всего прочего, с помощью двойного щелчка мыши по иконке инструмента на палитре) задаем цвет этого слоя — пусть он будет черным. Именно этот слой будет служить тенью. Еще один важный момент: степенью прозрачности тени можно управлять с помощью ползунка «Непрозрачность», который можно найти на палитре слоев. Это важно, если вместо белого у вас текстурный фон или даже фотография, и вам необходимо сделать так, чтобы тень не просто накладывалась на фон черными контурами, а было видно, какие именно участки изображения затеняются.



Уже неплохо, но резкость тени делает изображение несколько плакатным

Тень с размытыми краями выглядит более реалистично

Итак, у нас уже есть два текстовых слоя, причем черный (тот, который с тенью) лежит сверху. Рядом с кнопкой дублирования расположена кнопка с изображением направленной вниз стрелки. Она посылает выбранный слой на уровень ниже. Нажимаем на нее — и тень скрывается под оригинальным, первым текстовым слоем, но слой с тенью по-прежнему активен. Выбираем в палитре инструментов иконку с направленными в разные стороны стрелочками (либо нажимаем на клавиатуре букву «М»). Беремся за текстовый слой (в окне картинки, разумеется) и тянем его в сторону, смещая таким образом тень. Должно получиться нечто вроде изображенного на иллюстрации сверху.

А вот другой, более тонкий способ создания тени. Допустим, у нас снова имеются фон и текстовый слой. Инструментом «Выделение связанных областей» (клавиша «Z») выделяем фон вокруг текста и применяем операцию инвертирования выделения (пункт меню «Выделение → Инвертировать»). Теперь текст выделен.

Создаем пустой слой — не текстовый, а обычный. К выделению применяем последовательно операции увеличения и растушевки границ (из меню «Выделение»): для примера возьмем пять пикселей для увеличения и 15 — для растушевки.

Теперь переместим этот новый слой на один уровень ниже, под текстовый слой. С помощью инструмента «Ведро» заливаем выделенную область черным цветом. Стрелочками смещаем тень в нужное положение. В итоге посредством увеличения выделенной области и ее растушевки мы получили тень, имеющую плавные края.

| Создание пламени |

В GIMP довольно простым способом можно создать натуралистичное пламя. Во-первых, нам понадобится градиент, который, в свою очередь, создается следующим образом:

- ▶ Выбираем цвет фона — красный, цвет переднего плана — желтый (или же наоборот).
- ▶ Открываем окно списка градиентов, выбрав его либо на главной панели инструментов, либо в палитре параметров для инструмента «Градиент».

- ▶ В списке градиентов выбираем текущий бело-оранжевый градиент «Основной в фоновый RGB», нажимаем кнопку его копирования (оригинал редактировать нельзя) — это будет третья кнопка слева.

- ▶ Появилось окно правки копии градиента. Теперь нам нужно сделать так, чтобы красная составляющая не занимала столь незначительное пространство, то есть красную часть смещаем в сторону. Можно также добавить другие цветовые сегменты, например синий, если вам нужно, чтобы пламя в основании было немного синеватым.

- ▶ Сохраняем градиент и возвращаемся к окну с изображением. Там рисуем градиент так, чтобы внизу было желтое, а сверху — красное. Но пока это еще не пламя. Это дрова. И нам их надо поджечь.

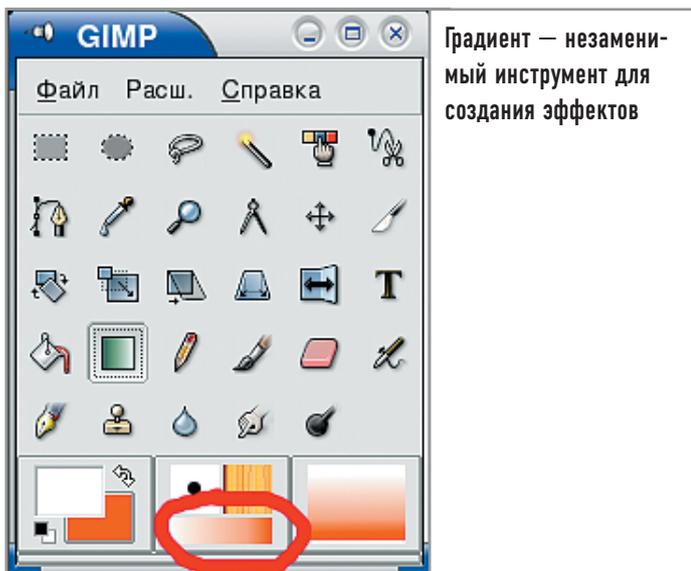
- ▶ Идем в меню «Фильтры → Искажения → IWarp». Откроется окно, в котором, изменяя параметры, щелкаем мышью в экране предварительного просмотра. Тут действия должны быть чисто интуитивными — просто нужно добиться в результате искажений, которые будут выглядеть как языки пламени.

Как вариант — при создании градиента сделайте его часть прозрачной, чтобы потом, при зарисовке области градиентом, эта прозрачность оказалась сверху. Таким образом, пламя будет выглядеть более натуральным, «небо» над ним останется прозрачным, что очень важно, если огонь у вас помещен на фотографию или рисунок, которые вам необходимо разместить позади пламени.

| Обработка фотографий |

Цифровая обработка фотографий — область достаточно обширная, и охватить все ее аспекты в этой статье, разумеется, нельзя. Я коснусь лишь самых распространенных задач, решаемых при работе со снимками.

Фотографии условно можно разделить на две категории по способу их попадания в компьютер. Первая категория — это отсканированные снимки, вторая — фотографии, импортированные с цифровых камер. Как правило, люди сканируют изображения довольно небрежно. Возможно, потому, что этот процесс довольно скучен, а подбирать оптимальные настрой-



Пламя получилось довольно абстрактное, но для оформительских работ пригодится

ки для каждой фотографии весьма утомительно. Да и не всякий сканер хорошо передает оригинальные цвета. Не ошибусь, сказав, что в большинстве случаев именно отсканированные снимки требуют большего труда по приведению их в надлежащий вид. Особенно это касается новых фотографий, сделанных дешевыми мыльницами со вспышкой в квартире.

Разумеется, цифровые снимки тоже бывают отвратительно-го качества, но так получается, что особо гнусные экземпляры удаляются в первые минуты после съемки, когда фотограф просматривает отснятый материал. А распечатанные на бумаге выкидывать жалко, их хочется исправить. И исправляют.

Базовых операций коснемся вкратце. У отсканированных фотографий надо отрезать темные края, используя инструмент кадрирования («Shift+C»). Цифровым изображениям такая обработка не грозит. Но и тем и другим может понадобиться функция переворота: «Изображение → Преобразование → пункты меню из серии «Повернуть/Вращать». И еще о кадрировании: в том же меню «Изображение» есть две функции — «Autocrop image» и «Zealous Crop». Они отвечают

за автоматическое кадрирование, однако справляются с этой задачей не очень хорошо, поэтому кадрирование вручную оказывается гораздо более эффективным.

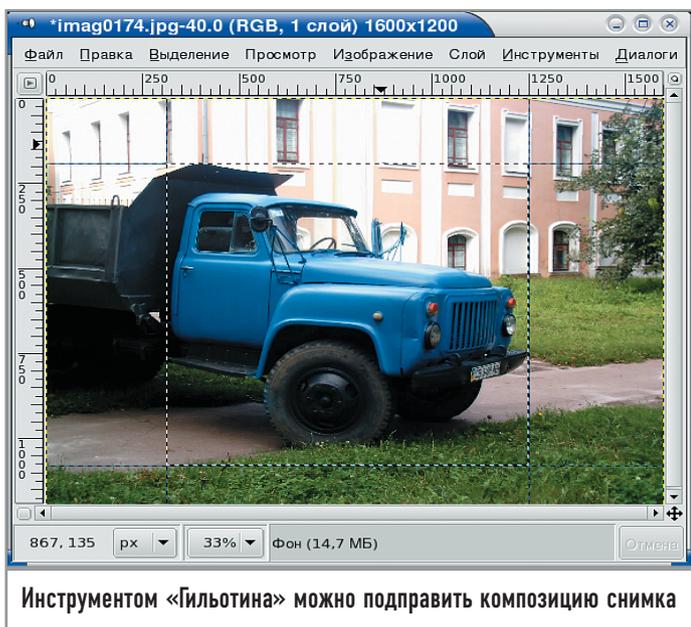
Завершая тему, упомяну о «Гильотине» («Изображение → Преобразование → Guillotine»). Вообще говоря, подобного рода устройства существовали задолго до того, как доктор Джозеф Гильотин из гуманных соображений предложил заменить палача с мечом на механическое устройство казни, что затем было воплощено в железе Антуаном Луи. Но гильотина в GIMP играет не столь устрашающую роль. Она всего-навсего рассекает на части... картинку.

В GIMP есть очень удобное средство под названием «направляющие». Если вы посмотрите на окно с открытым файлом, то заметите линейки по краям. Они находятся там не просто так. Если взяться мышью за такую линейку, то указатель мыши приобретет вид руки. Затем можно потащить мышью вниз, не отпуская кнопку, и вы увидите, как появится линия направляющей. Эту направляющую можно вытащить сколько угодно (а потом так же легко убрать с помощью пункта меню «Изображение → Направляющая → Remove all guides»). А еще можно создать направляющие из выделенной области: «Изображение → Направляющая → New guides from selection». После разметки картинки таким образом примените функцию гильотины, и GIMP разрежет изображение по линиям направляющих, поместив каждый кусочек в отдельный, специально созданный новый файл.

Если вы работаете с JPEG-изображениями, то измененные фотографии лучше сохранять в каком-нибудь другом формате, например BMP или TIFF, либо в полноцветном PNG, где сжатие происходит без потерь качества.

Исправление цветов

Основной мотив в процессе обработки фотографий — это цветокоррекция. Она необходима изображениям, снятым с неправильно заданными параметрами освещения. Во многих фотоаппаратах можно выбирать «погоду» — ясно, облачно, электрический свет, автоматика, да еще и уровень яркости устанавливать. Автоматика не всегда срабатывает хорошо,





Необработанная фотография имеет блеклые цвета



После «вытягивания» цветов снимок выглядит гораздо лучше

да и вручную, бывает, так «удачно» подберешь параметры, что потом задумчиво чешешь затылок. В случае работы с аналоговыми фотоаппаратами важную роль играет пленка, точнее ее характеристики. Поставили «двухсотую» пленку, которая отлично подходит для пасмурной погоды, отсняли солнечный день — что-то небо совсем не то. А вот при использовании самой дешевой пленки (не буду называть ее марку) изображения получаются с ярко выраженным красноватым оттенком, который на распечатанных фотографиях еще и усугубляется с течением времени.

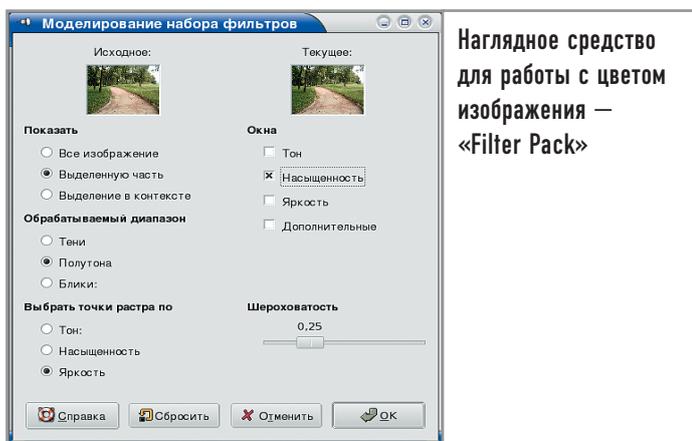
Удобнее всего исправлять цвета с помощью окна настройки уровней. Скрывается оно в меню «Слой → Цвет → Уровни». Здесь вы можете очень тонко настроить уровень каждого канала, а также отрегулировать яркость. На темных фотографиях отлично работает кнопка «Авто», которая автоматически подбирает значения уровней. Будьте осторожны — эта кнопка имеет обыкновение «пересвечивать» изображение, слишком усиливая его яркость. Но во многих случаях она оказывается очень полезной.

Впрочем, и сам снимок (вверху страницы) сделан не очень удачно — небо совсем блеклое, и чтобы исправить этот недостаток, пришлось бы, наверное, подкрашивать его вручную. Небо в оригинале выглядит совсем безнадежно.

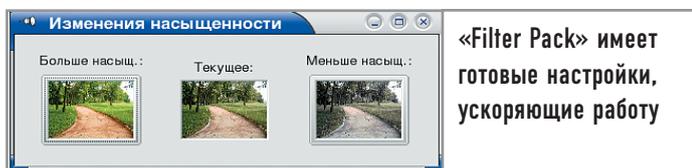
Кроме автоматического режима, разумеется, можно использовать ручную подстройку каналов. Например, если изображение красноватое, то в списке каналов выбираем красный и начинаем осторожно играть с ползунками в разделах «Уровни на входе» и «Уровни на выходе».

Другое полезное средство работы с цветами вы найдете в меню «Фильтры → Цвет → Filter Pack». Это набор фильтров, который позволяет изменять тон, осветление и насыщенность картинки. Кроме дополнительных окон подстройки параметров он еще и весьма нагляден.

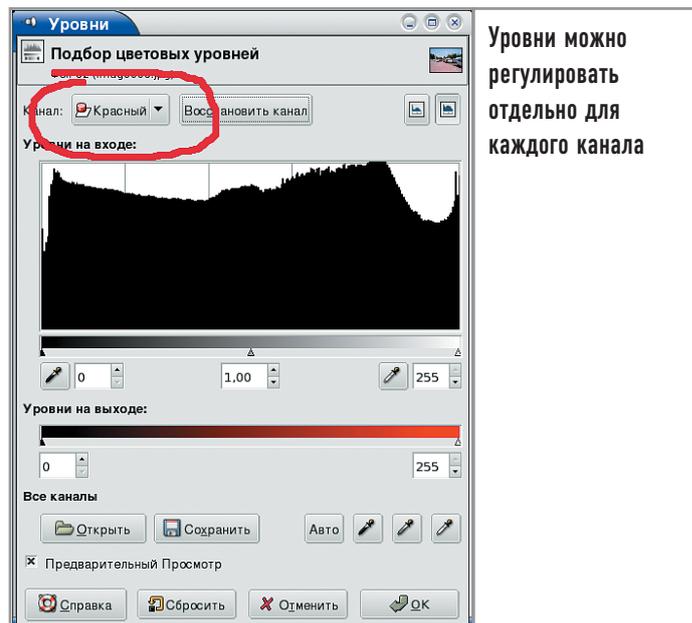
Что до исправления цветов с помощью кривых («Слой → Цвет → Кривые»), то проще добиться неких фантастических искажений цвета, нежели скорректировать цвета так, чтобы изображение не утратило натуральность. Впрочем, в умелых руках и кривые будут работать как нужно.



Наглядное средство для работы с цветом изображения — «Filter Pack»



«Filter Pack» имеет готовые настройки, ускоряющие работу



Уровни можно регулировать отдельно для каждого канала



Избавляемся от красного «вампирского» взгляда

Устранение эффекта красных глаз

Кто не сталкивался в фотографии с так называемым эффектом красных глаз? При расширенных в темноте зрачках свет вспышки отражается на кровеносных сосудах глазного дна. Существуют фотоаппараты с режимом предварительной вспышки, которая подается чуть раньше основной, чтобы зрачки успели сузиться. Однако если в комнате царит полумрак, а за окном нависли облака, эта хитрость оказывается бесполезной. Особенно если человек не смотрит прямо в объектив.

Кому-то нравятся вампирские глаза людей на фотографиях, но большинство желает от такой небывалой красноты избавиться. Существуют отдельные утилиты, выполняющие эту задачу автоматически. В GIMP красные глаза уничтожаются тоже довольно просто, но вручную. Метод таков:

- ▶ Загружаем картинку.
- ▶ Выбираем инструмент эллиптического выделения.
- ▶ Выделяем им красный глаз. Впрочем, наиболее точно выделить нужную область можно с помощью альфа-канала, преобразовав его потом в выделение.
- ▶ Применяем к выделенной области растушевку: «Выделение → Растушевка» (задав необходимый радиус). Таким образом, края выделенной области утрачивают свою четкость.
- ▶ Идем в меню «Слой → Цвет → Уровни», где начинаем снижать уровень красного канала до натурального почернения зрачка. Побочный эффект — отблеск на зрачке утратит красную составляющую и станет чуть зеленоватым. Поэтому если вам важен белый отблеск, то надо будет выделить его и осветлить или обесцветить. Замечу, что для бытовых фотографий, снятых с помощью мыльницы, это не имеет значения.

Удаление царапин, обрывов и дыр

Борьба с царапинами и подобными им последствиями влияния времени может пригодиться, если вы захотите отреставрировать какую-нибудь старую фотографию. Замечу, что это вовсе не обязательно. На мой взгляд, старые изображения лучше вообще не трогать, а оставлять как есть, потому что

любая реставрация подразумевает искажение. Впрочем, вы можете иметь и другое мнение на этот счет.

Итак, как же бороться с царапинами? Нам понадобится инструмент клонирования, именуемый «Clone tool». Вызывается он нажатием клавиши «С» либо из палитры инструментов.

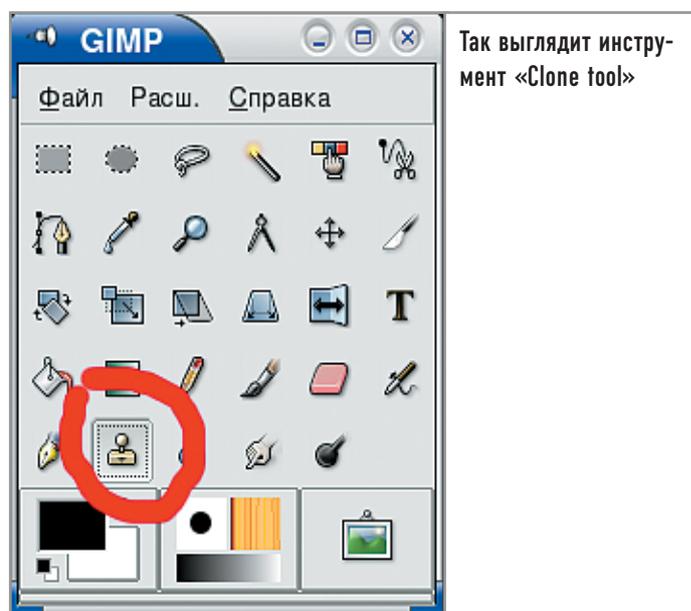
К сожалению, у меня нет под рукой старинной или даже старой фотографии, где были бы царапины, поэтому буду рассказывать словами, не иллюстрируя их картинкой.

Допустим, у нас есть какая-либо фотография, часть поверхности которой испорчена. Мы ее отсканировали и хотим что-нибудь сделать, чтобы улучшить ее вид. На фотографии непременно должны быть области, где изображение более-менее хорошо сохранилось.

Теперь пример: допустим, имеется фотография человека, стоящего на лесной поляне. Часть лица на снимке продырявлена, а по лесной чаще кто-то лет пятьдесят назад поводил циркулем или бритвой, оставив незабываемые следы. Мы же хотим от них избавиться.

Выбираем инструмент клонирования. Для начала исправим лес и кусты, на фоне которых стоит человек. Удерживая нажатой левую клавишу «Ctrl», щелкаем инструментом клонирования по той области, откуда будем брать пиксели неиспорченного изображения. Такая область обязательно найдется — не весь же лес попал под циркуль. Затем, отпустив «Ctrl», рисуем инструментом клонирования на испорченном участке кустов либо леса так же, как обычной кистью. При этом пиксели «хорошего» участка картинки переносятся и клонируются в «плохой». Между прочим, используются параметры текущей кисти (форма, размер и т. д.).

Схожим образом поступаем и с лицом человека. Сначала указываем ту область, где изображение нормальное, а затем закрасиваем этими пикселями испорченную часть снимка. Что это дает? Обычно очень трудно подобрать цвет и фактуру лица, но клонирование позволяет эту проблему обойти. Разумеется, инструмент клонирования — не панацея, и полезен он лишь тогда, когда детали не имеют значения. |



Алексей Кузьменков

С орбиты на экран десктопа

Словосочетание «DVB-карта» у многих, скорее всего, ассоциируется со спутниковым Интернетом. Однако мало кто из активных любителей дешевого трафика задумывается о том, что DVB-карту можно использовать и для просмотра спутникового телевидения.

В связи с появлением недорогого ADSL в крупных городах многие обладатели комплектов спутникового Интернета стали избавляться от них. Как оказалось, зря: DVB-карта, помимо использования по прямому назначению, может принести еще немало пользы. Кстати, тем, кто думает о приобретении ТВ-тюнера, я настоятельно рекомендую еще раз взвесить все за и против: подумайте, нужны ли вам постоянные помехи, неизбежно возникающие при приеме аналогового ТВ-сигнала? Может быть, за чуть большие деньги вы решите приобрести DVB-карту и спутниковую антенну? В цифровом спутниковом ТВ не бывает помех, а стандартным разрешением является 720x540 точек.

| Что необходимо? |

Прежде всего, давайте разберемся, что такое DVB-карта, или DVB-S. Она представляет собой обыкновенную PCI-плату с одним-единственным входом для антенного штекера. Впрочем, есть и USB-варианты, подходящие владельцам ноутбуков и barebone-систем, в которых не осталось свободного PCI-слота. Тем, кто думает о приобретении DVB-карты, я бы посоветовал

PCI-ресивер Technotrend SkyStar2. DVB-ресивер SkyStar2 — это довольно популярное решение для Интернета, однако и для просмотра спутникового телевидения он тоже отлично подойдет. Главное — не спутать его со SkyStar1, который стоит значительно дороже и, несмотря на похожее название, является, по сути, совсем другим устройством. Основное его отличие от SkyStar2 — наличие встроенного аппаратного MPEG2-декодера. Это, конечно, хорошо — нагрузка на центральный процессор во время просмотра ТВ минимальна, однако цена таких устройств значительно выше (по последним данным, она составляет примерно €215). SkyStar2 же стоит существенно дешевле — не более €70–75.

Кроме всего прочего при подключении телевизионной трансляции необходимо точно знать, на какой спутник настроена антенна, его частоту, поляризацию и символьную скорость потока.

| Установка драйверов |

Если ваша система имеет ядро версии 2.4.x, то вам придется скачать свежую версию комплекта драйверов с сайта LinuxTV

(www.linuxtv.org/download/dvb/linuxtv-dvb-1.0.1.tar.gz). Также потребуются исходные тексты используемого ядра. Его версию вы можете узнать с помощью команды `uname` с опцией `-r`:

```
# uname -r
```

2.4.28-gentoo-r1

Проследите за тем, чтобы ссылка `/lib/modules/версия_ядра/build` указывала на каталог с исходными текстами вашего ядра:

```
# ls -l /lib/modules/2.4.28-gentoo-r1/build
```

```
lrwxrwxrwx 1 root root 31 Apr 6 23:56 /lib/modules/2.4.28-gentoo-r1/build -> /usr/src/linux-2.4.28-gentoo-r6
```

Если карта имеет встроенный аппаратный MPEG2-декодер, убедитесь, что используемое вами ядро собрано с поддержкой опции «Video for Linux». Производители редко встраивают поддержку «Video for Linux» прямо в ядро, как правило, эта опция собирается отдельно в виде модуля. Посмотрите, существует ли поддиректория с именем `v4l` в каталоге `/lib/modules/версия_ядра/kernel/drivers/media`. Если же вам настолько повезло, что к DVB-карте прилагается пульт дистанционного управления, то стоит проверить, включена ли опция «Input Core Support». Еще раз оговорюсь, что это имеет смысл лишь в том случае, если на вашей карте есть MPEG2-декодер: владельцев SkyStar2 и подобных устройств это не касается.

Теперь можно начинать установку. Перейдите в режим суперпользователя, распакуйте архив с драйверами и выполните сборку:

```
# su
```

```
# cd /usr/src
```

```
# tar -zxvf ~/downloads/linuxtv-dvb-1.0.1.tar.gz
```

```
# cd linuxtv-dvb-1.0.1
```

```
# make
```

Если все прошло без ошибок, перейдите в директорию `driver` и попробуйте загрузить собранные модули:

```
# cd driver
```

```
# make insmod
```

Проверьте, загрузились ли они. Для этого выполните команду:

```
# lsmod
```

```
dvb_core          76456  1 skystar2
```

```
mt352             6276   1 skystar2
```

```
stv0299          10760  1 skystar2
```

```
nxt2002          8964   1 skystar2
```

```
mt312            7812   1 skystar2
```

```
i2c_core         17936  5
```

```
skystar2,mt352,stv0299,nxt2002,mt312
```

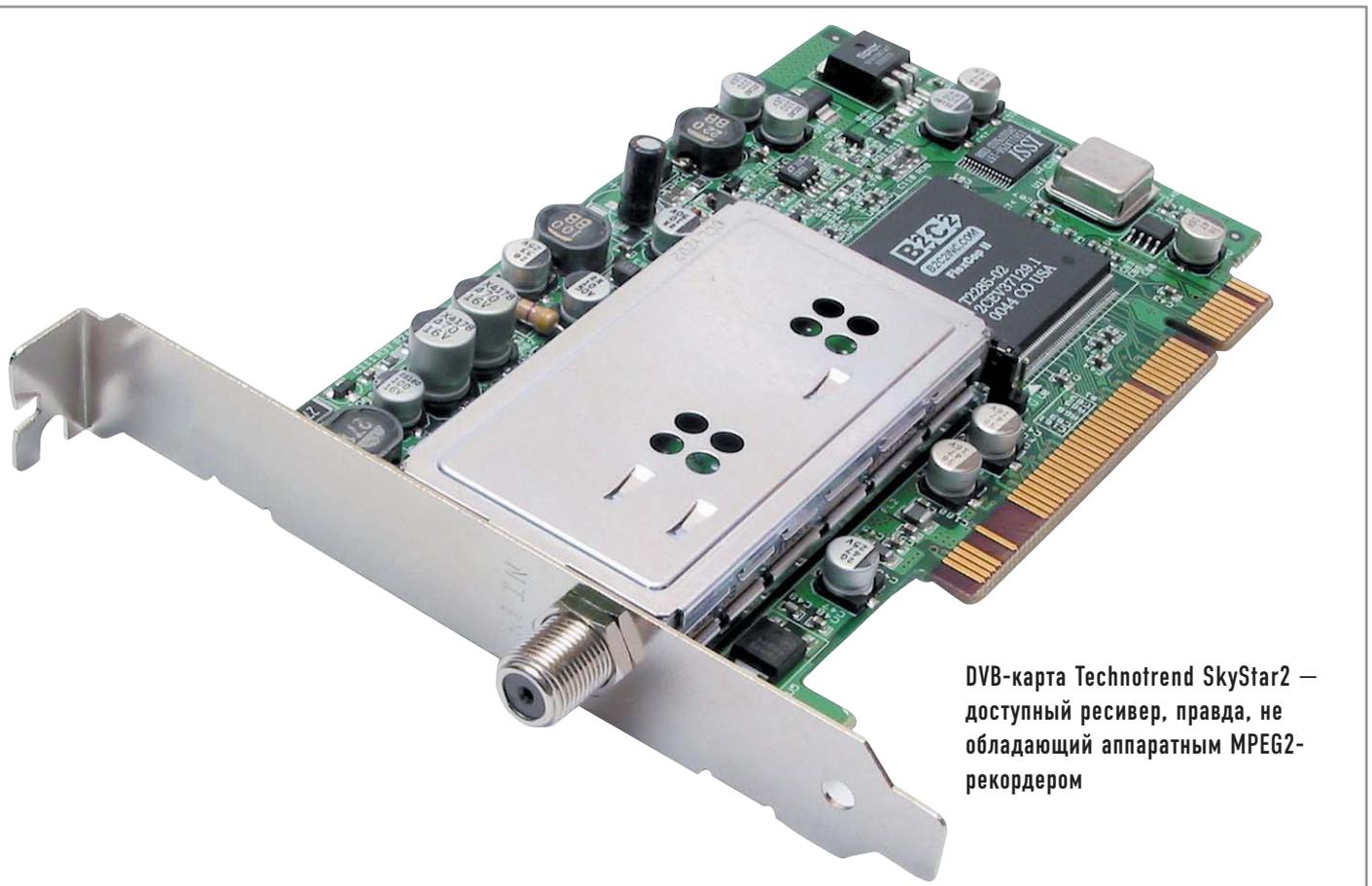
Когда вы убедитесь, что все работает, можно перенести модули в каталог `/lib/modules/версия_ядра/kernel/drivers/media/dvb`, а также скопировать тестовые утилиты в каталог `/usr/bin`:

```
# make install
```

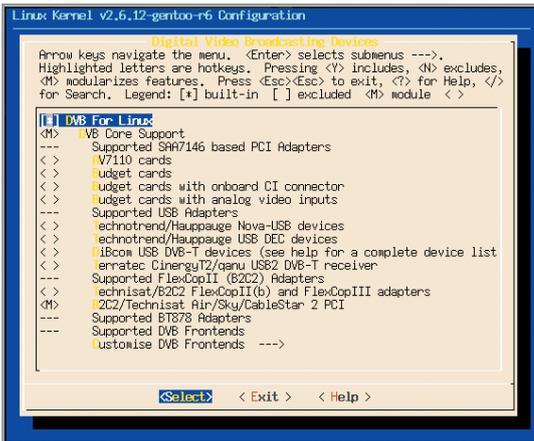
```
# cp ../apps/scan/scan /usr/bin
```

```
# cp ../apps/szap/szap /usr/bin
```

Если у вас не самый новый дистрибутив и в нем не используется система управления файлами устройств `devfs`, придется создать соответствующие файлы устройств. К счастью, вручную этого делать не нужно, так как с драйверами поставляется скрипт `makedev.nari`. Обратите внимание на то, что он требуется только в том случае, если не используется `devfs`.



DVB-карта Technotrend SkyStar2 — доступный ресивер, правда, не обладающий аппаратным MPEG2-рекордером



Включение поддержки DVB-устройств в ядре 2.6.12



Выбор фронтенда для ресиверов DVB-S

В системах, использующих devfs (можно проверить наличие файла /etc/devfsd.conf, для того чтобы определить, используется ли devfs), необходимо скопировать файл devfsd.conf, находящийся в директории driver, в каталог /etc/devfs/conf.d/dvb:

```
# mkdir /etc/devfs/conf.d/dvb
# cp devfsd.conf /etc/devfs/conf.d/dvb/
```

Обладателям ядер серии 2.6.x драйверы отдельно скачивать не надо, так как они уже интегрированы в ядро. Потребуется скачать лишь тестовые утилиты из пакета linuxtv-dvb-apps (www.linuxtv.org/download/dvb/linuxtv-dvb-apps-1.1.0.tar.bz2). Если разработчик вашего дистрибутива поленился включить в ядро поддержку DVB-устройств (проверьте директорию /lib/modules/версия_ядра/kernel/drivers/media/dvb), вам придется заняться его пересборкой. Настройки DVB-оборудования находятся в подменю конфигулятора ядра: «Device Drivers → Multimedia devices → Digital Video Broadcasting Devices». Вот тут полезно будет вспомнить, на каком чипе основан ваш DVB-приемник, так как большинство драйверов включают в себя поддержку сразу нескольких устройств. Для SkyStar2 следует отметить пункт «B2C2/Technisat Air/Sky/CableStar 2 PCI», причем желательно компилировать драйвер в виде модуля. Кроме того, необходимо включить поддержку нужного фронтенда. Для этого перейдите в меню «Customise DVB Frontends». Нас интересуют фронтенды для DVB-S: Conexant CX24110 based (stv0299), Philips TDA8083 based (stv0299), Philips TDA8044 or TDA8083 based (stv0299) и VLSI VES1893 or VES1993 based (mt312). Вообще считается, что со SkyStar2 ревизии 2.3 нужно использовать mt312, а для 2.6 — stv0299, однако на приемнике (SkyStar2 ревизии 2.3) одинаково хорошо работают все фронтенды. Их тоже желательно компилировать в виде модулей. Кроме самих драйверов обязательно включите поддержку RTC («Device Drivers → Character Devices → Enhanced Real Time Clock Support») и I2C («Device Drivers → I2C Support → I2C Support»). При сборке поддержка I2C не проверяется, однако если ее не включать, компиляция драйверов завершится неудачно. Впрочем, на ядрах версии 2.6.9 и ниже можно обойтись и без I2C — все и так нормально компилируется.

Теперь можно собрать утилиты для настройки DVB-ресивера. Но все компилировать не нужно, потребуются лишь scan и szarp:

```
# cd /usr/src
# tar -zxvf ~/downloads/linuxtv-dvb-apps-1.1.0.tar.gz
```

```
# cd linuxtv-dvb-apps-1.1.0/util/scan
# make
# cp scan /usr/bin/
# cd ../szarp
# make
# cp szarp /usr/bin/
```

Настройка устройств

Перезагрузитесь и подключите модуль, соответствующий вашей модели DVB-карты:

```
# modprobe skystar2
# lsmod
```

```
<...>
skystar2      28036    1
dvb_core     76456    1 skystar2
mt352        6276    1 skystar2
stv0299     10760    1 skystar2
nxt2002      8964    1 skystar2
mt312        7812    1 skystar2
i2c_core     17936    5
skystar2,mt352,stv0299,nxt2002,mt312
```

Также неплохо бы проверить сообщения ядра:

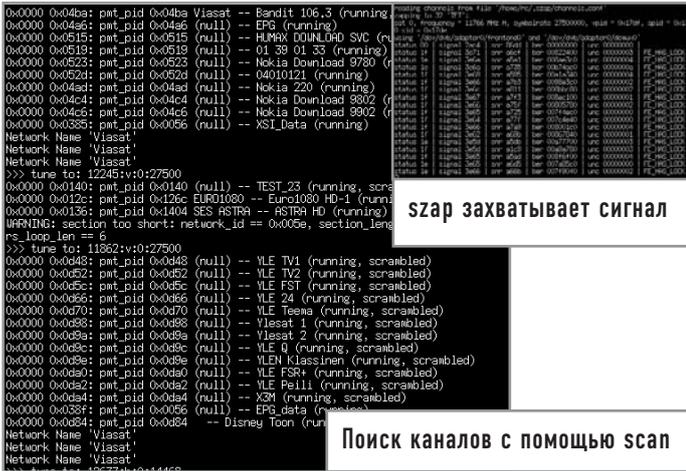
```
# dmesg | grep skystar
drivers/media/dvb/b2c2/skystar2.c: FlexCopII(rev.130)
chip found
drivers/media/dvb/b2c2/skystar2.c: the chip has 6 hardware
filters
```

Видно, что ядро признало DVB-карту. Теперь разберемся с файлами устройств. Проверьте директорию /dev/dvb/adapter0. Если она не пуста, значит все нормально. По умолчанию файлы в этой директории принадлежат пользователю root и группе video. Следовательно, для того чтобы пользователь мог работать с DVB-картой, его нужно внести в эту группу:

```
# su
# gpasswd -a username video
```

Сканирование транспондера и поиск каналов

Теперь, когда с настройкой драйверов DVB-карты покончено, можно приступать к сканированию транспондера и поиску открытых каналов. Для этого и служит утилита scan. На вход ей нужно подать специальный scan-файл с настройками спутника/



транспондера. Каждый транспондер пишется отдельной строкой, а формат файла такой: S <частота> <поляризация: горизонтальная (H) или вертикальная (V)> <скорость потока> AUTO.

В каталоге с исходными текстами утилиты scan есть поддиректория dvb-s, в которой содержатся scan-файлы для спутников Astra, Hispasat, Hotbird, Sirius, Telecom2, Telstar2, Thor и Turksat. Если вашего спутника в этом списке не оказалось, придется писать scan-файл самостоятельно на примере этих файлов. Вот так, например, выглядит написанный автором файл настроек для спутника Eutelsat-W3:

```
S 11554000 H 27500000 AUTO
S 11449000 V 27500000 AUTO
```

Сохраните созданный файл в /tmp и используйте его для сканирования утилитой scan:

```
# cd tmp
# scan Sirius > channels.conf 2> log
```

В результате получается файл channels.conf, содержащий данные всех каналов, на которые удалось настроиться, а также файл log, вмещающий «отладочную» информацию из стандартного потока ошибок. Файл channels.conf нуждается в доработке: необходимо удалить все закодированные каналы, в файле log они отмечены словом «scrambled». В идеале channels.conf должен содержать только те свободные каналы, которые вам нужны. Создайте в своем домашнем каталоге директорию /.szarp и скопируйте туда готовый channels.conf:

```
# cd ~
# make .szarp
# cp /tmp/channels.conf ~/.szarp
```

Для настройки на нужный канал используйте утилиту szarp с именем канала (szarp использует данные из ~/.szarp/channels.conf) и опцией -r:

```
# szarp TET -r
```

Что-то подобное изображенному на скриншоте вы должны наблюдать и у себя. FE_HAS_LOC и status If говорят о том, что DVB-приемник успешно настроился на нужный канал и уровень сигнала достаточно высок для комфортного просмотра ТВ. Чем ниже уровень сигнала, тем больше непринятых пакетов (unc). В идеале значение unc не должно превышать 5–10. Пока вы не завершите выполнение szarp комбинацией «Ctrl+C», ресивер будет принимать сигнал. Как только вы прервете выполнение szarp, фронтенд сразу же перейдет в спящий режим.

Такие условия работы предпочтительны, так как DVB-карта работает лишь тогда, когда это действительно необходимо. Если вы все-таки хотите, чтобы ресивер работал всегда, то загрузите модуль dvb-core с опцией dvb_shutdown_timeout=0, а szarp — с опцией -x, чтобы заставить его завершить работу, как только настройка на нужный канал будет выполнена.

Программы для просмотра

Если ваш DVB-приемник оборудован аппаратным MPEG2-декодером, то для просмотра можно использовать любую программу, работающую с ТВ-тюнерами, например XawTV. Учтите, что переключение каналов напрямую из XawTV работать не будет, ведь новый канал — это новая частота и поляризация (а иногда еще и новая скорость потока). Так что для переключения каналов придется пользоваться szarp, что очень неудобно.

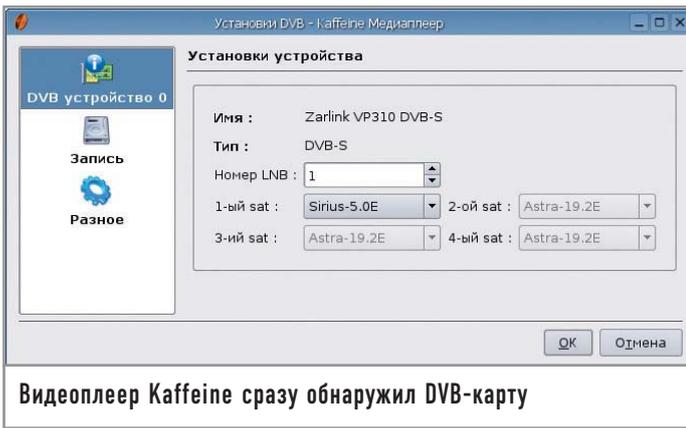
Владельцам SkyStar2 и подобных карт, не оснащенных MPEG2-декодером, XawTV уже не подойдет, ведь он не умеет работать со сжатым видеопотоком. В первую очередь следует попробовать mplayer: с недавних пор он умеет нормально декодировать потоки MPEG2-TS (transport stream). Mplayer поддерживает функцию «Timeshifting», то есть при просмотре спутникового канала вы можете нажать на паузу и отойти, чтобы налить себе кофе, а затем прийти и продолжить просмотр с того места, где вы прервались. А все то время, пока вы будете на кухне, mplayer будет принимать видеопоток и записывать его в файл — главное, не слишком увлечься: винчестер-то, наверное, у вас не резиновый. Для того чтобы пользоваться «Timeshifting», нужен mplayer версии не ниже 1.0-pre7 — в более ранних версиях эта функция работает плохо: после второго «похода на кухню» mplayer обычно завершает работу.

Предпочтительнее собрать mplayer из исходных текстов. Самую новую можно получить из cvs (подробнее об этом — на сайте www.mplayer.hu):

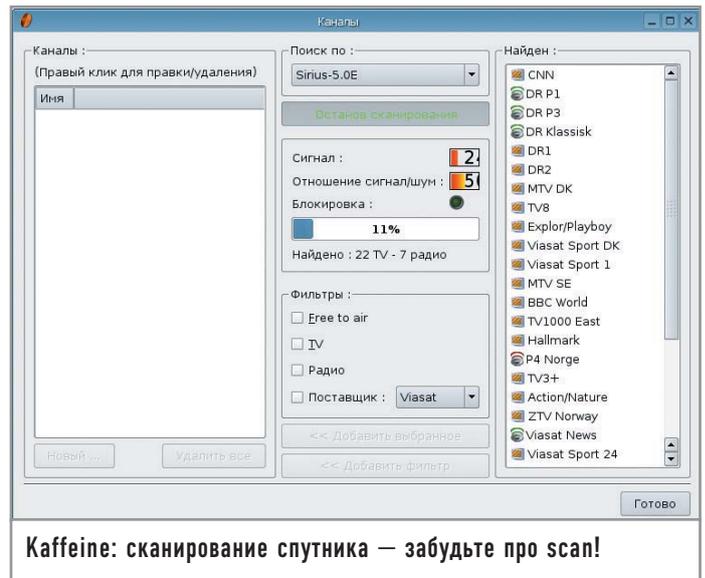
```
# ./configure --with-gui --with-dvbinclir=/usr/src/linuxtv-
dvb/include
# make
# make install
```

Для того чтобы mplayer смог принимать сигнал, запустите в одном терминале szarp, и, когда получите сообщения





Видеоплеер Kaffeine сразу обнаружил DVB-карту



Kaffeine: сканирование спутника — забудьте про scan!

«FE_HAS_LOC», из другого терминала — mplayer:

```
# mplayer — </dev/dvb/adapтер0/dvr0
```

Согласитесь, что это тоже не очень удобно. Впрочем, автор для себя написал несколько небольших скриптов (по одному на каждый интересный мне канал) приблизительно следующего содержания:

```
# !/bin/sh
```

```
szap TET -r&
```

```
sleep 2
```

```
mplayer — </dev/dvb/adapтер0/dvr0
```

```
killall szap
```

После настройки на нужный канал делается пауза в две секунды, которой достаточно для того, чтобы szap успел захватить сигнал. Затем запускается mplayer (все это время szap работает в фоновом режиме). Как только mplayer завершает свою работу, процесс szap убивается. Для таких скриптов можно создать отдельное подменю, что и сделано у меня во fluxbox.

И все равно не очень удобно — если вы хотите переключать каналы, нужно постоянно закрывать mplayer и обращаться к меню, чтобы открыть другой канал.

Гораздо проще просмотр спутникового ТВ организован в xine — он, как оказалось, теперь тоже поддерживает декодирование MPEG2-TS. Для того чтобы настроить xine на прием спутникового ТВ-сигнала, требуется всего лишь скопировать имеющийся файл channels.conf в поддиректорию .xine, расположенную в домашнем каталоге:

```
# cp ~/.szap/channels.conf ~/.xine/
```

Теперь попробуйте запустить xine и щелкнуть по кнопке «DVB». Переключаться между каналами можно с помощью клавиш «PageUp» и «PageDown», и при этом не нужно никакого szap: xine сам прочитает channel.conf и сам же захватит сигнал. Удобно, не правда ли?

У пользователей KDE есть отличная альтернатива — универсальный видеоплеер Kaffeine, в который не так давно была добавлена поддержка DVB-видео. Ранее существовал отдельный DVB-плеер для KDE, который назывался KaXTV, но теперь на сайте разработчиков KaXTV появилось объявление, что разработка этого продукта прекращена, однако все ее функции уже имеются в Kaffeine. Признаться, Kaffeine сразу поразил меня своей догадливостью — во время первого запуска он обнаружил DVB-карту и отобразил диалоговое окно, в котором попросил указать, с какого спутника я собираюсь

смотреть программы. Kaffeine умеет использовать движок xine, поэтому я надеялся, что и тут можно будет обойтись без szap. Каково же было мое удивление, когда я обнаружил, что кроме всего прочего Kaffeine имеет встроенную функцию сканирования транспондера! То есть можно обойтись не только без szap, но и без scan, а значит, и без нудной операции выборки открытых каналов — нужно всего лишь просканировать нужный спутник, поставить фильтр «Free to air» и выбрать пункт «Добавить фильтр». Теперь все открытые каналы у вас в списке. А еще можно добавить только радио, только ТВ или только каналы определенного провайдера. Kaffeine может записывать поток прямо в файл (не слишком увлекайтесь записью, если у вас не очень большой жесткий диск).

Ради интереса я решил попробовать MythTV — амбициозную новинку, которую так любят хвалить обладатели ТВ-тюнеров (она тоже поддерживает работу с DVB-ресиверами). MythTV требует для своей работы MySQL, и перед первым запуском вам нужно будет запустить MySQL-сервер, создать новую базу для MythTV, а после этого использовать mythtvsetup. К сожалению, ничего хорошего об этой программе я сказать не могу — при попытке настройки MythTV все время вылетала, выдавая сообщение об ошибке нарушения сегментации (segmentation fault).

Думаю, и так видно, что Kaffeine на сегодняшний день выглядит явно лучше остальных — оно и понятно, не зря разработчики KaXTV старались (их труды не пропали даром).

Кстати, на многих спутниках уже появились HDTV-каналы: телевидение высокого разрешения владельцам DVB-карт доступно уже сегодня! Правда, без мощного процессора и хорошей видеоплаты тут не обойтись: представьте себе, нужно декодировать MPEG2-видеопоток с разрешением 1988x1088 в реальном времени. Для комфортного просмотра HDTV вам понадобится процессор как минимум уровня Athlon 2500+. Впрочем, сейчас уже трудно найти новый процессор с частотой ниже 2 ГГц, и я думаю, что, как только HDTV получит достаточное распространение, возможность его просмотра появится у каждого пользователя. Ведь гораздо проще и дешевле приобрести DVB-тюнер, чем HDTV-телевизор. |

Обаяние минимализма

Отечественным пользователям хорошо знакомы такие дистрибутивы как ASP, ALT, Fedora и Mandriva. Однако загляните на сайт www.distrowatch.com и ради интереса ознакомьтесь с десятком самых популярных продуктов — возможно, вы удивитесь, но ни ASP, ни ALT там нет. Зато в «горячей десятке» плотно засел некий Gentoo Linux и, надо сказать, не покидает ее вот уже два года.

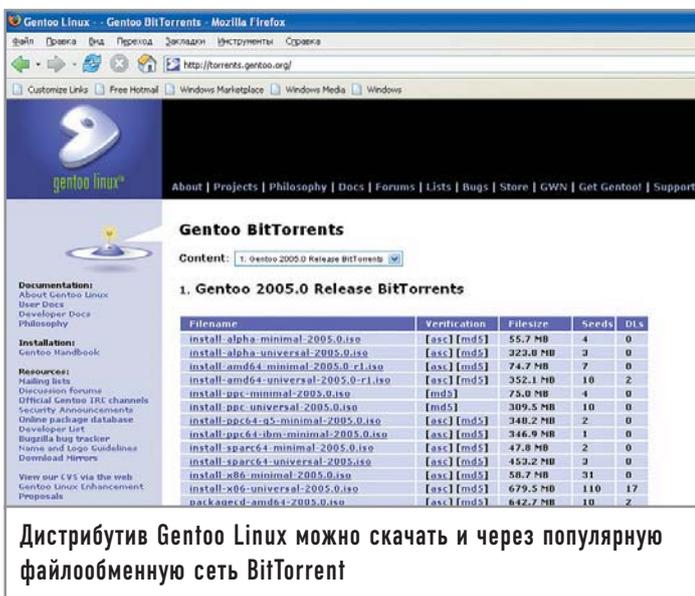
Проект Gentoo Linux был начат Даниэлем Роббинсом, ранее работавшим над Stampede Linux и FreeBSD. Ему никак не давал покоя тот факт, что механизм портов, так успешно применяемый в BSD-системах, не нашел своего отражения в мире Linux. И вот в 2002 году на свет появилась первая версия Gentoo Linux. С тех пор много воды утекло, Gentoo претерпел значительные изменения, а Даниэль ушел работать в Microsoft, однако развитие Gentoo продолжается, и с каждым днем этот дистрибутив находит все новых приверженцев.

Самое главное преимущество Gentoo, механизм портежей (portage), является своего рода наследником системы портов,

которую можно наблюдать, например, во FreeBSD/NetBSD. Portage — обновляющийся и дополняющийся каждый день он-лайн репозиторий приложений, а точнее их исходных текстов и соответствующих патчей и исправлений. Работая в Gentoo, вам совершенно необязательно ждать новых версий дистрибутива, так как при помощи портежей вы можете обновить дистрибутив и программы до самых последних версий. Причем делается это при помощи всего одной команды — emerge.

Вот что происходит после того, как пользователь вводит команду emerge program: сначала скрипт emerge проверяет, есть ли описание сборки (так называемый файл ebuild) этого приложения в локальном снимке дерева портежей, затем идет проверка зависимостей. При этом если не все зависимости были удовлетворены, то необходимые библиотеки и приложения автоматически добавляются в список установки, причем в нужном порядке. То есть, если приложению program для нормальной работы необходима библиотека libprog, она будет установлена перед сборкой program — то есть так, как требуют зависимости. Далее механизм портежей проверяет наличие исходных текстов linprog и program в директории /usr/portage/distfiles, а если они там не обнаружены, то самостоятельно скачивает их с нужных FTP-серверов. После этого emerge приступает непосредственно к сборке и установке в систему необходимого приложения.

Конечно же, сборка осуществляется с применением флагов оптимизации, указанных в файле /etc/make.conf. Стоит упомянуть, что все программы, находящиеся в дереве portage, тщательно проверяются, и, если какая-нибудь программа при компиляции с помощью какой-либо версии gcc будет, например, выдавать ошибку, к ней изготавливается патч, который накла-



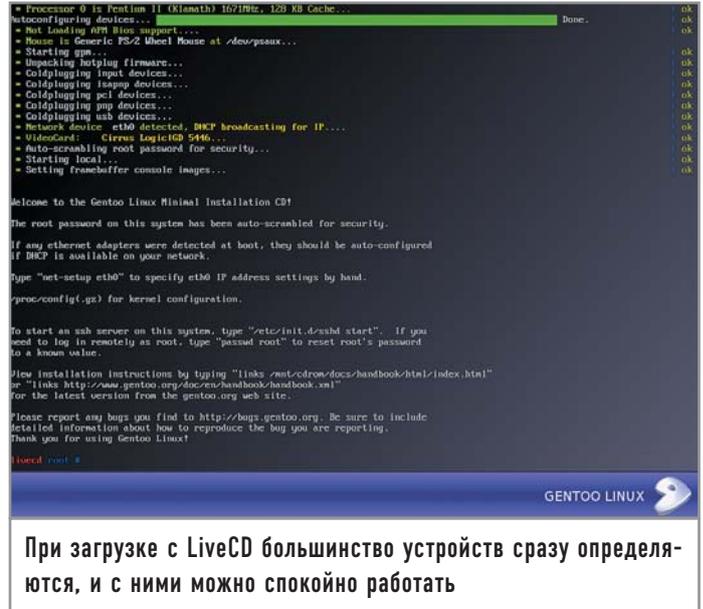
Gentoo Linux · Gentoo BitTorrents · Mozilla Firefox

Content: 1. Gentoo 2005.0 Release BitTorrents

1. **Gentoo 2005.0 Release BitTorrents**

Filename	Verification	Filesize	Seeds	DLs
install-alpha-minimal-2005.0.iso	[asc] [md5]	55.7 MB	4	0
install-alpha-universal-2005.0.iso	[asc] [md5]	323.0 MB	3	0
install-amd64-minimal-2005.0-r1.iso	[asc] [md5]	74.7 MB	7	0
install-amd64-universal-2005.0-r1.iso	[asc] [md5]	352.1 MB	10	2
install-ppc-minimal-2005.0.iso	[md5]	75.0 MB	4	0
install-ppc-universal-2005.0.iso	[md5]	309.5 MB	10	0
install-ppc64-ppc-minimal-2005.0.iso	[asc] [md5]	348.2 MB	2	0
install-ppc64-ibm-minimal-2005.0.iso	[asc] [md5]	346.9 MB	1	0
install-sparc64-minimal-2005.0.iso	[asc] [md5]	47.8 MB	2	0
install-sparc64-universal-2005.0.iso	[asc] [md5]	452.2 MB	3	0
install-x86-minimal-2005.0.iso	[asc] [md5]	58.7 MB	31	0
install-x86-universal-2005.0.iso	[asc] [md5]	679.5 MB	110	17
package-d-amd64-2005.0.iso	[asc] [md5]	842.7 MB	10	2

Дистрибутив Gentoo Linux можно скачать и через популярную файлообменную сеть BitTorrent



дывается на исходные тексты перед компиляцией. Таким образом, вы можете быть уверены, что все находящиеся в портелях программы работоспособны.

Надо сразу сказать, что Gentoo создан с прицелом на довольно опытного пользователя, который не поленится прочесть документацию и, ко всему прочему, обладает достаточным количеством свободного времени. Основной способ расширения функциональности системы — это сборка пакетов из исходных текстов. Да и сама система может быть полностью собрана из исходников — кстати, этот метод установки и является рекомендуемым. Возможно, при упоминании ручной сборки чего бы то ни было на читателя нахлынут воспоминания о том, как ему не удалось скомпилировать какую-нибудь программу, как компилятор почему-то выдавал кучу непонятных ошибок... Не стоит беспокоиться, в Gentoo Linux процесс сборки автоматизирован, но при этом у пользователя остается возможность держать его под своим контролем.

Итак, если у вас пылливый ум, крепкие нервы и достаточно времени — добро пожаловать в Gentoo Linux.

Пакеты или исходные коды?

Для начала вам необходимо сделать выбор — устанавливать базовую систему из уже скомпилированного пакета (stage3) либо целиком и полностью собирать из исходных кодов (stage1). В первом случае вам необходим официальный двухдисковый комплект Gentoo, во втором — сборка Gentoo LinuxCenter Edition или Gentoo Xor's. Предполагая, что читатель пока только начинает осваивать Gentoo Linux, мы бы посоветовали официальный комплект и, соответственно, установку из перекомпилированных пакетов. Обратите внимание, что официальный комплект существует в нескольких вариантах для различных архитектур: x86 (запускается на всем, что мощнее i368), i686 (требуется как минимум Pentium PRO), Pentium 4, Athlon XP, Athlon 64. Конечно же, следует выбирать, рассчитывая на то, какое железо имеется в наличии.

С комплектом Хог или LinuxCenter все проще — на диске пакетов почти нет, а исходные коды можно собрать на любой

платформе. Если вы предпочли именно этот вариант, то будьте готовы к тому, что на компиляцию уйдет гораздо больше времени, чем на установку из пакетов. Далее мы будем говорить об инсталляции Gentoo версии 2005.0.

Впечатляющее начало

Первое, на что стоит обратить внимание, — процесс установки системы. Как таковой инсталляционной программы в Gentoo Linux нет, а значит, вам никто не предложит откинуться на спинку кресла и узнать, что новая версия ОС стала еще проще и надежнее. Нет и скрипта, который шаг за шагом вел бы пользователя по незнакомым пока местам. Вместо всего этого после загрузки установочного Gentoo LiveCD перед вами предстанет пустая консоль в облике командного интерпретатора bash. Понятно, что это может вызвать шок даже у достаточно опытного пользователя. Но со временем вы и сами убедитесь, что отсутствие инсталляционной программы в Gentoo Linux скорее благо, нежели зло. Дело в том, что установка Gentoo Linux не всегда сводится к одним и тем же операциям, и нет такой инсталляционной программы (и вряд ли ее когда-нибудь напишут), которая по своей гибкости могла бы сравниться с обычной командной строкой. К сожалению, отсутствие красивого графического инсталлятора оказывается непреодолимой преградой для неопытного или просто нетерпеливого пользователя, которому не хочется читать документацию и понимать смысл такого подхода.

Между тем во время достаточно продолжительной установки с Gentoo LiveCD вам доступны шесть виртуальных консолей: на первой обычно проходит сам процесс инсталляции, а вот остальные совершенно свободны (между ними можно переключаться комбинациями клавиш «Alt+F1/2/3/4/5/6»), и вы можете использовать их для ознакомления с документацией, чтения романа «Война и мир» или интернет-серфинга при помощи консольного веб-браузера links.

Надо заметить, что большинство оборудования, такого как модемы, сетевые и звуковые платы, определяется в момент загрузки LiveCD и становится сразу же доступным.

Дисковые операции

Что следует сделать в первую очередь? Конечно же, правильно разметить жесткий диск. В Gentoo LiveCD для этого имеются две программы: fdisk и его более продвинутый аналог — cfdisk. Советуем воспользоваться cfdisk, так как она имеет псевдографическое подобие пользовательского интерфейса. Вам необходимо создать как минимум два раздела — корневой и swap. Размер swap-раздела в общем случае чуть больше удвоенного размера оперативной памяти. Размер корневого раздела должен быть не меньше 3 Гбайт. Swap желательно активировать сразу же после разметки диска, но для этого раздел должен быть отформатирован:

```
# mkswap /dev/hda1 (форматирование swap-раздела)
# swapon /dev/hda1 (теперь swap активирован)
```

Затем необходимо подготовить корневой раздел, на который и будет устанавливаться система. Но перед этим нужно выбрать тип файловой системы, где предстоит «жить» Gentoo. Целесообразно использовать Ext3 или ReiserFS, причем второй вариант предпочтительнее, но в принципе существенной разницы в этом нет:

```
# mkreiserfs /dev/hda2 (форматирование корневого раздела
и создание файловой системы ReiserFS)
# mkfs.ext3 /dev/hda2 (форматирование корневого раздела
и создание файловой системы Ext3)
```

Теперь, для того чтобы закончить подготовку, необходимо смонтировать корневой раздел в директорию /mnt/gentoo:

```
# mount /dev/hda2 /mnt/gentoo
```

Возможные варианты

На этом подготовительный этап закончен, и впереди вас ждет самое сложное — установка. Если у вас официальный двухдисковый комплект и вы планируете установить систему из пакетов, то нужно распаковать архив stage3 (он находится в директории /mnt/cdrom/stages) в корневой раздел. Вообще, на официальном установочном LiveCD есть несколько вариантов stage3 — выбирайте тот, который соответствует вашей модели процессора:

```
# cd /mnt/gentoo (переход в будущий корневой
раздел Gentoo)
```

```
# tar xjvpf /mnt/cdrom/stages/stage3-athlon-xp-
2005.0.tar.bz2 (распаковка базовой системы для процес-
соров Athlon XP)
```

Для того чтобы получить возможность устанавливать программы, нужно распаковать снимок дерева портежей, также имеющийся на установочном диске:

```
# tar xjvpf /mnt/cdrom/snapshots/portage-2005xxxx.tar.bz2 -C
/mnt/gentoo/usr/
```

Базовая система — это фундамент, на котором будет строиться ваш Gentoo. В случае со stage3 этот фундамент уже есть, однако если вы решили установить систему путем полной сборки из исходных кодов, то все значительно сложнее. Вместо stage3 вам нужно распаковывать архив stage1:

```
# cd /mnt/gentoo
# tar xjvpf /mnt/cdrom/stages/stage1-x86-2005.0.tar.bz2
# tar xjvpf /mnt/cdrom/snapshots/portage-2005xxxx.tar.bz2 -C
/mnt/gentoo/usr/
```

В отличие от stage3, stage1 — это только заготовка, из которой еще предстоит собрать базовую систему. Обратите внимание, что этот вариант подходит только в том случае, если у вас Gentoo Xor's или LinuxCenter Edition. В противном случае вам придется докачивать исходные тексты из Интернета (а их довольно много). Для установки Gentoo из Сети разумнее обратиться к официальному руководству по инсталляции.

Итак, на установочном диске имеется некоторое количество архивов с исходными текстами, и для того чтобы система смогла использовать их, архивы нужно скопировать в директорию /usr/portage/distfiles:

```
# cd /mnt/gentoo/usr/portage
# mkdir distfiles (директория distfiles отсутствует, ее нужно
создать)
# cp /mnt/cdrom/distfiles/* /mnt/gentoo/usr/portage/distfiles
(копирование архивов с исходными текстами)
```

Последний шаг (копирование distfiles) необходим и в том случае, если у вас официальный установочный LiveCD — на нем тоже имеется некоторое количество архивов. После этого нужно выполнить смену корневого раздела:

```
# chroot /mnt/gentoo (теперь система считает, что корневым
разделом является /mnt/gentoo)
```

```

cfdisk 2.12i

Disk Drive: /dev/hda
Size: 80026361856 bytes, 80.0 GB
Heads: 255 Sectors per Track: 63 Cylinders: 8729

Name  Flags  Part Type  FS Type  [Label]  Size (MB)
-----
hda1  Primary Linux swap / Solaris  551.10
hda2  Primary Linux ReiserFS  9927.92
      Unusable
      0.01*
hda3  Primary Linux ReiserFS  60521.61*
hda4  Primary Linux ReiserFS  9023.14

[bootable] [ Delete ] [ Help ] [Maximize] [ Print ]
[ Quit ]   [ Type ]  [ Units ] [ Urite ]
Illegal command
Toggle bootable flag of the current partition

```

Разметка диска с помощью программы cfdisk для опытного пользователя не составит особого труда

```

myrzilla rc # emerge -pv mythtv

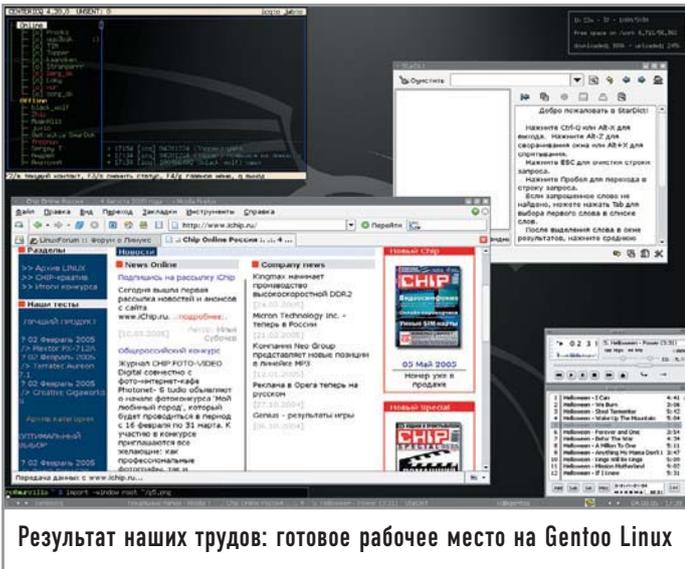
These are the packages that I would merge, in order:

Calculating dependencies ...done!
[ebuild N ] media-tv/linuxtv-dvb-1.0.1 0 kB
[ebuild N ] media-libs/libdwb-0.5.0-r1 249 kB
[ebuild N ] dev-db/mysql-4.0.24 +berkdb -debug -innodb +perl +readli
ne (-selinux) +ssl -static +tcpd 16,488 kB
[ebuild N ] media-tv/mythtv-0.18.1-r1 +alsa (-altivec) -arts -debug
+dwb -fontendonly =jack -joystick -led -lirc +mxi -nvidia +oggvorbis +op
engl +oss -unichrome 0 kB
[ebuild N ] perl-core/Storable-2.13 0 kB
[ebuild N ] dev-perl/Net-Daemon-0.38 0 kB
[ebuild N ] dev-perl/PIRPC-0.2016-r1 0 kB
[ebuild N ] dev-perl/DBI-1.46 363 kB
[ebuild N ] dev-perl/DBD-mysql-2.9003 0 kB

Total size of downloads: 17,102 kB
myrzilla rc #

```

При работе система portage показывает всю необходимую информацию о недостающих и установленных пакетах



Индивидуальный подход

Для тех, кто устанавливает систему из stage1, все только начинается. Чтобы собрать базовую систему с оптимизацией под имеющийся процессор, необходимо сначала отредактировать файл /etc/make.conf:

```
# nano -w /etc/make.conf
CFLAGS="-O2 -march=athlon-xp -pipe -fomit-frame-pointer"
CHOST="i686-pc-linux-gnu"
CXXFLAGS="${CFLAGS}"
MAKEOPTS="-j2"
```

CFLAGS — это флаги, которые будут использоваться компилятором C (gcc), CXXFLAGS — компилятором C++ (g++); march указывает оптимизацию под конкретный процессор (в нашем примере это Athlon XP). Вот флаги для других моделей процессоров: i386, i486, i586, i686, pentium, pentium-mmx, pentiumpro, pentium2, pentium3, pentium4, prescott, nocona, k6, k6-2, k6-3, athlon, athlon-tbird, athlon-4, athlon-xp, athlon-mp, winchip-sb, winchip2, c3. Думаю, в комментариях они не нуждаются — каждый узнает свой флаг по названию.

Опция -O2 задает средний уровень оптимизации (-O3 — высший), и менять его, по крайней мере на момент установки, категорически не рекомендуется.

Редактирование файла make.conf — возможно, самая сложная операция при установке из stage1. Теперь можно приступать к сборке:

```
# cd /usr/portage
# scripts/bootstrap.sh && emerge system
```

Надо сказать, что процесс этот занимает довольно длительный промежуток времени (несколько часов на современной машине), так что можно смело выключить монитор и заняться более важными делами.

Ядерные манипуляции

Но базовая система — это только полдела, ведь для нормальной работы Gentoo нужно установить ядро и загрузчик. На всех дисках по умолчанию имеется ядро gentoo (gentoo-sources). Чтобы установить его, воспользуйтесь портежами:

```
# emerge gentoo-sources
```

Как видите, ничего сложного в установке программ из портежей нет. О преимуществах этого подхода мы поговорим в конце статьи, а пока нужно скомпилировать ядро. Если вы ни разу этого не делали, рекомендуем воспользоваться утилитой genkernel — она соберет ядро, включив поддержку всех устройств в виде модулей:

```
# emerge genkernel (установка genkernel)
# genkernel all (сборка ядра в автоматическом режиме)
```

После этой операции в каталоге /boot появится образ свежеобработанного ядра — файл vmlinuz. Теперь нужно отредактировать файл /etc/fstab, для того чтобы система знала, какие разделы диска и какие файловые системы вы используете:

```
# nano -w /etc/fstab
/dev/hda2 / reiserfs noatime,exec 0 0
/dev/hda1 none swap sw 0 0
proc /proc proc defaults 0 0
none /dev/shm tmpfs defaults 0 0
```

Ну и наконец, для того чтобы система могла загрузиться, нужно установить загрузчик. В Gentoo включены как LILO, так и GRUB, и выбор тут зависит от вас. Мы продемонстрируем установку загрузчика на примере LILO:

```
# emerge lilo
# nano -w /etc/lilo.conf
<...>
image = /boot/vmlinuz (файл ядра Linux)
root = /dev/hda2 (корневой раздел)
label = Gentoo (имя пункта меню lilo)
# lilo (установка lilo в mbr)
```

Все, теперь можете смело перезагружаться в вашу новую систему Gentoo!

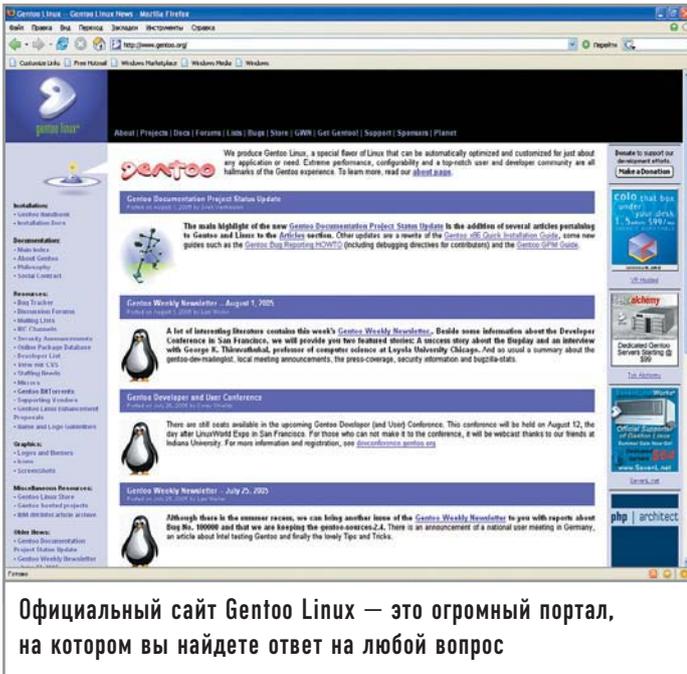
Хочется отметить, что приведенные выше краткие рекомендации не отменяют ознакомление с официальным руководством по установке. Мы преследовали цель показать новичкам, что их будет ожидать, если они захотят попробовать свои силы в Gentoo Linux.

Эффективная работа с portage

Закончив с инсталляцией, давайте посмотрим, как можно доукомплектовать систему, работая с механизмом portage. На первый взгляд все довольно просто: вы вводите команду emerge gnome, и скрипт emerge автоматически скачивает за вас из Интернета исходные тексты GNOME и всех зависимостей, компилирует их в нужном порядке и устанавливает. Однако ни о каком контроле сборки в этом случае не может быть и речи. Покажем на примере, как эффективнее всего пользоваться системой портежей:

```
# emerge -pv fluxbox
Calculating dependencies ...done!
[ebuild N ] media-libs/imlib-1.9.14-r3 0 kB
[ebuild N ] x11-wm/fluxbox-0.9.13-r1 -disablexmb -gnome +imlib -kde +nls +truetype -xinerama 1240 kB
Total size of downloads: 1240 kB
```

Опция -pv дает нам возможность посмотреть, от каких программ и библиотек зависит нужное приложение и сколько трафика придется потратить на скачивание необходи-



мых компонентов. Кроме того, показаны возможные опции сборки. Если emerge обнаруживает в каталоге distfiles нужный архив, то вместо размера архива показывает «0 kB». Из вывода emerge видно, что он обнаружил imlib, а вот оконный менеджер fluxbox придется скачивать. Для того чтобы загрузить все необходимое для установки fluxbox, нужно воспользоваться ключом -f:

```
# emerge -f fluxbox
```

Скрипт скачает исходные коды fluxbox и его зависимостей, если не обнаружит их в директории distfiles.

Теперь давайте коснемся опций сборки. Они отображаются сразу после номера версии пакета. В частности, fluxbox поддерживает следующие опции: disablexmb, gnome, imlib, kde, nls, truetype, xinerama. Например, при сборке с включенной опцией xinerama fluxbox будет возможна работа с двумя мониторами, а при включении kde/gnome появится поддержка трех для Gnome/KDE-приложений. Если перед названием опции стоит знак «+» — значит опция включена, если «-» — отключена. Опции сборки можно изменять при помощи переменной USE:

```
# USE="gnome kde -nls" emerge -pv fluxbox
Calculating dependencies ...done!
[ebuild N] media-libs/imlib-1.9.14-r3 0 kB
[ebuild N] x11-wm/fluxbox-0.9.13-r1 -disablexmb +gnome
+imlib +kde -nls +truetype -xinerama 0 kB
Total size of downloads: 0 kB
```

Если вы заметили, что часто используете какую-либо переменную, то проще всего включить ее в make.conf, добавив туда строку следующего вида:

```
USE="-qt -kde +gtk +gnome"
```

Теперь все программы, у которых есть такие опции, по умолчанию будут собираться в указанном варианте. Обратите внимание, что включение какой-либо опции при сборке программы, возможно, повлечет за собой установку «лишних» зависимостей.

Для того чтобы обновить нужную программу, воспользуйтесь опцией -u, например:

```
# emerge -u fluxbox
```

Таким же образом можно обновить и всю систему:

```
# emerge -u system
```

Информацию emerge берет из дерева портежей, так что, если у вас сохранен «снимок» дерева за прошлый год, не стоит удивляться тому, что версии программ будут слишком старыми. Для обновления дерева выполните:

```
# emerge --sync
```

Экономим трафик

Наверняка проблема чрезмерного расхода трафика для вас важна, если вы поддерживаете систему в актуальном состоянии, да еще и по модемному соединению. Если у вас есть архив fluxbox-0.9.9, то вам не обязательно скачивать полностью fluxbox-0.9.13, можно обойтись и малой кровью. Для этого предназначен инструмент getdelta.

```
# emerge getdelta (установка getdelta)
```

```
# echo FETCHCOMMAND="/usr/bin/getdelta.sh \${URI}"
\${URI}" >> /etc/make.conf
```

Вторая команда добавляет в файл /etc/make.conf строку FETCHCOMMAND="/usr/bin/getdelta.sh \\${URI}", чтобы emerge использовал getdelta (вместо wget) для скачивания исходных кодов; getdelta проверяет, присутствуют ли предыдущие версии нужного архива и, если они имеются, запрашивает с сервера только патч. Экономия трафика поражает, особенно при обновлении тяжеловесных программ, таких как OpenOffice, Mozilla и т. п. Таким же образом можно обновлять и дерево портежей, но для этого предназначен другой скрипт — emerge-delta-websync.

Приз за терпение

Читателя, дошедшего до этого места, возможно, интересует простой вопрос: «А зачем вообще это нужно, если можно взять любой другой дистрибутив и установить его за 20 минут?» Во-первых, собирая программы из исходных текстов с применением флагов оптимизации под ваш процессор, вы получаете увеличение производительности. Наиболее заметный прирост (до 30%) можно получить на процессорах Celeron, Duron и Athlon (не XP); на более современных конфигурациях это не так хорошо заметно. Однако одним из главных преимуществ Gentoo является гибкость. Простейший пример — установка популярного консольного файлового менеджера Midnight Commander (MC), который может быть скомпилирован с поддержкой Samba (сетевое окружение Windows) и X11 либо же без них. В первом случае кроме самого MC понадобятся также пакеты Samba и X11, которые тоже будут собраны из исходных текстов. В других дистрибутивах файловый менеджер уже будет скомпилирован разработчиками, причем опции его сборки останутся для вас тайной, пока вы не установите и не начнете использовать его.

В заключение можно сказать, что Gentoo дает невероятную гибкость, но при этом потребует от вас значительного времени на изучение всех тонкостей. |

Сергей Супрунов

Соискатель лавров Windows

Дистрибутив Linspire (в девичестве — Lindows) был задуман разработчиками как полноценная альтернатива операционной системе Windows на рабочем столе пользователя. Посмотрим, насколько им удалась эта затея.

| Введение |

Linspire — это Linux-дистрибутив, основанный на Debian. При его разработке ставилась задача создать максимально простую и удобную в работе для неподготовленного пользователя операционную систему и при этом сохранить все или по крайней мере основные преимущества Linux.

Для тестирования был выбран дистрибутив Linspire Five-0, наиболее свежий на данный момент. Поставляется он на одном диске, причем он же может выступать и в качестве LiveCD, позволяющего сначала посмотреть на систему, прежде чем устанавливать ее на жесткий диск.

| Установка |

Установив загрузочный диск в лоток CD-привода и перезагрузив машину, мы попадаем в меню загрузчика, предлагающего на выбор три пункта: установка или обновление системы на диске, запуск в режиме LiveCD и пункт «Advanced», который предоставляет возможность провести тестирование железа.

Выбираем установку и после нескольких минут подготовительной загрузки попадаем в диалог установки. Вопросы достаточно просты, хотя и на английском языке (можно выбрать несколько других — немецкий, португальский и т. д., но вот русского среди них пока что нет).

Установка максимально упрощена и не заставляет пользователя раздумывать над сложными вопросами даже там, где они не помешали бы. Вы можете выбрать установку системы на весь диск (как это обычно происходит при установке Windows) либо указать на нем конкретный раздел, который придется за-

ранее подготовить другими средствами, поскольку Linspire не предлагает никаких инструментов по переразбиению жесткого диска. В принципе это совершенно разумно, если не забывать, на какую целевую аудиторию рассчитан дистрибутив.

Здесь же, в расширенной установке, можно указать и тип файловой системы для будущего раздела. Правда, выбор невелик: предлагаются две версии ReiserFS — третья и четвертая (последняя находится в статусе экспериментальной).

Непосредственно перед началом копирования файлов (которое, к слову, занимает около семи минут) вам нужно будет придумать имя для своего компьютера, а также ввести пароль суперпользователя root (здесь он именуется «администратором»).

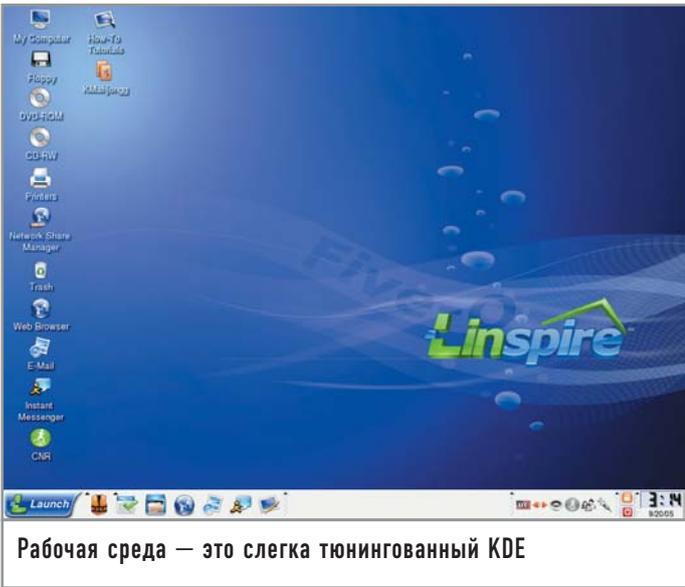
По окончании копирования система уходит на традиционную перезагрузку (порадовало, что Linspire сама отдает диск, снабжая этот процесс понятными даже школьнику картинками).

| Первая загрузка |

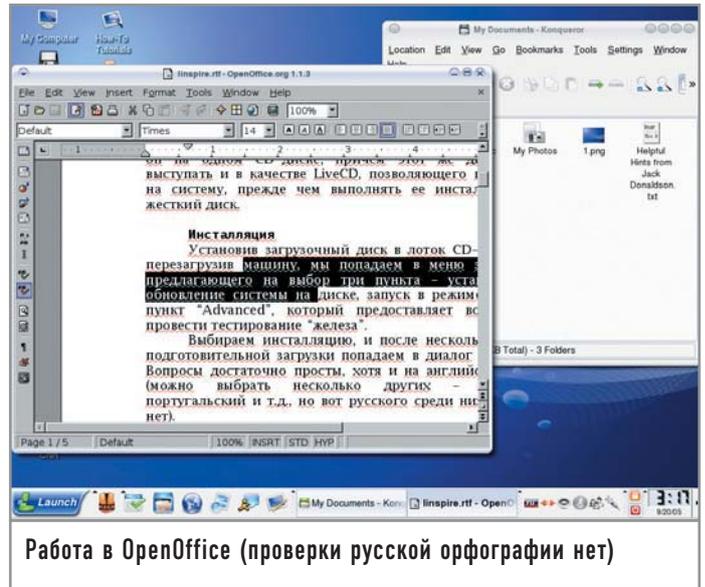
Итак, установка системы завершена. На все ушло около 20 минут, и если на этапе установки вы не отменили установку загрузчика в Master Boot Record (MBR), то увидите довольно симпатичное окно загрузчика (в основе которого лежит GRUB), в котором сможете выбрать загрузку Linspire, диагностику оборудования, а также загрузку других операционных систем, которые будут найдены на вашем жестком диске (или дисках, если их несколько).

При первой загрузке вам придется выполнить ряд действий:

- ▶ принять лицензионное соглашение;
- ▶ настроить уровень звука по умолчанию;
- ▶ выбрать часовой пояс и настроить текущее время.



Рабочая среда — это слегка тюнингованный KDE



Работа в OpenOffice (проверки русской орфографии нет)

Дополнительно вы сможете задать разрешение экрана, сменить пароль администратора, создать новых пользователей, настроить сеть и соединение dial-up. После этого вы попадаете в довольно симпатичный интерфейс операционной системы Linspire.

Графический интерфейс пользователя основан на KDE, так что у тех, кто знаком с этим оконным менеджером, никаких проблем с модификацией рабочего стола под свои предпочтения возникнуть не должно — все довольно традиционно настраивается в «Control Center».

| Справочная система |

При первой загрузке автоматически будет запущена обучающая программа. Вам подробно расскажут и покажут, как пользоваться системой, какие она имеет особенности и преимущества. В дальнейшем вы сможете в любой момент запустить ее из «Help Center» (How-To Tutorials).

| Состав дистрибутива |

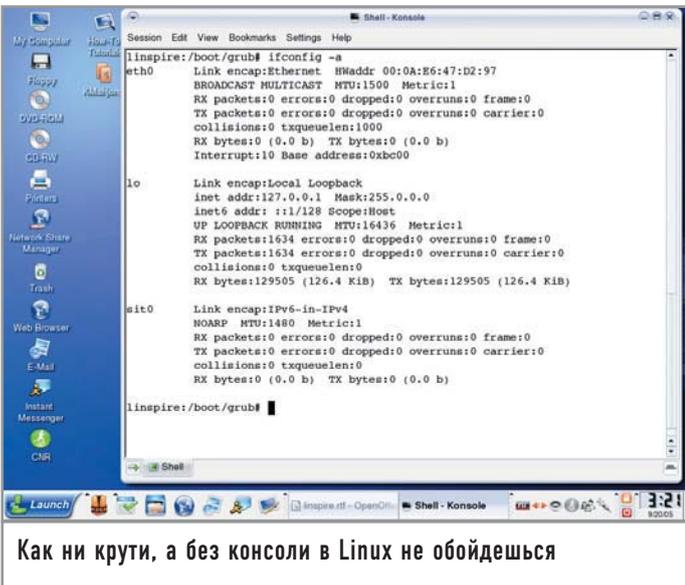
Что же можно встретить во вновь установленной Linspire? Все установленные программы сосредоточены в пункте главно-

го меню (кнопка «Launch») под названием «Run Programs». Здесь можно найти программы для работы с графикой, прослушивания музыки, а также ряд традиционных для KDE утилит, несколько игр. Порадовало наличие пакета OpenOffice.org.

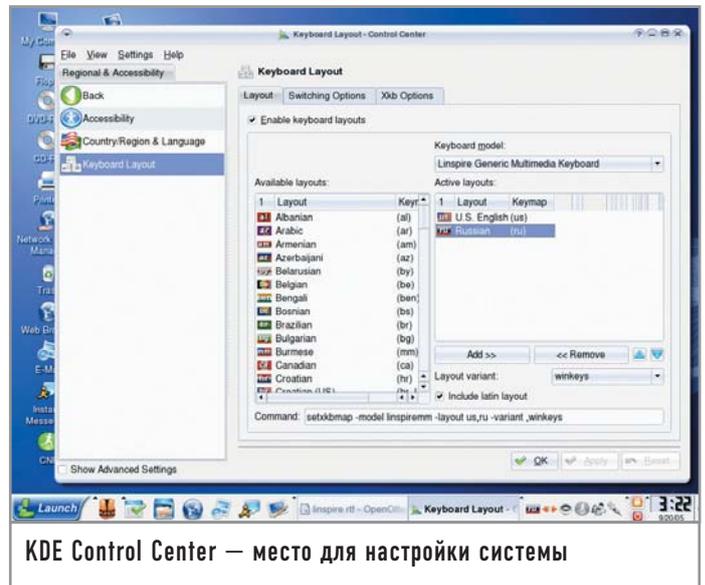
Несмотря на отсутствие русского языка, вы вполне можете выбрать соответствующую раскладку в KDE Control Center, что позволит вам в дальнейшем без особых проблем работать с русским языком (к слову, эта статья была написана в тестируемой системе Linspire только лишь с использованием имеющихся по умолчанию средств).

Для работы в Интернете имеется браузер из состава Linspire Internet Suite 1.6 на базе Mozilla 5.0. Есть и почтовый клиент из этого же пакета. Само собой, присутствует и Konqueror, так что вы можете выбрать его, если он вам больше нравится.

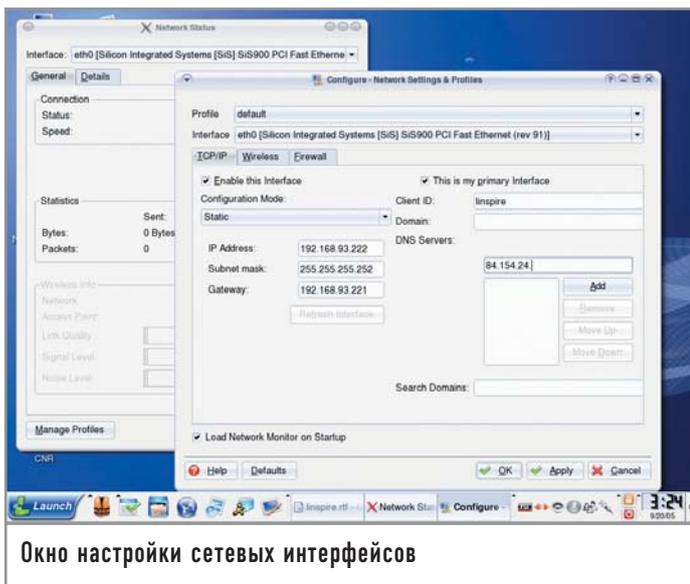
Из средств разработки в дистрибутиве присутствуют только продвинутые текстовые редакторы Emacs и Kate. Впрочем, в консоли вы можете в полной мере ощутить мощь Linux-систем, разрабатывая нужные для комфортной работы скрипты на Perl и Python. Кроме того, в системе имеется редактор vi.



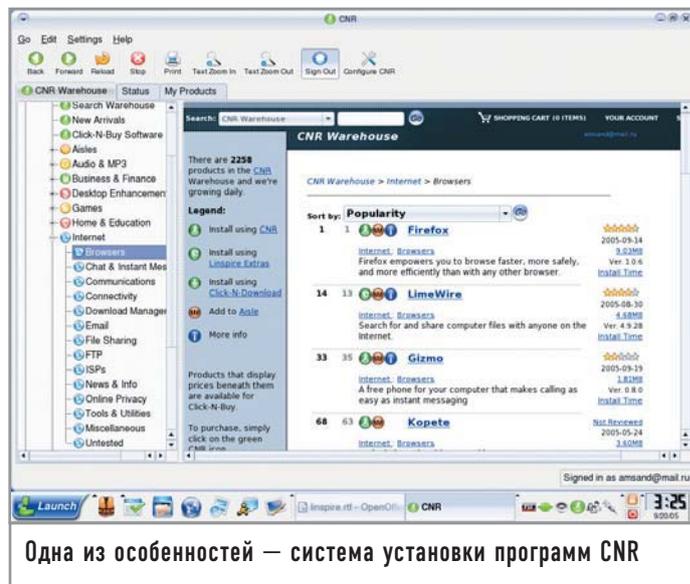
Как ни крути, а без консоли в Linux не обойдешься



KDE Control Center — место для настройки системы



Окно настройки сетевых интерфейсов



Одна из особенностей — система установки программ CNR

Установка дополнительного ПО

Однако вернемся из консоли (там, конечно, очень интересно, но вся соль дистрибутива все же не в ней) в GUI. Набора средств, имеющихся в системе сразу после установки, вполне достаточно для нетребовательных пользователей, которым от операционной системы нужны лишь традиционные «офис — мультимедиа — Интернет». Но рано или поздно возникает желание поставить новое приложение. Для этого Linspire предлагает свой подход.

Имя этому подходу — CNR (click-and-run). Данная система полностью оправдывает свое название: чтобы установить дополнительное приложение, достаточно соединиться с сервером CNR, выбрать из списка приложений нужное (на данный момент их число превышает 2000), почитать его описание и нажать на кнопку «CNR». Приложение будет скачано и установлено без лишних вопросов.

Правда, есть одно но — подписка на CNR платная. Сумма не столь уж и большая (выходит что-то около 50 рублей в месяц), тем не менее не каждый готов тратить свои кровные на то, что другие дистрибутивы позволяют делать бесплатно, хотя зачастую и с меньшим уровнем комфорта.

Попробовать CNR в работе можно и бесплатно, заполнив регистрационную форму. Вам будет предложено несколько приложений, которые вы сможете установить на свою систему. Вновь установленное приложение становится доступно через главное меню, где оно добавляется в соответствующую группу программ. Также на рабочем столе появляется иконка вызова этого приложения.

Консоль

Как и в любом дистрибутиве Linux, к вашим услугам окно терминала, где вы можете выполнять команды, не предусмотренные графическим интерфейсом. Например, здесь можно настроить брандмауэр iptables, выполнить диагностику сети (ping, netstat, traceroute и т. д.). Единственное, что несколько опечалило, — отсутствие традиционной справочной системы man. Впрочем, Linspire, судя по всему, и не предполагает активную консольную работу.

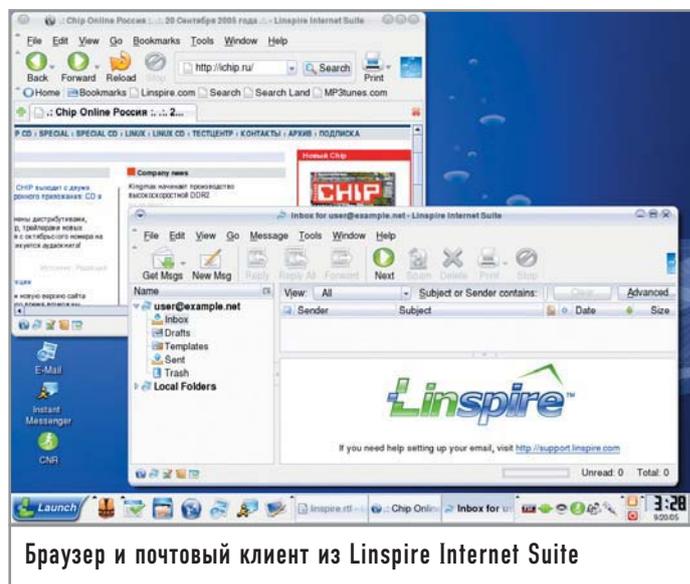
Заключение

Итак, после знакомства с Linspire 5.0 у нас сложилось вполне благоприятное впечатление об этой системе. Разработчики действительно приложили немало усилий для того, чтобы сделать работу с ней предельно простой и беспроblemной. Порадовал состав приложений, которые удалось вместить на один-единственный диск.

К недостаткам можно отнести тот факт, что базовая версия обойдется вам почти в \$50, плюс для комфортной работы нужно будет подписаться и на CNR. Также довольно неприятно отсутствие русификации.

Тем не менее Linspire вполне можно рекомендовать как альтернативу системам Windows для установки на офисные или домашние компьютеры, если вы хотите воспользоваться преимуществами Linux, но еще не готовы с головой погружаться в вопросы сборки собственных ядер, инсталляции приложений из исходных кодов и т. д.

В конце концов, основное завоевание Linux — свобода выбора системы, с которой вы хотите работать. |



Браузер и почтовый клиент из Linspire Internet Suite

Неповторимое лицо Mandriva

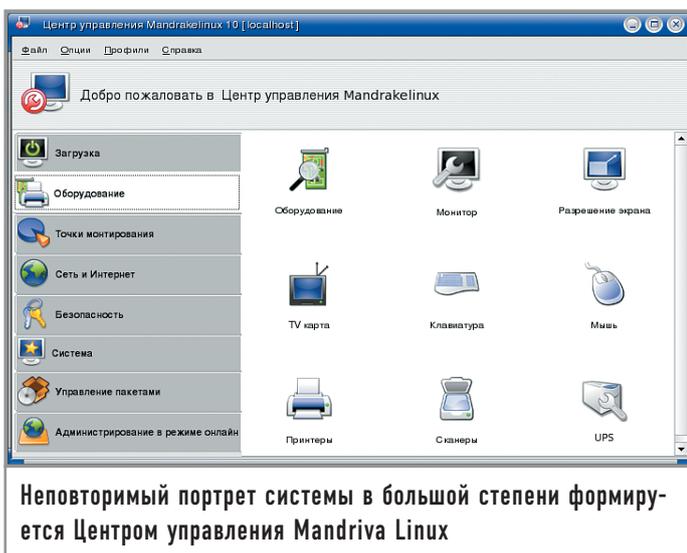
У каждого заслуженного бренда среди Linux-дистрибутивов есть какая-нибудь отличительная черта. Это может быть высокий уровень защиты или мощная система управления пакетами, мультимедийные возможности или производительность. Если говорить о Mandriva, то он обрел свою армию поклонников благодаря дружественному интерфейсу и простоте настройки, а главную роль в этом сыграл, несомненно, Центр управления Mandriva Linux.

Дистрибутивов Linux не просто великое множество — они очень разнообразные. Линейка ознакомительных, не требующих настройки продуктов LiveCD представлена такими громкими именами как Knoppix, Mandriva Move, Ubuntu. Дружелюбные помощники на вашем рабочем столе выходят из-под пера специалистов компаний Mandriva, Fedora Core, ALT Linux, Novell (бывшая SUSE). Те из них, что рассчитаны на глубокое освоение головоломки, например Slackware и Debian, осуществляют невольный «брейн-контроль» на входе, пропуская в свой мир только опытных пользователей. Особняком стоит дистрибутив-пазл Gentoo, который весьма долго устанавливается из исходных текстов, зато потом исправно служит верой и правдой.

Фирма Mandriva (ранее известная как Mandrakesoft) была основана во Франции в 1998 году Гаэлем Дювалем, который сегодня занимает в ней пост вице-президента. В это время на рынке «дружелюбных» дистрибутивов Linux доминировала Red Hat, внедряя в мир свободного программного обеспечения передовые технологии, такие как RPM — фирменный менеджер пакетов, сегодня уже ставший стандартом в большинстве дистрибутивов. Но французы — народ, очень тонко чувствующий эстетику, ценящий красоту; и группа энтузиастов из этой страны разработала план по созданию самого дружественного на свете и простого в освоении дистрибутива Linux. Номер первой версии этого продукта, являющегося ответвлением (fork) легендарного Red Hat 5.2, был выбран очень удачно — на единицу больше, чем у родителя. Так появился Mandriva 6.0. По-настоящему свое неповторимое лицо дистрибутив обрел в версиях 8.x, когда был отлажен его собственный графический инсталлятор, а разрозненные фирменные утилиты drake* объединились под одной общей панелью управления — Mandriva Control Center.

Генеральный штаб системы

Те, кто использует русскую (ru) локаль, называют Mandriva Control Center не иначе как Центром управления Mandriva Linux. Его физическая структура очень напоминает Центр управления средой GNOME, так как каждый значок, присутствующий на панели, является отдельным приложением, которое можно запустить из основного меню или консоли с помощью команд drake*. Но в этой статье мы рассмотрим Центр управления Mandriva Linux как единое целое. С точки зрения функциональности он похож на Панель управления Windows XP — с той лишь разницей, что настройки в



Mandriva Linux гораздо более разнообразны, и причина этого весьма понятна: в любом дистрибутиве Linux гораздо больше того, что можно настраивать, по сравнению с девственно чистой Windows XP.

Центр управления запускается через пункт меню «Настройка компьютера» и требует ввода пароля пользователя root. Слева можно видеть список закладок, а рабочее поле окна представляет собой набор значков, вызывающих конкретные настройки. Пройдемся вкратце по основным разделам.

| Загрузка (Boot) |

Автологин. Здесь настраивается рабочая среда, а также параметры пользователя, загружающиеся по умолчанию, но данную функцию можно и отключить. В этом случае при загрузке система каждый раз будет предлагаться вам сделать выбор пользователя и рабочей среды.

Загрузчик. Выбираете загрузчик — LILO или GRUB, указываете загрузочный раздел. Нажимаете на кнопку «Далее» и по шагам настраиваете все пункты. Например, можно изменить систему, загружающуюся по умолчанию. Опытные пользователи сразу заметят, что если, например, установлен загрузчик LILO, то данный пункт меню — не более чем фронтенд к `/etc/lilo/lilo.conf`.

Загрузочная тема. В этом разделе можно сменить заставку, появляющуюся при загрузке.

| Оборудование (Hardware) |

Оборудование. Если с железом возникают проблемы, чаще всего приходится обращаться именно к этому пункту. Автору этих строк, например, однажды довелось вручную выбирать звуковую плату, так как Linux опознала лишь встроенный в материнскую плату звуковой контроллер (забыл отключить, когда делал апгрейд). Напротив каждого элемента оборудования значится пункт меню «Запустить средство настройки». Например, у средства настройки жесткого диска — менеджер дисков, у видеоплаты — графический сервер, у CD-ROM — точки монтирования.

Далее следуют несколько пунктов меню, отвечающих за настройку графической системы на вашем компьютере. Все они, как правило, оперируют одним и тем же конфигурационным файлом — `/etc/X11/XF86config`.

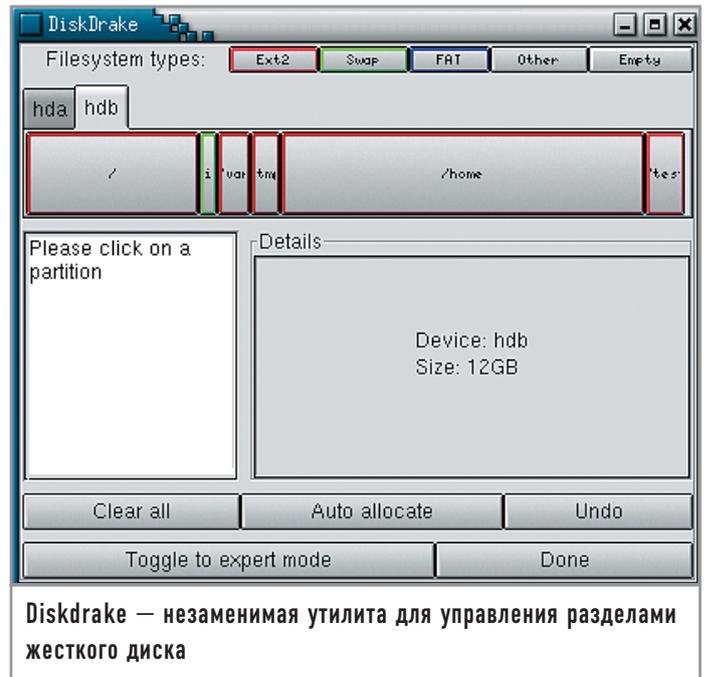
Монитор. В этом разделе можно указать стандартный PnP-монитор, поискать свою модель в списке и даже вручную указать частотные характеристики. В любом случае эта настройка влияет только на максимальное разрешение и соответствующие всем разрешениям частоты.

Разрешение экрана. Выбираете разрешение экрана и глубину цвета. Очень похожая утилита есть и в составе KDE.

Графический сервер. Все настройки графики собраны в одном пункте меню. Помимо этого из данного раздела можно сразу протестировать выбранные настройки.

ТВ-плата. Если у вас установлен ТВ-тюнер, в этом разделе можно выбрать и настроить плату.

Клавиатура. Задаете свой язык, а также раскладку клавиатуры, которая выбирается в меню «Оборудование». Mandriva 2005 поддерживает широкий модельный ряд мультимедийных клавиатур от различных производителей.



Мышь. Выбираете свой тип мыши. К сожалению, устройства, заряженные десятком кнопок и колесной формулой 4x4, в Linux пока стандартными средствами не поддерживаются.

Принтеры. Установка и настройка принтера. Вы можете запустить средство поиска локальных и сетевых принтеров. Список поддерживаемых моделей весьма широк, и, если вы не используете софтовый «win-принтер», для него, скорее всего, найдется подходящий драйвер.

Для принтеров предлагаются стандартные настройки, связанные с размером бумаги, качеством отпечатка, источником подачи, двухсторонней печатью.

Сканеры. Установка и настройка сканера происходят так же, как и в случае с принтером.

Благодаря фирменным утилитам настройка оборудования в Mandriva Linux становится задачей, доступной каждому. Вам даже не нужно будет в спешке менять компакт-диски с драйверами в своем накопителе, так как все драйверы Linux хранятся обычно на жестком диске.

| Точки монтирования (Mount Points) |

Разделы. Diskdrake — одна из самых мощных и полезных утилит, разработанная в недрах компании Mandriva. По возможностям она практически не уступает знаменитому дисковому менеджеру Partition Magic фирмы PowerQuest, который, правда, является закрытой платной программой и имеет версию только для Windows.

Diskdrake позволяет создавать разделы и форматировать их во множество файловых систем (в том числе закрытых, используемых, например, в Windows) и подобно Partition Magic может менять размеры разделов без потери данных.

Программа способна помочь в довольно распространенной ситуации: знакомый принес жесткий диск, и вам нужно его подключить. Автоматически он, разумеется, не смонтируется. Но Diskdrake покажет его, и останется только нажать на кнопку «Монтировать», чтобы увидеть содержимое устройства.

Здесь есть одна неприятная деталь. В `/etc/fstab` должна быть соответствующая строчка, которая позволяет монтировать данный тип подключения (`hdX`, где `X` может принимать значения от `a` до `d`). Заодно в `fstab` решается проблема с кириллицей.

Подключаемые устройства. Вы можете изменять точки монтирования накопителей CD-ROM, дисководов, флеш-карт, а также параметры монтирования. В программу заложены основные значения, которые можно регулировать установкой/снятием галочек в соответствующих чекбоксах. Кроме того, вы можете добавлять собственные параметры.

Точки подключения Samba. Программа сканирует сеть (разумеется, сеть должна быть настроена, а у вас должен быть в наличии сконфигурированный файл-сервер Samba) и отображает список доступных, «расшаренных» ресурсов. После этого вы можете указать, куда в файловой системе их необходимо монтировать. Если доступ к ресурсу требует ввода имени и пароля, можно сразу же здесь их и указать.

Точки монтирования NFS. Если в вашей сети используется протокол NFS, то монтирование ресурсов необходимо осуществлять с помощью этой программы. Настройка производится точно так же, как и у Samba.

Совместное использование локального диска. Можно разрешить или запретить пользователям открывать общий доступ к локальным ресурсам компьютера. Далее стоит выбрать, какие ресурсы позволено открывать — Samba или/и NFS.

Точки монтирования WebDAV. Если ваш компьютер играет роль веб-сервера, вы можете с помощью расширения WebDAV удаленно управлять его ресурсами, в частности изменять файлы.

| Сеть (Network & Internet) |

Новое соединение. Здесь сразу предлагается выбрать тип соединения, которое нужно настроить. В случае модемного соединения нужно будет согласиться с предложенным типом устройства и выбрать из огромного списка провайдера. Если ваш провайдер каким-то чудом не попал в этот список — не беда, создатели данной утилиты учтиво оставили пользователю возможность настроить этот параметр вручную. Далее нужно будет ввести название соединения, номер телефона, учетную запись, пароль и тип авторизации. Затем, если необходимо, заполнить IP-адрес, DNS и шлюз. И, наконец, указать, хотите ли вы автоматически запускать это соединение при загрузке. В списке доступных присутствуют ISDN, ADSL, беспроводное и кабельное соединения. Каждое из них в первую очередь предлагает выбрать устройство, благодаря которому будет осуществляться связь. Большинство описанных соединений работает через сетевые платы, поэтому нужно указывать еще и соответствующий драйвер.

Доступ в Интернет. Здесь можно поменять свое доменное имя, а также вручную ввести сведения о DNS-серверах.

Управление соединениями. Выбор устройства (сетевая плата, модем) и ручная настройка IP, маски сети, шлюза, DNS и домена.

Наблюдение за соединениями. Неплохой монитор сети. Графически и в цифрах показывает скорость соединения, время, входящий и исходящий трафик.

Удалить соединение. Эта опция предназначена для удаления выбранного вами соединения.

Прокси. Если вы выходите в Сеть через прокси-сервер, укажите его в этом пункте.

Совместное использование интернет-соединения. Пропишите устройство, с помощью которого вы выходите в Интернет (например, модем); затем устройство, через которое собираетесь предоставлять доступ другим желающим (например, сетевая плата). И, наконец, настройте сетевую плату для интернет-соединения. Если она уже ориентирована на выполнение других функций, ее придется перенастроить.

| Безопасность (Security) |

Уровень и проверки. При выборе этого пункта меню откроется дополнительное окно с рядом настроек.

► **Базовые опции.** Здесь нужно будет указать ваш уровень безопасности: от «низкого», который лучше всего использовать в полной виртуальной изоляции, до «параноидального», при котором, если уязвимости и остаются, то только программные, никак не связанные с настройками системы.

► **Сетевые опции** охватывают широкий спектр тонких сетевых настроек — от включения защиты от атак типа IP spoofing до проверки promiscuity-карт Ethernet.

► **Системные опции** также предлагают настроить большой список параметров безопасности — от разрешения/запрещения прямого входа в систему с правами пользователя root до установки сроков действия паролей и длительности периода, в течение которого учетные записи не будут активными.

► **Список периодических проверок** тоже велик — от диагностики баз данных RPM до ревизии прав файлов в локальных каталогах пользователей.

Права доступа. Здесь распределяются права доступа к различным каталогам файловой системы. По умолчанию все права принадлежат root, а пользователи имеют права на чтение почти везде (в `/root`, например, никто не имеет права даже заглядывать), а еще и на запись — в своих домашних каталогах `/home/<username>`.

Заметим, что с помощью утилиты `sudo` можно, к примеру, открыть пользователю (себе) доступ к командам `mount` и `umount`, а потом предоставить права на запись `/etc` (это даже не столь важно, если сразу прописать все в `fstab`). Теперь при подключении новых винчестеров, образов ISO, а также в других сферах применения команд монтирования переходить в режим суперпользователя не обязательно.

Firewall. В этом пункте выбираются типы соединений, которым можно выходить в Интернет, и для каждого из них прописываются разрешенные и запрещенные действия.

| Система (System) |

«Во всем виновата система», — сказал как-то в советское время один сантехник и угодил за решетку. Но мы с вами не будем винить систему во всех грехах, а грамотно ее настроим и заставим служить верой и правдой.

Меню. Так как с Центром управления Mandriva Linux может работать только пользователь root, то редактировать позволено лишь системное меню, доступное всем пользователям. В разделе «Окружение» можно выбрать три рабочие среды,

для которых будет осуществляться редактирование: KDE, GNOME и IceWM. Редактор позволяет выполнять все необходимые операции — создавать/удалять директории, пункты меню для приложений, добавлять приложениям описания и многое другое.

Оконный менеджер. Выбираем оконный менеджер и программу, которая будет рисовать окна: MkdKdm (версия менеджера kdm от Mandriva) или gdm.

Сервисы. Включение/выключение сервисов на ходу. Можно также дать команду сервисам запускаться при загрузке системы.

Шрифты. Настройка, установка и удаление шрифтов из системы. Если вы до последнего момента извлекали шрифты из Windows вручную, тайком поглядывая в какой-нибудь HOWTO, то в Mandriva об этом можно забыть — его Центр управления сделает всю работу за вас автоматически. От пользователя потребуется только указать каталог Windows со шрифтами.

Дата и время. В этом разделе вы настраиваете дату и время, включаете по желанию синхронизацию через Интернет с каким-нибудь сервером времени.

Логи. Здесь просматриваются журналы работы системы. В панели календаря можно указать временной интервал, за который они будут отображаться.

Консоль. Просто консоль с правами пользователя root.

Пользователи и группы. Очень удобный редактор, позволяющий добавлять и удалять пользователей и группы. Каждому пользователю задается имя, ID, группа (групп у пользователя может быть несколько), полное имя, оболочка входа, домашний каталог и статус. В отдельной закладке можно управлять группами — присваивать им имена и ID, вводить новых членов.

Резервные копии. Очень мощная утилита для создания резервных копий системы, включающая в себя несколько мастеров и диалоговых окон.

► Мастер настройки по шагам отвечает за резервное копирование. Сначала выбираете, что резервировать — всю систему или только отдельных пользователей (нужно указать, кого конкретно). Затем идет выбор метода резервирования — нарастающий или дифференциальный. В первом случае копируются будут только те файлы, которые изменились со времени последнего резервирования. Во втором будут копироваться файлы, измененные после первого резервного копирования. Теперь выбираете, куда копировать: на другой жесткий диск, в сеть, на CD/DVD или ленту. Настройка завершена.

► Расширенная настройка. Linux-версия игры «Что? Где? Когда?» — именно так называются подпункты меню. По большому счету, этот пункт фактически повторяет плейлист предыдущего с нажатой кнопкой «Shuffle» («Перемешать»).

► Просмотр конфигурации. В этом окне можно сразу включить процесс резервирования.

► Просмотр lastlog. Здесь доступен журнал последней операции резервирования.

► Зарезервировать немедленно. Тут и добавить нечего.

► Восстановить. Можно произвести операцию восстановления отдельного файла (есть встроенный движок поиска), целой копии и т. д. Здесь же указывается конкретный каталог, из которого будет проводиться восстановление.

Замечание: не пренебрегайте этой утилитой! Многие пользователи никогда не создают резервные копии системы, максимум копируя свои документы на CD-RW. Поверьте, это очень полезная возможность. Если вы ни разу не переставляли Linux по причине какого-либо сбоя, значит вы в ней просто не работали (или это делают за вас). Работая с правами root, вы подвергаете систему опасности, неосторожные действия могут привести ее к неминуемому краху. Иногда сбой ведет к переустановке. Вы проинсталлируете Linux и потратите вечер на восстановление настроек и приложений. Зачем усложнять себе жизнь? Сделайте резервную копию всей системы. В дальнейшем будут копироваться лишь изменения файлов, что займет считанные секунды (изредка минуты). Если случится сбой, восстановление пройдет за пару минут. Если вообще ничего не грузится, просто переставьте систему по минимуму (если, конечно, у вас нет диска наподобие Mandriva Move), потратив в результате те же несколько минут.

Пару лет назад ваш покорный слуга пробовал делать backup с помощью программы Acronis True Image 6.0 из Windows, которую она успешно восстанавливала, а вот Linux — нет. «Резервные копии» из Mandriva, напротив, здорово выручили.

| Управление пакетами |

Незаменимый и удобный набор утилит для управления RPM-пакетами. Здесь тоже есть несколько рубрик.

Установить, Удалить. Можете задать строку поиска необходимого пакета в базе или найти его вручную в списке слева. Справа вводится информация о каждом пакете — вплоть до того, какие файлы он приносит с собой на ваш жесткий диск. После этого выбираете необходимые пакеты и устанавливаете/удаляете их.

Обновления. Для работы этой утилиты необходимо интернет-соединение. Программа отыскивает на серверах обновленные пакеты, загружает и устанавливает их.

Менеджер. Здесь осуществляется управление носителями, на которых есть RPM-пакеты. По умолчанию менеджер настроен на диск (или диски) дистрибутива Mandriva Linux. Но никто не запрещает вам добавлять свои диски в этот список. При этом программа сканирует носитель на предмет наличия на нем RPM-пакетов и добавляет их в общую базу.

Помимо всего прочего Центр управления позволяет сохранять профили. Вы можете иметь несколько вариантов настройки системы. Но на этом уникальные возможности Mandriva не заканчиваются. Компания очень много трудится над художественным оформлением своих дистрибутивов. Несколько лет назад была разработана тема Galaxy, которая придавала окнам KDE и GNOME неповторимый «фирменный» вид. Позднее вторая версия этой темы стала еще краше. В дистрибутиве есть множество значков и иконок, нарисованных дизайнерами Mandriva. Все стандартные значки OpenOffice.org были заменены специальными, которые выглядят намного приятнее.

Mandriva Linux не зря называют самым дружелюбным дистрибутивом. Все его разработки направлены на простоту освоения системы, украшение ее внешнего вида. Работа никогда не останавливается, каждая последующая версия привносит что-то новое, еще дальше уводит в мир, в котором компьютер будет понимать вас с полуслова. |

Диалог с мобильным спутником

Эта статья рассчитана на достаточно продвинутых пользователей Linux, которые способны сами разобраться в подробностях. Рамки издания не позволяют детально рассмотреть вопросы обмена данными между настольным компьютером и КПК на базе Palm OS. Мы приведем краткий обзор существующих для этого средств.

| Общие сведения про обмен данными с Palm OS |

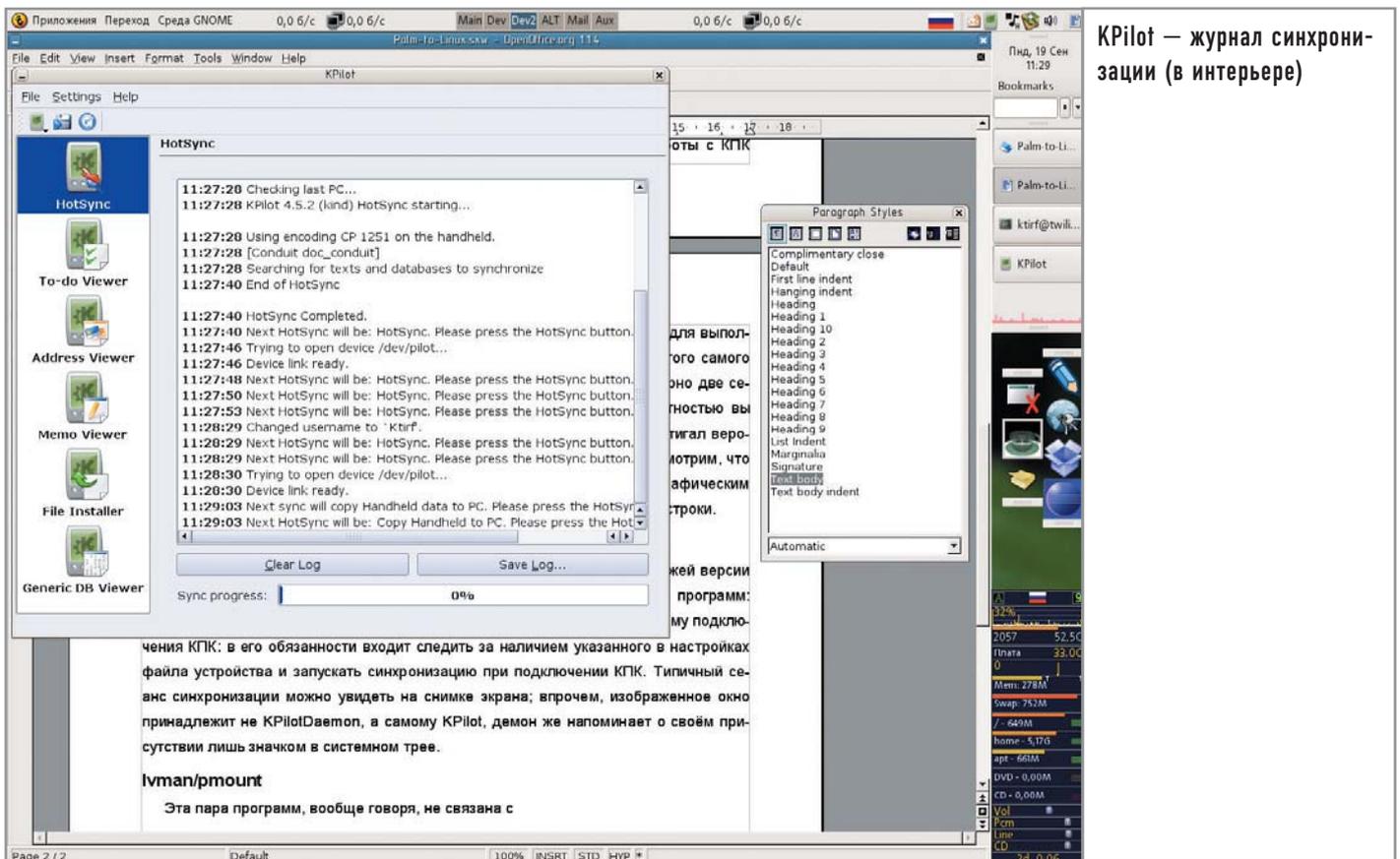
На сегодняшний день большинство КПК с операционной системой Palm OS комплектуются шнуром синхронизации с настольным ПК, рассчитанным на подключение через разъем USB. К более старым моделям можно найти шнур для COM-порта. Как ни странно, проблем с новым интерфейсом оказывается немного больше: дополнительные хлопоты возникают из-за того, что ядро Linux создает узлы в файловой системе /dev, соответствующие USB-устройствам, на лету и точно так же уничтожает их при отключении устройств. В результате, вообще говоря, нельзя с уверенностью сказать, как именно будет называться файл устройства при следующем подключении. Хуже того, КПК обнаруживается системой лишь после нажатия кнопки «HotSync», и файл устройства для него появляется примерно через пару се-

кунд после этого, что крайне неудобно, если программа, работающая с КПК, требует наличия файла устройства.

В отличие от USB файл, соответствующий COM-порту, есть всегда и никуда не девается, поэтому к нему можно жестко привязаться. Использование для передачи данных инфракрасного луча и Bluetooth очень похоже на USB (что касается работы с КПК) и сводится к использованию некоторого tty-подобного интерфейса, поэтому мы практически не будем о них упоминать. Начнем же с основы большинства способов синхронизации — пакета Pilot-Link.

| Pilot-Link |

Использование этого пакета — наверное, самый распространенный и универсальный способ работы с Palm. Он присутствует практически во всех современных дистрибутивах, рас-



KPIlot — журнал синхронизации (в интерьере)

считанных на использование в настольных системах, имеет довольно долгую историю и поддерживает множество моделей КПК. В его основе лежит одноименная библиотека, кроме того, в него входит набор программ, работающих в командной строке и позволяющих выполнять множество операций с КПК: передавать файлы PDB и PRC в обе стороны, делать резервное копирование и восстанавливать КПК из резервной копии, работать с адресной книгой, записками, фотографиями и т. д. Пользователям, которые предпочитают работать в командной строке, возможностей хватит.

Но никакой автоматизации процесса работы с КПК этот пакет не дает. Особенно остро это ощущается при работе с USB/IrDA/Bluetooth: для выполнения любой операции с КПК программы из Pilot-Link требуют наличия того самого файла устройства. Приходится сначала нажимать «HotSync», ждать примерно две секунды и после этого запускать нужную программу. С некоторой вероятностью вы угадаете, и Pilot-Link «поймает» подключенный КПК еще до того, как тот «потеряет терпение» (а оно теряется быстро). Автору этих строк удавалось ловить нужный момент где-то с вероятностью в 60–70%. Очевидно, что до удобства здесь далеко. Посмотрим, что с этим можно сделать. Для начала обратимся к двум (понятно каким) графическим средам, а затем попытаемся облегчить жизнь пользователям командной строки. Информацию о Pilot-Link вы можете найти по адресу www.pilot-link.org.

| KPIlot |

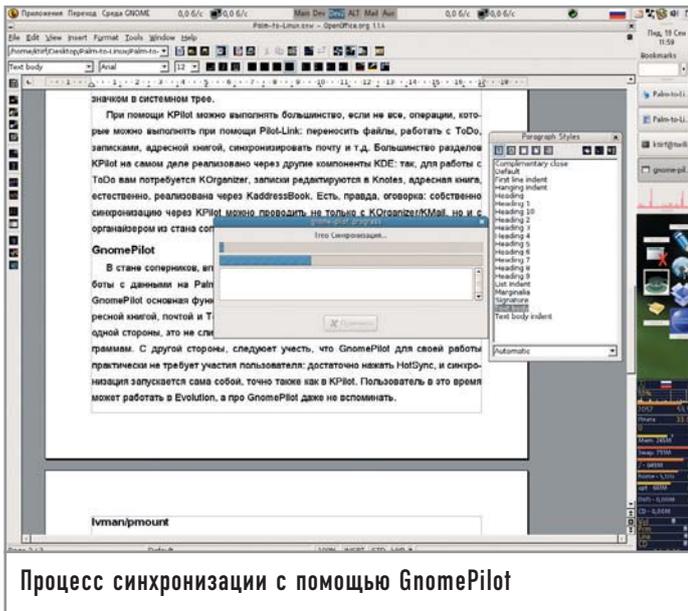
У пользователей KDE жизнь прекрасна: в любой мало-мальски свежей версии этой графической среды есть пакет

KPIlot, состоящий из двух программ — KPIlotDaemon и собственно KPIlot. KPIlotDaemon решает упомянутую ранее проблему подключения КПК: в его обязанности входит следить за наличием указанного в настройках файла устройства и запускать синхронизацию при подключении КПК. Типичный сеанс синхронизации можно увидеть на снимке экрана; большее окно принадлежит не KPIlotDaemon, а программе KPIlot, демон же напоминает о своем присутствии лишь значком в системном трее.

При помощи KPIlot можно выполнять большинство операций, доступных и при работе с Pilot-Link: переносить файлы, работать с ToDo, записками, адресной книгой, синхронизировать почту и т. д. Некоторые разделы KPIlot на самом деле реализованы через другие компоненты KDE: так, для работы с ToDo вам потребуется KOrganizer, записки редактируются посредством KNotes, адресная книга, естественно, реализована через KAddressBook. Есть, правда, оговорка: саму синхронизацию KPIlot может проводить не только с KOrganizer/KMail, но и с органайзером из стана соперников — Evolution. Страница программы — <http://pim.kde.org/components/kpilot.php>.

| GnomePilot |

В стане соперников, впрочем, тоже не бедствуют: здесь есть своя программа для синхронизации данных. Если сравнивать с KPIlot, можно сказать, что GnomePilot — это и есть демон синхронизации, а работа с адресной книгой, почтой и ToDo остается на долю модуля для программы Evolution. С одной стороны, это не слишком удобно: работа с КПК разнесена по нескольким программам. С другой стороны, GnomePilot для своей работы



Процесс синхронизации с помощью GnomePilot

практически не требует участия пользователя: достаточно нажать «HotSync», и синхронизация запускается сама собой — точно так же, как в KPilot. Пользователь в это время может работать в Evolution, а про GnomePilot даже не вспоминать.

К достоинствам GnomePilot можно отнести наличие в явном виде графического интерфейса для синхронизации не только через USB, но и последовательный и инфракрасный порты, и даже через Сеть. Кроме того, синхронизацию GnomePilot осуществляет с помощью набора подключаемых модулей (conduits), аналогичных модулям для ColdSync, о которых речь пойдет в конце статьи. В итоге GnomePilot оказывается более расширяемым и настраиваемым средством синхронизации, нежели KPilot, однако не содержит никаких средств для работы с информацией, хранящейся на КПК под управлением Palm OS. Подробнее о программе вы можете узнать на сайте <http://live.gnome.org/GnomePilot>.

| JPilot |

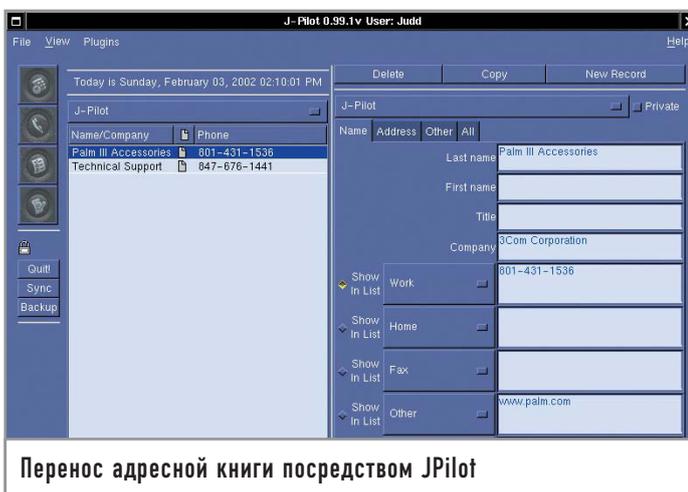
Ну хорошо, а что делать тем, кто не пользуется Evolution или KOrganizer? Без них KPilot и GnomePilot могут только пересылать файлы и осуществлять резервное копирование. Хотелось бы все-таки пользоваться наличием настольной ма-

шины с нормальной клавиатурой для работы с органайзером. Так вот, если вы не используете на настольной машине органайзер или если он не относится к вышеупомянутому мейнстриму, вам может пригодиться JPilot. Эта программа довольно точно (до некоторой степени) воспроизводит штатную утилиту от palmOne — PalmDesktop, работающую исключительно под операционными системами компании Microsoft. Как нетрудно догадаться, JPilot (www.jpilot.org) тоже использует для работы Pilot-Link, но в отличие от KPilot демона синхронизации здесь нет. Очевидное следствие — для успешной синхронизации вам придется ловить момент теперь уже для нажатия кнопки «Синхронизация» вместо «Enter» в командной строке.

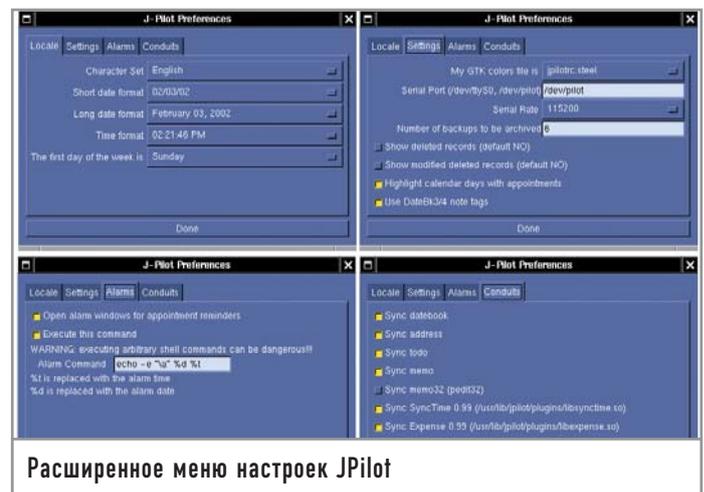
| ColdSync |

В заключение несколько слов об альтернативном проекте Pilot-Link, который решает вопрос работы с КПК не в смысле передачи данных, а в смысле синхронизации. Он тоже достаточно популярен, и его можно встретить во многих дистрибутивах. В отличие от Pilot-Link в пакете ColdSync (www.coldsync.org) нет библиотеки, а есть всего два исполняемых файла. Один из них выполняет роль универсального средства синхронизации, а второй является примитивным сервером имен (он нужен только при синхронизации с сетевой машиной).

ColdSync может работать как демон (не являясь, по сути, настоящим демоном), ожидая подключения КПК и синхронизируя его с настольной машиной. В этом состоит его выгодное отличие от Pilot-Link, для которого приходится использовать дополнительные средства автоматизации. Организация работы изначально направлена в первую очередь именно на синхронизацию, а не просто передачу данных туда или обратно. Весь процесс в ColdSync построен на специальных модулях, которые называются conduits (в переводе с английского — «канал, труба, трубопровод»). Каждый из них обычно занимается синхронизацией данных определенного рода, например только посылает почту, или только синхронизирует адресную книгу. В результате получается весьма гибкая и расширяемая архитектура. Аналогичные по назначению модули используются и в GnomePilot, но модули одной программы не подходят к другой. |



Перенос адресной книги посредством JPilot



Расширенное меню настроек JPilot

Возможна ли жизнь без Apache?

Популярные решения не всегда являются оптимальными. Это особенно применимо к веб-серверам. В данной статье мы постарались раскрыть слабые места Apache и попытались предложить достойную альтернативу.

Как это ни странно, но я не люблю HTTP-сервер Apache. Да, это самый распространенный на сегодняшний день HTTP-сервер (по данным Netcraft), но большей частью эта популярность объясняется не его удобством, а инерцией администраторов, которые его устанавливают, потом передают своим приемникам и т. д. (вспомним «сладкую парочку» LAMP — Linux/Apache/MySQL/PHP, которую многие начинающие системные администраторы считают панацеей от всех бед и единственно верным решением).

Чем не угодил Apache

Раньше эта инерция кое-как была оправдана — в стародавние времена медленного Интернета и протокола HTTP/1.0 не было альтернатив. А сегодня, когда контент стал более динамичным, скорости перешагнули гигабитный рубеж и количество новых протоколов уже не поддается учету, позиции Apache не так безоговорочны. Рассмотрим основные причины, почему Apache нельзя назвать совершенным HTTP-сервером и почему надо серьезно подумать перед его установкой и, возможно, выбрать более гибкое и удобное решение, чем стандартный LAMP.

► Неоправданно большое потребление памяти. Согласно информации, приведенной в статье Дениса Смирнова (www.freesource.info/wiki/Anti_Apache), в Apache версии 1.3.x каж-

дое параллельное соединение обрабатывается отдельной копией (дочерним процессом) Apache (в 2.x вместо копии используется нить исполнения, что несколько экономнее). Любая копия Apache требует для исполнения приблизительно 2 Мбайт памяти, то есть если нужно обработать 100 соединений, то для этого потребуется 200 Мбайт. Ну а если соединений несколько тысяч? Думаю, не у каждого есть в запасе несколько гигабайт свободной оперативной памяти (прибавим еще и SQL-сервер — в результате получаются совсем заоблачные цифры). А ведь несколько тысяч одновременных соединений совсем не редкость, многие форумы и популярные новостные сайты как раз имеют такую посещаемость, да еще и как среднее значение!

► Неправильная работа при большом количестве низкоскоростных подключений. Реализовано просто ужасно — на одно соединение выделяется копия объемом несколько десятков мегабайт, при этом средний размер передаваемых данных колеблется в пределах десятка килобайт. Согласитесь, память можно было бы расходовать и более экономно.

► Нерациональная работа со статическими данными. Отчасти это объясняется причинами, изложенными выше, и тем, что Apache уже староват для современного мира — например, передачу/чтение большого количества данных сегодня гораздо эффективнее делать не с помощью классических read/write, а более продвину-

тыми механизмами, например `sendfile()`, или использовать `epoll/kqueue` вместо стандартного `poll()`.

► Небезопасная модель работы. Вспомним еще и такие неприятности как модульная архитектура сервера без надлежащего разделения привилегий между модулями, и ситуация окажется совсем безрадостной. Либо нам нужен суперкомпьютер и постоянный аудит всех сторонних модулей к серверу (что не всегда возможно), либо нужно выбрать что-то другое.

Кое-как все эти недостатки можно причесать и поправить. Например, для кеширования ответов (и, как следствие, экономии памяти) использовать модули поддержки так называемого режима `reverse проху` (небезызвестный `mod_accel` Игоря Сысоева) или вообще специализированные решения — `squid`, `oops`. Но все остальные проблемы останутся с нами, особенно самая неприятная — невозможность стандартными средствами разделить в Apache понятия frontend (`reverse проху`, или статический сервер) и backend (динамический сервер) из-за «неисправимых преимуществ» смешивания этих качеств в одной программе.

| Реальный пример из жизни |

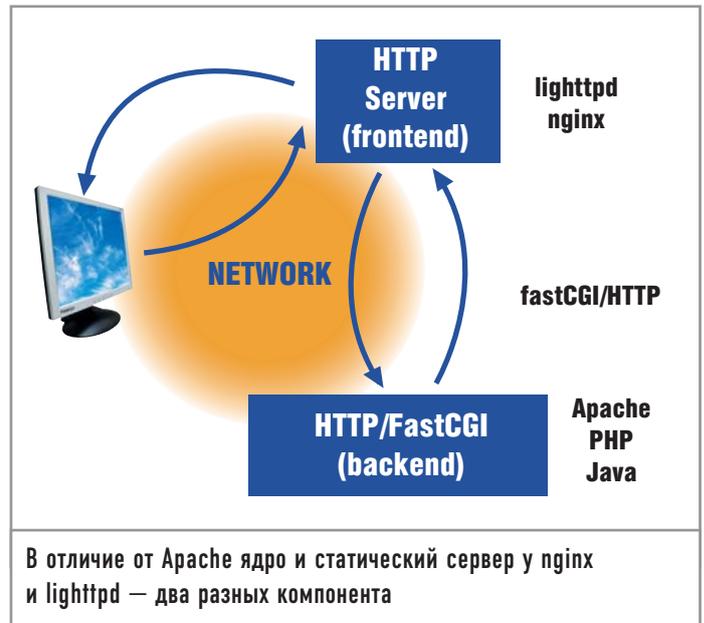
Чтобы усилить эффект внушения, приведу один пример — после замены связки Apache+`mod_php` на `nginx+php-fcgi` средний `loading` упал с 7 до 0,3! И это при том, что `nginx` не «подкручивался» по настройкам, а Apache был оптимизирован по всем пунктам — от конфигурации до флагов при сборке. Сайт, на котором производилась замена, имел постоянную посещаемость в районе нескольких тысяч разнородных запросов в час. Подозреваю, что если туда поставить `squid`, то нагрузку можно снизить еще больше. И при этом железо на сервере и его конфигурация далеки от идеала — это обычная машина с медленными IDE-дисками и небольшим объемом памяти, занятой другими, более «прожорливыми» программами.

| Список участников |

Итак, вы решились. Но что выбрать? Во-первых, это должен быть HTTP-сервер, удовлетворяющий целому ряду требований:

- поддержка CGI/FastCGI;
- поддержка современных возможностей передачи данных (как минимум `sendfile/sendfile64, epoll/kqueue`);
- малый расход памяти при большом количестве соединений;
- поддержка режима `reverse проху`;
- поддержка `url rewriting` (с функционалом не хуже `mod_rewrite`);
- поддержка `userdir/vhost` (для массового хостинга);
- поддержка базовых возможностей HTTP-сервера: SSI, `gzip`-сжатие данных, SSL, TLS, авторизация пользователей;
- безопасность работы (поддержка `chroot()`, `setuid/setgid`).

Я расскажу о тех серверах, которые либо использовал сам, либо слышал отзывы об их работе. Конечно, этот список не велик, и многие другие интересные проекты остались «за бортом», но по крайней мере мой рассказ поможет лучше ориентироваться среди альтернативных HTTP-серверов и выбрать то, что будет оптимальным решением. Все функциональные возможности рассматриваемых серверов объединены в единую таблицу, для каждого сервера имеется краткое описание. В конце статьи на примерах показаны принципы настройки каждого из них.



| Cherokee |

| Сайт проекта — www.0x50.org
 Автор — Alvaro Lopez Ortega |

Один из самых старых серверов: его разработка началась в 2001 году. Отличается гибкостью настройки и сборки (возможна даже сборка для встроенных устройств под названием `u-Cherokee`). Сервер имеет сильно модулированную архитектуру, где каждое расширение (`handler`) выполняет определенные функции (передача файлов, поддержка CGI/FastCGI, построение списка директорий и т. д.). Каждый модуль изолирован от «ядра» сервера, при этом есть возможность «сбросить» права при запуске или изолировать исполнение в отдельном каталоге с помощью вызова `chroot()`. К сожалению, работа сервера не отличается стабильностью — у меня он довольно часто аварийно завершал работу даже при включении стандартных расширений и возможностей, например поддержки `epoll`. Не совсем корректно реализована поддержка сжатия данных с помощью `gzip` — например, даже не проверяется, сможет ли клиент их обработать. Присутствуют ошибки в реализации SSL/TLS, причем довольно грубые. Но в целом, если вам нужен быстрый и нетребовательный к ресурсам HTTP-сервер для небольшого сайта с PHP/CGI/FastCGI, или если вы заинтересованы в разработке встроенного HTTP-сервера или клиент-серверных приложений на базе библиотек Cherokee, то этот проект может вас заинтересовать. Готовые сборки Cherokee есть в Linux-дистрибутивах Debian, Gentoo и в портах xBSD-систем — NetBSD, OpenBSD, FreeBSD.

| lighttpd |

| Сайт проекта — www.lighttpd.org
 Автор — Jan Kneschke |

Самый продвинутый из «легкого» семейства. Обладает всеми возможностями «взрослого» HTTP-сервера, включая поддержку массового хостинга, FastCGI, SSL/TLS и средства управления трафиком. Очень прост в конфигурировании, при этом не в ущерб функциональности. Для `lighttpd` в Сети существует очень много разнообразных HOWTO по настройке

и положительных отзывов об использовании, его собственная документация отличается подробным описанием всех опций и параметров. Поддерживается режим работы в изолированном каталоге с помощью вызова chroot().

| nginx |

| Сайт проекта — www.sysoev.ru/nginx

Автор — Игорь Сысоев |

Сайт дополнительной документации и полезные статьи по использованию этого сервера — www.nginx.info.

Очень динамично развивающийся проект. Хотя номер версии еще далек от заветного 1.x, сервер вполне функционален и стабилен в работе. По словам автора, различные версии nginx работают на многих загруженных сайтах и нескольких серверах Rambler. Как и lighttpd, он обладает всеми «взрослыми» возможностями, включая такие полезные как встроенный гео-модуль, позволяющий в зависимости от IP-клиента задавать различные действия, и модуль интеллектуального сжатия данных с помощью алгоритмов gzip/deflate. Сервер поддерживает все новшества ядер Linux/FreeBSD, такие как sendfile/sendfile64/kqueue/sendfilev, epoll для Linux, /dev/poll для Solaris и ассерт-фильтры для FreeBSD. По заверениям разработчика, на 10 000 неактивных keep-alive-соединений расходуется всего 2,5 Мбайт памяти! Если вы захотите собрать nginx самостоятельно, то будете приятно удивлены интересной системой сборки сервера — она отличается от стандартных autoconf/automake, но позволяет более гибко регулировать все шаги процесса, включая даже разновидности компилятора (кроме обычного gcc поддерживается icc, scc, asc и bcc). Еще одна интересная и полезная особенность nginx — это способ управления. В отличие от стандартных SIGTERM/SIGHUP для перезапуска и обновления конфигурации nginx можно управлять аж шестью сигналами, при этом дочерние процессы поддерживают дополнительно еще три сигнала. Все это позволяет выполнять плавную перезагрузку сервера в случае изменения конфигурации или завершения работы процессов. Таким образом, даже в случае большого количества открытых соединений клиенты даже не заметят «подмены» в работе сервера. К сожалению, пока на сайте проекта очень мало документации, так как Игорь больше занят разработкой, а не составлением описания работы, но этот недостаток легко восполняется русскоязычным списком рассылки, где можно задать вопрос разработчику и другим пользователям nginx. И, наверное, самый хороший источник информации по дополнительным параметрам работы — это непосредственно исходный код nginx: да-да, его можно просто читать и получать удовольствие от этого.

| Приступаем к конфигурации |

Напоследок я решил продемонстрировать, как можно настроить сайт на базе вышеперечисленных серверов. В качестве контента возьмем популярную программу imp (www.horde.org/imp), которая вполне годится на роль feature-rich-приложения. Она представляет собой основанную на протоколе IMAP почтовую систему. Написан IMP на PHP. В простейшем виде, наша конфигурация должна выглядеть так:

| Оригинальный httpd.conf |

```
#
# File: horde.conf
#
# This is the Horde Apache configuration file; it is included from the
# Apache httpd.conf file.
#
Alias /horde /var/www/html/addon-modules/horde/
Alias /horde/ /var/www/html/addon-modules/horde/

<Directory /var/www/html/addon-modules/horde>
    Options Indexes FollowSymLinks
    AllowOverride None
    order allow,deny
    allow from 127.0.0.1
</Directory>
<Directory "/var/www/html/addon-modules/horde/config">
    order deny,allow
    deny from all
</Directory>
<Directory "/var/www/html/addon-modules/horde/lib">
    order deny,allow
    deny from all
</Directory>
<Directory "/var/www/html/addon-modules/horde/locale">
    order deny,allow
    deny from all
</Directory>
<Directory "/var/www/html/addon-modules/horde/po">
    order deny,allow
    deny from all
</Directory>
<Directory "/var/www/html/addon-modules/horde/scripts">
    order deny,allow
    deny from all
</Directory>
<Directory "/var/www/html/addon-modules/horde/templates">
    order deny,allow
    deny from all
</Directory>
#
# File: imp.conf
#
# This is the IMP Apache configuration file; it is included from the
# Apache httpd.conf file.
#
<Directory "/var/www/html/addon-modules/horde/imp/config">
    order deny,allow
    deny from all
</Directory>
<Directory "/var/www/html/addon-modules/horde/imp/lib">
    order deny,allow
    deny from all
</Directory>
<Directory "/var/www/html/addon-modules/horde/imp/locale">
```

```

order deny,allow
deny from all
</Directory>
<Directory "/var/www/html/addon-modules/horde/imp/po">
    order deny,allow
    deny from all
</Directory>
<Directory "/var/www/html/addon-modules/horde/imp/scripts">
    order deny,allow
    deny from all
</Directory>
<Directory "/var/www/html/addon-modules/horde/imp/tem-
plates">
    order deny,allow
    deny from all
</Directory>

```

```

Handler common
    DocumentRoot
        /var/www/html/var/www/html/addon-modules/horde
        Allow from 127.0.0.1
    }
# Cherokee не поддерживает параметр deny from all, поэтому
ничего про это не пишем
# Rest of the virtual host configuration..
# Add your parameters here..
#
Extension php, php4 {
    Handler phpcgi {
        Interpreter /usr/bin/php-cgi
    }
}

```

Как видите, конфигурационный файл совсем небольшой, но все вроде работает.

Cherokee

К сожалению, Cherokee не поддерживает FastCGI, и пришлось использовать обычный CGI-модуль для PHP. Ниже приведен пример конфигурационного файла cherokee.conf:

```

DocumentRoot /var/www/html/var/www/html
Directory /horde {

```

lighttpd

Пример конфигурационного файла lighttpd.conf

```

# сначала подключаем нужные модули:
# access (для управления доступом)
# compress (для gzip-сжатия трафика)

```

Название сервера	Cherokee	lighttpd	nginx
Версия	0.4.25	01.03.14	0.1.41
Лицензия	GPL	BSD	BSD
Поддерживаемые ОС	Windows Linux OpenBSD FreeBSD NetBSD Sun Solaris	Linux *BSD ¹ SGI IRIX Windows (Cygwin)	Linux *BSD ¹ Tru64 UNIX HP/UX Windows
Поддержка sendfile()	•	•	•
Поддержка sendfile64	•	•	•
Поддержка sendfilev()	•	•	•
Поддержка epoll()	•	•	•
Поддержка kqueue()	–	•	•
Поддержка /dev/poll	–	•	•
Поддержка accept_filter	–	•	•
Поддержка TCP_DEFER_ACCEPT	•	–	•
Поддержка режима reverse proxy	–	•	•
Поддержка CGI	•	•	•
Поддержка FastCGI	–	•	• ²
Поддержка chroot() и сброса привилегий	•	•	–
Поддержка распределения нагрузки (load balancing)	–	•	•
Поддержка сжатия gzip/deflate	•	•	• ³
Поддержка vhost	•	•	•
Поддержка userdir	•	•	–
Поддержка SSL/TLS	GNUTLS/OpenSSL	OpenSSL	OpenSSL
Поддержка PCRE-синтаксиса	собственная реализация	через библиотеку pcre	через библиотеку pcre
Поддержка URL rewriting	•	•	•
Поддержка HTTP-авторизации	•	•	•
Ограничение скорости отдачи ответов	–	•	•
Стандартные возможности HTTP-сервера (SSI, пользовательские коды ошибок и т. д.)	частично (SSI не поддерживается)	•	•

¹ Подразумевается все семейство ОС BSD, включая Sun Solaris.

² nginx не умеет самостоятельно управлять FastCGI-сервером.

³ nginx наиболее правильно поддерживает этот режим, так как учитывает параметры запроса клиента.

```
# fastcgi (для поддержки FastCGI версии PHP)
# alias (для создания alias)
server.modules = (
    "mod_access",
    "mod_compress",
    "mod_fastcgi",
    "mod_alias",
    "mod_accesslog")
# задаем DocRoot
server.document-root = "/var/www/html"
# подгрузка fastcgi
fastcgi.server= ( ".php" =>
    "localhost" =>
    (
        "socket" => "/var/spool/lighttpd/php-fastcgi.socket",
        "bin-path" => "/usr/bin/php-cgi"
    )
)
# задаем alias
alias.url = ( "/horde/" => "/var/www/html/addon-
modules/horde/" )
# задаем ограничения на доступ
$HTTP["url"] =~ "^/(config|lib|locales|templates|po|scripts)/" {
    url.access-deny = ( "" )
    server.dir-listing = "disable"
}
$HTTP["host"] !~ "127.0.0.1"
    url.access-deny = ( "" )
}
```

Как видно, получилось даже лучше, чем с Apache.

| nginx |

Пример конфигурационного файла nginx.conf:

```
# передаем все запросы на php backend
# к сожалению, nginx не умеет самостоятельно запускать
# fcgi-сервер, поэтому
# запустим его через утилиту spawn-fcgi из lighttpd.
location ~* ^.+\.php$ {
```

Дополнительные источники информации

Полезные ссылки

Денис Смирнов, «В этом мире есть не только Apache»: www.freesource.info/wiki/Anti_Apache

Константин Лепихов, «PHP FastCGI»: www.freesource.info/wiki/Stat'i/PhpFastCGI?v=179h

Ralf S Engelschall, «Practical approaches for distributing HTTP traffic»: www.engelschall.com/pw/wt/loadbalance/article.html

Dan Kegel, «The C10K problem»: www.kegel.com/c10k.html

```
fastcgi_pass <fcgi host>:<fcgi port>;
fastcgi_index index.php;
fastcgi_param SCRIPT_FILENAME
/var/www/html/forum$fastcgi_script_name;
fastcgi_param QUERY_STRING $query_string;
fastcgi_param REQUEST_METHOD $request_method;
fastcgi_param CONTENT_TYPE $content_type;
fastcgi_param CONTENT_LENGTH $content_length;
# занимает ресурсы, без нужды не использовать
fastcgi_param REMOTE_ADDR $remote_addr;
fastcgi_param SERVER_PORT $server_port;
fastcgi_param REDIRECT_STATUS 200;
# специально для php-cgi без этого переменная
PHP_SELF
# будет пустая
fastcgi_param SCRIPT_NAME $fastcgi_script_name;
}
# теперь можно заходить по адресу http://.../imp
server {
    listen <ip>:80;

    sendfile on;
    tcp_nodelay on;

    gzip on;
    root /var/www/html/addon-modules/horde;
    index index.php;
    location = / {
        root /var/www/html/addon-modules/horde/imp;
        index index.php;
        allow from 127.0.0.1;
    }
    location / {
        allow from 127.0.0.1;
        rewrite ^/horde(.*)$ $1 break;
        rewrite ^/favicon.ico$ /graphics/favicon.ico break;
    }
    location ~* \.php {
        rewrite ^/horde(.*)$ $1 break;
    }
}
# настроим доступ
location ~ = ^/(config|lib|locales|templates|po|scripts)/ {
    deny all;
}
}
```

| Так есть ли жизнь без Apache? |

Из приведенных примеров видно, что конфигурационные файлы альтернативных HTTP-серверов проще, чем у Apache. И это без потери их функциональности. Так стоит ли идти на поводу у привычки и закостенелых взглядов и использовать привычный многим пользователям сервер Apache? Попробуйте одну из рассмотренных нами легковесных программ, и, возможно, сервер вашего популярного веб-ресурса еще долго не придется подвергать апгрейду. |

Сергей Яремчук

Игра на чуждом поле

Когда-то никто из пользователей серьезно и не задумывался о поддержке файловой системы NTFS в операционной системе GNU/Linux. С выходом Windows XP ситуация кардинально изменилась, теперь поддержка NTFS является одним из требований при выборе дистрибутива.

В принципе работа по поддержке NTFS в ядре Linux ведется уже давно. Так, еще в 1995 году был доступен патч для работы с этой файловой системой для ядер серии 2.0, а начиная с версии 2.1.74 поддержка NTFS была включена стандартно. На сегодняшний день практически все производители оснастили поддержкой NTFS свои дистрибутивы. Проблем с доступом к разделам NTFS не обнаружат пользователи Mandriva, SUSE, ALT Linux, ASPLinux, Slackware, Debian и прочих популярных программных продуктов. Исключение составляют лишь Red Hat, Fedora и их производные, которые, руководствуясь лицензионной чистотой, не включили поддержку данной ФС, хотя для них существуют RPM-пакеты, позволяющие получить доступ к ее разделам. А некоторые дистрибутивы, среди которых Phat Linux (phatlinux.com) и TopologiLinux (www.topologilinux.com), могут быть установлены на раздел, отформатированный под NTFS.

На сегодняшний день известны три проекта, позволяющие получить доступ к разделам NTFS. Два из них распространяются свободно, третий является коммерческим. При этом первые два пошли путем написания драйвера, а третий пытается решить эту проблему с помощью эмуляции.

| Драйверы проекта Linux-NTFS |

В настоящее время для GNU/Linux существуют два драйвера NTFS, разработанные в рамках этого проекта. Первый по умолчанию используется в ядрах серии до 2.4.18 и в 2.5.0-2.5.10. Он имеет ограниченные возможности по записи в раздел NTFS, но использовать его очень опасно, так как это может привести к разрушению файловой системы. Последняя версия — 1.1.22, разработка первой ветки уже приостановлена. Все усилия со-

средоточены на второй версии, которая была фактически переписана заново одним из разработчиков — Антоном Алтапармаковым. Этот драйвер имеет некоторые возможности по записи, но они весьма ограничены. Так, он может записать данные поверх уже существующего файла, но при этом не может изменить размер, добавить новые или удалить имеющиеся файлы. Новый драйвер, разрабатываемый Linux-NTFS (linux-ntfs.sourceforge.net), поддерживает NTFS версий 1.2, 3.0 и 3.1, Unicode и сжатые файлы. Свои проблемы есть и здесь. Драйвер не работает с зашифрованными файлами и квотами, игнорирует информацию безопасности. Положительная же его сторона — умение работать в мультипроцессорных конфигурациях.

Впрочем, винить разработчиков в этой не очень радужно складывающейся ситуации не следует. Строчение NTFS довольно сложное и напоминает базу данных: изменение в одном месте влечет за собой достаточное количество замен и во многих других объектах файловой системы, иначе она попросту будет разрушена. При этом система не документирована, и драйверы приходится писать фактически вслепую.

Новый драйвер включается в ядро начиная с версии 2.5.11, поэтому если у вас не сложилось с его поддержкой, то необходимо перекомпилировать ядро, включив нужные пункты. Для более ранних версий доступен патч.

Используемую версию драйвера узнать очень просто:

```
# dmesg | grep -i ntfs
NTFS driver v2.1.6b [Flags: R/O MODULE]
NTFS volume version 3.1.
```

Или другой вариант:

```
# grep -i ntfs /var/log/messages
```

Если вывод этих команд ничего не дал, вероятнее всего, NTFS попросту не поддерживается, что можно проверить, набрав:

```
# cat /proc/filesystems
```

Владельцам Red Hat и Fedora Core можно сразу идти на страницу linux-ntfs.sourceforge.net/rpm/index.html, где доступны откомпилированные RPM-пакеты с необходимыми модулями. Выбираете нужный и устанавливаете. После чего даете команду `/sbin/depmod -a` для загрузки модуля. Самая последняя версия драйверов доступна по адресу linux-ntfs.sourceforge.net/snapshots либо через репозиторий Bitkeeper (linux-ntfs.bkbits.net).

Монтирование разделов NTFS

Попробуем смонтировать раздел, но для начала узнаем его название:

```
# fdisk -l | grep -i ntfs
/dev/hda7      3522    3649    1028128+    7
HPFS/NTFS
```

Теперь можем монтировать этот раздел:

```
# mount /dev/hda7 /mnt/temp/ -t ntfs
```

Обратите внимание, что ключ `-t`, указывающий на файловую систему, в данном случае обязателен, иначе система не сможет сама определить ее и выдаст примерно такое сообщение:

```
# mount /dev/hda7 /mnt/temp/
mount: wrong fs type, bad option, bad superblock on /dev/hda9,
or too many mounted file systems
```

Посмотрим на опции, с которыми раздел монтируется по умолчанию; для этого введем:

```
# cat /proc/mounts | grep -i ntfs
/dev/hda7 /mnt/temp ntfs
rw,uid=0,gid=0,mask=0177,dmask=077,nls=koi8-r,errors=continue,mft_zone_multiplier=1 0 0
```

Мы узнали, что раздел смонтирован в режиме чтение/запись. Его можно изменить только на чтение, задав в командной строке опцию `-r` (или `-o ro`). Владелец является `root` (`uid=0`, `gid=0`), опция `errors` указывает, как будет себя вести система при возникновении ошибок. Существуют два варианта — `continue` (продолжает работу) и `recover` (пытается восстановить). В настоящее время поддерживается только замена загрузочного сектора резервным. Опции `mask` и `dmask` задают параметры доступа к файлам и каталогам, соответственно, возможно использование общей опции `umask`, задающей доступ к файлам и каталогам одновременно. NTFS хранит имена в Unicode, но драйвер переводит их в ASCII. Чтобы указать используемый язык, применяются две конструкции: `-o iocharset=`, или в новом варианте `-o nls=`. Для отображения русских имен используется `koi8-r`, возможно задание `utf8` (если ядро не поддерживает Unicode, то дополнительно используйте `utf8=true`). В нашем случае `nls` была выбрана автоматически, потому что при конфигурировании ядра эта кодировка оказалась прописанной по умолчанию (меню «File Systems → Native Language Support → Default NLS Option»). Параметр `mft_zone_multiplier` указывает на размер зарезервированной в `master file table` части, которая содержит информацию о файле. Первоначально он задается при форматировании NTFS, но в процессе эксплуатации может изменяться на лету. Цифра 1 является значением по умолчанию и соответствует 12,5% зарезервированного объема,

2 — 25%, 3 — 37,5%, 4 — 50%. Из всего вышесказанного следует, что полная опция монтирования может быть указана в таком виде (то, что следует после второй `-o`, в большей части не нужно и дано для примера).

```
# mount /dev/hda7 /mnt/temp/ -t ntfs -r -o nls=koi8-r -o
uid=500, gid=user, umask=0222
```

После этого с чтением данных проблем быть не должно, все имена будут отображаться нормально, а о записи читайте выше.

Утилиты для работы с NTFS

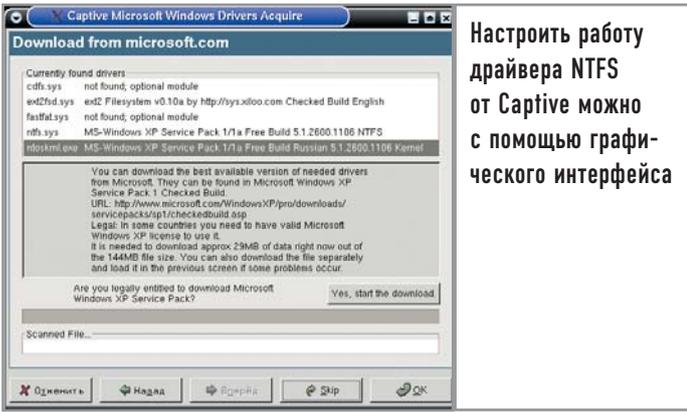
Кроме драйвера проект Linux-NTFS предоставляет также ряд утилит для работы с NTFS под Linux. Большая их часть ориентирована на разработчиков. Все утилиты доступны в пакете `ntfsprogs` (linux-ntfs.sourceforge.net/snapshots/ntfsprogs-200506061652.tar.bz2). После компиляции и установки в системе появятся десять утилит:

- ▶ **mkntfs** предназначена для создания NTFS 1.2 (поддерживается операционными системами Windows NT/2000/XP);
- ▶ **ntfsfix** создана для установки измененных драйвером разделов NTFS — нечто вроде `scandisk`, которая должна использоваться после каждой записи, чтобы предотвратить возможную потерю информации;
- ▶ **ntfsclnt** является аналогом стандартной Unix-утилиты `cat` и предназначена для чтения файлов в разделах NTFS;
- ▶ **ntfsclose** служит для клонирования, копирования, сохранения, создания резервного образа или восстановления раздела с файловой системой NTFS. Работает на уровне секторов диска и сохраняет только используемые данные; неиспользуемые данные заполняются нулями, что позволяет эффективно сжимать полученные образы. Утилита полезна для создания точных копий раздела и восстановления системы. Вот несколько примеров ее применения:

```
# ntfsclose --output system.img /dev/hda1 # создание копии
раздела
# ntfsclose --overwrite /dev/hda1 system.img # восстановление
раздел
# mount -t ntfs -o loop system.img /mnt/ntfsclose # а так
можно заглянуть внутрь созданного образа
```

- ▶ **ntfscluster** — утилита для идентификации файлов в указанном разделе или области NTFS. Работает в трех режимах: `info` (режим по умолчанию, показывает общую информацию об области NTFS), `sector` (отображает список файлов в заданном диапазоне секторов) и `cluster` (делает то же, что и предыдущий режим, только выводит список файлов в группе).

```
# ./ntfscluster /dev/hda7
bytes per sector:      512
bytes per cluster:    4096
sectors per cluster:   8
bytes per volume:     2212564992
sectors per volume:   540177
clusters per volume:  67522
initialized mft records: 56
mft records in use:   40
mft records percentage: 71
bytes of free space:   2207653888
```



Настроить работу драйвера NTFS от Captive можно с помощью графического интерфейса

```
sectors of free space: 4311824
clusters of free space: 538978
percentage free space: 99
bytes of user data: 151552
sectors of user data: 296
clusters of user data: 37
percentage user data: 0
bytes of metadata: 4759552
sectors of metadata: 9296
clusters of metadata: 1162
percentage metadata: 0
```

- ▶ **ntfsinfo** выводит атрибуты по номеру inode или имени файла;
- ▶ **ntfslabel** выводит или устанавливает метку файловой системы NTFS (может содержать до 128 Unicode-знаков);
- ▶ **ntfsls** — аналог Unix-утилиты ls (для Windows — dir) — выводит список файлов в разделе NTFS (монтировать необязательно);
- ▶ **ntfsresize** — а вот это действительно полезная утилита, предназначенная для изменения размера файловой системы NTFS, причем без потерь данных (о том, как работать с ней, мы поговорим чуть позже);
- ▶ **ntfsundelete** восстанавливает удаленные файлы на разделе NTFS; имеет три режима работы: scan (задан по умолчанию, осуществляет просмотр файловой системы на предмет наличия удаленных файлов, при нахождении которых выводит их список), undelete (пытается, насколько это возможно, восстановить утраченные данные, кроме сжатых и зашифрованных файлов) и sору (полезен по большей части при отладке, сохраняет данные MFT в файл):

```
# ./ntfsundelete /dev/hda7
```

Volume is dirty.

Run chkdsk and try again, or use the --force option.

Сообщение «Volume is dirty» может возникнуть после записи в раздел, изменения размера раздела и других возможных операций, связанных с изменением данных. После каждой такой операции во избежание ошибок рекомендуется проверка раздела средствами Windows:

```
# ./ntfsundelete /dev/hda7 --force
```

Volume is dirty.

Forced to continue.

Inode	Flags	%age	Date	Size	Filename
16	F..!	0%	1970-01-01	0	<none>
...					
51	FN..	100%	2005-06-18	5878	test.jpg

Files with potentially recoverable content: 9

Пробуем восстановить один из найденных файлов:

```
$/ntfsundelete /dev/hda7 -s -m test.jpg --force
```

Volume is dirty.

Forced to continue.

Inode	Flags	%age	Date	Size	Filename
51	FN..	100%	2005-06-18	5878	test.jpg

Files with potentially recoverable content: 1

Продолжается разработка еще нескольких утилит. Это ntfswipe, заполняющая нулями свободные части диска; ntfsdefrag — дефрагментатор файлов, каталогов и MFT; для проверки же диска будет использоваться ntfsck. Еще одна утилита — nttools — в перспективе позволит просмотреть, создать, изменить, копировать и находить требуемые значения (эквивалентны командам ntfsccp, ntfsgrep, ntfstouch, ntfsrm, ntfsrmdir). В дальнейших планах разработка утилиты ntfsdiskedit, которая позволит производить операции с дисковыми структурами NTFS.

Проект Captive NTFS

Иным путем пошли разработчики проекта Captive (www.jankratochvil.net/project/captive). Они написали оболочку для драйвера, используемого в Windows, что-то вроде Wine для NTFS. Поддерживаются как чтение, так и запись, правда, работает такой «драйвер» все-таки медленнее, чем предыдущий. Для работы Captive потребуются файл ntfs.sys и системный модуль ядра NT ntoskrnl.exe, которые можно взять с установленной Windows XP. Для защиты от возможного краха используется работа в пространстве пользователя при помощи модуля LUFs — Linux Userland File System (<http://lufs.sourceforge.net/lufs>). Стоит отметить, что Captive работает не со всеми версиями драйверов Windows (полный список проверенных в работе доступен на сайте). Во всяком случае, со всеми файлами, взятыми в русифицированных версиях Windows XP, работать не получилось.

Тогда необходимо будет взять файлы с сайта Microsoft. Но ситуация с их использованием двоякая: с одной стороны, они находятся в свободном доступе, с другой — предназначены для пользователей Windows. Во всех известных нам дистрибутивах, имеющих Captive, их приходится добывать самостоятельно. Для работы их следует положить в каталог /var/lib/captive. Так как драйверы Windows требуют особых привилегий вроде прямого доступа к железу, то полную стопроцентную эмуляцию реализовать не получится, потому что Unix-системы, естественно, будут защищаться от таких процессов. Во время установки создаются новый пользователь и группа captive, от имени которых и будут работать процессы, а сама установка по умолчанию запускается в изолированной среде CORBA sandbox и chroot-окружении. К сожалению, эти ограничения сделали невозможной одновременную работу сразу с несколькими разделами. Сами драйверы доступны как в исходных кодах, так и в перекомпилированном виде со статической линковкой. Работают они одинаково, проблема может возникнуть только с модулем lufs.o, который должен быть собран под определенную версию ядра. После установки для первоначального конфигурирования необходимо запустить утилиту captive-install-acquire,

которая проверит наличие необходимых библиотек и при необходимости закачает все нужное. Если все компоненты готовы к использованию, можно монтировать раздел:

```
# mount -t captive-ntfs /dev/hda7 /mnt/utils/
```

В случае возникновения ошибок вся информация будет доступна в файле `/var/log/messages`:

```
# cat /var/log/messages | grep captive-lufs
```

Если вам требуется автоматическое монтирование при загрузке системы, используйте скрипт `captive-install-fstab` с параметром `-add`, который автоматически добавит в файл `/etc/fstab` используемый вами раздел.

Драйвер от Paragon

Несмотря на то что практически вся информация по этому продукту закрыта, а сам драйвер является коммерческим, разработка от Paragon Software Group является наиболее функциональным решением для работы с NTFS-разделами под Linux, что, в общем, и не должно вызывать удивления, учитывая опыт работы этой компании. Доступен он в двух версиях — персональной и профессиональной. Как и у предыдущих проектов, поддерживаются все версии файловой системы NTFS, сжатые файлы и каталоги, а также жесткие диски объемом до 127 Гбайт. В персональной версии работа возможна только в режиме чтения, а в профессиональной доступна и запись. Демонстрационную версию драйвера (лицензия стоит \$69,95), поддерживающую только чтение, которую производитель разрешает использовать без регистрации в течение 30 дней, можно скачать с сайта www.ntfs-linux.com. Для установки потребуются исходные тексты ядра. Сам процесс очень прост. После распаковки архива запускаете скрипт `install.sh` (в интерактивном режиме — `./install.sh -interactive`, опцией `-iocharset=koi8-r` можно установить кодировку по умолчанию для раздела NTFS).

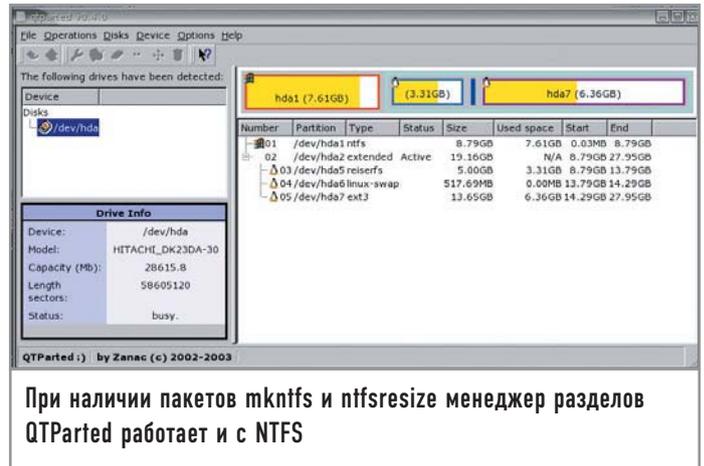
После этого, если не выбрано автоматическое монтирование при загрузке, можно смонтировать раздел вручную:

```
# mount -t ufsd /dev/hda7 /mnt/test_ntfs
# mount -t ufsd -o iocharset=koi8-r /dev/hda7 /mnt/test_ntfs
```

Изменение размеров NTFS в GNU/Linux

Теперь давайте попробуем разобраться, можно ли изменить размер раздела с файловой системой NTFS прямо под GNU/Linux, не прибегая к помощи посторонних утилит вроде Partition Magic? Доступная с 2002 года и входящая в проект Linux-NTFS утилита `ntfsresize` позволяет изменить размер раздела, не разрушив при этом данных. Поддерживаются все версии NTFS. Разработчиками сделано все, чтобы свести риск потери данных к минимуму. Для подстраховки проводятся всевозможные проверки, включая проверку на непротиворечивость данных. При возникновении проблем или каких-либо подозрений утилита отказывается производить изменение размера. При этом следует учесть, что `ntfsresize` не манипулирует размерами разделов, поэтому для начала необходимо воспользоваться утилитой `fdisk`.

Хотя уже имеются графические средства, использующие `ntfsresize`, будет полезно разобраться с работой первоисточни-



ка, поскольку в некоторых дистрибутивах изменить раздел можно только из командной строки. Для работы `ntfsresize` драйвер поддержки NTFS в ядре не нужен, утилита обращается напрямую к диску. Также, если вы хотите использовать ее в спасательной дискете или вам не требуются все утилиты, можно взять статически скомпилированную версию `ntfsresize` по адресу linux-ntfs.sourceforge.net/info/ntfsresize-static-1.9.4.tgz. Теперь давайте посмотрим информацию о NTFS-разделе.

```
# ./ntfsresize -i /dev/hda7
```

```
ntfsresize v1.9.1
```

```
NTFS volume version: 1.2
Cluster size: 1024 bytes
Current volume size: 1052803584 bytes (1053 MB)
Current device size: 1052803584 bytes (1053 MB)
```

```
Checking filesystem consistency ...
```

```
100.00 percent completed
```

```
Accounting clusters ...
```

```
Space in use: 5 MB (0,4%)
```

```
Estimating smallest shrunken size supported ...
```

```
File feature Last used at By inode
$MFT: 1 MB 0
```

```
You might resize at 4526080 bytes or 5 MB (freeing 1048 MB).
```

```
Please make a test run using both the -n and -s options before real resizing!
```

Команда выдала все о разделе NTFS и сообщила, что можно уменьшить раздел вплоть до 5 Мбайт. Но перед реальным изменением желательно прогнать тест.

```
# ./ntfsresize --no-action --size 500M /dev/hda7
```

Если в результате появится сообщение «The read-only test run ended successfully.», можно смело приступать к изменению размера. Если же команда выдала «ERROR:», лучше для начала исправить ошибки: спешка в данном случае ни к чему хорошему не приведет.

```
# ./ntfsresize -s 500M /dev/hda7
```

```
ntfsresize v1.9.1
```

```
NTFS volume version: 1.2
Cluster size: 1024 bytes
Current volume size: 1052803584 bytes (1053 MB)
Current device size: 1052803584 bytes (1053 MB)
New volume size: 499999232 bytes (500 MB)
```

```
Checking filesystem consistency ...
```

```

100.00 percent completed
Accounting clusters ...
Space in use:          5 MB (0,4%)
Needed relocations:   4394 (5 MB)
WARNING: Every sanity check passed and only the DANGEROUS
operations left.
Please make sure all your important data had been backed up in
case of an unexpected failure!
Are you sure you want to proceed (y/[n])? y
# если все данные сохранены, то соглашаемся
Are you sure you want to proceed (y/[n])? y
Schedule chkdsk for NTFS consistency check at Windows
boot time ...
Resetting $LogFile ... (this might take a while)
Relocating needed data ...
100.00 percent completed
Updating $BadClust file ...
Updating $Bitmap file ...
Updating Boot record ...
Syncing device ...
Successfully resized NTFS on device '/dev/hda9'.
You can go on to shrink the device e.g. with 'fdisk'.
IMPORTANT: When recreating the partition, make sure you
1) create it with the same starting disk cylinder
2) create it with the same partition type (usually 7, HPFS/NTFS)
3) do not make it smaller than the new NTFS filesystem size
4) set the bootable flag for the partition if it existed before
Otherwise you may lose your data or can't boot your computer
from the disk!
    
```

Проверить результат преобразования можно командой `./ntfsresize --info --force /dev/hda7`. Как вы сами можете убедиться, утилита свою работу выполнила, но изменила только размер самой файловой системы NTFS, размер же дискового раздела остался прежним (это можно легко проверить, введя команду `fdisk -l | grep -i ntfs`). Теперь же необходимо проделать еще несколько важных шагов. В начале неплохо на всякий случай сохранить MBR:

```
# dd if=/dev/hda of=hda.mbr bs=512 count=1
```

Теперь запустим `fdisk`. Конечно, было бы нагляднее воспользоваться `cfdisk`, но если указать ей новое значение раздела в мегабайтах, то она округлит его до ближайшего значения, которое, вполне возможно, будет меньше требуемой величины, что приведет к невозможности работы с разделом. А вот `fdisk` вполне корректно обрабатывает ситуацию.

```
# fdisk /dev/hda
```

Command (m for help): p

Disk /dev/hda: 30.0 GB, 30020272128 bytes

255 heads, 63 sectors/track, 3649 cylinders

Units = cylinders of 16065 * 512 = 8225280 bytes

Device	Boot	Start	End	Blocks	Id
/dev/hda1	*	1	242	1943833+	
83	Linux				
....					

```
/dev/hda7 3522 3649 1028128+ 7
```

HPFS/NTFS

Partition table entries are not in disk order

Удаляем раздел, на котором размещается NTFS, здесь это 7.
Command (m for help): d
Partition number (1-7): 7
 Создаем на его месте новый раздел размером 500 Мбайт, при этом начальный цилиндр обязательно должен совпадать, то есть, судя по выводу выше, он должен иметь номер 3522:

Command (m for help): n

First cylinder (3522-3649, default 3522): 3522

Using default value 3522

Last cylinder or +size or +sizeM or +sizeK (3522-3649, default 3649): +500M

Command (m for help): t

Partition number (1-7): 7

Устанавливаем тип раздела, для NTFS — 7:

Hex code (type L to list codes): 7

Changed system type of partition 7 to 7 (HPFS/NTFS)

Записываем изменения и выходим:

Command (m for help): w

The partition table has been altered!

Calling ioctl() to re-read partition table.

Если все эти операции были проделаны вами с LiveCD или дискетного дистрибутива, то вы сразу же можете просмотреть информацию о разделе:

```
# ./ntfsresize -i -f /dev/hda7
```

...

Current volume size: 499999232 bytes (500 MB)

Current device size: 509935104 bytes (510 MB)

Если же процесс происходил в работающей с жесткого диска системе, необходимо будет заново перечитать данные, для чего придется перезагрузиться.

Также для нормальной работы обязательно необходимо проверить файловую систему средствами Windows.

Как говорилось выше, уже имеются графические утилиты, в том числе и инсталляторы, позволяющие изменить раздел в наглядном и понятном любому пользователю виде, которые используют код `ntfsresize`. К таким приложениям относятся `DiskDrake` от Mandrakesoft, который может встретиться и в других производных от Mandrake дистрибутивах; а также `YaST` от SUSE, доступный и после установки. Некоторые дистрибутивы используют коммерческие утилиты для разбиения разделов: в `ASPLinux` это `PartitionExpert`, `Xandros` использует `PQDisk`. Нам кажется, что самыми удобными свободными графическими фронтендами, предназначенными для изменения разделов диска, являются `QTParted` (<http://qtparted.sourceforge.net>) и `GParted` (<http://gparted.sourceforge.net>).

Заключение

К сожалению, сам собой напрашивается вывод, что работа GNU/Linux с файловой системой NTFS пока еще далека от идеала, и когда появится достойное решение, удовлетворяющее всех пользователей, сказать пока трудно. Впрочем, процесс медленно, но верно движется вперед. |

Корпоративный каталог

Каталог LDAP (Lightweight Directory Access Protocol) — отличное средство для хранения различной статистической информации, например деловых контактов. Если надо организовать подобную службу во внутренней сети компании, все, что вам понадобится, — это дистрибутив ОС Linux.

Каталог по своей функциональности достаточно сильно напоминает обычную СУБД. Но в отличие от нее предназначен в основном для чтения и быстрого поиска информации внутри базы, а потому не поддерживает такие функции современных СУБД как, например, механизм транзакций или «реляционные» отношения между таблицами. Если продолжать проводить параллели все с той же СУБД, то информация, предоставляемая пользователю службой каталога, больше всего напоминает служебные таблицы-справочники, например «материалов» или «поставщиков». Новые записи в такие таблицы добавляются достаточно редко, а изменение уже имеющейся в них информации выглядит как удаление записи и добавление на ее место новой.

В сетях OSI (Open System Interchange networks) полная модель служб каталога описана стандартом X.500, который подробно рассматривает не только информационную модель хранения данных, но и сам протокол для оперирования ими — DAP (Directory Access Protocol). Однако модель каталога настолько богата, а DAP включает в себя такое количество самых разнообразных способов обработки данных, что его полная реализация на обычном компьютере — задача более чем нетривиальная. В основном именно по этой причине энтузиасты из Мичиганского университета, пользуясь рекомендациями консорциума ISODE, решили заняться разработкой нового, облегченного варианта DAP, который в итоге так и назвали — LDAP (Lite DAP).

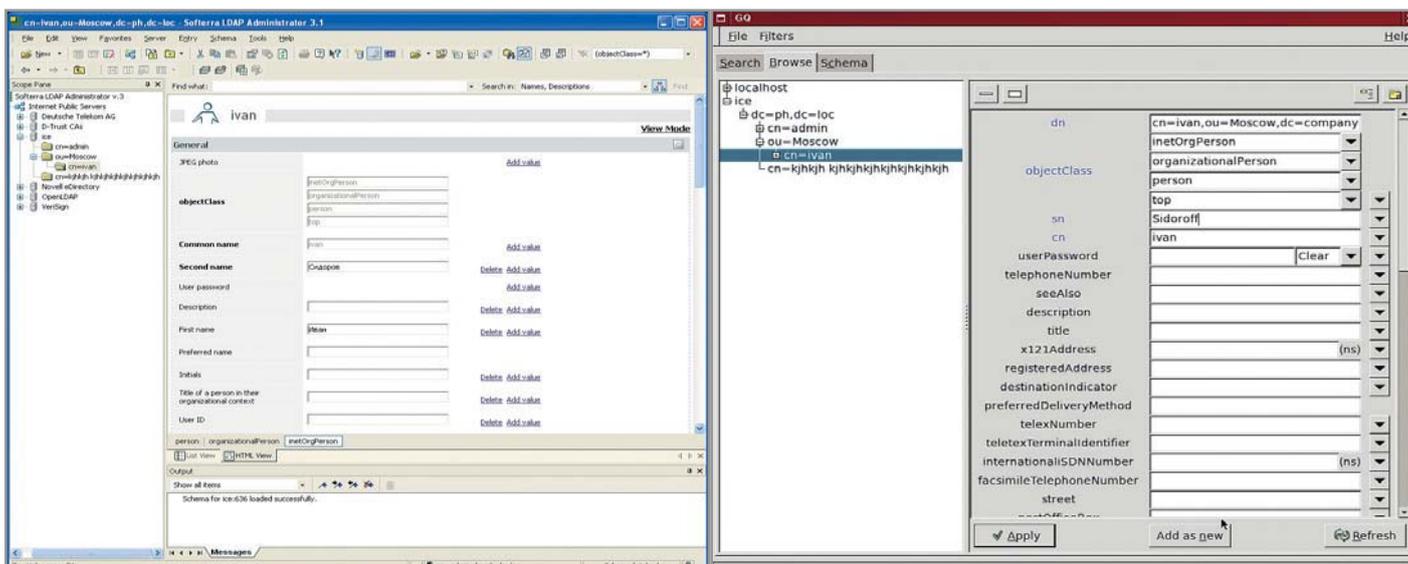
По сути, LDAP заимствовал из X.500 как модель хранения данных, так и протокол для работы с ними. Существенным же отличием от прародителя стало именно то, что LDAP был реализован только лишь для стека TCP/IP, а потому не поддерживал различные экзотические функции оригинального DAP. Еще раз отметим, что LDAP — это протокол, описывающий способ

доступа к информации, которая хранится в каталоге. Системой, которая хранит эту информацию, может быть любая СУБД. Однако как только СУБД начинает поддерживать интерфейс протокола LDAP, она может уже выступать и в роли службы каталога. Забегая немного вперед, можно сказать, что некоторые из служб LDAP-каталогов имеют реализацию интерфейса LDAP для обслуживания клиентов, а все данные хранят в какой-либо распространенной СУБД.

Если взглянуть на каталог изнутри, отдельные записи в нем удобнее всего рассматривать как объекты. Каждая запись в каталоге представляет собой набор атрибутов и их значений. Информация об атрибутах хранится в схеме каталога (directory schema), которая описывает хранящуюся в нем информацию с помощью классов (classes), используя при этом объектно-ориентированный подход.

Класс объекта (objectclass) описывает наименования атрибутов объекта, а также типы их значений. Некоторые атрибуты обязательны (must) для создания записи этого класса, остальные же являются опциональными (may). Кроме того, классы вполне могут наследовать атрибуты других классов. В качестве примера приведем описание структуры абстрактного класса account:

```
account OBJECT-CLASS
SUBCLASS OF top
MUST CONTAIN {
  userid}
MAY CONTAIN {
  description,
  seeAlso,
  localityName,
  organizationName,
```



Вносим новые контакты в каталог через LDAP Administrator и GQ соответственно

**organizationalUnitName,
host}**

Как видите, класс `account` наследуется от класса `top`. Для создания класса `account` должен быть указан необходимый атрибут `userid`, также могут быть обозначены необязательные атрибуты `description`, `seeAlso`, `localityName`.

Одна запись в каталоге может хранить атрибуты нескольких классов объекта. Информация о том, атрибуты каких классов хранит та или иная запись каталога, указывается отдельным, обязательным для каждой записи атрибутом `objectclass`. Другим обязательным атрибутом каждой записи считается ее DN (`distinguished name`). Это уникальный атрибут, который является своего рода ключевым полем и имеет мнемоническое представление, которое необходимо читать справа налево. Например:

DN: Cn=Admin, Ou=IT, dc=Company, dc=com

Формат обмена данными между сервером LDAP и клиентом или же между двумя серверами LDAP называется LDIF (`LDAP Data Interchange Format`). С помощью данного формата можно описывать значения атрибутов для вновь создаваемых объектов или изменения атрибутов для уже существующих. Подводя итог нашей немного затянувшейся вводной части, скажем, что примеры реализации и использования LDAP-совместимых каталогов сейчас можно встретить практически в любой операционной системе. В `Novell Netware` это `NDS` (`Novell Directory Service`). В `Microsoft Windows` — `AD` (`Active Directory`). Ну а в `Unix` — проект `OpenLDAP`.

Адресная книга на LDAP-сервере slapd

Как уже было сказано в самом начале, мы продемонстрируем установку LDAP-сервера на базе платформы `Linux`. Такой выбор был обусловлен сразу двумя факторами — доступностью дистрибутива и необходимых средств, а также относительной легкостью в установке, во всяком случае по сравнению с тем же `Microsoft Active Directory`. Итак, чтобы установить службу каталога, вам понадобится добавить в систему три следующих пакета:

- ▶ `openldap-servers` — сервер `slapd`;
- ▶ `openldap-clients` — клиентские утилиты (`ldapsearch`, `ldapadd`);
- ▶ `openldap` — общие библиотеки для утилит из первых двух пакетов.

После установки в каталоге `/etc/openldap` должны будут появиться следующие конфигурационные файлы:

- ▶ `/etc/openldap/ldap.conf` — настройки клиентских утилит;
- ▶ `/etc/openldap/slapd.conf` — настройки сервера `slapd`;
- ▶ `/etc/openldap/schema/*.schema` — схемы хранения данных.

Конфигурация сервера состоит из двух частей — глобальные директивы и директивы, относящиеся к каждой конкретной базе данных, в которой хранятся записи каталога. Изначально конфигурационный файл `slapd.conf` выглядит следующим образом:

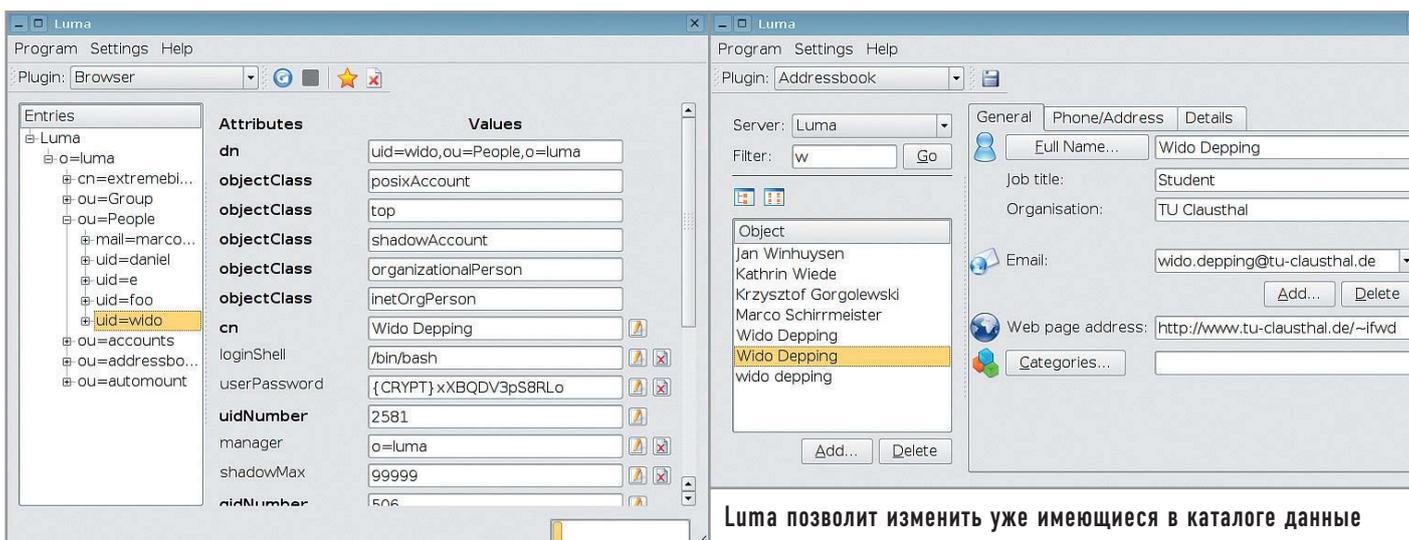
```
# Глобальные директивы
include /etc/openldap/schema/core.schema
include /etc/openldap/schema/cosine.schema
include /etc/openldap/schema/inetorgperson.schema
include /etc/openldap/schema/nis.schema
include /etc/openldap/schema/redhat/rfc822-MailMember.schema
include /etc/openldap/schema/redhat/autofs.schema
include /etc/openldap/schema/redhat/kerberosobject.schema

# Настройки БД на основе LDBM
database ldbm # БД на базе LDBM
suffix «dc=my-domain,dc=com» # Суффикс каталога
rootdn «cn=Manager,dc=my-domain,dc=com» # DN для root
directory /var/lib/ldap # Где хранятся файлы БД
index objectClass,uid,uidNumber,gidNumber,memberUid eq
# Атрибуты, для которых
index cn,mail,surname,givenname eq,subinitial
# строятся индексы
```

В текущую конфигурацию необходимо добавить еще одну директиву:

```
rootpw «secret» # Пароль для root
```

Кроме того, необходимо выбрать суффикс каталога, который был бы созвучен с названием вашей компании. Например:



Luma позволит изменить уже имеющиеся в каталоге данные

suffix «dc=company,dc=ru» # Суффикс каталога

Таким образом, суффикс указывал бы на принадлежность каталога к вашей компании, если бы этот каталог был частью другого, более глобального каталога. Однако, изменяя суффикс, надо не забывать и о смене «rootdn»:

rootdn «cn=admin,dc=company,dc=ru» #DN для root

После этого можно запускать сервер, используя скрипт, находящийся в /etc/init.d:

/etc/init.d/slapd start

После того как стартовал сервер, необходимо сделать первую запись в нашем каталоге, а именно для «dc=company,dc=ru». Для этого мы воспользуемся утилитой ldapadd. Как можно догадаться, эта консольная программа предназначена для добавления записей в каталог. Входная информация для нее должна быть представлена в формате LDIF, поэтому давайте создадим файл first.ldif следующего содержания:

```
dn: dc=company, dc=ru
objectClass: dcObject
objectClass: organization
dc: company
o: Company LTD, Russia
```

Как видите, структура LDIF-файла сама по себе довольно проста: слева указаны наименования атрибутов, справа — их значения. Обязательными являются атрибуты dn и objectClass, указывающие DN вновь создаваемой записи и набор классов, описывающих атрибуты, значения которых должна будет хранить эта запись. Далее должны быть указаны атрибуты, обязательные для каждого выбранного класса. В данном случае это dc — для dcObject и o — для organization. LDIF-файл также может содержать сведения о нескольких записях. Благодаря этому весь имеющийся LDAP-каталог может быть экспортирован в один LDIF-файл. Теперь запишем полученный файл в наш каталог:

```
ldapadd -D 'cn=admin,dc=company,dc=ru' -f first.ldif -W
Enter LDAP Password: secret
```

Опция -D указывает DN подключения к серверу. Чтобы осуществлять запись в каталог, необходимо подключиться с rootdn и паролем rootpw, которые мы указали в файлах конфигурации сервера. Теперь попробуем создать подкаталог для хранения записей адресной книги — файл addressbook.ldif:

```
dn: ou=addressbook, dc=company, dc=ru
```

```
objectClass: organizationalUnit
```

```
ou: addressbook
```

Ну а теперь сделаем первую запись с контактной информацией о конкретной персоне в файле ivan_smirnoff.ldif:

```
dn: cn=Ivan Smirnoff, ou=addressbook, dc=company, dc=ru
```

```
objectClass: person
```

```
objectClass: organizationalPerson
```

```
objectClass: inetOrgPerson
```

```
cn: Ivan Smirnoff
```

```
gn: Ivan
```

```
sn: Smirnoff
```

```
mail: ismirnoff@company.ru
```

```
ou: addressbook
```

```
telephoneNumber: +7-095-123-4567
```

```
mobile: +7-777-1234567
```

Атрибуты mail, ou, telephoneNumber и mobile не являются обязательными. После загрузки файла ivan_smirnoff.ldif мы получим адресную книгу в LDAP-каталоге, правда, пока что состоящую всего лишь из одной записи. Проверить работоспособность полученного каталога можно с помощью утилиты ldapsearch, которая отобразит содержимое каталога все в том же LDIF-формате:

```
ldapsearch -b «dc=company,dc=ru»
```

Если значения параметров в LDIF-файле используют не только латинский алфавит, то они должны быть представлены в кодировке UTF-8.

Настройка Outlook

Для того чтобы воспользоваться нашей адресной книгой, необходимо настроить почтовый клиент таким образом, чтобы он сверял вводимые пользователем адреса электронной почты с данными в LDAP-каталоге. Покажем это на примере наиболее распространенного почтового клиента Outlook Express. Для начала создадим в нем новую учетную запись для «Службы каталогов»: для этого потребуется перейти в меню «Сервис → Учетные записи». Далее на вкладке свойств «Дополнительно» для этой учетной записи необходимо указать базу поиска «ou=addressbook,dc=company,dc=ru» — то есть DN

нашей адресной книги, который при поиске будет являться суффиксом для всех контактов.

Теперь, если при создании нового письма в строке адреса написать лишь начальную часть имени, фамилии или электронного адреса персоны, Outlook Express, произведя поиск по каталогу, автоматически «допишет» адрес или же предложит на выбор несколько вариантов, в случае если результаты поиска окажутся неоднозначными. Аналогичным способом можно настроить и другие почтовые клиенты. Ключевыми параметрами здесь являются IP-адрес сервера и база поиска.

Редактирование каталога

Все существующие приложения для редактирования каталога, построенного на базе сервера OpenLDAP, можно разделить на три типа: консольные, графические и основанные на технологиях WWW/CGI. Пример использования консольных утилит мы демонстрировали немного выше. Однако такое решение может сгодиться только в том случае, если LDIF-файлы генерируются автоматически другим приложением, например на основании записей отдела кадров в корпоративной информационной системе (КИС). При достаточном опыте можно создавать LDIF-файлы и вручную, используя предварительно подготовленные для этого шаблоны. Однако если решение выходит на корпоративный уровень, а обновление информации может лечь на плечи секретариата или отдела кадров, то без хорошего графического клиента обойтись вряд ли получится.

LDAP Administrator 3.1

Наиболее качественно реализованный LDAP-клиент для Windows — LDAP Administrator от компании Softerra (www.ldap-administrator.com). Большое количество функций, удобный интерфейс, наличие мастеров, а также умение работать с LDAP-каталогами корпоративных масштабов делают этот продукт единственным достойным выбором для Windows-платформы.

GQ 0.6.0

Данный клиент создан для платформы X-Windows на основе библиотек GTK+ и является одним из первых GUI-клиентов для LDAP под Unix-системы. Как правило, его можно отыскать в любом дистрибутиве ОС Linux.

GQ обладает достаточной функциональностью для редактирования каталогов среднего по масштабам предприятия, поддерживает русский язык и создание новых записей на основе существующих (замена механизма шаблонов). Однако разработчики проекта или совсем удовлетворились созданным детищем, или попросту про него забыли, но так или иначе последний его релиз датирован ноябрем позапрошлого года.

LUMA 1.6

Это относительно новый и весьма перспективный LDAP-клиент для X-Windows, написанный на языке Python с использованием библиотек `python-ldap` и `python-qt`. Он обладает достаточно приятным и простым в работе интерфейсом, довольно функционален, а кроме всего прочего поддерживает русский язык. Программа имеет серьезные шансы стать оптимальным выбором для небольших компаний.

Заключение

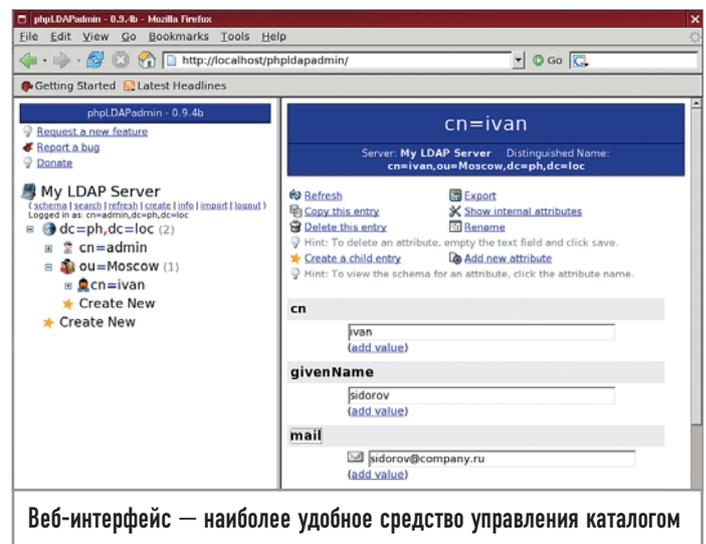
Несомненно, применение каталога в масштабах пусть даже небольшой организации облегчит жизнь не только пользователям, но и системным администраторам. Хотя бы по той простой причине, что LDAP-каталоги довольно легко масштабируемы, а благодаря использованию общего протокола практически не зависят от платформы реализации. Кроме того, при применении встроенных средств для репликации без особого труда можно построить собственную внутреннюю распределенную систему каталогов, которая будет полностью соответствовать территориальным разделениям филиалов вашей компании.

Альтернатива

Среди множества LDAP-клиентов, написанных на различных высокоуровневых языках программирования, отдельно следует выделить те, которые построены на основе технологий WWW/CGI. Приведем лишь несколько наиболее распространенных примеров: *LDAP-ABOOK* (<http://ldap-abook.sf.net>), написанный на Perl; *phpldapadmin* (<http://sf.net/projects/phpldapadmin>), а также *LABE* (www.savoirfairelinux.com/labe), основанные на PHP. К несомненным достоинствам таких клиентов стоит

Управление LDAP через веб-интерфейс

отнести независимость от платформы. То есть редактировать LDAP-каталог вы сможете при помощи любого браузера. С другой стороны, для того чтобы установить такие клиенты, понадобятся как минимум веб-сервер и модуль реализации протокола LDAP для того языка программирования, на котором написан клиент. Однако этот «недостаток» в умелых руках превращается в неоспоримое достоинство, позволяя программисту изменять вид LDAP-каталога под стандарты и нужды компании.



Веб-интерфейс — наиболее удобное средство управления каталогом

Grsecurity — ваша личная охрана

За безопасность нужно платить, а за ее отсутствие — расплачиваться. В этой статье речь будет идти о системе управления доступом, которая является дополнительным барьером для злоумышленника.

Сегодня мы займемся безопасностью вашей Linux. Конечно, она и так считается одной из самых надежных операционных систем, но все же и ее взламывают. Мы рассмотрим одну из самых популярных систем управления доступом — Grsecurity. Почему наш выбор пал именно на нее, ведь Grsecurity — далеко не единственная система управления доступом, существуют также LIDS, SELinux и RSBAC.

Самой строгой из всех перечисленных систем считается SELinux, но она и самая сложная в настройке, поскольку работа с ней подразумевает наличие у администратора определенного багажа знаний. Если вам нужно описание SELinux, в Интернете есть много HOWTO по ее установке на русском языке (<http://dkws.narod.ru/howto.html>).

Про LIDS написано много статей, руководств, ее описание можно встретить во многих книгах по Linux. И это неудивительно: LIDS была, наверное, самой популярной системой безопасности до появления SELinux. В любом случае найти хорошее описание LIDS особого труда не составит. Grsecurity — очень мощная система, но не так хорошо описанная, как LIDS и SELinux. Вот именно поэтому мы ее и рассматриваем.

Может, у вас возник закономерный вопрос: если SELinux — самая строгая система безопасности, то зачем же рассматри-

вать Grsecurity и LIDS? Ведь проще сразу установить SELinux. Конечно, это так, но SELinux — самая строгая только при условии правильной настройки. Если ее недостаточно умело отрегулировать, например, если у администратора не хватает опыта, то толку от этой системы не будет. Напротив, при правильной настройке (которая намного проще) системы Grsecurity и LIDS обеспечивают примерно такой же уровень безопасности, что и SELinux.

Прежде чем приступить к рассмотрению Grsecurity, поговорим о том, что такое вообще система управления доступом и как она должна работать.

| Управление доступом — зачем оно нужно? |

Что же не устраивало разработчиков систем управления доступом в обычной Linux? В этой ОС есть два типа пользователей — администратор и все остальные. Права обычных пользователей можно ограничить и с помощью штатных средств операционной системы. Но что может делать администратор — вы сами знаете. Если злоумышленник завладеет паролем администратора (как он это делает — это уже другой вопрос), то получит полную власть над системой. А вот

если на вашем компьютере установлена система управления доступом, то она не позволит ему сделать ничего, в крайнем случае он сможет произвести только те операции, которые не причинят системе ощутимого вреда.

Теперь вернемся к обычным пользователям. Их права ограничиваются, как правило, только на доступ к файлам. Также можно задать ограничение на использование системных ресурсов — дискового пространства (квоты), процессорного времени, установить максимальное число процессов. И все. Система управления доступом (СУД) может запретить пользователю выполнять кое-какие действия, то есть вы можете задать список программ, с которыми вправе работать тот или иной пользователь, а остальные приложения при попытке их запуска будут блокироваться. Во многих случаях даже обычным пользователям часто предоставляются чрезмерные полномочия. Например, зачем пользователю, который зарегистрировался в системе только для чтения почты, возможность компиляции исходного кода или запуска фоновых демонов? Также СУД решает проблему SUDO (когда отдельным пользователям предоставляются все полномочия root) — администратор может определить для каждого пользователя действия, которые он может выполнять с полномочиями root.

В Unix вы можете использовать одну из следующих моделей:

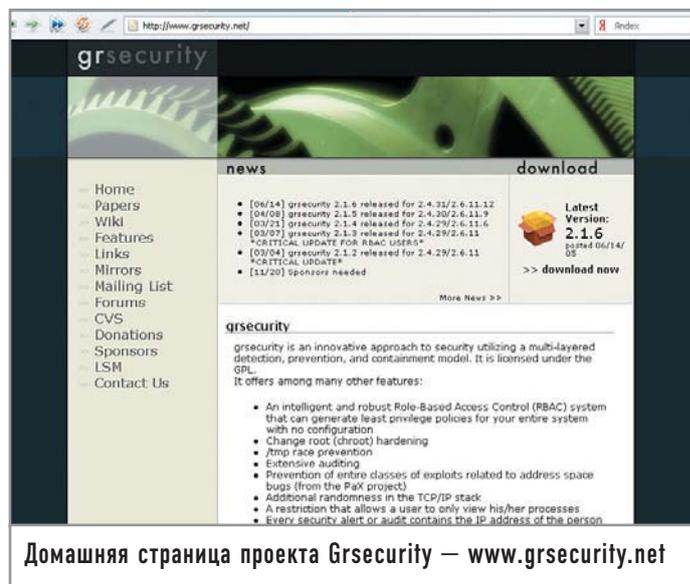
► Дискретное управление доступом (Discretionary Access Control, DAC). С данной системой мы все давно знакомы. Это обычная система пользователей и прав доступа, которая присутствует по умолчанию во всех Unix-системах. Здесь доступ к объекту (например, файлу) регулируется его владельцем с помощью прав доступа. Владелец может контролировать не только тех пользователей, которым положен доступ к объекту, но и режимы доступа (например, это могут быть чтение, запись или выполнение).

► Принудительное управление доступом (Mandatory Access Control, MAC). В этой системе доступ к объекту контролирует не его владелец, а администратор системы. Здесь власть администратора гораздо больше, чем в предыдущей модели. На практике в чистом виде MAC не существует — это только теоретическая модель, но на ее основе строятся некоторые системы управления доступом.

► Модель «тип-домен» (Domain Type Enforcement, DTE) основана на MAC, но с учетом концепции минимальных привилегий: процессу должны быть предоставлены минимально необходимые привилегии. В DTE объекты (файлы) формируют типы, а субъекты (процессы) — домены. Таблица DDT (Domain Definition Table — таблица определения домена) описывает, как домены и типы могут взаимодействовать друг с другом.

Кроме этих трех моделей в Unix часто используются списки управления доступом (Access Control List, ACL), управления доступом на базе ролей (Role-based Access Control, RBAC), возможности и модули защиты (Linux Security Modules, LSM).

С ACL вы наверняка не раз сталкивались при конфигурировании системы. Поэтому перейдем сразу к RBAC. В этой системе всем пользователям назначены одна или несколько ролей, которые он может выполнять в системе. Роль — это действия, которые разрешены пользователю. Например, ад-

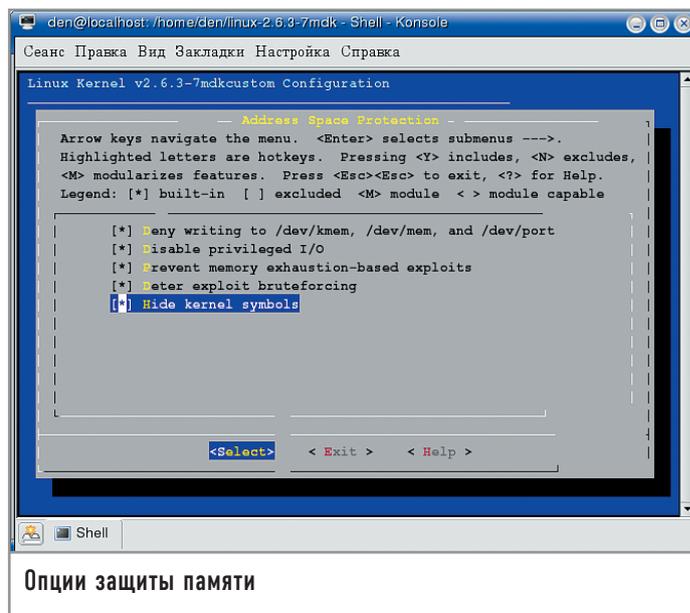
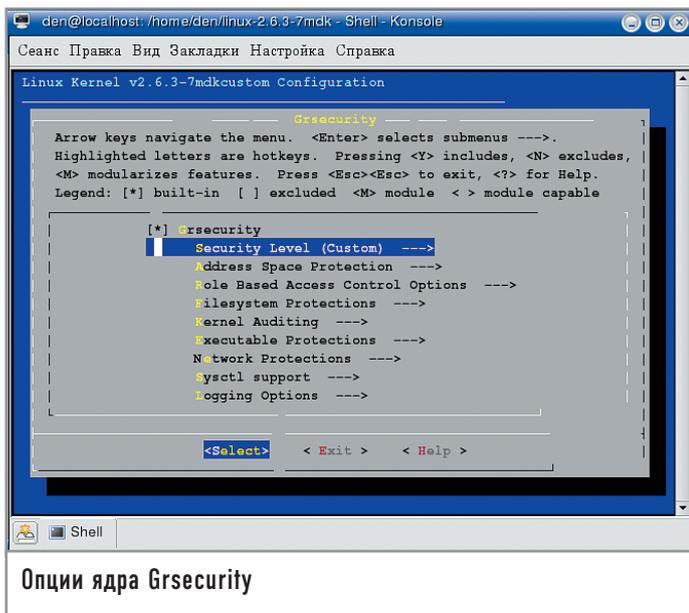


Домашняя страница проекта Grsecurity — www.grsecurity.net

министратор может назначить пользователя, ответственного за интернет-сервер — обычно это веб-мастер. Ему администратор разрешает изменение файла конфигурации, чтение протоколов, запуск, остановку и перезапуск сервера Apache, а также изменение каталога `/var/www/html` (как правило, это DocumentRoot).

Возможности (capabilities) появились давно — начиная с версии ядра 2.1, но мало кто с ними работал. Каждая возможность — это какое-то одно привилегированное действие пользователя root, например `CAP_KILL` — право «убивать» процессы других пользователей. Некоторым пользователям (которые должны выполнять администраторские функции) в зависимости от их нужд назначаются отдельные списки возможностей, то есть определяются действия, которые они могут выполнять от имени root. На данный момент Linux поддерживает 28 возможностей, семь из которых соответствуют POSIX-стандарту. Полный их список приведен в файле `/usr/include/linux/capability.h`.

Модули защиты (LSM) — это сравнительно молодое средство управления доступом. Его история началась с разработки различных систем управления доступом. Все они реализовывались в виде патчей ядра Linux. Но из-за отсутствия централизованной координации в результате получилось, что у каждого проекта защиты был свой патч для ядра, часто не совместимый 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 отметил, что для включения в ядро он будет рассматривать более обширную систему безопасности. Так появился проект LSM (Linux Security Modules). Его цель — предоставить разработчикам систем защиты общий (единый) интерфейс для реализации проектов, основанных на ядре. Реализация такой системы позволит меньше зависеть от ядра и не будет требовать его перекомпиляции, как в случае с патчами. Все LSM-модули используют для взаимодействия с ядром стандартный интерфейс, который не будет изменяться с выходом следующего релиза (не версии) ядра.



Модули защиты LSM — это не система управления доступом, фактически они представляют собой интерфейс, созданный для взаимодействия различных систем управления доступом и ядром. LSM предоставляет только структуру для реализации системы безопасности, воплощением же самой системы занимаются модули. Такой подход снимает проблему несовместимости между патчами отдельных разработчиков: LSM — один для всех, а модули могут быть совершенно разными у каждого разработчика.

В Linux 2.6 поддержка LSM обеспечивается по умолчанию (ее нужно только включить). Для того чтобы это сделать, в меню «Security Options» активируйте опцию «Enable Different Security Models».

Введение в Grsecurity

По понятным причинам мы не сможем рассмотреть все возможности Grsecurity, но надеемся, что приведенной ниже информации вполне достаточно для того, чтобы вы смогли установить и настроить эту систему.

Основная цель Grsecurity — сократить до минимума конфигурацию системы, поскольку сложные системы часто настраиваются неправильно, что приводит к возникновению потенциальных дыр в системе безопасности.

Вот некоторые усовершенствования, вносимые Grsecurity в вашу систему:

- ▶ улучшенное chroot-окружение, не позволяющее процессу выйти за его пределы;
- ▶ OpenBSD-рандомизация TCP ISN (номер последовательности TCP-пакета) и PID (Process ID);
- ▶ ACL (MAC);
- ▶ RBAC;
- ▶ защита стека PaX.

Для установки Grsecurity вам придется перекомпилировать ядро — Grsecurity реализована в виде патча ядра (судя по всему, в ближайшее время Grsecurity поддерживать LSM не будет), поэтому вам нужно загрузить патч для вашего ядра с сайта, расположенного по адресу www.grsecurity.net, пропат-

чить и перекомпилировать ядро. Кстати, пропатчить ядро можно с помощью команды:

```
/usr/src# patch -p0 < имя_файла_патча
```

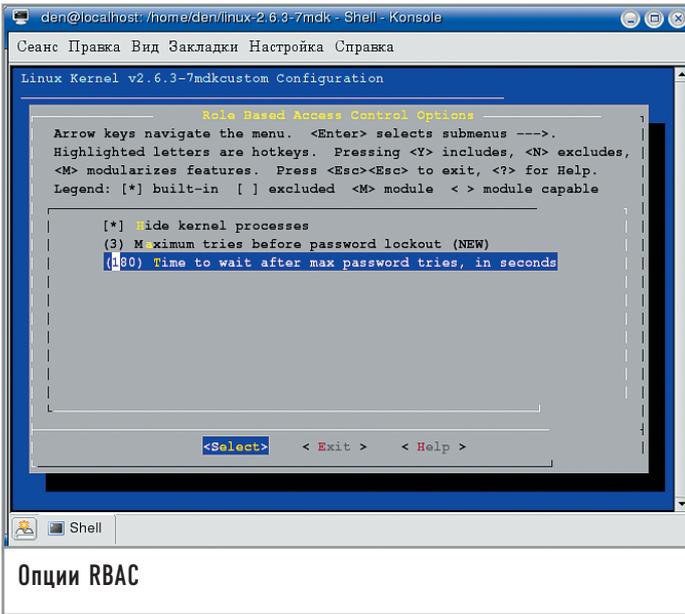
Имя файла патча мы специально не указываем, чтобы вы «по образу и подобию» не скачали патч для ядра автора. Вам нужно скачать патч для своего ядра — номер версии патча должен совпадать с версией ядра.

После применения патча запустите команду `menuconfig` (`make menuconfig`). В меню «Security Options» появятся опции Grsecurity. Включите их и перекомпилируйте как обычно ядро. Не исключено, что на упомянутом сайте не окажется патча именно для вашей версии. Тогда вам нужно будет загрузить с сайта www.kernel.org исходные коды того ядра, которое можно использовать с Grsecurity.

Мы рассмотрим почти все опции ядра Grsecurity, но сначала хотелось бы сделать одно важное замечание о группах пользователей. С помощью групп Grsecurity позволяет управлять ограничениями, которые накладываются на права входящих в группу пользователей. Идентификаторы групп (GID) могут выбираться произвольно, но по умолчанию Grsecurity использует идентификаторы в диапазоне от 1001 до 1005 включительно. Например, для того чтобы запретить пользователям создавать сетевые соединения, может применяться опция ограничения сокетов — по умолчанию для этой возможности используется GID 1004. Пользователи, входящие в группу с GID 1004, не смогут создавать сетевые соединения. Чтобы добавить пользователя в данную группу, необходимо ввести следующую команду:

```
# usermod -G users,1004 den
```

Для быстрой настройки Grsecurity вы можете выбрать один из трех уровней безопасности: низкий (low), средний (medium) и высокий (high). Подсказка подробно объяснит, какие функции будут включены в том или ином случае. Кроме того, можно настроить пользовательский (custom) уровень и установить все опции вручную. Мы, например, пойдем по пути «максимального сопротивления» и выберем именно пользовательский уровень.



Опции RBAC

| Address Space Protection (Защита адресного пространства) |

Опции данной группы используются для защиты памяти. Вы можете включить все опции кроме CONFIG_GKERNSEC_IO, поскольку она не совместима с XFree86.

Deny writing to /dev/kmem, /dev/mem and /dev/port. Символьные устройства /dev/mem и /dev/kmem позволяют пользователям уровня root непосредственно читать и записывать соответственно в системную память и память ядра. Процесс записи сам по себе обычно очень опасен, поскольку злоумышленник может использовать запись для загрузки модулей ядра прямо в память, даже если администратор отключил поддержку загружаемых модулей. Устройство /dev/port предоставляет прямой доступ к портам ввода/вывода, что также нежелательно. Данная опция не совместима с VMWare, поэтому, если вы используете этот эмулятор, не включайте ее.

Disable Privileged I/O. Данная опция позволяет отключить привилегированный ввод/вывод, то есть отключает системные вызовы ioregm() и iopl(). Опция не совместима с XFree86.

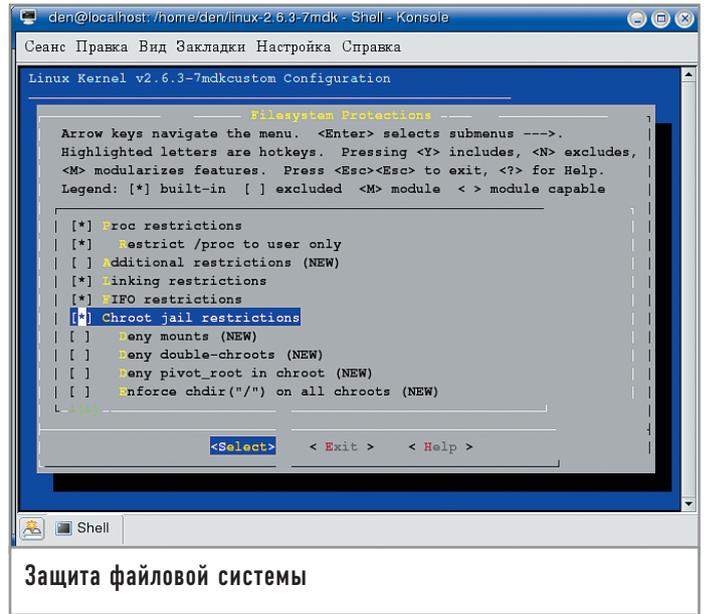
Remove Address from /proc/<pid>/[maps|stat]. Псевдофайловая система /proc предоставляет много информации о запущенных процессах. Злоумышленник может использовать ее в своих целях, поэтому желательно отключить меппинг памяти. При этом вам нужно использовать PaX. О том, что это такое, вы сможете прочитать на сайте <http://pax.grsecurity.net>. Пока не включайте эту опцию. Сделаете это тогда, когда узнаете, что такое PaX.

Hide Kernel Symbols. Скрывает символьную информацию ядра. Пользователи уровня root не смогут получить информацию о загруженных модулях и символах ядра, что позволяет защитить систему от атак памяти (например, переполнения буфера). Включите эту опцию.

| Role Based Access Control (RBAC) Options (Опции RBAC) |

При конфигурировании RBAC доступны следующие опции:

Hide Kernel Processes. Скрывает процессы ядра от обычных пользователей, но не от администратора. Включите эту опцию.



Защита файловой системы

| Filesystem Protection (Защита файловой системы) |

Proc Restrictions. Опция позволяет управлять доступом к /proc двумя способами.

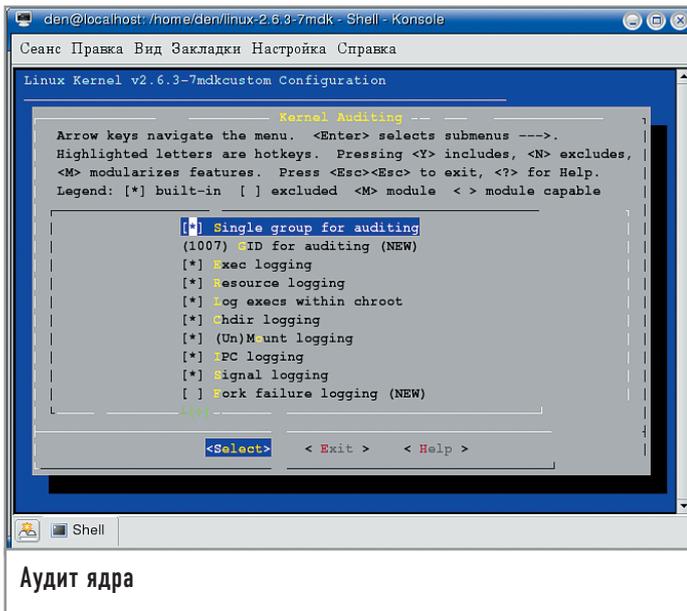
- **Restrict /proc to User Only:** обычные пользователи не смогут просматривать /proc-информацию о процессах других пользователей, а также данные о сети и ядре.

- **Allow Special Group:** создается специальная группа, которой можно просматривать /proc-информацию. GID этой группы должен быть указан в опции «GID for Special Group» (CONFIG_GRKERNSEC_PROC_GID).

Additional Restrictions. Опция добавляет дополнительные ограничения, запрещающие пользователям читать информацию о процессоре и устройствах (запрещает доступ к файлам /proc/cpuinfo и /proc/devices). Включите эту опцию: чем меньше информации окажется у злоумышленника, тем сложнее ему будет взломать систему.

Linking Restrictions. Если вы включите эту опцию, пользователь не сможет создавать жесткие ссылки на файлы, которые ему не принадлежат, а также переходить по символической ссылке, которая принадлежит другому пользователю.

FIFO Restrictions. FIFO (First In, First Out) — специальный тип потока, используемого для обмена данными между процессами. Включение этой опции запретит пользователям делать записи в FIFO из не принадлежащих им каталогов.



Аудит ядра

chroot Restrictions. chroot-окружение используется для организации «песочницы» — отдельной области файловой системы, к которой и только к которой процесс, запущенный непосредственно в «песочнице», может получить доступ. По сравнению с OpenBSD/FreeBSD chroot-окружение Linux довольно слабое (процесс, получивший права root или запущенный с этими правами, может легко выйти за его пределы), поэтому данная опция позволяет его несколько укрепить. Рассматриваемые далее опции помогут создать дополнительные ограничения для chroot-окружения. Мы, в свою очередь, можем только порекомендовать включить все опции, связанные с chroot-окружением.

| Kernel Auditing (Аудит ядра) |

Опции данного меню позволяют протоколировать определенные системные вызовы, например `execve()` и `fork()`.

Single Group for Auditing. Grsecurity может протоколировать действия пользователя (в частности, `exec`, `chdir`, `mount/unmount` и `IPC`). Будут протоколироваться действия не всех пользователей, а только тех, что входят в группу, GID которой указан в опции `GID for Auditing (CONFIG_GRKERNSEC_AUDIT_GID)`. Чтобы протоколировать действия конкретного пользователя, его нужно добавить в эту группу.

Exec Logging. Протоколирует все системные вызовы `execve()`, осуществленные пользовательским процессом. Вы сможете отслеживать все программы, которые запускаются пользователем.

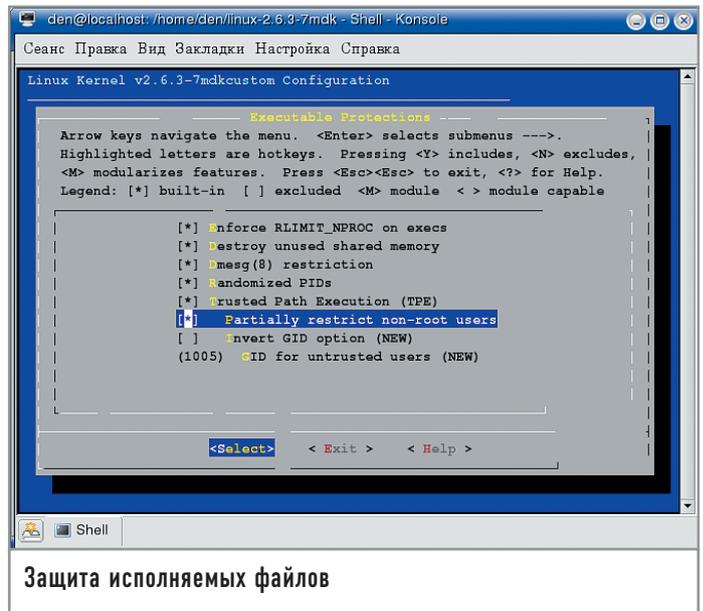
Resource Logging. Протоколирует попытки превысить лимит ресурса, например максимальное число процессов.

Log execs Within chroot. Протоколирует вызовы `execve()` в пределах chroot-окружения.

Chdir Logging. Протоколирует вызовы `chdir()`.

(Un)Mount Logging. Протоколирует монтирование и размонтирование файловых систем.

IPC Logging. Протоколирует разделяемую память, семафоры и очереди сообщений — все это относится к взаимодействию между процессами.



Защита исполняемых файлов

Signal Logging. Данная опция протоколирует поступление важных сигналов, например SIGSERV (segmentation Fault), которые могут сообщать о попытке взлома какого-либо приложения.

Fork Failure Logging. Протоколирует неудачные попытки системного вызова `fork()`. Как правило, вызов `fork()` проваливается, если превышен лимит ресурса. Это может быть признаком атаки типа `fork bomb`, когда злоумышленник вызывает огромное количество `fork`, чтобы помешать системе запустить законный процесс.

Time Change Logging. Протоколирует изменения, происходящие в системных часах.

| Executable Protection (Защита исполняемых файлов) |

Enforce RLIMIT_NPROC on exec. `RLIMIT_NPROC` (конфигурируется через `/etc/limits`) позволяет ограничить ресурсы пользователя. По умолчанию они проверяются только во время вызова `fork()`. Включение данной опции позволяет производить проверку ресурсов во время вызова `execve()`.

Dmesg Restriction. Включение данной опции позволяет выполнять программу `dmesg` только пользователю `root`.

Randomized PIDs. ID процессов будет назначаться не последовательно, а произвольно, что не позволит злоумышленнику предсказать PID демонов и других процессов.

Trusted Path Execution. Данная опция не позволит пользователю случайно запустить исполняемый файл (возможно, троян), записанный каким-либо другим пользователем. То есть ему разрешается выполнение программ только из тех каталогов, владельцем которых является непосредственно пользователь `root`.

| Network Protection (Защита сети) |

Larger Entropy Pools. Данная опция позволяет увеличить в два раза размер пулов, которые используются Grsecurity и многими другими приложениями. Данная опция должна быть обязательно включена.

Truly Random TCP ISN Selection. Если вы включите эту опцию, то стандартный способ генерации ISN, используемый в вашей операционной системе Linux, будет заменен настоящему случайным способом генерации ISN, который используется обычно в OpenBSD.

Randomized IP IDs. При включении данной опции рандомизации подвергнутся ID пакетов, которые передаются системой. По умолчанию Linux просто увеличивает ID каждого следующего пакета.

Randomized TCP Source Port. Опция, отвечающая за то, что генерирование номеров портов исходящих TCP-соединений будет происходить случайным образом.

Randomized RPC XIDs. Рандомизация идентификатора транзакций (XID), который используется протоколом RPC. В Linux значение XID просто увеличивается, что позволяет очень просто предсказать следующий XID.

Sockets Restrictions. С помощью трех подопций можно задать различные ограничения сокетов:

- ▶ **Deny Any Sockets to Group:** после включения данной подопции пользователи указанной группы (вам нужно упомянуть ее GID) не смогут связаться с другими узлами;
- ▶ **Deny Client Sockets to Group:** в этот раз пользователи данной группы не смогут выступать в роли клиентов, следовательно, для них невозможно будет установить соединение с другими узлами сети;
- ▶ **Deny Server Sockets to Group:** наиболее полезно к этой группе отнести пользователей, от имени которых запускаются различные сетевые сервисы, например FTP-сервер или Apache.

Последние две опции особенно сильно отражаются на FTP-сервере: при запрещении клиентских сокетов для передачи может использоваться только пассивный режим, а при запрещении серверных сокетов должен использоваться только активный режим.

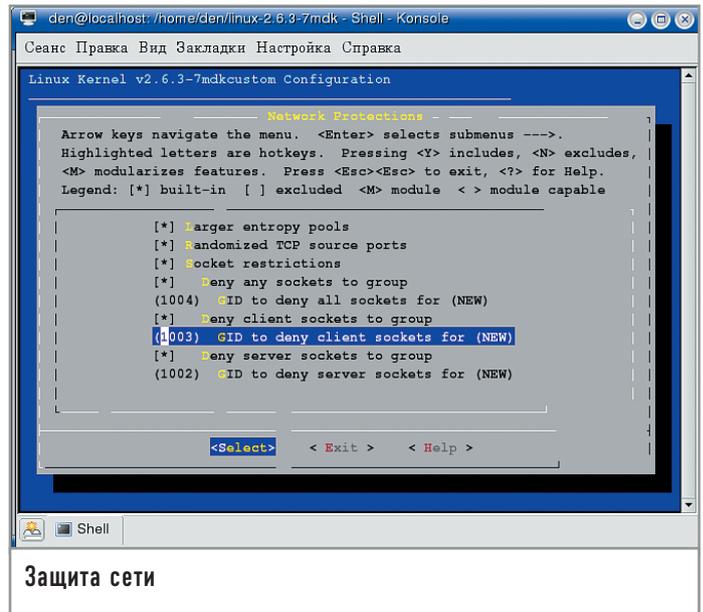
| **Sysctl Support** |

Если данная опция включена, большинство ограничений Grsecurity может быть включено/выключено с помощью файловой системы /proc (/proc/sys/kernel/grsecurity/) без всякой перекомпиляции ядра. Но мы не рекомендуем вам включать эту опцию, поскольку она может стать слабым местом в защите вашей сети.

Конфигурация ядра уже завершена. Теперь перекомпилируйте его, и ваша система будет готова к настройке Grsecurity. Управление доступом реализуется с помощью ACL, но вы не сможете найти опции ядра, связанные с управлением доступом, поскольку конфигурирование ACL осуществляется с помощью утилиты `gradm`. Ее можно скачать по адресу www.grsecurity.net.

| **Структура ACL** |

ACL (список управления доступом) находится в файле `/etc/grsec/acl`. Можно также самостоятельно включать в него дополнительные списки, используя C-стиль подключения заголовочных файлов (`<include>`), но рекомендуется хранить все списки в одном файле.



Защита сети

Структура ACL выглядит следующим образом:

```
Subject <путь к процессу> <необязательные режимы процесса>
{
    <файл объект> <необязательные режимы объекта>
    [+|-] <возможность>
    <имя ресурса> <«мягкий» лимит> <«жесткий» лимит>
    connect {
        <ip>/<сетевая маска>:<порт_от>-<порт_до>
    }
    <тип> <протокол>
}
    bind {
        <ip>/<сетевая маска>:<порт_от>-<порт_до>
    }
    <тип> <протокол>
}
    <имя ресурса> <«мягкий» лимит> <«жесткий» лимит>
}
}
```

Рассмотрим пример ACL для демона печати `cupsd`:

```
subject /usr/sbin/cupsd o {
    / h
    /etc/cups/certs
    /etc/cups/certs/0 wcd
    /etc/group r
    -CAP_ALL
    +CAP_CHOWN
    +CAP_DAC_OVERRIDE
    bind disabled
    connect disabled
}
```

Ограничения накладываются на объекты `/etc/group`, `/etc/cups/certs/0` и `/` путем указания режимов объекта. Для каталога `/etc/cups/certs` режим не задан.

| **Режимы субъекта (процесса)** |

В Grsecurity субъектом называется файл или процесс, к которому применяется ACL (в предыдущем примере — `/usr/sbin/cupsd`). Режимы субъекта позволяют управлять его поведением (см. таблицу 1).

К сожалению, рассмотреть PaX нам не позволяет объем данной статьи, но вы сможете познакомиться с ним на сайте, расположенном по адресу www.grsecurity.net.

| Режимы объектов |

Объектом является файл или каталог. Помните, что если для обозначения режима используется прописная буква, это означает протоколирование, если строчная — определяемое действие.

Для объектов могут использоваться режимы, перечисленные в таблице 2.

| Возможности |

Список доступных возможностей был приведен чуть выше. В дополнение к ним в Grsecurity используется псевдоним CAP_ALL, обозначающий все возможности. Обычно его применяют для запрета всех возможностей, а впоследствии для разрешения необходимых. Например, сейчас мы запретим все возможности, а потом разрешим возможность изменения владельца файлов:

```
-CAL_ALL
+CAP_CHOWN
IP ACL
```

Grsecurity представляет концепцию IP ACL: ограничения IP-адресов, протоколов, типов сокетов, связанных с процессом:

```
connect {
    <ip>/<сетевая маска>:<порт_от>-<порт_до>
    <тип> <протокол>
```

Таблица 1. Режимы субъекта

Режим	Поведение
b	Включает учет процесса
d	Защищает псевдофайлы /proc/<pid>/fd и /proc/<pid>/mem этого процесса
h	Скрытый процесс; его могут «увидеть» только процессы, для которых установлен режим v
k	Может «убивать» процессы, защищенные режимом p
l	Включает режим обучения для процесса
o	Отменяет ACL-наследование (см. дальше)
p	Защищает процесс; его может «убить» только процесс в режиме k или другой процесс, принадлежащий этому субъекту
r	Удаляет ограничения ptrace
v	Может «видеть» скрытые процессы
A	Защищает разделяемую память этого процесса; доступ к разделяемой памяти могут получить только процессы, принадлежащие этому субъекту
C	Если процесс «забил тревогу», он будет «убит». Если с этим процессом связан какой-то IP-адрес, также будут «убиты» все процессы, относящиеся к этому IP-адресу
K	Если процесс «забил тревогу», он будет «убит»
T	Не позволяет процессу выполнить любой троянский код
P	Отключает PaX-функцию PEGEEXEC для этого процесса
S	Отключает SEGMEHEX (PaX)
M	Отключает MPROTECT (PaX)
R	Отключает RANDMMAP (PaX)
G	Отключает EMUTRAP (PaX)
X	Включает RANDEXEC (PaX)

```
}
bind {
    <ip>/<сетевая маска>:<порт_от>-<порт_до>
    <тип> <протокол>
}
```

В обоих случаях синтаксис идентичен: IP-адрес, соответствующая ему маска (если маска не указана, подразумевается /32), после этого следует диапазон портов (младший и старший порты разделяются дефисом). Например, диапазон 0–65 535 используется по умолчанию. Тип сокета может быть следующим: sock, dgram, raw_sock или any_sock. Последним идет протокол. Можно использовать любой протокол, указанный в файле /etc/protocols.

В следующем примере мы ограничим соединения с субъектом в сети 192.168.1.x, разрешим только stream-сокеты в непривилегированном режиме:

```
connect {
    192.168.1.1/24:1025-65535      stream      tcp
}
```

Если же вы не хотите управлять ни bind, ни connect, можно их отключить:

```
bind      disabled
connect   disabled
```

| Ограничение ресурсов |

Ограничение ресурсов процесса задается в следующем виде:

```
<имя ресурса> <«мягкий» лимит> <«жесткий» лимит>
```

С доступными ресурсами вы можете ознакомиться в таблице 3. Почти все эти ресурсы вы уже могли использовать в файле /etc/limits. Для некоторых из них нужно задать время. По умолчанию оно задается в миллисекундах, что не очень удобно. Можно использовать единицы s (секунды), m (минуты) или d (дни). Для ресурсов, где лимит нужно задавать в байтах, удобно использовать единицы K (1000 байт), M (1 000 000 байт) и G (1 000 000 000 байт). Если ограничений не накладывается, нужно указать строку unlimited.

Примеры ограничения ресурса:

```
RES_FSIZE      200M    300M
RES_NPROC      2          3
RES_NOFILE     10         10
```

| Наследование субъекта |

Grsecurity поддерживает наследование ACL для упрощения его самого. Наследование применяется ко всем субъектам, за исключением тех, которые не были специально отмечены флагом o. Лучше всего продемонстрировать наследование на примере:

```
{
    /
    /bin rx
    /etc r
    /tmp rw
    -CAP_ALL
    connect disabled
    bind disabled
}
```

```
/usr/local/bin/someapp {
    /h
    /etc rw
    /var h
    +CAP_SETUID
}
```

При обработке второго ACL Grsecurity проверит существование ACL для каждого компонента пути (в этом примере для /usr/local/bin, /usr/local, /usr и /). В данном случае ACL существует только для субъекта /. Его ACL будет унаследован субъектом /usr/local/bin/someapp.

Предыдущий пример может быть переписан так:

```
/{
    /
    /bin rx
    /etc r
    /tmp rw
    -CAP_ALL
    connect disabled
    bind disabled
}
/usr/local/bin/someapp {
    /h
    /bin rx
    /etc r
    /tmp rw
    -CAP_ALL
    +CAP_SETUID
    connect disabled
    bind disabled
}
```

Возникший конфликт двух ролей (например, в первом случае для /etc используется режим r, а во втором — режим rw) будет решен в пользу субъекта (то есть здесь будет использоваться режим /etc rw).

Роли

Одна из новейших функций Grsecurity — RBAC — предоставляет администратору дополнительный контроль над субъектами. В данной реализации RBAC ключевым понятием является роль: один или более субъектов могут выполнять какую-либо роль — вы можете объявить один объект несколько раз для разных ролей.

Роль описывается следующим образом:

```
role <имя роли> <необязательные режимы роли>
```

Роль default можно использовать, чтобы применить ACL к пользователям, у которых нет роли, например:

```
role default
subject / {
    / r
    /opt rx
    /home rwxcd
    /mnt r
    /dev
    /dev/grsec h
```

```
/dev/urandom r
/dev/random r
-CAP_ALL
connect 192.168.1.0/24:22 stream tcp
bind 0.0.0.0 stream dgram tcp udp
```

```
}
role admin sA
subject /r {
    /rwcdmxi
}
```

В данном примере мы запретили пользователям локальной сети регистрироваться на удаленных машинах с помощью SSH и наложили ограничения на доступ к некоторой части файловой системы. Роль admin перезаписывает субъект /, устанавливая более слабые ограничения.

Для дополнительной безопасности можно также определить роль role_allow_ip, разрешающую соединения с определенным адресом:

```
role_allow_ip <ip>/<сетевая маска>
```

Например:

```
role_allow_ip 192.168.10.1/32
```

Автоматическая генерация ACL

Для автоматического создания файла правил Grsecurity предоставляет режим обучения. Даже если вы предпочитаете создавать файлы правил вручную, мы рекомендуем использо-

Таблица 2. Режимы управления объектами

Режим	Описание
a	Объект может быть открыт для добавления информации. Информация помещается в конце объекта
c	Разрешает объекту создавать каталоги
d	Разрешает удаление каталога
h	Скрывает объект
i	Если устанавливается на исполняемый файл — ACL субъекта будет унаследован во время выполнения этого файла
m	Объекту можно создавать SUID/SGID-файлы
p	Нельзя использовать ptrace для этого объекта
r	Объект может быть открыт для чтения
s	Не протолировать отклоненные попытки доступа к объекту
t	Объект можно просмотреть с помощью ptrace, но нельзя изменить
w	Объект может быть открыт для записи или добавления данных
x	Разрешает выполнение объекта
A	Протолирование успешного добавления информации в объект
C	Протолирование успешного создания каталога
D	Протолирование успешного удаления каталога
I	Протолирование успешного наследования ACL
R	Протолирование успешного чтения данных
M	Протолирование успешного создания SUID/SGID-файла
W	Протолирование успешной записи
X	Протолирование успешного выполнения

Таблица 3. Доступные ресурсы

Режим	Описание
RES_AS	Ограничение адресного пространства (в байтах)
RES_CORE	Максимальный размер файла core (в байтах)
RES_CPU	Максимальное процессорное время (в мс)
RES_DATA	Максимальный размер данных (в байтах)
RES_FSIZE	Максимальный размер файла (в байтах)
RES_LOCKS	Максимальное число блокировок файлов
RES_MEMLOCK	Максимальное число блокировок памяти (в байтах)
RES_NOFILE	Максимальное число открытых файлов (минимально допустимое число открытых файлов равно 3 — для STDOUT, STDIN и STDERR)
RES_NPROC	Максимальное число процессов
RES_RSS	Максимальный RSS (в байтах)
RES_STACK	Максимальный размер стека (в байтах)

вать режим обучения — он послужит фундаментом для вашего собственного файла.

Если у вас еще не установлена программа `gradm`, сейчас самое время это сделать (для установки используются команды `./configure`; `make`; `make install`). После того как вы напишете «`make install`», вас попросят ввести пароль, который будет использоваться для администрирования ACL-системы. Не вводите пароль пользователя `root`!

Для запуска автоматического обучения запустите `gradm` со следующими опциями:

```
# gradm -F -L /etc/grsec/learning.log
```

Во время обучающего режима все действия системы будут запротоколированы в файл `/etc/grsec/learning.log`. После завершения обучающего режима `Grsecurity` обработает протокол и на его основании сгенерирует ACL.

В обучающем режиме, так же как и в случае со `systrace`, важно поработать с приложением, которое вы хотите защитить (с субъектом), в самых разных режимах, чтобы система `Grsecurity` запротоколировала абсолютно все действия, которые может выполнять субъект. Например, для веб-сервера вы должны не только запросить HTML-страницу, но и разные сценарии, написанные на языках PHP и Perl. Также не забудьте запустить сценарий, работающий с MySQL. Для веб-браузера вам нужно будет посетить различные сайты, не обделяйте вниманием также HTTPS-ресурсы и сайты со скриптами (JavaScript, Jscript и т. д.).

Во время обучающего режима не производите действий администратора, например запуск и остановку сервиса, модифицирование учетной записи, установку и удаление про-

Таблица 4. Режимы роли

Режим	Описание
A	Роль администратора, различные ограничения, например на использование <code>ptrace</code> , снимаются
g	Роль группы
G	Разрешается использование утилиты <code>gradm</code>
l	Режим обучения роли
N	Для этой роли не требуется аутентификация
s	Специальная роль, для которой не применяются ACL
T	Обучение TPE (Trusted Path Execution)
u	Пользовательская роль

граммного обеспечения, иначе они запишутся в протокол и будут считаться разрешенными.

Обучающий режим должен длиться целый день (также будут запротоколированы задачи `cron`). После этого отключите контроль доступа с помощью следующей команды:

```
# gradm -D
```

Password:

Введите пароль, который вы указали при установке программы. Чтобы `gradm` сгенерировал ACL по полученному протоколу, используйте опцию `-O`:

```
# gradm -F -L /etc/grsec/learning.log -O /etc/grsec/acl
```

```
Beginning full learning 1st pass...done.
```

```
Beginning full learning role reduction...done.
```

```
Beginning full learning 2nd pass...done.
```

```
Beginning full learning subject reduction for user root...done.
```

```
Beginning full learning subject reduction for user den...done.
```

```
Beginning full learning subject reduction for user snmps...done.
```

```
Beginning full learning 2rd pass...done.
```

```
Beginning full learning object reduction for subject /...done.
```

```
Beginning full learning object reduction for subject /bin/bash...done.
```

```
Beginning full learning object reduction for subject /bin/cat...done.
```

```
...
```

```
Beginning full learning object reduction for subject /...done.
```

```
Beginning full learning final pass...done.
```

Создание ACL — довольно утомительная и рутинная задача, поэтому даже опытным пользователям `Grsecurity` мы рекомендуем использовать встроенный режим обучения для создания начального ACL. Далее мы рассмотрим непосредственно создание ACL для демона `sshd`.

После многочасовой работы `Grsecurity` в режиме обучения был получен следующий ACL:

```
subject /usr/sbin/sshd o {
    /                               h
    /bin                             h
    /bin/bash                         x
    /dev                              h
    /dev/log                          rw
    /dev/ptmx                         rw
    /dev/pts                          h
    /dev/pts/1                       rw
    /dev/tty                          rw
    /etc                              r
    /etc/grsec                        h
    /lib                              rx
    /proc                             h
    /proc/1817                       h
    /proc/1819                       h
    /usr                              h
    /usr/lib/libcrack.so.2.7          rx
    /usr/lib/libglib-1.2.so.0.0.10   rx
    /var                              h
    /var/empty/sshd                  h
    /var/log                          h
    /var/log/lastlog                  rw
    /var/log/wtmp                     w
```

```

/var/run/utmp          rw
/root
-CAP_ALL
+CAP_DAC_OVERRIDE
+CAP_SETGID
+CAP_SETUID
+CAP_SYS_CHROOT
+CAP_SYS_TTY_CONFIG
bind 0.0.0.0/32:0 dgram ip
connect 192.168.9.100/32:53 dgram udp
}

```

Задать правила для /proc/<число> мы не можем, поскольку число (PID) будет изменяться при каждом запуске процесса. Поэтому, чтобы удалить записи для /proc/<число>, необходимо написать следующее:

```

/proc

```

Обратите внимание: вы должны снять флаг h с /proc. По аналогии, терминал не всегда будет pts1, поэтому нужно обеспечить rw-доступ (чтение/запись) ко всем /dev/pts:

```

/dev/pts          rw

```

Правило bind сейчас разрешает привязку к любому порту и любому интерфейсу. Учитывая, что мы настраиваем демон SSH, можно явно указать порт — 22:

```

bind 0.0.0.0/32:22 dgram ip

```

Инструкцию connect также необходимо подкорректировать. В ней перечислены адреса компьютеров, с которыми субъект может соединяться (но не те, которые могут соединяться с субъектом!). Наш автоматически сгенерированный ACL содержит правило, разрешающее субъекту соединяться с сервером имен (192.168.9.100). Однако ведь серверов имен может быть несколько. Обычно их два, а наше правило разрешает соединение только с одним сервером — с которым соединялась программа во время режима обучения. Если этот сервер недоступен, программа будет обращаться к другому серверу имен, указанному в конфигурационных файлах системы, но не сможет найти нужную информацию о нем, поскольку второй сервер не указан в ACL. Указать два (или более) сервера можно так:

```

connect {
    10.0.0.1:53 dgram udp
    10.0.0.2:53 dgram udp
}

```

Если серверов больше двух и вам лень указывать их все, можно использовать следующую инструкцию, разрешающую подключение к любому серверу имен (порт 53):

```

connect 0.0.0.0/32:53 dgram udp

```

Не забывайте, что SSH-демону может понадобиться соединение с демоном AUTH (IDENT), работающим по порту 113. В момент обучения такого соединения не произошло, поэтому в автоматически сгенерированном ACL и слова нет о IDENT. Исправим это:

```

connect {
    192.168.0.0/8:113 dgram ip
    10.0.0.0/24:113 dgram ip
}

```

Как и в предыдущем случае, можно разрешить соединение с любым IDENT-сервером:

```

connect 0.0.0.0/32:133 dgram up

```

Окончательный листинг ACL для субъекта /usr/sbin/sshd выглядит следующим образом:

```

subject /usr/sbin/sshd o {
    /                h
    /bin             h
    /bin/bash        x
    /dev             h
    /dev/log         rw
    /dev/ptmx        rw
    /dev/pts         rw
    /dev/tty         rw
    /etc             r
    /etc/grsec       h
    /lib             rx
    /proc            h
}
/usr                h
/usr/lib/libcrack.so.2.7 rx
/usr/lib/libglib-1.2.so.0.0.10 rx
/var                h
/var/empty/sshd
/var/log
/var/log/lastlog   rw
/var/log/wtmp      w
/var/run/utmp      rw
/root
-CAP_ALL
+CAP_DAC_OVERRIDE
+CAP_SETGID
+CAP_SETUID
+CAP_SYS_CHROOT
+CAP_SYS_TTY_CONFIG
bind 0.0.0.0/32:22 dgram ip
connect {
    192.168.9.100/32:53 dgram udp
    192.168.9.200/32:53 dgram udp
    192.168.0.0/16:113 dgram udp
}
}

```

Однако это еще не все. В нашем примере используется режим субъекта o. Можно задать несколько дополнительных режимов, например: h — чтобы скрыть процесс от ненужных глаз (но тогда не забудьте «спрятать» объект /var/run/sshd.pid); A — чтобы защитить разделяемую память процесса; T — для защиты от троянов; X — для активации RANDEXEC. Учитывая новые режимы, первая строка описания субъекта будет выглядеть так:

```

subject /usr/sbin/sshd ohATX {

```

Также желательно ограничить сбои процесса, чтобы их было не более двух в течение часа:

```

RES_CRACH          2          60m

```

Надеюсь, что данная статья поможет вам при настройке Grsecurity. Любые вопросы можете смело направлять по адресу электронной почты: dhsilabs@mail.ru. |

Enigma для жестких дисков

Причины, побуждающие нас к шифрованию данных, бывают очень разными. Личная ли это паранойя (авторский случай) или реальная необходимость — не имеет значения. Важно, что проблему надо решать, чем мы и попробуем заняться.

Разовые решения в стиле шифрования с помощью KGpg мы рассматривать не будем, хотя их удобно использовать, например, для безопасной передачи данных по электронной почте (хотя еще логичнее применять для этого стандартизированный почтовый OpenPGP). То есть своя направленность у них есть, но для серьезных дел они не годятся.

Настоящее применение начинается тогда, когда вам необходимо иметь постоянно зашифрованные файлы, которые будут расшифровываться только при необходимости работы с ними. При этом, конечно, все это должно быть удобным.

Сразу отметим, что на сегодняшний день существует интересная альтернатива описываемому далее методу шифрования — EncFS. Это файловая система, работающая поверх FUSE, однако FUSE появится в Linux только к версии 2.6.14, ядро придется пересобирать, а также устанавливать библио-

теки, сам EncFS... К тому же это означает появление все большего числа слоев, что не сулит ничего хорошего в плане стабильности (хотя конкретных претензий у автора к EncFS нет). Инструменты же для шифрования через dm-crypt (правда, вряд ли все) наверняка уже есть в вашем дистрибутиве, так что остается только ими воспользоваться.

| dm-crypt |

Для начала немного истории. Начиная с Linux 2.4.x и по сей день ядро предоставляет возможность шифрования loopback-устройств (man losetup), которые отображают произвольные файлы как блочные устройства. Это рабочий способ, который использовался длительное время, но у него есть масса недостатков. Во-первых, cryptoloop, исходя из своей архитектуры, далеко не так стабилен, как хотелось бы, а во-вторых, код

cryptoloop поддерживается довольно слабо, поэтому ошибки в нем — явление довольно закономерное.

dm-crypt появился в Linux как результат эксперимента с новой подсистемой device-mapper. Его реализация значительно стабильнее и лучше, нежели cryptoloop.

К сожалению, необходимо отметить один минус старого cryptoloop, унаследованный dm-crypt, — использование режима сцепления блоков CBC (существует альтернатива в виде ECB, но возможность его применения для хоть скольких-нибудь больших разделов даже не рассматривается) с известным инициализационным вектором. При наличии хорошего ключа это не доставляет особых хлопот, только жаль, что более продвинутые режимы работы блочных шифров в Linux пока не реализованы. Впрочем, dm-crypt, в отличие от cryptoloop, позволяет их добавлять без особых проблем, так что будем надеяться, что скоро увидим улучшения (и под скрежет винчестеров будем перешифровывать разделы — CBC уже точно никуда не денется).

Требования

Во-первых, в вашем ядре должна быть включена поддержка «Device Mapper» («Device Drivers → Multi-device support (RAID and LVM) → Device mapper support») и там же «Crypt target». Во-вторых, у вас должен быть установлен пакет пользовательских утилит device-mapper (sources.redhat.com/dm). В-третьих, в вашем ядре должна быть включена поддержка каких-либо блочных алгоритмов шифрования. И «last but not least» (последнее по списку, но не последнее по значению): в /dev/mapper должен находиться файл control. Если вы используете udev — никаких проблем, но в статическом /dev его может и не оказаться, и тогда придется отправиться на поиски архива с новым /dev.

Еще одно, самое главное требование: будьте крайне аккуратны и осторожны, обязательно создайте резервные копии всех важных данных где-нибудь за пределами шифруемого жесткого диска (особенно если собираетесь шифровать именно разделы). При работе с разделами винчестера цена ошибки всегда очень высока.

Два метода

Для dm-crypt существуют два пользовательских приложения/формата работы с шифрованными дисками — старый (и можно честно сказать, что заброшенный) и новый. Старый вырос из простейшей утилиты cryptsetup (www.saout.de/tiki-wiki/tiki-index.php), которая сначала была реализована вообще в shell и лишь затем обрела реализацию на C. Это объясняется очень просто, поскольку фактически, имея на руках инструмент dmsetup, вы уже можете создавать зашифрованные диски, а сам cryptsetup всего лишь облегчит процесс взаимодействия с dmsetup.

Cryptsetup-luks, доступный на сайте <http://luks.endorphin.org>, совместим со старым cryptsetup, но привносит в нашу жизнь новый стандарт формата зашифрованных дисков — LUKS.

Формат LUKS значительно упрощает работу тем, что использует в качестве ключа для шифрования не заданный па-

роль, а случайные данные из /dev/random, позволяет задавать несколько ключей и менять ключи без полной перешифровки носителя. Плюс ко всему LUKS создан с прицелом на стандартизацию, и, судя по всему, он действительно может стать стандартным форматом не только в GNU/Linux, но и в других дистрибутивах.

Подготовка

Cryptsetup работает только с блочными устройствами, поэтому, если вы хотите использовать в качестве носителя файл, вам придется вспомнить, как пользоваться утилитами dd и losetup, выглядит все это приблизительно так:

```
rik-note:~ # dd if=/dev/urandom of=some-file count=100000
100000+0 входных записей
100000+0 выходных записей
rik-note:~ # ls -l some-file
-rw-r--r-- 1 root root 51200000 2005-09-12 23:30 some-file
rik-note:~ # losetup /dev/loop1 ~/some-file
```

Теперь можно работать с /dev/loop1. Кстати говоря, если вы собираетесь шифровать раздел жесткого диска, то его тоже настоятельно рекомендуется предварительно заполнить данными из /dev/urandom, вне зависимости от того, использовался данный раздел ранее или это новый винчестер. Далее мы будем рассматривать шифрование раздела винчестера, что гораздо надежнее, нежели cryptoloop (если есть возможность обойтись без него, сделайте так).

Шифрование в стиле люкс

Итак, переходим к самому вкусному — собственно шифрованию. Для этого в cryptsetup необходимо ввести дополнительные действия, а именно:

- ▶ **luksFormat** <устройство> [<файл ключа>] — создание нового раздела LUKS;
- ▶ **luksOpen** <устройство> <имя> — открытие раздела LUKS, отображение его на <имя>;
- ▶ **luksDelKey** <устройство> <номер ключа> — удаление ключа с заданным номером с раздела LUKS;
- ▶ **luksAddKey** <устройство> [<файл ключа>] — добавление ключа к разделу LUKS;
- ▶ **luksUUID** <устройство> — вывод UUID устройства LUKS;
- ▶ **isLuks** <устройство> — проверка устройства на наличие LUKS;
- ▶ **luksClose** <имя> — закрытие раздела LUKS;
- ▶ **luksDump** <устройство> — вывод полной информации о разделе LUKS.

У cryptsetup также есть несколько полезных опций:

- ▶ **-c** — используемый алгоритм шифрования;
- ▶ **-y** — проверка пароля (при создании пароль будет просто запрошен дважды);
- ▶ **-d** — файл ключа;
- ▶ **-s** — длина мастер-ключа в битах.

Все остальное мало интересно, однако доступно через cryptsetup -help.

В ядре Linux реализована масса алгоритмов шифрования, однако LUKS поддерживает только их часть (впрочем, огра-

ничения формата поддерживаемых шифров более чем достаточно) — AES, Twofish, Serpent, CAST5 и CAST6. Алгоритмы используемого ядра и длину ключей можно узнать по адресу /proc/crypt0 (обратите внимание, что длина ключей там указывается в байтах, и шифры, вынесенные в модули, необходимо сначала подгрузить).

Насчет выбора шифров/длины ключей могу сказать только одно — AES 128 бит. Нет, конечно, если вы хотите, можете выбирать то, что захочется, но на сегодня это самый лучший вариант. Алгоритм Rijndael (основа AES) реализован в ядре для x86 на ассемблере и прекрасно оптимизирован, а 128 бит вам хватит на ближайшие лет сто (если опять же нам на голову не свалятся квантовые компьютеры, которые эту картину значительно подкорректируют). Итак, за дело. Шифровать будем раздел /dev/hda6.

```
rik-note:~ # dd if=/dev/urandom of=/dev/hda6
dd: запись в `/dev/hda6': No space left on device
192718+0 входных записей
192717+0 выходных записей
rik-note:~ # cryptsetup -c aes -s 128 luksFormat /dev/hda6
WARNING!
=====
```

```
This will overwrite data on /dev/hda6 irrevocably.
Are you sure? (Type uppercase yes): YES
Enter LUKS passphrase:
```

Теперь /dev/hda6 готов работать как LUKS-раздел. Стоит отметить, что здесь мы применили парольную защиту, однако ничто не мешает воспользоваться безопасными ключами, сгенерированными /dev/random, которые можно носить с собой, например, на флеш-накопителе. В этом случае делаем примерно следующее:

```
rik-note:~ # dd if=/dev/random of=/media/usb-
flash/hda6_key count=1
0+1 входных записей
0+1 выходных записей
rik-note:~ # cryptsetup -c aes -s 128 luksFormat /dev/hda6
/media/usb-flash/hda6_key
WARNING!
=====
```

```
This will overwrite data on /dev/hda6 irrevocably.
Are you sure? (Type uppercase yes): YES
```

Теперь у нас есть «люксовый» раздел, что дальше? А дальше необходимо отобразить его в виде нормального (расшифрованного) блочного устройства и работать с ним точно так же, как с обычным разделом.

```
rik-note:~ # cryptsetup luksOpen /dev/hda6 hda6
Enter LUKS passphrase:
key slot 0 unlocked.
rik-note:~ # ls -l /dev/mapper/hda6
brw-r----- 1 root root 254, 0 2005-09-13 00:18 /dev/map-
per/hda6
```

Имя раздела, введенное в качестве последнего параметра к luksOpen, становится именем устройства в /dev/mapper.

Если вы использовали вариант с бинарным ключом, стоит поступить следующим образом:

```
rik-note:~ # cryptsetup -d /media/usb-flash/hda6_key
luksOpen /dev/hda6 hda6
key slot 0 unlocked.
```

Теперь мы можем отформатировать раздел /dev/mapper/hda6 в любую удобную файловую систему (здесь выбирайте сами), например mkfs.reiserfs /dev/mapper/hda6, и примонтировать его вполне традиционным способом вроде mount /dev/mapper/hda6 /mnt/hda6/.

Все, теперь можно работать с /mnt/hda6 точно так же, как с любой файловой системой. Начать стоит, конечно, с переноса сверхсекретных сведений.

После того как работа с секретными данными будет завершена, необходимо отмонтировать раздел (umount /mnt/hda6/) и закрыть отображенный раздел LUKS с помощью команды cryptsetup luksClose hda6. Теперь в /dev/mapper уже не будет видно расшифрованного устройства, а /dev/hda6 враги могут читать, пока им не надоест.

Для того чтобы добавить второй ключ к разделу LUKS, необходимо воспользоваться командой cryptsetup luksAddKey /dev/hda6. После этого вам будет предложено ввести один из уже зарегистрированных, а также еще и новый ключи. Забудьте, если вы использовали бинарный ключ, то это надо указать в -d следующим образом:

```
cryptsetup -d /media/usb-flash/hda6_key luksAddKey /dev/hda6
```

| Пожалейте пальчики! |

У такого решения есть несколько неприятных минусов — для работы с cryptsetup необходимы права root, и работать с cryptsetup приходится в консоли.

К сожалению, здесь вытащить из рукава козырь я не могу, но поведаю вам один адрес, по которому эту проблему решают: www.flyn.org/easycrypto/easycrypto.html. Этот проект берет на вооружение уже хорошо известную нам связку HAL+GNOME Volume Manager, чтобы обеспечить красивое решение: как только HAL обнаружит на разделе «файловую систему» LUKS, он сообщит об этом GNOME VM, который сможет спросить у вас пароль. Дальше все вполне понятно: зная пароль, отображается дешифрованный раздел, на нем находится файловая система, которая монтируется на свое место — работает все та же связка HAL и GNOME VM. В рамках этого же проекта разрабатывается и графический инструмент для полноценной работы с разделами LUKS, который поможет существенно упростить этот процесс.

| Еще секретнее! |

Приведенного выше метода, конечно, достаточно для абсолютного большинства случаев применения шифрования. Однако различные соображения могут натолкнуть вас на мысль о шифровании сначала раздела /home, а потом и всей корневой файловой системы (в конце концов, в /etc тоже имеются некоторые интересные данные). И если о первом можно сильно не беспокоиться, каждый раз заходя из консоли с правами root, проводя монтирование и потом уже загрузку нормального профиля, то со вторым так просто ничего не выйдет. Но выход все же есть.

Для того чтобы устроить шифрование корневой файловой системы, необходимо завести отдельный маленький раздел /boot, где будут лежать незашифрованный загрузчик, ядро и initrd — минимум, не представляющий собой никакого интереса для следствия.

Как думаете, что мы забыли? Забыли мы, конечно же, swar, который также надо шифровать, потому что через него данные могут утекать ведрами. Единственное но — сохранять ключ для swar нам ни к чему, мы будем каждый раз использовать первый попавшийся.

Для переноса существующих разделов / и /home придется сначала куда-то их скопировать (все это желательно делать с LiveCD), потом создать на их месте зашифрованные разделы и перенести файлы на новое место.

Итак, действующие лица: hda6 — раздел /boot (предполагается, что он был создан еще при установке дистрибутива), hda7 — /, hda8 — /home.

Если основные действия проводились в параллельной системе, установленной на том же винчестере, можно воспользоваться каким-нибудь LiveCD. Выводы команд пропущены, иначе не хватило бы места:

```
# mount /dev/hda7 /mnt/hda7/
# mount /dev/hda8 /mnt/hda8/
# mkdir /mnt/hda7/etc/cryptokeys
# dd if=/dev/random of=/mnt/hda7/etc/cryptokeys/home_key
count=1
# chmod 400 /mnt/hda7/etc/cryptokeys/home_key
# cp -ax /mnt/hda7/* /media/usb-60/ubuntu-crypt/root/
# cp -ax /mnt/hda8/* /media/usb-60/ubuntu-crypt/home/
# umount /mnt/hda{7,8}
```

Крайне желательно, чтобы создаваемые копии также хранились на защищенных дисках. Информация скопирована, начинаем создание зашифрованных разделов и перенос данных:

```
# dd if=/dev/urandom of=/dev/hda7
# dd if=/dev/urandom of=/dev/hda8
# cryptsetup luksFormat /dev/hda7
# cryptsetup luksFormat /dev/hda8 /media/usb-60/ubuntu-crypt/root/etc/cryptokeys/home_key
# cryptsetup luksOpen /dev/hda7 hda7
# mkfs.ext3 /dev/mapper/hda7
# mount /dev/mapper/hda7 /mnt/hda7/
# cp -a /media/usb-60/ubuntu-crypt/root/* /mnt/hda7/
# rm -fr /media/usb-60/ubuntu-crypt/root/
# cryptsetup -d /mnt/hda7/etc/cryptokeys/home_key luksOpen
/dev/hda8 hda8
# mkfs.reiserfs /dev/mapper/hda8
# mount /dev/mapper/hda8 /mnt/hda8/
# cp -a /media/usb-60/ubuntu-crypt/home/* /mnt/hda8/
# rm -fr /media/usb-60/ubuntu-crypt/home/
# umount /mnt/hda8/
# cryptsetup luksClose hda8
```

Теперь данные находятся на своем месте, осталось решить проблемы загрузки такой конфигурации. Пора сотворить initrd, ваш подход к нему может отличаться от моего:

```
# mount --bind /dev/ /mnt/hda7/dev/
```

```
# chroot /mnt/hda7
# cd
# mkdir ramdisk
# dd if=/dev/zero of=initrd bs=4096 count=2048
# mke2fs -F ./initrd
# mount -o loop ./initrd ramdisk
# cd ramdisk
# mkdir {bin,lib,dev,proc,mnt}
# ln -s bin/sbin
# cp -a /bin/{bash,mount,mkdir} bin/
# cp -a /sbin/modprobe bin/
# ln -s bash bin/sh
# cp -a /usr/sbin/cryptsetup bin/
# cp -a /sbin/pivot_root bin/
# cp -a /dev/hda* dev/
# cp -a /dev/{console,null,tt,device-mapper} dev/
# mkdir dev/mapper
# cp -a /dev/mapper/control dev/mapper/
# ldd bin/* | egrep -o [^/]lib[a-zA-Z]+\.\so\.. | xargs whereis |
egrep -o ./+\.\so[\.\0-9]* | tr " " "\n" | sort | uniq | xargs -i cp -a
{} lib/
# ls -dpl lib/* | egrep -o "\->\ lib.+" | sed "s/> //" | sort | uniq
| xargs whereis | egrep -o ./+\.\so[\.\0-9]* | tr " " "\n" | sort |
uniq | xargs -i cp -a {} lib/
# cp -a /lib/ld-linux.so.2 lib/
# cp -a /usr/lib/libgpg-error.so.0* lib/
Главное тут, чтобы в lib были все необходимые для запуска
приложений библиотеки, пара диких команд эту проблему
вроде бы решает, но не стесняйтесь проверить работоспособность
в chroot: перезагрузка — дело утомительное. Если необходимо
добавить модули — самое время этим заняться; у меня это вылилось
в следующий список действий:
# mkdir -p lib/modules/2.6.10-5-386/
# cp -p /lib/modules/2.6.10-5-386/* lib/modules/2.6.10-5-386/
# cd lib/modules/2.6.10-5-386/
# mkdir -p kernel/drivers/{ide,md}
# mkdir -p kernel/fs/{ext3,jbd}
# mkdir -p kernel/net/unix/
# mkdir -p kernel/arch/i386/crypto/
# cp -a /lib/modules/2.6.10-5-386/kernel/drivers/ide/* kernel/drivers/ide/
# cp -a /lib/modules/2.6.10-5-386/kernel/drivers/md/* kernel/drivers/md/
# cp -a /lib/modules/2.6.10-5-386/kernel/fs/ext3/* kernel/fs/ext3/
# cp -a /lib/modules/2.6.10-5-386/kernel/net/unix/* kernel/net/unix/
# cp -a /lib/modules/2.6.10-5-386/kernel/fs/jbd/* kernel/fs/jbd/
# cp -a /lib/modules/2.6.10-5-386/kernel/arch/i386/crypto/aes-i586.ko
kernel/arch/i386/crypto/
# cd ~/ramdisk
# vim sbin/init
```

Вбиваем в /sbin/init скрипт приблизительно следующего содержания (у каждого пользователя все это может выглядеть сугубо индивидуально, особенно что касается части драйверов; можно скомпилировать это все статически и забыть про модули вообще):

```
#!/bin/sh
export PATH=/bin:/sbin:/usr/bin:/usr/sbin
echo "Loading some modules..."
modprobe ide-core&
modprobe via82cxxx&
modprobe ide-generic&
modprobe ide-disk&
modprobe dm-mod&
modprobe dm-crypt&
modprobe ext3&
modprobe aes-i586&
wait
echo "Mapping root file system LUKS device"
for count in 1 2 3
do
  cryptsetup luksOpen /dev/hda7 rootfs
  if [ $? -eq 0 ]
  then
    break
  else
    if [ "$count" = "3" ]
    then
      echo "Sorry, system halted!"
      exit 0
    fi
  fi
done
echo "Mounting root file system"
/bin/mount -n -t ext3 /dev/mapper/rootfs /mnt
cd /mnt
pivot_root . initrd
exec chroot . /bin/sh <<- EOF >/dev/console 2>&1
umount initrd
blockdev --flushbufs /dev/ram0
exec /sbin/init
EOF
```

Завершаем создание initrd:

```
# chmod 755 sbin/init
# cd
# umount ramdisk
# gzip initrd
# exit
# mount /dev/hda6 /mnt/hda6/
# cp /mnt/hda7/root/initrd.gz /mnt/hda6/
# umount /mnt/hda6
```

Добавляем в menu.lst (конечно же, обязательно при использовании GRUB):

```
title Ubuntu-crypted
kernel (hd0,5)/vmlinuz-2.6.10-5-386 root=/dev/ram0 rw
initrd (hd0,5)/initrd.gz
```

И корректируем fstab (в /mnt/hda7/etc, разумеется), получается нечто вроде:

```
proc /proc proc defaults 0 0
/dev/mapper/rootfs / ext3 defaults 0 1
/dev/hda6 /boot ext3 defaults 0 2
/dev/mapper/homefs /home reiserfs defaults 0 2
/dev/mapper/swapfs none swap sw 0 0
/dev/hdc /media/cd udf,iso9660 ro,user,noauto 0 0
```

Теперь еще один очень важный момент — модификация загрузки, для того чтобы получить swar и /home. Тут могут быть вариации от дистрибутива к дистрибутиву, но в Ubuntu (должно 1:1 переноситься в Debian) я для этого создал два скрипта в /mnt/hda7/etc/init.d/ — cryptofs и uncryptofs. Содержимое cryptofs:

```
#!/bin/sh
cryptsetup -d /etc/cryptokeys/home_key luksOpen /dev/hda8
homefs
head /dev/urandom | cryptsetup -c aes -s 128 -h plain create
swapfs /dev/hda5
mkswap /dev/mapper/swapfs
```

Заметьте, здесь используется старый подход к шифрованию с cryptsetup, преимущества LUKS для swar совершенно излишни. Содержимое uncryptofs:

```
#!/bin/sh
cryptsetup luksClose homefs
cryptsetup remove swapfs
mkswap /dev/hda5
```

На этом компьютере параллельно установлен другой дистрибутив, который не очень будет обрадован мусору у себя в swar, поэтому не будем его огорчать. Далее:

```
# cd /mnt/hda7/etc/init.d
# chmod 755 cryptofs uncryptofs
# cd ../rcS.d/
# ln -s ../init.d/cryptofs S06cryptofs
# cd ../rc0.d/
# ln -s ../init.d/uncryptofs S45uncryptofs
# cp -a S45uncryptofs ../rc6.d/
```

Вот, собственно, и все. Можно демонтировать все разделы и перезагружаться. Конечно, если брать шифрование всей системы, то все не так уж и просто, но ведь и не слишком сложно, как могло бы показаться на первый взгляд! Более того, мы специально провели этот эксперимент на Ubuntu GNU/Linux 5.04 без перекомпиляции ядра.

Как вы смогли убедиться, ядро можно использовать дистрибутивное — никаких проблем, лишь бы все было скомпилировано в модулях.

А в заключение хотелось бы напомнить, что шифрование — всего лишь один из кирпичиков безопасности, решающий сугубо свои и, надо заметить, очень узкие задачи. Поэтому ни в коем случае не думайте, что, зашифровав данные, вы сможете от всего защититься. Гораздо лучше будет обновить систему, настроить брандмауэр, убить ненужные сервисы, и далее в том же духе. Также можно начать и с чтения соответствующей литературы.

И помните: безопасность — дисциплина комплексная. |

Анатолий Лебедев

Великая огненная стена

Для того чтобы обеспечить сетевую безопасность в современных пользовательских Linux-системах, совершенно необязательно вручную прописывать все правила iptables. Большую часть рутинной работы может взять на себя Firestarter.

Массовое внедрение широкополосного доступа в Интернет не может не радовать. Как вы сами прекрасно знаете, поставщиков подобных услуг сегодня существует довольно много. К сожалению, далеко не все из них готовы обеспечить безопасность своим пользователям. Почему это происходит — отдельный вопрос. В некоторых сетях проблемами безопасности заниматься просто некому или, что зачастую более верно, никому не хочется. Подобная ситуация свойственна многим только что образовавшимся домашним локальным сетям, организаторы которых ставят перед собой только одну задачу — ничего не делая заработать как можно больше денег. И выполняют они ее с рвением, достойным всяческих похвал.

Пользователям же, которым не посчастливилось оказаться участниками одной из таких сетей, остается только вспом-

нить известную истину — если хочешь, чтобы что-то было сделано хорошо, то сделай это сам. И, вооружившись свободным пакетом Firestarter, начать стоит собственную крепость, проникнуть в которую злоумышленнику как из локальной, так и из внешней сети будет весьма проблематично. Но перед тем как мы приступим непосредственно к работе, необходимо немного сказать о самой программе. Firestarter — полностью свободное решение, позволяющее, не используя консоль и не изучая достаточно сложный синтаксис системного брандмауэра ядра Linux, задавать политику безопасности для каждого активного сетевого интерфейса.

Установка

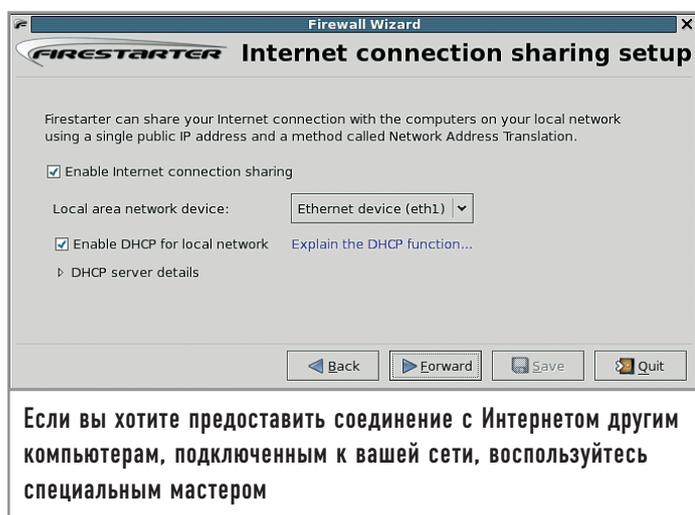
Для начала необходимо скачать дистрибутив программы. Последнюю версию продукта можно найти на сайте разработчиков, который располагается по адресу www.fs-security.com, в разделе «Download».

Как обычно существует несколько вариантов сборки для различных дистрибутивов:

- ▶ Fedora Core
- ▶ Mandrake
- ▶ Debian
- ▶ Slackware
- ▶ Gentoo

Ну и, конечно же, в исходных текстах для тех пользователей, которые любят собирать все программы самостоятельно. Так как наиболее удобным и качественным дистрибутивом на сегодня является Fedora Core, скачаем пакет для него. Установка в этом случае происходит следующим образом:

```
rpm -Uvh firestarter-1.0.1-1.i386.rpm
```



После выполнения этой операции в основном меню GNOME или KDE (в зависимости от того, что вы используете) появится иконка Firestarter. Щелкнув по ней, запустите установленное приложение.

Для работы программе необходимо иметь административные полномочия, поэтому сразу после запуска Firestarter перед вами появится окно с предложением ввести пароль для пользователя root. В случае если пароль верен, запустится специальный мастер, который поможет произвести первоначальное конфигурирование программы. После того как она завершит опрос всех установленных в компьютере сетевых интерфейсов, вам потребуется указать, какой из них для чего используется (внешняя сеть, внутренняя). В этом же диалоговом окне можно настроить NAT (Network Address Translations) для работы всех присутствующих в квартире машин через установленное соединение с сетью, а также задачу IP-адресов через DHCP-сервер.

| Интерфейс |

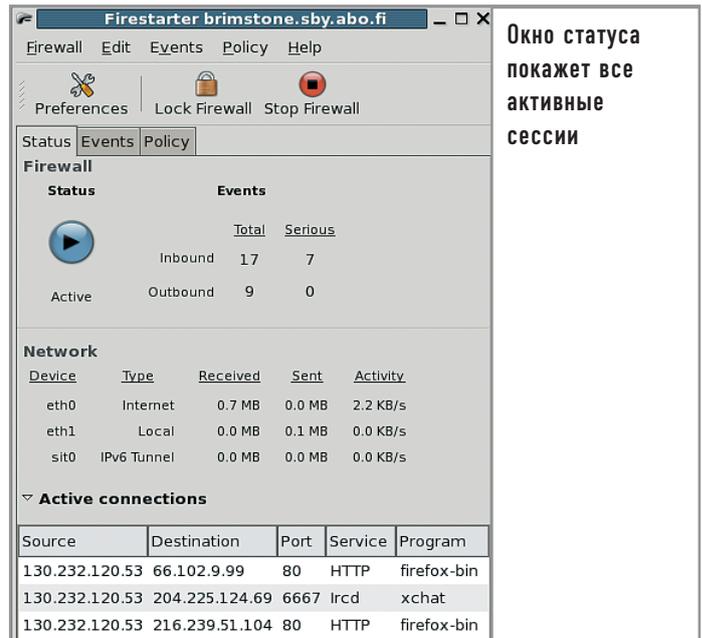
После окончания работы мастера пользователь попадает в главное окно программы. Его содержимое разбито на три основные группы, в каждой из которых располагаются те или иные компоненты.

| Статус |

Первая группа называется «Статус» — в ней отображается список интерфейсов и статистика по ним. Если развернуть скрытые поля под списком интерфейсов, можно будет увидеть все текущие соединения.

| События |

Вторая группа носит название «События» и представляет собой таблицу, в которой отмечены все соединения. События отображаются и обновляются в таблице в режиме реального времени, и по ним доступна вся необходимая информация. Все соединения разделены на три типа, и каждый из них



Окно статуса покажет все активные сессии

отмечается в таблице отдельным цветом. Черным отмечены текущие соединения, серым — заблокированные пакеты. Красным цветом выделены те события, которые могут трактоваться как попытка проникновения к внутренним службам системы и которые требуют к себе особого внимания пользователя.

Если щелкнуть правой клавишей мыши по какому-либо событию, появится контекстное меню, в котором будут предложены действия в зависимости от выбранной политики:

- ▶ разрешить соединения с внешнего адреса;
- ▶ разрешить использовать эту службу для всех;
- ▶ разрешить использовать этот сервис с этого внешнего адреса.

Или же:

- ▶ разрешить соединения с этим внешним адресом;
- ▶ разрешить исходящие соединения на внешний адрес для этого сервиса;
- ▶ разрешить соединения с этого адреса внутренней сети.

iptables

Создание правил вручную

iptables — пакетный фильтр, встроенный в ядра ОС Linux начиная с версии 2.4. Это серьезная система анализа пакетов, позволяющая реализовать все, что может потребоваться от брандмауэра:

- ▶ строить firewall, фильтруя пакеты по адресам;
- ▶ использовать NAT для обеспечения общего доступа в Интернет;
- ▶ использовать NAT для создания прозрачного прокси-сервера;

- ▶ строить систему маршрутизации на основе правил QoS.

Вся работа с правилами обычно выполняется одноименной утилитой **iptables**. Так, например, для того чтобы посмотреть список правил, участвующих в обработке пакетов, потребуется ввести следующую команду:

```
iptables -L
```

Для просмотра правил при работе с NAT необходимо явно указать таблицу:

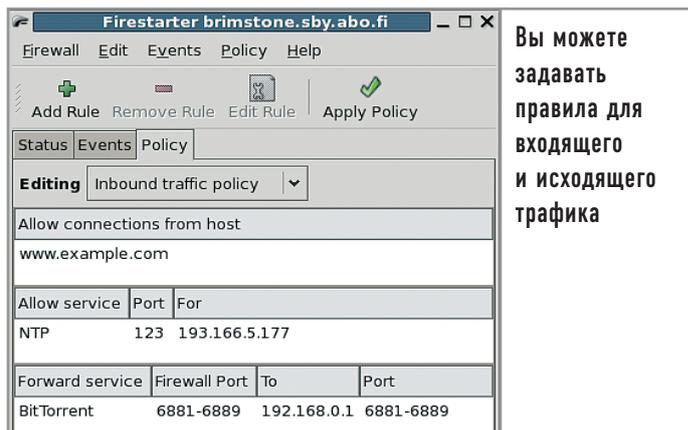
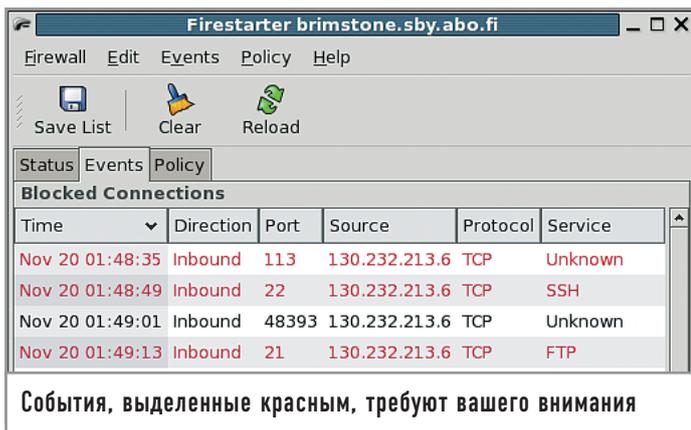
```
iptables -t nat -L
```

Для добавления нового правила используется параметр **-A**:

```
iptables -A INPUT -s 10.11.12.13 -j DROP
```

Данная команда добавляет правило для обработки входящих пакетов с адреса 10.11.12.13 — все пакеты с него будут выкидываться из обработки и не дойдут до получателя. Если вы захотите получить более подробную информацию о работе утилиты **iptables**, то сможете найти ее в интерактивном справоч-

ном руководстве **man**. Кроме того, в Интернете имеется перевод детальной инструкции по построению брандмауэров с использованием **iptables**. В ней вы найдете примеры конфигурационных файлов для решения самого различного рода задач, а также сможете познакомиться со всеми параметрами командной строки. Найти ее можно в Интернете по адресу www.opennet.ru/docs/RUS/iptables/index.html.



Политики

Третья группа позволяет редактировать правила обработки входящего и исходящего трафика. Чуть ниже — список адресов, для которых разрешено прохождение данных. Последняя часть управляет трафиком, проходящим во внутреннюю сеть.

Работа

По своей сути Firestarter — программа для управления firewall, встроенным в ядро Linux, в отличие от большинства решений для ОС Windows, которые полностью выполняют функции фильтра пакетов. Таким образом, при работе с программой мы задаем правила, по которым брандмауэр принимает или отклоняет пакеты. Пакеты в терминологии Firestarter делятся на те, что приходят из внешней сети (Интернет) и из внутренней (LAN). То есть, выбирая ту или иную строку в списке на вкладке «Политика», мы управляем политикой доступа к ресурсам нашей сети из Интернета или же политикой, по которой принимается решение о передаче информации из локальной сети в Интернет.

Для любопытных пользователей разработчиками пакета реализована возможность детально рассмотреть, что же именно создал Firestarter и откуда появляются все эти правила работы с пакетами. В директории `/etc/firestarter` можно найти скрипты, которые вызываются при запуске программы и которые, собственно, создают правила. При желании в эти скрипты можно добавлять дополнительные правила. Например, если в вашей домашней сети гуляет большое количество широкоэвещательных пакетов от Windows-компьютеров на порты 135–139, можно добавить правило, блокирующее их. Это избавит вас от большого количества мусорных сообщений в журнале программы.

При работе в сетях, в которых провайдер не заботится о вашей безопасности, стоит закрыть для доступа все типовые сервисы, которыми вы не пользуетесь. К ним можно отнести SMTP, HTTP, FTP, TELNET, RPC и прочие — они могут быть включены по умолчанию и использоваться другими пользователями сети в своих целях. Кроме того, стоит быть очень внимательным при работе с сервисами, обеспечивающими проксирование трафика. Так, например, открытый на весь мир HTTP-проxy запросто может вас разорить, но если это средство использовать с умом, можно сэкономить некоторую сумму денег за счет кеширования. Подобные при-

меры можно привести в отношении многих других служб. Если для вас это не совсем очевидно, в этом случае лучше довериться мастеру, который предложит использовать общую политику для обеспечения безопасности среднестатистической системы, причем применяемые в ней правила можно считать оптимальными.

Заключение

Использование подобных программ весьма неоднозначно оценивается критиками. Опытные системные администраторы просто не понимают, зачем они нужны. Они привыкли работать в консоли и хорошо себе представляют, как настраивать подобные службы вручную.

Другие говорят, что базовая настройка безопасности должна производиться дистрибьютором операционной системы. Позиция авторов описанной сегодня программы не всегда пользуется поддержкой, но, судя по популярности подобных средств для Windows, она обязательно найдет своих поклонников. Можно лишь пожелать несколько достойных конкурентов этому продукту, так как грамотная конкуренция поможет улучшать и совершенствовать его.

СТРИМ

Обход системы фильтрации

За прошлый год проект компании «MTU Интел» охватил более ста тысяч домашних пользователей. Подключая клиентов посредством ADSL-канала, компания выдает всем реальные адреса. Это весьма удобно, так как всегда можно добраться до своего домашнего компьютера, находясь, например, на работе. Но при этом стоит учесть, что они, проявляя заботу о пользователе, не пропускают пакеты на порты, являющиеся базовыми для большинства сервисов. Так, например, установив FTP-сервер на стан-

дартный порт, вы не сможете получить к нему доступ. Тот же принцип относится к HTTP и многим другим службам. Полный список фильтруемых портов можно получить на странице <http://stream.ru/s-filter>. Если вам необходимы «смотрящие» во внешнюю сеть те или иные службы, почти всегда можно настроить их через нефильтруемый порт. Тот же HTTP-сервер Apache можно заставить работать на порту 8008, используя директиву `Listen 8008` в файле настройки сервера `httpd.conf`.

Незаметный слуга

В предыдущем номере мы рассматривали `udev` и то, как он решает свою задачу — именование устройств, однако в наш динамичный век, когда все должно работать по мановению руки, этого явно недостаточно.

Аппаратура должна работать и при этом никого не утруждать своими настройками, пристройками и подстройками: воткнули USB-флеш — вот вам содержимое, подсоединили фотоаппарат — вот вам снимки, подключили принтер — он уже должен быть доступен для печати в CUPS. Для этого необходимо нечто большее, чем `udev`, — нужен условный «кто-то», кто сидел бы и наблюдал за нашей аппаратурой не только на предмет появления/исчезновения, но и следил за ее состоянием, предоставлял бы информацию о классах устройств в понятном для приложений виде. Именно этим и занимается HAL.

| Что такое HAL? |

HAL расшифровывается как Hardware Abstraction Layer, то есть прослойка абстракции от аппаратуры. Как видно из названия, его основное дело — свести пользовательские приложения с устройствами, которые находятся в системе. Для этого HAL собирает массу информации об устройствах из `hotplug`, `udev`, `sysfs` и своей собственной базы данных об устройствах, которую пользователь может самостоятельно изменять.

Как обстояли дела до его появления? Приложения обращались к ядру (чаще всего посредством библиотек) напрямую и получали информацию об аппаратуре оттуда. Однако ядро не знает всего, а даже когда знает, применяет удобные с точки зрения ядра абстракции — именно так наша фотокамера может оказаться всего лишь еще одним блочным устройством в `/dev`, мало чем отличающимся, например, от винчестера. А ведь это фотокамера, это снимки, разве не было бы приятно увидеть запущенным менеджер фотоальбома сразу после подключения камеры? Еще раз вспомним про `udev` — теперь наши устройства могут называться в `/dev` как угодно, а приложения должны

понимать, что, например, `/dev/my_mega-printer` все-таки является принтером, и на нем можно и нужно печатать.

HAL решает эту проблему ведением базы данных устройств, где все они представлены в виде объектов, каждому из которых присвоен свой уникальный идентификатор. Для любого объекта существует набор свойств в виде пар «ключ-значение», эти свойства собираются с различных уровней представления устройства, но все они могут быть одинаково интересны для приложений. Именно эти пары можно посмотреть в менеджере устройств Fedora Core 3/4 (или просто воспользовавшись командой `lshal`).

Наверное, самый интересный для устройств ключ — `sarabilities`, в котором хранится набор качественных характеристик устройства. Так, например, у всех принтеров в `sarabilities` обязательно будет записано «`printer`», а у всех фотокамер — «`camera`». Тем самым приложения могут спросить HAL о том, какие устройства определенных классов (интересующих приложение) есть в системе и в какие соответствующие файлы `/dev` надо бить челом.

| Принципы работы |

При этом работает HAL на удивление просто: на сегодня он регистрирует скрипты-помощники в `udev` и `hotplug`, которые поставляют информацию об изменениях в аппаратуре. Хотя готовятся изменения, которые не только сделают взаимодействие более правильным (уже не придется работать с `hotplug`, HAL будет просто подхватывать всю информацию, которую уже успел собрать `udev`) и более быстрым (для передачи информации от `udev` к HAL будут использоваться сокеты; тесты с `udevstart` показывают, что время обработки может сократиться в два раза).

Но, конечно, информацией от `udev` и `hotplug HAL` не ограничивается, иначе в чем был бы смысл! Во-первых, HAL самостоятельно исследует доступную информацию для новых устройств в `sysfs`, что заполняет еще сразу несколько свойств, после чего в дело включается мощный механизм файлов информации об устройствах (`fdi`). Путем сопоставления уже полученной разными путями информации с описанными там устройствами HAL может добавлять новые свойства: именно так для карт-ридеров определяется тип поддерживаемых карточек, а для плееров по модели из информации в `sysfs` — данные о поддерживаемых форматах файлов.

Плюс к этому HAL может выполнять активное исследование. Устройства хранения данных, таким образом, проверяются на наличие типов имеющихся файловых систем, ведь эту информацию никак не предоставляет ядро, аналогично диагностируются устройства ввода, дискеты и прочая аппаратура.

Но даже это еще не все, HAL может динамически отслеживать состояние устройств и их свойства, для чего у него имеется механизм вызова `add-ons` (добавлений-включений). Это программы, которые работают весь период существования какого-то типа устройств в системе, соответственно, они запускаются из HAL при подключении первого устройства своего типа и закрываются им же после удаления последнего подобного устройства из системы.

В поставку HAL входит несколько включений: для ACPI, карт-ридеров, а также мышей USB CSR. Они просто выполняют активный опрос устройств и изменяют свойства в случае обновления состояний.

Именно таким образом мы можем узнать о появлении карточки в карт-ридере — сам он об этом ничего не сообщает (конечно, зависит от модели), аналогично выполняется мониторинг параметров ACPI и других. Важно, что любое приложение, запрашивающее информацию у HAL, всегда получает точные и актуальные сведения.

Взаимодействие с приложениями

Сам HAL ничего не навязывает пользовательским приложениям, то есть, куда что монтировать, как что отображать — это не его дело, он всего лишь предоставляет удобную в использовании базу данных устройств. Пользовательские приложения могут работать с механизмом вызовов (`callout`) HAL, для того чтобы реализовать необходимые действия.

Если вернуться к избитому примеру с USB-флеш, то после его обнаружения нам бы хотелось иметь возможность монтировать его, не упражняя пальчики набором команд в консоли. Учитывая ненавязчивость HAL, для этой простой задачи уже есть два решения.

Одно из них применяется в Fedora Core и называется `fstab-sync` — эта программа просто заносит новое устройство в `/etc/fstab`, а дальше работа происходит в стандартном режиме. Второе уходит своими корнями в Debian и зовется `rmount` — это программа, позволяющая непривилегированным пользователям монтировать разделы, не описанные в `fstab`, `rmount` устанавливается как бинарный `suid`. Так что с точки зрения безопасности здесь присутствует какая-то доля риска, хотя у

такого подхода есть удобное преимущество, заключающееся в наличии утилиты `rmount` не только для взаимодействия с HAL; к тому же постоянно изменять `fstab` тоже некрасиво.

В любом случае, здесь уже есть из чего выбрать, а все, что касается вызываемых HAL программ, находится в следующих разделах: `/etc/hal/device.d/` (добавление/удаление устройств), `/etc/hal/capability.d/` (добавление/удаление возможностей устройств), `/etc/hal/property.d/` (добавление/удаление/изменение свойств устройств).

Но этим функционал HAL не ограничивается, он может работать и активным гидом, рассказывая о появлении/исчезновении устройств всем интересующимся приложениям через шину D-BUS. Впрочем, это было бы не слишком интересно в свете наличия `udev` (но стоит помнить, что `udev` не экспортирует информацию на D-BUS, хотя так было в ранних его версиях), поэтому самое интересное заключается в том, что точно таким же способом HAL может сообщать приложениям об изменениях состояний устройств.

Кроме того, HAL доступен и через D-BUS по адресу `/org/freedesktop/Hal/Manager`, описание интерфейсов вместе с исходниками HAL находится в `doc/spec/hal-spec.html`, там же есть пример на Python, использующий этот интерфейс.

Новые возможности

Все это позволяет делать то, что раньше казалось трудно выполнимым, и это легко почувствовать, используя последние версии GNOME — именно эта настольная среда сегодня явно лидирует в освоении всех прелестей работы с HAL. Его используют библиотека для доступа приложений к файловым системам `GnomeVFS`, что позволяет создавать красивый вид по адресу «`computer:///`» (заметьте, что отображаемые иконки там четко соответствуют типу накопителя), апплет `NetworkManager`, который сам разбирается в том, какое сетевое подключение необходимо использовать в конкретный момент времени (владельцы ноутбуков оценят его особенно), а также `GNOME Volume Manager`, позволяющий задавать необходимые действия при появлении разных типов накопителей.

Впрочем, потихоньку HAL проникает и в KDE. Пользователи KDE 3.4.x, возможно, уже знают про новый `ioslave media:/`, предоставляющий доступ к устройствам хранения данных. Актуальность списка устройств и возможность их беспрепятственного монтирования обеспечивается именно HAL (в связке с `rmount` или `fstab-sync`). Даже свежий K3b теперь обновляет список доступных для записи устройств, используя информацию, полученную от HAL.

Кстати говоря, в будущем HAL, возможно, возьмет на себя функцию выбора и загрузки драйверов устройств, а также сможет предупредить пользователя об отсутствии того или иного драйвера в системе (а если продумать дальнейшую интеграцию с пакетным менеджером на предмет установки, будет очень красиво). Поэтому HAL на самом деле — только лишь начало настоящего Plug&Play в GNU/Linux, по мере адаптации к нему различных настольных сред и приложений мы увидим еще много интересного, а аппаратура... просто работает — разве нужно что-то еще? |

Посредник D-BUS

Понятие настольной среды подразумевает нечто цельное и тесно переплетенное между собой, однако среда формируется приложениями, а они как-то должны друг с другом общаться, интегрируясь в нее. В мире KDE для этого давно и небезуспешно используется DCOP, но вот в других настольных средах (например, GNOME) дела все время обстояли не так хорошо.

Конечно, существовала возможность коммуникации посредством CORBA, SOAP или XML-RPC, но первый очень уж тяжеловат и неудобен (причем, надо заметить, что и KDE, и GNOME прошли этап его использования за время своего существования), а оставшиеся не так быстры, как хотелось бы, да и предназначены они для совсем других целей.

До недавнего времени GNOME использовал Bonobo, который, в свою очередь, был основан на CORBA, но в связи с зависимостью от GObject более Bonobo нигде не использовался, да и тяжесть CORBA по-прежнему давила на всю среду. А ведь хотелось бы, чтобы свободно общаться между собой могли не только приложения одной среды, но и разных.

На это и направлен новый стандарт D-BUS, разрабатываемый обществом freedesktop.org. Продукт легковесен и очень быстр в работе, он не зависит от конкретной среды, но при этом прекрасно интегрируется в каждую из них. D-BUS также легко доступен из разных программных сред (Glib, Java (GCJ), Mono, Qt, Python) и прозрачен для сети.

Что такое D-BUS?

D-BUS — это механизм IPC (InterProcess Communication), который предоставляет шину для передачи сообщений, ну а если быть точнее, то сразу несколько шин. Первая и самая главная — системная шина, она создается уже при старте демона D-BUS, и с ее помощью происходит общение различных демонов. Она хорошо защищена от посторонних, и пользовательские приложения, хоть и могут подключаться к ней, все же будут значительно ограничены в том, какие сообщения они смогут туда посылать (в то же время они могут многое «услышать»).

Реальная же рабочая лошадка D-BUS — сессионная шина, создаваемая для любого пользователя, авторизующегося в сис-

теме. Для каждой такой шины запускается отдельная копия демона, и именно посредством нее будут общаться приложения, с которыми работает этот пользователь.

Каждое сообщение D-BUS, передаваемое по шине, имеет своего отправителя и своего получателя, их адреса называются путями объектов, поскольку D-BUS предполагает, что каждое приложение состоит из набора объектов, а сообщения пересылаются не между приложениями, а между объектами этих самых приложений. Для идентификации объектов используются пути, именуемые в стиле Unix. Так, например, сам D-BUS доступен по адресу `/org/freedesktop/DBus`.

Каждый объект может поддерживать один или более интерфейсов, которые представлены здесь в виде именованных групп методов и сигналов — аналогично интерфейсам Glib, Qt или Java. Именуются они привычным для OO-программистов образом, например рассматриваемый D-BUS экспортирует интерфейс `org.freedesktop.DBus`.

Однако объектов и интерфейсов маловато для реализации некоторых интересных возможностей, и D-BUS также предусматривает концепцию сервисов. Сервис — уникальное местоположение приложения на шине. При запуске приложение регистрирует один или несколько сервисов, которыми оно будет владеть до тех пор, пока самостоятельно не освободит, до этого момента никакое другое приложение, претендующее на тот же сервис, занять его не сможет. Именуются сервисы аналогично интерфейсам, а сам D-BUS экспортирует, соответственно, сервис `org.freedesktop.DBus`.

Принципы работы

Сервисы делают доступной еще одну функцию — запуск необходимых приложений в случае поступления сообщений для

них. Для этого должна быть включена автоактивация, и в конфигурации D-BUS за этим сервисом должно быть закреплено одно приложение. Тогда D-BUS сможет его запустить при появлении сообщения.

После закрытия приложения ассоциированные сервисы также разрегистрируются, а D-BUS посылает сигнал о том, что сервис закрыт. Другие приложения могут получать такие сигналы и соответствующим образом реагировать.

После подключения к шине приложение должно указать, какие сообщения оно желает получать, путем добавления масок совпадений (matchers). Маски представляют собой наборы правил для сообщений, которые будут доставляться приложению, фильтрация может основываться на интерфейсах, путях объектов и методах. Таким образом, приложения будут получать только то, что им необходимо, проблемы доставки в этом случае берет на себя D-BUS.

Сообщения в D-BUS бывают четырех видов: вызовы методов, результаты вызовов методов, сигналы и ошибки. Первые предназначены для выполнения методов над объектами, подключенными к D-BUS; посылая такое сообщение, вы выдаете задание объекту, а он после обработки обязан вернуть вам либо результат вызова, либо ошибку через сообщения соответствующих типов. Сигнальные же сообщения, как им и полагается, ничуть не заботятся о том, что и как делается объектами, они вольны воспринимать их как угодно (равно как и не получать их вовсе).

Естественно, для нормальной работы вызовов методов между различными средами/языками D-BUS стандартизует типы передаваемых данных (параметров). Мы не будем их перечислять, поскольку это не имеет смысла, однако отметим, что интерфейсы D-BUS строго типизированы, и D-BUS всегда выполняет проверку корректности передаваемых по интерфейсу данных, не доверяя никому.

Системная шина D-BUS защищена от вредных приложений. Каким же образом это реализовано? Ответ располагается по адресу /etc/dbus-1/system.d/ — там можно обнаружить различные конфигурационные файлы, с помощью которых осуществляется очень гибкая настройка доступа к тем или иным сервисам и интерфейсам. Впрочем, если этой гибкости мало, можно сделать еще больше, подключив возможность использования контекстов безопасности SELinux. В базовый протокол D-BUS также встроены методы аутентификации клиентов, обращающихся с вызовами, однако они практически не используются.

Пример работы

Посмотрите на следующий пример использования механизма D-BUS из Python: код взят из документации к другому любимому нами компоненту — HAL:

```
#!/usr/bin/python
import gtk
import dbus
def device_added(interface, signal_name, service, path, message):
    [udi] = message.get_args_list ()
    print 'Device %s was added'%udi
```

```
def device_removed(interface, signal_name, service, path, message):
    [udi] = message.get_args_list ()
    print 'Device %s was removed'%udi
bus = dbus.Bus (dbus.Bus.TYPE_SYSTEM)
hal_service = bus.get_service ('org.freedesktop.Hal')
hal_manager = hal_service.get_object
('/org/freedesktop/Hal/Manager',
'org.freedesktop.Hal.Manager')
bus.add_signal_receiver (device_added,
'DeviceAdded',
'org.freedesktop.Hal.Manager',
'org.freedesktop.Hal',
'/org/freedesktop/Hal/Manager')
bus.add_signal_receiver (device_removed,
'DeviceRemoved',
'org.freedesktop.Hal.Manager',
'org.freedesktop.Hal',
'/org/freedesktop/Hal/Manager')
gtk.main()
```

Запустите такой скрипт и поиграйте с устройствами — подключите/отключите мышку, принтер, сканер, еще что-нибудь. Все эти действия будут сопровождаться сообщениями от скрипта, поскольку он ловит сигналы device_added и device_removed от HAL через D-BUS (на системной шине). Согласитесь, отловить эти сообщения совсем несложно, зато представьте, сколько всего полезного можно сделать (если чуть-чуть расширить скрипт на предмет более подробных расспросов HAL об устройстве)!

Применения

Разработчики KDE пока еще только начали осваивать D-BUS, что, в общем-то, и понятно: DCOP не намного хуже D-BUS (надо отметить, что последний учел опыт первого в разработке), но с ним KDE работает уже давно. Однако есть проект перевода DCOP на использование D-BUS в качестве транспорта, да и, как уже было сказано, освоение уже пошло (все, что относится к взаимодействию с HAL в KDE, также протекает посредством D-BUS).

А вот GNOME уже активно использует D-BUS в своих приложениях, одни примеры его применения вместе с HAL многого стоят (что интересно, HAL без D-BUS был бы довольно скучен, но и D-BUS смог по-настоящему показать свои прелести только на HAL). Однако этим его использование отнюдь не ограничивается — во-первых, GNOME сейчас активно избавляется от Bonobo, и здесь D-BUS начинает тихо и без лишнего шума выполнять свою основную работу, пусть и не с первого взгляда заметную пользователю.

Во-вторых, наличие удобного транспорта уже навело некоторых разработчиков на мысли о том, что с его помощью можно легко обобщать многие части общего кода приложений, создавая сервисы D-BUS, которые фактически становятся очень интересной альтернативой обычным разделяемым библиотекам, которые, в свою очередь, будут доступны из любых языков и сред. Поэтому, как и в случае с HAL, все самое интересное еще впереди. |

Фотоальбом про запас

Язык программирования Python благодаря ясному и продуманному синтаксису помогает легко и непринужденно решать проблемы системного администрирования и практические задачи, оставляя свободное время. Это позволяет быть в хорошем настроении (ведь все получается быстро) и уделять внимание окружающим.

Если программирование — ваш хлеб насущный, то скорость кодирования позитивно скажется на доходах. А в моем случае небольшой скрипт на Python значительно улучшил мою, так сказать, семейную жизнь. Позвольте рассказать об этом подробнее.

Ради девушки своей мечты, я, как и многие мужчины, был готов к проявлению самых лучших творческих способностей. Но я не умею ни петь, ни рисовать, ни играть на музыкальных инструментах. Только компьютер у меня всегда под рукой и Python, конечно. И это помогло! Я создал скрипт для резервирования фотоколлекции на Gmail. Так программист выиграл у диджея! Ура!

В данной статье мы не просто подробно разберем написание программы, но и рассмотрим весь процесс поэтапно — от постановки задачи до готового решения. Более того, после небольших изменений вы сможете использовать этот скрипт для создания резервных копий как своих, так и пользовательских данных. И пока компьютер работает, оглянитесь — вокруг много красивых девушек, ждущих вашего внимания. Важное замечание: не забудьте купить цветы!

Вводные данные

Самостоятельно сделанные фотографии — самое ценное, кроме сногшибательной внешности, что есть у (теперь уже) моей девушки. По последним подсчетам, их около 1800 штук. Все снимки хранятся на винчестере, который может уничтожить «сокровища» в любой момент. Но это можно предотвратить.

Наша цель — написать такую программу, которая будет заниматься проверкой всех файлов в каталоге и отправлять новые или измененные файлы заданных типов по почте на

Gmail. Плюс, конечно, неплохо было бы научиться чему-нибудь новому, а также получить компоненты (модули) для повторного использования. Вообще, один из самых эффективных способов реализовать задуманное, в том числе и в программировании, — четко представлять конечный результат и промежуточные шаги, ведущие к цели.

Для нашей программы итог представлен в виде четырех файлов. Непосредственно ее текст находится в файле `jpg2gmail.py`. Входные данные — рабочий каталог, настройки сети, почтовый адрес и что-нибудь еще, что может понадобиться, — в файле `jpg2gmail.ini`. В `jpg2gmail.db` мы станем хранить данные об отправленных файлах. Тогда `jpg2gmail.log` будет содержать журнал работы нашей программы.

Мы можем использовать наш скрипт в качестве импортируемого модуля таким образом:

```
>>> import jpg2gmail
>>> print jpg2gmail.test() # выдаст количество файлов для
отправки
>>> jpg2gmail.upload(5) # отправит пять файлов
```

Или вот так из командной строки:

```
python jpg2gmail.py test
# распечатает количество файлов, готовых к отправке
python jpg2gmail.py upload 10
# отправит десять файлов
```

Предварительная подготовка

Для хранения параметров создадим файл `jpg2gmail.ini` следующего содержания:

```
[main]
# адрес почтового ящика хранилища
destination_address = your.name@gmail.com
# адрес, присвоенный провайдером
from_address = myname@provider.com
# SMTP-сервер провайдера
smtp_server = smtp.your.provider.com
# директория с нужными файлами
work_dir = /home/tanya/images
```

Теперь разобьем нашу задачу на несколько мелких и прямо из интерпретатора Python найдем лучший способ их решить. Во-первых, как нам получить вводные данные?

```
>>> from ConfigParser import ConfigParser
>>> config = ConfigParser()
>>> config.read("jpg2gmail.ini")
>>> destination_address = config.get('main' , 'destination_address')
    А как обойти все файлы в каталоге?
>>> import os                # импортируем стандартный модуль
>>> print os.walk.__doc__    # читаем документацию к нужному
методу и ...
# ... узнаем, что, для того чтобы получить список всех файлов в
директории, нужно сделать так:
>>> result = []
>>> for dirpath, dirnames, filenames in
os.walk("/home/tanya/images"):
...     for file_name in filenames:
...         result.append(os.path.join(dirpath,
file_name))
>>>
```

Где хранить сведения об отправленных файлах? Используем стандартный модуль anydbm:

```
<listing>
>>> help("anydbm")          # узнаем, как пользоваться
модулем
>>> import anydbm           # импортируем его
>>> db = anydbm.open("jpg2gmail.db", "w") # открываем файл
данных
```

База данных db работает как обычный словарь (dict). В качестве ключа используем полный путь к файлу, значение — дата последней модификации. Это предполагает, что при отправке файла мы делаем следующее:

```
>>> db[file_path] = str(os.path.getmtime(file_path))
>>> # os.getmtime выдает дату последней модификации файла
```

Информация о факте отправки сохраняется в jpg2gmail.db. Тут необходимо остановиться и немного подумать. Может, вы найдете лучшее решение?

Определяем, нужно ли отправлять данный файл? Верное решение складывается из ответов на три других вопроса.

Заданного ли типа этот файл? Если нет, пропускаем его:

```
>>> import mimetypes
>>> mimetypes.guess_type(file_name)[0] == 'image/jpeg'
    Был ли этот файл уже загружен?
>>> db.has_key(file_path)
```

Если да, проверим, изменялся ли он после последней загрузки?

```
>>> str(os.path.getmtime(file_path)) == db[file_path]
```

Давайте подумаем, как превратить файл в электронное сообщение с присоединенными данными? Для меня это был самый сложный вопрос, но, немного поискав в Сети, я обнаружил готовый пример и практически без изменений использовал его здесь. Не хочу повторяться, поэтому объяснения последуют позже, когда доберемся до класса MessageManager.

Как отправить сообщение по электронной почте? Убедитесь сами, что проще простого:

```
>>> import smtplib
>>> S = smtplib.SMTP(smtp_server)
>>> S.sendmail(from_address, [destination_address], "message
as string")
```

Теперь группируем задачи и создаем сущности для их решения. То есть описываем классы, методы и функции. Для начала создадим файл jpg2gmail.py следующего содержания:

Листинг

```
00 #!/usr/bin/python
01 # -*-coding:utf8 -*-
02 # jpg2gmail.py
03 ""
04 Usage:
05 $ python jpg2gmail.py test
06 $ python jpg2gmail.py upload 10
07 or
08 >>> import jpg2gmail
09 >>> jpg2gmail.test()
10 >>> jpg2gmail.upload()
11 ""
12 import MimeWriter, base64, StringIO, smtplib, os,
mimetypes, sys, anydbm, ConfigParser, datetime
```

Комментарии

```
00 Стандартное начало для скриптов на Python:
указан путь к интерпретатору
01 Указаны кодировки, в которых набраны неанглийские
символы
02 Название файла
03-11 Документация к модулю. Мы всегда можем к ней обратиться
через переменную __doc__
12 Импортируем все нужные для работы модули
```

| Класс, управляющий сообщениями, — MessageManager |

На MessageManager в нашем случае возложены две задачи — создать текстовое сообщение, содержащее прикрепленный файл и пригодное для отправки через SMTP-сервер, и отослать его.

Листинг

```
15 class MessageManager:
16     """Create message from file path and send them via smtp
server """
17     def __init__(self, smtp_server, from_address, to_address):
18         self.sender = smtplib.SMTP(smtp_server)
19         self.from_address = from_address
20         self.to_address = to_address
21
22     def make_message(self, file_path):
```

```

23     file_name = file_path.split(os.path.sep)[-1]
24     subject=file_path.split(os.path.sep)[-2]
25
26     message = StringIO.StringIO()
27     writer = MimeWriter.MimeWriter(message)
28
29
30     writer.addheader('MIME-Version', '1.0')
31     writer.addheader('Subject', subject)
32     writer.addheader('From', self.from_address)
33     writer.addheader('To', self.to_address)
34
35     writer.startmultipartbody('mixed')
36
37     part = writer.nextpart()
38     body = part.startbody('text/plain')
39     body.write('Created by ... \n')
40
41     part = writer.nextpart()
42     part.addheader('Content-Transfer-Encoding',
43 'base64')
44     body=part.startbody('application/jpg;
45 name=%s' % file_name)
46
47     fp=open(file_path, 'rb')
48     body.write(base64.encodestring(fp.read()))
49     fp.close()
50
51     writer.lastpart()
52     return message
53
54 def send_message(self, message):
55     return self.sender.sendmail(self.from_address,
56 [self.to_address], message.getvalue())

```

Комментарии

15 Начало описания класса

16 Строка документации. Получаем доступ так: `myinstance.__doc__`

17 Описываем метод `__init__` — конструктор экземпляра класса. Этот метод вызывается только тогда, когда необходимо создать экземпляр класса

18 Указана сущность, ответственная за отправку. Мы будем использовать ее в строке 53

19, 20 Сохраняем данные для дальнейшего использования

22 Начало функции `make_message`. Обратите внимание на обязательное использование ключевого слова `self`

23 Получаем имя файла из полного пути к нему. Это последний элемент массива, который имеется после разбиения пути к файлу с помощью команды `split`. `os.path.sep` выдает разделитель для текущей операционной системы ("`/`" — для Unix, "`\\`" — для Windows)

24 Тема почтового сообщения — имя папки: получаем его, разбив путь к файлу и выбрав предпоследний элемент

26–38 Используем готовое без лишних вопросов

39 Тело почтового сообщения заполняем по своему усмотрению

40–44 Здесь представлены строки, необходимые для создания сообщения

45–47 Открываем файл, кодируем его содержимое по стандарту `base64`, записываем в сообщение и закрываем документ

48–51 Без комментариев — все и так понятно

52 Отправляем сообщение. `Message` — экземпляр класса `StringIO`. Получаем из него строку методом `getvalue()`

53 Возвращает пустой словарь, если все прошло успешно. Если есть проблемы, словарь содержит данные об ошибках. Обработываем в строке 95

| Класс `Logger`, отвечающий за ведение лога |**Листинг**

```

54 class Logger:
55     def __init__(self, file_name):
56         self.file_name=file_name
57     def write_line(self, txt):
58         fp = open(self.file_name, 'a')
59         fp.write("%s %s \n" %
60             datetime.datetime.now().strftime('%d/%m/%Y:%H:%M:%S'),
61             txt)
62         fp.close()

```

Комментарии

55–56 В конструкторе передаем имя файла для ведения журнала

58 Открываем файл для записи. Если бы вместо `'a'` было `'w'`, все данные перезаписывались бы. А так добавляются строки

59 Записываем строку в журнал: `datetime.datetime.now()` выдает текущее время; `ftime('%d/%m/%Y:%H:%M:%S')` позволяет получить строку вида `4/10/2005:2:50:46`, как в логах Apache

60 Закрываем файл

| Класс-обработчик — `Handler` |

Данный класс определяет, есть ли необходимость отсылать файлы из полученного списка, а затем отправляет выбранные и отмечает в базе факт их отправки.

Листинг

```

61 class Handler:
62
63     def __init__(self, message_manager, logger, db_file_name):
64         self.manager = message_manager
65         self.logger=logger
66         self.db_file_name = db_file_name
67
68
69     def get_db(self):
70         if os.path.isfile(self.db_file_name):
71             return anydbm.open(self.db_file_name, 'w')
72         return anydbm.open(self.db_file_name, 'c')
73
74     def isSuitable(self, file_path):
75         if mimetypes.guess_type(file_path)[0] !=
76 'image/jpeg':
77             return 0
78
79         db = self.get_db()
80         if db.has_key(file_path):
81             if str(os.path.getmtime(file_path))==
82 db[file_path]:
83                 db.close()
84                 return 0
85         db.close()
86         return 1
87
88     def mark_as_uploaded(self, file_path):
89         db = self.get_db()
90         db[file_path] = str(os.path.getmtime(file_path))
91         db.close()

```

| Класс-обходчик — Walker |

Основная задача этого класса — просмотреть все файлы, находящиеся в данной директории (включая также и поддиректории), и запомнить пути к тем из них, которые позднее будут отправлены по электронной почте.

Листинг

```

101 class Walker:
102     def __init__(self, handler):
103         self.handler = handler
104         self.files_for_upload = []
105
106     def go(self, root):
107         for dirpath, dirnames, filenames in os.walk(root):
108             for file_name in filenames:
109                 file_path = os.path.join(dirpath, file_name)
110                 if self.handler.isSuitable(file_path):
111                     self.files_for_upload.append(file_path)
    
```

Комментарии

- 101-103 Начинаем описывать класс. Конструктор получает экземпляр класса Handler в качестве аргумента
- 104 Создаем контейнер для хранения списка файлов, готовых к отправке
- 106 Указываем путь к директории, где хранятся нужные файлы
- 107-108 Обходим все файлы в директории
- 109 Вычисляем полный путь к файлу
- 110-111 Если файл подходящий, добавляем путь к нему в контейнер

| Функция, которая проверяет аргументы, переданные в командной строке |

Листинг

```

112 def test_arguments(log, argv):
113
114     def usage():
115         print >>>sys.stderr, 'don\'t understand arguments'
116         print >>>sys.stderr, __doc__
117         sys.exit(2)
118
119     if len(argv)<2 or argv[1] not in ['test', 'upload']:
120         usage()
121
122     if argv[1] == 'upload' and len(argv)==3:
123         try:
124             int(argv[2])
125         except:
126             usage()
    
```

Комментарии

- 112 Нужно проверить, правильно ли переданы аргументы в командной строке. Эта функция вызывается из main (мы рассмотрим ее далее)
- 114 Иногда полезно определить функцию в функции
- 115 Система отвечает на стандартный вывод фразой «don't understand arguments» — это означает, что она не понимает указанных аргументов. И ...
- 116 ... объяснит, как лучше поступить, напечатав документацию к модулю (см. комментарии к строкам 03–12)
- 117 Вывод сообщения ОС, что не все так хорошо, как нам хотелось бы

```

90
91     def process(self, files, qty=None):
92         if qty:
93             files = files[:qty]
94         for file_path in files:
95             error_dict = self.manager.send_message(self.manager.make_message(file_path))
96             if error_dict:
97                 self.logger.write_line('can\'t send file %s, %s ' %
98                                     (file_path, str(error_dict)))
99             else:
100                 self.mark_as_uploaded(file_path)
101                 self.logger.write_line('%s was uploaded' %
102                                     file_path)
    
```

Комментарии

- 61 Начало описания класса
- 63-66 Передаем конструктору экземпляр MessageManager; экземпляр Logger и имя файла — базы данных, в которых хранятся сведения об отправленных фотографиях
- 69 Метод для получения доступа к базе данных
- 70-71 Если файл существует, открываем его для записи
- 72 Если файла нет, создаем его.
Но при этом в любом случае возвращаем экземпляр для работы с базой данных
- 74 Смотрим, нужен ли файл указан
- 75-76 Если это не JPG-файл, то возвращаем 0 — он нам не подходит. Соответственно, отправлять его не нужно
- 78 Получаем доступ к базе данных
- 79 Есть такой ключ? То есть, мы этот файл уже отправляли?
- 80 Если отправляли, надо выяснить, был ли изменен документ с момента последней отправки.
(В качестве ключа здесь мы используем полный путь к файлу, а в качестве значения — дату последней модификации.)
Если дата последней модификации совпадает с данными, сохраненными в базе данных, то отправлять этот файл не нужно — возвращаем 0
- 81, 83 Не забываем закрывать базу. В противном случае, если компьютер даст сбой, то уже отправленные файлы придется отсылать снова. Выполнив команду db.close(), мы сохраняем данные на винчестер
- 84 Во всех остальных случаях файл нужно отправлять. Теперь возвращаем 1
- 86 Отмечаем факт отправки файла
- 87/89 Открываем/закрываем базу
- 88 Заносим в базу информацию об отправленном файле. Ключ — путь, значение — дата последней модификации
- 91 Метод для отправки группы файлов:
files — список с путями к файлам, которые нужно отправить; qty — необязательный параметр, указывающий количество файлов для отправки.
Зачем это нужно? Все просто. Если, например, 1000 файлов готовы к отправке, и мы отошлем их все за один раз, то провайдер может воспринять такую активность с нашей стороны как рассылку спама, что приведет к блокировке почтового ящика. Поэтому лучше отправлять письма порционно
- 92 Параметр передан?
- 93 Обрезаем список файлов
- 94 Перебираем все файлы из списка
- 95 Даем команду экземпляру MessageManager создать сообщение из файла (make_message) и отправить его (send_message).
Если все прошло успешно, error_dict окажется пустым
- 96-97 Если найдена ошибка, заносим данные в журнал
- 98-100 Ошибки нет — отмечаем факт успешной отправки

- 119-120 Если показано меньше двух аргументов (первый — это всегда путь к исполняемому в данный момент файлу), то это указывает на ошибку.
Или если второй аргумент — не test или upload, то это тоже можно расценивать как ошибку
- 122-126 Если второй аргумент — upload, а третий невозможно преобразовать в целое число, то здесь тоже ошибка

| Функция main |

Управление программой происходит отсюда.

Листинг

```

127 def main(argv=None):
128     script_dir = os.path.dirname(sys.argv[0])
129     log = Logger(os.path.join(script_dir, 'jpg2gmail.log'))
130
131     if argv is None:
132         argv = sys.argv
133     test_arguments(log, argv)
134
135     try:
136         config = ConfigParser.ConfigParser()
137         config.read(os.path.join(script_dir,
138                                "jpg2gmail.ini"))
139         destination_address = config.get('main',
140                                         'destination_address')
141         from_address = config.get('main',
142                                   'from_address')
143         smtp_server = config.get('main', 'smtp_server')
144         work_dir = config.get('main', 'work_dir')
145         message_manager =
146         MessageManager(smtp_server, from_address,
147                        destination_address)
148         handler = Handler(message_manager, log,
149                            os.path.join(script_dir, 'jpg2gmail.db'))
150         walker = Walker(handler)
151         walker.go(work_dir)
152     except:
153         for msg in sys.exc_info():
154             log.write_line(msg)
155             sys.exit(2)
156
157     if argv[1] == 'test':
158         print '%s files ready for upload' % len(walker.files_for_upload)
159
160     if argv[1] == 'upload':
161         if len(argv)==3:
162             handler.process(walker.files_for_upload,
163                             int(argv[2]))
164         else:
165             handler.process(walker.files_for_upload)

```

Комментарии

- 127 Главная функция — точка входа в программу
- 128 Вычисляем путь к директории, в которой хранится исполняемый файл. Именно в ней должны находиться файлы jpg2gmail.ini, jpg2gmail.db и jpg2gmail.log
- 129 Создаем экземпляр класса Logger. os.path.join используется для вычисления полного пути к файлу журнала
- 131 Если аргументы командной строки не переданы напрямую, то получаем их с помощью модуля sys

- 133 Проверяем правильность аргументов.
- 135 Выражение try (и except в 149) поможет найти ошибки
- 136-141 Получаем текущие параметры из файла jpg2gmail.ini
- 143-148 Создаем основных игроков и отправляем Walker в путь. В этой ситуации можно слегка почувствовать себя современным магом
- 149-152 Если произошли ошибки, заносим сведения о них в журнал и выходим, обобщив операционной системе, что у нас проблемы
- 154-155 Если получена команда test, распечатываем количество готовых к отправке файлов
- 157 Если получена команда upload ...
- 158 ... и аргументов передано три (например, python jpg2gmail.py upload 7)
- 159 ...даем команду на отправку указанного выше количества файлов
- 160-161 Если количество файлов не задано, отсылаем все файлы, готовые к отправке

Дополнительные функции, которые вызываются при импорте модуля, выглядят вот так:

```

>>> import jpg2gmail
>>> jpg2gmail.test() # распечатает количество
# файлов, готовых к отправке
>>> jpg2gmail.upload(2) # отправит два файла
>>> jpg2gmail.upload() # отправит все файлы, готовые к отправке

```

Листинг

```

162 def test():
163     main([None, 'test'])
164
165 def upload(qty=None):
166     if qty is None:
167         main([None, 'upload'])
168     else:
169         main([None, 'upload', qty])
170
171 if __name__ == '__main__':
172     sys.exit(main())

```

Комментарии

- 162 Эта функция дает команду вывести на печать все файлы, готовые к отправке
- 163 Вызываем функцию main, эмулируя список аргументов командной строки. Первый аргумент — путь к скрипту — в этом случае не используется
- 165 Функция дает команду на отправку файлов
- 166-169 Если передан необязательный аргумент qty (количество), переназначаем его функции main вместе с командой upload. Если qty нет, то просто выполняем команду upload
- 171 Условие __name__ == '__main__' выполняется, только если скрипт запускается из командной строки, в случае импорта другим модулем этого не происходит
- 172 Закрывает сессию интерпретатора и возвращает результат выполнения функции main. Если main вернет 2, то все узнают, что что-то пошло не так

Ну вот мы и добрались до конца. Поздравляю! Если вы разобрались в этом не совсем простом примере, то в следующем номере журнала вас будет ждать приз — небольшой программный продукт (естественно, открытый), благодаря которому от клиентов не будет отбоя. |

Секреты системы

Очередная десятка полезных советов от Дениса Колисниченко. На этот раз рекомендации так или иначе касаются аппаратного обеспечения вашего компьютера.

Оборудование

Linux и флеш-диск

Флеш-диски становятся все более популярными. Этому есть вполне логичное объяснение: они очень удобны — вы можете купить накопитель в виде брелока, кроме того, он может быть встроен в обычные часы или цифровой фотоаппарат. Вам не нужно больше носить с собой громоздкие компакт-диски — вы можете приобрести компактный флеш-драйв и в любой момент скопировать на него не только важные рабочие документы, но и свежескачанный фильм. К тому же цены на флеш-накопители постоянно падают, и сейчас они уже вполне доступны любому пользователю. В Windows 2000/XP работать с флеш-диском очень просто: достаточно подключить его к ПК через USB-разъем, и операционная система тут же определит наличие нового устройства. А как же дело обстоит в Linux?

Здесь вам нужно как всегда лишь один раз настроить и забыть — в дальнейшем вы сможете спокойно пользоваться вашим флеш-диском. Для поддержки USB-накопителей в Linux используется модуль `usb-storage`. Его можно запустить с помощью следующей команды:

```
# service usb start
```

Но если сервис USB у вас не поддерживается, придется вручную загружать этот модуль с помощью команды `modprobe`, но перед этим необходимо установить модуль USB-контроллера. Названия этих модулей различаются в зависимости от производителя контроллера. Например, модуль для контроллеров Intel именуется `uhci-hcd` (или `usb-uhci` — для ядра 2.4):

```
# modprobe uhci-hcd
```

```
# modprobe usb-storage
```

Если все будет работать правильно, вам нужно прописать загрузку этих модулей в файл `/etc/modules.conf`, чтобы каждый день не загружать их вручную. Например, в моей системе поддержка USB в `/etc/modules.conf` обеспечивается такой вот командой:

```
probeall usb-interface usb-uhci ehci-hcd
```

После загрузки модулей вам нужно установить программу `hotplug` (если она, конечно, по каким-то причинам до сих пор не установлена в вашей системе). При подключении нового флеш-диска `hotplug` автоматически найдет его и ассоциирует со SCSI-устройством (для флеш-дисков выполняется эмуляция SCSI). Обычно накопитель ассоциируется с устройством `/dev/sda`, но если у вас уже есть SCSI-диск, то флеш может стать, например, устройством `/dev/sdb` и т. д. Чтобы узнать, как именно его определила система, необходимо заглянуть в системный журнал, размещенный в `/var/log/messages`. В нем должны появиться соответствующие записи, ясно указывающие на то, с каким устройством ассоциирован флеш-диск:

```
kernel: usb 1-2: new full speed USB device using address 2
```

```
kernel: scsi0 : SCSI emulation for USB Mass Storage devices
```

```
kernel: Vendor: PQI Model: JoyTone U800 Rev: 1.00
```

```
kernel: Type: Direct-Access ANSI SCSI revision: 02
```

```
kernel: SCSI device sda: 502528 512-byte hdwr sectors (257 MB)
```

```
kernel: sda: assuming Write Enabled
kernel:
```

```
/dev/scsi/host0/bus0/target0/lun0: p1
kernel: Attached scsi removable disk sda
at scsi0, channel 0, id 0, lun 0
kernel: USB Mass Storage device found
at 2
```

В данном случае видно, что мы подключили флеш-накопитель объемом 256 Мбайт, и он ассоциируется с устройством `/dev/sda`.

Теперь нам осталось только подмонтировать это устройство в качестве обычного диска. На нем будет всего один раздел — `sda1` (или `sdb1`), который мы, собственно, и станем использовать:

```
# mount /dev/sda1 /mnt/flash-disc
```

В случае если на флеш-диске вообще нет разделов, необходимо монтировать весь `/dev/sda`. Только после этого с ним можно будет работать. Однако у вас должен существовать каталог `/mnt/flash-disc`. Вы также можете добавить следующую строчку в `fstab`:

```
/dev/sda1 /mnt/flash-disc vfat
rw,users,noauto,ioccharset=koi8-r,codepage=866,umask=0 0 0
```

Посредством этой команды мы указали, что флеш-диск будет содержать файловую систему VFAT, как в Windows, поэтому для него нужно задать кодировки `ioccharset=koi8-r,codepage=866`, как и для обычного Windows-раздела. Файловая система флеш-накопителя монтируется в режиме чтения/записи (`rw`); опции `users` и `noauto`, соответственно, разрешают монтирование накопителя обычным пользователям и запрещают это делать в автоматическом режиме. Опция `sync` запрещает кеширование записи во избежание ситуаций, когда вы уже вытащили диск, а данные физически на него еще не записались.

Настройка системы

Пусть Linux немного поспит

Как вы догадались, ниже речь пойдет о реализации спящего режима в Linux. В мире Windows этот режим давно стал привычным, и мы попробуем реализовать эту функцию в нашей ОС.

Для чего используется спящий режим? Он позволяет сохранить содержимое памяти на жесткий диск, после чего питание компьютера выключается. Включение же питания полностью восстанавливает состояние операционной системы: запускаются программы и открываются все файлы (даже несохраненные) — таким образом, пользователь может просто продолжать свою работу, причем сразу, а не дожидаясь перезагрузки системы.

Спящий режим поддерживается во всех ядрах 2.6. Ядро 2.4 для поддержки спящего режима нужно пропатчить, но мы этот вариант рассматривать не будем — почти все современные дистрибутивы основаны на ядре 2.6, поэтому в большинстве случаев вам не нужно ничего делать дополнительно. Но, скорее всего, вам придется увеличить раздел подкачки. Сохранение содержимого памяти происходит в разделе подкачки. Ясно, что его размер должен быть равен как минимум объему оперативной памяти, а еще лучше, чтобы он был больше. Например, если на вашем компьютере установлено 512 Мбайт ОЗУ, будет хорошо, если на раздел подкачки отведется хотя бы 640 (512+128) Мбайт.

Для загрузки сохраненного содержимого памяти нужно передать ядру параметр `resume2=имя_раздела`, или просто `resume=имя_раздела` (выбор зависит от версии ядра, поэтому нужно поэкспериментировать). Название раздела, как правило, соответствует имени раздела подкачки, например `/dev/hda3`. Данный параметр необходимо прописать в файле конфигурации вашего загрузчика LILO/GRUB, чтобы ядро использовало его по умолчанию.

После этого алгоритм работы ядра будет следующим: если найден сохраненный образ памяти — ядро загрузит

его; если не найден — будет продолжена обычная загрузка.

Рассмотрим одну неприятную ситуацию. Существует способ загружать систему с нуля, даже если сохранен образ памяти. Для этого надо просто убрать параметр `resume (resume2)`, но при следующей загрузке снова будет обнаружен образ, и система попытается его загрузить. Но поскольку вы уже работали с системой в прошлый раз, может быть повреждена файловая система. Ведь во время перехода в спящий режим не сбрасывается дисковый кеш, поэтому при «пробуждении» системы на дисках, куда производилась запись, пока система «спала» (когда вы загрузили ее с нуля), будет разница между тем, что уже записано, и тем, что содержится в дисковом кеше, сохраненном при переходе в спящий режим. Так что, если вам пришлось загрузиться без «пробуждения» системы, нужно очистить раздел подкачки с помощью команды `mkswap /dev/hdXX`. После этого вы можете уже не удалять каждый раз параметр `resume` при загрузке системы.

К чему все это было сказано? Ведь если я использую спящий режим, значит он мне нужен, и необходимости загружаться с нуля не возникнет. На самом деле все немного не так... Спящий режим в Linux — пока недостаточно надежная вещь, и в некоторых случаях система может просто-напросто не «проснуться». Вот тогда вам и понадобится загрузка без `resume`. К тому же спящий режим в Linux не может работать на многопроцессорных машинах, не поддерживает работу со SCSI, а кроме того, имеет ряд других ограничений. И еще: если у вас загружен модуль `nvidia` (поддержка видеоплат от NVIDIA), первая версия спящего режима (`swsusp`, когда вы вводите параметр `resume`, а не `resume2`) не работает. Сейчас проверим, функционирует ли у вас спящий режим. Если вы выполнили все перечисленные выше рекомендации, для перехода в спящий режим можно использовать одну из двух приведенных ниже команд:

```
# echo 4 > /proc/acpi/sleep
# echo disk > /sys/power/state
```

Если в вашей системе имеется сценарий `hibernate`, корректнее использовать именно его — тогда вам не придется гадать, какая команда лучше.

Настройка системы

Восстановление загрузчика Linux после переустановки Windows

Если на вашем компьютере установлены одновременно Linux и Windows, то примерно раз в полгода (или даже чаще) станет повторяться ситуация, когда Windows начнет перезаписывать загрузчик Linux. Попросту говоря, во время очередной переустановки операционная система Windows как всегда не заметит посторонний загрузчик и удалит его, даже не спросив вас, хотите ли вы это делать.

Итак, Windows вы уже переустановили. Не делать же этого и с Linux... Как восстановить загрузчик? Если вы при установке создали Rescue-дискету, загрузитесь с нее и введите команду:

```
# lilo -
если у вас установлен загрузчик LILO
# grub-install /dev/hda -
если у вас установлен загрузчик GRUB
```

Как поступить, если дискеты у вас нет, а переустанавливать Linux не хочется? В этом случае можно воспользоваться LiveCD, например Gentoo LiveCD или Knoppix. Его можно приобрести в интернет-магазине. Понимаю, проще переустановить систему. Но все равно купите этот диск — через некоторое время он вам обязательно понадобится. Итак, если у вас есть LiveCD, в его командной оболочке последовательно введите следующие команды:

```
# mkdir /mnt/my_linux
# mount /dev/hdXX /mnt/my_linux
# chroot /mnt/my_linux
# lilo или
grub-install /dev/hda
```

Разберемся теперь, что здесь написано. Первая команда создает каталог `my_linux` — чтобы было куда монтировать наш Linux-раздел. Вторая же монтирует раздел, на который установлена ваша ОС, к каталогу `/mnt/my_linux`. Третья команда изменяет корневую

файловую систему. После этого каталог `/mnt/my_linux` становится корнем вашей системы. Теперь можно вводить команды, как будто вы загрузили обычную ОС Linux. Только предварительно нужно установить загрузчик. В зависимости от используемого загрузчика введите команду `lilo` или `grub-install`. После этого нажмите комбинацию клавиш «Ctrl+Alt+Del».

Настройка системы

Меняем место жительства

Нет, переезжать в другой город мы не будем, а просто попытаемся переместить нашу систему с одного жесткого диска на другой.

Предположим, что у нас есть два жестких диска — `/dev/hda` и `/dev/hdb`. Первый — оригинальный, а второй — тот диск, на который нужно записать систему. Для большей определенности скажем, что на первом диске у нас три раздела — `/dev/hda1` (`/`), `/dev/hda2` (`/home`) и `/dev/hda3` (`swap`).

На втором диске тоже нужно создать три раздела. Конечно, желательно, чтобы совпадали и файловые системы. Например, если первые два раздела на оригинальном диске у вас относятся к файловой системе Ext3, то и на втором нужно создать два соответствующих раздела — сделать это можно с помощью команды `fdisk` или `сfdisk`. После этого лучше сразу перезагрузить компьютер. Будем считать, что новые разделы называются `/dev/hdb1`, `/dev/hdb2` и `/dev/hdb3`.

При перезагрузке укажите параметр ядра `single` — система будет загружена в однопользовательском режиме. После этого нужно отформатировать созданные разделы. Форматирование выполняется посредством команды `mkfs.ext3`, если вы используете Ext3. Если же отдаете предпочтение XFS или ReiserFS, вам необходимо запустить команды `mkfs.xfs` или `mkfs.reiserfs` соответственно. Отформатировать раздел подкачки можно с помощью команды `mkswap`. После этого надо смонтировать раздел `/dev/hdb1` как `/mnt/root`, а затем скопировать содержимое корневой файловой системы командами:

```
# cp -ax / /mnt/root
```

Приступаем ко второму разделу. Подмонтируем его как `/mnt/home` и скопируем данные на него:

```
# cp -ax /home /mnt/home
```

Все, что теперь вам осталось, — это просто отредактировать конфигурационный файл загрузчика на новом винчестере и записать его в загрузочный сектор на разделе нового жесткого диска, а не в MBR! При этом не запутайтесь в обозначениях жестких дисков — `hda` и `hdb`.

Тестирование оборудования

Проверяем железо

В Windows, наверное, все мы хоть раз в жизни использовали какую-нибудь программу для проверки железа. Как правило, это продукты производства сторонних разработчиков. В Linux же все, что нужно, всегда под рукой — часто все необходимые приложения входят в состав дистрибутива:

- ▶ `hdparm` — используется для установки параметров жесткого диска; кроме того, ее же можно применять для вычисления скорости работы диска;
- ▶ `hddtemp` — осуществляет мониторинг температуры жесткого диска (правда, далеко не все модели HDD оснащены датчиками температуры);
- ▶ `smartmontools` — предназначена для мониторинга жестких дисков, поддерживающих технологию S.M.A.R.T.; правда, такие устройства достаточно редко встречаются и, как правило, относительно дорого стоят;
- ▶ `badblocks` — проверяет жесткие диски на наличие битых секторов;
- ▶ `bonnie` — проводит тестовые испытания жестких дисков на производительность;
- ▶ `memtest86` — осуществляет проверку оперативной памяти;
- ▶ `screentest` — служит для настройки монитора;
- ▶ `сrubburn` — используется для стресс-теста процессора.

Тут все понятно, кроме стресс-теста. Что это такое? Суть данного метода заключается в том, что процессор загружается какой-либо работой. Это может быть компиляция большой, даже

огромной, программы, например ядра. Если ваша система работает нестабильно, например из-за перегрева процессора или плохой памяти, через определенное время обязательно произойдет ошибка компиляции, причем не важно, какая именно — синтаксическая или какая-то другая.

Продуктивность

Уменьшение времени запуска OpenOffice

Наверное, вы заметили, что любое приложение из этого пакета запускается, откровенно говоря, медленно. Чтобы ускорить этот процесс, нужно установить утилиту быстрого запуска OpenOffice. Помните, что-то подобное было в Microsoft Office, только в случае с OpenOffice данная утилита действительно работает. Называется она OpenOffice Quick Starter. Пакет, содержащий это приложение, называется `oooqs` или `oooqs-kde` — в зависимости от дистрибутива. В случае если вы используете GNOME, он будет именоваться `ooqstart-gnome`. В большинстве дистрибутивов эти пакеты имеются. Все, что вам нужно сделать, — установить нужный пакет и наслаждаться быстрым запуском OpenOffice.

Настройка системы

Воспроизведение неподдерживаемых форматов

Во многих дистрибутивах Linux мы не можем воспроизвести, казалось бы, привычные форматы — DivX, Windows Media (WMV), QuickTime, DVD и иногда даже MP3. Причина всему этому — всевозможные лицензионные ограничения, нарушение которых не допускается. Но выход из этой ситуации все же есть: распространять тот или иной кодек в составе дистрибутива запрещено, но никто не мешает загрузить его и использовать в свое удовольствие.

В большинстве случаев кодекы выполнены в виде динамических библиотек Windows — DLL-файлов. Чтобы их использовать в Linux, нужен `mplayer` — кросс-платформенное приложение, позволяющее воспроизводить различные

форматы мультимедийной информации. Найти программу mplayer в Интернете не проблема, поэтому вы и сами справитесь с этой задачей. А вот кодеки отыскать намного сложнее. Я рекомендую загрузить пакет essential, который доступен по адресу в Интернете: www.mplayerhq.hu/homepage/dload.html. Давайте установим этот пакет:

```
$ tar -jxvf essential-20050216.tar.bz2
$ sudo cp essential-20050216/*
  /usr/lib/win32/
```

Первая команда распаковывает кодеки, а вторая копирует их в каталог /usr/lib/win32. Чтобы кодеки стали доступными, вам нужно перезапустить mplayer. С воспроизведением DVD ситуация тоже неоднозначная. Некоторые дистрибутивы, особенно новые (но не все), воспроизводят DVD без проблем. А проигрыватели других дистрибутивов (например, xine) вообще отказываются работать с DVD. Проблема заключается в том, что проигрыватели просто не поддерживают этот формат. Чтобы изменить ситуацию, нужно перекомпилировать проигрыватели из исходных кодов. Прочитать обо всем этом можно в Сети, по адресу: www.geniusweb.com/LDP/HOWTO/html_single/DVD-Playback-HOWTO.

И еще: если у вас не воспроизводится DVD, возможно, с вашим дистрибутивом все нормально, просто компьютер слишком слабый. Для проигрывания DVD нужна машина с минимальной конфигурацией Pentium III 500 МГц и 256 Мбайт оперативной памяти. Также убедитесь, что включен DMA (прямой доступ памяти) для вашего DVD-привода.

Продуктивность

Просмотр документов MS Word в консоли

Для просмотра и редактирования документов MS Word с успехом используется OpenOffice Writer, но иногда нам нужно быстро просмотреть документ (например, вспомнить номер телефона, который указан в самом конце), и запускать тяжеловесный Writer не хочется, или же OpenOffice вообще не установлен.

Для просмотра документов MS Word нам понадобятся две утилиты — wv и

w3m. Первая — это конвертер документов Word (мы его будем использовать для преобразования их в формат HTML), а вторая — универсальный просмотрщик/браузер. Как вы догадались, w3m позволяет просматривать HTML-файл, созданный первой программой. Оба приложения входят в состав современных дистрибутивов, правда, могут быть не установлены по умолчанию. Для их инсталляции надо воспользоваться менеджером пакетов.

Итак, для успешного просмотра документа document.doc воспользуемся одной большой командой:

```
# wvWare -x /usr/lib/wv/wvHtml.xml
  document.doc | w3m -T text/html
```

Попробуйте-ка запомнить и быстро ввести эту команду — за это время успеют загрузиться две копии OpenOffice. Наша же цель — быстрый просмотр документов. Поэтому создадим сценарий viewword для просмотра документов Microsoft Word:

```
# !/bin/bash
wvWare -x /usr/lib/wv/wvHtml.xml $1
2>/dev/null | w3m -T text/html
```

Теперь запишем сценарий в каталог /usr/local/bin, чтобы он стал доступным для всех пользователей. Не забудьте, что он должен быть исполняемым:

```
# chmod +x /usr/local/bin/viewword
```

Для просмотра документа document.doc далее надо ввести команду:

```
$ viewdoc document.doc
```

Продуктивность

Просмотр документов PDF в консоли

Принцип реализации этого совета схож с предыдущим. Как и в случае с MS Word, сначала мы преобразуем PDF-документ в формат HTML, а потом просмотрим его с помощью текстового браузера.

Для преобразования PDF-документа в формат HTML будем использовать программу pdftohtml. Она входит в состав многих дистрибутивов, но если на вашем компьютере она не установлена, можно загрузить ее с сайта <http://pdftohtml.sourceforge.net>. Для просмотра созданного программой HTML мы будем использовать браузер

elinks — для этого случая он подходит лучше, чем w3m.

Итак, для просмотра файла file.pdf будем использовать следующую команду:

```
# pdftohtml -q -noframes -stdout file.pdf
  | elinks
```

Теперь опять создадим сценарий, автоматизирующий работу:

```
# !/bin/bash
pdftohtml -q $1 ~/temp.html
elinks ~/temp.html
```

И назовем его viewpdf (не забудьте сделать его исполнимым и скопировать в каталог /usr/local/bin). Теперь для просмотра PDF-файлов будем использовать команду:

```
# viewpdf <имя_файла>.pdf
```

Настройка системы

Как установить графический фон загрузчика GRUB?

Много интересных картинок, пригодных для установки в качестве фона загрузчика GRUB, можно найти по адресу: <http://ruslug.rutgers.edu/~mcgrof/grub-images/images/working-splashimages>. Чтобы не загружать все эти картинки, можно просмотреть их уменьшенные изображения (<http://ruslug.rutgers.edu/~mcgrof/grub-images/images>), а уже после загрузить понравившуюся картинку. Выбранное изображение нужно поместить в каталог /boot/grub:

```
# cp image.xpm.gz /boot/grub
```

Далее необходимо изменить файл конфигурации GRUB — /boot/grub/grub.conf. Для этого можно воспользоваться вашим любимым текстовым редактором. Для того чтобы поменять фон, нужно добавить или изменить следующую строку:

```
splashimage=(hd0,0)/boot/grub/image
.xpm.gz
```

С файлом картинки ясно. А вот значение (hd0,0) вам, возможно, придется изменить; (hd0,0) — это начальный раздел первого диска (Primary Master) в вашей системе. Очень сомневаюсь, что Linux установлена у вас именно туда — как правило, этот раздел бывает занят Windows.

После перезагрузки вы увидите новый фон для вашего загрузчика. |

Спасатель данных

Спасательные круги вышли из моды — да здравствуют спасательные флеш-карты! Если Windows «тонет», вы можете с помощью миниатюрного USB-спасателя вернуть ваши данные к жизни.

Что нам понадобится

Самое необходимое

- ▶ USB-донгл на 128 Мбайт (www.damnsmalllinux.org)
- ▶ WinImage 7.0 (www.winimage.com)
- ▶ Damn Small Linux 1.2.1 (www.damnsmalllinux.org)
- ▶ Загрузочный образ для USB-брелока

Как ни странно, иметь персонального спасателя данных в связке ваших ключей, который всегда готов прийти на помощь, может сегодня практически каждый из нас. О каком спасателе идет речь? Все очень просто. Это сохраненный на брелоке USB Flash вариант операционной системы Linux, который можно запустить на большинстве ПК. Если Windows не хочет загружаться, вы сможете быстро спасти важные документы, которые находятся на жестком диске. Мини-система умеет еще кое-что: у нее есть собственный браузер, текстовый редактор и редактор таблиц. Вы можете добавить к этому игры, чтобы не скучать во время копирования спасаемых данных.

Все это становится возможным благодаря маленькой бесплатной разновидности Knoppix. Она называется Damn Small Linux (DSL) — то есть «чертовски маленький Линукс» — и занимает на USB-карте ровно 50 Мбайт.

Здесь есть лишь одно маленькое неудобство: некоторые компьютеры не любят загружаться с USB-накопителя. Они поддерживают такую опцию, но она, тем не менее, не работает. Вина в этом плохая стандартизация материнских плат. Однако не унывайте раньше времени. Если ваш ПК не «грузится» с

флеш-драйва, мы предлагаем вам альтернативу. Linux можно записать на Mini CD (болванку 8 см). Диск прекрасно помещается в кармашке для кредитных карт вашего бумажника. Подробнее об этом читайте в блоке «Сетевой ПК в кошельке».

Кстати, с помощью окружения BartPE можно изготовить USB-брелок с Windows. Для этого вам потребуется флеш-карта объемом 256 Мбайт или обычный CD, загрузка с которого занимает гораздо меньше времени, чем с USB-носителя.

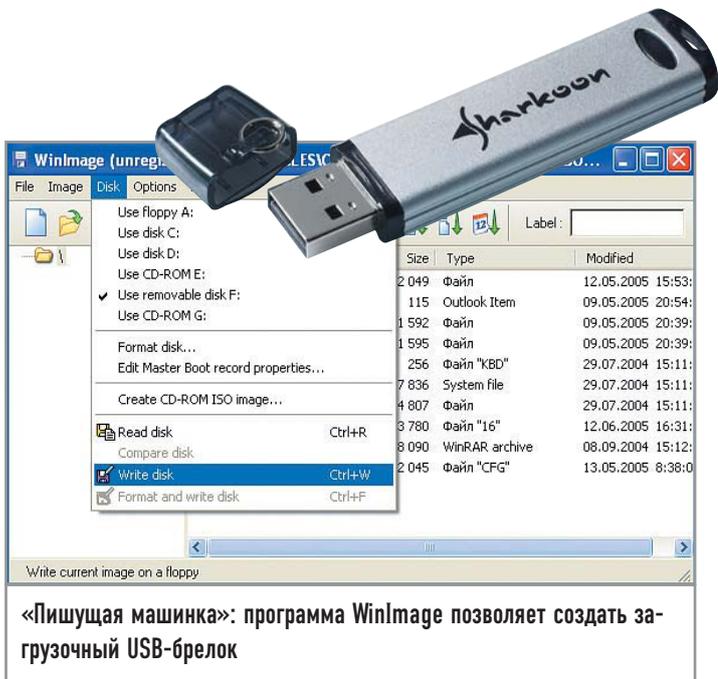
1. Форматируем флеш-брелок

После того как вы подключили его к компьютеру, щелкните правой кнопкой мыши в окне «Мой компьютер» по символу флеш-карты и выберите опцию «Форматировать». В качестве файловой системы задайте FAT. При желании введите название (метка тома) и щелкните по кнопке «Начать». Через несколько секунд ваша карта будет отформатирована.

2. Копируем DSL-загрузчик

Чтобы сделать флеш-накопитель загрузочным, скопируем на него Boot Image. Для этого используем файлы bootfloppy-usb.img и dsl-1.2.1.iso, которые можно найти на одном из серверов, перечисленных на сайте www.damnsmalllinux.org/download.html. Кроме того, вам понадобится WinImage 7.0 с сайта www.winimage.com. Эта программа распространяется условно бесплатно, и вы можете свободно использовать ее в течение 30 дней.

Запустите программу WinImage и откройте в ней файл bootfloppy-usb.img. В меню «Disk» выберите строчку «Use removable Disk E:». Буква, обозначающая диск, может варь-



«Пишущая машинка»: программа WinImage позволяет создать загрузочный USB-брелок

роваться в зависимости от того, как Windows назовет вашу флеш-карту. После этого вновь откройте меню «Disk → Write Disk» и на все последующие сообщения с вопросами от программы ответьте «Да». WinImage запишет файл, и ваш флеш-брелок станет загрузочным.

3. Устанавливаем Linux

Теперь откройте в WinImage с помощью меню «File → Open» файл dsl-1.2.1.iso, а также USB-брелок в обычном Проводнике Windows. Выделите все файлы, которые вы видите в правом окне WinImage, и перетащите их в каталог USB-накопителя.

4. Загружаем и настраиваем

Перезагрузите компьютер и нажмите клавишу «Delete», чтобы попасть в BIOS. Найдите очередность устройств, с которых может быть загружен компьютер. В качестве первого устройства выберите «USB», «USB Device» или нечто похожее. Сохраните изменения и снова перезагрузите компьютер.

Теперь компьютер загружается с USB-брелока. На требование подтвердить загрузку нажмите «Enter» — и Linux запу-

стится. Обычно Linux сразу загружается с графической оболочкой. Однако на некоторых компьютерах для этого не подходит разрешение монитора. В таком случае сделайте в командной строке следующую запись:

xsetup

Выберите первую из предложенных возможностей и следуйте указаниям. В заключение введите:

startx

5. Копируем файлы

Для того чтобы скопировать данные из раздела жесткого диска Windows XP, сначала щелкните по пункту «Xterminal». Откроется консоль, в которой необходимо будет ввести следующие строки:

mount./mnt/hda1

mount./mnt/sda1

sudo.emelfm

С их помощью первый раздел жесткого диска (hda1) и USB-брелок (sda1) интегрируются в систему, и запускается файловый менеджер Emelfm, который отображает структуру каталогов. Они необходимы для копирования файлов. Слева в файловом менеджере выберите /mnt/hda1 — вы увидите содержание жесткого диска. Справа перейдите в каталог /mnt/sda1. Выделите слева необходимый файл, нажмите «Сору», и он скопируется на флеш-карту. По окончании работы не вынимайте флеш-брелок из разъема сразу, сначала выключите компьютер.

6. Добавляем дополнительные инструменты

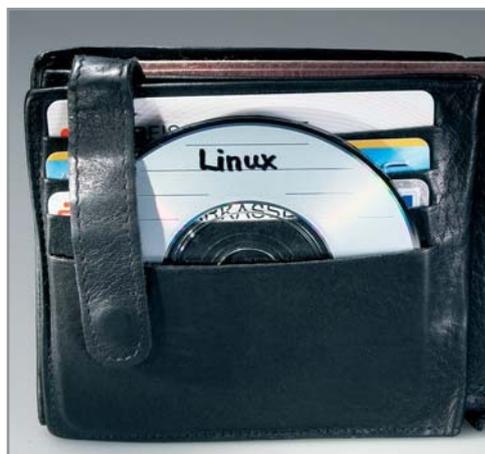
На страницах интернет-сайта, расположенного по адресу [ftp://ibiblio.org/pub/Linux/distributions/damnsmall](http://ibiblio.org/pub/Linux/distributions/damnsmall), вы найдете дополнительные инструменты и игры. Загрузить их довольно просто: отложите файлы с расширением .DSL в самый нижний уровень каталога флеш-брелока, и при следующем старте они запустятся автоматически. |

Записываем DSL на CD

Сетевой ПК в кошельке

Если компьютер не загружается с флеш-карты, не спешите вывешивать белый флаг. Скопируйте Damn Small Linux вместо USB-карты на Mini CD. Недостаток: вы не сможете скопировать спасенные файлы на тот же носитель информации или добавить на него дополнительные программы. Но если вы все же загрузили Linux с CD, можете подключить флеш-карту и скопировать документы или переслать их по электронной почте.

Вам понадобится болванка Mini CD, которая стоит не намного больше 12-сантиметрового аналога. Любая программа для записи CD, совместимая с файловой системой ISO, запишет ваш DSL на такую болванку. С помощью Nero это делается так: в диалоговом окне «New Compilation» щелкните «Open» и выберите образ ISO «dsl-1.2.1.iso». Затем щелкните «Burn», и Nero запишет диск.



В бумажнике: на маленькой болванке хватает места для Mini-Linux, она же, в свою очередь, помещается в кармашке для кредитных карт

View Picture

View Picture — просмотрщик графических файлов, построенный на SDL, включающий в себя весь джентльменский набор функций: здесь и слайдшоу, и поддержка полноэкранный ре-

жима, и масштабирование, и произвольная навигация по списку изображений. Примечательно (такое действительно встречается не часто), что спустя четыре года забвения разработчик проекта Эрик Гринвальд снова вернулся к своему детищу и выпустил его обновленную версию.

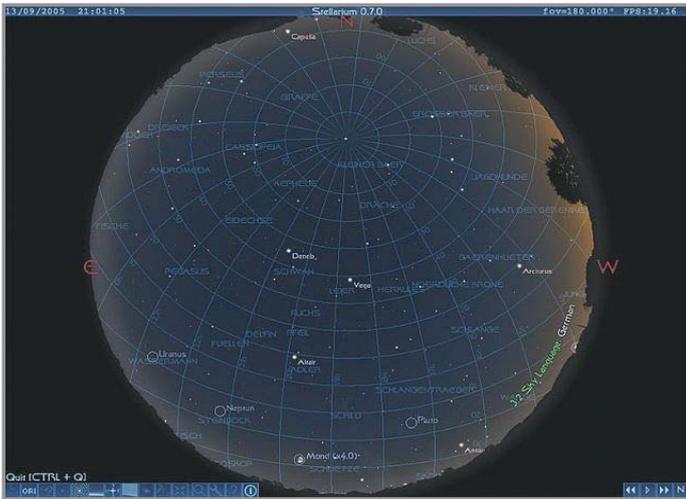
Графической оболочки у программы нет, так что, если вам надо посмотреть изображение, просто наберите команду `vp image.jpg`

Лицензия: GPL

Язык интерфейса: английский

Домашняя страница:

<http://math.missouristate.edu/~erik>



Настольная обсерватория

Качественное открытое ПО подобного рода — своеобразная изюминка Linux. Любой пользователь KDE знаком с KStars. Celestia же впечатляет предлагаемой экскурсией по Солнечной системе... Мы хотим обратить ваше вни-

мание на очередную версию виртуального планетария Stellarium. Эта программа имеет характерные отличительные особенности, благодаря которым вы никогда не спутаете ее с указанными выше продуктами. Например,

на картах звездного неба можно лицезреть контуры мифологических персонажей — Ориона и Льва, Большой Медведицы и Девы... Путешествие по небосводу, заполненному более чем 120 тысячами мерцающих звезд (из каталога Hipparcos), наверняка станет увлекательным и незабываемым. Фотореалистичный Млечный путь и падающие звезды, движение планет и их спутников в режиме реального времени, более 70 туманностей — все это не оставит равнодушным даже далекого от астрономии человека. Помимо прочего, Stellarium позволяет менять как весь ландшафт целиком, в том числе и на собственный, так и его отдельные элементы. Приложение замечательно отображает смену

времени суток — закат, восход Солнца и Луны; имеет систему автоматической адаптации яркости неба к глазу, используя психологическую модель; масштабирует небесные светила так, как если бы вы наблюдали их в бинокль или небольшой телескоп. Немаловажным фактом можно считать и то, что Stellarium используется даже в планетариях (более подробную информацию можно найти на сайте www.digitaliseducation.com). Хотя этот программный продукт создан на основе OpenGL, он отличается разумной требовательностью к аппаратному обеспечению.
Лицензия: GPL
Интерфейс: английский
Домашняя страница: www.stellarium.org

The friendly interactive shell

Чего может не хватать в традиционной консоли Linux? Разработчик The friendly interactive shell (fish) решил, вероятно, что интерактивности и дружелюбности много не бывает. Как это получилось на деле — решать пользователю. Но вряд ли предлагаемый программой ассортимент элементов заставит задуматься о неудобстве. Например, специально для начинающих пользователей в fish предусмотрен вызов страниц с документацией в окне веб-браузера, а в числе дополнительных

функций — поддержка буфера обмена X, подсветка синтаксиса, проверка ошибок, удобные вкладки и многое-многое другое. Автор проекта Аксель Лиленкрантц исправно совершенствует свое детище от версии к версии. Кроме того, на домашней странице fish размещено огромное количество дополнительной информации, инструкции по сборке из исходников, практические советы по установке пакетов, а в помощь новичкам предлагается подборка документации.

```

Edit View Terminal Tabs Help
shellboy -> case -h
opsis
switch VALUE; [case [WILDCARD...]; [COMMANDS...];...] end

cription
The switch statement is used to perform one of several blocks of
commands depending on whether a specified value equals one of several
wildcarded values. The case statement is used together with the switch
statement in order to determine which block should be performed.

#!perl
If the variable $animal contains the name of an animal, the following
code would attempt to classify it:

switch $animal
  case cat
    echo evil
  case wolf dog human moose dolphin whale
    echo mammal
  case duck goose albatros
    echo bird
  case shark trout stingray
    echo fish
end

If the above code was run with $animal set to whale, the output would
be mammal.

```

Лицензия: GPL

Язык интерфейса: английский

Домашняя страница:

<http://roo.no-ip.org/fish/index.html>

Свежий блеск FVWM



Чрезвычайно удобный для воплощения самых смелых идей (вспомним хотя бы Metisse или Mezzo в Symphony), FVWM показывает новое оригинальное лицо. FVWM-Crystal объединяет ROX-Filer в качестве файлового менеджера; xterm,

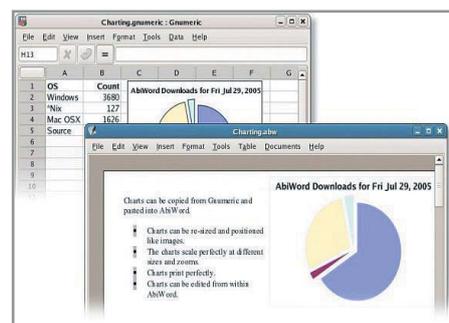
aterm, mrxvt или urxvt как эмуляторы терминала; осуществляет поддержку мультимедиапроигрывателей Music Player Daemon (MPD) и X MultiMedia System и некоторых полезных опций, таких как инструменты для отрисовки

фона рабочего стола (Esetroot, hsetroot или habak) и снятия снимков экрана. Безусловно, некоторые функции FVWM-Crystal унаследовала от FVWM, однако появилось и много нового. Меню, панели и диалоговые окна представлены в виде полупрозрачных элементов интерфейса, управление которыми теперь можно осуществлять с помощью гибких настроек, а также при желании менять им оформление. С клавиатуры посредством более 90 различных комбинаций можно манипулировать приложениями, например упомянутыми ниже проигрывателями. Кроме того, специальный скрипт на Python отыскивает приложения и выстраивает панель и меню для организации удобного доступа к ним. Также в FVWM-Crystal включена расширенная поддержка управления окнами терминалов, возможна работа с gettext и конфигурационными файлами.

Лицензия: GPL
Язык интерфейса: английский
Домашняя страница: <http://fvwm-crystal.berlios.de>

Этот кросс-платформенный текстовый процессор не нуждается в особом представлении, он давно нашел своих почитателей и теперь только продолжает радовать новыми технологическими наработками. Не обошлось без них и в новом релизе. AbiWord 2.4 может на лету схватывать орфографические и грамматические ошибки — правда, пока только в англоязычном тексте (между тем разработчики уверяют, что появление поддержки правописания других языков не за горами). Улучшена работа с графическими элементами: теперь текст мягко обтекает все неровности, уверенно заполняя прозрачные области изображения, и раду-

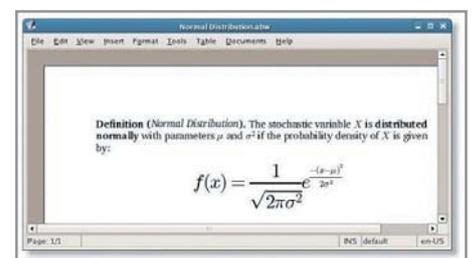
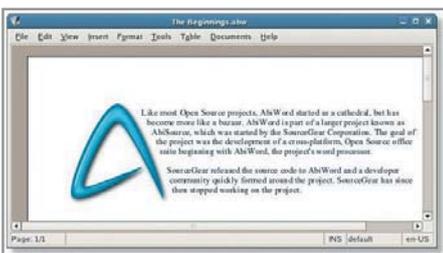
AbiWord 2.4



ет глаз. В новой версии AbiWord можно редактировать уравнения. Они вставляются с использованием синтаксиса MathML или LaTeX — здесь не обошлось без участия Mozilla, которая предоставила установщик шрифтов семейства Computer Modern, упрощающий управление математическими выражениями. Теперь процессор с легкостью справляется и с диаграммами Gnumeric — сложные достаточно просто скопировать в документ AbiWord,

а более простые можно создавать самостоятельно средствами самого процессора. Кроме того, разработчики улучшили фильтры для работы с форматами Word Perfect, XHTML, RTF и MS Word. И, конечно, главная особенность представленного релиза, которую мы умышленно оставили «на десерт», — поддержка нового открытого формата Open Document.

Лицензия: GPL
Язык интерфейса: английский
Домашняя страница: www.abisource.com





Содержание



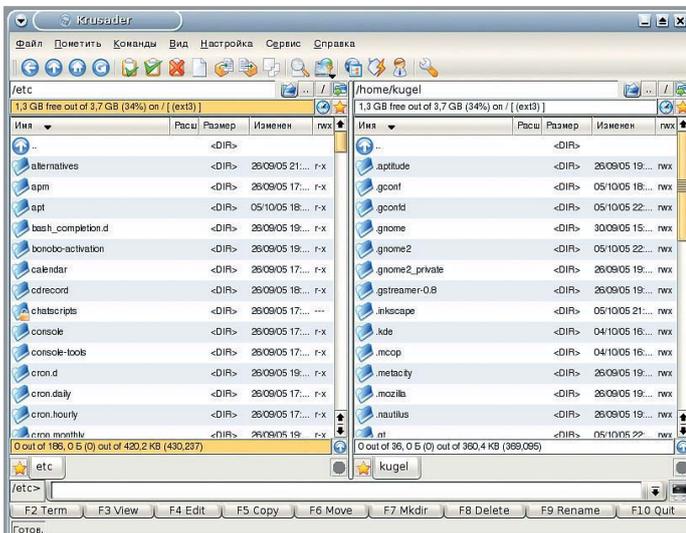
На диске, приложенном к журналу, находятся свежие версии самых интересных, на наш взгляд, программ. Объем необъятное, конечно же, нельзя, но мы постарались осветить темы мультимедиа и разработки, офисных приложений и администрирования. Естественно, не были обделены вниманием и игры. Кроме того, большинство упоминаемых в материалах номера программ вы сможете найти в разделе «Журнал». За подготовку диска благодарим сотрудников компании СИТКИТ. Именно они собрали этот диск, наполнив его исходными кодами программ из своего архива и обеспечив их всеми необходимыми зависимостями.

Krusader

Мало найдется программ, которые изобилуют таким большим количеством реализаций, как файловые менеджеры.

И в каждой из них разработчики пытаются ускорить рутинные операции по работе с файловой системой и усовершенствовать кажущийся почти идеальным интерфейс.

Интерфейс не утратил актуальности, и именно так представлен пользователю внешний вид Krusader.

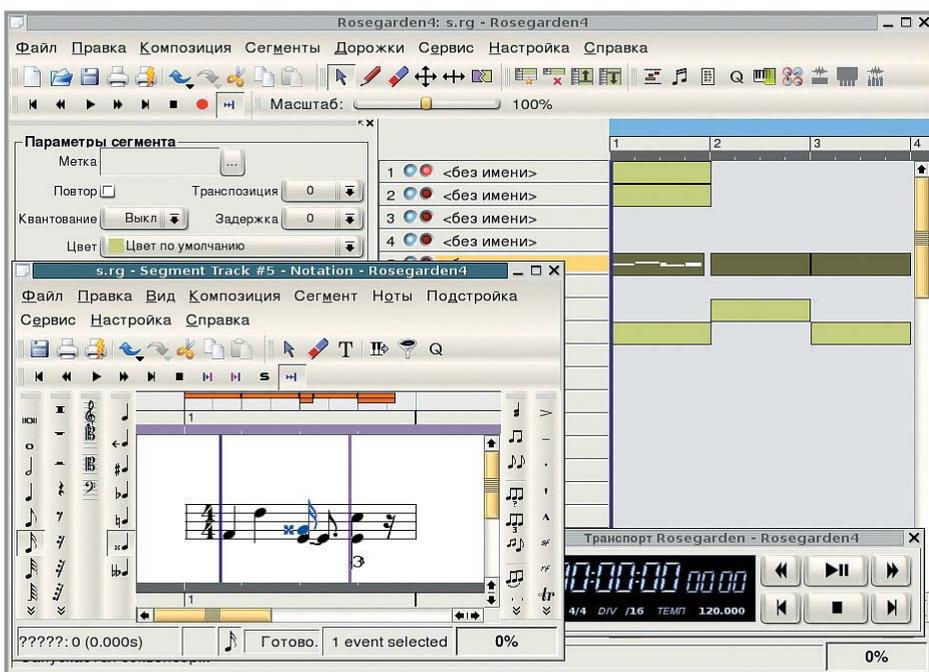


К его достоинствам следует отнести поддержку монтируемых файловых систем, FTP, современный модуль поиска, просмотр/редактирование текста, синхронизацию каталогов, поддержку сравнений файлов по содержимому, наглядный режим отображения файлов в виде разноцветных прямоугольников разной величины. Несмотря на все вариации и нововведения, классический двухпанельный

интерфейс не утратил актуальности, и именно так представлен пользователю внешний вид Krusader. К его достоинствам следует отнести поддержку монтируемых файловых систем, FTP, современный модуль поиска, просмотр/редактирование текста, синхронизацию каталогов, поддержку сравнений файлов по содержимому, мощные функции пакетного переименования и многое другое. Возможности по работе с архивами очень широки. Krusader поддерживает форматы TAR, ZIP, BZIP2, GZIP, RAR, ACE, ARJ и RPM. Очень удобно вводить консольные команды: специальный режим включается при нажатии на иконку в левом нижнем углу.

www.krusader.org

Rosegarden4 1.0

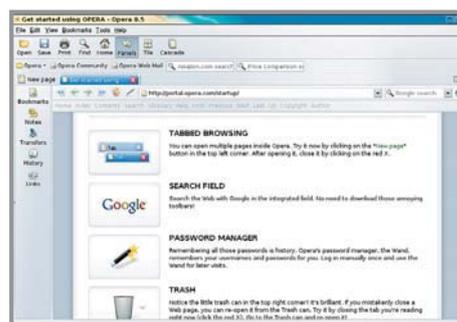


Долгое время пользователи Linux были обделены мощными музыкальными редакторами наподобие Cubase. Но потом появилась Rosegarden, четвертая инкарнация которой оказалась вполне зрелым продуктом. Программа выполняет функции MIDI-аудиосеквенсора и нотного редактора. Вывод звука осуществляется через ALSA, JACK и aRts. Поддерживаются плагины DSSI и, что особенно ценно, VSTi. На лету можно

применять эффекты LADSPA, обеспечивающие реверберацию, эквалайзер и (теоретически) все что угодно. Записанная MIDI-партия может быть сконвертирована в нотную запись. Интерфейс программы не вызовет затруднения у тех, кто уже работал с другими секвенсорами, а качественный перевод, выполненный Александром Прокудиным, не позволит запутаться в настройках. www.rosegardenmusic.com

Opera 8.5

До недавнего времени популярность браузера Opera уступала бесплатному Firefox. Теперь же появились все основания предполагать, что ситуация изменится: в конце сентября разработчики Opera Software сообщили, что отныне их продукт будет распространяться бесплатно. Opera — один из самых быстрых и гибких в настройке браузеров. Программа имеет встроенный блокировщик рекламных окон, позволяет на лету отключать и включать графику, а также может похвастаться технологией масштабирования веб-страниц. Еще одна изюминка — управление с помощью жестов мыши. Надо сказать, что Opera — это не только браузер, но и почтовый клиент со встроенным спам-фильтром, IRC-клиент, менеджер закладок, утилита для чтения RSS-новостей и даже клиент пиринговой сети BitTorrent. <http://opera.com>



Soviet Unterzögersdorf

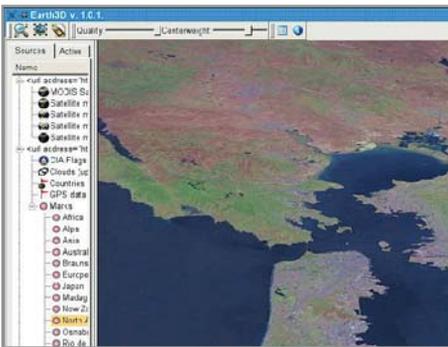
Весьма странно, что после развала СССР «Советский Унтерцугерсдорф» остается последней республикой, следующей заветам Ильича, и упорно пытается проложить дорогу в светлое будущее. Но вот беда, территория этого оплота социализма, расположенного где-то в Австрии, составляет всего 2,5 км², развитие промышленности сводится к перевозу гальки, а со всех сторон государство окружают «успешные» западные страны. Перед вами весьма странный квест, напоминающий «Штирлица». Удивление вызывает вполне достоверное воссоздание пост-социалистического быта с его разрухой, бесхозяйственностью и, вместе с тем, пафосным официозом. Как благополучные австрийцы могли такое со-

здать — непонятно. Тем не менее игра вовсе не мрачная и местами даже смешная. Специфический колорит ей придает озвучка. Такое ощущение, что попадаешь в ситуацию Арнольда Шварценеггера в фильме «Красная жара», где он говорит на ломаном русском языке. Здесь тоже главный герой говорит по-русски, к тому же вы можете читать титры на немецком языке. Если вам роднее и понятнее анг-



лийский, в разделе [language] конфигурационного файла впишите строку: translation=english. www.monochrom.at/suz-game/index_en.htm

Earth3D 1.0.3



Необыкновенную популярность снискал в последнее время сервис map.google.com, а также клиент для работы с ним — Google Earth. Но вот беда, программа существует только для Windows и Mac OS X. Впрочем, пользователям Linux отчаиваться рано. У них тоже есть возможность посмотреть на Землю с высоты полета спутника. Программа, реализующая это, называется Earth3D. Она отображает земной шар, который можно вращать, приближать и удалять. Дополнительные уровни данных представляют, например, флаги стран и географические названия. Используется информация с серверов НАСА, USGS и ЦРУ. Карты загружаются по требованию через Интернет, поэтому необходимо иметь быстрое соединение.

www.earth3d.org

Clam AntiVirus 0.87

Уберечься от вирусов поможет этот программный пакет — один из лучших антивирусов для Unix-платформ. Он состоит из гибкого и масштабируемого многопоточного демона, сканера, запускаемого в режиме командной строки, и инструмента автоматического обновления через Интернет. Основное назначение Clam AntiVirus состоит в интеграции с почтовыми серверами (сканирование вложений в почтовые сообщения). База данных по вирусам обновляется несколько раз в неделю — по уровню оперативности поддержки программа приближается к коммерческим продуктам. Основой Clam AntiVirus служит разделяемая библиотека, которую можно использовать и в других приложениях.

www.clamav.net



Battle for Wesnoth



Что такое «Веснот», за который нам предлагают сражаться в названии этой игры? Это столица сказочного королевства, существующего в мире, напоминающем место действия саги «Властелин колец». Как и полагается, он населен эльфами, гоблинами, людьми и т. д. Зачем же нам надо рисковать жизнью ради трона этого королевства? О перипетиях в семье царствующих особ в подробностях

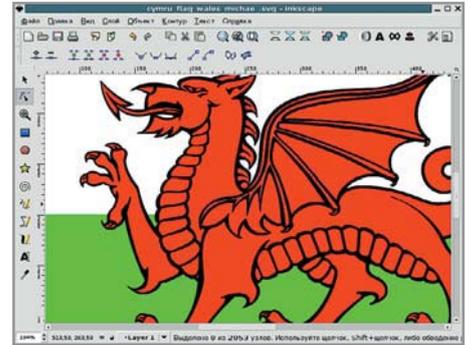
будет рассказано во вступительных титрах. Нужно лишь помнить, что мы мстим за коварно убитого короля-отца и пытаемся свергнуть незаконно присвоенную власть. В общем, наше дело правое.

Сразиться предстоит по правилам классических пошаговых стратегий с игровым полем, разбитым на шестиугольники. Игровой процесс целиком состоит из военных действий. Экономическая часть заключается в необходимости завоевывать опорные пункты — деревни. За каждую из них выдаются звонкие монеты, на которые можете нанять воинов, обладающих силой и специальными умениями (лечение, магия, скрытность и т. д.). При достижении определенного опыта юниты продвигаются по службе. Меняются и их внешний облик, и характеристики.

www.wesnoth.org

Inkscape 0.42.2

Среди некоммерческих векторных редакторов эта программа является, пожалуй, наилучшей. Она близка по возможностям к таким грандам как FreeHand и Adobe Illustrator и основана на формате SVG. Вы можете редактировать SVG-файл непосредственно в XML-редакторе или ри-



совать с помощью набора примитивов, кривых Безье или каллиграфического пера. Последний инструмент довольно интересен. Он позволяет динамически с помощью клавиш курсора менять толщину штриха и наклон пера. Неплохо развиты в программе средства заливки и градиентов. Освоить Inkscape за максимально короткий срок помогают неплохая справочная система и отображающиеся внизу главного окна контекстные подсказки.

www.inkscape.org

Список программ

Интернет

BitTorrent, Firefox, Opera, Psi

Веб-разработка

FKEditor, Gallery, phpBB, phpMyAdmin, SquirrelMail, Woodoo Chat

Администрирование

Ethereal, Ettercap, TightVNC

Обработка звука

Audacity, LAME, MP3Gain

Игры

DOSBox, Earth3D, Scorched3D, ScummVM, Torcs

Бонус

ClamAV, Inkscape, Kernel 2.6.12, Krusader, StarDict

Журнал

Cherokee, Gaim, GnomeICU, Grsecurity, Jack, Kaffeine, Licq, LIDS, Lighttpd, Linuxtv, Lsm, Mathopd, mplayer, MythTV, Nginx, NVU, Oolite, Rosegarden, SIM, Stellarium, Suz, Sylpheed, Tovid, Battle for Wesnoth, XAMPP, XawTV, Xine

Киберпространство и свобода в нем

В дебатах по поводу противостояния свободного и коммерческого ПО нередко звучат необдуманные и даже истеричные высказывания. Нам кажется, что в этом небольшом эссе достаточно взвешенно раскрываются корни некоторых проблем и показываются пусть тривиальные, но правильные методы борьбы с ними.

Существует большое количество разных лицензий на программное обеспечение. Некоторые из них лишают человека права копировать программу, изучать, модифицировать и распространять результаты изменений ее исходного кода.

Компьютерные науки имеют черты, сходные с кулинарией. Готовя пищу, мы используем рецепты. Будем ли мы строго соблюдать написанное в поваренной книге или нет, зависит только от нас. Можно исключить некоторые специи, потому что так посоветовал доктор, или изменить что-либо на свой вкус. Если получившееся блюдо понравится другу или подруге, мы даем им копию модифицированного рецепта. Представьте, что вас лишают права знать, из каких продуктов приготовлено блюдо, которое вам предстоит съесть, запрещают изменять рецепты и делиться ими. Что вы будете чувствовать? Что чувствует пользователь программы, находящийся в зависимости от владельца прав на нее? Мало того что, подписывая проприетарную лицензию, вы лишаетесь права делать копии программы для знакомых (которые с этической точки зрения этого заслуживают), вы еще и не можете быть уверенными в том, что используемое вами ПО не содержит вредоносного кода, ворующего ваши секреты. При использовании проприетарной программы нужно отказаться от мысли, что компьютер делает то, что хотите вы, а не третье лицо [1]. Несомненно, подобное состояние дел способствует нарастанию социальной напряженности, проявлениями которой в киберпространстве можно считать компьютерные вирусы, взломы серверов, от которых страдают простые пользователи, лишаясь информации, работы, средств к существованию, получающие серьезнейший стресс от осознания собственной уязвимости и бессилия [2].

Есть два полярных метода борьбы с угрозами, которые влечет за собой социальная напряженность:

- ▶ усиление контроля за технологиями;
- ▶ приобщение людей к культуре.

Усиление контроля за технологиями (например, *trusted computing*) может снизить риск, но, став инструментом давления на разработчиков и пользователей свободных операционных систем и прикладных программ, еще больше усугубит напряженность в обществе. Налицо замкнутый круг: одни стараются усилить системы защиты, другие пытаются их сломать [2].

Но выход есть. Нужно приобщать людей к культуре — то есть тому, что создано человеческим разумом и руками. Несомненно, движение за свободу киберпространства [3] — явление культуры. Используя свободное программное обеспечение, мы учимся взаимопомощи, учимся ценить свою свободу. Открывая путь в мир культуры, мы не делаем из человека изгоя (потенциального преступника), мы делаем из него члена общества, которому близки этические ценности и который понимает, что даже невинная на первый взгляд забава — взлом соседского компьютера с оставленным на нем шутивным сообщением — является неправильной и аморальной.

Данный подход к решению проблемы доказал свою жизнеспособность. Обучение детей карате (или другим боевым искусствам) не влечет за собой волны детской преступности, так как учителями в школах боевых искусств являются люди, для которых важны духовные и этические ценности (в любом боевом искусстве цель духовного совершенствования первична). В компьютерных сетях многих университетов нет жесткой политики безопасности, а есть сообщество людей, для кого моральные ценности и возможность учиться гораздо важнее выгоды, которую можно получить, своровав чужие секреты.

Развивая идеи свободы и взаимопомощи, мы заботимся о нашем будущем, будущем наших детей и внуков. |

[1] www.audio-video.gnu.org/audio/rms-speech-curitiba.ogg

[2] www.cs.berkeley.edu/~bh/hackers.html

[3] www.gnu.org