

RAILS

Почему Linux – лучший выбор **с.42**

PICASA

Новый фотоальбом от Google **с.12**

DVD ВНУТРИ



Все Ubuntu



Mandriva One



Inkscape и другие!

LINUX

FORMAT

ЧИТАЮТ ВСЕ ПОЛЬЗОВАТЕЛИ LINUX

**№9 (83)
сентябрь 2006**

ubuntu

READER AWARDS

Лучшее открытое ПО – ваш выбор! **с.40**

Неуклонный восход самого популярного дистрибутива Linux – чем же Ubuntu так хорош? **с.30**

LaTeX

Старый, но не устаревший **с.100**

ПРИВИВКА OPENSOURCE:

Открытое ПО с сертификацией для предприятий **с.46**



СТРЕЛЯЛКА

СВОИМИ РУКАМИ

Начните программировать на C++ вместе с нами! **с.70**



LINUX FORMAT

К вашим услугам...

Зачитавшись спецрепортажем про Ubuntu от Майка и Грэма мы задались следующим вопросом: «Говорят, что с одного из африканских языков Ubuntu переводится как «гуманизм». Но что же оно значит на самом деле?» Вот что нам ответили в Команде LXF:



Пол Хадсон
Слово «ubuntu» ничего не значит. Его выбрали, так как оно созвучно «Red Hat».



Грэм Моррисон
«Единственное богатство, которое останется с вами навсегда – это то, что вы отдали» (Марк Аврелий).



Эфрейн Эрнандес-Мендоса
Мы – мир! Мы – дети!



Майк Сондерс
Ubuntu значит «апельсиновый Gnome с начинкой из сладкого Debian».



Ребекка Смолли
ubuntu суш. 1. гуманизм 2. цветовая схема, популярная в 1970-х 3. Самодовольное чувство, возникающее у кодеров после установки Linux.



Эндрю Григори
По-африкански это значит «акулы с лазерными лучами на головах» Осторожнее!



Марк Бейн
На самом деле ubuntu значит «никогда не говори, что используешь Windows».



Нейл Ботвик
Ubuntu – это древнеафриканское слово, означающее «я не смог настроить Slackware».



Энди Ченел
Ubuntu значит «коричневый». Он мог бы стать ближе к дошколятам, если бы назывался Ubuntu – это было бы забавнее.



Нам один год!



Уважаемый читатель! Вы держите в руках юбилейный двенадцатый номер нашего журнала. Да, да... Linux Format исполнился ровно один год

«Первый в России ежемесячный журнал на русском языке, целиком посвященный операционной системе Linux» был официально представлен в ходе выставке LinuxWorld 2005 в Москве, в самом начале осени. По удачному стечению обстоятельств дебют оказался двойным (LinuxWorld также проходил у нас в стране впервые) и, как теперь можно с уверенностью сказать, удачным. За LinuxWorld последовал LinuxLand/SofTool'05, где состоялась презентация Linux Format для широкой, не обязательно знакомой с Linux, публики. Мы считаем это событие не менее важным, поскольку первая и главная из миссия журнала – пропаганда свободного ПО в России, а миссионерством, как известно, полагается заниматься среди аборигенов, а не среди христиан.

Время шло, и журнал не стоял на месте. Мы, как это ни банально звучит, постоянно стремились улучшить выпускаемый нами продукт: избавиться от опечаток и других «детских болезней», привлечь к работе над материалами отечественных авторов (ведь кто знает ситуацию с Linux в России лучше, чем мы сами?), разместить как можно больше программ на прилагаемом к журналу диске и т.д. Конечно, все эти усовершенствования были бы невозможны без активного сообщества читателей, которые дискутировали на форуме, присылали нам письма на редакционный ящик, высказывали пожелания на выставках и конференциях. Спасибо вам – Linux Format в его текущем виде, в том числе, и ваша заслуга!

Один год – далеко не преклонный возраст и, конечно же, мы не собираемся расслабляться. Linux не стоит на месте и Linux Format будет прилагать все усилия, чтобы не отстать от времени – с вашей, разумеется, помощью. Уверен, мы еще встретимся на этой странице, чтобы отметить пяти-, десяти-, ... -летие LXF. Ну а пока – приятного чтения!



С уважением,
СИНИЦЫН ВАЛЕНТИН
главный редактор LinuxFormat в России

КАК С НАМИ СВЯЗАТЬСЯ

Письма для публикации:

letters@linuxformat.ru

Подписка и предыдущие номера:

subscribe@linuxformat.ru

Техническая поддержка:

answers@linuxformat.ru

Проблемы с дисками:

disks@linuxformat.ru

Общие вопросы: info@linuxformat.ru

Website: www.linuxformat.ru

Адрес редакции:

Россия,

Санкт-Петербург, ул. Гончарная, 23, офис 54

Телефон редакции: (812) 717-00-37

Дополнительная информация на стр.118

МИССИЯ ЖУРНАЛА

- Пропаганда свободного ПО в России
- Продвижение решений с открытым кодом в бизнес-сообществе
- Поддержка российского Open Source сообщества
- Организация трибуны для разработчиков свободного ПО
- Обратная связь между разработчиками и потребителями ПО



Содержание

LINUX
FORMAT

Добро пожаловать в *LinuxFormat* – ваш гид в мире Linux!

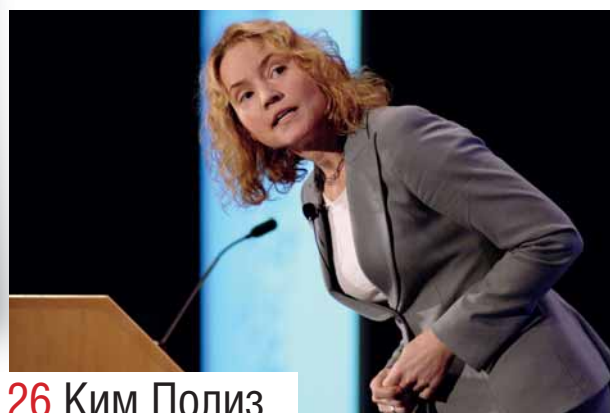
LXF9(83) СЕНТЯБРЬ 2006

»» ЧИТАЙТЕ В ЭТОМ ВЫПУСКЕ



UBUNTU НАВСЕГДА! 30

- 13** **Обзор Kopete 0.12.0**
Почему вы должны попробовать его прямо сейчас
- 40** **LXF Reader Awards 2006**
Результаты голосования!
Ну что же ты, Crack Attack?
- 64** **Учебник LVM**
Виртуальные разделы: стань хозяином своего жесткого диска
- 100** **Учебник LaTeX**
Истинная TeXнология – по вашим многочисленным просьбам



26 Ким Полиз

МЫСЛИ
ВСЛУХ

«Говорят, что в Linux мало игр — но он сам может быть не менее интересен и уж заведомо более полезен»



51 Евгений Балдин о новом подходе к установке

«Красота – страшная сила, особенно если мы хотим познакомить со свободным ПО как можно больше людей»



36 Марк Шаттлворт о прелестях Ubuntu



08 М-р Картер притаился в углу

04 НОВОСТИ

Все обо всем

08 COLD WAR

Старая добрая игра в прятки

10 TURBOLINUX FUJI

Популярен в Японии

11 TREND MICRO SERVER PROTECT

Проверка на вирусы в реальном времени

12 PICASA 2

Новый фотоальбом от Google

13 КОРЕТЕ 0.12.0

Вкусный IM-клиент

14 VIM 7.0

Наконец-то!

16 DISTROWATCH

Мокроносый дистрибутив

18 HOTPICKS

Лучшее, что может предложить Web

24 ЧТО ТАКОЕ... GEGE?

Амбиции разработчиков Gimp

26 SPIKESOURCE

Сертификация для предприятий

30 UBUNTU НАВСЕГДА

Почему он так хорош

38 СНОВА АРТИТУДЕ

Взгляд из командной строки

40 LXF READER AWARDS

Ваш голос имеет значение

42 RUBY ON RAILS

Web-разработка – это просто

44 БРОНИРОВАННЫЙ LINUX

AppArmor и с чем его едят

48 RIP – LIVE CD СВОИМИ РУКАМИ

Немного теории для удачного старта

52 ПЕРВЫЕ ШАГИ

Компьютер для ребенка

56 INKSCAPE

Танец в SVG

60 OOO BASIC

Советы бывалых

64 РАЗБИВКА HDD

Получите максимум – используйте LVM

68 XDMCP ЗА 10 МИНУТ

Тонкий клиент на дому

70 3D-ИГРА

Перемещаемся по нашему миру

96 Hardcore Linux

Аутентификация пользователей

78 PYTHON PRO

Снова о web-программировании

82 QT/KDE

Настоящее приложение KDE

86 MAXIMA

Упрощение выражений

91 ROOT

Нет, это не то, о чем вы подумали

96 UNIX API: СОКЕТЫ

Самое удачное изобретение BSD

100 LATEX НОВАЯ СЕРИЯ

Культовая система верстки

104 WINK

Создаем screencast

106 ВОПРОС? ОТВЕТ!

Мы вам поможем

112 ДИСК LINUX FORMAT

Ubuntu 6.06 + Mandriva One

119 ПОДПИСКА!

Полная информация для жителей России и СНГ

На диске

DVD полный новейших программ **112**



>> НА ДИСКЕ

UBUNTU 6.06 Релиз Dapper Drake с долгосрочной поддержкой – специальная версия, включающая Kubuntu, Xubuntu и дополнительные программы!

MANDRIVA ONE Живая версия одного из самых популярных настольных дистрибутивов. С возможностью установки!

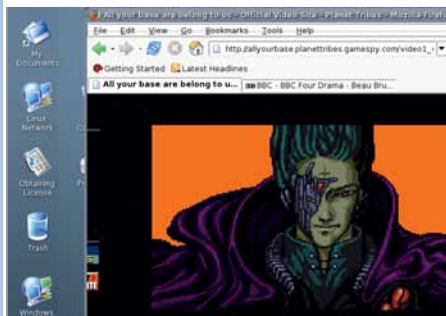
INKSCAPE 0.44 Новая версия векторного графического редактора. Читайте учебник Дмитрия Кирсанова!

RIP Проблемы с системой? RIP LiveCD спешит на помощь!

ИГРЫ Многопользовательская стрелялка Nexiuz и комедийный футбол Slam Soccer



Перед использованием диска ознакомьтесь с инструкцией и вырежьте обложку своего DVD на **стр. 113–114**



10 Turbolinux Fuji



44 Защитим Linux

ПОДПИШИСЬ СЕГОДНЯ!

LINUX FORMAT

Подробности на сайте
www.linuxformat.ru



Главные новости

• Java открывается • Greenphone от Trolltech • СОЛО для Unix • Рэймонд призывает к компромиссу с закрытым кодом

Зеленый телефон от Trolltech

Мобильный телефон специально для разработчиков открытого ПО



Норвежская компания

Trolltech анонсировала выпуск Qtopia Greenphone – «первого открытого Linux-устройства для разработчиков приложений». Новинка была продемонстрирована на выставке LinuxWorld в Сан-Франциско и должна поступить в продажу в ограниченном количестве уже в середине сентября.

Что же такое Greenphone? В первую очередь, это мобильный телефон стандарта GSM/GPRS, разработанный китайской фирмой Yuhua Teltech на базе процессора XScale. Устройство имеет небольшие размеры (приблизительно 10x5x1,5 см), оснащено цветным QVGA-дисплеем с разрешением 320x240 пикселей, 64 МБ оперативной и 128 МБ флэш-памяти, расширяемой посредством карт SD. Помимо этого, Greenphone снабжен 1,3-мегапиксельной камерой, поддержкой Bluetooth и портом mini-USB.

Основной интерес представляют начинка и позиционирование устройства. Greenphone – это не конечный продукт, а часть комплекта разработчика (SDK), в который, помимо прочего, входит платформа Qtopia Phone Edition и встроенный дистрибутив Linux неизвестного пока производителя, под управлением которого и работает телефон. Наверное, излишне говорить, что все компоненты SDK (кроме частей, ответственных за поддержку baseband-процессора) распространяются под дуальными или открытыми лицензиями. Trolltech прикладывает все усилия, чтобы Greenphone был доступен всем разработчикам, независимо от их статуса. Как утверждают в компании, разнообразные условия лицензирования Greenphone учитывают как интересы телекоммуникационных компаний, так и требования разработчиков Open Source. Ожидается, что благодаря Greenphone последние смогут испытывать свои встроенные

приложения на настоящем оборудовании: все, что для этого требуется – подключить телефон к компьютеру, используя упомянутый выше порт mini-USB и обновить «прошивку» (firmware). Это, в свою очередь, должно привести к образованию вокруг Qtopia Phone Edition активного сообщества. «Скажу вам по секрету: отдать телефон в руки разработчиков Open Source – это именно то, чего я хочу», – сообщил технический директор компании Бенуа Шиллингс (Benoit Schillings).

<http://www.trolltech.com/gppress>



Sun открывает Java

Платформы Java 2 Standard и Micro Edition будут доступны по одобренной OSI лицензии



14 августа в Сан-Франциско,

где в эти дни проходила выставка LinuxWorld, состоялась небольшая пресс-конференция, в ходе которой представители компании Sun Microsystems Ричард Грин [Rich Green, вице-президент Sun по программному обеспечению], Лори Толсон [Laurie Tolson, вице-президент по продуктам и программам для разработчиков] и Алан Бреннер [Alan Brenner, вице-президент по мобильным и встроенным системам] детализировали планы по открытию Java.

В разряд открытого ПО будут переведены платформы Java SE (Standard Edition) наряду с Sun-реализацией Java ME (Micro Edition), предназначенной для встроенных устройств. Java ME и первые крупные компоненты Java SE будут опубликованы как

Open Source уже к концу 2006 года. Не спеша назвать конкретные наименования, компания отметила двух вероятных кандидатов – компилятор javac и виртуальную машину HotSpot VM. Остальные части JDK будут открыты в начале 2007 года. Как и в случае с Solaris, открытость продукта будет не стопроцентной – права на некоторые участки кода принадлежат другим компаниям. Такие компоненты будут распространяться в двоичной форме. Возможно, со временем их удастся переписать или заменить наработками проектов GNU Classpath или Apache Harmony, однако в настоящий момент этот вопрос остается открытым, точно так же как и вопрос выбора лицензии – известно лишь, что она будет одобрена OSI. Можно предположить, что Sun выберет разработанную для Solaris лицен-

зию CDDL, хотя официальных заявлений на этот счет пока сделано не было.

«Самое важное, что мы не занимаемся этим в изоляции. Мы хотим перенять опыт успешных открытых проектов и приглашаем сообщество Java-разработчиков подключиться к процессу. Без помощи со стороны сообщества, Java не сможет в полной мере воспользоваться преимуществами открытой модели разработки», – подчеркнула Джин Эллиотт [Jean Elliott], директор Sun по маркетингу решений для разработчиков. Для нас же с вами это значит, что Java сможет, наконец, быть включена даже в самые «разборчивые» дистрибутивы, и Ричард Столлмен не скажет ничего против.

<http://community.java.net/jdk/opensource/>

КОРОТКОЙ СТРОКОЙ

- Компания Mandriva анонсировала планы по включению трехмерного рабочего стола (XGL или AIGLX на выбор) в Mandriva Linux 2007.
- Зак Расин (Zack Rusin) из компании Trolltech объявил о начале работы над еще одной архитектурой трехмерного рабочего стола – Glucose.
- Компания Real Software реализует в Linux легальную поддержку форматов Windows Media. Это стало возможным благодаря спецификациям, полученным ею от Microsoft в ходе дела о недобросовестной конкуренции.
- Вышли в свет KDE 3.5.4 и Freespire 1.0
- В России начались продажи SUSE Enterprise Desktop/Server 10
- Команда Firefox направилась в Редмонд, чтобы вместе с инженерами Microsoft обеспечить совместимость Firefox и Windows Vista.

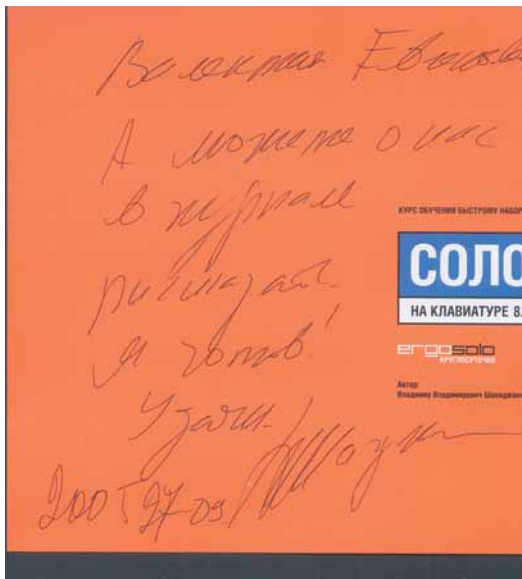
СОЛО для Unix



К создателю программы «СОЛО на клавиатуре» Владимиру Владимировичу Шахиджаняну, как, впрочем, и к любой другой яркой личности, можно относиться по-разному. Зачастую мнения оказываются диаметрально противоположными, однако и сторонники, и оппоненты Владимира Владимировича должны признать, что «СОЛО», фактически, является синонимом «программы для обучения слепой десятипальцевой печати», по крайней мере, на территории бывшего СССР. Число людей, прошедших обучение по методике Шахиджаняна, оценивается ее авторами в двести тысяч человек. Внушительная величина, однако она обещает стать еще больше — компания «ЭргоСОЛО» (www.ergosolo.ru) выпустила Unix-версию «СОЛО на клавиатуре»!

В момент подготовки этого материала к печати для загрузки с сайта компании была доступна версия Beta 3. Программа базируется на дисковой версии «СОЛО на клавиатуре» 8.8 для Windows и разрабатывается Дмитрием Дмитриевичем Стасюком. Список зависимостей минимален: Unix-подобная операционная система с X-сервером не ниже X11R6, Gtk+ 2.4.0, а также GtkHTML 2.6.0 или более поздней версии. Под термином «Unix-подобная операционная система» в настоящее время понимается Linux, но к моменту чтения вами этих строк ситуация может измениться: в ближайших планах разработчиков — поддержка FreeBSD 5.4 и 6.0, а в перспективе — Mac OS X, OpenBSD, OpenSolaris, QNX и даже GNU/

Вот он — диск с дарственной надписью.



Hurd. Перечень дистрибутивов, для которых доступны бинарные пакеты, также впечатляет: в него уже входят ASP Linux 11, Fedora Core 5, SUSE 10, Mandriva 2006; Debian 3.1, Ubuntu, ALT Linux, Slackware 10.2, Arch и Gentoo планируются в недалеком будущем. Для «СОЛО» очень важна широкая поддержка различных дистрибутивов, поскольку став родным приложением Unix, продукт не перестал быть проприетарным, так что собрать его из исходных текстов под свою систему не получится. Однако, поддержка множества различных дистрибутивов, как справедливо заметил Алексей Федорчук, может отрицательно сказаться на качестве пререлизного тестирования. Можно порекомендовать разработчикам распространять «СОЛО» в виде пакетов *Autopackage* (www.autopackage.org) или простого tar-архива с бинарными файлами, благо зависимостей не так уж и много.

Вернемся к возможностям программы. Как утверждают авторы, «СОЛО» для Unix

использует для хранения результатов тот же самый формат, что и Windows-версия (8.8 или 8.1 с возможностью автоматической конвертации), а значит, вы можете спокойно переезжать в Linux, не боясь потерять «все, что нажато непосильным трудом». Дождитесь лишь финальной версии — если только заранее не готовы помочь в тестировании, оттянув на себя некоторый процент ошибок. Функционал программы может быть расширен с помощью подгружаемых модулей; не возбраняется и подключать новые курсы по мере их появления.

Главный редактор нашего журнала случайно встретился с Владимиром Владимировичем на прошлогодней выставке SofTool. На память об этой встрече у нас остался диск «СОЛО на клавиатуре» 8.8 с дарственной надписью и предложение рассказать об этой программе на страницах журнала. Быть может, пришла пора им воспользоваться?

www.ergosolo.ru

Алексей Федорчук

Обеспокоен недавним заявлением м-ра Реймонда.



КОММЕНТАРИИ

Linux: открытость и проприетаризм

“**Может ли Linux включать в себя какие-то не вполне свободные элементы? В отношении дистрибутивов ответ дан явочным порядком: трудно найти среди них такой, в котором не было бы ни одного пакета, распространяемого под не вполне свободной, в понимании FSF, лицензией.**

Ядро же Linux до настоящего времени проприетарных компонентов не содержало. Закрытые драйверы устройств «прикручивались» к нему в качестве внешних модулей «от производителя». И вдруг из уст Эрика Реймонда прозвучали слова о необходимости компромисса с разработчиками коммерческих программ. То есть — о возможности включения в ядро проприетарных драйверов устройств. Может ли быть такое, и если может, то — в какой форме?

Встраивание в ядро закрытых фрагментов видится нереальным: это перевернет всю схему его разработки, создававшуюся полтора десятилетия. Включение в ядро открытого, но не распространяемого свободно кода теоретически возможно. Однако это вызовет изменение лицензирования ядра и, скорее всего, приведет к мгновенному его разветвлению на две части — собственно свободную и свободную ограниченно. А если к сегрегации дистрибутивов добавится еще и сегрегация на уровне ядра — картина получается апокалиптическая. Так что остается надеяться, что компромисс между свободой и проприетаризмом, если и будет достигнут, примет какие-то иные формы.

alv@posix.ru



Реймонд призывает к компромиссу с закрытым кодом

Иначе Linux рискует оказаться на обочине истории, считает экс-президент OSI



Эрик Реймонд (Eric Raymond), одна из ключевых фигур мира Open Source и бывший президент Open Source Initiative (OSI), утверждает, что Linux следует менее настороженно относиться к программам с закрытым кодом. В своем выступлении на конференции LinuxWorld он отметил, что отсутствие полноценной поддержки современных устройств, таких как *iPod*, не добавляет Linux привлекательности в глазах конечного пользователя. Если так пойдет дальше, то с течением времени отставание Linux (в смысле поддержки новых устройств) от платформ с коммерческими драйверами

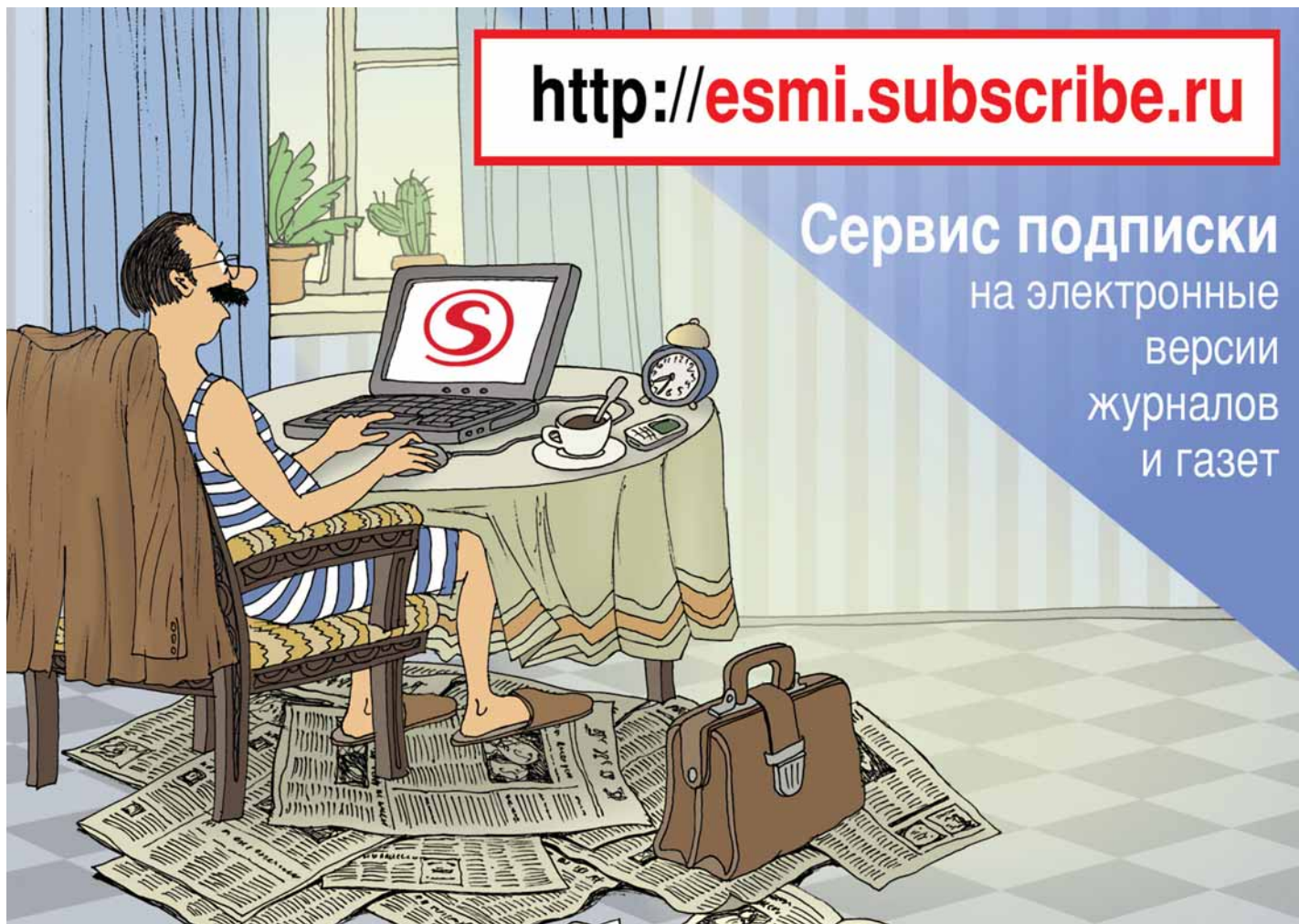
будет становиться все более заметным. Реймонд также назвал работу с закрытыми драйверами «необходимым компромиссом». Свое согласие с этим мнением высказали Йон «Мэддог» Холл (Jon «maddog» Hall), представители Intel и Google.

Комментарий LXF: эта новость, как и другие высказывания подобного рода, мгновенно всколыхнула Linux-сообщество, как в России, так и за рубежом. Пользуясь случаем, мы хотели бы напомнить всем, что компьютеры создавались для решения задач, стоящих перед их владельцем, а не как орудие какой бы то ни было идеологии, поэтому идея «если кое-что не работает в

Linux — это беда Linux, а не кое-чего» имеет полное право на существование. Кроме того, свобода (software freedom), о которой так много говорится на страницах нашего (и не только нашего) издания означает именно свободу, а не «осознанную необходимость». И если вы голосуете за пресловутый «свободный мир», то должны признавать за производителями ПО право делать с их кодом все, что им вздумается — в частности, закрывать от посторонних глаз. Если вы считаете, что производитель обязан сделать для вас то-то и то-то — оставайтесь: вы требуете не свободы, а новой тирании.

<http://esmi.subscribe.ru>

Сервис подписки
на электронные
версии
журналов
и газет



СИСТЕМНЫЙ администратор

Клонировем Windows с помощью Symantec Ghost

Насколько неуязвима ваша беспроводная сеть?

Active Directory вместо рабочей группы

Настраиваем DSPAM – ваш личный спам-фильтр

Как спасти данные, если отказал жесткий диск

Модифицируем BIOS

Все ли возможности ClamAV вы используете?

Что важно знать об IP-телефонии

Админские сказки

www.SAMAG.ru

В «Системном администраторе» вы не прочтете о:

- котировках валют
- сплетнях
- погоде
- политике
- развлечениях



В вашем распоряжении:

- опыт лучших IT-специалистов
- новые идеи и полезные советы
- самые эффективные решения в области системного и сетевого администрирования



Подпишитесь сейчас!

Роспечать – 20780, 81655
Пресса России – 87836
Online-подписка – www.linuxcenter.ru

Время подписки
ограничено!

Обзоры



Новинки программного и аппаратного обеспечения в описании наших экспертов

ЧТО ДЕЛАТЬ?



Евгений Балдин наслаждается произведениями Горького: «Что я сделать могу для людей? – вскричал Данко и вырвал из груди своей сердце..., а потом упал и умер.»

Для людей можно сделать много чего, но в данном случае конкретизируемся. Например: «Что можно сделать для развития сообщества Open Source?» Есть очевидные ответы на этот вопрос, такие как: писать программы и выпускать их под открытой лицензией, заниматься переводом уже существующих проектов, создавать свободную документацию. Но это технические решения.

Пришло время для политических, социальных решений. Точнее не пришло — оно всегда было, просто теперь имеющиеся технические возможности могут обеспечить развитие идей свободы за пределами сообщества.

«СООБЩЕСТВО, КОТОРОЕ НЕ РАЗВИВАЕТСЯ – ГИБНЕТ. ЭТО ЗАКОН ПРИРОДЫ: ЛИБО ВВЕРХ, ЛИБО ВНИЗ.»

Сообщество, которое не развивается – гибнет. Это закон природы: либо вверх, либо вниз. Развитие возможно только за счёт обучения молодого поколения, потому что стариков не переубедишь. На бога, государство и за границу надеяться бесполезно, поэтому наиболее действенный метод – это личный пример. Понятно, что это не просто – школа это та ещё структура, но другого пути нет. Не надо заниматься этим всё своё время: кружок, дополнительные занятия, помощь при администрировании. Только это должно быть постоянным, чтобы был определенный день и час, в который можно прийти и задать вопрос или спросить совета. Подобные начинания держатся на людях и вы вполне можете оказаться среди них.

ЧТО НОВОГО?

08 COLD WAR

Прекрасная графика, скрытые перемещения и рентгеновские камеры – лучшее, что может предложить Linux скучающим по *Splinter Cell*

10 TURBOLINUX 11

Лучший дистрибутив из Японии – теперь на 50% больше пиццания во время загрузки.

Исследует наш штатный новичок **Эндрю Грегори**

11 TREND MICRO SERVER-PROTECT 2.5

Linux не боится вирусов, так почему бы не использовать его в качестве отстойника для чересчур нежных Windows-клиентов?

12 PICASA 2

Любима пользователями Windows на зависть всем остальным. Но все меняется – Google портировал *Picasa* в Linux и теперь мы тоже можем любить ее. Ну, почти...

13 КОРЕТЕ 0.12.0

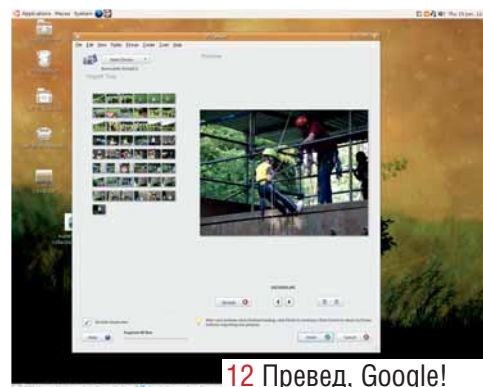
Грэм снова дал претенденту от KDE более высокую оценку, чем конкуренту от Gnome. Ну а чего вы ждали?

14 VIM 7.0

В череде долгостроя Open Source *Vim 7* занимает почетное третье место. Ну ладно, ладно, занимал. Знакомьтесь с новой версией!



08 Бах, бах!



12 Превед, Google!

НАШ ВЕРДИКТ: ПОЯСНЕНИЕ

Все продукты оцениваются по 10-балльной шкале. 4 обычных параметра оценки: возможности, производительность, простота использования и соотношение «цена/качество», но для свободного ПО последний параметр может быть заменен на оценку документации. Независимо от набора категорий, мы



всегда вычисляем общий рейтинг, подводящий итог нашим высказываниям.

Продукты, выделяющиеся из основной массы, получают престижную награду *Linux Format Top Stuff Award*. Выбираются самые-самые – и только лишь высокая оценка здесь еще ни о чем не говорит.

Для тестирования серверов используется наша собственная разработка, LFXBench 2004, состоящая из 4 основных подсистем: Multi CPU, Single CPU, RAM и Hard Disk. Оценки усредняются и дают общий рейтинг.

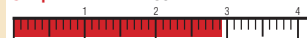
1 очко означает, что тестовый компьютер равен нашему эталону: Pentium 4 1,8ГГц, 512 Мб оперативной памяти и жесткий диск IDE. 2 очка означает, что он вдвое быстрее.

Все тесты выполняются под управлением Red Hat Enterprise Linux 3 AS на соответствующей платформе (x86, AMD64, Itanium). Код компилируется при помощи GCC, если не указано иное.

РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

MULTI CPU	6.07
SINGLE CPU	3
RAM	2.17
HARD DISK	0.46

ОБЩИЙ РЕЙТИНГ 2.93



ИГРА «В ПРЯТКИ»

Cold War (Холодная война)

Мы отправили **Грэма Моррисона** за железный занавес: пусть посмотрит леденящий кровь триллер от Linux Game Publishing.

САМОЕ ГЛАВНОЕ

Бей и беги, пока не обнаружен. Больше всего похоже на Splinter Cell или Metal Gear Solid под Wine.

- **РАЗРАБОТЧИК:** Linux Game Publishing
- **САЙТ:** linuxgamepublishing.com
- **ЦЕНА:** около \$55



Десятилетия, проведенные в мрачной тени холодной войны, никому, похоже, не мешают использовать эту тему в качестве фона для книги, фильма или видеоигры. Но разработчик последнего релиза от Linux Game Publishing (зашифруем ее как LGP) знает об этом предмете больше, чем мы, благополучно отсидевшиеся за Георгианским песчаником в Бате, Англия [местоположение офиса английского издания LXF, — прим. ред.]. Основанная в 2001 г. Mindware Studios приветствует вас из центра шпионажа времен холодной войны: из Праги, Чехия.

Mindware выпустила компьютерную игру *Cold War* для ничего не подозревающей общественности в конце прошлого года. Между делом LGP в рекордный срок конвертировала ее для работы в вашей любимой подрывной ОС, при единственном условии: ваш жесткий диск потяжелее на 1.7 Гб.

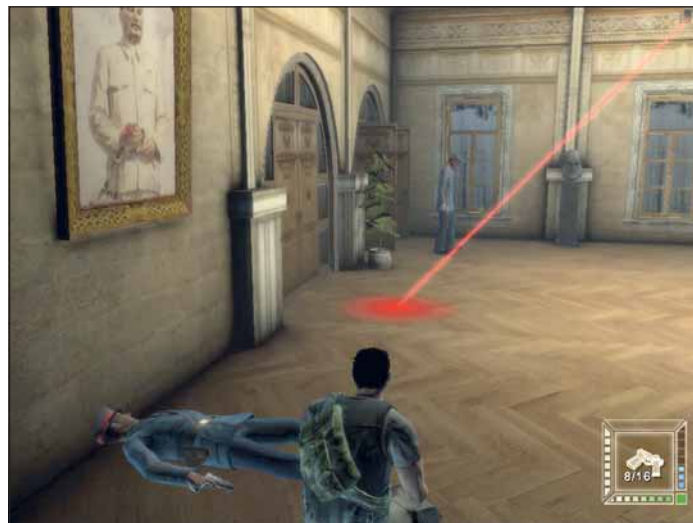
Посмотрев изрядное количество фильмов про Джеймса Бонда, мы поднатерели в

распознавании подсознательных моментов и стиля, присущих эпосу о холодной войне. Игра *Cold War* ни в коей мере не разрушает готовые штампы. Место действия где-то в морозной России? Есть. Загнивающие символы тоталитарного режима? Есть. Вы брошены в тюрьму поклонниками Кафки за преступление, которого не совершали? Есть. Лейтмотив в исполнении балалайки? Нет, этого нет. Но все равно, картина ясна. Наш главный герой — журналист по имени Мэтью Картер [Matthew Carter], проникший на тайное заседание правительства в расчете на Пулитцеровскую премию.

Именно здесь мы вступаем в игру, после короткого знакомства с управлением в процессе протаскивания Картера в Мавзолей Ленина (есть!), чтобы подслушать разговор.

Управление осуществляется привычной комбинацией клавиш WASD на клавиатуре, а мышь, как всегда, выбирает действия. На экране — вид от третьего лица; графика исключительная (по крайней мере, на современном оборудовании), благодаря движку *Quake 3*. Рендеринг каждой сцены превосходен, используется мягкий свет с умеренным сиянием на ярко освещенных объектах — выглядит практически на уровне технологий с высоким динамическим диапазоном.

Разработчики постарались воссоздать каждую сцену действия в игре с максимумом реализма: например, не имея фото-



Реализм обстановки и коммунистическая атрибутика создают завораживающую атмосферу... только враги туговаты на ухо.

графии Мавзолея Ленина, они взяли за модель погребальный склеп бывшего коммунистического лидера Чехословакии Клемента Готтвальда (Klement Gottwald) в Праге. Создается ощущение настоящего пребывания в помещении, восприятия текстур строения и гробницы, а также подземных лабиринтов и проходов. И это — один из наиболее привлекательных аспектов игры, характерный для каждой локации. Пусть даже игра не всегда совершенна, сознание того, что каждое место действия смоделировано на основе реального существующего, разжигает воображение.

остаться незамеченным, приходится держаться в тени, прячась за подходящими объектами. В игре имеются три степени приседания, выбрать которые можно с помощью колесика мыши; чем ближе к земле, тем лучше. Когда вы приближаетесь к определенным объектам, появляется меню действия (action menu), и нужно срочно нажать на правую кнопку мыши, чтобы совершить какой-то маневр, например, послать врага в нокаут. Скоро вы подберетесь к часовым у Мавзолея и вышибете из них дух, прежде чем они смогут что-либо предпринять.

Не забудьте, что Картер умеет создавать собственную технологию, подбирая подручные предметы. Вы платите очками Tech Points за создание каждого объекта, а на ранних стадиях игры можно разве что изготовить пластиковые пули из пластиковых бутылок. Даже при наличии оружия Картер не спешит убить кого-то по-настоящему.

Но и в целом идея Tech Points не работает, отчасти из-за того, что вам редко приходится рассчитывать на свои доморощенные решения.

Сюжет становится понятен сразу же, как только наш герой попадает в ленинскую гробницу. Вы — заложник международного заговора, который необходимо раскрыть до получения биле-

Игра в прятки

Под ногами солдат в полушубках, патрулирующих периметр Мавзолея, по-настоящему скрипит снег. Однако вы не можете выхватить оружие и отправить их в небытие, ведь это шпионский триллер, а Мэтью Картер — личность совестливая. Индикатор в правом нижнем углу экрана показывает, насколько сильно вы бросаетесь в глаза. Чем лучше вас видно, тем вероятнее, что какой-нибудь плохой парень из игры вас подстрелит. А вам этого не надо, тем более, что телосложение у Картера хиловатое: две-три пули — и он покойник. Чтобы



Отличное освещение сопровождается мягким сиянием (впервые применено в сиквеле Deus Ex), сглаживающим темные края.

СКРЫТНОСТЬ РЕШАЕТ ВСЕ



1 Лучше всего общаться с часовыми, напад на них сбоку или сзади.



2 Вы всего на шаг от часового, а он и ухом не ведет. Невысок его AI.



3 Избавьтесь от часового, выбрав в меню действий Knock Out. *Do svidaniya!*

ТРЕБОВАНИЯ

Требования *Cold War* весьма высоки, особенно к видеокarte.

Рекомендуемый минимум:

- Процессор 1.7 ГГц x86
- 384 МБ ОЗУ
- 1.6 Гб жесткого диска
- DVD-ROM
- Видеокarta с аппаратным ускорением и 128 МБ памяти
- OSS/ALSA-совместимая звуковая карта

та до Сибири в один конец. Хотя, по нашему мнению, разработчики упустили шанс устроить действие в парке Горького, нам весьма понравились сцены в мрачной тюрьме на Лубянке (есть!). Именно здесь мы впервые вошли во вкус игры, когда Картер пытался сбежать от своих тюремщиков из КГБ.

Вы быстро научитесь красться по коридорам, перед каждой пробежкой ожидая, пока отвернется караульный. Придется даже втыкать в лежащих без сознания

караульных шприц, чтобы они подольше пробыли в этом состоянии, дав вам возможность продолжить свою деятельность. Цели всегда ясны. Вариантов немного: иногда вам может понадобиться отыскать компьютерный терминал, чтобы отключить лазер сканирующего устройства системы безопасности или внутреннего телевидения. Плохие парни не блещут умом, поскольку они разместили все эти терминалы как раз там, где вам и надо, но зато у вас имеется шанс произвести разведку других местностей с помощью камер. (Это еще сыграет свою роль на других стадиях игры). Однако чаще всего вы натываетесь на запертую дверь и вынуждены отправиться на поиски ключа, либо взломать ее, перед тем как перейти к следующей секции.

Одна из самых больших проблем в этой игре – искусственный интеллект. При игре в прятки жизненно важно, чтобы другие персонажи в игре вели себя как можно более естественно. Но вы можете спокойно убегать в прямой видимости сидящих в паре метров от вас караульных, а уж это неприемлемо.

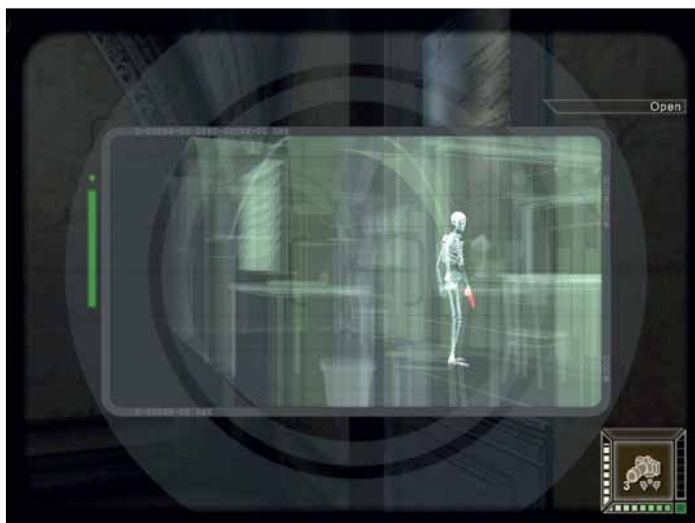
Искусственная глупость

То же самое с определением звуков. Другие игры в жанре «стелс» (вспоминается *Metal Gear Solid*) используют крайне чувствительный порог, который будит AI [искусственный интеллект, – прим. пер.] при малейшем вашем шорохе. В *Cold War* наш герой может расстрелять пару обойм из Калашникова, а караульный за углом и ухом не поведет. А если вас все-таки услышат, то караульные поведут себя довольно ненатурально. Сначала они неистовствуют, но если вы продолжаете прятаться, мало-помалу унимаются, пока в конце концов кто-нибудь не буркнет: «Наверное, крыса». И это не позволяет *Cold War* достичь уровня напряжения, подвластного другим игры данного жанра.

Но хватит о негативных аспектах. Мы оставили самое крутое напоследок. Наша стандартная журналистская камера таинственным образом превратилась в последнее достижение российских высоких технологий – камеру с рентгеновскими X-лучами! Нажав клавишу X, вы можете изменить свой взгляд на мир, превратив его в прозрачную проволочную рамку, через которую вы можете смотреть сквозь стены и видеть вокруг скелеты караульных. Очень грамотный трюк, он добавляет оригинальности всему происходящему. На самом деле мест, где эта камера понадобится, всего несколько, но нет сил устоять перед искушением заглянуть за закрытую дверь или узнать, что там под вами, посмотрев сквозь пол. Увы, постоянно бегать с рен-

тгеновской камерой-намордником нельзя, потому что время пользования ею ограничено. В порядке компенсации рентгеновская камера позволяет дистанционно включать огнетушители, а это частенько бывает актуальным.

Если бы наш журнал был посвящен компьютерным играм, и страница за страницей отводились бы обзорам игр под Linux, мы бы дали *Cold War* прохладный отзыв. Атмосфера – не единственно важное в игре, и можно найти те же игровые функции, куда лучше реализованные в других играх, например, *Splinter Cell*, *Deus Ex* и *Metal Gear Solid*. Но ни одна из этих игр не создавалась изначально под Linux, поэтому мы рассматриваем *Cold War* как единственную в своем роде – и она нас впечатлила. Графика отличная, а игровая атмосфера захватывает, заставляя вас переходить на новые и новые уровни. Да и сюжет не так уж плох. **LXF**



У русских караульных изящный костяк – прямо как у Марии Шараповой.



Улыбнитесь! Вас снимает телевидение.

ВЕРДИКТ LINUX FORMAT

ИГРАБЕЛЬНОСТЬ	7/10
ГРАФИКА	10/10
ДОЛГОВЕЧНОСТЬ	6/10
ЦЕНА	8/10

Для Windows игра посредственная, но это настоящая ахула в маленьком прудике игр под Linux.

РЕЙТИНГ **8/10**



ДИСТРИБУТИВ LINUX

Turbolinux Fuji

Платить \$39 за кодеки Windows Media? **Эндрю Грегори** только что попробовал «это».

САМОЕ ГЛАВНОЕ

Домашний дистрибутив с медиа-уклоном, рассчитанный на пользователей Windows. Аналоги: Linspire или Ubuntu.

- **ВЕРСИЯ:** 11
- **РАЗРАБОТЧИК:** Turbolinux
- **САЙТ:** www.turbolinux.com
- **ЦЕНА:** \$39



Turbolinux почти не встречается в Британии [да и в России тоже, — прим. ред.], но на другой половине глобуса ситуация иная: за последние 12 месяцев компания провела миллионные продажи Linux-дистрибутивов правительственным институтам, инженерным проектам и учебным заведениям. Продажи велики в Японии и огромны в Китае. Доберется ли новейшая настольная версия Turbolinux Fuji до наших берегов?

Mongoose, инсталлятор Turbolinux, стартует в текстовом режиме, переходя затем в графический, и инсталлирует дистрибутив за 35 минут на нашей машине 1.4 ГГц Celeron, 512 МБ ОЗУ. Возможности переработать жесткий диск нет, что для дистрибутива, частично предназначенного для пользователей Windows, выглядит по меньшей мере странно.

После серии громких надоедливых гудков Fuji загружает KDE 3.4.2. Стандартная тема напоминает Windows: кнопка К-меню смахивает на кнопку «Пуск» Win XP, а на рабочем столе есть даже значки «My Documents» и «My Computer». Пользователя Linux это несколько смущает — My Computer нам совершенно ни к чему; но действуют ярлыки точно так же, как в Windows XP.

Расчёт на пользователей Windows не только поверхностный. Сетевой протокол *Samba* (обозначенный Windows Networking для новичков, никогда прежде им не пользовавшихся), доступный по щелчку

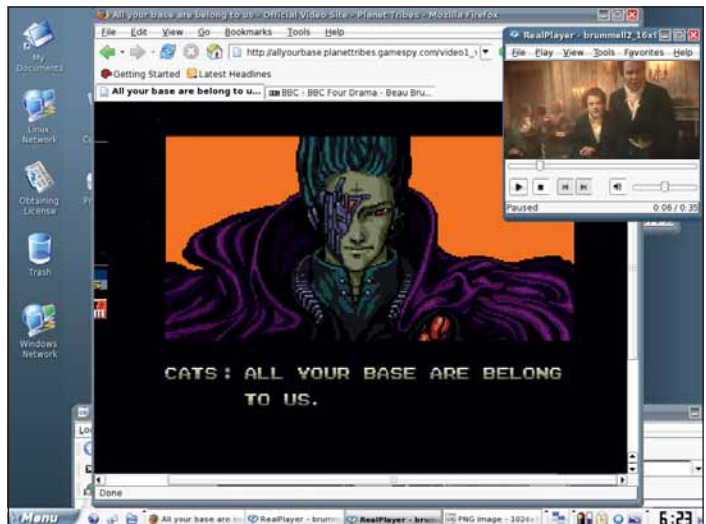
с рабочего стола, заработал в нашей сети сразу же, без всякой настройки. Ещё лучше то, что Fuji по умолчанию монтирует Windows-разделы, а значит, My Computer обеспечивает доступ ко всем (поддерживаемым) файлам на жестком диске. С технической точки зрения в этом нет ничего особенного, зато желающие перейти с системы на систему могут не опасаться потери доступа к старым файлам.

Гибридная ОС

Turbolinux объявляет себя гибридной системой, имея в виду способность к получению несвободного ПО, например, *Macromedia Flash*, *RealPlayer*, Java и Windows Media. Конечно, часть долларов, уплаченных за коробку с Turbolinux, идёт напрямик в Редмонд; но зато для многих домашних пользователей Turbolinux предлагает прямую замену Windows. *Gtkpod* и *Juk* завершают мультимедийную шеренгу.

Единственное, чего здесь не хватает — DVD-плеера. *Power DVD*, который включался в прежние версии, теперь выделен в платный модуль и доступен только через менеджер пакетов *TurboPlus*. Ах нет, не хватает еще одной важной вещи: Fuji поддерживает некоторые закрытые кодеки, но только не тот, что нужен всем — MP3.

Слишком много «но» в этом обзоре. Меню понятны, но дурно организованы и полны опечаток — причем тех же, что мы замечали ещё два года назад, в предыдущей версии (*Gimp* здесь — «Image manipulation tool»). Файлы помощи, диалоговые окна и пункты меню переведены на псевдоанглийский язык, из-за чего весь дистрибутив выглядит неряшливо. Это касается не только стиля, но и сути: Turbolinux стремится стать солидным дистрибутивом с претензией на стабильность, а в итоге вы платите за дистрибутив вчерашнего дня. Fuji имеет 30-дневный срок поддержки, что само по себе неплохо, но если вы не сильны в японском,



Flash, *RealPlayer*, темы и значки в стиле Windows, комический японо-английский: вот вам Turbolinux в двух словах.

вам вряд ли удастся выудить из сайта что-нибудь дельное.

Итак, к чему же мы пришли? Можно пользоваться *Flash*, Java, MP3 и закрытыми драйверами для Linux с помощью некоторых хитрых манипуляций с командной строкой. К несчастью, новичков консольный интерфейс отпугивает, на это и рассчитывает Turbolinux. Но мы скорее порекомендовали бы *EasyUbuntu* (см. стр. 50). Вот он — лучшая реклама Linux: денег не стоит, притом намного проще в использовании.

Turbolinux разочаровал нас своими недостатками, и это грустно. Без сомнения, он и впредь будет иметь успех в Японии и Китае, ведь людям нравятся вещи, созданные на их родном языке. Но вряд ли этот дистрибутив заметят где-либо ещё.

LXF

СОСТАВ TURBOLINUX

- Ядро 2.6.13
- KDE 3.4.2
- Firefox 1.0.6
- Glibc 2.3.5
- GCC 3.4.3
- OpenOffice.org 2.0.2
- Thunderbird 1.0.7
- X.org 6.8.2

ВЕРДИКТ LINUX FORMAT

ФУНКЦИОНАЛЬНОСТЬ	4/10
ПРОИЗВОДИТЕЛЬНОСТЬ	7/10
ПРОСТОТА ИСПОЛЬЗОВАНИЯ	5/10
ЦЕНА	2/10

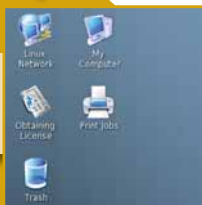
Несколько неплохих идей, но даже после двухлетней разработки дистрибутив выглядит незаконченным.

РЕЙТИНГ **4/10**

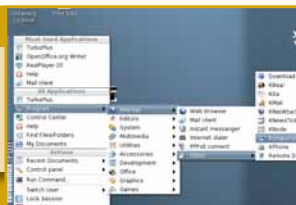


СВОЙСТВА НАВСКИДКУ

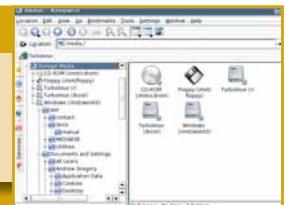
Беглецы с Windows почувствуют себя как дома.



Организация меню Fuji оставляет желать лучшего.



Доступны все ваши прежние файлы (кроме MP3).



АНТИВИРУС

ServerProtect от Trend Micro для Linux 2.5

Грэм Моррисон знакомится с платной программой для очистки Windows с помощью Linux.

САМОЕ ГЛАВНОЕ

Серверная программа защиты файлов в реальном времени. Аналоги: *Sophos AV (LXF78)*, *ClamAV* или *F-Secure AV*.

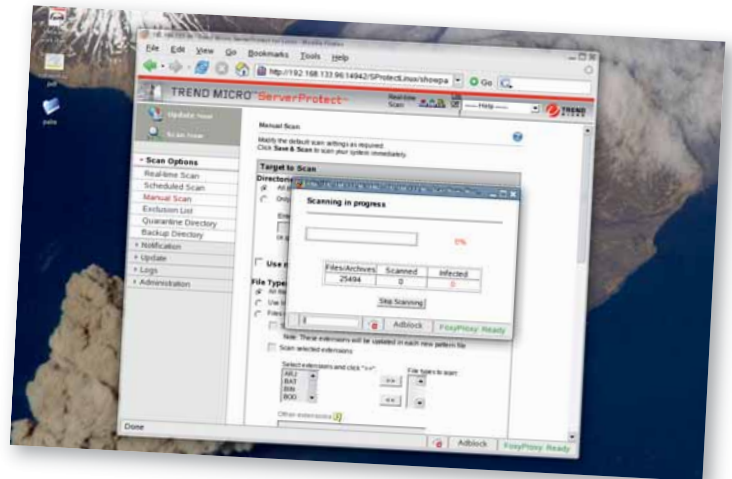
- **РАЗРАБОТЧИК:** Trend Micro
- **САЙТ:** <http://uk.trendmicro-europe.com>
- **ЦЕНА:** От \$1500 на одну клиентскую машину

где маловероятно наличие вирусов. Для Linux здесь нет ничего, за исключением легкого кивка в сторону архивов *tar* и *Gzip*. Единственный Linux-сервер с установленным антивирусом поможет сэкономить на лицензионных отчислениях за настольные программы для Windows от Norton или McAfee.

Поддерживаемые платформы

В отличие от свободных антивирусов, *ServerProtect* весьма разборчив в дистрибутивах, с которыми может работать. Выбор ограничен: Red Hat Enterprise Linux 4, SUSE Linux Enterprise Server 9, Linux Desktop 9 от Novell или Debian 3.1. Кроме того, у вас должна быть особая версия ядра 2.6 для одно- или многопроцессорной машины (хотя по умолчанию Debian 3.1 использует ядро 2.4). Как и большинство антивирусных программ, эта полагается на специальный бинарный модуль ядра – в данном случае *splxmod*.

Если ваша система удовлетворяет приведённым требованиям, то инсталляция проходит сравнительно гладко. Вы просто запускаете скрипт, который задаёт вам обычные вопросы о лицензии. Как и в других антивирусах от Trend Micro (см. *InterScanVirusWall* в нашем обзоре антивирусов LXF75), основной интерфейс доступен из браузера – как через старый добрый HTTP, так и с помощью шифрованного HTTPS. Конечно, это имеет смысл для сервера, так как на нём ПО настраивается удалённо. Вряд ли вы на своём сервере установите KDE, но если это всё-таки случится, то к вашим услугам некоторые дополнительные пункты меню. В *ServerProtect* это называется Quick Access Console Menus (Консольные меню быстрого доступа), в действительности же это не что иное, как



Можно просканировать систему и вручную, но лучше дать *ServerProtect* свободу действий. Найдя что-нибудь, программа известит вас почтой!

ссылки на несколько консольных программ, выполняющих всю работу; их легко перенести и в Gnome. Консольные программы действительно очень полезны и позволяют вручную сканировать файлы и директории, а также запускать и останавливать сервер.

Как это работает?

Web-интерфейс очень прост, изучать в нём особенно нечего. Здесь всего три страницы, с которыми доведется работать с некоторой регулярностью. Первая из них позволяет вручную сканировать файловую систему, выбирать отдельные директории или всю систему полностью. На второй странице устанавливается регулярность обновления антивирусной базы данных – можно установить значение от нескольких часов до одного раза в день. База эта весьма богата, она общая для многих продуктов Trend Micro. На третьей странице можно просмотреть протоколы работы антивируса. В отличие от предыдущих версий, эти сообщения теперь могут быть отправлены на ваш почтовый ящик.

Одно из преимуществ *ServerProtect* перед другими антивирусами – особая способность сканирования в реальном времени. Файлы подвергаются проверке во время открытия, загрузки, выгрузки и даже во

время исполнения. Как только файл появляется в файловой системе, *ServerProtect* тут же находит его. Вирусы перехватываются даже в частично загруженных временных файлах и файлах, перемещаемых через шифрованные SFTP-соединения, то есть ещё до записи на диск – это действительно впечатляет и выделяет программу из общего ряда.

Без сомнения, *ServerProtect* сможет проверить любой файл, прошедший через ваш Linux-сервер, и сделает это быстро и эффективно. **LXF**

ВЕРДИКТ LINUX FORMAT

ФУНКЦИОНАЛЬНОСТЬ	7/10
ПРОИЗВОДИТЕЛЬНОСТЬ	8/10
ПРОСТОТА ИСПОЛЬЗОВАНИЯ	8/10
ЦЕНА	7/10

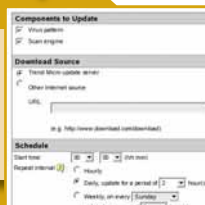
Превосходное качество проверки, но стоит поработать над интеграцией с почтой и прокси.

РЕЙТИНГ 7/10

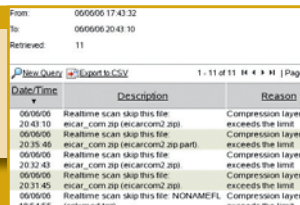


СВОЙСТВА НАВСКИДКУ

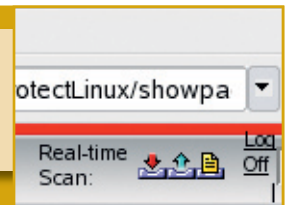
Проверка на вирусы и обновление вирусной базы по расписанию.



Инфицированные файлы удаляются или изолируются.



Сканирование в реальном времени можно применить и к работающим программам.



МЕНЕДЖЕР ФОТОГРАФИЙ

Picasa 2

Пойте аллилуйя и воздевайте руки к небесам: Google сдержал обещание выпустить программу для работы с фото под Linux. Суперновость или холостой выстрел? **Энди Ченнел** хватает камеру...

САМОЕ ГЛАВНОЕ

Бесплатное приложение для управления фотографиями, приобретенное Google в 2004 году. Аналоги: *F-Spot* и *DigiKam*.

- **РАЗРАБОТЧИК:** Google и Codeweavers
- **САЙТ:** <http://picasa.google.com/linux>
- **ЦЕНА:** Бесплатно по закрытой лицензии

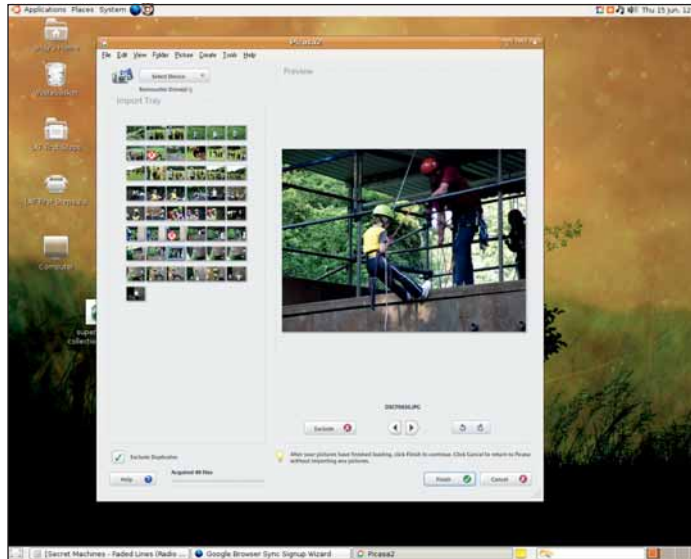


Linux благословлен двумя превосходными программами для работы с фото — *F-Spot* и *DigiKam*; так что же может выложить на стол Google, чего у нас пока не было? Фирма столкнулась с напором критики по поводу Linux-версии своего адаптированного ПО для фотографий: вместо того, чтобы стать целиком родным портом приложения, *Picasa* под Linux полагается на эмулятор Windows, *Wine*. А некоторые просто сочили, что приложение написано с использованием *Wine* API (в противовес *Qt* или *GTK*) и таким образом исходно может рассматриваться — в некотором роде — как приложение Linux. Но какой бы ни была политика, главное, что *Picasa* теперь доступна, а 200 с лишним патчей Google к проекту *Wine* дадут возможность другим разработчикам портировать свои Windows-продукты в Linux.

Вай-вай-вай, Wine

Наследие *Wine* сказало в сером цвете элементов пользовательского интерфейса *Picasa* — панелей инструментов, меню и диалоговых окон, но основное приложение не следует стандартам дизайна Windows, поэтому оно вовсе не казалось неуместным в тестовых рабочих столах KDE 3.5 и Gnome 2.14. Проблемы прорисовки окон иногда становятся настоящей напастью в приложениях *Wine*, но здесь они в основном взяты под контроль, за исключением тех случаев, когда надо изменить размер окна.

При первой загрузке *Picasa* предложит или прочесть всю вашу систему в поисках



Picasa хорошо интегрируется с Ubuntu, несмотря на уродливые эффекты Wine и избыток серого цвета. Да и импорт изображений отличный!

изображений, или ограничиться рабочим столом. На просмотр нашей библиотеки, состоящей из 1500 изображений, ушло примерно 35 секунд, но во время этого процесса с приложением можно работать. Google приложил немалые усилия, чтобы *Picasa* нормально работала с *Gphoto*, а это означает, что мы без проблем подключили нашу камеру через USB. Изображения переносятся в буфер, где их можно посмотреть и повернуть, а потом уж принять решение: стоит ли импортировать изображение. Функции удаления изображений из камеры после их импорта нет (это неприятно, но не смертельно), зато есть функции добавления имени папки, описания и даты.

Как только вы импортировали изображения, появляется куча инструментов для работы с ними — цветокоррекция, кадрирование, удаление эффекта «красных глаз» и тому подобное; несколько симпатичных функций настройки; несколько базовых эффектов, например, зернистость пленки. Есть и хитроумные инструменты для презентации результатов, включая пересылку

изображений по электронной почте (очень полезно для пользователей Flickr, потому что подлинной интеграции с этим сервисом нет), инструмент коллажа, отправку изображений на печать и слайд-шоу. Сервис электронной почты может использовать любую почтовую программу или соединиться с Gmail, причем настраивает размер изображения для снижения трафика. Кнопка Order Prints использует отношения Google с разнообразными сервисами печати и предлагает доступ к восьми сервисным центрам печати в Великобритании (но не в России, — прим. ред.).

Googlinux

Производительность скорее приемлемая, чем потрясающая. Попытка быстрой навигации при просмотре 1000 цифровых фото вызвала заметное торможение, хотя дисплей и старался не отставать, даже на быстрой машине (Athlon 2500+ с 512 МБ ОЗУ и видеокартой 128 МБ). Однако подобное мы замечали в большинстве приложений для работы с фотографиями, и недостаток

ГРЭМ СЧИТАЕТ...



«Picasa неплоха для нетребовательных пользователей, но я предпочитаю управлять моими фото через Beagle, а загружать их через DigiKam. Wine тут ни к чему.»

этот постепенно исчезал по мере взросления программ.

В качестве заявления о намерениях Google (вам решать, филантропических или злобных) *Picasa* довольно хороша. Тот факт, что столь известная фирма выпустила ПО под Linux, без сомнения, поднимет репутацию ОС как жизнеспособной альтернативы для домашнего пользования.

Picasa немного медленнее, чем *F-Spot*, когда доходит до навигации по большой коллекции фото, и в ней нет никаких инструментов Flickr, но она более чем компенсирует этот недостаток другими функциями, особенно в сфере профессиональной печати. Документация тоже первоклассная, и все это вместе хорошо интегрируется с KDE 3.5, Gnome 2.14 и *Firefox*.

Если вы аллергик на проприетарное ПО, крапивница вам гарантирована. Если же отсутствие исходного кода вас не беспокоит, то попробуйте *Picasa* действительно стоит. **LXF**

ВЕРДИКТ LINUX FORMAT

ФУНКЦИОНАЛЬНОСТЬ	7/10
ПРОИЗВОДИТЕЛЬНОСТЬ	6/10
ПРОСТОТА ИСПОЛЬЗОВАНИЯ	8/10
ДОКУМЕНТАЦИЯ	8/10

Очень хорошее приложение для домашнего пользования, с весьма умными функциями.

РЕЙТИНГ **7/10**



СВОЙСТВА НАВСКИДКУ

Возьмите кучу картинок, и коллаж из них будет фоном рабочего стола.



Возьмите кучу картинок и пошлите их по электронной почте как вложения.



Возьмите кучу картинок, пошлите их чужеземцам, и... вам придет распечатка по почте.



ИНТЕРНЕТ-ПЕЙДЖЕР

Kopete 0.12.0



Пусть в Чили это слово значит «выпивон с друзьями»; для Грэма Моррисона *Kopete* – солидный IM-клиент.

САМОЕ ГЛАВНОЕ

Клиент мгновенных сообщений, способный работать с несколькими разными протоколами. Аналоги: *Gaim* или *Skype*.

- **РАЗРАБОТЧИК:** Duncan Mac-Vicar Prett и команда разработчиков *Kopete*
- **САЙТ:** <http://kopete.kde.org>
- **ЦЕНА:** Бесплатно по лицензии GPL



Мы часто критикуем разработчиков KDE за увлечение сверхфункциональностью, но в некоторых случаях им удаётся соблюсти чувство меры. *Kopete* – один из таких случаев. Она может соединить вас одновременно с десятком разными службами мгновенных сообщений, причем делает это, почти не озадачив пользователя. Здесь есть внушительный пакет модулей, настраиваемый пользовательский интерфейс, разнообразные оконные темы и на удивление плавный скроллинг.

Центр вселенной *Kopete* – окно со списком «друзей» (buddy list). Именно здесь вы найдёте свой перечень контактов. Можно создать иерархическую структуру каталогов для друзей, родни и коллег, добавляя контакты как с помощью небольшого Мастера, так и через «приглашения», которые ваши респонденты посылают через собственные программы мгновенных сообщений.

Конечно, все эти способности целиком зависят от используемых служб сообщений. *Kopete* поддерживает их во множестве (см. врезку «Переговорные сети» ниже), от IRC и Jabber до трицы наимоднейших на сегодняшний день протоколов, Microsoft's Messenger, Yahoo's Messenger и AOL's instant messenger (AIM). К несча-

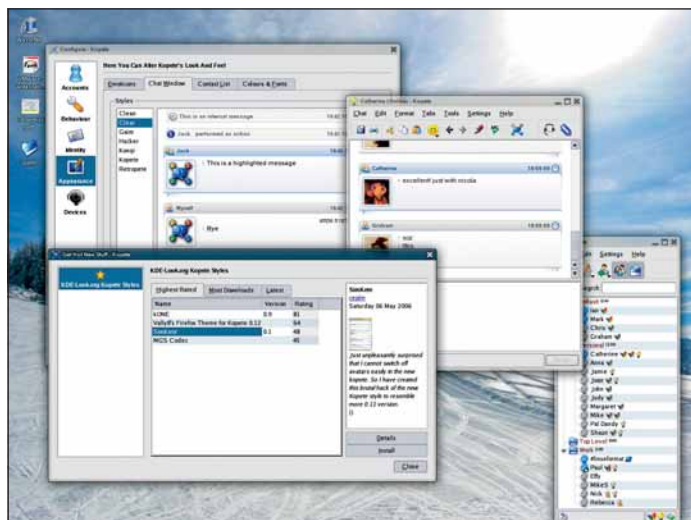
стью, все три – частные, что чрезвычайно усложняет отслеживание внутренних изменений и работу со скрытыми функциями для сторонних клиентов вроде *Kopete*. В результате некоторые подключения проваливаются, а больше всего страдает обмен файлами – особенно между *Kopete* и *MS Messenger*. Получить файлы от клиентов *MS Messenger* мы ещё смогли, а вот отправить – уже нет.

С Jabber, открытым протоколом, таких проблем не возникает. *Kopete* бесперебойно работает с Jabber messaging от Google через Google Talk, поэтому просто здорово, если у ваших друзей есть Gmail.

Мета-контакты

Службы сообщений, которыми пользуются ваши респонденты, отображены маленькими значками рядом с их именами. Но действительно по-умному *Kopete* управляется с этим скоплением контактов и протоколов при помощи так называемых мета-контактов (они позволяют одному контакту иметь несколько разных протоколов). Например, если у Пола есть учётные записи в Jabber, AIM и Yahoo, для каждого из этих протоколов можно создать отдельную запись в списке. С помощью мета-контакта подтащите все протоколы к его имени и наблюдайте за их состоянием по значку протокола – существенная экономия места, и пользоваться удобно. Чтобы связаться с Полом через Jabber, просто щёлкните значок Jabber напротив его имени.

Разработчики *Kopete* не сидят сложа руки: последние несколько месяцев они потратили на отладку нескольких новых



Kopete может быть маленьким значком, а может занять весь рабочий стол.

функций для нынешней версии 0.12. Самая заметная из них – Hot New Stuff (Горячие Новинки). Это (ужасный) KDE-термин для загружаемого контента, включенного в приложения, а в *Kopete* он означает новейшие значки, темы и оконные стили чатов с www.kde-look.org. Загружаются Новинки с помощью двух дополнительных кнопок в панелях настройки окон «Emoticons» и «Chat». В отличие от *SuperKaramba*, где соединения с Hot New Stuff иногда срываются, в *Kopete* всё работает надёжно – и это прекрасная возможность освежить вид вашей программы.

Ещё одно заметное новшество – экспериментальное включение Jingle, голосового расширения Jabber для работы с Google Talk. Нам удалось добиться некоторых успе-

хов при общении *Kopete-Kopete*, но вот попытки поговорить через *Kopete* с Google Talk неизменно проваливались.

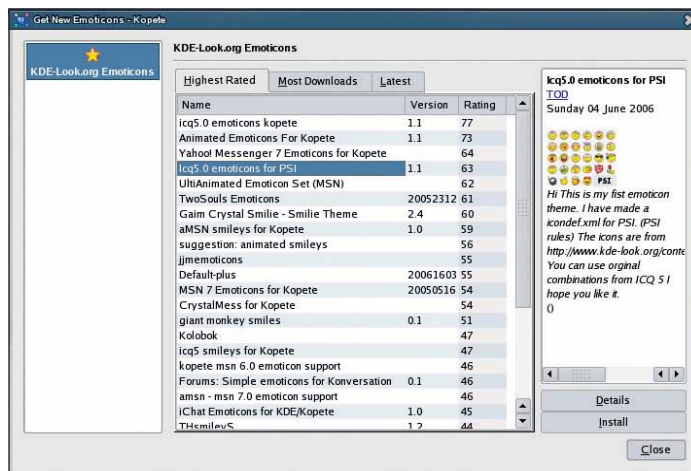
На грустной ноте заканчивать не хотелось бы, ведь *Kopete*, наряду с *Gaim*, один из лучших клиентов мгновенных сообщений для любой системы. Если уж на то пошло, можно сказать, что нынешняя версия *Kopete* превзошла Gnome-соперника. И вообще, есть целый ряд причин, почему *Kopete* – это хорошо: проверка правописания в реальном времени, оконные стили в XML, система чатов-вкладок в единственном окне, модули перевода на разные языки, текстовые эффекты, IRC и мгновенные сообщения из единственного приложения, шифрование переговоров, плавные графические переходы, отправка сообщений на мобильные телефоны.

Ну, как тут не попробовать *Kopete*? **LXF**

ПЕРЕГОВОРНЫЕ СЕТИ

Kopete работает со следующими протоколами:

- AIM
- Gadu-Gadu
- Novell's GroupWise
- ICQ
- IRC
- Jabber
- MSN Messenger
- SMS
- WinPopup
- Yahoo



Hot New Stuff позволяет загружать значки и темы для *Kopete*.

ВЕРДИКТ LINUX FORMAT

ФУНКЦИОНАЛЬНОСТЬ	9/10
ПРОИЗВОДИТЕЛЬНОСТЬ	9/10
ПРОСТОТА ИСПОЛЬЗОВАНИЯ	9/10
ДОКУМЕНТАЦИЯ	7/10

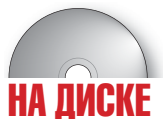
До твёрдых 10 баллов не хватает только функции Buddy Pounce, как у Gaim.

РЕЙТИНГ 9/10



ТЕКСТОВЫЙ РЕДАКТОР

Vim 7.0



Самый артистичный из всех редакторов: в нем больше горячих клавиш, чем может упомянуть Пол Хадсон.

САМОЕ ГЛАВНОЕ

Консольный текстовый редактор с тысячами скрытых возможностей. Аналоги: *Emacs*, *Nano* или *Joe*.

- **РАЗРАБОТЧИК:** Команда Vim
- **САЙТ:** www.vim.org
- **ЦЕНА:** Бесплатно под GPL-совместимой лицензией



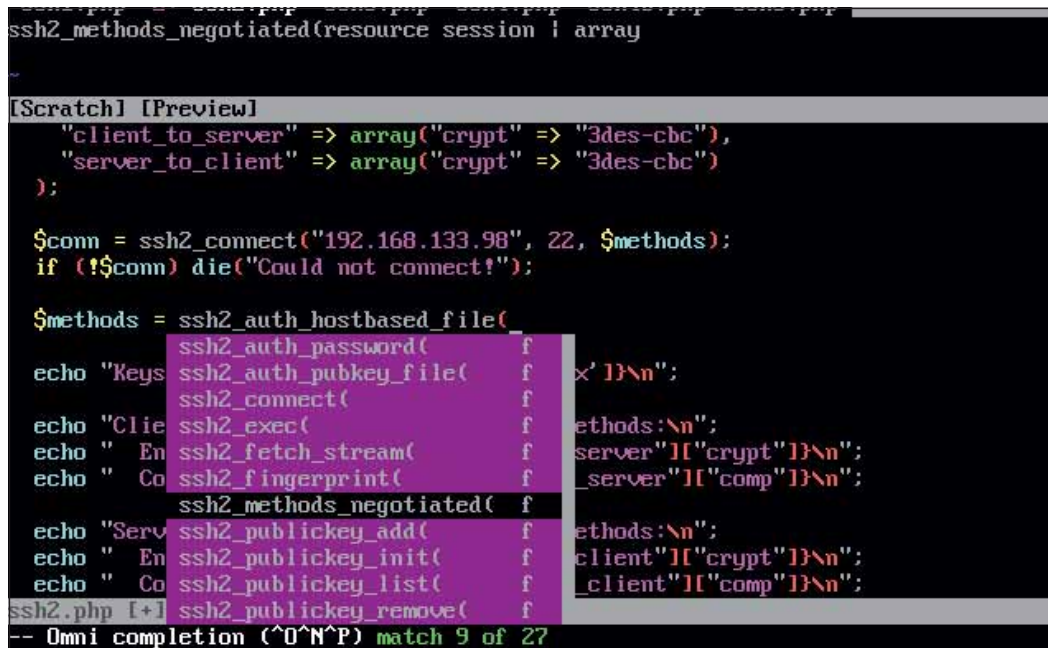
Пусть он еще не научился варить кофе, зато все остальное новая версия маститого редактора *Vim* уже умеет. Многие считают *Vim* такой же неотъемлемой частью Linux, как и *Bash*. Стандартная черно-серая схема сохраняется со времен его предшественника *Vi*, и *Vim* по-прежнему стартует в режиме совместимости; поэтому, запустив 7.0, вы вправе усомниться в наличии каких-либо изменений.

Не сомневайтесь, их предостаточно. Если вы редактируете в *Vim* случайный файл конфигурации, то, скорее всего, не заметите никакой разницы. Если же вы продвинутый пользователь *Vim*, то приглотаетесь к вкладочному редактированию, автозавершению кода, проверке правописания «на лету», новой системе грег, дереву откатов и встроенному профилированию скриптов для выявления медленных участков кода. И это далеко не все удивительные новшества.

Умные дополнения

Некоторые новые для *Vim 7* функции известны уже давно, но здесь они реализованы настолько гладко, что почти не требуют изучения. Взять, к примеру, новую систему вкладочного редактирования. *Vim* всегда умел (и умеет) редактировать несколько файлов сразу, но это выглядело убого и было сложно в освоении. Теперь принят набор вкладок в стиле *Opera*, которые появляются при одновременном открытии нескольких документов. Каждая вкладка несет название открытого в ней файла, а к названию несохраненных добавляется знак **+**. Просто, но такого-то мы и ждем от редактора подобного класса.

Задачи нововведений в области автозавершения кода и проверки правописания кажутся полностью противоположными: того, кому нужно автозавершение, вряд ли заботит проверка правописания, и наоборот. Но в этом весь *Vim*: у него есть что предложить любому пользователю, а своих услуг он не навязывает. Если вы вводите



Подсветка синтаксиса плюс автозавершение кода и проверка правописания делают *Vim 7.0* почти что экстрасенсом...

текст через *Vim*, то проверка правописания весьма тщательна, это не просто сравнение вашего текста со списком слов: например, подсвечиваются неверно введенные слова, предложения, начинающиеся не с заглавной буквы, слова, чуждые вашей стране (сравните 'center' и 'centre'), и редкие слова (т.е. те, которые вы, вероятно, ввели случайно). Если слово введено неверно, *Vim* предложит список слов для замены, составленный из замен, использованных прежде. Чем чаще применялась замена, тем выше она размещается в списке.

Для редактирования

Автозавершение кода, для первой попытки, действует неплохо. Неудобство состоит лишь в активации по клавишам **Ctrl+X** и **Ctrl+O**, в противоположность более привычным открытым скобкам. Зато выдается список параметров функций, и в нем присутствуют пользовательские переменные, что дает *Vim* фору перед прочими средами разработки, которые нам встречались.

ОК, вкладки, автозавершение и проверка правописания — безусловные хиты, но этого нельзя сказать о *Vimgrep* и о дереве откатов (по крайней мере, пока). Система *Vimgrep* задумана неплохо, но на практике трудноосуществима. Идея состоит в том, что вам достаточно набрать **:Vimgrep некоторый_образец*.все_что_угод-**

но, а *Vim* разыщет все файлы с указанным образцом и откроет их для редактирования в своих вкладках. Как будто разумно. Можно даже приказать *Vim* открыть лишь первый из найденных файлов, набрав **:Vimgrep**. Однако по неизвестным причинам записать найденные файлы в новую вкладочную систему невозможно: файлы то загружаются, но вам по-прежнему понадобится клавиатурная черная магия для навигации по ним.

Дерево откатов в теории кажется весьма тонким замыслом. Вообразите ситуацию: вы набрали большой отрывок текста и кое-что в нем переделали. Поразмыслив, вы решили, что первоначальный вариант был лучше, и нажали **Undo** (Откат), чтобы отменить изменения. Поработали еще — и вспомнили, что среди тех самых изменений был жутко удачный параграф, но, увы, он был затерт с началом набора текста после отката, и ваш шанс отменить откат нажатием **Redo** (Отмена отката) упущен. Согласен, этот сценарий надуман: ветвление откатов отнюдь не самая актуальная задача. Круто, конечно, набрать 'earlier 20m' и получить свой документ таким, каким он был 20 минут назад, или прыгать по откатам и отменам до бесконечности, но особого смысла тут не просматривается.

Мы не имеем ничего против всех этих новшеств. Если они вам не нужны, то вы

даже не заметите их существования. А если вы продвинутый пользователь *Vim*, то возьмите и скомпилируйте исходные тексты: новые функции слишком хороши, стоит ли полгода ждать, пока их включат в ваш дистрибутив? **LXF**

ВЕРДИКТ LINUX FORMAT

ФУНКЦИОНАЛЬНОСТЬ	8/10
ПРОИЗВОДИТЕЛЬНОСТЬ	9/10
ПРОСТОТА ИСПОЛЬЗОВАНИЯ	8/10
ДОКУМЕНТАЦИЯ	9/10

Грамотное исполнение сделало это обновление приятным. А уж без вкладочного редактирования и вовсе не обойтись.

РЕЙТИНГ **8/10**



ТЕПЕРЬ В СУТКАХ 25 ЧАСОВ!

Не хватает времени в течение суток? IBM представляет сервер IBM System p5 510Q Express. Его новая технология Quad-Core Module значительно повышает производительность системы. А это означает увеличение объема операций в секунду. Это инновации, имеющие значение. И они позволяют Вам использовать время с максимальной отдачей.



Товар сертифицирован

IBM System p5 510Q Express

Особенно подходит для приложений, критически важных для бизнеса, таких, как SAP® и WebSphere®, работающих в AIX 5L™ или Linux®.

От **\$12 052***

4 процессора POWER5+ с частотой 1,65 ГГц

8 ГБ памяти

2 диска по 73 ГБ

Сервисный процессор

Резервирование по охлаждению

Двухпортовый контроллер Ethernet 10/100/1000 Мбит/с

ОС AIX с поддержкой на 1 год

Стандартная гарантия: 3 года с обслуживанием на месте¹

Реклама

IBM System Storage



IBM System Storage DS4700 Express

Высокопроизводительная технология подключения Fibre Channel 4 Гбит/с

До 33,6 ТБ физической емкости хранения

Мощные средства управления системой и данными

Возможность наращивания емкости от уровня рабочей группы до уровня предприятия за счет подключения шести дисковых модулей расширения DS4000 EXP810

От **\$21 500***

IBM System Storage



Ленточный автозагрузчик IBM System Storage TS3100 Express

Поддержка ленточных накопителей LTO Ultrium 3

Подключение по интерфейсам LVD SCSI или Fibre Channel 4 Гбит/с

Возможность установки до 22 картриджей общей емкостью 7,8 ТБ без компрессии

От **\$10 500***

**УЗНАЙТЕ БОЛЬШЕ
О СЕРИИ ПРОДУКТОВ
IBM EXPRESS**

[ibm.com/
eserver/ru/express3](http://ibm.com/eserver/ru/express3)

8 495 258 6348,

8 800 2006 900

(звонок по России бесплатный)

* Все указанные цены установлены IBM для розничных продаж и были действительны на момент публикации данного материала. Стоимость IBM System p5 510Q Express в рублях составляет 324 922 рубля по курсу доллара 26,96 рублей. Стоимость IBM System Storage DS4700 Express в рублях составляет 579 640 рублей по курсу доллара 26,96 рублей. Стоимость IBM TotalStorage 3100 Express в рублях составляет 283 080 рублей по курсу доллара 26,96 рублей. Цена в рублях может изменяться при изменении курса доллара. Цены могут отличаться в зависимости от конфигурации. Посредники устанавливают собственные цены; цены для конечного потребителя могут варьироваться. Предложения действительны при условии наличия товаров. Количество товаров на складе ограничено. IBM оставляет за собой право изменять цены без уведомления. В базовую стоимость не всегда входит жесткий диск, операционная система или другие компоненты. Информацию о текущих ценах в Вашем регионе Вы можете узнать в представительстве IBM или у партнеров IBM. Аппаратные продукты IBM производятся из новых деталей или из новых и исправных деталей, бывших в употреблении. Это не влияет на гарантийные условия. Телефонная поддержка может оплачиваться отдельно. При возникновении локальных проблем, прежде чем направлять специалиста к месту возникновения неполадок, IBM проведет диагностику с последующей попыткой решения проблемы удаленно. Гарантия распространяется только на некоторые компоненты. IBM, логотип IBM, System p5, WebSphere, AIX 5L, POWER5+ и OpenPower являются зарегистрированными товарными знаками International Business Machines Corporation в США и/или других странах. Linux является товарным знаком Линуса Торвалдса в США и других странах. Наименования других компаний, продуктов и услуг могут быть товарными знаками или знаками обслуживания третьих лиц. © 2006 IBM Corporation. Все права защищены.



Distrowatch

Ежемесячный обзор новых разработок дистрибутивов Linux.

БРАВО, FEDORA



Ладислав Боднар
основатель,
начальник и
сотрудник сайта
DistroWatch.com.

После появления релизов трех основных дистрибутивов многие пользователи Linux задают один и тот же вопрос: какой дистрибутив лучше? На этот вопрос не просто ответить, однако множество обзоров, блогов и форумов дают достаточно информации о состоянии дел в том или ином проекте.

Пожалуй, самым большим разочарованием стал Ubuntu 6.06.

«SUSE LINUX 10.1 ВСЕ ЕЩЕ СТРАДАЕТ ОТ СТАРЫХ ОШИБОК.»

Отложенный на шесть недель под флагом «долгосрочной поддержки» и «дополнительной отладки», он подавал большие надежды. Однако в итоге пользователи столкнулись с проблемами в новом инсталляторе, временем загрузки 20 минут и серьезными неполадками печати (лидер проекта Марк Шаттлворт отвечает на эти претензии на стр. 36).

SUSE Linux 10.1 также был отложен, однако финальный релиз по-прежнему содержит серьезные ошибки, возникающие в новой системе управления программами. Кроме того, хотя трехмерные эффекты Xgl и Compiz и впечатляют, эти технологии еще недостаточно созрели для серьезного промышленного применения.

С оглядкой на три основных релиза, приз за наиболее стабильный и свободный от ошибок дистрибутив Linux во втором квартале этого года мы отдаем Fedora Core 5. Хотя тут отсутствуют какие-то серьезные новые возможности, команда разработчиков Fedora 5 заслуживает благодарности за создание прекрасного спроектированного продукта, доставляющего удовольствие от его использования и не содержащего неприятных ошибок.

Если у вас есть что сказать о трех основных дистрибутивах Linux, пришлите мне свое мнение по почте:

ladislav.bodnar@futurenet.co.uk

Риппу-люб

Puppy Linux 2.0 – быстрый Live-дистрибутив.



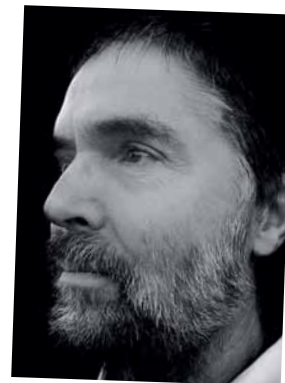
Puppy Linux – дистрибутив для знатоков. В отличие от большинства других проектов,

ориентированных на устаревшие компьютеры и представляющих собой хотя и небольшие, но полнофункциональные операционные системы, Puppy Linux создан с нуля: каждый его компонент отобран вручную, а каждый кандидат в приложения придирчиво проверялся по критерию отношения размер/функциональность. В результате Риппу превратился в дистрибутив с преданным и быстро растущим сообществом пользователей.

Чтобы понять Риппу, нужно понять его создателя – Барри Каулера [Barry Kauler]. Проживающий в сельской части западной Австралии 56-летний основатель проек-

та, бывший лектор университета, является поклонником простого и ясного дизайна и противником раздутого от возможностей ПО, характерного для наших дней. В свое время он создал полнофункциональный векторный редактор для Windows размером менее 80 КБ. «Видимо, я был помешан на простоте и компактности и считал все слишком медленным и громоздким», вспоминает Барри.

Философия Каулера становится ясна сразу после загрузки дистрибутива. Поскольку 70-МБ ISO-образ работает как LiveCD, дистрибутиву не требуется установка на жесткий диск, чтобы продемонстрировать его скорость и мощь – этот CD мог бы использоваться на компьютере вообще без устройства хранения данных. Более того,



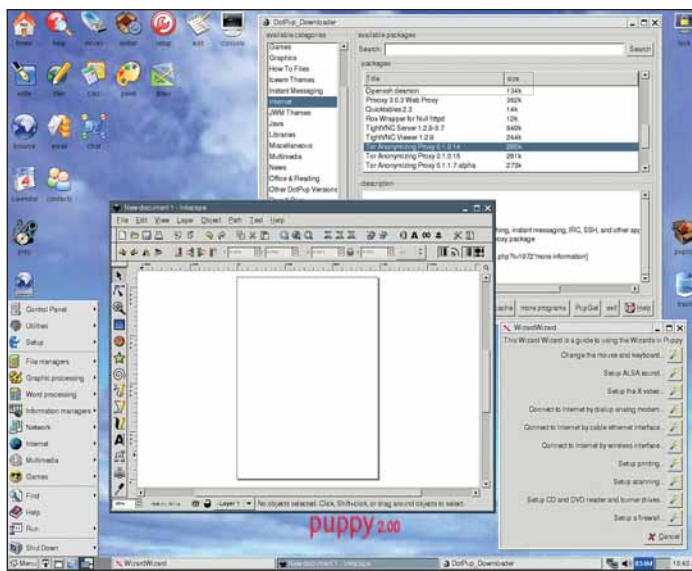
Барри Каулер, автор Puppy Linux.

сразу после загрузки дистрибутив помещает себя в память, и привод CD-дисков освобождается для других целей. В итоге Риппу чрезвычайно быстр, что очень важно для устаревших компьютеров.

Один из лучших аспектов Риппу – для каждой задачи по настройке какой-либо части системы предлагается специальная утилита, включая настройку беспроводной сети и брандмауэра. Как вы и ожидаете, поддерживается установка дистрибутива на жесткий диск, а в сам Live CD включены дополнительные программы. Разработчики также создали так называемый мультисессионный CD/DVD – единственный Live-дистрибутив, поддерживающий запись данных на перезаписываемые CD- или DVD-диски.

Puppy Linux сделан с любовью – это красивая и мощная операционная система, вручную доведенная до совершенства. www.puppylinux.com

«PUPPY LINUX СОЗДАН С НУЛЯ, ОТБОР ПРОШЛИ ВСЕ ЕГО КОМПОНЕНТЫ.»



Легкий, быстрый, полнофункциональный, расширяемый – вот Puppy Linux 2.0.

В РЕЛИЗЕ 2.0

- Оконный менеджер: *Rox-Filer*
- Текстовый процессор: *AbiWord*
- Электронная таблица: *Gnumeric*
- Графика: *Inkscape*
- Медиа-проигрыватель: *Gxine*
- Интернет: *SeaMonkey*

Эти программы предоставляются по умолчанию, однако с помощью утилит управления пакетами *PupGet* и *DotPup* легковесного Щеночка-Риппу можно запросто превратить в матерого ротвейлера.

Свободный как...?

Freespire 1.0 — «очередной» дистрибутив на базе Debian.



Freespire — новый проект, созданный Linspire — анонсировал предварительный

план, по которому версия дистрибутива 1.0 ожидается в сентябре, а 1.1 будет выпущена до конца года. Первый релиз ответвится от стабильной версии Linspire 5.0; предполагается, что он пройдет только краткую фазу бета-тестирования, длительностью всего месяц. Идея заключается в наивозможно скорейшем выпуске первого релиза, чтобы привлечь новых пользователей и разработчиков.

«Настоящая» разработка Freespire начнется с версии 1.1, в которую войдет несколько новых возможностей, вклю-

рии появления нового релиза — итак, можно ожидать, что во Freespire 1.1 войдет ядро 2.6.17, KDE 3.5 и GCC 4.1.

По отработанной другими дистрибутивами схеме Freespire 1.1 должен образовывать основу для новой версии Linspire 6.0 — коммерческого дистрибутива, дата релиза которого также запланирована на конец 2006-го года.

Довольно спорным было анонсирование Freespire как первого дистрибутива Linux, в который за дополнительную плату будут включены легально лицензированные медиа-кодеки, драйвера устройств и приложения с закрытым кодом. Кое-кто из апологетов свободного ПО заявил, что



Создатели Freespire верят в идею предоставления пользователям выбора закрытых дополнений.

В любом случае Freespire обещает быть интересным проектом. Если все пойдет по плану, то репозитории Freespire откроются для публичного доступа в конце июля (возможно, как раз к моменту чтения этих строк), а первая бета-версия станет доступна для скачивания в начале августа. Несмотря на критику со стороны некоторых записных линуксоидов, разработчики Linspire уже продемонстрировали свои возможности по созданию удобной и продуманно-дружественной операционной системы, ориентированной на выполнение повседневных задач пользователя.

www.freespire.org **LXF**

ХИТ-ПАРАД ДИСТРИБУТИВОВ

10 самых посещаемых страниц на DistroWatch.com в июне (среднее число визитов в день)

1	Ubuntu Linux	3,102	↔
2	SUSE Linux	1,632	↑
3	Fedora Core	1,248	↓
4	Mepis Linux	1,057	↑
5	Damn Small Linux	822	↔
6	Mandriva	809	↓
7	Knoppix	759	↓
8	Debian GNU/Linux	716	↓
9	Kubuntu	622	↑
10	PCLinuxOS	576	↑

DistroWatch.com отслеживает популярность дистрибутивов, основываясь на количестве визитов на страницы сайта, посвященные конкретным дистрибутивам. Хотя эти цифры и не отражают настоящее количество инсталляций, они являются индикатором популярности дистрибутива на данный момент времени.

«В ЛЮБОМ СЛУЧАЕ FREESPIRE ОБЕЩАЕТ БЫТЬ ИНТЕРЕСНЫМ.»

чая новый Click-N-Run клиент для легкой установки пакетов, совместимость с Linux Standards Base 3.1, новый модуль KDE для выбора приложений при автостарте, утилиты переключения пользователей и утилиту поиска для рабочего стола. Набор пакетов будет синхронизирован с тестовой веткой Debian, которая к тому времени должна достичь стабильного состояния, в преддвее-

такой шаг противоречит духу Free Software, особенно учитывая имя 'Freespire'. На что создатели Freespire ответили, что установка любого платного компонента будет только опциональной, а легкость доступа к этим программам может склонить менее технически грамотных пользователей к миграции на Linux, увеличивая, таким образом, пользовательскую базу.

ЦЕЛАЯ ГОРСТЬ BSD

Разнообразие Live-дистрибутивов BSD

Сразу после первого публичного релиза Knoppix LiveCD в 2003-м году компьютерный мир был буквально наводнен загрузочными Linux LiveCD. А как насчет тех, кто предпочитает BSD? Существуют ли Live CD на базе FreeBSD, OpenBSD или NetBSD? Конечно! Давайте посмотрим, что у нас есть в наличии.

• На базе FreeBSD

FreeSBIE и Frenzy — два великодушных Live-дистрибутива на основе FreeBSD. **FreeSBIE** разрабатывается итальянской группой пользователей BSD и представляет собой настольную систему общего назначения с Fluxbox или Xfce в качестве оконного менеджера. Текущая стабильная версия — 1.2, а версия 2.0 для i386 и AMD64 находится в процессе интенсивной разработки, и, возможно, к момен-



ту чтения этих строк релиз уже вышел. Frenzy — украинский проект, созданный Сергеем Можайским — более специализированный дистрибутив, содержащий коллекцию сетевых утилит и утилит безопасности для опытных администраторов и задач восстановления системы.

www.freesbie.org <http://frenzy.org.ua>

• На базе OpenBSD

У тех, кто предпочитает безопасную OpenBSD, есть два варианта. Первый — это **Anonym.OS**, LiveCD с утилитами шифрования данных и анонимного подключения к Интернету. Разработанная американской исследовательской фирмой, специализирующейся на безопасности, **Anonym.OS** — прекрасный выбор для пользователей, желающих защитить свои данные при походах по коварной всемирной паутине. Второй проект — **OliveBSD** — создан французским хакером Габриэлем Падерни (Gabriel Paderni). Это настольная ОС, с Fluxbox в качестве оконного менеджера, приложениями для Интернет-серфинга, обмена сообщениями, обработки графики и медиа-проигрывателями.

<http://sourceforge.net/projects/anonym-os>
<http://g.paderni.free.fr/olivebsd>



Через меню Frenzy легко найти множество сетевых утилит и утилит безопасности.

• На базе NetBSD

Ну и, конечно, поклонники **NetBSD** — системы, знаменитой своей переносимостью на дюжину архитектур — также не оставлены без внимания. Официальный Live-дистрибутив NetBSD называется **NetBSD Live** (последняя версия 2.0), кроме того, есть и настольная LiveCD-система **Newbie** с Fluxbox. Обе поддерживают только архитектуру i386.

www.netbsd.org/gallery/products.html
<http://arudius.sourceforge.net>

Hot Picks



Лучшее на планете новинки открытого ПО!

Майк Сондерс

Пописывает в Celtx очередной роман: The Adventures of Nate!



Здесь мы проводим обзор некоторых из наиболее популярных программ в мире.

Каждый месяц мы прочесываем тысячи проектов с открытым кодом и отбираем для вас самые новые, самые изобретательные и просто отличные. Большая часть программ, попавших в обзор, записана на прилагаемый к журналу диск, но мы также предоставляем и web-ссылки, чтобы вы могли скачать новейшие версии.

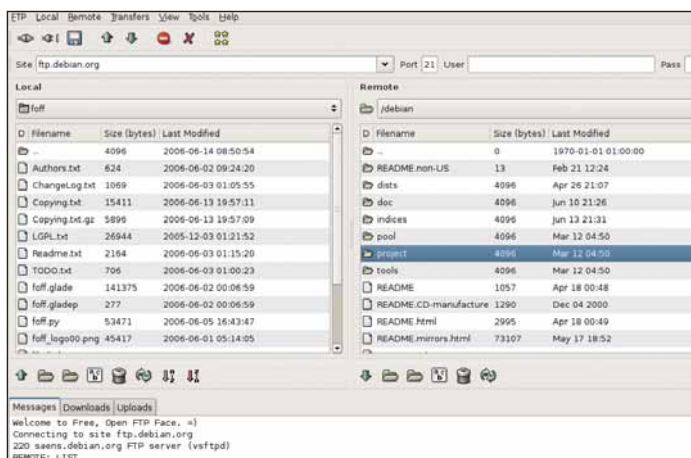
Если у вас есть идеи о том, какие проекты с открытым кодом нам стоит включить в обзор, пишите нам на адрес:

linuxformat@futurenet.co.uk

FTP-КЛИЕНТ

Foff

• ВЕРСИЯ 0.99.2 • WEB <http://foff.sourceforge.net>



Foff применяет традиционный подход: двухпанельный интерфейс.

Джефри Баккер – автор **Foff** – недоволен текущим положением дел с FTP-клиентами под Linux. Он считает, что только малая их часть следует принципу «простота и удобство», вместо этого предлагая запутанные и замусоренные интерфейсы, заслоняющие простоту, присущую FTP – ведь это всего-навсего передача файлов. Протокол FTP в наши дни с появлением загрузки в web и P2P-обмена данными используется уже не так широко, но хорошие графические клиенты по-прежнему нужны, поэтому Баккер создал **Foff** (Free Open FTP Face).

Если у вас уже есть Python, GTK и PyGTK, вы можете просто распаковать файл **Foff-0.99.2.tar.gz** и запустить **Foff.py** из получившегося каталога. Сразу становится ясно, что на дизайн интерфейса повлиял другой популярный клиент – **GFTP**: Баккер с похвалой отзывается об этом клиенте в документации, а **Foff** сохраняет его аккуратный и удобный дизайн. Меню наверху позволяет ввести адрес сервера (а также любую требуемую комбинацию имени пользователя и пароля), а под ним располагаются две навигационные панели, позволяющие выбрать файлы для скачи-

вания или закачивания. Это традиционный дизайн, который отлично работает, хотя не помешала бы поддержка drag'n'drop.

При использовании **Foff** не возникает никаких сюрпризов – вы передаёте файлы, отслеживаете состояние соединения через панель обратной связи и просто делаете то, что вам нужно. Однако приложение имеет пару дополнительных возможностей, которые выделяют его на фоне других клиентов.

доходит до просмотра файлов, они создают ощущение, что у **Foff** очень хорошая производительность.

Кроме того, в **Foff** присутствует контекстное меню, доступное по правой кнопке мыши – оно позволяет вам на лету упаковывать и распаковывать архивы. Нет нужды запускать отдельное приложение или затевать вылазку в командную строку – операция доступна по простому щелчку мышью. Если разработчик позволит вам добавлять свои действия, одним щелчком запуская требуемую утилиту командной строки, он, определённо, выиграет. Базовые файловые операции, например, переименование и удаление файлов, также интегрированы, что опять-таки позволяет вам избежать ненужных переключений в другие программы.

Документация пока что отсутствует, но интерфейс **Foff** достаточно интуитивен, и любой, кто имел дело с графическими FTP-клиентами, разберётся и с ним. Единственная проблема, которую мы увидели в **Foff** – он не запоминает размеры окна или файловых панелей после выхода из программы, поэтому каждый раз после запуска размер придётся регулировать заново.

«СООБЩЕСТВУ LINUX ПО-ПРЕЖНЕМУ НУЖНЫ ХОРОШИЕ FTP-КЛИЕНТЫ.»

В него встроен просмотрщик текстовых файлов и файлов изображений – идея в том, чтобы **Foff** выглядел и работал одинаково на любой платформе (версия для Windows также доступна) и чтобы вам не требовалось запускать никаких дополнительных программ. Просмотрщики настолько просты, что проще и не бывает, но когда

Тем не менее, это хороший знак: получается, что других серьёзных проблем в **Foff** на данный момент нет. Эта программа – отличный выбор, если вам не нужны навороты **GFTP**, а нужен небольшой и нетребовательный клиент, работающий без каких-либо проблем. Ждём версию 1.0!

HOT PICKS В ЭТОМ РАЗДЕЛЕ

Celtx	22
DOSBox	23
Foff	18
GFast File	20
KBilliards	21
Lighttpd	23
Mplay	19
NeoDraughts	21
RIB	19
Telak	20

ОБРАТИТЕ ВНИМАНИЕ НА ПОБЕДИТЕЛЯ HOT PICKS

Все, что попадает в раздел **HotPicks**, заслуживает самого пристального внимания.



Однако, каждый месяц мы выбираем всего один самый яркий проект.

Побеждает лучший!

МЕДИА-ПРОИГРЫВАТЕЛЬ

Mplay

• ВЕРСИЯ 1.0-beta1 • WEB <http://mplay.sourceforge.net>

MPlayer – по праву один из наиболее популярных медиа-плееров под Linux: он поддерживает большое количество файловых форматов, хорошо работает и содержит больше настроек, чем вам когда-либо может понадобиться (см. map-страницу!). Но из консоли его использовать не так удобно – много технической информации и мало возможностей пов-

лиять на процесс, если файл уже запущен. *Mplay* пытается решить эту проблему, предоставляя дружелюбный текстовый интерфейс для *MPlayer* с использованием плей-листов и разноцветных индикаторов статуса.

Для установки *Mplay* распакуйте файл **Mplay-1.0-beta-1.tar.gz**, перейдите в получившийся каталог и в командной

строке наберите **./install**. Вам нужно делать это под суперпользователем **root**, и в Ubuntu здесь вас поджидает проблема (во время выполнения *sudo* система не переключает **\$HOME** на **/root**). Чтобы обойти эту проблему, удалите первые три строки из инсталляционного скрипта – там, где стоит: **if ! [\$HOME = /root]**. Затем наберите **./install**, и когда программа будет собрана (для этого требуется Perl), можно попробовать её запустить.

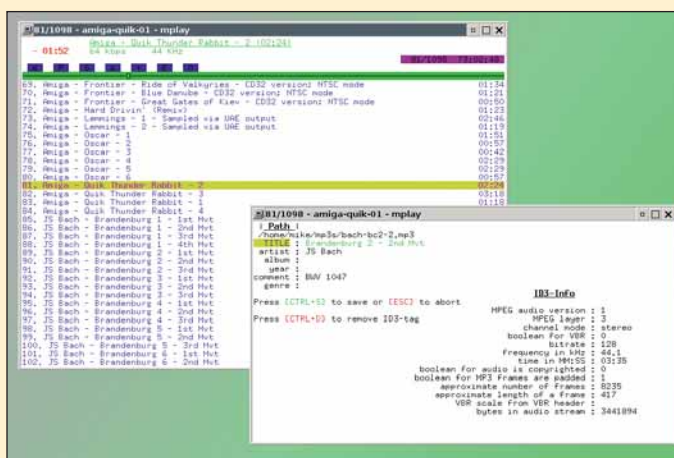
Хотя *MPlayer* может работать и с видеофайлами, интерфейс *Mplay* больше подходит для музыки, будь то формат MP3, Ogg или WAV. Наберите **Mplay <filenames>** – загрузится разноцветный плей-лист, внутри которого вы можете перемещаться с помощью курсора. Во время проигрывания файла текущую позицию в нём (отображаемую на индикаторе сверху) можно менять левой и правой клавишами курсора. Вы в любой момент можете нажать Esc и выйти из приложения, а когда вы запустите *Mplay* в следующий раз, он начнёт проигрывать последнюю песню ровно с того места, на



котором находился в момент выхода из программы.

В *Mplay* также включён микшер (нажмите x) и эквалайзер (d), а также возможность редактирования ID3-тэгов (e) и переключения между несколькими плей-листами. Эти функции описаны в файле помощи, доступ к которому можно получить, нажав h. Вы даже можете настраивать цвета различных элементов экрана (прогресс-индикатора, информации о треке, громкости звука и т.п.).

Разработчики нашли идеальный баланс между поведением программы по умолчанию и возможностями по настройке, так что если вы ищете что-то получше, чем интерфейс командной строки *MPlayer*, – попробуйте эту программу.



Mplay в режиме просмотра плей-листа (ну что за смесь – саунд-трек к *Frontier* и *Bach*!); на переднем плане – приятный встроенный редактор ID3-тэгов.

НАСТОЛЬНЫЙ ПЕЙДЖЕР

RIB

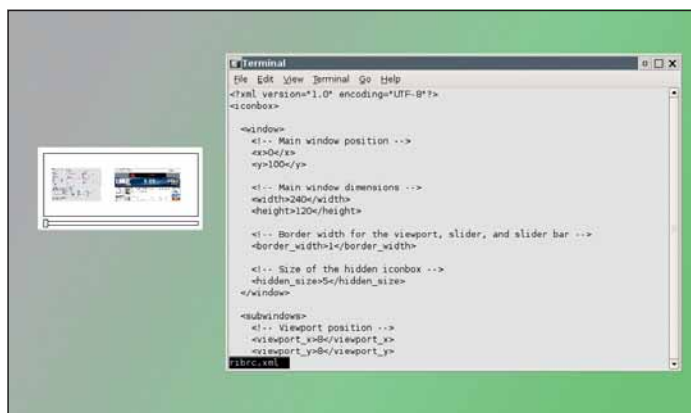
• ВЕРСИЯ 0.4 • WEB www.cae.wisc.edu/~hoeiz/iconbox.html

Давным-давно, ещё до того, как KDE и Gnome стали основными кандидатами для рабочего стола (и задолго до того, как мы могли хотя бы помыслить о *Xgl* и *Compiz*), существовал только один привлекательный на вид оконный менеджер – *Enlightenment*. Он поддерживал просвечивающее перемещение окон и симпатичный стартовый экран, а всего приятнее отображались уменьшенные копии запущенных вами приложений. Переключатель экранов в Linux всегда впечатлял новичков, особенно на фоне простеньких иконок и панелей задач.

RIB – Rob's Iconbox сделал эту функциональность доступной почти во всех оконных менеджерах, поддерживающих спецификацию EWMH (т.е. в любой популярный оконный менеджер, релизы которого выходили в последние несколько лет). Для его сборки вам понадобятся зависимости – это

lmlib2 и пакеты разработчика для X. Почти в каждом дистрибутиве в репозитории эти пакеты есть. Результирующий двоичный файл имеет размер всего 31 КБ и настраивается с помощью файла конфигурации в формате XML – скопируйте **RIBrc.xml** из архива в **~/RIBrc.xml** перед тем как запустить программу.

Впервые запустив *RIB*, вы, скорее всего, будете усиленно искать, где же его окно – программа отображает только небольшой черно-белый прямоугольник с линейкой прокрутки внизу. Но стоит вам начать минимизировать окна, как *RIB* оживает, отображая небольшие копии экрана минимизированных приложений. Теперь вы можете кликнуть на эти картинки, чтобы восстановить приложение на рабочем столе. По сути, он работает как панель задач, но гораздо более наглядно, чем при обычном подходе с текстом и иконкой.



Командно-строчный Mini-Me: иконки у *RIB* крошечные, но изображаемые ими приложения легко распознать.

Дальше – больше: теперь у вас есть крутая функциональность *Enlightenment*, доступная в любом оконном менеджере. Но в рукаве у *RIB* припасено ещё несколько трюков, которые станут доступными после редактирования вышеупомянутого файла **RIBrc.xml**. Вы можете настроить размер и позицию рамки с иконками или назначить ей выделенную зону на экране, чтобы развёрнутые приложения её не перекрывали, а также назначить время, за которое

должны генерироваться копии экранов программ. Увеличение этого времени должно приводить к улучшению качества иконок, но в нашем тесте это не дало никакого эффекта.

RIB – миниатюрное, быстрое и разумное графическое приложение для отображения страниц, которое будет отличным дополнением к любому ПК на базе Linux (особенно если у вас легкий оконный менеджер).

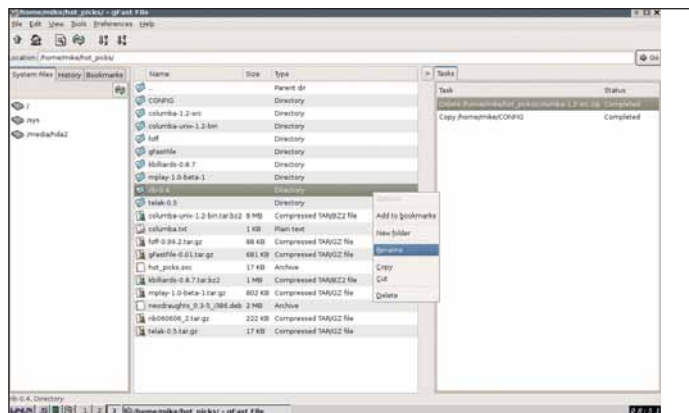
ФАЙЛОВЫЙ МЕНЕДЖЕР

GFast File

• ВЕРСИЯ 0.01 • WEB <http://gfastfile.sourceforge.net>

Мы знаем, что вы подумали: очередной файловый менеджер? Что ж, для улучшений всегда есть место, и Дэниэль Капитен [Daniel Capitan], разработчик *GFast File*, считает, что козырь его файлового менеджера – отсутствие интерактивности. Вместо того, чтобы копировать файлы, ждать, пока операция закончится, а затем приступать к другому заданию, *GFast File* использует фоновую очередь – мож-

GFast File написан на C# с интерфейсом GTK, и чтобы запустить его, вам понадобятся Mono и GTK#. К счастью, для запуска этого приложения не нужно проходить через какую-либо процедуру установки: просто распакуйте архив **gFastFile-0.01.tar.gz** и запустите **gfastfile.exe** в получившемся каталоге. Если всё пойдёт хорошо, вы увидите главное окно со знакомым интерфейсом файловой навигации.



Задания на операции с файлами помещены на правой панели.

в файловой системе. Справа находится панель файловой навигации – причём с типами файлов ассоциированы действия, но они не настраиваются через интерфейс; поклонникам KDE придётся смириться с запуском приложений Gnome!

Очередь задач *GFast File* действительно удобна: начните копировать большой файл или каталог, и эта операция появится в крайней правой панели, показывающей, выполняется ли задание или уже завершено, а также хранящей историю выполненных заданий. Можно ставить задания в

очередь и забывать о них до их завершения, избегая появления бесконечных диалогов на вашем рабочем столе.

GFast File имеет некоторые проблемы с надёжностью – пару раз во время нашего тестирования он завис, поэтому Капитен не рекомендует использовать его с критически важными файлами. Однако это всего лишь версия 0.01, и работы ещё много – следите за этим приложением и ждите нашего обзора в **HotPicks Повторный визит** в будущих номерах *LXF!*

«МОЖНО СТАВИТЬ ЗАДАНИЯ В ОЧЕРЕДЬ И ЗАБЫВАТЬ О НИХ ДО ИХ ЗАВЕРШЕНИЯ.»

но создать стек заданий. Некоторые другие файловые менеджеры также позволяют выполнять несколько действий одновременно, но не так элегантно, как это небольшое сообразительное приложение.

Левая панель содержит вкладки для переключения между System Files (корневая директория и точки монтирования), плюс историю ранее посещённых каталогов и закладки для часто используемых путей

ПРОСМОТР ИЗОБРАЖЕНИЙ

Telak

• ВЕРСИЯ 0.5 • WEB <http://julien.danjou.info/telak.html>

Telak – небольшое приложение, предназначенное для вытаскивания изображений из Интернета и помещения их на «корневое окно» (заставка рабочего стола, на X-жаргоне). Вы спросите: «А

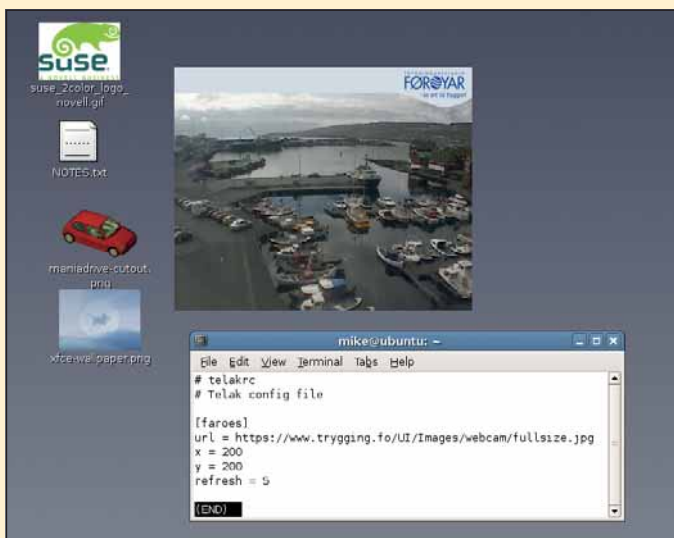
зачем?». Ну, большинство из нас имеют статические изображения на рабочем столе: трёхмерные пейзажи, фотографии мест, где мы были, и т.п. Однако в последние годы появилась тенденция ожив-

лять рабочий стол – например, сводками новостей, монитором погоды или апплетами, как в *SuperKaramba*. Пусть вы не хотите отвлекаться от основной работы, но почему бы не иметь оперативную информацию на экране, если есть незанятое место – и это позволит сделать *Telak*.

Чтобы скомпилировать и установить его, достаточно ввести **make** и затем **make install** (под суперпользователем). Для сборки программы из исходного кода понадобятся *lmlib2* и *libcurl*. После установки вручную создайте конфигурационный каталог *Telak*, выполните команду **mkdir .Telak** в вашем домашнем каталоге (войдя как обычный пользователь), а затем скопируйте файл **Telakrc.sample** из исходного архива в **~/Telak/Telakrc**. Это файл конфигурации, содержащий список URL-адресов и размеры изображений – вы можете использовать любой формат, поддерживаемый *lmlib2*, включая JPEG, PNG, GIF и многие другие.

В чём же практическая польза *Telak*? Прежде всего, вы можете настроить его на периодическое скачивание изображений из Интернета на ваш рабочий стол. Это может быть график курса акций, если вы следите за ситуацией на рынке, или репортаж с любимого вами места при помощи web-камеры. *Telak* позволяет указывать интервал обновления изображений, и если вы не хотите, чтобы он забил вам трафик, выставьте секунд 300. Если у вас широкополосный канал, а изображение меняется довольно быстро, попробуйте поставить 10 секунд.

Файл **Telakrc** позволяет указывать точные URL для изображений (или путь, если вы используете картинки с жёсткого диска), их ширину, высоту и положение на экране. Если вы желаете раскидать по своему рабочему столу сразу несколько картинок, добавьте в файл соответствующее количество записей – *Telak* справится с задачей легко и быстро.



Регулярно обновляемое изображение с web-камеры: восход на Фарерах.

НАСТОЛЬНАЯ ИГРА

NeoDraughts

• ВЕРСИЯ 0.3 • WEB <http://ndraughts.sourceforge.net>

По последним данным, на свете больше версий компьютерных шашек, чем автомобилей в Беларуси, и это число неуклонно растёт. Вы можете найти даже маленькие шашки для мобильных телефонов — разработчикам определённо осталось оттачивать мастерство только на искусственном интеллекте (а для среднего игрока и одного мегагерца достаточно, чтобы соперник оказался трудным). Поэтому хакеры *NeoDraughts* решили порадовать глаз трёхмерными боями вместо ходов по скучным двухцветным клеткам.

Те, кто помнит игру-прародителя *BattleChess*, в *NeoDraughts* почувствуют себя как дома. Традиционные шашки заменены пехотой с автоматами, готовой превратить обычный ход в эпизод насилия. Пока в игре есть только один режим — игра против компьютера, которая начинается на доске, превращённой в парящий в небесах над горами магический ковёр-самолёт. Игра не особо требовательна к ресурсам: для неё будет достаточно даже графического чипа от Intel.

Прижав правую кнопку мыши и потянув курсор, вы можете вращать камеру вокруг игровой зоны; вставка в виде двумерного поля позволяет переместить камеру к конкретной клетке. Игра проходит так же, как и в любых компьютерных шашках: кликните на клетку, где стоит фигура, которой вы делаете ход, затем кликните туда, куда вы хотите пойти. Ваш боец переместится вперёд и будет ждать хода соперника.

Манёвры в игре получились лучше всего: ваш игрок поливает компьютерного противника яростными залпами, а компьютерная армия укомплектована лазерами. Игра не так зрелищна, как *BattleChess* с его мириадами спецэффектов, но по ходу действия привносит немало веселья.

Если вы регулярно практикуетесь в шашки, то вам не составит труда обыграть компьютер в *NeoDraughts*: он выбрасывает ходы очень быстро и часто попадает в скверную позицию. Однако разработчики используют этот простой искусственный интеллект только для тестирования 3D-движка, так что, возможно, вскоре мы увидим другие игры. А как насчёт ультра-жестокоего варианта игры — «Змейки и лесенки»: *NeoSnakes And Ladders*? Есть желающие? Так или иначе, если вам захотелось сыграть во что-то поживее, чем традиционная настольная игра, и без серьёзного соперника, это шанс повеселиться.



Пейзаж игровой зоны настраивал на дурные шутки про шашки, но мы решили от них воздержаться.



Способ быть крутым #437: опоздав на встречу, скажите «Извините за задержку — заигрался в пул в джаз-клубе».

БИЛЬЯРД

KBilliards

• ВЕРСИЯ 0.8.7 • WEB www.hostnotfound.it/kbilliards.php

Бильярд, бесспорно, самая популярная игра в пабах: тут налицо должный баланс между мастерством и удачей, а у аутсайдера всегда есть шанс. Когда трудный удар получился, это выглядит очень круто, и нет большего удовольствия, чем забить все шары, даже не дав сопернику поддержать за кий.

Именно поэтому игра не получила достойного воплощения в мире компьютерного моделирования; правда, Арчер Маклин [Archer Maclean] немало поработал над этим для Amiga и PC. *KBilliards* — ещё одна попытка сделать симулятор бильярда, и как вы уже догадались по его имени, он использует библиотеки KDE (которые понадобятся для его сборки из исходного кода).

Если вы поклонник британского бильярда, вы, возможно, расстроитесь, не увидев красных и жёлтых шаров — *KBilliards* по умолчанию использует систему подсчёта, в которой последовательные попадания в лузу шаров разного цвета накапливают очки. Используя левую кнопку мыши, вы кликаете по столу, направляя красную штриховую линию в направлении удара, а правой кнопкой мыши выбираете темп удара. Перемещать красную линию можно и клавишами курсора, но, к сожалению, здесь не предусмотрен точный контроль.

Звуки соударений шаров весьма правдоподобны, зато удары о борт похожи на удары о слабо натянутую резиновую ленту, а звук закатывания в лузу и вовсе напоминает комедийные эффекты.

Весёлая фоновая мелодия, исполняемая на пианино, довольно быстро закликивается, но её можно и отключить. В остальном всё нормально, и звуковые эффекты вполне пристойны.

Во время игры шар двигается по столу весьма убедительно, но иногда возникает ощущение, что стол довольно медленный. Возможно, это намеренное воссоздание неоднократно залитой пивом скатерти, которая годами собирала пыль и грязь. Если вы хотите побаловаться с силами трения, можете слегка изменить физику игры и ускорить движение шаров (Settings > Preferences). Установите константу 'Deceleration constant' в значение 0.000001, и вы увидите, как шары буквально летают по столу.

В *KBilliards* есть режим тренировки, позволяющий вручную расставлять шары, а также режим двух игроков для игры против компьютера. К сожалению, соперник-компьютер не обдумывает удары, так что вы увидите шквал попаданий в лузу и промахов. Пока нет возможности выставить уровень компьютерного игрока — он либо нереально точен, либо выбирает бессмысленно-случайное направление удара. Тем не менее игра неплоха! *KBilliards* — приятная бесплатная модель бильярда, идеальная для блиц-партии, не занимающей больше пяти минут.

МЕДИА-ПАКЕТ

Celtx

• ВЕРСИЯ 0.9.5.1 • WEB www.celtx.com

Когда Netscape открыл исходный код своего web-браузера в форме *Mozilla*, команда разработчиков собиралась двинуть его в новом направлении, создав не просто набор Интернет-программ, а платформу. Они разработали XUL (XML UI Language) в надежде, что программисты создадут новые приложения вокруг *Mozilla*, которые помимо прочего будут переносимы.

К сожалению, XUL не оправдал надежд, а команда *Mozilla* вернулась к Интернет-источкам и создала *Firefox* и *Thunderbird*.

Поэтому мы были очень рады услышать про *Celtx* (произносится «келтикс») от читателя LXF Марка Холдера [Mark Holder]. Это медиа-пакет, созданный Greyfirst Corp по технологии XUL, и одно из первых больших приложений на платформе *Mozilla* (если не считать среды разработки *ActiveState*

Поддержка его пользователей осуществляется через форумы и электронную почту. Чтобы установить *Celtx*, скопируйте файл *Celtx.tgz* с нашего диска, распакуйте его в ваш домашний каталог и запустите *Celtx* в получившемся каталоге.

Celtx предназначен для написания сценариев к фильмам или пьес, но может быть приспособлен также и под книги или даже радиопередачи. Приложение поможет создать сценарий, персонажей и декорации, и содержит некоторые функции менеджера, например, график работ. После старта *Celtx* открывает симпатичный интерфейс — легко догадаться, что внутри *Mozilla* и XUL, но разработчики придали приложению индивидуальность.

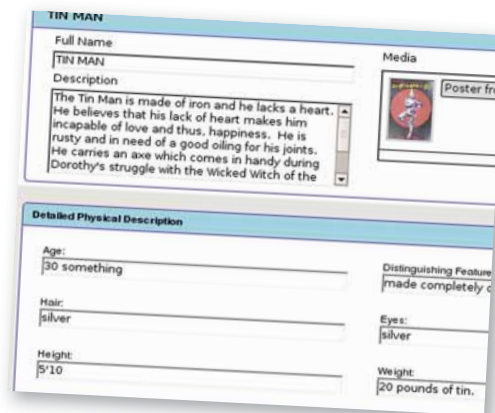
Лучший способ исследовать функции *Celtx* — открыть учебный пример проекта,

сценарий, титульный лист, персонажи, декорации, график работ и отчеты. Разработчики также встроили удобный web-браузер на основе *Gecko*, и вы можете отыскать в Google или Wikipedia варианты создания 'dramatic tension' (драматическое напряжение) или 'movie cliches' (киношные штампы) и многое другое, не выходя из проекта (вдобавок это снижает требования к памяти). Части интерфейса хорошо сосстыкованы, с пользой расходуют пространство экрана, не захлывая его, и не отвлекают вас от задачи.

Интрига затевается

Редактор сценариев *Celtx* очень способный и форматирует текст различными способами, выделяя диалоги, действия и ремарки. Можно выполнять вручную основные операции форматирования (жирный шрифт, курсив, подчеркивание); предусмотрена и проверка орфографии.

Если вы вводите в сценарий новый персонаж, он тут же чудесным образом появится на отдельном экране *Characters*, где вы можете привязать к нему портрет, указать его характерные черты и задать линию поведения — это очень удобно при работе с соавторами, чтобы они не «искажили образ» в ваше отсутствие.



Опишите характер вашего персонажа, чтобы ничего потом не перепутать.

На экране *Scene Details* перечисляются ключевые события, персонажи-участники и их настройки в каждой части проекта; по мере развития событий можете в окне *Schedule* (базовый календарь) записывать, что и как делать, время действия и т.п. Наконец, после создания сложной комбинации сцен, персонажей и подбора актеров вы можете взглянуть на итоговый отчет, отображающий состав вашей работы. Заодно составляется примерный список требуемых материалов — например, бутафория, музыка, животные и даже трюки, если вы готовите сцену в стиле Джеки Чана.

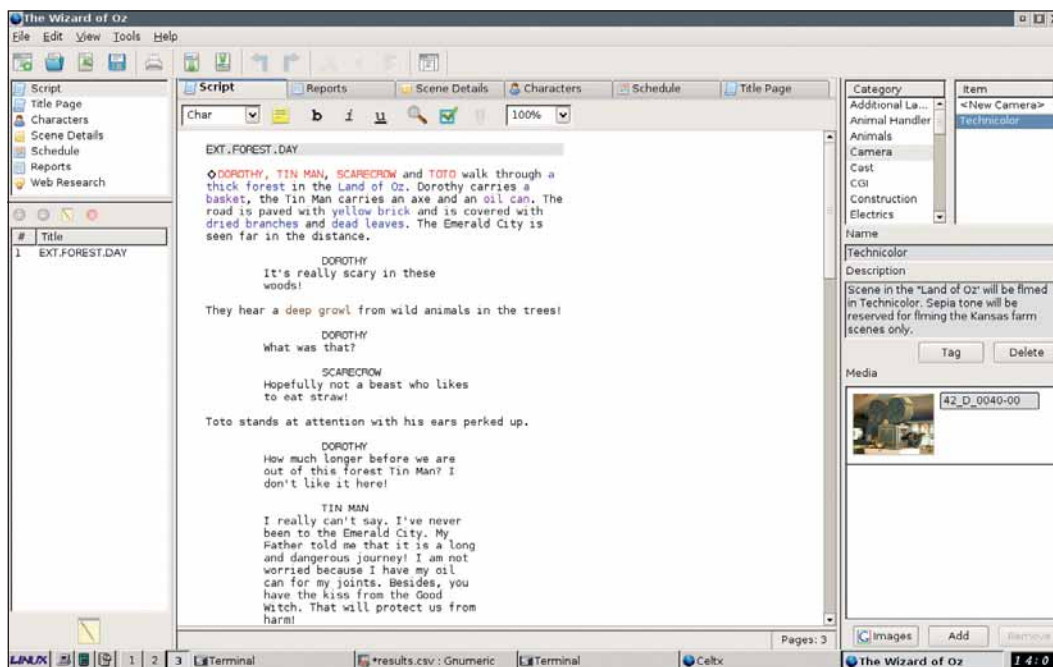
Celtx поддерживает набор онлайн-возможностей для публикации сценария и генерирует его версии в формате HTML и PDF. От возможности закачать вашу работу на сервер *Celtx* и пригласить других к участию в проекте просто дух захватывает, и сайт проектов <http://pc.Celtx.com> уже завален работами. Пока что документации почти нет, кроме недоделанных описаний с сайта *Celtx* wiki, но программа настолько проста в использовании, что проблем не возникает.

Хотя *Celtx* еще не достиг версии 1.0, многие писатели в сети уже сравнивают его с *Final Draft*, который считается промышленным стандартом для сценаристов. Но даже если вы всего лишь работаете над небольшим рассказом и вам не нужен полный набор возможностей *Celtx*, вам точно стоит попробовать его в действии: его способность организовать и каталогизировать персонажей, сцены и сценарии очень помогает не упускать из под контроля ваш рассказ и стыковать его части между собой. Огромное спасибо Greyfirst за разработку отличного бесплатного приложения и создания вокруг него сообщества.

«ПОМОГАЕТ В СОЗДАНИИ СЦЕНАРИЕВ, ПЕРСОНАЖЕЙ И ДЕКОРАЦИЙ.»

Komodo, — прим.ред.] *Celtx* работает на Linux, Mac OS X и Windows и выпущен под лицензией CePL — созданной на основе MPL (*Mozilla Public License*), так что его можно бесплатно использовать и распространять.

The Wizard of Oz (Волшебник страны Оз): затеяв новый проект, вы окажетесь наедине с набором пустых текстовых полей и разделов. В верхнем левом поле помещен виджет-список различных аспектов про-



Celtx в режиме редактирования сценария; работа проекта иллюстрируется примером.

HOTPICKS ПОВТОРНЫЙ ВИЗИТ

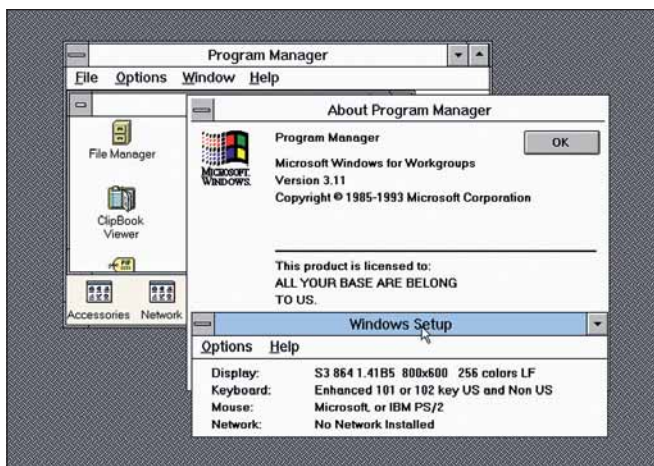
ЭМУЛЯТОР DOS И PC

DOSBox

• ВЕРСИЯ 0.65 • WEB <http://dosbox.sourceforge.net>

Это не просто пересмотр DOSBox, а пересмотр пересмотра! Мы описывали этот эмулятор DOS и PC в LXF48, затем ещё раз в LXF63. Тогда была версия 0.63, сейчас 0.65. Вы, возможно, считаете, что это означает всего лишь мелкую подвижку, но на самом деле проделан существенный объём работы. DOSBox создаёт виртуальный ПК на вашем компьютере, но, в отличие от Qemu или Vochs, исключительно для запуска классических игр DOS.

Главное отличие от версии 0.63 – поддержка захвата видео с экрана и сохранение его в формате AVI. Также добавлена поддержка VESA-режимов 4/15/16/32BPP для лучшей совместимости видео и исправлено виртуальное VGA-устройство, что позволит запускать больше игр. DOSBox никогда не приходилось стыдиться своей производительности, но оптимизация эмулированного процессора продолжалась, и те, кто отмечал подтормаживания игр в прошлых релизах,



Цель DOSBox – игры, но можно запускать и «серьёзные» программы, типа Windows 3.1 (хотя тут больше подошёл бы DOSEmu).

теперь обнаружат некоторое улучшение. Реализована поддержка эмуляции VCP1 и исправлены тонны ошибок. DOSBox всё

хорошеет, и для эмуляции классических игр DOS в мире нет лучшего решения – оно работает.

WEB-СЕРВЕР

Lighttpd

• ВЕРСИЯ 1.4.11 • WEB www.lighttpd.net

Обзор версии 1.3.6 Lighttpd мы делали не так давно (LXF63). С тех пор проект обзавёлся новым релизом, не говоря уж о небольших обновлениях и улучшениях. Этот компактный и нетребовательный к памяти web-сервер восхитил нас первоклассной документацией, простой установкой и про-

изводительностью в четыре раза большей, чем Apache при обслуживании статических страниц. Благодаря крошечному размеру его даже ухитрились приспособить к наладонным устройствам вроде Sharp Zaurus – это впечатляет.

Среди новых возможностей версии

1.3.6 – модуль WebDAV, поддержка вложенных условий в конфигурационном файле, поддержка интерфейса шлюза Simple Common Gateway Interface (SCGI), модуль виртуальных хостов на базе MySQL и сортировка в mod_status на базе JavaScript. Вдобавок проведена огромная работа по оптимизации производительности сервера, а многие модули теперь документированы лучше, чем Британская библиотека.

Маленьких web-серверов существует бесчисленное множество, но преимущество Lighttpd в его зрелости, широком использовании и регулярном обновлении. Он становится популярным выбором для программистов Ruby on Rails, так что в его будущем можно не сомневаться. Если ваш web-сервер сильно нагружен, переведите его на Lighttpd – прирост производительности гарантируется. LXF

Directory Listing for /upload/

date	size	type	name
2003-05-11 18:57:44	272	directory	./
2005-01-30 20:21:34	9056	directory	../
2003-05-11 10:20:35	46036	image/jpeg	0049.jpg
2003-05-11 10:20:21	188349	application/pdf	307.pdf
2003-05-11 10:19:52	1100416	application/pdf	81tx+d.pdf
2003-05-11 10:21:58	18308	application/octet-stream	B34b.JPG
2003-05-11 10:22:41	50037	application/x-gzip	access_log_2003-02-18[1].txt.gz
2003-05-11 10:20:08	1040331	application/pdf	cederqvist-1.11.3.pdf
2003-05-11 18:57:44	2338	text/css	style.css

Если нет файла index.html, Lighttpd автоматически генерирует аккуратный список файлов.

ТАКЖЕ ВЫПУЩЕНЫ

Новые и обновленные программы, заслуживающие внимания...

- **WikiBlog 0.10** CMS система с Ajax и wiki-интерфейсом www.wikiblog.com
- **NanoVM 20060613** Ультра-маленькая виртуальная машина Java (меньше 8 КБ) www.harbaum.org/till/nanovm
- **PyQt 4.0** Привязки Python для Qt www.riverbankcomputing.co.uk/pyqt
- **DupeFinder 1.0.2** Очистка файлов-дубликатов <http://freshmeat.net/projects/dupefinder>
- **Paymaster 0.9.6** Приложение для расчёта зарплаты и управления кадрами www.treshna.com/paymaster
- **Fluxbox 1.0rc** Стильный и быстрый оконный менеджер www.fluxbox.org



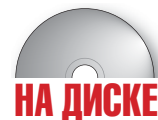
- **MultiChat 0.8** Клиент для чата и общения <http://software.pt-link.net>
- **GmNotify 0.7.3** Нотификатор почтовых сообщений Gmail в системном трее <http://gmnotify.sourceforge.net>
- **Dolphin 0.5.1** Сфокусированный на удобстве файловый менеджер KDE <http://enzosworld.gmxhome.de>
- **Wired 0.2.2** Профессиональный пакет для создания музыки <http://wired.is.free.fr>
- **Gnome Hearts 0.1** Карточная игра с компьютерным соперником www.jejik.com/hearts



- **Reft 0.3** 'Ridiculously Easy File Transfer' – до смешного простая пересылка файлов www.raftkind.com/jon/showproject.php?id=17
- **F4L 0.2.1** Среда разработки Macromedia Flash <http://f4l.sf.net>
- **Serpentine 0.7** Запись аудио-CD <http://s1x.homelinux.net/projects/serpentine>
- **KSquirrel 0.6.3** Просмотр изображений KDE <http://ksquirrel.sourceforge.net>
- **Cjots 0.20** Ведение заметок на консоли <http://cjots.sourceforge.net>
- **Sylpheed-Claws 2.3.0** Почтовый клиент с GTK-интерфейсом www.sylpheed-claws.net

Что за штука...

GEGL?



Не один год нам сулили, что *GEGL* (*Generic Graphic Library*) будет огромным скачком в развитии *Gimp*, но по жизни мы ждем его до сих пор. Майкл Дж. Хэмел пробует сунуть туда нос.

>> Гегель какой-то... не надо мне этого. Пусть *Gimp* сделает поддержку CMYK, как в *Photoshop*. Когда она наконец появится?

Тогда же, когда и *GEGL*. Видите ли, у разработчиков *Gimp* возвышенный принцип: «всё делать правильно», а не «наскоро сляпать». Поддержка CMYK не была их изначальной целью – они собирались просто добавить 16-битные цветовые каналы. Но когда прикинули фактический объём работ, осознали: потребуется полная переделка внутренней архитектуры. Вот *GEGL* и является такой архитектурой.

>> И что, переделка RGB в CMYK – это целый проект?

Нет, новый проект требуется для преобразования *Gimp* из доведка рабочего стола в мощный графический пакет. *GEGL* не просто изменит работу с цветом – улучшится также управление памятью, благодаря мозаичной обработке и новым процессам работы с графикой; будет осуществляться протоколирование всех ваших действий, чтобы вы потом могли легко воспроизвести их.

На самом деле перевести изображение из RGB в CMYK проще простого. Но это лишь часть проблемы.

>> «Спят усталые игрушки, книжки спят...»

Подождите, не засыпайте: всё и правда просто! RGB и CMYK – два типа цветовых моделей. Они являются подмножествами гаммы видимых цветов для различных типов носителей. Модель RGB основана на цветах, воспроизводимых люминофорами экрана дисплея. Модель CMYK описывает ситуацию с воспроизведением цветов в технологиях печати.

>> Стоп-стоп. Что такое цветовая модель?

Цвет – тема довольно сложная, но попробую изложить её кратко. Цвет – это излучаемый (как в мониторе), либо отраженный (как на печатном оттиске) свет. Восприятие определенного цвета зависит от многих факторов, например, от температуры источника отражаемого света. Цветовая модель – как раз RGB или CMYK, запомните – просто абстрактный способ представления цветов с помощью наборов

И как скажет вам команда разработчиков *Gimp*, работа в различных цветовых пространствах требует определенной сложности от графического редактора. *GEGL* предоставит возможность выбора работы в любом цветовом пространстве – вместо простого преобразования файлов, и все это войдет в состав *GEGL*, чтобы пользователю не пришлось беспокоиться о дополнительных модулях. Звучит как китайская грамота?

“ПРЕОБРАЗОВАТЬ RGB В CMYK ПРОЩЕ ПРОСТОГО. НО ЭТО ЛИШЬ ЧАСТЬ ПРОБЛЕМЫ.”

чисел. Числа лишь маркируют определенные точки в пространстве цвета, ограничивая ту часть общей гаммы цветов, которую способна охватить данная цветовая модель. Вы всё ещё здесь?

>> Пока да.

Хорошо. Визуально цветовая модель обычно представляется в трехмерном виде: представьте куб, раскрашенный всеми цветами радуги – это цветовое пространство, координатные оси которого определяются цветовой моделью. По ним мы и выбираем нужный цвет. [Вы, может быть, встречали такие обозначения – R255G255B255; в 8-битной модели RGB (т.е. 8 бит данных отводится на значение цвета одной точки в одном канале, что дает $2^8=256$ градаций цвета по каждой из трех осей пространства – от 0 до 255) оно означает точку с максимальным значением красной, зеленой и синей составляющих, т.е. белый цвет – прим.ред.]

>> Угадали.

Существует очень важная цветовая модель – CIE L*a*b [CIE – Commission International d'Eclairage, Международная комиссия по освещению, – прим. ред.]. Это международный стандарт, определяющий полный набор цветов, который могут воспринимать наши глаза (L означает освещенность, а и b – соответственно, красно-зеленая и желто-синяя оси). RGB – уникальная по отношению к CIE L*a*b модель, потому как многие другие модели (включая CMYK) необходимо преобразовать в RGB, чтобы стало возможным их однозначное соотнесение с цветами CIE L*a*b.

Поэтому думайте о RGB и CMYK как о средствах воспроизведения цвета, определенного L*a*b, но только при определенных условиях.

>> Что же это за условия?

Гамма цветов одной цветовой модели (скажем, RGB) отличается от гаммы другой (скажем, CMYK).

Распечатайте на принтере фотографию на принтере и поднесите ее к экрану. Цвета не совпадают? *Gimp* необходимо сообщить об условиях печати на принтере для того, чтобы полученный отпечаток полностью соответствовал изображению на мониторе. Это отнюдь не простая задача.

>>> И GEGL, стало быть, её решит. А с чего мы заговорили о принтерах?

Я же говорю: разработчики обнаружили, что поддержка CMYK потребует куда больше работы, чем просто добавление нового пункта меню. Конечно же, *GEGL* решит проблему. Код *GEGL* позволит *Gimp* работать в различных цветовых пространствах. А в сочетании с информацией о принтерах (также известной как ICC-профили) поможет уравнивать печатные оттиски и изображение на экране. Это очень важно для тех, кто работает в области полиграфии, а также при редактировании видео в *Gimp*.

>>> Вот как, редактирование видео?

Теоретически — да. Но в действительности вы, возможно, будете выполнять работу с помощью дополнительного модуля, который будет загружать части фильма для редактирования. Вы не будете сразу загружать весь фильм в *Gimp*. В видеомонтаже свои вопросы, с которыми должен разбираться *GEGL*.

>>> Правда? Расскажите.

Хорошо. Давайте взглянем на задачу с точки зрения конечного пользователя. Представьте, что открываете отсканированную фотографию в *Gimp*. Одно из первых ваших действий — коррекция гистограммы Уровней. Если вы используете Автоподстройку в диалоге Уровни, то пиксели, близкие к черному, станут чёрными, а близкие к белому станут белыми, а все прочие, что находятся посередине, распределятся так, чтобы образовать хорошую гистограмму. Однако на некоторых изображениях это может привести к пробелам на гистограмме.

>>> Видал я такое, но в голову не брал. Что эти пробелы значат?

Это значит, что из-за ошибок в округлении при вычислениях изображение теряет качество. При работе с фотографией это не страшно. Но при видеомонтаже, когда у вас 60 кадров в секунду, об этом стоит побеспокоиться.

>>> И как это можно исправить?

GEGL будет использовать 16 бит на канал, а не 8, как сейчас. Дополнительные разряды уменьшат ошибки округления и позволят построить более гладкие гистограммы. В результате коррекция цвета будет произведена точнее. Благодаря 16-битным цветовым каналам, *GEGL* обеспечит и улучшенную обработку изображений.

>>> Чудесно. Только... что такое цветовой канал?

Помните RGB? R означает красный канал, G — зеленый, B — синий. При использовании *GEGL* вы сможете работать с другими типами цветовых каналов, пока же *Gimp* знает только эти (а также альфа-канал, он же канал прозрачности). Каждый 8-битный канал может определить цвет при помощи целого числа от 0 до 255, т.е. мы получаем более чем 16 млн. цветов — вроде бы солидная цифра.

С *GEGL* (при 16-битном цвете) эта цифра увеличится до 65535 значений на канал, что даст свыше 281 триллиона цветов — гораздо больше, чем способен различить глаз. Преобразование цвета пикселя происходит по определенной формуле, причём результат может оказаться значением с плавающей точкой, тогда как каналы оперируют только целыми числами. Ошибки округления при работе с 8 битами вызывают появление пробелов на гистограмме уровней. При работе с 16 битами такие пробелы должны исчезнуть.

>>> Значит, *GEGL* осчастливит редакторов видео и печатных плакатов; мы получим грамотные цветовые пространства.

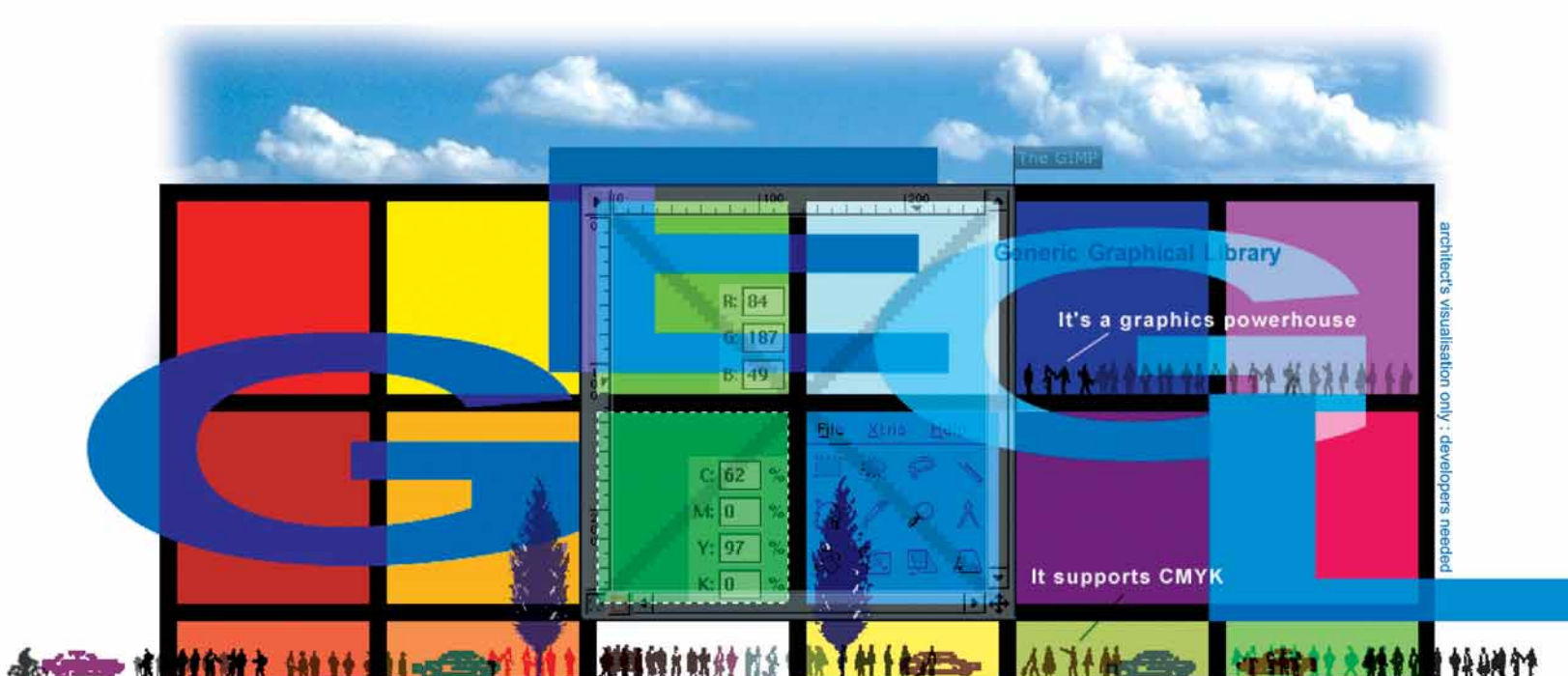
Как я уже сказал, *GEGL* не просто разрешит вопросы с цветами. *GEGL* — полноценная система обработки изображений. *Gimp* будет использовать ее только как внутреннюю часть для задач обработки. Некоторые другие функции *GEGL* обеспечат чтение и запись графических файлов, геометрические операции, преобразование Фурье и свертки. Также будет предоставлен мозаичный механизм для работы с очень большими изображениями. В конечном счёте будет возможна распределенная обработка изображений по сети.

>>> К чему все это приведет?

К тому, что *Gimp* будет работать быстрее. Следует признать, что для среднего пользователя эти улучшения будут пустым звуком, кроме улучшенного управления цветом (т.е. добавления поддержки CMYK). Но для промышленного применения это означает возможность распределения работы по сети. Программы 3D-рендеринга уже используют такую возможность, потому что они выполняют трудоемкие вычисления. При работе с очень большими изображениями или большим числом слоев (как, например, при работе с видео) *GEGL* позволит распределить нагрузку по сети и применять общие схемы обработки ко множеству изображений.

>>> Поэтому разработка и длится так долго? *GEGL* обещали уже довольно давно.

Вы правы: хотя *GEGL* был анонсирован в 2000 году, его все еще нет. Это даже ещё не даже публичный релиз (хотя CVS-срез находится на диске к журналу). Проекту требуются разработчики, чтобы выполнить список поставленных задач (домашняя страница проекта — www.GEGL.org, а список задач — на сайте http://cvs.gnome.org/viewcvs/*checkout*/GEGL/TODO). В настоящий момент только один активный разработчик вносит изменения в CVS. *GEGL* выпустят, как только проект получит достаточно разработчиков и выполнит поставленные цели. Вот так всё просто. **LXF**



LXF ИНТЕРВЬЮ



ПРИВИВКА OPENSOURCE:

«УКОЛОЛСЯ И ПОШЕЛ!»

Налицо стопки денег. Взаправду. Сертифицированные, интегрированные стеки открытого исходного кода. Если предприятие собирается использовать Linux, их понадобится много, а SpikeSource, под руководством Ким Полиз [Kim Polese] и Муругана Пала [Murugan Pal], их предоставляет.

Вобразите, что вы сидите на званом обеде, а рядом с вами – Майк Сондерс. И вдруг, дежурно поболтав о семье и о работе, Майк раздражается комментарием по поводу *Firefox*, оказывается, что вы оба – ярые поклонники открытого кода, к взыску потрясению остальных гостей. Майк рассказывает вам, насколько продвинулась современная технология, и намекает, что Linux в его нынешнем составе может сэкономить тысячи в вашем бизнесе. Но для вас это – просто хобби, и вы пока не используете его в своем бизнесе по следующей причине: кто

будет интегрировать все ПО и настраивать его? Кто все проверит и сертифицирует? В конце концов, ребята из *MySQL* или *Apache* этим заниматься не будут. А кто будет осуществлять поддержку вашего программного обеспечения и сопровождение?

И тогда Майк вам поведает – если, конечно, не отвлечется на приставку Nintendo в хозяйской гостиной – что вам нужна фирма *SpikeSource* из Редвуд-Сити (Redwood City), Калифорния: она продает «готовые решения для бизнеса», облегчающие предприятиям задачу перехода на Open Source, и привлекла к себе большой интерес не только благо-

даря своим важным и интригующим услугам, но и благодаря тому, что ее поддерживают люди с громкими именами – например, Брайан Белендорф [Brian Behlendorf] и Тим О'Рейли [Tim O'Reilly]. Выстоит ли их бизнес-модель? Будет ли от нее отдача для сообщества? И вообще, что такого важного в тестировании? Пол Хадсон встретился с руководителем фирмы, Ким Полиз [Kim Polese] (той самой, которая придумала название Java – а может, это и не она придумала) и техническим директором Муруганом Палом [Murugan Pal] (на фото – в центре), чтобы во всем разобраться.

Linux Format: Итак, *SpikeSource*. Чем вы занимаетесь?

Ким Полиз (КП): Муруган и Рей [Лейн, Ray Lane, со-основатель *SpikeSource*] весной 2003 года разглядели насущную потребность, вызванную ростом популярности открытых программ: управление ими – и, в частности, организация взаимодействия – становилось все более дорогостоящим для предприятий. Муруган понял, что для решения этой проблемы необходима новая технология: автоматизированные тесты, которые обеспечивают и нам, и предприятию независимость от поставщика и дают клиенту выбор и гибкость. Целью было дать потреби-



КИМ ПРО «ОЧЕРЕДНОЙ ДОТ-КОМ»:

«Я заподозрила бы то же самое, не будь это реальной проблемой, для которой у нас есть решение.»

телю широкий выбор операционных систем, баз данных, серверов приложений, выбора языков и многие десятки компонентов, которые в то время стали появляться.

И тогда Муруган принялся создавать команду, а команда создала рабочую среду, и сегодня мы проводим 26 000 тестов по шести операционным системам, шести языковым средам исполнения, почти 80 компонентам, 189 файлам конфигурации, 273 параметрам... Перед нами – масштабная проблема, поэтому технологический процесс невозможен без автоматизации.

Это было первое открытие. А второе было таким: нужна не только автоматизация, но и вовлечение... невозможно построить тестовую среду, которая работала бы со всей комбинаторикой возможностей, если вы – одна фирма. Вам нужна помощь коллективного опыта сообщества. Так что это – двусторонний канал: тестовая среда была спроектирована не только для автоматизации всего, начиная с отслеживания ошибок в

базах данных и до тестовой сертификации сборок, поставок и обновлений, но и для интеграции с результатами тестов от сообщества, так что мы непрерывно расширяем базу знаний о конфигурациях.

LXF: Последние три года о *SpikeSource* было не слышать. Полагаю, вы занимались хождением по инстанциям.

Муруган Пал (МП): Точно. С мая 2003 года и до тех пор, пока Ким не вошла в Совет директоров в августе 2004, мы работали в [венчурной фирме] *Kleiner Perkins* и многие попросту не знали, чем мы занимаемся. А мы создавали эту нашу тестовую среду, а затем собрали команду архитекторов или основателей других компаний.

LXF: Вы позиционируете себя как лидера тестирования Open Source. Это несколько неожиданно!

КП: Не уверена, что мы это делаем. Мы не можем объявить себя лидерами тестирования Open Source, потому что тестирование требует участия коммиттеров, разработчиков и проч., которые

на самом деле собираются и объединяют свои знания.

МП: Мы называем это совместным тестированием. Это единственный способ достичь широкого охвата. В основном это – производное от концепции участия в архитектуре, которую проповедует Тим О'Рейли. Нам нужно, чтобы к нам приходили люди и принимали участие в тестировании вместе с нами, потому что сами мы этого сделать не можем. Как мы можем пойти и клонировать 30 лет работы и 15 000 сотрудников, работающих в *Microsoft*? Единственный способ – работать совместно с сообществом. Недавно к нам пришли ребята из числа работающих над тестами для *Perl*, и сказали: «Эй, у нас тут 900 тестов, можно, мы передадим их вам, а вы их проведете?» Еще один пришел к нам и сказал: «Я написал тесты для *Perl*. Если вы решите поделиться этими технологиями с потребителями, может, вы и их туда включите?» Вот именно это нам и надо.

КП: Мы не разгуливаем повсюду, объявляя себя лидерами...

LXF: А на web-сайте именно так.

МП: Я думаю, наша цель – привлечь участников.

КП: Нет, мы не...мы очень болезненно относимся к тому, что нас...

LXF: Называют лидером? Или что вы сами себя называете лидером?

КП: К заявлениям, что мы – лидер. Мы внедрили эту автоматизированную систему тестирования и, по моему мнению, сделали это уникальным способом, но в то же время мы открыты для изучения нового и готовы поделиться нашими знаниями с сообществом.

LXF: Один из слайдов на презентации Ким, где я присутствовал, вообще-то меня запутал. Вы показали временные рамки релизов – по-моему, это были *Red Hat*, *Apache*, *MySQL*, *Ajax* и что-то еще, и я подумал: «Это бессмысленно». *Ajax* не тот продукт, у которого есть релизы, *Red Hat* уже следит за *Apache* и *MySQL*. И я хочу знать, где же вы входите в это уравнение, потому что релизы *MySQL* никого не волнуют, все думают о релизах *Red Hat*.



«КП: Целью того слайда было продемонстрировать разнообразие релизов и всевозможных компонентов, и что предприятия остаются один на один с постоянной необходимостью интеграции, им приходится иметь дело с новыми релизами и заплатками. А формализованной структуры, как, например, в Microsoft, где можно собрать все заплатки, обновить их и затем предоставить, просто нет.

LXF: Ну, есть ведь Red Hat Enterprise Linux: вы платите за нее, и компания дает вам все продукты и все заплатки целых семь лет, верно?

МП: На самом деле ключевой момент здесь в том, что Red Hat осуществляет поддержку

избыток динамических библиотек, дублирующих друг друга.

Это – проблема DLL hell [ада динамических библиотек, – прим. пер.]. Вы даже не знаете, что там внутри – даже у тех версий, которые поддерживаются официально – поэтому мы их просматриваем и удаляем избыточные копии. Еще вы говорили об Ajax или Red Hat, и в том-то и красота всей модели в целом – в ней присутствует гибкость, а люди хотят сами использовать разные программы.

LXF: Red Hat может сказать, что они не поддерживают программы типа *Geronimo* потому, что они узко целевые. Сможете ли вы конкурировать с ними по таким базовым продуктам, как *MySQL*?

чтобы вместе с ним работали JK2, Tomcat, *Connector/J* и *MySQL*. И потребителям вовсе не нужны отдельные базы данных по ошибкам, им нужен один человек, которому можно позвонить. Если у потребителей уже сложились отношения с Red Hat и *MySQL*, и, может быть, со *SpikeSource* по поводу всех этих «длиннохвостых» компонентов, то они позвонят в *SpikeSource*, а мы будем полагаться на поддержку Red Hat, существующую для *MySQL*, и сами свяжемся с ними, чтобы решить вопрос.

LXF: А если кто-то из ваших клиентов отправится в Oracle, приобретет у них *Oracle*, Red Hat Enterprise Linux, а заодно и поддержку всего этого, а потом позвонит им и скажет: «Здравствуйте, Oracle, мой сер-

вер *Netscape Enterprise Server*, потом еще дополнительные модули NSAP, *Oracle PL/SQL*, соединение с базой данных *Oracle*, база данных, работающая на Solaris или HP-UX. Если эта цепочка ломалась, нельзя было позвонить в одно какое-то место. Эта взаимосвязь по-прежнему остается проблемой, и в открытом коде, и в коммерческих программах. Но кое в чем открытый код дает потребителям больше возможностей: если инженер потребителя умнее или способнее наших инженеров, угадайте, что произойдет? Они сами устранят неисправность. И они сообщат нам об этом, а мы это протестируем и подтвердим. Вот почему в тестировании настолько важно участие.

LXF: А какие части вашего стека приложений действительно с открытым кодом?

КП: Все, полностью.

LXF: Java – часть вашего стека?

КП: Мы сертифицируем по Java, как и по Perl, Python, PHP...

LXF: Но это – часть вашего профессионального ядра?

МП: Если вы установите наше основное ПО на сегодня, для Java VM есть отдельная лицензия, потому что мы должны соблюдать права Sun. Но на все остальные части, как уже упоминала Ким, мы придерживаемся модели сквозного лицензирования. Это означает, что если мы имеем дело с *Apache Software License (ASL)*, или *BSD*, или *GPL*, то мы их сохраняем.

LXF: Вашим клиентам не нужен юрист, чтобы просматривать все индивидуальные лицензии на наличие нарушений?

КП: Сейчас есть фирмы, например, *Black Duck* и другие, которые берут на себя эту задачу. Это не наше дело, но мы с ними сотрудничаем. Например, у нас есть инструмент *asset manager*, который мы интегрировали с калькулятором лицензий *Black Duck*.

LXF: Вы пытаетесь представить сообществу. Какой поддержки вы ожидаете?

КП: Участия в тестировании. Мы ищем тех, кто будет участвовать в наших тестах и внести в них свой вклад.



определенных продуктов, как, например, *Apache*, а потребители хотят получить разные программы, им нужен *Geronimo* для работы в их среде, и чтобы *JBoss* тоже работал в их среде. Red Hat этого не поддерживает, по крайней мере, не в виде интеграции.

Во-вторых, есть такая вещь, как библиотеки классов, как, например, *Struts* и *Hibernate* и микроядро *Spring*, на сегодняшний день официально они не продвигаются, хотя потребители ими уже пользуются. Есть несколько фирм, которые, как нам известно, используют 38 компонентов этих библиотек классов; любопытно, что наш *Spike Asset Manager* (это всего-навсего простой скрипт *Python*, инвентаризирующий все компоненты с открытым кодом), когда его запускаешь для всех этих штук, обнаруживает в этих компонентах шестикратный

МП: Прежде всего, это не вопрос конкуренции. Это – вопрос сосуществования, а не замещения. Для нас Red Hat – важный партнер, есть немало серверов, работающих на Red Hat, они лидируют на рынке, и для нас очень важно партнерство с ними.

В то же время, *MySQL* и Red Hat... если у них уже сложились отношения, то для нас это хорошо, потому что потребителю сначала нужен *Apache*, а потом нужно,

вер не работает. У меня стоит *Geronimo* от *SpikeSource*, можете мне помочь?» *А Oracle* ответит: «Звоните в *SpikeSource*, а не нам». И, мне кажется, вряд ли кому понравится такое услышать.

МП: Видите ли, я сам из Oracle, и этого навиделся. Давайте забудем пока об *Open Source*, и *Geronimo* с *Tomcat*. Вернемся к тому, что у нас было в 1996 году. У нас была операционная система Solaris, сер-

КИМ О ПОТРЕБНОСТИ В SPIKESOURCE:

«Предприятия постоянно вынуждены иметь дело с новыми релизами и заплатками. А структуры нет.»

МУРУГАН О ВОЗМОЖНОСТЯХ:

«Потребителям нужно варьировать ПО. Red Hat этого не поддерживает.»

LXF: То есть вы хотите, чтобы люди давали свои результаты *SpikeSource*?

МП: Да, но взаимно. Как говорит Ким, это всегда встречное движение. Если посмотреть на тестирование, как на процесс, то имеется концепция, называемая тестирование с охватом всего кода: вы проводите тест и проверяете, сколько строк кода, или ветвей, или патчей было протестировано. Для C, C++ и Java такие инструменты есть, а для PHP такого не было. И поэтому один из инженеров [в *SpikeSource*] применил инструмент для охвата PHP, и открыл его исходный код; его можно найти на *SourceForge*. Мы списались с Энди Гутманом [Andi Gutmans] и Зеевом Сураски [Zeev Suraski], им это понравилось.

Теперь, когда этот инструмент создан, мы уже начали работу по системному применению с нашими партнерами по тестированию. Есть несколько фирм, с которыми мы работаем. Они протестировали *phpMyAdmin* и *phpPgAdmin*, это инструменты администрирования баз данных на PHP, которые раньше никогда не тестировались. Для проверки мы сгенерировали 1900 тестов, и результаты вернулись в проект. И этот инструмент *Spike PHPCoverage*, или написанный нами инструмент *asset manager*... все, что мы предлагаем, будет, как говорит Ким, с открытым исходным кодом.

LXF: Ваш совет директоров похож на издание Кто есть Кто: Брайан Белендорф, Митчелл Бейкер [Mitchell Baker], Тим О'Рейли... А что каждый из них дает?

МП: Они не только известны и уважаемы в отрасли, они активно помогают нам формировать нашу компанию. Потому что их тоже волнует вопрос возможностей, и они, как и мы, хорошо понимают, что нет Руководства или Сборника правил, которым можно было бы следовать. Это совет не по принципу «давайте пригласим самую верхушку», и не список людей, которые хорошо смотрятся на web-сайте. Каждый из них действительно переживает за то, что мы пытаемся создать.

МП: Ким была очень практична. Консультационный совет был задуман ею; она сразу же создала его, как только стала руководителем, и это очень способствовало нашему успеху. Мы им говорили одно: «Мы хотим, чтобы вы использова-

ли *SpikeSource*, чтобы осуществить вашу страсть ускорить продвижение открытого кода, любым способом, каким пожелаете». Ларри [Розен, Larry Rosen] мог способствовать в лицензировании, Стив Вебер [Steve Weber] — с социологической точки зрения. Наша цель — помочь им реализовать свою страсть в нашей фирме.

LXF: Я думаю, люди видят венчурных капиталистов, и большие имена, проходящие в связи с фирмой, чей адрес пишется «точка-ком», и слова типа «продукт», и думают: «Хмм, опять это слово, небось начался новый бум дот-комов». Что бы вы сказали для их успокоения?

МП: Я бы заподозрила то же самое, не будь реальной проблемы, для решения которой у нас имеется инструментарий. Иными словами, очень многие «дот-комовцы» [от англ. Dot-com — .com, — прим. пер.] в конце девяностых занимались созданием продуктов, которые на самом деле не решали особых проблем. В нашем случае Муруган и Рей определили проблему и поняли, что для ее решения нужен новый подход, так что с точки зрения технологии и того, как мы работаем с сообществом и строим фирму, по-настоящему основанную на архитектуре вовлечения... именно этим мы намерены заниматься.

МП: А в доказательство приведу такой вот простой пример. *Sleepycat*, которые выпускают *Berkeley DB* обратились к нам и дали нам 9 000 тестов. И мы тратим нашу электроэнергию и пропускную способность нашей сети для подтверждения работы на нескольких платформах, чтобы гарантировать взаимодействие. Вот где основной момент. Наше оправдание и наш приговор — помогать людям в этом процессе — в отличие от доткомовских компаний, наша модель — поставка программ в качестве сервиса [так, чтобы] люди могли загрузить это интегрированное ПО.

Раньше людям, которые ничего не понимали в этом, требовалось недели три на интеграцию. А тем, кто уже знаком и имеет определенный опыт, понадобится от восьми до 24 часов на интеграцию основного ПО. А в нашем случае нужно всего лишь 15 минут. Все уже готово к работе, и они могут видеть то, с чем им придется работать.

МП: Если мы можем сэкономить фирме деньги, и предоставить им все преимущес-

тва и гибкость открытого кода без всякого риска; и если мы можем работать с сообществом и содействовать проектам вроде *JetSpeed*, *Struts* и *Postgres*, и всем прочим, в получении более широкой сертификации на многочисленных платформах; публиковать тесты, сделав их достоянием всего сообщества, чтобы все могли извлечь из них пользу, то для меня именно это и называется успешной работой. Мы будем решать настоящие проблемы с точки зрения бизнеса, и, надеюсь, поможем сообществу в создании более богатой среды для всех.

LXF: Итак, вот что вы думаете в идеале: кто-то приобретает Red Hat Enterprise Linux, потом приходит в *SpikeSource* — и ему дают заверченный вариант. Вы говорите, что не заинтересованы в продвижении исключительно открытого ПО? Я заметил, что почти половина ваших машин работает под Windows. Говорите ли вы: «*Apache*, *MySQL*, *PHP*, *Windows*?» Или просеиваете ПО в поиске решения с открытым кодом?

МП: Мы действительно независимы от поставщиков — если наш клиент скажет, что хочет использовать, скажем, *SugarCRM* на *Windows*, мы не будем против. Если он захочет интегрировать *Apache* и *JBoss* с *Oracle*, мы тоже не будем против. Мы сертифицируем любое сочетание компонентов, если оно популярно и удобно для предприятия и в мире открытого кода.

МП: Но в то же время в идеале наши инженеры работают под Linux и используют *OpenOffice.org*. Мы работаем на *Samba*, *Postfix* и сервере электронной почты *Courier*, и это экономит наши средства. Потому что в конце концов речь идет о гибкости по отношению к клиенту, но и об экономии денег тоже.

LXF: А как вам видится будущее продуктов Open Source? Кроме, конечно, того, что *SpikeSource* станет огромной фирмой!

МП: А что мы имеем в виду под продуктом?

LXF: Ну, точнее, как вам видится превращение открытого ПО в продукт в будущем?

МП: С одной стороны, мы видим бесконечное изобилие компонентов с открытым кодом, которые и дальше будут генерироваться сообществом и приниматься предприятиями: это процесс долгосрочный, и он никуда не денется. А это означает, что и сложности будут опять возникать, и проблемы по поводу интеграции и взаимодействия потребуют нового подхода. Это — одна сторона, которая нам видится.

А еще нам видится, что открытый код будет повсюду. Взгляните на новое поколение приложений с открытым кодом, например, *Sugar*, *JasperReports*, *Alfresco*... почти каждый день появляется что-то новенькое. На мой взгляд, любое мыслимое приложение будет так или иначе представлено в виде открытого ПО. Так что будет выбор. Нам представляется, что открытый код проникнет не только в крупные организации: от этого выиграют и малые и средние предприятия, я полагаю, это будет способствовать развитию экономики, мы увидим, как открытое ПО встраивается в новые устройства, оборудование... в моем представлении Open Source будет везде, где нужны будут программы. **LXF**

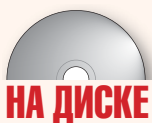


«Это похоже на вечеринку в складчину.»
Читайте полный вариант интервью с Ким и Муруганом на www.linuxformat.co.uk/mag/spikesource.html.

ubuntu

НАВСЕГДА

Ubuntu – это не дистрибутив, а образ жизни. Откуда он пришел? Как можно взять от него все? И каковы намерения его супер-богатого основателя? Расследование ведут **Грэм Моррисон и Майк Сондерс.**



НА ДИСКЕ

НА ЭТИХ СТРАНИЦАХ:

Руководство по установке Ubuntu 6.06. с. 32	с. 32
Обзор рабочего стола.....с. 33	с. 33
Easy и Enterprise.....с. 34	с. 34
Лучшие элементы Ubuntu.....с. 35	с. 35
Интервью с Марком Шаттлвортом.....с. 36	с. 36
Что будет в Edgy Eft.....с. 37	с. 37

И увидел Бог, что есть Debian, и что это хорошо. Что ж, это хорошо, если вы можете ждать два или три года между релизами и не

бояться текстового инсталлятора. Зато у Debian есть заскоружное сообщество, которое было, да и сейчас остается, полностью преданным свободе ОС – куда преданнее, чем многие другие сообщества вокруг Linux-дистрибутивов.

Затем, в конце 90-х, активный разработчик Debian из Южной Африки, Марк Шаттлворт (Mark Shuttleworth), ухватил удачу за хвост на пике «дот-комов». Он собрался потратить часть своего богатства на укрепление роли Open Source на глобальной сцене ПО, и решил, что с Debian он этого сделать не сможет. Шаттлворт задумал создать дистрибутив Linux с ориентацией на предоставление пользователям настольных систем лучших решений, с регулярными выпусками, без бесконечных политических дебатов и огромных интервалов между новыми версиями.

Ubuntu родился из желания Шаттлворта создать Linux для каждого. Он использует Debian как основу и наследует множество его особенностей, например, непредвзятый

подход к проприетарным программам (через использование неподдерживаемых репозиториями) и колоссальную доступность пакетов.

Шаттлворт называет себя «самопровозглашенным пожизненным диктатором» и спонсирует разработку Ubuntu через Canonical Ltd. Он даже вложил 10 млн. долларов из собственных средств в Фонд Ubuntu, гарантирующий продолжение проекта даже в случае, если с Шаттлвортом (или с Canonical) случится нечто скверное. Результат – сильное, организованное сообщество, контролирующее направление развития дистрибутива и сосредоточенное на конечном пользователе.

Что такое Ubuntu

С его положением на вершине хит-парада DistroWatch.com и растущим с каждым новым релизом числом скачиваний, Ubuntu является феноменом: новый взгляд на Linux, фокусирующийся на конечном пользователе, и сообщество, всегда готовое помочь. Заинтересованность Шаттлворта в распространении свободного ПО в развивающихся странах привело к сильному акценту на языковую поддержку, типа проекта Canonical Rosetta (<https://launchpad.net/rosetta>), призванного упростить пользователям перевод программ на их родной язык.

Никто не остался за бортом мира Ubuntu: дистрибутив свободен для загрузки, и вы можете бесплатно получить CD-диски. К разработке проекта может присоединиться каждый, при наличии желания и умения.

Команда Ubuntu не желает ограничивать доступность свободного ПО. Независимо от того, сидите ли вы посреди Лондона или ждете, пока диски Ubuntu придут по почте в удаленную ангольскую деревушку, вы всегда получите самый лучший Linux.

Савубона, Ubuntu!

[По-зулусски это значит «Племя приветствует тебя, Ubuntu!» – прим. пер.]. Просто не верится, что два года назад Ubuntu был крошечным «ответвлением» от проекта Debian, который даже не дошел до релиза. А сегодня это самый популярный дистрибутив на DistroWatch.com, имеющий миллионы пользователей, шумное сообщество и коммерческую поддержку. Как он достиг этого так быстро?

Ubuntu появился на сцене открытого ПО в среду, 20 октября 2004 года. Номер версии представляет год и месяц релиза, отсюда и Ubuntu 4.10.

МЕНЯЮЩЕЕСЯ ЛИЦО UBUNTU

Ubuntu 4.10 «Warty Warthog» [Бдительный Бородавочник]	20 октября 2004 г.	вердикт LXF60: 8 из 10
Ubuntu 5.04 «Hoary Hedgehog» [Седой Дикобраз]	8 апреля 2005 г.	вердикт LXF68: 8 из 10
Ubuntu 5.10 «Breezy Badger» [Бойкий Барсук]	13 октября 2005 г.	вердикт LXF74: 9 из 10
Ubuntu 6.06 «Dapper Drake» [Шустрый Селезень]	1 июня 2006 г.	вердикт LXF81: 9 из 10



От бородавочника (выше) к селезню (справа).

В традиции Debian давать своим релизам кодовые имена из *Истории игрушек* (см. LXF70), а Шаттлворт любит придумывать имена разработкам, отражающие ту черту характера, которую он хочет видеть в каждом релизе. Первый был назван Warty Warthog [«Бдительный бородавочник» – африканское животное, см. <http://ru.wikipedia.org/wiki/Phacochoerus> – прим. пер.]. В Ubuntu

wiki Warty описывался как «функциональный, но не изящный», зато он заложил прочный фундамент для всех последующих релизов Ubuntu. Он был основан на рабочем окружении Gnome 2.8, выделялся золотисто-коричневым одеянием, ставшим синонимом Ubuntu, требовал минимальных усилий по настройке и имел потрясающую совместимость с оборудованием.

А главное, Warty был украшен голыми людьми! Гениальный рекламный ход: ассортимент фоновых изображений Ubuntu включал со вкусом сделанные «фотошопные» рисунки людей без одежды (мы же не могли сказать «гимповые», верно?). Изображения не перешли в следующий релиз, но интерес, который они возбудили в Интернете, безусловно, остался. Мы убрали фоновые изображения из нашего обзора в LXF60, чтобы никого не шокировать, и дали Ubuntu 8 из 10 баллов в декабре 2004-го с вердиктом: «Ubuntu предоставляет надежное и мощное решение, поистине не имеющее изъянов». Люди скачивали Ubuntu тысячами.

Одним из основных мотивов отделения от Debian было желание выпускать два релиза в год, чтобы успевать включать все новейшие программы. Под руководством Шаттлворта Ubuntu сумел развиваться быстрее, чем Debian, подверженный метаниям курса. Хороший источник финансирования тоже способствовал успеху.

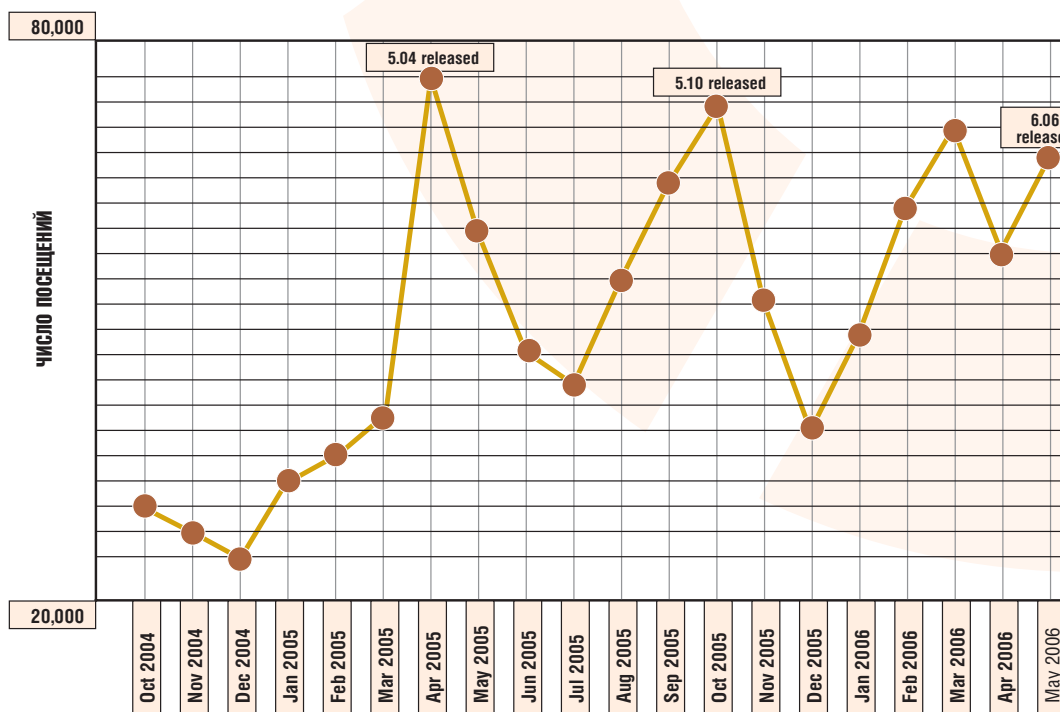
Последователь Warty был выпущен спустя шесть с половиной месяцев. Версия 5.04, Hoary Hedgehog («Седой дикобраз»), была другим зверем, отличающимся не только управлением пакетами, стилями и темами зрелой настольной системы, но

также отношением к своим разработчикам. Наиболее важным дополнением была совершенно новая версия дистрибутива, предоставляющая альтернативный рабочий стол с установочного диска. Kubuntu (на языке бемба, принятом в Замбии, это означает «к человечеству»), использует KDE вместо Gnome, и был выпущен параллельно с Ubuntu 5.04. Шаттлворт и компания разослали убедительное сообщение, что это не раскол сообщества Ubuntu – кому нравится KDE, может поставить его с официального благословения.

На nive просвещения

Открытое отношение сохраняется, и сообщество не боится создавать ответвление от проекта, если чувствует, что на него есть спрос. Так произошло со следующими двумя релизами, и Ubuntu проникал в области, где требовались преимущества стабильной платформы. Breezy Badger («Бойкий Барсук»), выпущенный в октябре прошлого года, сопровождался еще одним официально одобренным дистрибутивом, под названием Edubuntu. Он разработан для образовательных учреждений и, наряду с лучшими образовательными программами, предоставляемыми разработчиками

«СВОБОДНОЕ ПО ПОЗВОЛЯЕТ ЛЮБОМУ УМНОМУ ПАРНЮ В МИРЕ ВСТАТЬ НА ПЛЕЧИ ПРЕДЫДУЩИХ УМНЫХ ПАРНЕЙ И СПЛЯСАТЬ СВОЙ МАЛЕНЬКИЙ ТАНЕЦ ИННОВАЦИИ.» МАРК ШАТТЛВОРТ



Число визитов в день, полученных Ubuntu на www.DistroWatch.com с момента его появления.

Open Source, включает тонкий клиент. Идея заключается в том, что школы с ограниченным финансированием могут объединять в сеть старые ПК при минимуме расходов.

Вот мы и подошли к последнему релизу, Dapper Drake («Шустрый Селезень»), где к Gnome, KDE и образовательному дистрибутиву присоединился Xubuntu, который использует рабочий стол Xfce и является превосходным способом получить один из лучших Linux-дистрибутивов для старых компьютеров. Dapper Drake – знаковый релиз. Он считается достаточно стабильным, чтобы гарантировать долгосрочную поддержку, и спроектирован на базе опробованных и протестированных технологий Linux. Кроме того, он выпущен как раз тогда, когда наконец-то появился стабильный офисный пакет (*OpenOffice.org*), а также почтовый клиент и web-браузер, которые не отпугивают пользователей, лучше знакомых с другими операционными системами. Ubuntu повзрослел.



«Шесть шагов к Darpper Drake»

Ubuntu 6.06 в этом месяце есть на вашем DVD. Это «живой» (Live) дистрибутив, и вы можете опробовать его перед тем, как «прописать» на своем жестком диске. Для инсталляции, описываемой ниже, вам понадобится как минимум 5 ГБ места на винчестере и 256 МБ ОЗУ (а лучше 512 МБ). На соседней странице – несколько моментов, на которые стоит обратить внимание в вашей новой системе Ubuntu.



Вставьте наш DVD в привод и перезагрузите свой ПК (убедитесь, что у вас настроена загрузка с DVD, а не с жесткого диска). Нажмите **Enter**, чтобы загрузить дистрибутив; если возникнут проблемы с графикой, перезагрузите ПК и выберите Safe Graphics Mode (нажмите стрелку вниз, а затем – **Enter**).



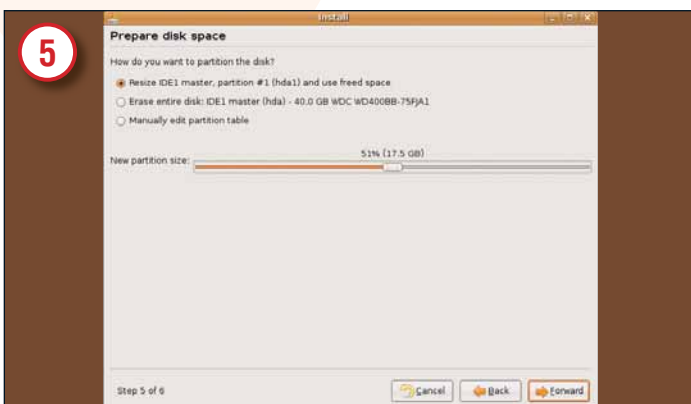
Ubuntu загрузит с DVD рабочий стол Gnome. Вы можете приступить к запуску приложений, но не забывайте, что это «живой» дистрибутив, так что любые изменения будут потеряны при перезагрузке! Запустите инсталлятор, двойным щелчком по иконке Install в левом верхнем углу. Теперь вы готовы к копированию Ubuntu на ваш жесткий диск.



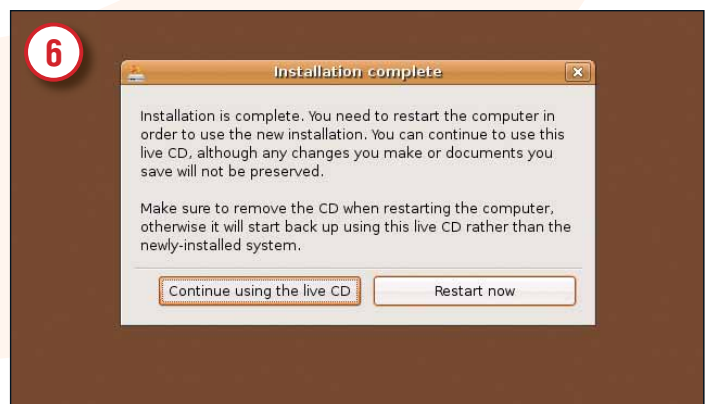
Сначала выберите язык, затем щелкните «Далее» – и попадете на этот экран выбора часового пояса. Меняя масштаб карты мира, подберитесь к ближайшему к вам городу и убедитесь, что часы установлены правильно. После этого снова щелкните «Далее», подтвердив правильность раскладки клавиатуры. Опять жмите «Далее».



Вам нужно имя пользователя и пароль, чтобы обезопасить Ubuntu. Введите сверху ваше полное имя, затем короткий логин и пароль (не забудьте его!). Вы можете указать имя компьютера (хоста), если знакомы с настройкой сети; иначе примите предложенное по умолчанию и щелкните «Далее».



Пора определиться с местом на диске для Linux. Если у вас есть пустой диск, или диск с большой (более 5 ГБ) свободной областью, вы можете позволить Ubuntu автоматически создать разделы Linux. Вы также можете изменить размеры существующих разделов. Если здесь вы зашли в тупик, посмотрите наше руководство к диску (стр. 113).



Щелкните «Далее», чтобы подтвердить настройки, и Ubuntu начнет копироваться. Появится это окно. (Зайдите в меню Система > Выход [System > Log Out], если передумали). Перезагрузите компьютер, вынув DVD перед стартом, и Ubuntu запустится с жесткого диска. Войдите под заданным на 4-м шаге именем пользователя, и вы готовы к работе!

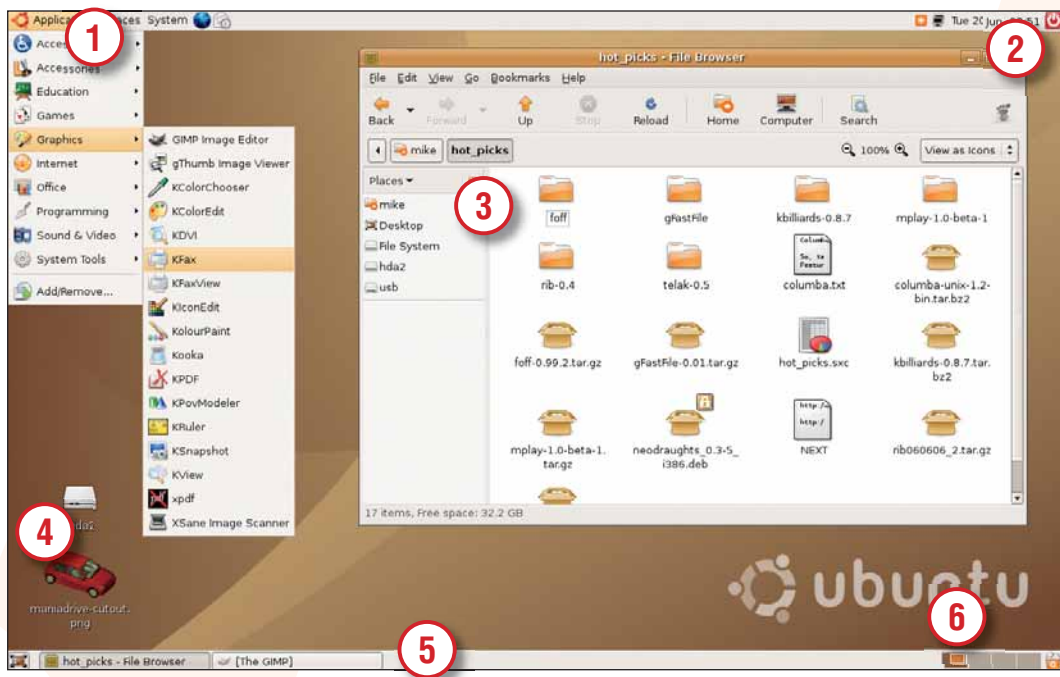
Коричневый — это красиво

Рабочий стол по умолчанию в Ubuntu 6.06 — Gnome 2.14, с характерной яркой оранжево-коричневой темой. Расположение элементов во многом похоже на типовую установку Gnome, или, если хотите, на Gnome в других дистрибутивах, например, Fedora Core, и в нем очень легко ориентироваться. Если это ваш первый Linux, или вы привыкли к дистрибутиву с рабочим столом KDE, обратитесь к нижеприведенному краткому обзору главного. (Дополнительная помощь — в меню Система > Справка).

1 МЕНЮ
Три пункта меню слева сверху позволяют вам входить в Applications (любые установленные программы), Places (расположение файловых систем) и System (системные настройки). Большинство программ, добавленных с помощью менеджера пакетов Synaptic, будет отображено здесь.

2 СИСТЕМНЫЙ ЛОТК
В верхнем правом углу размещены часы, настраиваемые правой кнопкой мыши, наряду с иконками системного лотка (например, состояние сети). Красная кнопка завершает сеанс работы — щелчок по ней выводит диалоговое окно, предлагающее выбор: выключение, перезагрузка или возврат к окну входа, где можно выбрать другой рабочий стол, включая KDE (щелкните Параметры, затем Выбрать сеанс...).

3 NAUTILUS
Nautilus — это файловый менеджер Gnome. Обратите внимание на организацию предварительного просмотра, когда иконка



файла показывает часть его содержимого, как для текстовых файлов на рисунке.

4 ИКОНКИ
Чтобы проще было добираться до определенных файлов, вы можете перетаскивать их на рабочий стол. Опять-таки, для графических файлов используется стильная миниатюра.

5 ПАНЕЛЬ ЗАДАЧ
Внизу располагается панель задач, для переключения между запущенными программами. Можете щелкнуть правой клавишей по кнопкам панели, чтобы закрыть приложения, которые бывают и невидимыми.

Иконка внизу слева сворачивает все открытые окна, очищая вам рабочий стол.

6 ПЕРЕКЛЮЧАТЕЛЬ
Справа от панели задач помещен плейджер. Он предоставляет вам «виртуальные рабочие столы» — дополнительные места для ярлыков, позволяющие избежать нагромождения. Вы можете использовать одно «рабочее место» для интернет-приложений, другое — для офисных программ, и т.д.

Советы по настройке
Для настройки вашей установки Ubuntu, зайдите в меню «Система», пункт

«Администрирование». Здесь находится множество инструментов для изменения параметров системы, например, настроек сети или экрана. Большинство из них попросит ввести ваш пароль, чтобы убедиться, что вы имеете право параметров.

Также в меню «Система» есть подменю «Параметры», которое позволяет навести косметическую красоту на дистрибутив — шрифты, темы, звуки и так далее. Все вместе, эти утилиты дают широкие возможности управлять всем дистрибутивом, и пока вы не захотите учинить что-то особо продвинутое, вам не потребуется вторгаться на территорию командной строки.



КАК ДОБАВИТЬ БОЛЬШЕ ПРОГРАММ

У Ubuntu есть тысячи пакетов программ, готовых для установки через Интернет. Благодаря фундаменту Debian, почти каждая существующая открытая программа доступна, и, к счастью, совсем не сложна в установке! Лучший инструмент для добавления приложений — Synaptic, мощный менеджер пакетов: в нем есть полезные функции поиска, способные находить именно то, что вы хотите.

Еще до запуска Synaptic, вам понадобится добавить дополнительные репозитории программ Ubuntu. По умолчанию дистрибутив настроен на доступ к небольшому числу популярных и полностью поддерживаемых приложений; но есть гораздо больше программ, собранных сообществом Ubuntu, и к ним мы тоже хотим создать доступ. Откройте терминал (Приложения > Стандартные > Терминал) и введите:

```
sudo gedit /etc/apt/sources.list
```

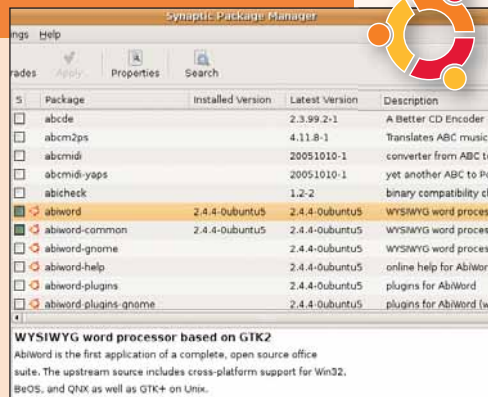
Затем введите ваш пароль, и откроется текстовый редактор. Здесь мы настроим инструменты для работы с пакетами, чтобы добраться до дополнительных пакетов — репозиториях Universe и Multiverse, которые гораздо богаче, чем подключенная по умолчанию секция Main. Добавьте в конец этого файла следующие строки:

```
deb http://gb.archive.ubuntu.com/ubuntu/
dapper universe
deb-src http://gb.archive.ubuntu.com/
ubuntu/ dapper universe
deb http://gb.archive.ubuntu.com/ubuntu/
dapper multiverse
deb-src http://gb.archive.ubuntu.com/
ubuntu/ dapper multiverse
```

[Чтобы использовать русские зеркала, замените gb на ru. Кроме того, можно не редактировать файл sources.list непосред-

ственно, а указать нужные репозитории в самом Synaptic — «Настройки > Репозитории» — прим. пер.]

Сохранив файл, выйдите, затем в терминале введите `sudo apt-get update`, чтобы система узнала о внесенных изменениях. При этом из Интернета будут загружены списки новых пакетов. Когда процесс завершится, щелкните Система > Администрирование > Программа управления пакетами Synaptic, и запустится менеджер пакетов. Здесь вы можете просмотреть доступные программы, используя Разделы слева или нажав кнопку Искать сверху. Когда вы найдете программу, которую хотите установить, щелкните правой кнопкой мышки по ее имени в списке и выберите Отметить для установки. Нажмите кнопку Применить на панели инструментов, и Synaptic скачает ее из сети. Если вы совсем



Synaptic: ваше окно в бескрайний мир свободного ПО.

запутались, нажмите [F1], чтобы получить онлайн-справку!



« Понять Ubuntu глубже

Получить лучшую поддержку файловых форматов с помощью Easy... а как насчет создания своей собственной интернет-империи со звездного корабля Enterprise Ubuntu?

EasyUbuntu

При всем великолепии Ubuntu людей подстерегает одна проблема: отсутствие в поставке коммерческих кодеков или программ. Если вы попытаетесь прослушать MP3 под Ubuntu, или, например, посмотреть DVD, то очень быстро упреетесь в стену — они просто не работают «из коробки».

Дело в том, что Ubuntu избегает «закранных» файловых форматов и приложений, так что мы, конечные пользователи, получаем полностью свободное (в смысле свободы распространения) ПО, без мороки с лицензиями и патентами. Однако свобода имеет обратную сторону: некоторые функции, например, воспроизведение MP3, закрыты патентами, не включены в установку по умолчанию.

Если воспроизведение MP3 не защищено патентами в вашей стране, хитроумное сообщество Ubuntu поспособствует его включению без особых сложностей, через EasyUbuntu (<http://easyubuntu.freemcontrib.org>). Эта небольшая программа выполняет всю тяжелую работу по восполнению недо-

стающей функциональности дистрибутива, автоматически скачивая и устанавливая программы из Интернета, так что вам не понадобится тратить годы на поиск в Synaptic или посещение сайтов сторонних разработчиков.

Просто введите следующую команду в терминале (пройдите в «Приложения > Стандартные > Терминал»):

```
wget http://robotgeek.org/eu/easyubuntu-3.01.tar.gz
tar -zxf easyubuntu-3.01.tar.gz
cd easyubuntu
sudo python easyubuntu.py
```

Учтите, что EasyUbuntu регулярно обновляется, и если у вас эти команды не работают, посетите сайт проекта и прочтите новые инструкции.

Вы увидите диалоговое окно со вкладками (см. рисунок справа вверху), где «птичками» надо отметить желаемые функции. Удобно, что EasyUbuntu поможет также установить дополнительные свободные программы, которые обычно с трудом



Воспроизведение MP3, поддержка DVD и Java — одним щелчком мыши.

поддаются настройке — например, проигрыватель MIDI. Закончив, щелкните ОК, и EasyUbuntu все сделает сам. [Готовьтесь к достаточно большим объемам трафика — только на кодеки плюс MIDI потребуются более 50 МБ. Впрочем, перед началом

загрузки информация по объему закачки будет выведена, и у вас запросят подтверждение, — прим. пер.] По завершении работы выйдите из системы и снова зайдите, после чего сможете испробовать все раздобытые функции!

Enterprise Ubuntu

Марк Шаттлворт удивил всех, когда при подготовке к релизу Dapper объявил, что хочет конкурировать с Red Hat и Novell на корпоративном рынке. И первые плоды этой стратегии стали видны, когда Dapper Drake вышел с долгосрочной поддержкой. Другими словами, Canonical будет в течение пяти лет предоставлять

обновления безопасности, что очень важно для каждого ИТ-менеджера, желающего установить Ubuntu в преуспевающей фирме — ту же стратегию использовали Novell и Red Hat для корпоративных версий своих дистрибутивов.

Но Ubuntu козырнул тузом из рукава: заключил стратегический альянс с Sun

Microsystems, самым олицетворением корпорации. Sun хочет использовать серверную редакцию Ubuntu на своем оборудовании SPARC и даже предустанавливать ее для своих покупателей. Итог — серверная версия Ubuntu 6.06, выполненная специально для Sun SPARC64.

Серверная версия Ubuntu не изобилует пакетами, подобно типичной установке Linux, но она невероятно полезна для быстрой установки программ, которые могут потребоваться среднему серверу. Она пригодится и домашним

пользователям, хотя SPARC-версия — это, пожалуй, чересчур. Серверная версия может использоваться как простой файловый сервер, или даже как основа для домашнего медиа- или интернет-сервера. И, если вы решите, что вам нужен полноценный рабочий стол Linux, установка — это лишь вопрос репозитория.

Вы можете найти различные установочные CD с Dapper, включая серверные версии, на <http://releases.ubuntu.com/dapper>.



IMAGE COURTESY: SUN MICROSYSTEMS INC

Кое-кто в сообществе удивился сделке между Sun и Марком Шаттлвортом (вы видите Марка на майской конференции JavaOne).

ЧТО ЛЮБИТ ДЭН

Что больше всего нравится в проекте хакера Ubuntu Дэну Холбаху (Dan Holbach)?



«Больше всего мне нравится в Ubuntu полная прозрачность разработки и старательное сотрудничество разработчиков.»

Если вы хотите знать, что происходит в процессе разработки, или принять участие в проектировании или принятии решений,

вы можете Просто Сделать Это! Просто чудесно видеть, как люди что ни день присоединяются к командам, вносят свои идеи и добавляют захватывающие и ценные аспекты в Ubuntu во всевозможные его разделы.»

Великолепная четверка

Ubuntu – цельный проект, но именно эти четыре функции делают его дистрибутивом Linux, равно достойным и начинающих пользователей, и экспертов.



Цветовая гамма Ubuntu распространяется и на экраны входа и выхода из системы.

Превосходный вид

Марк Шаттлворт считает, что приглядный рабочий стол – это функция, и мы с ним согласны. Ubuntu узнается с ходу, потому что художники рискнули сменить цветовую гамму, и коричневая цветовая схема, явно просветлевшая в Dapper Drake, стала четким символом дистрибутива.

Совместимость с Debian

Благодаря использованию в Ubuntu формата пакетов Debian и насыщенности хорошо поддерживаемых онлайн-репозиториями, для Ubuntu доступно больше программ, чем для любого другого дистрибутива (ну, то есть, всех, что мы смогли припомнить). Кроме того, Ubuntu настраивается так, что добавлять сторонние и неподдерживаемые репозитории сравнительно несложно. Это решение бросает вызов Fedora и SUSE, которые умышленно усложняют установку всего, связанного с патентами или авторским правом. Все, что вам нужно в Ubuntu – правильная библиотека и законные права использовать ее.

Synaptic также заслуживает внимания, ведь он делает получение всех программ практически безболезненным. Через его графический интерфейс вы можете даже добавлять и делать доступными свои собственные репозитории.

Launchpad

Launchpad подобен Олимпийской Деревне для атлетов Ubuntu. Это центральный

репозиторий для разработчиков Ubuntu, состоящий из множества разнообразных инструментов, аккуратно завернутых в web-интерфейс.

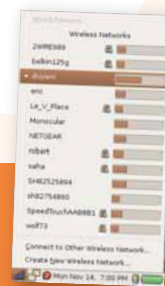
Большую часть этих приложений написал сам мистер Шаттлворт. При взгляде на главную страницу, где список новых приложений стиснут списками лучших разработчиков слева и крупнейших жертвователей справа, может показаться, что их не так уж и много. Но за главной страницей прячутся залежи функциональности для тех, у кого есть учетная запись (и каж-



Launchpad – центр вселенной сообщества разработчиков Ubuntu.

ЧТО ЛЮБИТ МАРК

Что больше всего нравится в проекте основателю Марку Шаттлворту?



«*NetworkManager* – мы не включили его по умолчанию, потому что сейчас он, похоже, работает примерно в 70% случаев, в зависимости от вашей Wi-Fi-карты и так далее. Но просто выполните `sudo apt-get install network-manager`, затем поместите апплет *NetworkManager* в

вашу сессию. Затем вы выберете вашу сеть Wi-Fi из списка – вы увидите их все вместе с уровнями сигнала. *NetworkManager* поддерживает WEP, WPA и кучу других вещей. Если он работает у вас, это здорово.»

дый может получить ее бесплатно). Для обычных пользователей самой полезной функцией будет инструмент оповещения об ошибках *Malone* (помните Buggy Malone?), который спроектирован, чтобы каждый мог «поднять шум», обнаружив в Ubuntu проблему. Сейчас накопилось почти 50 тысяч сообщений об ошибках, и *Malone* позволяет группам разработчиков брать их исправление на себя либо поручать другим членам команды. Ошибки могут варьироваться от банального «#50322: Пожалуйста, включите игры в Kubuntu» до фатального «#345086: Ядро 2.6.15 падает при загрузке на машине PPC».

Если сообщения об ошибках – не ваш конек, есть другой способ помочь – это перевод текста вашего любимого приложения на другой язык. Естественно, вы должны в совершенстве владеть выбран-

ными вами языком, зато это – замечательный способ внести свой вклад. Если программа зарегистрирована в проекте *Rosetta*, вы можете набрать ваш перевод для каждой строки непосредственно в приложении. Вам даже не потребуется запускать приложение, хотя это поможет лучше понимать, о чем вы пишете. Вы также можете следить за продвижением перевода на ваш язык каждого зарегистрированного приложения.

Долгосрочная поддержка

Область, где Ubuntu затмевает Debian – это долгосрочная поддержка (Long-Time Support). С LTS-релизом Dapper Drake вы можете рассчитывать на пять лет обновлений безопасности. Это почти вечность для Linux, и это на два года больше, чем предлагает Debian – налицо смещение приоритетов Ubuntu от чисто «настольных» амбиций к признанию важности серверной версии. Конечно, это отнюдь не повредит и школьным программам Ubuntu, и правительственным инициативам, где пятилетняя поддержка горячо приветствуется.

В отличие от предыдущих релизов, Dapper Drake взял на себя обязательство по обратному портированию важных приложений, чтобы они работали в этом проекте. Сюда не входят основные новые функции, подобные получаемым с новым релизом Gnome или KDE, но могут включаться небольшие обновления, исправляющие не одни проблемы безопасности. CUPS – прекрасный пример, ибо Dapper поставляется с самой современной версией 2.0. Если в следующих версиях может решиться какая-то из проблем версии 2.0, портированная версия станет доступной для пользователей Ubuntu.

«Человек со звезды»



Марк Шаттлворт разбогател, продав свою фирму Thawte, занимающуюся интернет-безопасностью, фирме VeriSign в 1999-м году. Затем он потратил часть своего кошелька на экскурсию в космос. Его последующее отделение от Debian, высокая популярность и, разумеется, миллионы нервировали некоторых бескомпромиссных членов сообщества FOSS, и даже вызывали недоверие к его мотивам. Но **Майк** и **Грэм** встретились с ним, чтобы поговорить о курсе Ubuntu, и нашли тихого, скромного человека, чья страсть — движение Open Source.

LinuxFormat: Насколько важен внешний вид Ubuntu?

Марк Шаттлворт: «Прелесть» — это функция, особенно когда мы пытаемся донести открытое ПО до более широкого сообщества людей. Мы должны сделать так, чтобы людям захотелось показать его своим друзьям, своим родителям. Дедушек и бабушек не заинтересуешь супернадёжным ядром — они заранее убеждены в полной надёжности (или в полной ненадёжности) компьютеров, и этим их не завлечь.

LXF: Выйдет ли Edgy Eft в октябре, или задержка с Dapper сдвинет и его?

МШ: Дата релиза все еще обсуждается. Мы сдвинули выпуск Dapper, потому что это релиз с долгосрочной поддержкой, которая требует высочайшего качества. Сделаем ли мы Edge полным шестимесячным релизом, с постоянным отставанием от Gnome на два месяца, или нагоним один месяц, а то и целых два? Похоже, есть сильная поддержка идеи «догнать и перегнать», и тогда Edgy станет «остроактуальным» релизом.

LXF: Где вы видите Ubuntu спустя десять лет?

МШ: Я надеюсь, что Ubuntu будет стандартной опорой, факелоносцем свободного ПО для некомпьютерных специалистов. Но я вовсе не собираюсь подрывать идею Ubuntu как стандартной опоры свободного ПО для разработчиков и профессионалов. Я не считаю Ubuntu версией Linux «для

тупых» — но наряду с ультрасовременностью и ориентированностью на разработчиков, надеюсь, Ubuntu будет продвигать свободное ПО в новые области, куда оно пока не проникло.

LXF: В журнале за этот месяц Ладислав Боднар пишет об обратной связи, обнаруженной в сети на последних нескольких выпусках дистрибутивов. Он отметил у Dapper больше отчетов о проблемах, чем у последних версий Fedora или SUSE...

МШ: Это интересно. Мы подметили, что размер нашего сообщества удваивается от релиза к релизу — и мы получаем отчеты о таких ошибках, к которым не привыкли, потому что люди теперь используют Ubuntu непривычными способами. Так что границы привычного постоянно отодвигаются. Мы обязаны расширять наше сообщество разработчиков, чтобы управиться с этим.

Если вы получаете сообщения об ошибках от вдвое большего числа источ-

ников, о примерно удвоенном числе различных вещей, нужно иметь достаточно разработчиков для «переключения контекста» на эти различные области. Этот баланс становится чрезвычайно важен. Я не беспокоюсь о качестве Dapper: я думаю, это лучший релиз, который мы когда-либо выпускали. Но я понимаю, что он не идеален, и могу допустить, что мы обнаружим крайности, с которыми мы никогда не сталкивались в предыдущих релизах.

LXF: Что движет людьми, стоящими за разработкой Ubuntu?

МШ: Я думаю, у нас у всех в определенной степени эгоистичные мотивы — мы хотим сделать нечто, чем сможем поделиться, чем сможем гордиться, и на чем будут наши имена. Эти парни построили то, что они подарят своим друзьям, о чем будут рассказывать, что станет частью того, что они делают. Это потрясающе — быть причастным к такому.

Я видел в сообществе даже определенное перегорание: ребята, рванувшие в

разработку, забросив многое другое в своей жизни. Это опасно: если это не работа, это и не должно быть работой. Мне нравится видеть парней, которые любят быть причастны к чему-то и вносят огромный вклад, но сохраняют здравомыслие и сами задают себе темп. Многие из команд начинают дозревать до понимания этого.

Часто случается и другое: у нас есть парни, пробывшие с нами достаточно долго, чтобы стать настоящими лидерами различных команд. И вы можете ясно видеть различие между командой сообщества, где этот настоящий лидер появился, и другими, где нет. Мы учимся распознавать и поощрять таких лидеров.

LXF: Есть ли в Dapper Drake какие-нибудь сюрпризы?

МШ: Есть несколько прикольных кусочков — видео от Нельсона Манделы, например! [в нем Мандела рассказывает о значении слова «ubuntu»].

К тому же, вы знаете, некоторым из скринсейверов требуются фотографии, и

«Я НИКОГДА НЕ СЧИТАЛ UBUNTU ВЕРСИЕЙ LINUX «ДЛЯ ТУПЫХ»»

На переднем краю

Для популярного дистрибутива большая редкость, когда направление развития определяется крутыми хакерами из команды разработчиков – но так и планируется в следующем релизе Ubuntu.

Разработчики Ubuntu по праву гордятся достижениями Dapper Drake. Они все-таки сделали то, что намеревались, затеяв дистрибутив Ubuntu: настольную систему Linux, простую в использовании, и, что важнее, масштабируемую. Будь то средний домашний ПК или офис, забытый машинами, Ubuntu станет превосходным решением.

Linux может наконец-то восприниматься серьезно, во многом благодаря обещанной долгосрочной поддержке Ubuntu: *на пять лет*. Подобные обязательства достигаются определенной ценой, и здесь это – стабильность. Выпуск Dapper Drake был отложен, чтобы убедиться, что никакие проблемы стабильности не приведут к банкротству. Это означает отказ от многих появившихся технологий Linux ради более обкатанных и протестированных программ.

Но стабильность – не единственная мотивация Ubuntu: в конце концов, это Linux для обычных людей. А большинство из нас хочет, как только просохнет картинка на нашей новой установке, получить новые функции – и разработчики Ubuntu, Марк Шаттлворт в

частности, не исключение. Поэтому следующий релиз Ubuntu будет иным. У него не будет долгосрочной поддержки, и, возможно, он рискнет стабильностью, но зато вы получите лучшее из кузницы Linux, где новые идеи и разработки выкует молодой Edgy Eft, в переводе – рискованный тритон.

Ubuntu не меняет направление: просто Dapper Drake идет под маркой стабильности, в то время как команда Ubuntu оторвется на парочку других релизов. Если вы бизнесмен, Dapper наиболее пригоден в качестве настольной системы; если вы домашний пользователь, Edgy даст вам лучшие разрабатываемые пакеты Linux в неподражаемом стиле Ubuntu. Вы, видимо, получите только 18 месяцев обновлений, но многие из нас все равно еще до этого перейдут на следующий хитовый дистрибутив.

Шаттлворт не указывал, что должно оказаться в Edgy, кроме элементарной совместимости для некоторых корпоративных партнеров Ubuntu. Мистер Шаттлворт дал сообществу волю использовать коллективную фантазию и предложил каждо-

му (включая вас) вносить идеи. Крайний срок для предложений наступил 12 июня, и вслед за этим технический совет Ubuntu свел их воедино и разложил по приоритетам, основываясь на соответствии целям Ubuntu, уместности в Edgy и предполагаемой степени готовности каждой включаемой функции на время выхода релиза.

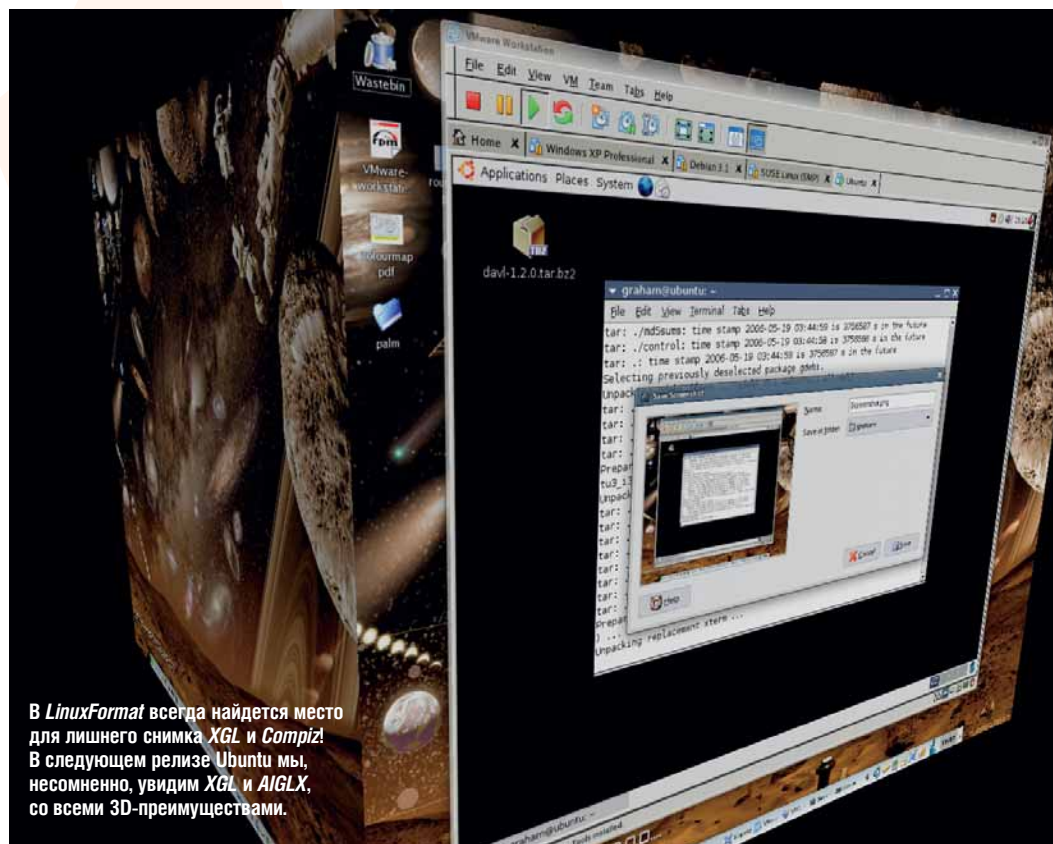
Ясно одно – новая версия Ubuntu будет выглядеть хорошо. Можно не сомневаться, что это будет рабочий стол с OpenGL-ускорением, использующий *Compiz*, и *XGL* или *AIGLX*. Вы можете также ожидать лучшей интеграции настольного поиска с помощью Beagle, и, вероятно, версию *Xep*, где каждый сможет вызвать виртуальную машину с рабочего стола – Ubuntu в Ubuntu. Да, и Шаттлворт мечтает также о мультимедиа-редакции, которая включит все проприетарные кодеки, на попытки установить которые многие из нас тратят уйму времени. Но каким бы мы его не получили, Edgy будет еще одним достойным восхищения релизом. Давайте просто надеяться, что в этот раз задержек не будет! **LXF**

Узнайте мнение Марка о двоичной совместимости и мультимедиа-версии Ubuntu на www.linuxformat.co.uk/mag/shuffeworth2.html

если их не ткнуть в фото, они возьмут тестовые изображения экрана? Мы выложили пару фотографий из тех, которые вам понравятся – изображения из космоса! А прочее? Придется уж вам покопаться в этом... [Смеется.] У нас есть прекрасная первоапрельская идея для людей, занимающихся процессом разработки Edgy+1... Жду не дожусь Первого апреля!

LXF: Позвольте, угадаю: все посинеет? Оконным менеджером по умолчанию будет *TWM*?

МШ: [Смеется] *TWM* с *Compiz*!



В *LinuxFormat* всегда найдется место для лишнего снимка *XGL* и *Compiz*! В следующем релизе Ubuntu мы, несомненно, увидим *XGL* и *AIGLX*, со всеми 3D-преимуществами.



aptitude Search [Full options](#)
 Search on: Package names only Descriptions
 Source package names

- About Debian
- News
- Getting Debian
- Support
- Development
- Site map
- Search

Package Search Results

You have searched for packages named *aptitude* in all suites, all sections, and all architectures.

Found 1 matching packages, displaying package 1.

Package 1

СНОВА АРТITUDE: РЕЖИМ КОМАНДНОЙ СТРОКИ

Одна из отличительных особенностей дистрибутивов семейства Debian – разнообразие средств управления пакетами. Однако в последнее время в качестве такового рекомендуется *aptitude* – надстройка над *apt*, работающая в текстовом режиме. Она предполагает два метода использования – интерактивный и командный. Первый был подробно описан Тихоном Тарнавским (LinuxFormat, #5(79), 2006). **Алексей Федорчук** остановится только на командном методе, затронутым в том материале лишь вкратце.

Командный метод использования *aptitude* будет непривычен тому, кто знаком с утилитами *apt-get* и *apt-cache*: конструкция ее директив предполагает наличие оператора и, для некоторых из последних, также аргумента – имени пакета или ключевого слова.

Особенности *aptitude* в сравнении с утилитами *apt* проще всего рассмотреть на примере конкретных командных директив.

Резонные люди обычно начинают работу с пакетами поиском нужного для установки. В случае с *aptitude* это делается так:

```
$ aptitude search keyword
```

ответом на что будет список всех пакетов, в названии или описании которых имеется указанное ключевое слово, с краткой характеристикой. Почти как в *apt*, но вывод *aptitude* содержит информацию о текущем состоянии пакета. Например:

```
$ aptitude search term
```

выдаст примерно такой список (рис. 1)
 Пакеты, маркированные литерой **i** (от installed), уже установлены в системе, а помеченные литерой **p** (от purge) – не установлены или удалены «вчистую» (как – будет говориться далее). Кроме того, в этой колонке могут присутствовать марки **c** (от clean), которой отмечены

пакеты удаленные, следы которых (в виде конфигурационных файлов), однако, сохранились, и **v** (от virtual), которой обозначаются так называемые виртуальные пакеты, представляющие собой просто списки пакетов реальных.

Для установленных пакетов возможна еще и дополнительная маркировка – литерой **A** (от automatic); таким образом помечаются пакеты, установленные автоматически в качестве зависимостей других пакетов.

Следующий этап – получение информации о тех пакетах, которые можно заподозрить в полезности. Этой цели служит оператор *show*, требующий аргумента в виде имени пакета, например:

```
$ aptitude show mltterm
```

выведет весьма подробные сведения о пакете *mlterm* (рис. 2):

- имя и версию;
- статус – установлен ли пакет и, если установлен, то как: собственноручно или автоматически, в качестве зависимости;
- приоритет пакета и раздел репозитория, к которому он приписан;
- имя и e-mail сопровождающего;
- размер пакета в распакованном виде;
- зависимости пакета, предложения по дополнительным компонентам и конфликтующие пакеты;
- назначение и функциональность.

В *aptitude*, в отличие от *apt*, в число зависимостей включаются не только обязательные (*depends*), но также и рекомендуемые (*recommends*) пакеты.

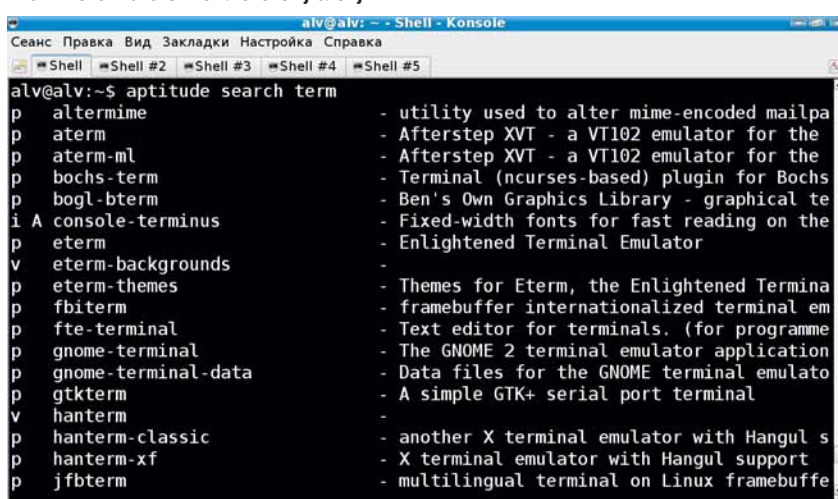
Очевидно, что все операторы команды *aptitude*, служащие получению информации о пакетах, могут выполняться от лица обычного пользователя. Следующие же операторы требуют уже привилегий администратора – в дистрибутивах семейства Ubuntu они получаются посредством команды *sudo*.

Установка выбранных пакетов осуществляется посредством оператора *install*, требующем в качестве аргумента имени пакета:

```
$ sudo aptitude install pkg_name
```

Оператор *install* команды *aptitude*, в отличие от одноименного из *apt-get*, устанавливает не только «строгие» зависимости пакета (собственно *depends*), но и часть «мягких» (*recommends*), то есть все, что перечислено в качестве зависимостей в выводе оператора *show*. На

Рис. 1. Поиск пакетов по ключевому слову.



усмотрение пользователя остается только установка «мягких» зависимостей из категории **suggest** (то есть «предложений» оператора **show**). Хотя, как мы увидим далее, такое положение вещей можно изменить.

Установка версий пакетов осуществляется в соответствии с локальным их кэшем, каковой время от времени (а также после подключения дополнительных репозиториях) нуждается в обновлении. Это осуществляется посредством оператора **update**, в аргументах не нуждающегося.

«Испорченный» по какой-либо причине (например, неаккуратным вмешательством в конфигурационные файлы) пакет можно «починить».

Команда

```
$ sudo aptitude reinstall pkg_name
```

вернет его в первоначальное состояние.

Установленные пакеты, оказавшиеся не нужными, могут быть удалены. Директива

```
$ sudo aptitude remove pkg_name
```

удалит указанный в качестве аргумента пакет с сохранением его конфигурационных файлов. Именно такая ситуация и маркируется литерой **c** в выводе команды **aptitude search**. Полная же очистка системы от всех следов пакета достигается оператором **purge**:

```
$ sudo aptitude purge pkg_name
```

В этом случае пакет в выводе команды **aptitude search** маркируется литерой **p**.

Важно, что оба оператора удаления — **remove**, **purge**, — деинсталлируют не только пакет, указанный в качестве аргумента, но и все те, что были установлены автоматически в качестве его зависимостей — разумеется, только в том случае, если в системе не осталось других программ, которые от них зависят. Уже одно это является веским аргументом в пользу предпочтения **aptitude** перед классическими инструментами **apt**.

aptitude позволяет выполнить и тотальное обновление системы — этой цели служат операторы **upgrade** и **dist-upgrade**. Как и в **apt-get**, первый обновит все установленные пакеты в том случае, если это не влечет за собой новых, противоречащих имеющимся, зависимостей. Оператор же **dist-upgrade** выполнит принудительное обновление системы. В обоих случаях удалению подвергнутся также автоматически установленные пакеты, от которых больше ничего не зависит.

Позволяет **aptitude** избавиться и от промежуточных продуктов собственной жизнедеятельности — скачанных из Сети deb-архивов; для этого предназначены операторы **clean** и **autoclean**.

И, наконец, еще пара операторов, не имеющих аналогов в инструментарии **apt**: **markauto** и **unmarkauto**. Первый помечает пакет или их группу как установленную автоматически в качестве зависимостей. Так, командой

```
$ sudo aptitude markauto '-slibs'
```

в качестве автоматически установленных будут помечены все пакеты с компонентом **libs** в имени — то есть практически все библиотеки, следствием чего явится автоматическое удаление неиспользуемых библиотек после деинсталляции последнего зависимого от них пакета.

Если же для некоторых библиотек это по каким-либо причинам нежелательно, их можно «размаркировать» командой

```
$ sudo aptitude unmarkauto pkg_name
```

переведя таким образом в категорию пакетов, установленных собственноручно, и, следовательно, могущих быть удаленными только явным образом.

Кроме операторов, командная директива **aptitude** предусматривает использование опций. Они весьма многочисленны, но не обязательны, и потому я остановлюсь только на самых, с моей точки зрения, интересных и полезных. Более подробные сведения об опциях можно получить посредством

```
$ man 8 aptitude
```

Ранее уже говорилось, что по умолчанию при использовании **aptitude** с любыми операторами инсталляции устанавливаемый пакет тянет за собой не только «жесткие», обязательные, зависимости, но и изрядную часть «мягких» — тех, которые сопровождающий пакета посчитал нужным включить в категорию **recommends**, что не всегда желательно.

Избежать принудительного выполнения «рекомендаций» можно с помощью опции **-R**, данной в командной директиве установки конкретного пакета. Если же игнорирование «рекомендаций» требуется всегда — это можно зафиксировать в конфигурационном файле. Как именно — будет сказано чуть ниже, пока же допустим, что мы уже изменили умолчательное поведение **aptitude** — теперь операторы инсталляции учитывают только «жесткие» (**depends**) зависимости.

Для некоторых пакетов все же желательно установить все рекомендуемые зависимости — например, в случае малознакомых пакетов, с которыми просто лень разбираться. В этом

```

alv@alv:~$ aptitude show mltterm
Пакет: mltterm
Новый: да
Состояние: не установлен
Версия: 2.9.2-6ubuntu1
Приоритет: необязательный
Раздел: universe/x11
Сопровождающий: Tomohiro KUBOTA <kubota@debian.org>
Размер в распакованном виде: 418k
Зависимости: libc6 (>= 2.3.4-1), libfontconfig1 (>= 2.3.0), libfreetype6 (>= 2.1.5-1), libfribidi0 (>= 0.10.7-1), libglib2.0-0 (>= 2.9.3), libgtk2.0-0 (>= 2.8.0), libx11-6, libxft2 (> 2.1.1), libxrender1, zlib1g (>= 1:1.2.1), mltterm-common (>= 2.9.1)
Рекомендуются: mltterm-tools
Предлагаются: unifont, xfonts-efont-unicode, ttf-kochi-gothic, ttf-baekmuk, ttf-arphic-bsmi00lp, ttf-arphic-gbsn00lp, ttf-freefont, tl-cyrillic, mltterm-im-uim, mltterm-im-iiimf, mltterm-im-m17nlib, mltterm-im-scim

```

Рис. 2. Получение информации о пакете.

случае можно прибегнуть к опции **-r** (**--with-recommends**), которая инвертирует действие опции **-R** — то есть заставит установить все рекомендуемые зависимости.

Должен предупредить: применение опции **-R** к установленной системе Ubuntu/Kubuntu требует осторожности. Базовая ее инсталляция осуществляется по принципу «плюс **recommends**». И применение к ней **aptitude -R** делает как бы «ненужными» многие пакеты. Одни из них — действительно (на мой взгляд) лишние. Однако в «черный список» могут попасть и нужные библиотеки. Так что перед тем, как нажать **Enter** в ответ на предложение

Хотите продолжить? [Y/n/?]

внимательно прочтите весь предшествующий ему вывод.

Тем не менее, вполне возможно, что по разрешении указанных противоречий, опцию **-R** все же захочется сделать умолчательной.

Для этого нужно внести изменения в конфигурационные файлы **aptitude**. Вообще-то **aptitude** обращается к тем же настройкам, что и **apt** (**/etc/apt/sources.list**, **/etc/apt/apt.conf**), однако имеет и собственный **~/aptitude/config**. По умолчанию он пуст, но может быть отредактирован по потребностям. В частности, для придания опции **-R** статуса по умолчанию, в этот файл следует внести такую строку:

```
aptitude::Recommends-Important «false»;
```

Можно, напротив, сделать так, чтобы при установке пакета автоматически инсталлировались также и «предлагаемые» (**suggest**) зависимости. Это достигается строкой

```
aptitude::Suggests-Important «true»;
```

Вообще-то опций конфигурирования для **aptitude** предусмотрено великое множество — и многие из них применимы не только к командному, но и к интерактивному режиму, позволяя настроить внешний вид интерфейса и многое другое. Ознакомиться с полным набором опций конфигурирования **aptitude** и их умолчательными значениями можно в официальной документации — она включена в состав дистрибутива и находится в каталоге **/usr/share/doc/aptitude/html/{lang}/**. Здесь под **{lang}** подразумевается язык документа — кроме английской (**en**) версии, в репозитории Ubuntu существуют переводы его на французский, финский и чешский языки; кстати, в репозитории Debian русской версии этого документа также не обнаруживается. А текущие настройки можно посмотреть в файле **/usr/share/aptitude/aptitude-defaults**.

В общем, после знакомства с **aptitude** можно сделать вывод, что она предоставляет все возможности, обеспечиваемые утилитами **apt-get** и **apt-cache**, плюс ряд дополнительных, подчас неочевидных, удобств, позволяющих, в частности, содержать пакетное хозяйство в стерильной чистоте. И потому использование ее в командном режиме предпочтительно перед указанными средствами комплекта **apt**.

Правда, тут я хотел бы отметить: совместное использование **aptitude** и **apt** видится мне нежелательным. То есть, по завершении настройки **aptitude** и приведении пакетного хозяйства к соответствию с ними, к командам **apt-get** и **apt-cache** лучше не прибегать. Команда **apt-cache** будет выдавать информацию в соответствии с настройками **aptitude**, а не умолчаниями **apt** — в частности, зависимости в выводе **apt-cache show** будут показываться так, как они описаны в файле **~/aptitude/conf**.

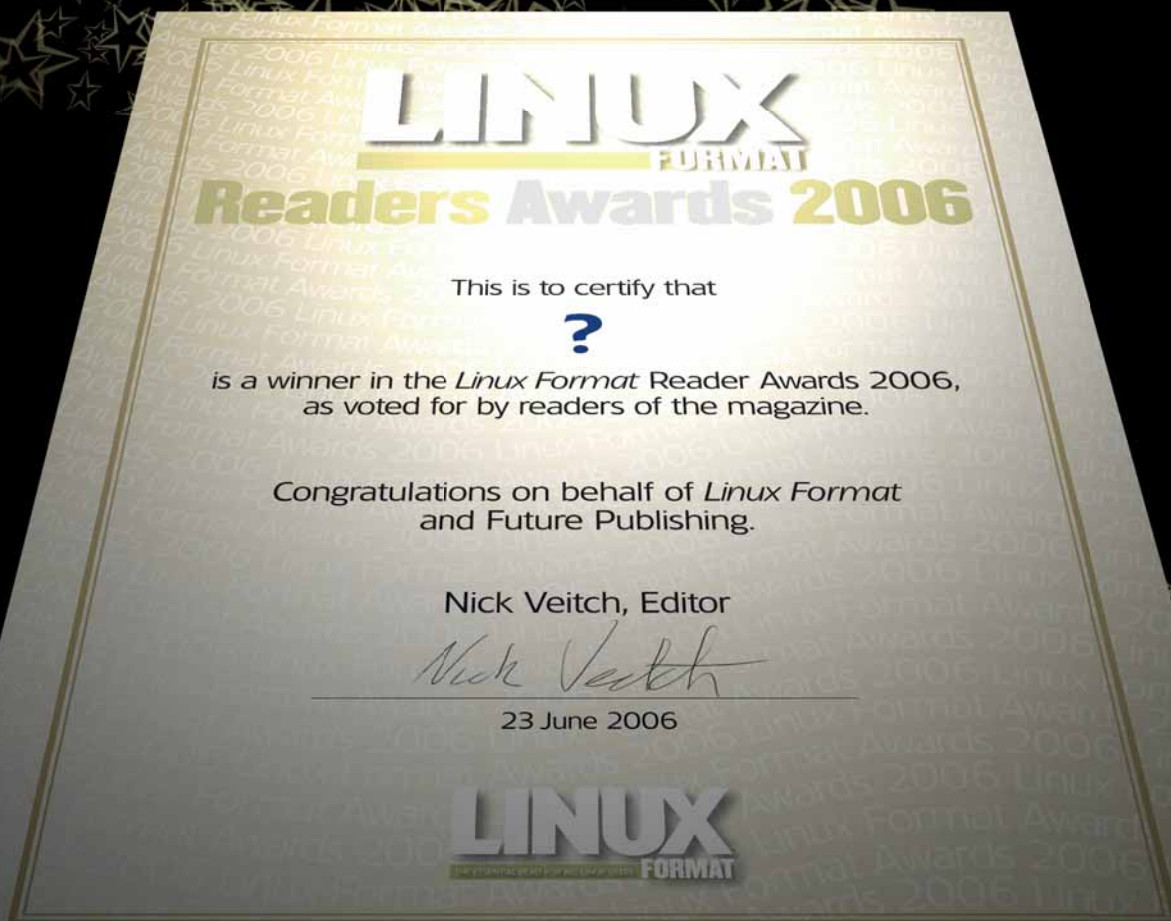
Что же до интерактивного режима **aptitude**, требующего запоминания многочисленных горячих клавиш, — это на любителя. Хотя в некоторых случаях — например, при очистке системы от категорий пакетов — и он может оказаться незаменимым.

В заключение я хотел бы выразить признательность Тихону Тарнавскому, благодаря которому я преодолел скептическое отношение к этой замечательной программе. **LXF**

LINUX FORMAT

Reader Awards 2006

В вашем распоряжении было четыре месяца, чтобы, выбрав из тысяч проектов, отдать за кого-то свой голос. А сейчас Пол Хадсон убавит свет и наденет смокинг, чтобы открыть имена... победителей приза читательских симпатий LXF Reader Awards 2006.



Лучший дистрибутив

Победитель: Ubuntu

Одного взгляда на обложку этого номера хватит, чтобы понять: мы предвидели, что вы выберете Ubuntu лучшим дистрибутивом года. Кто-то полюбил его за понятный дизайн, кто-то – за всеобъемлющий выбор пакетов, кто-то – за бесплатную систему рассылки Shiplt CD, но для большинства из нас Ubuntu – все это вместе взятое и объединенное в один пакет. Однако SUSE лишь на пять процентов отстал от Ubuntu, так что почивать на лаврах не стоит.



Лучшее Интернет-ПО

Победитель: Firefox

Легко обогнав соперников, победителем в номинации второй год подряд стал *Firefox*, набравший две трети голосов. Следующий кандидат, *Joomla*, набрал только 14% голосов. Это просто отлично для грядущего релиза 2.0 и еще раз доказывает, что пользовательское сообщество по-прежнему стоит за *Firefox*. Отлично, *Огненный Лис*!



Лучшее настольное приложение

Победитель: Firefox

Не удовлетворившись наградой в одной номинации, *Firefox* отнял награду в номинации Best Desktop Application у прошлогоднего победителя, *Gnome*. *KDE*, *OpenOffice.org* и *Amarok* – все произвели отличное впечатление, но против такого монстра, как *Firefox*, никому не устоять!

Лучшая игра для Linux

Победитель: Frozen Bubble

Как мы ни агитировали за могучую *Crack Attack!*, победить в этой сложнейшей категории ей не удалось. *Unreal Tournament 2004*, спустился на 3-е место, позади *SuperTux*. Ну, а на вершине пьедестала, обогнав всех на шесть пунктов, встал *Frozen Bubble*, классическая головоломка написанная на Python. Что ж, до 2007 года, *Crack Attack!*..



Свободное программное обеспечение – проект года

Победитель: KDE

Претенденты на нашу самую престижную награду шли в этом году буквально голова к голове. *Firefox*, уже отмеченный в двух номинациях, недобрал всего лишь 6 голосов до получения еще и этой награды. И KDE проскочил у него под носом! Будем надеяться, что KDE 4.0 оправдает наши чаяния!

Лучшее серверное ПО

Победитель: Apache

С самого начала в этой категории неизменно побеждал один и тот же продукт. И вот опять награду в номинации вы присудили *Apache*, 4-й год подряд собирающему больше голосов, чем все другие претенденты, вместе взятые.



Лучшее средство разработки

Победитель: GCC

Что ж, *GCC* и *Eclipse* всегда лидировали в этой категории, но награду может получить только один проект, и, хотя перевес был очень незначителен, победу вы присудили *GCC*. Он по-прежнему вездесущ, а сейчас стал еще более быстрым и совместимым, чем когда-либо.

Лучшая корпорация

Победитель: Novell

Эту награду обычно получали Red Hat и IBM, но не теперь: набрав больше голосов, чем они вместе взятые, победил Novell, ставший одной из ведущих корпораций, выступающих за свободное ПО. Теперь, когда сотни инженеров трудятся над разработкой ядра, Gnome, KDE, Samba, Mono и над многими другим, Novell стал очень популярен в сообществе – и это неудивительно. Сможет ли Canonical догнать его?



Лучшее офисное ПО

Победитель: OpenOffice.org

Интернет долго гудел после того, как в двух наших последних выпусках мы хорошенько врезали *KSpread*, и вы, похоже, разделяете наше мнение: *KOffice* набрал лишь 7% ваших голосов. Притом *KOffice* был вторым номинантом в данной категории, и это означает только одно: *OpenOffice.org* победил с отрывом более 84% голосов.

Лучшая поддержка оборудования

Победитель: Nvidia

Без хорошего драйвера ваша видеокарта превратится всего лишь в цветную плату. 3-й год подряд вы выбираете Nvidia, с чем мы полностью согласны: *Doom 3*, *Quake 4*, *Unreal Tournament* и другие игры безотказно работают на этих картах, благодаря неустанной поддержке Nvidia. А вот ATI не удалось даже войти в список претендентов...

Лучшее Linux-устройство

Победитель: Nokia 770

Известный также как Награда Невоспетому Герою, этот приз присуждается фирме, проекту или продукту, наиболее продвинувшим внедрение Linux в портативные устройства. Прошлогодний победитель, Trolltech, на этот раз стал вторым – ему не хватило чуть-чуть, а приз взяла Nokia за свой превосходный 770 Internet Tablet.



Лучший хостер или провайдер

Победитель: UKLinux.net

О, как много интернет-компаний беззаботно существуют, игнорируя своих пользователей и делая вид, что Linux не существует в природе. Именно поэтому так важна эта категория: она отмечает и награждает фирмы, поддерживающие пользователей Linux. DreamHost, новый номинант, вплотную приблизился к вершине пьедестала, но в этом году фанфары звучат в честь UKLinux.net – постоянного фаворита.

Лучший справочный ресурс

Победитель: Google

Несмотря на сильную конкуренцию со стороны ресурсов-конференций, таких, как LinuxQuestions.org или пользовательские форумы Gentoo и Ubuntu, на вершину пьедестала в этот раз попал Google, набрав полновесные 50% голосов.

Лучший гаджет года

Победитель: iPod

Эта категория всегда была нашей любимой по вполне очевидным причинам. Пусть Nokia 770 популярны, но их слишком мало выпущено для нужного числа голосов. Так что награду получает пользующийся многолетним признанием Apple iPod. Да, он – коммерческий, но благодаря маленькому волшебству от RockBox вы можете даже воспроизвести OGG! LXF



Reader Awards 2006

ЧТО ВЫ ДУМАЕТЕ?

Все наши лауреаты получат вожденный сертификат Linux Format – и бесценную поддержку читательской аудитории LXF. Спасибо вам за все ваши голоса. Результаты голосования уже не изменить, но мы предлагаем вам место, где вы можете обменяться мнениями о победителях и проигравших: пишите нам письма на lxf.letters@futurenet.co.uk или на форумы английского сайта www.linuxformat.co.uk.

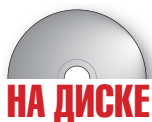
ИЗОБРАЖЕНИЕ IPOD ЯВЛЯЕТСЯ СОБСТВЕННОСТЬЮ APPLE

Новые рельсы Linux

ILLUSTRATION: NILS DAVEY

Правда ли, что Ruby + Rails + Linux – магическое уравнение, революционизирующее мир web-разработки? Пол Хадсон проводит собственное расследование...

WARNING
CODE CONSTRUCTION
IN PROGRESS



НА ДИСКЕ

- Ruby 1.8.4
- Ruby on Rails 1.1.3
- RubyGems 0.8.11

«Мощные web-приложения, разработка которых раньше занимала недели или месяцы, теперь создаются за дни.» – Тим О'Рейли.

Все мы знаем, что концепция открытого кода приносит компьютерным технологиям свободу, но также и пренебрежительное отношение к рыночным мифам, столь распространенным в мире закрытого ПО. Если что-то и вправду быстрее или обладает большими возможностями или легче в использовании, то мы хотим увидеть это своими глазами. Пусть вы с осторожностью воспринимаете похвалы, возносимые Ruby on Rails – ничто не мешает взять его из CVS и попробовать: это не будет стоить вам ничего, кроме затрат на web-соединение.

Проект начинался весьма скромно. Сам по себе язык программирования Ruby появился в 1995-м году как очередной скриптовый язык, позаимствовавший некоторые идеи из Perl, Eiffel, Smalltalk и Python. Однако его создатель Юкихио Матsumото [Yukihiro Matsumoto] ввел в язык собственный синтаксис, настоящую объектную ориентированность и продвинутую функци-

ональность, редко встречающуюся в других языках. Несмотря на это, Ruby не стяжал большого числа поклонников и рассматривался как легковесный конкурент для известного триумвирата парадигмы LAMP – PHP, Python и Perl.

С появлением Ruby on Rails ситуация в корне изменилась. Пару лет назад Давид Хайнемайер Ханссон [David Heinemeier Hansson] использовал Ruby в разработке приложений для своей фирмы 37 signals, занимающейся web-дизайном. Все они имели одну общую черту – им нужен был быстрый и легкий в использовании интерфейс, не требующий больших усилий в изучении, но предоставляющий мощную функциональность. После работы над этими приложениями Ханссон решил выделить свои наработки в отдельный пакет Rails и выпустить его под свободной лицензией.

С этого момента Rails и сам язык Ruby быстро приобрели известность как новый способ быстрой разработки web-сайтов с базой данных.

Сквозь мифы

Хотя Rails несомненно предоставляет новый интересный способ web-разработки, но этот проект создал вокруг себя густое облако маркетинговой пыли. Отчасти это объясняется относительной молодостью Rails (его релиз состоялся в прошлом году), отчасти связано с самой природой Сети, а отчасти с ощущением «крутости», созданным вокруг проекта элегантным сообществом Rails.

Конечно, не всякий рекламный миф обязательно неправда, но все же заявления требуют тщательной проверки. Как, например, в случае с фразой Тима О'Рейли – правда ли, что теперь проекты, разработ-

ВСТАЕМ НА РЕЛЬСЫ

Если вам нравится, как звучит фраза: «инфраструктура, оптимизированная для счастья программиста», то Rails, возможно, для вас. Ваш дистрибутив, наверно, содержит элементы Rails, как то: сам Ruby и менеджер пакетов RubyGems. Установить Rails можно

с помощью *RubyGems*. Некоторые дистрибутивы (их не много) предлагают Ruby on Rails прямо в основном списке пакетов, но проще всего использовать RubyGems, это позволит вам отслеживать обновления. www.rubyonrails.org/down

«НЕКОТОРЫЕ ЗАКАЗЧИКИ ДАЖЕ ПРОСЯТ ИМЕННО RUBY ON RAILS.»

ка которых занимала недели (или месяцы), можно создавать за считанные дни? Возможно; однако реальная длительность может быть разной. Например, один из авторов официального блога Rails сказал о недавнем проекте на базе Rails: «Кстати, двое программистов написали это на Ruby on Rails всего за шесть недель!». Звучит так, как будто бы при использовании обычного стека LAMP эта работа заняла бы годы, но какой же проект-мастодонт требует столько времени? Как выясняется, этим новым проектом был Интернет-магазин. Корзина очень приятная, ничего не скажешь, но должна ли реализация какой бы то ни было корзины двумя спецами на любом языке программирования занимать шесть недель?

Rails — это, конечно, не серебряная пуля для web-программирования, но мы обнаружили, что там сделан ряд шагов, на которых можно кое-чему научиться и применить это знание в своей работе. Если, как и большинство дальновидных фирм, вы предпочитаете не замыкать свой бизнес на единственную web-технологии, мы рекомендуем вам предоставлять решения на базе Rails наряду с PHP, Perl и JSP. Работая с Rails, вы не только получаете доступ к модной технологии (некоторые заказчики даже просят именно Ruby on Rails), но ваши инженеры заодно получают шанс увидеть, в чем, собственно, Rails лучше, и воспользоваться этим в других проектах.

Выбирая Р***

Большинству web-разработчиков хорошо знакомы основы стека LAMP. L-часть (Linux) позволяет вам выбирать из широкого спектра дистрибутивов — Red Hat, SUSE, Gentoo и сотни других. P-часть также довольно гибка; здесь свою популярность доказали языки PHP, Perl, Python, но и для Ruby место найдется.

Однако части A (*Apache*) и M (*MySQL*) уже не так гибки, поэтому опасения, что Ruby on Rails является конкурентом LAMP, видимо, беспочвенны. Если убрать *Apache* и Microsoft IIS, то на оставшуюся часть рынка придется менее чем 10% web-серверов: несущественная доля. Что касается *MySQL*, то он стал стандартом де-факто для баз данных web-сайтов, несмотря на то, что его конкурент *PostgreSQL* имеет больше возможностей и более надежен. Так что Rails

вряд ли сможет раскатать корабль LAMP. К тому же Хайнемайер Ханссон дал понять, что он предпочитает использовать базы данных в качестве не более чем пассивного хранилища, а бизнес-логику держать на уровне Rails. Этот принцип дал преимущество *MySQL* над *PostgreSQL*, но также привел к потере масштабируемости многих сайтов. И, чтобы облегчить существование в стеке LAMP/R, примите основную рекомендацию — запускать Rails на платформе Unix, это на худой конец обезопасит вложения в ваш сервер.

При всем при том на сайте Rails гордо заявляют, что вся основная команда разработчиков ведет разработку в OS X при помощи инструмента с закрытым кодом — редактора *TextMate*. *TextMate* — потрясающий текстовый редактор, и нам стыдно признать, что у нас нет его эквивалента с открытым кодом для людей, собирающихся использовать Rails.

Где Rails действительно выигрывает, так это в философии «не болтай, а покажи». Идея здесь в том, что читать чужой код — занятие довольно скучное. Вместо этого разработчики Rails создали несколько видеороликов, показывающих как они вживую создают web-страницы — печатают код, делают изменения, просматривают результаты и т.п. (www.rubyonrails.org/screencasts). Скорострельный цикл разработки кода Rails отлично подошел для видео — можно и вправду наблюдать, как приложение создается прямо у вас на глазах.

Переход

Сравнивать языки программирования непросто, поскольку такое сравнение во многом зависит от вашего прошлого опыта. Некоторые языки, например, PHP, хотя бы имеют сходство с уже привычными, например, Java и C. Но Ruby выглядит совершенно уникально, поэтому вам придется изучать его с нуля. Утешает только тот факт, что в Rails объем кода, который вам придется писать, сведен к минимуму.

Некоторые опытные разработчики предают анафеме «закулисную» магию — они любят иметь явный контроль над всем. Но сколько раз вы готовы писать один и тот же цикл для печати строки таблицы? Сколько раз вы готовы писать HTML для вставки дат? Это небольшие нюансы, которые



Давид Хайнемайер Ханссон закончил факультета бизнеса и компьютерных наук в Копенгагене и периодически появляется на постерах, посвященных Open Source.

ФОТО ЯВЛЯЕТСЯ СОБСТВЕННОСТЬЮ JAMES DUNCAN DAVIDSON/RELLY

Rails может убрать из вашей повседневной работы, позволив вам сфокусироваться на более высокоуровневых вещах: заставить ваш код работать, и работать хорошо.

Кое-кто уже считает Rails серьезным сдвигом парадигмы, но для этого Rails должен сделать побольше, чем отвести от себя подозрения в чрезмерности рекламы. Лучше думать о Rails как о еще одном ярком кусочке в мозаике программных инструментов web-разработки и еще одной причине выбрать свободное ПО вместо продуктов Microsoft. **LXF**





Армированный Linux

AppArmor – отличный, гибкий инструмент безопасности для Linux. Его создатель, **Криспин Коуэн**, поведает вам о нем.

ОБ АВТОРЕ

Криспин Коуэн (Crispin Cowan) основал Immunix, фирму, стоящую за *AppArmor*, и в настоящее время работает в Novell



разработчиком средств безопасности для платформы Linux и приложений Linux от Novell. Среди его достижений – соавторство в изобретении метода выбора безопасного момента для наложения заплаток – ‘time-to-patch’.



НА ДИСКЕ

- **AppArmor 2.0 для Ubuntu и Slackware**

Основная беда безопасности приложений – невозможно сказать, как поведет себя ваше приложение в нестандартных условиях: число состояний и вариантов ввода слишком велико, чтобы исчерпывающе протестировать даже небольшую программу. Когда это выливается в проблему? Когда некто обнаруживает, что определённый набор входных данных приводит к некорректному поведению программы, и разрабатывает приложение, эксплуатирующее найденную уязвимость (подобные приложения так и называют – эксплойты, exploits).

Для решения проблемы эксплойтов и разработан *AppArmor*. *AppArmor* обеспечивает безопасность приложений, заставляя их делать только то, что они обязаны делать и не более того. Конкретно, *AppArmor* контролирует файлы и функции POSIX.1e (см. **man 7 capabilities**), к которым приложение имеет доступ, поскольку контроль над этими ресурсами и привилегиями на вашей машине и есть мечта злоумышленника. В данной статье мы разьясим, как настроить и использовать *AppArmor* для защиты ваших программ и системы от атак.

AppArmor полностью интегрирован в SUSE Linux 10.1, SLES 9 SP3 и SUSE Linux Enterprise 10. Он устанавливается и работает по умолчанию, но использует скромный набор профилей (об этом позже). Намного больший набор профилей можно найти в `/etc/apparmor/profiles/extras/`; вы можете скопировать их в `/etc/apparmor.d`, где хранятся активные профили. В SUSE, интерфейс пользователя *Yast* содержит кнопку *AppArmor*, которая выводит графический эквивалент описываемых здесь инструментов *genprof* и *logprof*.

AppArmor вступил в жизнь как проприетарная система, но после приобретения Immunix Inc в 2005 году фирмой Novell приложение выпускается по лицензии GPL. С тех пор *AppArmor* был портирован на Slackware, Ubuntu и Pardus Linux, турецкий дистрибутив.

Для установки *AppArmor* в Ubuntu, выполните

```
sudo dpkg -i apparmor-* libapparmor1_2.0-0ubuntu2_i386.deb linux-image-2.6.15-21-386_2.6.15-21.32mr1_i386.deb
```

Заметьте, что эти пакеты включают новое ядро, поскольку для поддержки *AppArmor* необходима небольшая заплатка для ядра. Ubuntu основан на Debian, так что пакеты прекрасно подойдут для Debian,

MEPIS и прочих систем на базе Debian. Если вы хотите использовать *AppArmor* в другой системе Linux, потребуется его портировать – см. врезку «Двигаемся дальше» на странице 47.

Обезопасим NTPD

Теории достаточно; обратимся к примеру. Рассмотрим NTPD, Network Time Protocol Daemon. Он обеспечивает синхронизацию вашей системы с главным сервером вашей локальной сети или с интернетом – важная задача! Однако для изменения вашего системного времени NTPD потребуются привилегии суперпользователя (**root**), а также открытый сетевой порт, чтобы узнавать время на главном сервере. В итоге получается убойная комбинация: открытый сетевой порт и доступ к процессам с правами суперпользователя – классический пример из руководства по безопасности. Злодей, нашедший уязвимость в коде NTPD, сможет атаковать демона NTP и получить права суперпользователя на вашем компьютере.

Права суперпользователя отнять нельзя – ведь тогда NTPD не сможет менять время; открытый сетевой порт удалить тоже нельзя – тогда NTPD не сможет узнавать время. Тут-то и поможет *AppArmor*, ограничив привилегии NTPD посредством про-

фия безопасности. Вот *AppArmor*-профиль для программы *ntpd*:

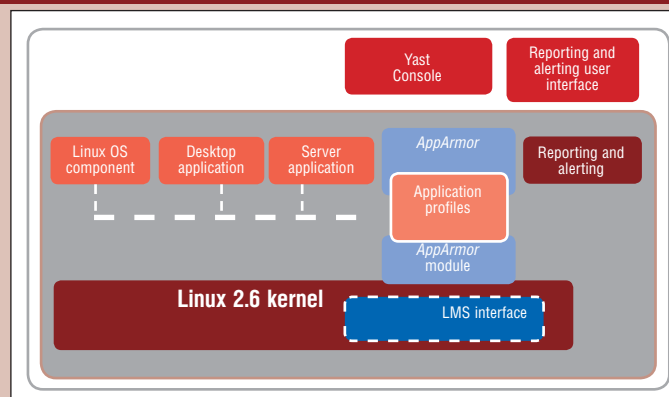
```

/usr/sbin/ntpd {
#include <abstractions/base>
#include <abstractions/nameservice>
capability ipc_lock,
capability net_bind_service,
capability setgid,
capability setuid,
capability sys_chroot,
capability sys_resource,
capability sys_time,
/drift/ntp.drift rwl,
/drift/ntp.drift.TEMP rwl,
/etc/ntp.conf r,
/etc/ntp/drift* rwl,
/etc/ntp/keys r,
/etc/ntp/step-tickers r,
/tmp/ntp* rwl,
/usr/sbin/ntpd rix,
/var/lib/ntp/drift rwl,
/var/lib/ntp/drift.TEMP rwl,
/var/lib/ntp/drift/ntp.drift r,
/var/lib/ntp/var/run/ntp/ntpd.pid w,
/var/log/ntp w,
/var/log/ntp.log w,
/var/run/ntpd.pid w,
}
    
```

Сейчас *ntpd* запускается от имени *root*, то есть имеет доступ ко всем функциям POSIX.1e. Но этот профиль ограничивает доступ *ntpd* семью функциями, описанными в коде. Будучи *root*-процессом, *ntpd* мог бы читать, записывать и запускать любой файл в файловой системе, но теперь ему доступны только файлы, описанные в профиле.

Что же произойдёт, при наличии данного профиля, если злоумышленник припас программу-экспloit, направленную против *ntpd*? Будьте спокойны: ничего. У него нет прав на выполнение, а значит, простой shell-код [«полезная нагрузка» эксплойта, выполняющая задуманные злоумышленником действия, — прим. ред.] вида *exec(sh)* не пройдет. Если *ntpd* зачем-либо потребует командную оболочку (*shell*), злоумышленник сможет запустить ее, но только из-под профиля, и сможет читать и записывать лишь файлы, принадлежащие *ntpd* на основании профиля. Короче говоря, злоумышленник сможет разве что изменить ваше системное время. *AppArmor* разрешил *ntpd* выполнять только конкретные операции — в литературе по безопасности это называется запуском приложения с наименьшими привилегиями.

КАК УСТРОЕН APPARMOR



- **Ядро Linux 2.6** Имеет новый интерфейс модулей безопасности Linux (Linux Security Modules), допускающий подключаемые модули безопасности. Процессы пользователя обращаются к ядру, которое при помощи всех загруженных LSM-модулей проверяет корректность доступа.
- **Модуль ядра AppArmor** Обеспечивает реальную политику безопасности, сверяя обращение программы с профилем и запрещая либо разрешая действие.
- **Профили приложений** Содержат вашу политику безопасности и описывают, что позволено делать каждому приложению.
- **AppArmor** Набор программ уровня пользователя для отображения и управления профилями *AppArmor*. Наиболее важным из них является синтаксический анализатор (*parser*), который читает профили, преобразует их в структуры данных, понятные ядру, и переправляет их модулю *AppArmor*.
- **Отчеты и предупреждения** Предоставляет отчеты о конфигурации *AppArmor* и предупреждает о попытках вторжения.
- **Yast-интерфейс** В системах без *Yast*, *AppArmor* может управляться посредством его родного интерфейса командной строки.

КАК ОБЕЗОПАСИТЬ АРАСНЕ И ДРУГИЕ ПРИЛОЖЕНИЯ

В мире огромное число приложений — фактически бесконечное, ведь люди не перестают писать код. И если вы собираетесь использовать систему безопасности, контролирующую деятельность приложений, очень важно, чтобы поддержка совместимости политик безопасности с программным обеспечением была простой, особенно при наличии постоянно прибывающих новых приложений.

Профили *AppArmor* достаточно просты: нетрудно сообразить, как создать их вручную. Фактически, когда в 1998 г. появился *AppArmor*, именно так и делалось. Сейчас *AppArmor* поставляется с обучающейся системой: вы запускаете ваши приложения как обычно, а *AppArmor* следит и запоминает, что требуется разрешать приложению.

Выбрав приложение, для которого нужно разработать профиль, вы отправляете модулю ядра исходный вариант профиля, функционирующий в режиме 'complain' («жалоба»): нарушения политик не блокируются, а записываются в log-файл (журнал). Затем вы запускаете приложение обычным способом, и оно какое-то время работает, заодно заполняя журнал событий, характеризующих его поведение. После этого вы применяете к созданному журналу анализатор *AppArmor*, который задаст вам интерактивные вопросы о добавлении событий в окончательный профиль.

Фабричные профили

Скрипты, поставляемые с *AppArmor*, выполняют за вас большую часть работы. Посмотрим, как можно создать профиль для вашего web-сервера и его web-приложений. Выполните от имени суперпользователя (*root*) `genprof /usr/sbin/httpd2-prefork`. Если для *Apache* ещё нет профиля, *genprof* создаст его, а если есть, то воспользуется уже существующим. Загрузив профиль в ядро в *complain*-режиме и попросив вас запустить ваше приложение с другого терминала или иным способом, *genprof* примется создавать журнал событий. В случае *Apache*, поупражняйтесь с `/etc/init.d/apache stop` и `/etc/init.d/apache`.

Затем посетите ваш web-сайт и займитесь обслуживаемыми web-страницами. Как вы скоро увидите, посещать каждую отдельную страницу не обязательно, хотя это довольно просто, если использовать «паучка», например: `wget --mirror http://example.com`. Зато следует запустить каждый исполняемый файл, относящийся к сайту. Посетив достаточное для представления количество web-страниц, вновь выполните `/etc/init.d/Apache stop` и вернитесь к *genprof*.

Нажмите S, чтобы попросить *genprof* просканировать журнал событий, который находится в `/var/log/audit/audit.log`, и *genprof* задаст вам серию вопросов о том,

как обрабатывать события. Вопросы задаются по блокам; первый — об управлении дочерними процессами, запускаемыми из программы, для которой создается профиль. Выберите из следующего:

Inherit (наследование) Дочернее приложение (потомок) запускается с тем же профилем безопасности, что и родитель. Идеально для служебных программ; пример — сценария командной оболочки, запускающего программу *sr* с целью скопировать файл.

Profile (профиль) Потомок имеет свой профиль. Осторожно: этот профиль имеет действие во всей системе, так что для его создания вам придётся проде-



```

crispin@groo:~ - Shell - Konsole <5>
Session Edit View Bookmarks Settings Help

[[A]]llow / (D)eny / (G)lob / Glob w/(E)xt / (N)ew / Abo(r)t / (F)inish

Profile: /usr/sbin/httpd2-prefork
Path: /etc/apache2/conf.d/apache2-manual.conf
Mode: r
Severity: 3

1 - /etc/apache2/conf.d/apache2-manual.conf
[2 - /etc/apache2/conf.d/*]

[[A]]llow / (D)eny / (G)lob / Glob w/(E)xt / (N)ew / Abo(r)t / (F)inish

Profile: /usr/sbin/httpd2-prefork
Path: /etc/apache2/conf.d/apache2-manual.conf
Mode: r
Severity: 3

1 - /etc/apache2/conf.d/apache2-manual.conf
2 - /etc/apache2/conf.d/*
[3 - /etc/apache2/**]

[[A]]llow / (D)eny / (G)lob / Glob w/(E)xt / (N)ew / Abo(r)t / (F)inish
    
```

TTY-интерфейс *AppArmor*: здесь можно настроить профили и без *Yast*.



AppArmor позволяет вам создавать профили самим, но его система мастеров (в данном случае, в YaST) упростила эту задачу.



лать большую работу. Если этого потомка (например, web-приложение, обычно запускаемое web-сервером) использует только одна программа, то выбор Р является безопасным. Если потомок используется разными программами, придётся погонять его во всех режимах, чтобы правильно определить его поведение для профиля.

Unconfined (неограниченный)

Потомок запускается вообще без профиля. Это очень опасный выбор, предполагающий, что у потомка нет уязвимостей. Он существует преимущественно для того, чтобы позволить системным администраторам удаленно управлять AppArmor, потому как программам с ограничениями манипуляции с AppArmor не разрешены.

Deny (запрещено) просто не позволяет родителю запускать этого потомка.

В каждом из блоков вопросов также имеются дополнительные опции **abort**, которая отменяет все изменения и выходит из *genprof*, а также **Finish**, которая обрывает создание профиля на данном этапе и осуществляет выход.

Следующий блок вопросов касается функций POSIX.1e. Здесь имеются опции **Allow (Разрешить)** приложению доступ к этим функциям, или **Deny (Запретить)**. Если вы уверены, что ваше приложение не

атаковалось в момент обучения, то **Allow**, скорее всего, верный ответ. Однако если профиль создается на машине, открытой для Интернета, ваше приложение вполне может быть атаковано во время «обучения», и здесь нужна осторожность. Создание профиля приложений подобно обучению детей: они учатся на примерах, поэтому не делайте ничего дурного в их присутствии. Последний блок вопросов касается доступа к файлам. Кроме простых **Allow** и **Deny**, здесь есть опции:

Glob this access («Глобализовать» доступ) Каждое нажатие клавиши G заменяет на шаблон очередной уровень «хвоста» полного (включающего путь) имени файла. Одно нажатие G превратит */srv/www/htdocs/images/star.gif* в */srv/www/htdocs/images/**, второе – в */srv/www/htdocs/**, и т.д.

Glob with extensions («Глобализовать» с расширением) Создает шаблон из последней части пути и расширения, так что */usr/local/myfile.html* превратится в */usr/local/*.*html*.

New path specifier (Описание нового пути) Добавьте свой собственный путь путем создания шаблонов вручную, используя *, ?, [,], -, { и }.

Получив ответ на все вопросы, *genprof* вновь попросит перейти к сканированию журнала; на сей раз нажмите F. Программа *genprof* завершится, переведя профиль Apache в режим подавления (enforce-режим). Вам не потребуется перезапускать Apache; AppArmor может изменять статус профилей на фоне запущенного процесса. Теперь ваша система функционирует с ограничениями AppArmor согласно созданной вами политике. Эта политика будет работать и после перезагрузки – вы можете посмотреть ее в */etc/apparmor.d/usr.sbin.httpd2-prefork*.

Общие правила

Вопросы о файловом доступе, на которые вы отвечали, лишь номинально относятся к запретам. Намного важнее возможность создать общее правило доступа к файлам. Простейший случай – работа с временными (temporary) файлами, которым обычно намеренно даются случайные имена. Заменив последнюю часть пути шаблоном, вы будете уверены, что программа будет нормально работать, а не закроется с ошибкой, потому что имена временных файлов с каждым запуском меняются.

Другой пример обобщения – имена библиотек. Например, если программе необходим доступ к */usr/lib/libdb_cxx-4.3.so*, наш друг *genprof* может предложить альтернативу */usr/lib/libdb_cxx-*.so*, чтобы профиль работал и после обновления версии библиотеки. Пользователю предо-

ставляется компромисс: широкое использование шаблонов сделает профили более удобными и безотказными, но платой за это будут большие разрешения и, как следствие, снижение безопасности. Но даже с широким использованием шаблонов профили AppArmor все еще более безопасны, чем без AppArmor.

Вот ещё один пример: *genprof* замечает, что событие доступа имеет соответствие в каталоге **abstractions**, и предлагает, скажем, нечто вроде **#include <abstractions/namespace>**, тем самым обеспечивая все правила доступа, необходимые для DNS-запросов и локальных UID-запросов.

Под надзором

Программа *genprof* помещает отметки в журнале и, таким образом, обучается только на событиях, происходящих в течении данной сессии *genprof*. Но вы можете обучать программу хоть целую неделю, невзирая на перезагрузки. Команда **complain /usr/bin/моя_программа** или **complain /etc/apparmor.d/usr.bin.моя_программа** переведет профиль программы *моя_программа* в complain-режим и оставит его в этом состоянии, а **complain /etc/apparmor.d/*** переводит все профили вашей системы в complain-режим.

Запустите ваше приложение на сколь угодно долгое время, позаботившись, чтобы все журналы сохранялись и не удалялись при ротации. Когда вы посчитаете, что набрали достаточное количество событий, используйте программу *logprof* для обработки журнала так же, как делает *genprof* – только *logprof* прочитает весь файл журнала. *logprof* имеет ключи командной строки, указывающие, где искать журналы, с какого места журнала начинать анализ и где находится профиль, подлежащий обновлению.

Чтобы полностью разделить процессы обучения и создания профиля, отработайте все обучающие режимы для генерации журнала событий на большой тестовой машине в QA-подразделении, а затем просто отправьте набор профилей и log-файлов по электронной почте эксперту по безопасности, который при помощи *logprof* преобразует события в профили. По завершении выполните команду *enforce*, которая работает точно так же, как команда *complain*, но переводит все профили в режим 'enforce' («подавление»).

Эти команды AppArmor могут быть выполнены только из командной оболочки root, причём оболочка должна быть свободна от ограничений AppArmor. Все команды также имеют псевдонимы, начинающиеся с **aa-**, таким образом вы сможете обнаружить все команды AppArmor в вашей системе, просто набрав **aa-<TAB>** в root-строке.

СОВЕТЫ ПО ПРОФИЛЯМ APPARMOR

- При обучении AppArmor, всегда оставляйте окно терминала открытым, выполнив команду **tail -f /var/log/audit/audit.log**, чтобы видеть в реальном времени, чем занимается AppArmor.
- Если вам показалось, что AppArmor где-то навредил, грегните файл **APPARMOR /var/log/audit/audit.log**. Если в нём ничего нет, то причина проблемы – не AppArmor. Если есть, то AppArmor объяснит, почему он что-то заблокировал. Еще лучше запустить **logprof**, чтобы AppArmor нашел блокирующее событие и спросил, хотите ли вы добавить его в

список разрешенных для данного профиля.

- Команда **complain /etc/apparmor.d/*** быстро переведет всё в complain-режим, разрешающий все действия, так что AppArmor вообще ничего не будет блокировать. И наоборот – **enforce /etc/apparmor.d/*** блокирует всё.
- Прежде чем создавать новый профиль, загляните в */etc/apparmor/profiles/extras/* – нет ли готового профиля, чтобы взять его в качестве исходного: это значительно сократит работу. AppArmor прекрасно улучшает существующие профили.

КАК ОБЕЗОПАСИТЬ СВОЮ СИСТЕМУ И РАБОЧИЙ СТОЛ

Существует две системы политик безопасности. Системы Черного списка перечисляют запрещенные вещи*, а всё остальное разрешается; таким образом, это системы, разрешающие по умолчанию. Системы Белого списка определяют то, что разрешено, а всё остальное запрещается, то есть это системы, запрещающие по умолчанию.

Понятно, что системы Белого списка безопаснее, поскольку злоумышленник не может их обойти, просто приискав экзотический способ нагадить, который не пришёл в голову составителям списка. Но системы Черного списка удобнее для пользователей, потому что они не блокируют вас всякий раз, когда вы изобретаете новый способ применения своего ПК.

AppArmor обходит дилемму «безопасность–удобство», используя гибридную модель. При использовании профиля для конкретной программы AppArmor действует как Белый список: разрешены только указанные файлы и функции, и профиль не позволит злоумышленнику подsunуть программе новый вариант работы во вред. Но на уровне системы AppArmor является Черным списком: ограничиваются только программы, обладающие профилем, и их потомки, а прочие программы работают без ограничений.

Но как же определить, какую программу следует ограничить? Это зависит от вашей модели угрозы. Обязательно должны быть ограничены права всех приложений, имеющих нефильтруемый ввод данных, через которые можно осуществить атаку, потому что AppArmor никак не защищает программы без ограничений.

Открыты сетевые порты

Очень часто встречающаяся модель угрозы – сетевая: плохой парень «откуда-то оттуда», взламывающий ваш сервер или настольный компьютер. AppArmor предлагает обороняться программой *unconfined*. Она сканирует вашу машину на наличие открытых сетевых портов, находит прослушивающие их программы и выводит их список, а также их профили, если они есть. Если вы создали профили для всех программ, отмеченных в отчете *unconfined*, все открытые сетевые порты «упрутся» в профили AppArmor, то есть политика AppArmor полностью определит, что сетевой злоумышленник сможет сделать на вашей машине, даже если для всех прочих программ вашей системы вы профилей не создавали.

Вот листинг результата команды *unconfined* в защищенной системе (web- и mail-сервер в берлоге автора):

```
4406 /sbin/dhcpd confined by '/sbin/dhcpd (enforce)'
4758 /sbin/portmap confined by '/sbin/portmap (enforce)'
4758 /sbin/portmap confined by '/sbin/portmap (enforce)'
5339 /usr/sbin/sshd confined by '/usr/sbin/sshd (enforce)'
5479 /usr/sbin/httpd2-prefork confined by '/usr/sbin/httpd2-prefork (enforce)'
6151 /usr/sbin/dhcpd confined by '/usr/sbin/dhcpd (enforce)'
6151 /usr/sbin/dhcpd confined by '/usr/sbin/dhcpd (enforce)'
6263 /usr/lib/postfix/master confined by '/usr/lib/postfix/master (enforce)'
6263 /usr/lib/postfix/master confined by '/usr/lib/postfix/master (enforce)'
6329 /usr/sbin/xinetd confined by '/usr/sbin/xinetd (enforce)'
6329 /usr/sbin/xinetd confined by '/usr/sbin/xinetd (enforce)'
8767 /usr/sbin/cupsd confined by '/usr/sbin/cupsd (enforce)'
8767 /usr/sbin/cupsd confined by '/usr/sbin/cupsd (enforce)'
```

ПК на замке

Описанный профиль подходит для сервера сетевой группы, поддерживающего web, электронную почту и печать. А как с безопасностью настольных ПК? Используется тот же подход: ограничить все приложения, подверженные атакам из сети. То есть любые приложения, работающие с сетевыми данными.

Некоторые из этих приложений имеют прямой вход из сети, например, почтовый клиент. Другие получают данные из сети опосредованно (например, *OpenOffice.org*, использующийся для открытия DOC-файлов, которые некто выслал вам в виде почтового вложения). Несмотря на отсутствие прямого соединения с сетью, они осуществляют сетевой ввод и тем создают нештатную опасность. Анализатор *unconfined* поможет и здесь, но учтите, что он не сможет найти приложения с чрезвычайно изменчивыми открытыми портами (некоторые IM-клиенты) и, естественно, приложения с опосредованным соединением (такие, как OOo).

Наша настольная политика защищает систему от сетевых атак, но ничего не делает для защиты от консольного пользователя, поскольку в нашем представлении пользователь – человек опытный, который выложил немалые деньги за свой ПК и полностью его контролирует. Однако на рабочей станции переговорного пункта или в общественном киоске консоль становится

основным источником опасности. Для устранения угрозы следует выявить все приложения, предоставляемые терминалом, и убедиться, что все они имеют профиль AppArmor. В данном случае клавиатура и мышь обязаны рассматриваться как угроза, точно такая же, какой сетью является для нашего сервера. Только создав профили для всего, до чего может добраться и использовать злоумышленник, мы можем считать свой долг выполненным.

Гибкость Черного списка всей системы AppArmor позволяет обезопасить только те программы, которые внушают беспокойство. Платой за удобство является необходимость определения таких программ,

«ПРАВА ВСЕХ ПРИЛОЖЕНИЙ С НЕФИЛЬТРУЕМЫМ ВВОДОМ ДОЛЖНЫ БЫТЬ ОГРАНИЧЕНЫ.»

для чего надо представить себе источники угрозы. Многие из нас ожидают угрозу преимущественно из сети, вот почему программа *unconfined* столь полезна. По-программный Черный список AppArmor разрешает каждой программе делать только то, что ей полагается, и ничего другого, игнорируя атаки злоумышленников. Комбинация списков помогает вам контролировать работу вашей системы легко и эффективно. Наслаждайтесь! **LXF**

ДВИГАЕМСЯ ДАЛЬШЕ

Если вы не являетесь пользователем SUSE или Debian, но хотите добавить брони AppArmor в вашу систему Linux, то программу можно портировать. Исходный код (<http://forge.novell.com/modules/xfmod/project/?apparmor>) не зависит от архитектуры и поддерживается на всех аппаратных платформах SUSE Linux (x86, x86-64, Itanium, PowerPC, и Z-series) – один из разработчиков AppArmor в свое свободное время портировал его на свой Sharp Zaurus PDA (ARM). Вот шаги для успешного портирования:

1. Установите на ядро заплатку, экспортирующую семафор пространства имен для ядра 2.6.15 или выше.
2. Убедитесь, что модуль подготовлен для используемого вами ядра.
3. Убедитесь, что скрипты инициализации вашего дистрибутива могут монтировать securityfs.
4. Создайте пакет для вашего дистрибутива аналогично RPM для SUSE и Debs для Ubuntu.

5. Создайте или измените профили в соответствии с вашим дистрибутивом. В частности, существует небольшой набор важных профилей, которые должны быть подправлены, особенно для набора библиотек **ld.so**. Удачи!

Проект AppArmor также пытается привлечь больше участников из сообщества Open Source. Если вы администратор или пользователь Linux, вы можете запускать приложения для своих целей и публиковать результаты в списке рассылки *apparmor-general* (<http://forge.novell.com/mailman/listinfo/apparmor-general>). Если вы разработчик приложений, подумайте над разработкой и распространением профилей AppArmor с вашим приложением. Намного легче обслуживать профили AppArmor, если они являются частью процесса разработки приложения. Или вы можете помочь в разработке новых возможностей AppArmor. Узнайте больше, посетив <http://en.opensuse.org/Apparmor>.



Recovery is possible

СПАСЕНИЕ УТОПАЮЩИХ КАК ДЕЛО РУК САМИХ УТОПАЮЩИХ

Создать свой собственный LiveCD совсем не сложно – надо только иметь под рукой подходящую основу и понимать базовые принципы. Первое «безвозмездно, то есть даром» предоставляет RIP, а разобраться со вторым вам поможет статья **Владимира Попова**.

Зачем?

Вряд ли в настоящее время требуется объяснять, что такое Live CD. CD-диск (или flash-накопитель, что в данном случае не имеет принципиального значения), несущий на себе «самодостаточную» ОС, неоднократно признавался специалистами в сфере программного обеспечения одним из наиболее примечательных решений последнего десятилетия. «Спасательные» (rescue) системы, инсталляционные диски дистрибутивов, демо-версии ОС и отдельных приложений, «целевые» (обеспечивающие решение определенного круга задач) системы – все они, по сути, представляют собой Live CD.

Подобное разнообразие сначала приводит к вопросу: «И как же это делается?», – а затем и к размышлениям: «А не создать ли мне свой собственный Live CD?». Если вы профессионально занимаетесь обслуживанием вычислительных систем, то потребность в собственном Live CD почти наверняка есть. Аналогично, если вам часто случается проводить демонстрации на «чужих» компьютерах, то история создателя Knoppix Клауса Кнопфера (Klaus Knopper) может показаться интересной. Одним словом, тема представляется достойной обсуждения.

Есть два основных способа научиться что-то делать: получить фундаментальные знания и только потом перейти к практике или идти от прототипа, осваивая знания и получая навыки одновременно, изучая прототип и создавая собственную «версию». Нет смысла спорить об эффектив-

ности обоих способов, но «веселее» все-таки второй. По крайней мере, если поставленная цель – создание собственного Live CD.

Почему – RIP?

По данным Distrowatch.com, на настоящий момент известно 224 Live CD, из которых 176 – активно поддерживаются. Который выбрать в качестве прототипа для собственной разработки? Как сказал один пролетарский поэт: «делать жизнь – с кого?» Я бы рекомендовал RIP Кента Роботти (Kent Robotti). Вопреки мрачному названию («R.I.P.» – традиционная надпись на надгробьях в англоязычных странах, представляющая собой аббревиатуру печальной фразы «Rest In Peace» – «покойся с миром»), назначение этого Live CD исключительно оптимистично, а название означает ни много, ни мало «Recovery Is Possible», то есть: «Восстановление – возможно!».

В актив RIP можно занести следующее:

- компактность: iso-образ в 32 МВ – совсем не много по нынешним временам;
- современность, которая отчасти predeterminedена компактностью: поддерживать актуальность ядра, библиотек и утилит для дистрибутива с сотней приложений намного проще, чем для дистрибутива, количество приложений в составе которого перевалило за две тысячи;
- рациональность: RIP быстр, независим от привода CD после загрузки и полезен, поскольку сохранность данных –

проблема общая и для программиста, и для дизайнера, и для бухгалтера.

Еще одно обстоятельство, свидетельствующее в пользу RIP – личность хранителя. Кент Роботти известен как автор многих патчей и нескольких программ: *pkgtool2*, *make_uninstall*, *mnsetup*, *grubconfig*, *backup-mbr*. Приведенный список однозначно говорит о том, что автор хорошо ориентируется в различных Linux-дистрибутивах (*pkgtool2/make_uninstall* предлагают универсальный способ инсталляции приложений и конвертации *tgz/rpm/deb*), сведущ в коммуникациях (*mnsetup* автоматизирует установку ПО, необходимого для организации mail и news), занимался вопросами загрузки (*grubconfig*) и не понаслышке знаком с проблемами потери информации на HDD (*backup-mbr*).

Мне, во всяком случае, всегда было интересно знакомиться со взглядами Кента на текущие возможности ядра и алгоритм загрузки, список «первоочередных» утилит и средства восстановления файловых систем. До сих пор разочарован я не был ни разу. Поэтому и предлагаю, вслед за Кентом Роботти, быстрое действие предпочесть универсальности, а компактность – красоте графической среды.

Принципы

Более десяти лет прошло с тех пор, как с «легкой руки» Вернера Алмсбергера (Werner Almesberger) и Ганса Лермена (Hans Lermen) в коде ядра Linux-1.3.73 появилось устройство *initrd* (initial RAM disk). Потребность в послед-

нем объяснялась просто: корневая файловая система Linux может располагаться, грубо говоря, «черт-те где». Помимо само собой разумеющихся HDD и FDD, это могут быть SCSI, Zip и CD-ROM накопители, сетевые ресурсы и т.д. и т.п. Что же, ядро должно изначально поддерживать все эти устройства? Накладно, но возможно. А как быть, если монтированию корневой файловой системы обязательно предшествуют операции настройки (сетевые, например)? Вместо усложнения ядра было предложено снабжать его «начальной файловой системой в памяти». В конце концов, если ядро как-то попало в память (от тонкостей аппаратной реализации загрузки абстрагируемся), то почему бы тем же способом в память не загрузить файловую систему, содержащую все необходимое для монтирования уже «настоящей» файловой системы?

Практически, `initrd` загружается в память загрузчиком и «видится» ядром как устройство `/dev/initrd`. Устройство это блочное, то есть «в первом приближении» — диск, который можно монтировать, что, собственно, ядро и делает, оказавшись в памяти (загрузка `initrd` всегда предшествует загрузке ядра, хотя конфигурационные файлы загрузчиков и намекают на обратный порядок). Далее ядро разархивирует содержимое этого смонтированного диска и переносит его на устройство `/dev/ram0` (или `/dev/rd0` при использовании `devfs`, о чем, похоже, в настоящее время можно уже забыть). Именно `/dev/ram0` и становится корневой файловой системой (по крайней мере, в первой фазе загрузки) при использовании `initrd`.

Стандартное использование `initrd` предполагает на следующем этапе выполнение файла `/linuxrc`, в функции которого входит прежде всего загрузка модулей, необходимых для обеспечения доступа к «настоящей» корневой файловой системе (на устройствах SCSI, SATA, NFS и т.п.), операции настройки, если требуется, и, наконец, монтирование этой самой «настоящей» корневой файловой системы. В отсутствие файла `/linuxrc` ядро запустит `/sbin/init`, как это происходит в случае загрузки без `initrd`. Корневой файловой системой в этом случае останется `/dev/ram0`.

Сообразительный читатель уже, вероятно, догадался, что если `initrd` будет содержать все необходимое для работы системы (в том числе и `/sbin/init`, разумеется), а загрузка будет происходить с CD-ROM, то первая фаза загрузки при стандартном использовании `initrd`, представляет собой нечто иное, как загрузку Live CD. Вот так внедрение `initrd` «между делом» обеспечило возможность существования систем, корневая файловая система которых (полностью или частично) расположена в памяти.

О том, что корневая файловая система всех Live CD располагается в памяти, все уже и так догадались: где же еще можно обеспечить возможность записи, если в вашем распоряжении только read-only CD? Другое дело, что объем памяти вычислительной системы (а значит, и объем `/dev/ram0`) может не соответствовать пожеланиям к количеству и составу доступных для запуска приложений. Тогда придется вспомнить, что, во-первых, целый ряд каталогов корневой файловой системы совершенно не нуждаются в возможности записи в них, а во-вторых, монтировать можно не только к каталогам `/mnt/*` или `/media/*`.

Ничто не мешает иметь в `initrd` «пустыми» такие традиционно объемистые каталоги, как `/opt` или `/usr/X11R6`, чтобы потом монтировать к ним нужные файловые системы. Беда в том, что файловая система CD (iso9660) не всегда удовлетворяет требованиям, предъявляемым к тем же `/opt` или `/usr/X11R6` (ограничения на формат имен файлов, отсутствие необходимых атрибутов и т.п.), но и это преодолимо, если вспомнить о существовании таких замечатель-

ных устройств, как `/dev/loopN`. Для каталога `/opt`, например, определите нужный размер:

```
du -sh opt
создайте файл в соответствии с определенным размером (X — размер в килобайтах):
dd if=/dev/zero of=opt.img bs=Xk count=1
mke2fs -F -m0 -b 1024 opt.img
смонтируйте его, как loopback-устройство:
mount opt.img /mnt/image -o loop
```

Перенесите в `/mnt/image` содержимое `/opt`, и после демонтажа в вашем распоряжении будет файл `opt.img`, которым можно воспользоваться уже при работе с файловой системой Live CD. Просто подмонтируйте его к каталогу `/opt`:

```
mount /mnt/cdrom/img/opt.img /opt -o loop
Каталог /opt, таким образом, принадлежит файловой системе на /dev/ram0, а вот содержимое его находится на CD, в файле cdrom/img/opt.img.
```

Воспользовавшись возможностями файловой системы `cramfs` (compressed ram file system) из состава канонического ядра или `cloop` (compressed loop), как это сделано в Knoppix, файлы-образы каталогов можно еще и сжать. Именно так и получается, что CD емкостью 700 МБ предлагает 2 Гб программного обеспечения.

Вопрос заключается в том, нужно ли это при создании собственного Live CD. Кент Роботти, например, считает это излишним: то, что является необходимостью для `demo-CD`, для `rescue-систем` совсем не обязательно. Не обязательно это и для многих целевых систем: `MoviX` и `GeexBoX`, например, представляющие собой, фактически, мультимедийные центры, также полностью размещаются в памяти. Не будем и мы углубляться в проблемы Live CD, обращающихся к CD-приводу во время своей работы.

Для того, чтобы ядро Live CD «вписывалось» в представленную выше схему загрузки, оно должно поддерживать разделяемые области памяти, RAM-диски, `initrd`, файловые системы `proc` и `tmpfs` и ту файловую систему, на которой создавался `initrd` — не обременительные, в общем-то, требования.

Коснувшись ядра, перейдем к такой его важнейшей функции, как поддержка оборудования. Определение оборудования не является задачей, сколько-нибудь специфичной для Live CD. Современное «дистрибутивостроение», повсеместно отказываясь от `devfs`, а вслед за ним и от `hotplug`, все чаще предлагает пользователю воспользоваться ядром с «универсальным», сравнительно полным набором модулей драйверов, дополнив его, в крайнем случае, `initrd`. Это стало возможным после того, как ISA-устройства вышли из употребления, а вместе с ними отпала нужда и в модулях, способных конфликтовать при попытке обращения к одним и тем же аппаратным ресурсам.

Кроме того, пренебречь наличием в памяти ненужного 50-килобайтного модуля значительно проще, если этой памяти не 16 Мб, как бывало, а 256 Мб, что в настоящее время является нормой. Фактически, планируя собственный Live CD, выбирать приходится между двумя возможностями:

- использовать доступный определитель оборудования (`kudzu` из `RedHat`, `hwdetect` из `Arch Linux`, `hwsetup` из `Knoppix` и т.п.);
- не определять оборудование вообще, положившись на ядро, включающее в себя большую часть драйверов.

Что-то мне подсказывает, что большинство предпочтет второй вариант, хотя, возможно, я и ошибаюсь.

Последним «фундаментальным» вопросом является загрузка Live CD. Если пользователи MS Windows, как правило, плохо представляют, как это происходит, то о линук-

соидах, обычно, так сказать нельзя. Конфигурационные файлы `LILO` и `Grub` вполне читабельны и рано или поздно пользователь Linux узнает, что BIOS сначала загружает в память и запускает на выполнение загрузчик, который, в свою очередь, загружает ядро ОС и, возможно, `initrd`. Тот же загрузчик передает ядру параметры, один из которых имеет незатейливое имя `root` и указывает на местоположение корневой файловой системы. Загрузка Live CD отличается лишь несколькими моментами:

- в качестве загрузчика используются `isolinux` или `grub`, но не `LILO`;
- применение `initrd` обязательно, поскольку, как мы выяснили, «initial RAM disk» и является, собственно, корневой файловой системой;
- из этого следует, что параметр загрузки `root` должен указывать на `/dev/ram0`.

Отказ от `LILO` понятен: этот загрузчик не предназначен для использования с CD. `isolinux`, напротив, именно для этого придуман и долгое время был единственным загрузчиком, использовавшимся при создании Live CD. Начиная с версии 0.95, для загрузки с CD стало возможным использовать `Grub`. Принимая во внимание универсальность этого загрузчика, именно его бы я и предпочел, но у `isolinux` есть одна сильная черта — возможность иметь несколько `help`-файлов (по числу F-клавиш). В остальном оба упомянутых загрузчика равноценны (только в отношении загрузки с CD, разумеется: в широком смысле `Grub` как загрузчик не имеет себе равных).

Реализация

Ознакомившись с изложенными выше принципами, можно перейти непосредственно к RIP и посмотреть, как они реализованы в данном случае. Начнем с `initrd`, представляющего собой сжатый образ корневой файловой системы. Не вдаваясь в детали и следуя рекомендациям Кента, выполните следующие команды:

```
mkdir livecd_root
cd livecd_root
gzip -dc ../rootfs.cgz | cpio -iumdv
```

Предполагается, что каталог, в котором планируется развернуть корневую файловую систему RIP, будет называться `livecd_root`, а сам файл `rootfs.cgz` находится уровнем выше. Понятно, надеюсь, что команда `gzip` просто разархивирует `rootfs.cgz`, а команда `cpio` превращает полученный файл-образ в дерево каталогов.

В полученном дереве каталогов без труда угадывается `Slackware`, впрочем, Кент и сам говорит об этом. Хорошо, что `Slackware`, а не `SUSE` или `Debian`, потому что модификация `Slackware` потребует от пользователя только знания принципов, по которым загружаются все BSD-системы. `/etc/inittab` адресует нас к `/etc/rc.d/rc.S` и `/etc/rc.d/rc.M`, а всю дальнейшую информацию можно получить просматривая эти файлы: спасибо Патрику (`Patrick Wolkering` — автор `Slackware`) за исчерпывающие комментарии, спасибо Кенту за его выбор.

Для осуществления обратной операции (создания сжатого образа из дерева каталогов) выполните команды:

```
cd livecd_root
find . | bin/cpio -v -o -H newc | gzip -9 > ../rootfs.cgz
```

Использование `find` вызвано тем, что для архивирования образа нужно использовать не тот `cpio`, что имеется в вашей системе, а тот, который входит в состав RIP.

Как компилировалось ядро из состава RIP, можно узнать, познакомясь с содержимым файла `kernel.txt`. Отметить можно следующее:

- ядро предполагает поддержку CPU, известных как "generic x86";

- не поддерживаются Software Suspend, APM и CPU Frequency scaling (управление частотой CPU), но поддерживается ACPI;
- не поддерживаются PCI Express и PCI Hotplug, но поддерживаются ISA и PCMCIA/CardBus;
- поддерживаются сокеты Unix и TCP/IP, но не поддерживаются IrDA, Bluetooth и Wi-Fi;
- поддерживаются практически все известные ядру 10, 100 и 1000 Mbit сетевые адаптеры;
- поддерживаются практически все блочные устройства (IDE, включая SATA, SCSI, RAID и LVM, FireWire);
- не поддерживаются ISDN, Sound, Hardware Monitoring и Multimedia devices;
- поддерживаются порт PS/2 и мышь, но не поддерживаются Touchscreens и Joysticks;
- поддерживаются практически все известные ядру файловые системы;
- поддерживается USB.

На мой взгляд, приведенный список как раз и соответствует задачам, которые предстоит решать rescue-cd. Если же ваши потребности существенно отличаются, то ничто не мешает использовать в составе Live CD свое собственное ядро.

При сборке ядра Кент полностью отказался от загружаемых модулей: ядро при этом «выросло» до 3-х мегабайт, зато отпала необходимость в определении оборудования и, соответственно, подзагрузке модулей. Решение, быть может, не оптимальное, но для Live CD, на мой взгляд, вполне приемлемое.

Осталось выяснить, как это все загружается. ISO-образы RIP, которые можно взять на <http://www.tux.org/pub/people/kent-robotti/looplinux/rip/>, используют *Grub*, хотя еще совсем недавно в качестве загрузчика выступал *isolinux*. В настоящее время в состав RIP входят и *Grub*, и *isolinux*, а скрипт *mkiso.sh* создает iso-образ с использованием того загрузчика, который указан в качестве первого параметра. Для создания собственного Live CD создайте каталог, содержащий подкаталог **boot**, а в нем – подкаталог **grub**. В подкаталог **boot** поместите файлы **kernel** и **rootfs.cgz**, в подкаталог **grub** – файлы из одноименного подкаталога исходного диска RIP. Если имя созданного каталога **rip-iso**, то запуск *mkiso.sh*, в каталоге уровнем выше **rip-iso** выполните так:

```
sh mkiso.sh grub rip-iso . rip_grub.iso
```

В результате файл **rip_grub.iso** будет создан в текущем каталоге.

Для *isolinux* вместо подкаталога **grub** в **boot** создайте подкаталог **isolinux**. Разумеется, первым параметром для *mkiso.sh* в данном случае также будет **isolinux**, а имя создаваемого файла логично будет изменить на **rip_isolinux.iso**.

Запись диска можно осуществить любым доступным способом, а при желании, даже непосредственно из RIP,

поскольку последний содержит и *mkisofs*, и *cdrecord* (и *dvdrwrecord* с *cdwrtool*, впридачу).

Практикум

Какие же изменения стоит внести в оригинальный RIP, чтобы он в большей мере соответствовал вашим потребностям?

Любителям использовать мышь можно рекомендовать «включить» *gpm*. Для этого, в соответствии с правилами Slackware, файл **/etc/rc.d/rc.gpm** сделайте исполняемым:

```
chmod +x etc/rc.d/rc.gpm
```

Предполагается, что команда издается в созданном нами каталоге **livecd_root**. Поскольку **rc.gpm** ссылается на **/dev/mouse**, то в каталоге **/dev** создайте символическую ссылку:

```
ln -s input/mouse0 mouse
```

Исполняемым также стоит сделать файл **/etc/rc.d/rc.font** – это даст возможность после загрузки шрифта с кириллицей читать на родном языке. Шрифт, разумеется, также можно установить свой. Для этого переписите в каталог **/usr/share/kbd/consolefonts/** свой любимый консольный шрифт (рекомендую *terminus*), и внесите соответствующие изменения в **/etc/rc.d/rc.font**. Например:

```
setfont -v ter-k16f -m koi8-r_to_uni
```

При наличии кириллицы, файл **/etc/issue** имеет смысл заменить на свой собственный. Содержимое этого файла выводится перед приглашением «login».

Приверженцам *Midnight Commander* можно посоветовать сохранить файл **~/mc/ini** в каталоге **/root/.mc/**. Тогда *mc* в составе RIP унаследует ваши личные настройки.

Если предполагается использовать Live CD для подключения к удаленным NFS или SMB ресурсам, не забудьте загрузить *portmap* (сделайте исполняемым **/etc/rc.d/rc.portmap**). Всем, кто не очень знаком со Slackware, вообще стоит изучить содержимое каталога **/etc/rc.d/** – скрипты Патрика отлично прокомментированы и вы, возможно, узнаете из них нечто полезное.

Существенно изменит RIP замена ядра системы на свое собственное. Мне, например, кажется не лишним включение поддержки framebuffer (**CONFIG_FB=y**), драйвера *vesa* (**CONFIG_FB_VESA=y**) и возможности переключения видеорежимов (**CONFIG_VIDEO_SELECT=y**), поскольку консоль *vga* (640x480) на 17..19 дюймовых LCD мониторах выглядит изрядным анахронизмом.

Обратите внимание, что, отказавшись от ядра Кента Роботти в пользу канонического, вы лишитесь поддержки некоторых контроллеров PATA, REISER4 и SQUASHFS, поскольку эти опции – результат наложения независимых патчей. Впрочем, это относится только к RIP-0.15 и ядру 2.6.16 – для другой версии и другого ядра отличия могут быть иными.

Разумеется, вы можете установить в RIP пакеты Slackware. В общем – делать все, что вы обычно делаете со своей «основной» системой. Проверить работоспособность

добавленного приложения можно, даже не переписывая Live CD. Выполните

```
chroot livecd_root
```

и вашей корневой файловой системой (вплоть до *exit*) станет файловая система будущего Live CD.

И немного об использовании...

В заключение, несколько реверансов в адрес Кента Роботти, которые, быть может, обратят ваше внимание на содержимое RIP.

Нет смысла, как ни странно, обсуждать средства восстановления данных из состава RIP – тема достойна отдельного разговора. Ограничимся констатацией факта, что список этот достаточно полон, всегда в актуальном состоянии и не ограничивается миром Linux, вплоть до утилит модификации NTFS-разделов и изменения паролей CMOS/BIOS и Windows NT/XP SAM.

Последняя версия пакета *ntfsprog* обеспечит монтирование с возможностью записи разделов NTFS, утилита *ntfsresize*, как вы уже догадались, позволит безопасно менять размеры таких разделов, а *ntfscione* попытается создавать их копии (вопреки утверждению Microsoft о невозможности этого). Одним словом, интересующимся восстановлением данных стоит познакомиться с опытом Кента.

Несколько слов по поводу *Grub*. Мало того, что Кент Роботти всегда использует самую современную версию этого загрузчика, так она у него еще и «усилена» возможностью чтения с разделов Reiser4 и NTFS, дополнена утилитами использования под DOS/Windows и конфигуратором собственного производства.

Не оставлены без внимания и *ldlinux/isolinux*: скрипт и подробные инструкции по поводу того, как создать RIP на USB-накопителе – к вашим услугам.

Особого упоминания заслуживает документация в составе RIP. Написанного собственно Кентом не так и много, но содержимое каталога **rootfs/usr/doc** прекрасно дополняет комплект *man*, который также присутствует в RIP. Именно так: 30-мегабайтный RIP включает в себя **man**-ы всех используемых программ плюс документацию в формате plain text для тех случаев, когда *man* приложения отсутствует или недостаточен. Известная фраза «Изучая Debian, ты изучаешь Debian. Изучая Slackware, ты изучаешь Linux», как нельзя лучше подходит к RIP, документация которого – маленькая энциклопедия Linux.

Что касается ядра в составе RIP, то оно хоть и не «на все случаи жизни», но может быть очень полезным. Как определяет конфигурацию вашего оборудования последняя стабильная версия? Не появилась ли в ядре поддержка адаптера, которой вы уже давно ждете? Не улучшилась ли поддержка SATA-винчестеров? Воспользовавшись последней версией RIP и возможностями *sysfs*, все это можно выяснить, не прибегая к сборке нового ядра. Для кого-то это может быть заманчивым... **LXF**

Recovery is possible

Учебники >>

Наши эксперты помогут вам с любым приложением Linux

А ВАМ КУДА?



Евгений Балдин
Цитирует belgorod.lug:
«Чем бы дети ни
тешились, лишь бы
GPL не нарушалось.»

GNU/Linux идёт на рабочий стол. А куда ещё ему деваться? Появляются «завлекалки» в виде XGL, решается вопрос всех времён и народов: «Кто есть простой пользователь и чего его душевнее угодно?»

Потихоньку начинают прорезаться недоуменные голоса и другого рода. Джим Матцан (Jim Matzan) написал эссе на тему, что GNU/Linux явно движется куда-то не туда (Linux desktop distributions are headed in the wrong direction), а именно в сторону «визуальщины», а не в сторону технологичности. Так ли это страшно?

Вряд ли. Просто набралась критическая масса тех, кому интересна именно визуальность. Они бы в любом случае не улучшали технологичность. Вполне нормально, когда люди занимаются тем, что им нравится. Остаётся только проблема, как совместить эти две культуры на одной базе.

Очень просто. Не нравится — не смотри. Если человек давно освоился в окружении GNU/Linux, то набор подходящих команд и приёмов у него уже есть — нет необходимости пересекаться с программами противоположного лагеря. Единственная проблема — это сложность выбора окружения при первой установке, пока ещё не знаешь, что конкретно необходимо.

Я предлагаю для таких случаев внедрить ещё один способ начальной установки, когда пользователь не выбирает программы, а отвечает на отвлечённые вопросы вида: «Вам шашечки или ехать?», или «Насколько быстро Вы набираете текст? Можете ли Вы набирать вслепую?». Подобный приём часто применяется в текстовых RPG для выбора персонажа. Говорят, что в Linux мало игр — но он сам может быть не менее интересен и уж заведомо более полезен.

В ЭТОМ ВЫПУСКЕ МЫ ИЗУЧИМ:

52 ПЕРВЫЕ ШАГИ

Энди Чэннел настраивает Ubuntu для своей дочери

56 INKSCAPE

Анимация в векторно-графическом стиле: Дмитрий Кирсанов расскажет подробности

60 OOO BASIC

Марк Бейн снова надоедает вам со своими советами

64 LVM

Разбейте свой жесткий диск с помощью Нейла Ботвика

68 XDMCP ЗА 10 МИНУТ

Тонкие клиенты с доставкой на дом в исполнении Пола Хадсона

70 3D-ИГРЫ

Пол Хадсон программирует перемещение и улучшает ландшафт

74 HARDSCORE LINUX

RAM: именно так компьютер Грэма Моррисона узнает своего хозяина

78 PYTHON PRO

Это можно делать не только на PHP! Сергей Супрунов разрабатывает гостевую книгу

82 QT/KDE, ЧАСТЬ 6

Сегодня Андрей Боровский расскажет, чем настоящие KDE-приложения отличаются от «почти настоящих»



56 Танцы маленьких векторов

86 МАХИМА, ЧАСТЬ 3

Тихон Тарнавский — о том, как Maxima может упростить вашу жизнь... и алгебраические выражения

91 АНАЛИЗ ДАННЫХ, ROOT

Однажды команде PAW стало скучно... Евгений Балдин изучает, что из этого вышло

96 UNIX API, ЧАСТЬ 3

Сокеты — одна из наиболее удачных находок Unix, портированная почти во все ОС. Андрей Боровский покажет, как с ними работать.

100 LaTeX

Евгений Балдин начинает серию статей о культовой системе компьютерной верстки

104 WINK

Видеопособие — это просто! Научитесь делать их вместе с Сергеем Яремчуком

НОВАЯ СЕРИЯ

СОВЕТ МЕСЯЦА

Мы часто сталкиваемся с проблемой иллюстраций для игровых обзоров, подобных Cold War на стр. 8. Игры захватывают клавиатуру и дисплей и, если разработчики не предусмотрели специальной встроенной функции, получить снимок экрана не так-то просто. В некоторых играх (в том числе, Cold War) поддерживается оконный режим, но вам все равно необходимо каким-то образом отобрать у нее клавиатуру и передать управление рабочему окружению, прежде чем вы сможете использовать утилиты Gnome или KDE.

СНИМОК ЭКРАНА

Решение существует! Несмотря на то, что доступ к рабочему столу может быть закрыт, в вашем распоряжении по-прежнему остаются другие виртуальные терминалы. Нажмите **Ctrl+Alt+F1** — вы перенесетесь на текстовый экран с приглашением войти в систему. Это наследие тех времен, когда Unix использовался преимущественно как многопользовательская ОС; термин «виртуальный» отмечает тот факт, что все терминалы, на самом деле, работают на одной машине. К другим терминалам можно перейти, заменив **F1** на **F2-F6**, а вернуться к вашему рабочему столу

позволит комбинация **Ctrl+Alt+F7**.

Какое отношение все это имеет к экранному снимкам? Если вы добрались до командной строки, вы можете запечатлеть экран с помощью одной из многочисленных утилит пакета **ImageMagick**. Попробуйте следующую команду:

```
chvt 7;sleep 10;import -display :0.0  
-window root image.png
```

Она переключит вас на 7 терминал (chvt 7), подождет 10 секунд и затем сохранит экран в файл **image.png** с помощью команды **import**. Готово!



ПЕРВЫЕ ШАГИ. СЕРИЯ «LINUX ДЛЯ НОВИЧКОВ»

Ubuntu Рабочий стол для детей

Ох уж эти дети! Они любопытны, наивны и, увы, склонны к вредительству. Энди Ченнел расскажет, как оградить детей от Интернета, а ваш компьютер – от детей, сконструировав для них отдельный рабочий стол.

МЕСЯЦ НАЗАД



Я показывал, как работать с приложениями и документами намного быстрее.



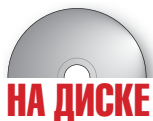
Всех нас греет мысль, что наши Linux-системы гораздо безопаснее, чем любые Windows-компьютеры или Mac. Хакерам нелегко взломать Linux, но даже если это им удастся, большой беды не будет, благодаря эффективному контролю доступа к приложениям. Однако одним аспектом безопасности в Linux некоторое время пренебрегали, по сравнению с другими ОС: речь идет о родительском контроле.

На наше счастье, поставщики дистрибутивов дальновиднее сисадминов и хакеров, и сейчас они взялись за этот вопрос. На этом уроке мы воспользуемся новыми технологиями для создания модифицированного рабочего стола специально для детей. Мы организуем дружелюбное рабочее место, в котором ребята смогут запросто посещать любимые web-сай-

ты и запускать программы; а вы сможете быть уверены, что они не угодят в сомнительные закоулки Интернета и не угрожат операционную систему. Мы добавим в систему новых пользователей, настроим их права, запретим доступ к локальным устройствам, а также создадим ярлыки сайтов на рабочем столе и настроим фильтры для работы в Интернете.

Я использовал последнюю версию Ubuntu (6.06, есть на нашем диске), но все описанные действия и приемы вы можете также применить и к любому другому настольному дистрибутиву Linux. Родственный Ubuntu проект Edubuntu располагает специальными инструментами, которые могут запретить ученикам в школах доступ к определенным web-сайтам. Кое-какие из этих инструментов просочились и в основной репозиторий Ubuntu – они-то нам и понадобятся.

ШАГ 1 – ДАЙТЕ ИМ ПРОСТОР



НА ДИСКЕ

- Ubuntu 6.06
- Willow 3.13

Первым делом нужно выделить детям пользовательское пространство на диске. В Ubuntu это делается через редактор учетных записей пользователей (User Account Editor), который вы найдете в меню System > Administration > Users and Groups. В окне редактора есть поля для названия учетной записи, полного имени пользователя, пароля и других данных.

Заполнив все поля, вы можете настроить «привилегии пользователей», которые логичнее было бы назвать «запреты пользователям», потому что мы собираемся именно запрещать. Проще всего это сделать, выбрав на вкладке User Privileges те опции, которые мы хотим запретить. Очевидно, что нужно сразу закрыть доступ к администрированию системы (Execute System Administration Tasks), а также запретить пользователю добавлять в систему принтеры, монтировать дискеты и посылать факсы. Не стоит недооценивать способности детей к пользованию компьютером – итог может быть плачевным: а) потеряны резуль-

тат многочасовой работы и важные настройки; б) вы выглядите идиотом перед коллегами или в) диск забит всяким хламом. Отметив нужные пункты и удовлетворившись результатом, жмите ОК. Создастся новый пользователь со своей директорией в /home, к которой будут применены сделанные нами ограничения.

Если вам нужно настроить учетные записи для большого числа ребят – например, в школе – пригодится средняя вкладка Advanced в окне Users and Groups: с ее помощью создаются профили, которые можно одним щелчком применить к создаваемым или уже существующим пользователям. Этот инструмент предоставляет все рассмотренные средства для ограничения прав пользователей. Можно также настроить права группы и использовать вкладку User Privileges для подстройки прав отдельных пользователей. Например, можно разрешить доступ к iPod'ам подросткам и запретить его детям младшего возраста. Понятно, что все эти изменения возможны только при наличии у вас



Сделав учетную запись каждому потомку, вы предотвратите ссоры между ними насчет того, чья картинка будет фоном рабочего стола.

администраторских привилегий, поэтому тщательно скрывайте свой пароль от детей - иначе грош цена всем ограничениям! Другая часть настроек в User and Groups касается, естественно, управления группами, хотя здесь настройки будут посложнее. Но мне кажется, неплохо будет создать отдельную группу под названием Kids (Дети) эксклюзивно для ребят и разрешать доступ к определенным папкам сразу целой группе, а не каждому пользователю по отдельности.

САМЫЕ ЧАДОЛЮБИВЫЕ ПРОГРАММЫ

Все они доступны через менеджер пакетов вашего компьютера или в Интернете.

- **GCompris** – Дедушка развлекательно-обучающих программ для Linux. Включает огромную подборку игр для обучения детей работе на компьютере и знакомству с цифрами и буквами, головоломки, логические игры и многое другое...
- **Tuxpaint** – Чудесный графический редактор с открытым исходным кодом. Поставляется с отличной коллекцией картинок вроде монет, планет, пингвинов, которые можно добавить к своим рисункам.
- **Childsplay** – Новый конкурент GCompris, позволяющий, по словам разработчиков, общаться с

компьютером «и граучи». Проект пока еще совсем молод.

• **GnuChess** – Если уж Джон Стюарт Милл мог говорить по-латыни в возрасте трех лет, то шести- и семилетки точно в состоянии научиться играть в шахматы. Это отличная программа.

• **Zsnes** – Если вы, как и домочадцы Энди Ченнела, только что заново открыли для себя радости SNES, Zsnes (на картинке) станет прекрасным открытым эмулятором для запуска ваших любимых игр в Linux.



ШАГ 2 – РАСЧИЩАЕМ РАБОЧИЙ СТОЛ

Настроив учетные записи, займемся индивидуальными рабочими столами пользователей. Большинство параметров может быть изменено «владельцами» учетных записей – пусть наслаждаются возможностью сменить фон рабочего стола, выбрать хранитель экрана по своему вкусу и т.п. Но прежде чем дать деткам волю, лучше кое-что подправить.

Во-первых, хотя стандартный рабочий стол Ubuntu очень лаконичен, он содержит значки подключенных дисков, а это нам ни к чему. Чтобы убрать их, запустите из терминала (Applications > Accessories > Terminal) аналог реестра Windows – *gconf-editor*. Теперь перейдите к разделу Apps > Nautilus > Desktop, где имеется список разных опций. Снимите галочку с ключа Volumes_visible (значки устройств пропадут) и поставьте галочки напротив ключей Home_icon_visible и Trash_icon_visible, чтобы добавить соответствующие иконки на рабочий стол. Для быстрого доступа к дисковым разделам, CD и прочему, включите отображение иконки «Компьютер» – аналога «Моего компьютера» от Microsoft (отметьте галочкой ключ Computer_visible). Конечно, можно оставить рабочий стол вообще пустым, но согласитесь, что быстрый доступ к файлам и Корзине пригодится. Теперь *gconf-editor* можно закрыть (File > Quit).

В Ubuntu (и других дистрибутивах на базе Gnome – например, Fedora Core) можно полностью переделать структуру меню с помощью редактора *Alacarte Menu Editor* (Applications > Accessories > Alacarte Menu Editor). Но я готовлю рабочий стол для девочки, которой нужно всего несколько приложений (web-браузер, пара игр и музыкальный проигрыватель), и будет логично создать нижнюю панель в стиле док OS X и поместить туда значки этих программ, тогда пользователь сможет

запускать их одним щелчком. Туда же добавим апплет списка окон (для переключения между приложениями) и часы с календарем. Гnome позволяет это сделать очень легко – потребуется лишь несколько раз щелкнуть правой кнопкой мыши и выбрать нужные пункты меню или значки.

Сперва поменяем расположение панелей, по умолчанию расположенных вдоль верхней и нижней границы рабочего стола Gnome. Избавимся от нижней панели, щелкнув правой кнопкой мыши по свободному пространству в пределах панели и выбрав пункт **Delete This Panel...** Теперь щелкните правой кнопкой мыши по верхней панели и выберите **Properties (Свойства)**. Так как нижней панели больше нет, можно перетащить на ее место верхнюю (содержащую кое-какие полезные штуки). Измените значение пункта **Orientation** на **Bottom (низ)** и сделайте панель побольше, установив ее размер равным 48 пикселям (ребенку будет проще попасть по значку).

Вы можете сделать панель более стильной, щелкнув на вкладке **Background (Фон)**, выбрав цвет и установив режим полупрозрачности. Хорошей идеей будет также увеличить размер шрифта в подписях иконок на рабочем столе, чтобы ребенку было легче их читать. Для этого перейдите к пункту **System > Preferences > Font**, где регулируется размер и начертание шрифта для значков рабочего стола. Не рекомендую увеличивать кегль пунктов меню: крупный шрифт мешает работе в приложениях.

Вернемся к панели: можно настроить ее на автоматическое свертывание за нижний край экрана (пункт **Autohide**), чтобы максимально использовать рабочее пространство; однако по моим личным наблюдениям, это особой пользы не приносит и лишь затрудняет семилетнему ребенку общение с компьютером.

ШАГ 3 – ДОБАВЛЯЕМ ДРУЖЕЛЮБНЫЕ ДЕТЯМ ЭЛЕМЕНТЫ

Пора добавить нужные элементы на панель. Для начала, щелкните правой кнопкой мыши по свободному месту на панели и выберите пункт **Add to panel...** Вы увидите внушительный список апплетов, которые можно разместить на ней. Найдите в самом верху пункт **Application Launcher (Запуск приложений...)**. Как вы уже заметили, этот пункт предоставляет список всех установленных приложений, которые ранее были доступны из главного меню; теперь просто найдите

нужное вам приложение. Я начну с *Firefox*, который находится в разделе **Internet**. Выберите его значок и нажмите кнопку **Add (Добавить)**. На панели появится новый элемент, по щелчку на котором будет запускаться web-браузер. Щелкните правой кнопкой мыши по значку для вызова контекстно-зависимого меню. Перейдите в свойства значка (**Properties**). Здесь мы можем изменить иконку и команду запуска.

Заменим стандартную иконку программы на более «детскую».





Настроить *Firefox* на нужный сайт очень просто: укажите ссылку и смените иконку.



Ubuntu поставляется с огромным количеством иконок, да к тому же их полно на сайте www.gnome-look.org. Темы иконок поставляются в виде архивов с расширением ***.tar.gz**, которые можно прямиком пере-

тащить в окно настройки тем (System > Preferences > Theme). Нажав на кнопку **Theme Details** (Подробнее о теме), мы можем изменить тему иконок глобально, однако сейчас нас интересует наша конкретная иконка. Поэтому в окне свойств значка щелкните по его иконке (по умолчанию в Ubuntu для *Firefox* установлен синий глобус) и выберите другое изображение из числа представленных в списке, либо нажмите **Browse** (Обзор) для выбора графического файла.

Большинство иконок Gnome хранятся в директории **/usr/share/icons/имя_темы**, и вы сможете увидеть их список при нажатии кнопки **Open** (Открыть). Обычно иконки представлены в нескольких размерах – выберите размер, который ближе всех к высоте вашей панели.

Как правило, все изменения в настройке Gnome применяются в реальном времени, поэтому выбранная иконка сразу же появится на экране. В качестве иконки можно использовать любой PNG-файл, чтобы оформить рабочее место на свой вкус – очень удобно!

Через окно свойств значка можно изменять также и команду запуска приложения – можно создать, к примеру, несколько значков веб-браузера (с разными иконками) для запуска разных сайтов. Для этого добавьте на панель, как было описано выше, еще один значок Firefox и откройте его свойства. В разделе **Command** (по умолчанию там написано **firefox %u**) добавьте URL сайта, заключив его в кавычки.

```
firefox %u "http://www.bbc.co.uk/cbeebies/"
```

Присвойте значку другую иконку, ассоциирующуюся с сайтом, и теперь даже трех- или четырехлетний человек с мышью сможет с минимальной помощью загрузить любимый сайт.

ШАГ 4 – НЕОБХОДИМЫЕ ОГРАНИЧЕНИЯ

Теперь нужно позаботиться, чтобы ребенок (или дети) сидел себе в своей песочнице, и не тянулся своими замурзанными пальчиками к информации других пользователей. На данном этапе закроем доступ к местам, куда дети соваться не должны. Работа в Linux основана на системе прав, определяющих, кто может заходить в определенные директории, и что можно в них делать.

«БЕРЕГИТЕ ВАЖНЫЕ ДАННЫЕ ОТ ЗАМУРЗАННЫХ ДЕТСКИХ ПАЛЬЧИКОВ.»

Чтобы не пускать детей в директорию **/home**, щелкните правой кнопкой мыши по домашней папке (Ubuntu автоматически присваивает

ей имя пользователя), выберите пункт **Properties** (Свойства) и перейдите на вкладку **Permissions** (Права). После строк, указывающих на владельца файла и группу, идет таблица 3x3, позволяющая указать, кто имеет доступ к директории, может сохранять в ней изменения и запускать программы. Чтобы оставить все права только за собой, снимите галочки со всех пунктов, кроме тех, что относятся к вам (**Owner**). После этого другие пользователи не просто не смогут войти в вашу директорию – для них она как будто не будет существовать.

Конечно же, вы можете настроить права ваших детей таким образом, что сможете просматривать их файлы так, как захотите.

Чтобы упростить юным пользователям обмен документами, я рекомендую создать еще одного пользователя, назвав его **Share** (или как-то вроде этого) и сделать его домашнюю директорию общедоступной, поставив все галочки в таблице прав. В дальнейшем я буду хранить в этой директории фотографии, музыку и фильмы для общего доступа, чтобы каждый член семьи мог их посмотреть и послушать.

ШАГ 5 – СОЗДАЕМ ГИБКИЕ ФИЛЬТРЫ ДЛЯ РАБОТЫ В ИНТЕРНЕТЕ

Запретить доступ к локальным папкам несложно – гораздо сложнее ограничить доступ к определенным сайтам в Интернете. Тем не менее, в нашем распоряжении есть две системы под Linux, предоставляющие родителям дополнительный контроль над сайтами, которые детям разрешено посещать. Это **DansGuardion** (<http://dansguardian.org>) и **Willow** (www.digitalnumber.com/willow). Первая программа очень хороша, но она подразумевает опыт работы с прокси-серверами (в частности, со *Squid*), и хотя разработчики постарались облегчить установку и настройку программы, она не слишком подходит для начинающих или случайных пользователей. Если вам все же хочется ей воспользоваться, читайте статью Марко Фиоретти [Marco Fioretti] в разделе **Hardcore Linux** в *LXF81*.

С *Willow* все наоборот. Программа доступна в виде пакета для Ubuntu, устанавливается автоматически, не требует знания прокси и очень проста в использовании. Я взял пакет для Ubuntu с <http://nanomad.altervista.org> (а вы можете сэкономить трафик и устано-

вить программу с нашего диска), и он без проблем установился через стандартный менеджер пакетов дистрибутива. По сути, вся установка свелась к скачиванию пакета и нажатию на кнопку **Install Package** (Установить пакет).

Установщик закрывает все окна *Firefox*, и когда браузер загрузится вновь, он будет работать уже через фильтр. Для установки ограничений разработчики тестировали программу на начальной школе в США, поэтому можно считать, что стандартные настройки фильтра подойдут для ребенка от пяти до восьми лет.

Фильтр работает по байесовскому принципу, то есть анализирует визуальное оформление и текст сайта, а затем сравнивает его с образцами, которые уже были отмечены ранее как допустимые или недопустимые. Это лучше, чем система черных и белых списков (для запрещенных и разрешенных сайтов соответственно), поскольку ни один список не в силах уследить за развитием Интернета. Тем не менее, у фильтра иногда случаются ложные срабатывания, и иногда необходимо слегка



Опытному пользователю легко отключить Willow, но младшим детям это вряд ли удастся.

подправить настройки фильтра.

Для начала позаботимся, чтобы у каждого пользователя браузер запускался через фильтр. Для этого зайдите в систему под учетной записью ребенка, запустите браузер и перейдите в окно его настроек (Edit > Preferences). Во вкладке General (Основные) щелкните по кнопке Connection Settings (Параметры соединения). Обычно здесь настроено только прямое подключение к интернету, но нам нужен фильтр Willow, поэтому щелкните по переключателю Manual Proxy Configuration (Настройка прокси вручную) и введите адрес HTTP-прокси – localhost. Номер порта должен быть 8000, а пункт «Использовать этот прокси-сервер для всех протоколов» – отмеченным. Теперь, когда кто-нибудь попытается зайти на сайт, запрещенный с точки зрения Willow, браузер отобразит «объяснительную» страницу, оформленную в красивых оттенках желтого.

Если вы разбираетесь в ручной правке HTML, можете сделать свою страницу, появляющуюся вместо «запрещенного» сайта, изменив конфигурационные файлы. Это делается от имени суперпользователя при помощи такой команды:

```
gedit /usr/share/Willow/domainfilter.py
```

Отредактировать настройки не очень просто, хотя мелкая правка, вроде исправления опечатки в слове «innappropriate» на «запрещенной» странице, не составит труда. У меня прекрасно получилась блокировка целого ряда сайтов «для взрослых», причем мудрый Willow позволяет при этом открывать сайты о раке груди или о роли нацистов во Второй мировой войне. Бывает, что фильтр и ошибается; в таких случаях его надо «учить». Например, настройки по умолчанию заблокировали доступ к www.gnome-look.org (материалы по изменению внешнего вида Gnome), и этот портал пришлось вручную добавить в «белый» список.

Чтобы открыть (весьма скупой) интерфейс Willow, перейдите в меню к System > Preferences > Willow Administration Interface, введите имя root и пароль Willowadmin. Эти параметры пока нельзя изменить через графический конфигуратор, тем не менее, известно, что программист Willow Трэвис Уоткинс (Travis Watkins) получил в конкурсе Google Summer Of Code грант на доработку графического интерфейса к Willow, поэтому следите за домашней страничкой проекта – в течение ближайших месяцев на ней могут появиться важные обновления. Пока что придется обходиться существующими настройками.

Ссылка Filter приведет вас к странице с тремя параметрами: фильтрация конкретных URL (блокирование/доступ), фильтрация доменов (то же, но применительно к целым сайтам) и, что наиболее интересно, фильтрация по контенту. В последнем случае вы задаете URL страницы, которую Willow пропустит через байесовский алгоритм, после чего будет лучше знать ваш нрав. От вас требуется всего лишь вставить URL в строку good (хороший) или bad (плохой) и нажать кнопку Apply Changes (Применить изменения). Таким образом, вы можете создать фильтр по своему вкусу и не зависеть от политических и этических взглядов посторонней организации или человека.

Willow нельзя назвать самой безопасной системой фильтрования, потому что ничто не мешает пользователю зайти в настройки соединения Firefox и отключить фильтрацию. Однако в данном случае нам важно предотвратить непреднамеренное попадание на некоторые сайты, а абсолютная защищенность браузера не столь важна. Так что для наших целей Willow подходит идеально.

О ПОЯСНЕНИЯХ

Советы, изложенные в нашем руководстве, обезопасят ваших детей и ваш компьютер – но лишь до поры до времени. Упорный и активно осваивающий компьютер тинэйджер вполне способен обойти поставленные вами ограничения, но это не повод пускать все на самотек. Поставив компьютер в место, где ходит вся семья, вы сильно сократите тайные эксперименты с Интернетом, но очень важно (как и вообще в воспитании) строить отношения с ребенком на доверии и уважении, чтобы дети не стеснялись приходить к вам, если, например, они что-то намудрили с браузером. В частности, не скрывайте от детей, какие вы принимаете меры и почему вы считаете их важными.

ШАГ 6 – БЛОКИРОВКА РЕКЛАМЫ НА САЙТАХ

Последнее, что нас волнует – это интернет-реклама. Да, многие сайты не будут правильно работать без рекламы, но они, как правило, не для детей, хоть и имеют завлекательный вид: тут могут быть ссылки на сайты мошенников, казино и брачные порталы. В составе Firefox имеется очень грамотный фильтр всплывающих сообщений, но вы можете установить расширение Adblock и заблокировать даже ту рекламу, с которой не справится обычный фильтр.

Adblock можно установить через обычную систему расширений браузера (Tools > Extensions > Get More Extensions), загружающую файлы с сайта Firefox. Это расширение хорошо себя зарекомендовало за гибкость в фильтровании рекламы. Вы можете установить фильтрацию либо по содержанию рекламы (например, маска .gif заблокирует все GIF-файлы), либо по ее родительскому сайту (если вам нужно отсеять весь контент с сайта DoubleClick, используйте маску doubleclick.net).

Установленный Adblock доступен в виде значка в нижней части окна браузера. Щелчок по надписи Adblock покажет список элементов на текущей странице, поддающихся блокировке. Выберите один из них и нажмите OK, после чего на основании вашего выбора включится фильтрация. Правый щелчок по значку вызывает контекстно-зависимое меню с вездесущим пунктом Preferences. В настройках можно менять параметры фильтров, упорядочить блокировку flash-баннеров и составить «белый» список разрешенных страниц или сайтов. Последняя опция наиболее интересна, так как она позволяет контролировать, какую рекламу смотреть вам или вашим детям. Если вы, например, хотите поддержать сайт, который зарабатывает на жизнь рекламой – скажем, www.linuxformat.co.uk, а также знаете, что некоторые баннеры действительно полезны и



Перед вами – Настоящий рабочий стол Принцессы. Или программистки, если она захочет таковой стать...

информативны, вы можете зайти на страницу с этими баннерами, щелкнуть правой кнопкой мыши на значке Adblock и выбрать Whitelist This Page (Добавить эту страницу в «белый» список) или Whitelist This Whole Site (Добавить весь сайт в «белый» список).

Дети и компьютеры отлично ладят друг с другом, и с помощью системы настройки прав в Linux, «умных» открытых программ, а также здравого смысла, вы можете спокойно оставить детей наедине с Linux-компьютером и не бояться, что они сотрут важные документы или скачают 50 гигабайт всякой дряни. LXF

ЧЕРЕЗ МЕСЯЦ

Базами данных обычно занимаются администраторы, но они могут оказаться полезными и простым смертным.



ПРАКТИЧЕСКИЕ ЗАНЯТИЯ ВЕКТОРНАЯ ГРАФИКА

Inkscape Оживите свой рисунок

ЧАСТЬ 2 Дмитрий Кирсанов покажет, как создать простую GIF-анимацию. Скажем, аватар форума LXF?

**МЕСЯЦ
НАЗАД**



Я представил вам образец визитной карточки 101.



Наверное, вы подумали: «Дмитрий, мы знаем вас как поборника *Inkscape*, но вы же не хотите сказать, что это эквивалент Flash под Linux?». Ну, не совсем. Но ведь язык SVG изначально создавался как для статической, так и для анимированной векторной графики. SVG умеет создавать самые разные виды анимации – когда-то его даже называли «Flash killer». К сожалению, даже сейчас, спустя несколько лет со времени создания SVG, в сети очень мало SVG-анимаций – большинство выполнены в форматах Flash или анимированных GIF. Тому есть несколько причин, и главная из них, вероятно, отсутствие стандартного плеера, способного составить конкуренцию *Macromedia Flash*.

Очень жаль: лёгкость создания и редактирования объектов, чудесное Каллиграфическое перо и набор векторных инструментов на все случаи жизни делают *Inkscape* идеальным инструментом для создания большинства анимаций: баннеров, аватар и заглавий блогов. Создавать анимированные SVG (бегающие строки, плавные переходы и др.) *Inkscape*

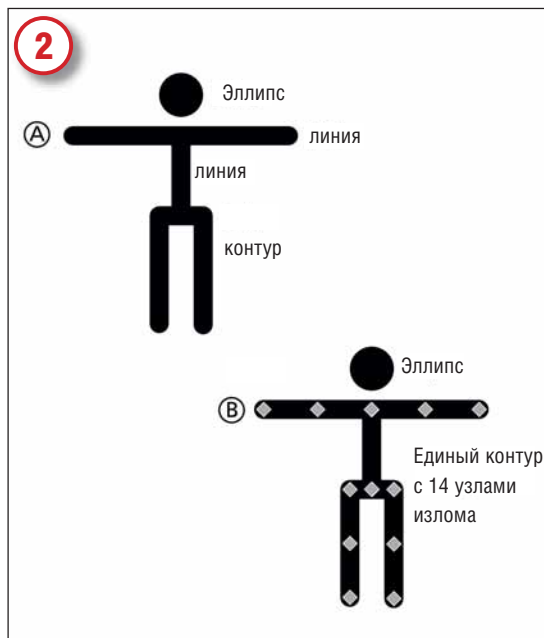
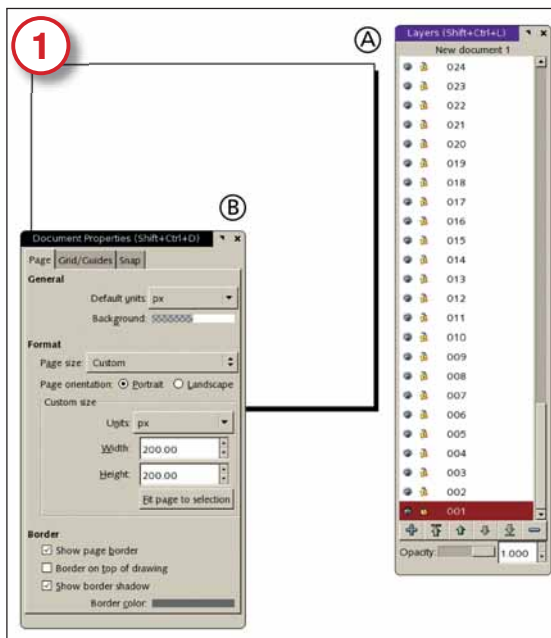
умел с первого дня, только это умение до сих пор не применялось. Но это не значит, что вы не можете создать в *Inkscape* статические рисунки, а затем соорудить из них анимацию с помощью другой программы. На этом занятии я продемонстрирую два способа создания анимации с помощью *Inkscape*, технократический и художественный.

Для этого потребуется свежая версия 0.44, она есть на вашем диске. Понадобятся и другие программы, перечисленные в шаге 6. Мы будем создавать простые GIF-анимации, которые сможет отобразить любой браузер. Найти их можно на сайте LXF, www.linuxformat.co.uk/inkscape. Имена файлов я буду упоминать в тексте, вы найдёте их также в разделе Magazine на диске.



НА ДИСКЕ

- *Inkscape 0.44*
- Файлы проекта

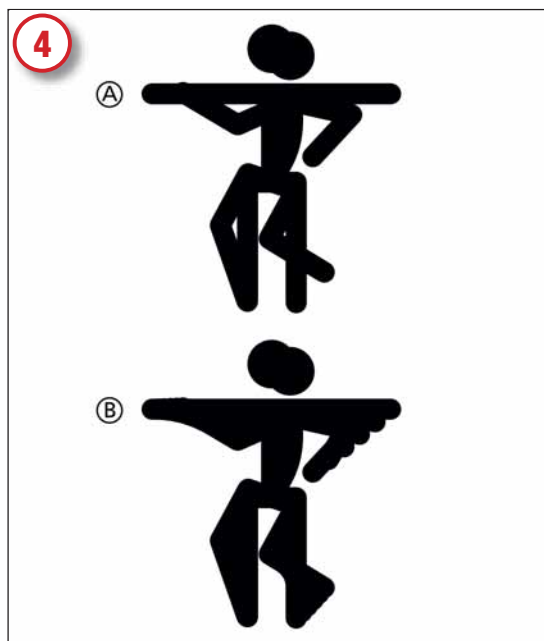
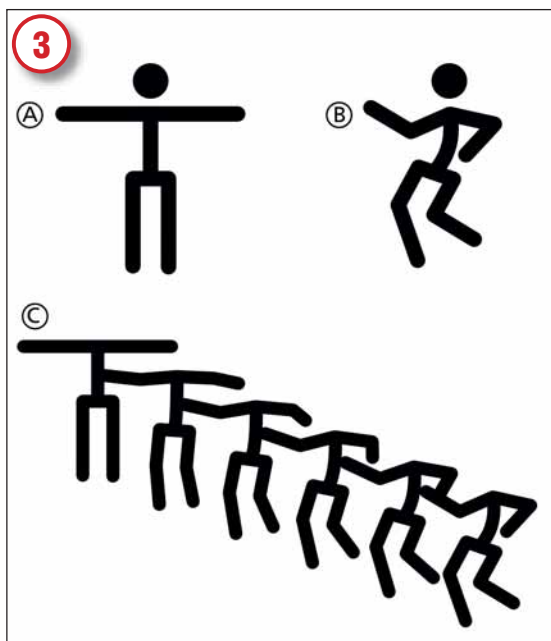


Припасти стопку слоёв

Лучше всего создавать анимационные заготовки в *Inkscape*, помещая их в отдельные слои. Новое диалоговое окно *Layers* (Слои) в 0.44 упрощает этот процесс (А). Создавать много слоёв вручную (*Layer > Add, Слой > Добавить*) утомительно, поэтому сделайте это один раз и сохраните конфигурацию как шаблон в *~/Inkscape/templates* для многократного использования (в моём шаблоне 100 чистых слоёв). Через *File > Document Properties* (Файл > Свойства документа) установите размер вашей анимации (В); наша будет размером 200x200.

Нарисовать фигурку

Сделаем анимацию танцующего человечка – сойдет примитивная фигурка наподобие показанной на рисунке (А), нарисованная с помощью трёх простых линий (строго горизонтальные или вертикальные линии чертите инструментом *Pen, Перо*, с нажатым *Ctrl*) и эллипса для головы. Для удобства интерполяции (см. дальше!) соедините голову и тело командой *Path > Combine* (Контур > Связать), а в инструменте *Node* (Узел) при нажатой *Ctrl* щелчками мыши добавьте узелки в коленные и локтевые сгибы фигурки (В).



Промежуточные изображения

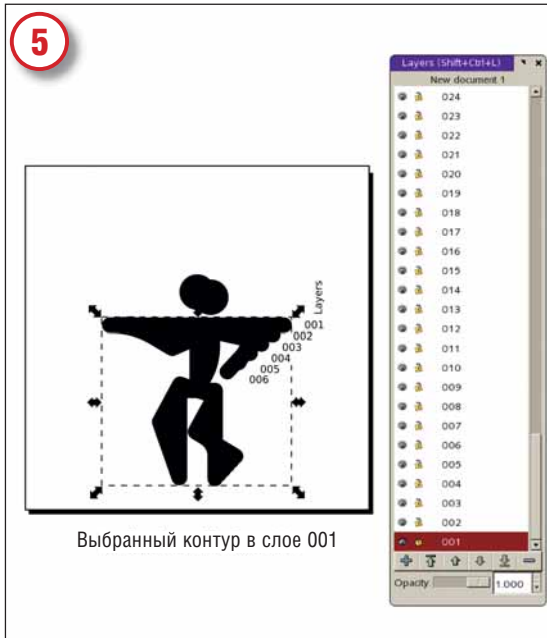
Сделайте копию фигурки (А), передвиньте её вправо и, пройдясь по её суставам инструментом *Node* (Узел), придайте ей позу весёлого танца. Затем вставим промежуточные кадры (так называемый «*tweening*») между двумя «ключевыми фигурами». Выделите обе фигурки и скомаундите *Effect > Generate From Path > Interpolate* (Эффекты > Создать из контура > Интерполировать) (С). Задайте количество промежуточных шагов (скажем, четыре), выберите *Method 1*, а если хотите – установите ненулевое значение *Exponent* (величины шага) для нелинейного ускорения или замедления. Головы к телам тоже можно присоединить в процессе интерполяции за четыре шага.

Собрать клип

Если кадры интерполяции вас не устраивают – удалите всё и исправьте ключевые фигурки. Теперь пора поместить анимацию в нужное место на холсте и распределить кадры между слоями. Удалите промежуточные кадры и поместите вторую ключевую фигуру прямо поверх первой (А). Выделите обе перекрывающиеся фигурки и поместите их в нужное место на холсте. Подгоните размеры с учётом других элементов изображения. Затем интерполируйте фигурки и головы снова, прямо на месте (В).



<<



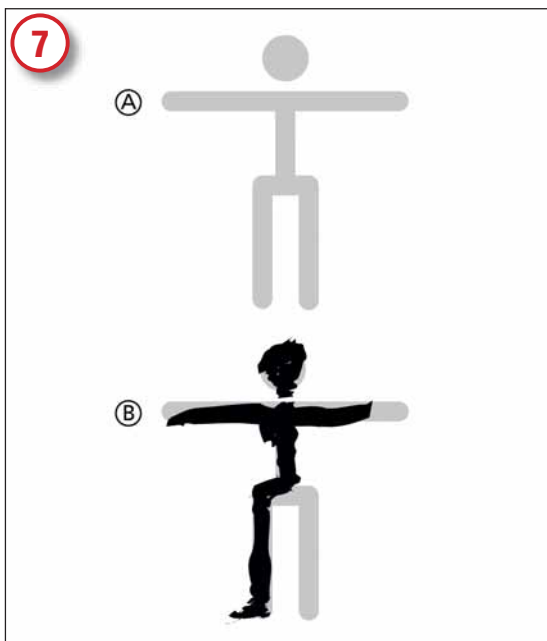
Из кадров – в слои

Следующая (кропотливая) задача – отмена группирования промежуточных фигурок и ручной перенос каждого изображения в отдельный слой от 002 до 005 (ключевые фигурки занимают слои 001 и 006). Можно воспользоваться **Shift+Page Up/Page Down** для перемещения объекта вниз или вверх на один слой, или **Cut > Paste In Place (Вырезать > Вставить в...)**. Затем скопируйте слои с 002 по 006 в слои 007–011 в обратном порядке, чтобы вернуть танцора в его первоначальную позицию. Убедитесь, что все слои видимые, затем сохраните SVG-файл.



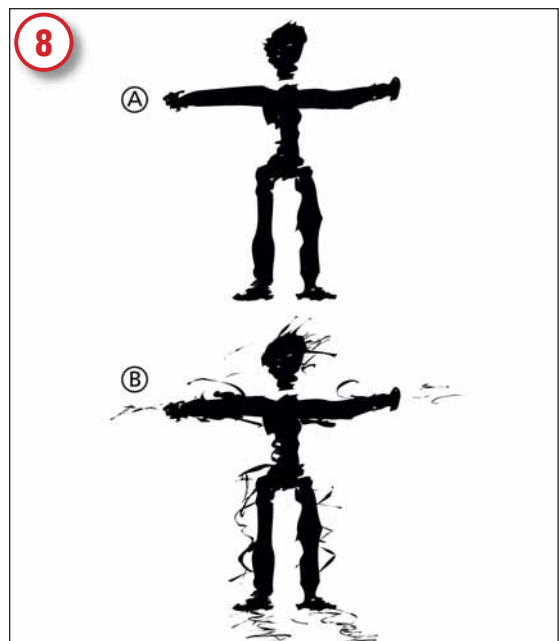
Экспорт и окончательная сборка

Можно сделать это и вручную, но в *Inkscape* есть удобный пакет параметров командной строки. Я написал простой сценарий оболочки для экспорта слоёв. Сценарий принимает в качестве параметров имя исходного файла и номера начального/конечного слоёв, затем экспортирует и собирает кадры в GIF-файл. Кроме *Inkscape*, сценарий обращается к программам *convert* из *ImageMagick* (www.imagemagick.org) и *range* из *num-utils* (<http://suso.suso.org/programs/num-utils>).



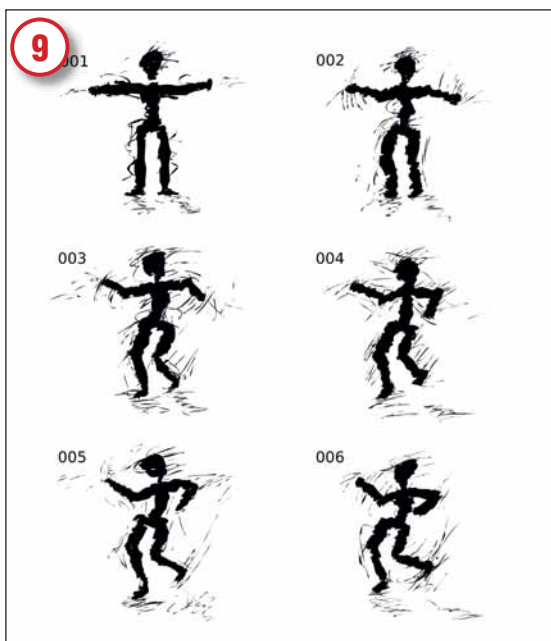
Рисование

Результат нашей работы (*sticky.gif*) в браузере выглядит хотя и мило, но не слишком выразительно. Угловатый стиль хорош для технической анимации, но танец в таком исполнении не впечатляет. Можно ли это исправить? Можно! Скройте все слои, кроме 001, выделите фигурку и уменьшите интенсивность цвета (A). Затем вооружитесь Каллиграфическим пером, задайте ширину (*width*) 20 с «дрождью» (*tremor*) 40 и рисуйте поверх угловатой фигурки (B), стараясь оживить и «развеселить» её.



Штриховка

Закончив, удалите первоначальный скелет, он вам больше ни к чему. Результат (A) неплох, но выглядит как-то сиротливо на безукоризненно белом фоне. Давайте исправим это. Уменьшите ширину штриха до 1 и добавьте несколько случайных штрихов вокруг танцора, чтобы подчеркнуть движения его тела и теней на полу (B). Когда вы увидите его танцующим, случайность штрихов покажется живой и естественной.



Нарисуем оставшиеся кадры

Теперь другие кадры. Золотое правило аниматора – «не копируй!»! Перемещение и масштабирование частей или всей фигуры, вместо рисования её всякий раз заново, постепенно убивает ощущение естественности, а ваша анимация «деревенеет». Не ленитесь: чем больше вы рисуете, тем легче становится это делать. Закончив, можете просмотреть свою анимацию, раскрыв все слои, выделив всю группу фигурок и поместив их в нужное место холста (см. [rough.gif](#)).



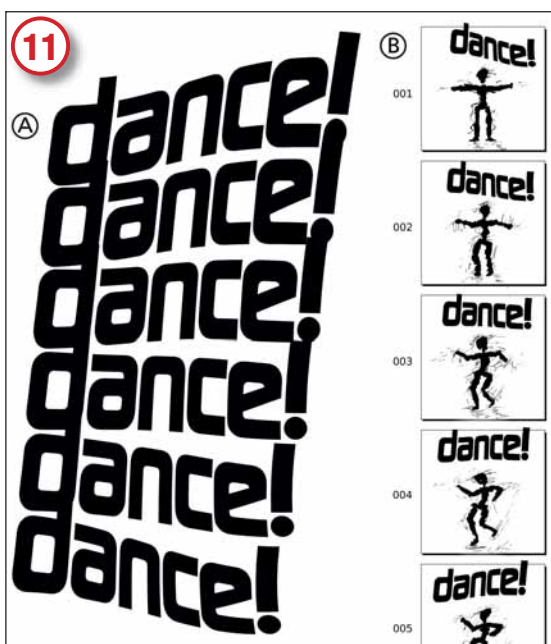
Изогнутые слова

Следующая задача – добавить к нашей анимации текстовый баннер. Чтобы текст двигался вместе с нашей фигуркой, а не стоял неподвижно – попробуем ещё одно новое свойство *Inkscape* 0.44, сдвиг узлов. Создайте текстовый объект в вашем любимом шрифте (A), преобразуйте его в кривые, переключитесь в инструмент *Node* (Узел), затем выделите все узлы (**Ctrl+A**) (B) и при нажатой **Alt** потяните за один из них. Вся надпись плавно изогнётся и растянется, словно знамя на ветру (C).

ПОЛЕЗНЫЕ СОВЕТЫ

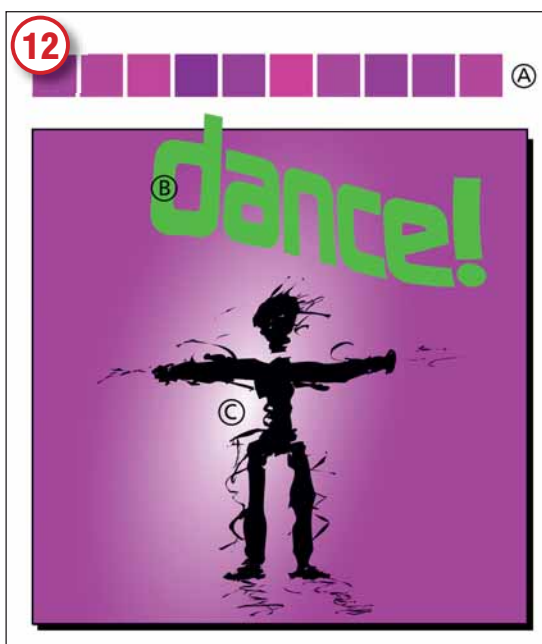


- Если вам нравятся шероховатые линии инструмента *Каллиграфическое перо*, но в рисовании вы не сильны – можно достичь впечатляющих результатов, просто обводя готовые рисунки. Импортируйте изображение на ваш холст, сделайте его полупрозрачным, затем рисуйте поверх него, стараясь выделить самые важные места и игнорируя остальное.
- Для сборки анимированного GIF вместо программы *convert* из *ImageMagick* можно воспользоваться *Gifsicle* (www.lcdf.org/gifsicle). Её преимущество в том, что она может, оптимизируя анимацию, уменьшить размер файла.
- Если вы хотите создать анимацию в формате *Flash*, то самый простой способ сделать это – конвертировать *SVG* в статические *SWF*-кадры (например, с помощью *svg2swf* от http://robba.net/1996_svg2swf), затем собрать кадры в анимацию, для чего я рекомендую *SWF Tools* (www.swftools.org, см. *LXF80*).



Расслоение текста

Теперь сделайте две копии текстового объекта, изогните их в противоположные стороны, как развевающийся флаг, затем интерполируйте их (A). Затем, точно так же, как мы делали с танцующей фигурой, вставьте по одному кадру в каждый слой, помещая их на холст выше фигуры (B). Чтобы видеть, где находится изображение предыдущего слоя, используйте функцию контроля интенсивности в диалоговом окне слоёв, чтобы временно сделать предыдущий слой полупрозрачным. Снова экспортируйте и соберите анимированный GIF ([with-banner.gif](#)).

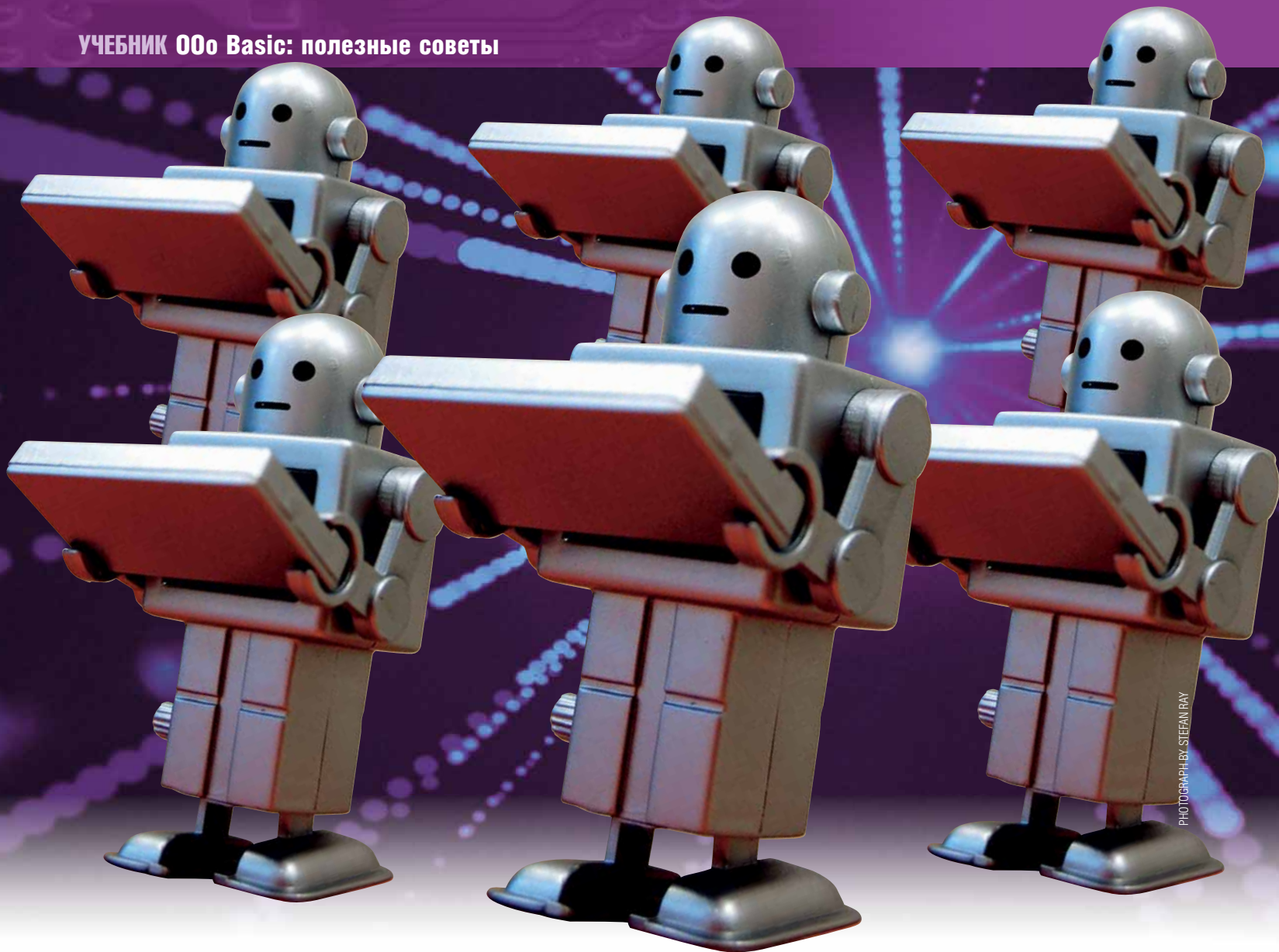


Добавим цвет

Окрасим по-разному фон каждого кадра, чтобы получить эффект дискотек со стробоскопическим освещением. Залейте прямоугольник исходным цветом и используйте диалоговое окно *Tile Clones* (*Расположение клонов*) для разложения его на несколько оттенков (A). Раскрасьте надпись «dance!» контрастным ярким цветом и тоже варьируйте его от кадра к кадру (B). Наконец, добавьте эллиптический градиентный «блик» на заднем плане, передвигая, изменяя размеры и поворачивая его от кадра к кадру, придав им дополнительную энергию (C). Окончательную версию см. на диске ([final.gif](#)). Удачи! [LXF](#)

ЧЕРЕЗ МЕСЯЦ

На следующем занятии мы продолжим тему созданием в *Inkscape* простой Интернет-странички.



PHOTOGRAPH BY STEFAN RAY

АВТОМАТИЗАЦИЯ OPENOFFICE.ORG СЕРИЯ: «СЦЕНАРИИ НА BASIC»

OOo Basic: полезные советы

ЧАСТЬ 4 Продвинем знания о макросах с помощью мудрых советов Марка Бейна.

МЕСЯЦ НАЗАД



Мы изучали, как с помощью OOo Basic извлечь максимум из работы с базой данных.



За последние три выпуска я показал вам несколько своих любимых приемов написания макросов для *Writer* и *Calc*, входящих в состав открытого офисного пакета *OpenOffice.org*. Мы автоматизировали выполнение задач и создание отчетов, писали в документы из командной строки, ускорили получение полезной информации из базы данных... На сей раз я поднесу вам на блюде 14 полезных советов для работы с OOo Basic – они помогут вам повысить эффективность макросов, вне зависимости от программы офисного пакета, с которой вы их используете.

1 Правильная структура данных

Строго говоря, это совет скорее не для OOo Basic, а для программирования вообще: подумайте о своих данных до того, как будете с ними работать – в дальнейшем это избавит вас от множества проблем.

Рассмотрим один из распространенных примеров использования макроса в OOo: вы захотели создать справочник имен и адресов. Если вы новичок в создании баз данных, то, скорее всего, вы создадите одну таблицу, содержащую, скажем, пять полей – имя, адрес, телефон, мобильный телефон и email. На вид вполне разумно; но давайте немного подумаем. Представьте, что вы имеете дело с большой организацией.

Может оказаться, что несколько людей проживают по одному адресу – в результате получится дублирование данных. Согласитесь, что это не здорово: гробится больше дискового пространства, чем надо, и существует большая вероятность того, что данные могут быть неверными.

Взгляните на следующий пример:

Таблица - Покупатели	
Имя	Адрес
Джо Твайтс,	Ансэнк, Камбрия, CA11 9TG
Джозеф Твайтс,	Ансэнк, Камбрия, CA10 9TG

Зная, что один Ансэнк имеет почтовый индекс CA11, а другой CA10, и что Твайтс – довольно распространенная фамилия в этом районе, мы не можем сказать, два ли это разных человека по имени Джо Твайтс или же он один, но в почтовом индексе была допущена ошибка. При проектировании баз данных такие ошибки могут быть сведены к минимуму.

О проектировании баз данных я рассказывать не собираюсь – этот вопрос требует отдельного рассмотрения и достоин целой книги. Но существует простое правило, которым вы можете воспользоваться: проектируйте вашу базу данных так, чтобы каждое поле вводилось только один раз. Поэтому в нашем примере вместо одной таблицы можно использовать две:



НА ДИСКЕ

• Код примеров статьи

Таблица - Покупатели

ID	Имя	Фамилия	Адрес_id
1	Джо	Твайтс	1
2	Джозеф	Твайтс	2

Таблица- Адреса

ID	Город	Графство	Почтовый_индекс
1	Ансэнк	Камбрия	CA11 9TG
2	Ансэнк	Камбрия	CA10 9TG

Теперь можно видеть, что это два разных покупателя с одинаковым именем, но проживающих по разным адресам. Кому интересно, Джо (владелец передвижной лавки видеофильмов, который заявляется к нам домой каждый вторник) на самом деле живет в Лэнгветби, но ради нашего примера я это подправил.

2 Максимум пользы от базы данных

При правильно построенной базе данных вы сможете избежать одной из позорнейших ошибок: набора значений, забитого в макрос. Предположим, у вас есть форма для добавления адресов в таблицу, а одна из колонок называется County (графство). Можно и просто забить все данные в таблицу — но зачем это нам? Куда как лучше использовать диалоговое окно и макрос, пусть они вкальвают.

Вернемся к вводу данных: очевидно, лучшим решением является создание ниспадающего списка из названий округов. Другие варианты реализации: А — написать макрос, содержащий массив названий, из которых потом составляется список, или В — хранить имена округов в таблице базы данных, а затем заполнять список из нее. Если вы выбрали вариант А — получите общественное «у-у-у», а если В — чувство глобокого морального удовлетворения.

Почему вариант В лучше? Потому, что вещи имеют свойство меняться со временем («Когда я служил под знаменами герцога Камберлендского, внучек, никаких ваших Камбрий и в помине не было, а был Камберленд»). Не будете же вы редактировать код каждый раз при добавлении новых данных! Поэтому храните данные в базе и поручите макросу сделать работу за вас. Это подводит нас к следующему совету:

3 Автоматизация диалоговыми окнами

Если вам комфортно работать с данными из командной строки, замечательно. Но если нет, или вы хотите поручить работу ещё кому-то, подумайте об использовании диалоговых окон. Таким способом возможно добавить данные, написав макрос, который составляет запрос **insert** и берет значения из поля ввода, таким образом:

```
SQL = "insert into county (name) values ('" + textbox.value +
"'" )"
```

Вы также можете изменить данные, используя список, поле ввода и запрос **update**:

```
SQL = "update county set name = '" + textbox.value + "'" _
+ " where name = '" + listbox.value + "'" "
```

4 Работайте с OOo 2.0

Прежде чем приняться за создание приложений, удостоверьтесь, что работаете с *OpenOffice.org 2.0*. Может оказаться, что ваш любимый дистрибутив по умолчанию не содержит последней стабильной версии офисного пакета — если вы, например, используете Debian Sarge, то по умолчанию будет установлен *OOo 1.1.3*. Прежде чем приступать к работе, вам следует установить версию *2.0*. Почему? Потому что в этой версии были расширены возможности *OOo Basic*; в частности, теперь вы можете использовать код для добавления пунктов меню.

Если вы создали макрос в *OOo 1.x.x*, а затем обновились до *OOo 2.x.x*, то обнаружите, что ваши модули исчезли — нажитое непосильным трудом кануло в никуда... Не отчаивайтесь, ничего не потерялось. Запустите ваш любимый терминал, перейдите в домашний каталог и наберите **ls -la**. Среди множества каталогов вы увидите каталоги с похожим названием:

```
drwxr-xr-x 3 bainm bainm 4096 1999-10-20 04:08
```

```
.openoffice
```

```
drwx----- 3 bainm bainm 4096 2006-05-31 20:30
```

```
.OpenOffice.org2
```

Разработчики *OpenOffice.org* не волки и ваших макросов не съели — просто они хранятся в другом каталоге. Если вы просмотрите содержимое этих каталогов, то встретите похожие названия: **.openoffice/1.1.3/user/basic/Standard** и **.OpenOffice.org2/user/basic/Standard**. Это каталоги, где находятся ваши модули и диалоги, поэтому прежде чем кодировать, удостоверьтесь, что *OOo 2.0* имеет доступ к старым макросам:

```
mv .OpenOffice.org2/user/basic/Standard .OpenOffice.org2/
user/basic/Standard_old
```

```
ln -s .openoffice/1.1.3/user/basic/Standard OpenOffice.org2/
user/basic/Standard
```

Перезапустите *OpenOffice.org*, теперь он должен обнаружить ваши макросы.

5 Запуск макросов из меню

Вы уже привыкли запускать макросы с помощью кнопки Run в окне Basic (там, где вы набираете код). Вы также знаете, как запустить макрос через меню **Tools > Macros > Macro...** Мило, но раздражающе долго, особенно если макрос часто используется. Почему бы не добавить ссылку на полезный макрос в меню *OOo*? Отличная идея!

Перейдите к меню *OOo*, нажмите на **Tools** и выберите **Configure...** *OOo* отобразит диалог настроек — проверьте, что выбрана вкладка **Menu**. В нижней половине экрана выберите макрос для добавления в меню. В верхней половине экрана найдите место для нового пункта меню, а затем нажмите **New**, чтобы создать его.

Теперь вы можете запросто вызвать макрос, когда захотите. Если

«ХРАНИТЕ ДАННЫЕ В БАЗЕ И ПОРУЧИТЕ МАКРОСУ СДЕЛАТЬ РАБОТУ ЗА ВАС.»

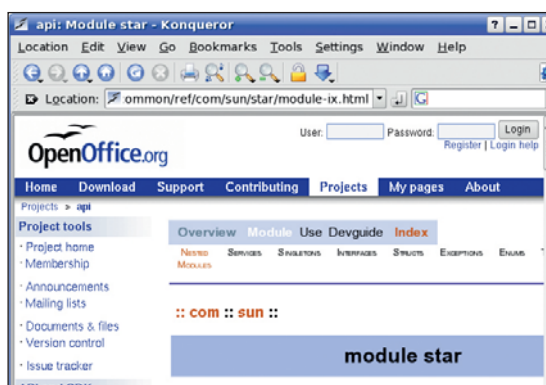
вы любите клавиатуру больше, чем мышь, то назначьте горячие клавиши для быстрого вызова макроса: снова отправляйтесь в диалог *OOo Configuration*, но теперь выберите вкладку **Keyboard**. Чтобы запустить макрос при открытии документа... угадали: используйте диалог *Configuration*. Если вы выберете закладку **Events**, то увидите всевозможные события, к которым можно назначить макрос — все, что надо сделать, это выбрать **Open Document**.

6 Добавьте меню автоматически

Если вы используете или обновились до *OOo 2.0*, то сможете создавать дополнительные пункты меню из макроса. Чтобы это сделать, используйте свойство *com.sun.star.beans.PropertyValue*. Например, если вы хотите добавить процедуру **showMessage** из модуля с именем **OOobasic4**, можете использовать следующий способ:



Заберитесь в сеть, чтобы получить больше информации о модулях, встроенных в *OpenOffice.org*.



ВОЛШЕБНОЕ ТАИНСТВЕННОЕ ПУТЕШЕСТВИЕ ДЖО

Практическое применение макросов *OpenOffice.org*

Помните, в совете номер один я говорил вам о Джо Твайтсе? Пусть Джо на микроавтобусе совершает объезд потенциальных покупателей. Остановившаяся у очередного дома, он открывает заднюю дверь своего микроавтобуса, чтобы показать последние видеокассеты, DVD и игры. Представьте деревянную панель, встроенную в микроавтобус. Посреди стены стоит 17-дюймовый Flatron, а на нем крутятся сразу три *Хроники Нарнии* – внимательно рассмотревшись, вы увидите под панелью PC с клавиатурой и мышью.

Любопытствуете, что за компьютер? Он работает на Debian 3.1 (благодаря *LXF70*), для проигрывания DVD используется *Kaffeine*; и компьютер, и монитор работают через преобразователь Belkin DC-AC.

Мило; но какое это имеет отношение к *OpenOffice.org*? Хороший вопрос. Перед тем как отправиться на работу, Джо запускает документ *OOo Calc* с макросом, выполняющимся по событию 'Open Document'. Этот макрос почти такой же, как разработанная нами процедура *connectToDatabase* – разница только в том, что в нем стоит обработка ошибок. Зачем? Потому что на этот раз главная база *MySQL* находится не на

PC в автобусе, а на компьютере в доме Джо. Главный компьютер Джо использует беспроводную связь, а PC в автобусе имеет небольшую Wi-Fi антенну. Теперь, если Джо дома, то макрос подсоединится к базе данных; в противном случае *Calc* просто откроет документ.

Определив доступность базы данных (то есть возможность установления соединения), макрос начинает просмотр листа в *Calc* (он называется **Daily data**). Ячейка A1 содержит дату. Если эта дата раньше текущей, макрос переходит ко второму листу и запускает процедуру: **uploadYesterdayRoute**. Когда этот макрос завершит работу, другой макрос, **downloadTodayRoute**, опросит базу данных и скачает информацию о сегодняшних адресах, которых надо посетить – эта информация включает в себя видеокассеты, DVD и игры, взятые напрокат клиентами.

Путешествуя по стране, Джо обновляет таблицу *Calc*, записывая, кто и что у него брал. Если он кого-то не застанет дома, то вызовет другую процедуру, **printReminder**, которая изменит данные о прокате и перенесёт этот адрес в маршрут следующей поездки. Процедура также распечатает эту ин-

формацию на принтере, расположенном в кабине (чтобы оставить ее клиенту). На следующее утро процесс повторяется. Для Джо это значит, что у него имеется полная история всех его клиентов. Более того, он может провести анализ и понять, что для него прибыльнее: например, жители Камбри любят боевики, и т.д., и ориентироваться на аудиторию, которую собирается посетить в назначенное время.

Ладно, а зачем сюда приплели Джо? Затем, что многие крупные фирмы тратят на разработку систем, подобных придуманной нами для Джо, огромные деньги. А вы можете сделать это бесплатно с помощью Linux и своих знаний.



Впечатление художника: ты лучше всех, Эффи!

```
<< Menuitem(0).Name = "CommandURL"
Menuitem(0).Value = "macro:///Standard.OOobasic4.showMessage()"
Menuitem(1).Name = "Label"
Menuitem(1).Value = "Do whatever you want direct from the menu"
Menu.insertByIndex( 0,Menuitem)
```

Чтобы понять, как это работает, взгляните на макрос **setUpMenu** с нашего диска. Опережаю ваш вопрос: удалять уже существующие пункты меню нельзя, можно только добавлять свои собственные, как элементы в обычных меню *OOo*.

7 Разбивайте строки кода

Это совет по улучшению читабельности вашего кода. Читабельность... есть ли такое слово? Короче, вы её улучшите, разбивая длинные строки кода на несколько с помощью символа нижнего подчеркивания. Например,

```
Menuitem = CreateMenuitem( "macro:///Standard.OOobasic4.showMessage()", "Show a test message" )
    стоит превратить в
Menuitem = _
CreateMenuitem( "macro:///Standard.OOobasic4.showMessage()", _
"Show a test message" )
```

```
+ ", address.town, address.county, address.postcode" _
+ " where customer.address_id = address.id"
```

Повторюсь: разбиение длинной строки кода на несколько строк никак не повлияет на работоспособность кода, но зато такой код будет проще понять.

8 Помните о модулях и библиотеках

Вместо того, чтобы сохранять всю свою работу в огромном модуле, попробуйте разбить макросы на отдельные осмысленные функции, запишите их в несколько модулей, а модулям дайте имена, отражающие их суть: **Макрос1**, **Макрос2**, **Макрос3** и так далее – это, согласитесь, нудно и непонятно. Если вы дадите «говорящие» имена своим макросам (например, назовёте их **СоздУчетЗапись** или **ОбслужитьПокупателя**), то в будущем скажете мне спасибо. Но не увлекайтесь – не пытайтесь создать легион модулей, в каждом из которых будет по одному или два макроса.

Когда у вас наберётся много модулей, можете сгруппировать их в библиотеки. Опять-таки, попытайтесь дать им осмысленные имена.

9 Не скупитесь на комментарии

Когда вы пишете макрос, то думаете, что все понятно и логично. Но когда вы вернетесь к нему через полгода-год, изрядно подзабыв, что, собственно, делали, код может показаться непонятным. Я бы порекомендовал вам по ходу работы делать заметки – просто объясните, чего вы пытаетесь достичь.

Как добавить комментарий? Для этого начните строку со слова **REM** или одиночной кавычки:

```
REM Это комментарий
' И это
А это уже не комментарий
```

Вы также можете добавить комментарий в конце строки кода `msgbox "Это сообщение" 'а это комментарий`

С другой стороны, не пытайтесь закомментировать всё, не то ваш код затеряется в пояснениях:

```
REM Цель данного куска кода состоит в том,
REM чтобы сообщить пользователю информацию. В небольшом окне отобразится текстовое сообщение
REM Пользователь должен подтвердить, что он прочел сообщение.
REM В противном случае выполнение программы
```

«НАЗЫВАЙТЕ МАКРОСЫ ИМЕНАМИ, ОТРАЖАЮЩИМИ ИХ СУТЬ.»

На работу кода не влияет, но выглядит лучше – ну, то есть, читабельнее.

Использование нижнего подчеркивания не поможет, если вы попытаетесь расчленив строку вроде этой:

```
SQL = "select customer.firstname, customer.surname, address.town, address.county, address.postcode where customer.address_id = address.id"
```

Чтобы разбить её на несколько строк, воспользуйтесь символом «плюс»:

```
SQL = "select customer.firstname, customer.surname" _
```

```
REM продолжено не будет.
```

```
msgbox "Ну здравствуй!"
```

10 Используйте встроенные диалоговые окна

Совершенно незачем делать всю работу по созданию диалоговых окон – лучше позаимствовать уже встроенные в *OpenOffice.org* решения. Предположим, вы хотите, чтобы пользователь выбрал папку для сохранения результатов. Все, что вам надо – это диалог *FolderPicker*:

```
Dim FolderDialog, UserFolder
```

```
FolderDialog = CreateUnoService("com.sun.star.ui.dialogs.  
FolderPicker")
```

```
UserFolder = FolderDialog.Execute()
```

Чтобы узнать, какие ещё предлагаются диалоги, проконсультируйтесь на сайте *OpenOffice.org* или загляните на <http://snipurl.com/rh5z>. На этой странице вы найдёте подробное описание каждого из диалогов. Их немного, но из них можно извлечь немалую пользу и сберечь свое время.

Раз уж зашла речь о **com.sun.star**... На самом деле это набор встроенных модулей *OOo*. Вы, несомненно, заметили, что используя какой-либо объект *OOo*, мы всегда применяем следующий формат:

```
MyObject = CreateUnoService("com.sun.star.foo.bar.object")
```

Это довольно просто, но иногда случается, что вы не можете определить, какие именно ещё объекты вам пригодятся. Снова, вы найдёте всю дополнительную информацию о списке модулей в **com.sun.star** на странице, посвящённой *OpenOffice.org*, по адресу <http://snipurl.com/rh61>.

11 Форматируйте отчеты

За последние месяцы мы увидели, как писать в *Writer*, считать в *Calc* и использовать данные из баз данных. Вы также можете использовать макрос для того, чтобы сделать вашу работу более презентабельной с минимумом усилий. Представьте, что вы заполнили таблицу из базы данных, выполнили всю необходимую работу (с помощью макроса, конечно), а теперь готовы распечатать свой отчет для менеджера/учителя (который уж обязательно повысит вам зарплату/оценку за вашу чудесную работу). Что вы обычно делаете с документом перед тем, как его распечатать? Я предполагаю, что добавляете заголовок, номер страницы, общее число страниц и дату. Это делается просто:

```
oDoc = ThisComponent
```

```
oPageNumber = oDoc.CreateInstance( "com.sun.star.text.  
TextField.PageNumber" )
```

```
oPageCount = oDoc.CreateInstance( "com.sun.star.text.  
TextField.PageCount" )
```

```
oDateTime = oDoc.CreateInstance( "com.sun.star.text.  
TextField.DateTime" )
```

Теперь хорошо бы сделать нижний и верхний колонтитулы:

```
oStyles = oDoc.getStyleFamilies().getByName( "PageStyles" )
```

```
oPStyle = oStyles.getByName( "Default" )
```

```
oHeader = oPStyle.RightPageHeaderContent
```

```
oFooter = oPStyle.RightPageFooterContent
```

Вы можете поместить текст в любое место колонтитула:

```
oHeader.getCenterText().setString( "LXF83 Article" )
```

Однако функции (например, **DateTime**) обрабатываются по-другому, с применением курсора:

```
oCursor = oHeader.getRightText().createTextCursor()
```

```
oHeader.getRightText().insertTextContent( oCursor,  
oDateTime, True )
```

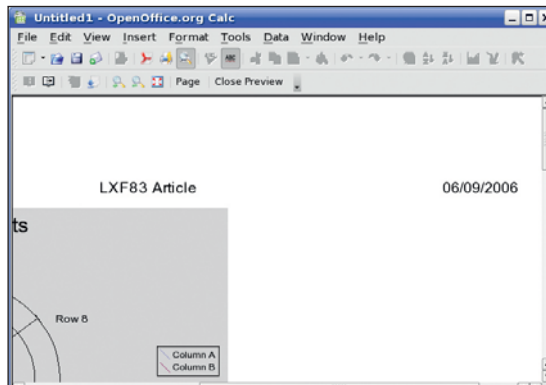
Используя оба этих метода, можете создавать всевозможные вставки:

```
oFooter.getRightText().setString( "Page " )
```

```
oCursor = oFooter.getRightText().createTextCursor()
```

```
oCursor.gotoEnd( False )
```

```
oFooter.getRightText().insertTextContent( oCursor,  
oPageNumber, True )
```



Предпросмотр печати поможет увидеть колонтитулы, созданные вашим макросом.

```
oCursor.gotoEnd( False )
```

```
oCursor.setString( " of " )
```

```
oCursor.gotoEnd( False )
```

```
oFooter.getRightText().insertTextContent( oCursor,  
oPageCount, True )
```

Наконец, необходимо обновить страницу, чтобы новая информация отобразилась:

```
oPStyle.RightPageHeaderContent = oHeader
```

```
oPStyle.RightPageFooterContent = oFooter
```

12 Объединяйте документы

Не просто создавайте новые документы – если у вас их несколько и нужно их скомпоновать, воспользуйтесь курсором для создания единого документа. Такая задача возникает, когда над проектом работает несколько человек:

```
oCursor.gotoEnd(false)
```

```
oCursor.BreakType = com.sun.star.style.BreakType.PAGE_  
BEFORE
```

```
oCursor.insertDocumentFromUrl(SrcFile, argsInsert())
```

Очевидно, эти файлы можно брать из какого-то каталога или читать их из таблицы базы данных, а затем с помощью цикла, перебирая каждый документ, добавлять его к основному документу.

После объединения документов желательно создать раздел Содержание (Content). Это поддается автоматизации:

```
oDoc = ThisComponent
```

```
oCurs = oDoc.getText().createTextCursor()
```

```
oCurs.gotoStart(False)
```

```
oDoc.getText().insertTextContent(oCurs, oIndex, False)
```

13 Ищите всё, что вам нужно

Может, и не всё, но если хотите посмотреть в алфавитном порядке список встроенных в *OpenOffice.org* функций, то отправляйтесь на <http://snipurl.com/rh67>. Вы увидите каждую функцию, свойство, сервис и объект, к которому они принадлежат.

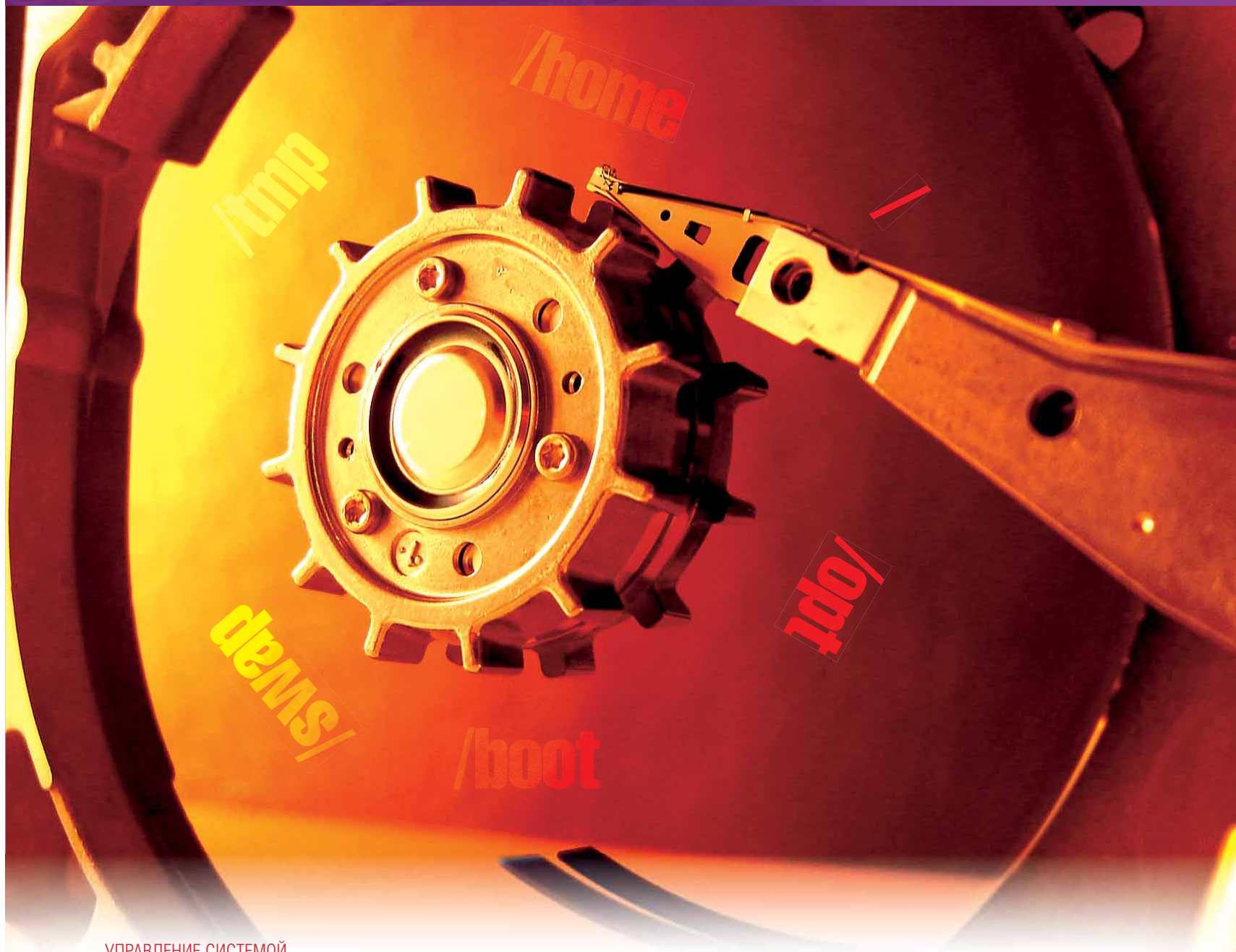
14 Учитесь у других

Хороший способ изучать *OOo Basic* – это смотреть, как пишут макросы другие люди: можете с ними не соглашаться, но возьмите на вооружение полезные идеи. Можно, конечно, запрячь Google, запросив 'OpenOffice.org macros', но для начала стоит посмотреть на макросы на странице проекта *OOo*. Примеры вы найдёте на <http://codesnippets.services.OpenOffice.org>.

На данном уроке мы рассмотрели всего несколько примеров, но я надеюсь, что их достаточно для понимания, как легко делаются эффективные макросы. У вас есть все ресурсы, посвященные *OOo*, и вы сможете сами создать красивые решения. **LXF**

**ЧЕРЕЗ
МЕСЯЦ**

Мы увидим, как легко преобразовать макросы Excel в *OOo Calc*.



УПРАВЛЕНИЕ СИСТЕМОЙ

LVM Заводим виртуальные разделы

Долой тиранию фиксированных разделов диска – Нейл Ботвик покажет, как достичь гибкости при помощи управления логическими томами (Logical Volume Management).



Как управлять разделами? Спросите 10 пользователей Linux о лучшем способе разбиения жесткого диска – итогом будет не меньше 11 разных ответов. Вдобавок все ответы, вероятно, правильные: у каждого метода свои достоинства, все зависит от ваших намерений по использованию системы. Проблемы возникают немного позже, когда реальность расходитсся с планами и вы замечаете нехватку свободного места в одном разделе и ненужный избыток – в другом.

Размеры разделов можно изменить при помощи утилиты *QtParted*, но она (и инструменты, ею используемые) ограничивается отодвиганием конца раздела. Вы можете усечь раздел, чтобы высвободить место для нового раздела, или увеличить раздел, включив следующее за ним пустое пространство. Если надо расширить раздел **/usr**, вклинившийся между разделами **root** и **home**, то вам не повезло: придется произвести резервирование данных, удалить и вновь создать разделы, а потом восстановить данные из резервной копии.

К счастью, решение существует – это менеджер логических томов (LVM), представляющий собой программный слой между разделами и накопителями. Ваши разделы становятся виртуальными, а не привязанными к фиксированному положению на диске. Сегодня я покажу вам, как использовать LVM для создания раздела (далее называемого логическим томом), изменения его размера или удаления без проблем и забот о физическом размещении. Можно даже добавлять и удалять целые диски, не затрагивая вашу файловую систему.

Работаем со слоями

LVM поставляется в большинстве дистрибутивов – или как часть стандартного набора программ, или в программном репозитории (пакет обычно называется **lvm2**). Там содержатся все используемые инструменты, но учтите, ядро тоже должно поддерживать LVM. Многие дистрибутивы опять-таки предоставляют подходящее ядро, но если

ТРЕБУЮТСЯ ПРАВА ROOT

Все команды LVM, используемые в данном уроке, требуют запуска с правами суперпользователя (root). Нужно будет открыть терминал и ввести

```
su -
# а затем пароль суперпользователя. С очевидным исключением команд отображения, большинство LVM-команд деструктивны: они удаляют или затирают данные вашего жесткого диска, иногда и тайком. Помните присказку «компьютер делает то, что вы просите, а не то, что вы думаете», и будьте осторожны при запуске представленных здесь команд! Вероятно, хорошей идеей будет экспериментировать где-нибудь на нейтральной территории, например, на отдельном жестком диске. Если у вас такого нет, то для тестов вполне сгодятся приличной емкости USB-брелок или карта памяти в карт-ридере.
```

Можно также создать LVM на loор-устройстве. Это малоприменимо в реальности, но является хорошим способом экспериментировать, не трогая разделы вашего жесткого диска. Все, что вам нужно – это свободное место для создания loор-файла:

```
dd if=/dev/zero of=lvmtest bs=1M count=8192
losetup /dev/loop1 lvmtest
pvcreate /dev/loop1
vgcreate etc.
```

Здесь создается файл размером 8 ГБ, присоединяется к loор-устройству и используется в качестве физического тома для LVM.

вы делаете все самостоятельно, то вам необходимо включить следующие опции:

```
CONFIG_MD=y
CONFIG_BLK_DEV_DM=y
```

Их можно найти в инструменте конфигурирования ядра: Device Drivers > Multi-Device Support (RAID и LVM) > Device Mapper Support

Имеется две версии Управления Логическими Томами: изначальная – LVM и последняя – LVM2. LVM2 в основном имеет обратную совместимость с LVM, но когда на этом уроке я говорю о LVM, то подразумеваю LVM2.

LVM предоставляет логические тома трех уровней: физические тома, группы томов и собственно логические тома. В самом низу находятся физические тома – это разделы жесткого диска, на которых хранятся данные (**/dev/hda1**, **/dev/sdb3** и т.п.). Можно также создать физический том, охватывающий весь диск, в этом случае вам не нужно вначале разбивать этот диск на разделы. Один или несколько физических томов объединяются в группы – второй уровень. Вы можете рассматривать группу томов как виртуальный жесткий диск, который может иметь разделы более чем на одном физическом носителе. И, наконец, поверх групп томов создаются логические тома.

Все группы томов разбиваются на один или несколько логических томов, точно так же, как физический жесткий диск разбивается на один или несколько разделов. Основное отличие в том, что все становится более гибким – от размера виртуальных дисков и разделов до соглашений по именованию, которые вы используете. Здорово? Давайте начнем с низшего уровня: создадим несколько физических томов.

На дне

Небольшое предупреждение: начиная с этого момента, все приводимые команды требуют прав суперпользователя (см. врезку **Требуются права root** вверху). Поехали. Итак, физические тома создаются при помощи *pvcreate*; тап-страница содержит множество опций, но обычно требуется указать только имя используемых раздела(-ов) или диска(-ов). Вот хорошие примеры этой команды:

- *pvcreate /dev/hda5*
- *pvcreate /dev/md0*
- *pvcreate /dev/sda7 /dev/sdb*

Первая из них создает физический том на отдельном разделе; вторая делает то же самое на RAID-разделе. Третий пример создает физические тома в разделе первого диска и на всем втором диске. Если вы хотите создать физический том во весь диск, то последний при этом не

должен содержать никаких разделов: если они есть, то *pvcreate* откажется работать с ним.

Разделы можно удалить при помощи любого инструмента разметки диска, например *fdisk*, или командой

```
dd if=/dev/zero of=/dev/sdb bs=512 count=1
```

Обязательно убедитесь, что вы работаете с нужным диском – очень трудно будет восстановить данные, если вы удалите таблицу разделов не того диска.

Работа с группами

Вообще-то наш друг *pvcreate* только подготавливает устройство для использования LVM – теперь вам необходимо создать одну или несколько групп томов. Для стандартного настольного ПК обычно вполне достаточно одной группы. Для масштабных или специализированных применений, например, серверов с интенсивной нагрузкой, может понадобиться больше. Но сам процесс одинаков, вне зависимости от количества созданных вами групп:

```
vgcreate myvg /dev/hda5 /dev/sdb
```

Здесь создается отдельная группа томов с именем **myvg**, состоящая из раздела на **hda** и целого **sdb** (в предположении, что они были подготовлены *pvcreate*). Кстати, вы можете спокойно проигнорировать любые ошибки, связанные с CD или DVD-приводами: это LVM сканирует ваши диски и выдает ошибку для любого устройства, содержащего диски только для чтения или без дисков.

Хотя общепринято давать группам томов нумерованные имена (**vg00**, **vg01**...), вы можете называть их по-своему. Например, если

«МОЖНО ДАЖЕ ДОБАВЛЯТЬ ЦЕЛЫЕ ДИСКИ, НЕ ЗАТРАГИВАЯ ФАЙЛОВУЮ СИСТЕМУ.»

раздел вашей операционной системы размещается в одной группе, а сами вы работаете с другой, вы можете назвать их **System** и **Work** соответственно. Другая причина для использования своих собственных имен – избежание конфликтов: вдруг вам потребуется подсоединить другое устройство с группой томов под именем **vg00**, чтобы извлечь данные – возникнет путаница.

Чтобы проверить, создана ли группа томов, выполните

```
vgdisplay myvg
```

Выведется информация о вашей группе томов. Запустите ее без аргументов, чтобы увидеть параметры всех групп.

Наконец-то логические тома

Теперь к третьему слою; команда создания логических томов лишь немного длиннее, чем команда для создания группы томов. Вот она:

```
lvcreate --name myvol --size 5G myvg
```

Опция **--name**, как пояснил бы вам Шерлок Холмс, дает имя тому. Она не обязательна, но если ее опустить, том получит имя вида "volNN" – значимое имя, безусловно, лучше. Аргумент **--size** задает размер тома



КАК УВИДЕТЬ ТОМА И ГРУППЫ

Вы увидите, что можно создать несколько физических томов, групп томов и логических томов. Приятно, что есть команды, помогающие отследить, что происходит: *pvdisplay*, *vgdisplay* и *lvdisplay* показывают параметры ваших физических томов, групп томов и логических томов соответственно. При запуске команды без аргументов выводится список всех объектов данного типа. Вы можете ограничиться конкретными томами и/или группами, указав их в команде – например, так:

```
vgdisplay myvg # информация об одной группе
lvdisplay myvg # информация обо всех логических томах в группе
lvdisplay /dev/myvg/myvol system/usr # информация об указанных логических томах
```

Если у вас всего одна группа томов, можете сократить число символов в команде, присвоив имя группы переменной окружения **LVM_VG_NAME**. После этого будет использована именно она, даже если в соответствующей команде имя группы не указано.

« (ну, это же элементарно) и указывается он в мегабайтах, или по вашему выбору К, М, G или T (килобайты, мегабайты, гигабайты или терабайты, дорогой мой Ватсон). Вы должны указать или `--size`, или `--extents`. Extent (непрерывная область на диске) – это наименьшая единица данных в группе, аналогичная блоку физического диска. Размер по умолчанию – 4 МБ.

Ваш новый том находится в `/dev/volumegroup/volumename`, и команда

```
mke2fs -j /dev/myvg/myvol
```

создаст на новом томе файловую систему ext3.

Повтором команды `lvcreate` вы можете добавлять логические тома, пока хватает свободного места на диске, но при этом следует помнить несколько моментов. Как вы скоро увидите, LVM чрезвычайно упрощает изменение размера логических томов, а вот изменение размера раздела проходит гладко только при его увеличении. Уменьшение размера раздела в лучшем случае рискованно, а для некоторых файловых систем невозможно (это зависит от распознавания вашей файловой системы программой разметки – если у вас очень экзотическая файловая система или старая версия GParted, вряд ли это сработает).

Поэтому задавайте размер ваших логических томов в точности таким, как надо – завышать его, исходя из соображений «а вдруг потом потребуется больше», бессмысленно. По тем же причинам, хотя под группу томов и можно грохнуть все свободное место, крайности в этом нет – разве что при подлинной необходимости. Неиспользованное пространство можно приписать к любому логическому тому в любой момент. Однако не увлекайтесь таким подходом: большинство файловых систем при заполнении «под завязку» вынуждены фрагментировать файлы. Наилучшая производительность достигается при заполнении на 75–80%. Рассчитайте размер каждого тома и добавьте к нему еще 30%, чтобы дать файловой системе некоторое пространство для толкотни.

Добавляем пространства

Пока мы создавали одни логические тома, которые, похоже, не имеют особых преимуществ по сравнению с традиционными физическими. Но это только начало. Если вы решите, что вам нужно побольше места на **myvol**, нужно сделать всего лишь

```
lvresize --size +2G /dev/myvg/myvol
```

Это увеличит размер логического тома на 2 Гб. Начало `/dev/` не обязательно, но означает, что вы можете использовать завершение по клавише TAB в командной оболочке, чтобы убедиться, что имя верное.

Должен отметить, что эта команда изменяет размер только тома, а не его файловой системы. Увеличение размера тома, содержащего смонтированную файловую систему, вполне безопасно, но если вы хотите уменьшить размер тома, то прежде уменьшите размер файловой системы по крайней мере на ту же величину. При увеличении размера сначала увеличиваем размер тома, а затем файловой системы. Некоторые файловые системы позволяют увеличить их без размонтирования (а вот XFS разрешает изменение размера только в смонтированном состоя-

ФИЛЬТРАЦИЯ НЕПОДХОДЯЩИХ УСТРОЙСТВ

Хотя сообщения о якобы ошибках на `/dev/cdrom` или другом читающем устройстве безвредны, они изрядно раздражают. Чтобы LVM не обращался к `/dev/cdrom`, добавьте следующую строку в `/etc/lvm/lvm.conf`:

```
filter = [ "rl/dev/cdrom" ]
```

r означает reject (ignore) (отказать, игнорировать), а за ним следует регулярное выражение, обозначающее игнорируемые устройства. Эта строка, возможно, в файле уже есть, но в виде комментария – так раскомментируйте ее!

Видимо, придется ее и переместить: LVM использует первую же встретившуюся строку фильтра, соответствующую устройству, а обычно это строка

```
filter = [ "a/.*/" ]
```

разрешающая все. Поместите фильтры для конкретных устройств выше нее, и она станет актуальна только для неуказанных девайсов.

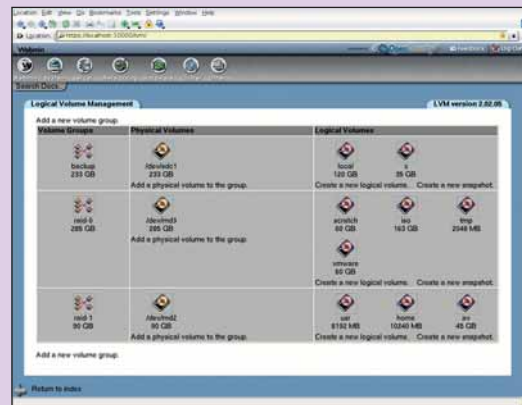
Другой вариант – ограничиться только отдельными устройствами, проигнорировав остальные. Попробуйте, например:

```
filter = [ "al/dev/sdl", "al/dev/mdl", "rl.*" ]
```

Тут содержатся сразу 3 фильтра. Первый разрешает SCSI- и SATA-устройства (если у вас SATA-диски и PATA DVD-привод), второй – программные RAID-девайсы, а третий запрещает все остальное. Исправив `lvm.conf`, запустите `vgscan` для обновления кэша, где LVM хранит результаты сканирования устройств.

ГРАФИЧЕСКОЕ АДМИНИСТРИРОВАНИЕ

Я сосредоточился на командной строке потому, что это обычный способ администрирования LVM. Если исключить инструменты, специфичные для дистрибутивов, лучшим GUI-инструментом администрирования LVM является вездесущий *Webmin*. Он содержится в репозиториях большинства дистрибутивов и, возможно, уже установлен на вашем компьютере. Вы можете найти LVM-модули в разделе Оборудование (Hardware) пакета *Webmin*.



LVM-модуль для Webmin обеспечивает графический интерфейс управления томами и группами.

нии). Попробуйте специализированные инструменты изменения размера файловых систем (например, `resize2fs`, `resize_reiserfs`, `xfs_growfs` etc) или графический инструмент вроде *QParted*.

Подразумевается, что у вас хватает свободного пространства в вашей группе томов. LVM не может наколдовать пространства из ниоткуда, но он может облегчить добавление дополнительного диска в ваш компьютер, добавив затем его пространство к имеющейся группе(ам) томов. Например,

```
pvccreate /dev/hdb
vgextend myvg /dev/hdb
```

добавит весь `/dev/hdb` к существующей группе томов **myvg** (столь же просто вы можете создать раздел на диске и добавить его в группу томов). Мы можем использовать это пространство для создания дополнительных логических томов или для увеличения размера существующих. Использование LVM означает, что одну файловую систему можно разместить на сколь угодно большом количестве физических разделов и дисков, обеспечивающих требуемую память.

Уменьшение разделов

А как насчет перемещения и удаления элементов LVM? Для начала, логические тома могут быть удалены командой

```
lvremove myvg/myvol
```

Перед этим вы должны размонтировать раздел, иначе `lvremove` откажется выполнить ваш запрос. Удаленное пространство возвращается в группу томов.

Аналогично, `vgremove` удаляет группу томов целиком, но только после удаления всех ее логических томов командой `lvremove`.

Из группы томов можно также удалять отдельные физические тома. Это требует некоторой осторожности, так как перед удалением нужно убедиться, что на этом томе или диске не содержится данных из принадлежащих группе томов. Но это также означает, что вы в основном можете перенести систему на диск большего размера практически без простоя, кроме вызванного физическим переподключением дисков (или даже и без этого, если вы используете диски с «горячим» подключением). Я добавил «в основном», потому что любым не-LVM разделом потребуются заниматься отдельно. Дистрибутивы вроде Fedora Core размещают `/boot` на отдельном разделе, а все остальное на LVM. Поскольку в процессе работы системы `/boot` не требуется, то скопировать его достаточно легко.

Чтобы скопировать остатки системы, скажем с **hda5** на **hdb5**, создайте физический том и добавьте его в группу томов, как описано выше. Теперь вам нужно убедиться, что все данные этой группы томов размещены на новом диске. Во-первых, удостоверьтесь, что места достаточно:

```
pvdiskdisplay /dev/hda5 /dev/hdb5
```

Убедитесь, что свободный физический размер (Free PE) нового диска больше, чем занимаемый физический размер старого (Allocated PE). Если это так, то перенесите данные командой

```
pvmove /dev/hda5 /dev/hdb5
```

заменяя, естественно, имена устройств в соответствии с вашей системой.

`pvmove` может перетаскивать данные довольно долго, но не пугайтесь: его действия не мешают вам работать за компьютером. Если событие вроде отключения питания или краха системы прервет процесс, запустите потом `pvmove` без параметров, и процесс продолжится с момента остановки. Используя `pvdiskdisplay`, проверьте, что на старом диске нет занятого места, после чего удалите его из группы:

```
vgreduce myvg /dev/hda5
```

```
pvrmove /dev/hda5
```

Первая команды удаляет раздел из группы томов (вы можете в этом убедиться при помощи `vgdisplay`). Вторая просто удаляет у раздела метку LVM.

Для перемещения данных отдельного логического тома следует выполнить:

```
pvmove --name myvg/myvol /dev/hda5 /dev/hdb5
```

Имя тома-назначения не обязательно; если вы его опустите, `pvmove` перенесет данные в свободные физические области текущей группы томов. Это весьма полезно, если у вас несколько физических томов, а удалить надо только один.

Команда `pvmove /dev/hdaN` очистит раздел N, переместив данные в свободные области группы, которой он принадлежит.

Прекрасные снимки

Создание резервной копии работающей файловой системы не обходится без проблем. Резервирование — процесс не мгновенный, поэтому всегда существует вероятность, что файловая система за это время изменится. Особенно это касается резервирования **/var**, где файлы в каталогах типа **/var/log** или **/var/lib** могут изменяться ежеминутно (зачастую так и происходит). К счастью, LVM предоставляет измученным сисадминам опцию моментальных снимков логических томов. Взгляните сюда:

```
lvcreate --size 1G --snapshot --name mybackup myvg/myvol
```

Этот код создает новый логический том с именем **mybackup**, который является снимком нашего тома **myvol**. Содержимое **/dev/myvg/myvol** в момент выполнения команды `lvcreate` теперь располагается в **/dev/myvg/mybackup**. Оба тома находятся в одной группе томов, поэтому перед созданием резервной копии в группе должно быть обеспечено свободное место. Вам не потребуется полный размер исходного тома, поскольку в снимке хранятся только изменения, сделанные с момента его выполнения. Мы обошлись 1 ГБ, хотя размер **myvg/myvol** — 7 ГБ. В документации LVM рекомендован размер 15–20% от исходного, но он зависит от того, насколько быстро меняются данные и как долго вам нужно хранить снимок. Если снимку не хватит места, он будет испорчен, так что отведите ему побольше.

Когда снимок станет не нужен, вы сможете удалить его так же, как и любой другой том, а освободившееся пространство возвратится группе томов. Сколько места снимок фактически использовал, можно выяснить при помощи `lvdisplay`. Вот грубый пример резервирования рабочего раздела на ленту при помощи снимков:

```
lvcreate --size 500M --snapshot --name var-backup /dev/
```

```
myvg/var
```

```
mount /dev/myvg/var-backup /mnt/tmp
```

```
tar c /mnt/tmp >/dev/tape
```

```
umount /mnt/tmp
```

```
lvremove /dev/myvg/var-backup
```

ЗАГРУЗКА С LVM

100% LVM-систему создать нельзя. Как минимум **/boot** обязан находиться на стандартном физическом разделе, чтобы загружать ядро с поддержкой LVM, да еще необходимы кое-какие программы для инициализации LVM-томов. Следовательно, нужно либо разместить `boot`-раздел на физическом томе, либо записать соответствующие команды на начальный RAM-диск в **/boot**. Fedora Core использует последний подход, то есть все разделы, кроме одного небольшого, могут управляться LVM.

В LVM можно поместить и раздел подкачки (`swap`), но веских причин для этого нет. Если вам понадобится временно увеличить его размер, можете создать логический том и добавить его к существующему разделу:

```
lvcreate --size 4G --name swap myvg
mkswap /dev/myvg/swap
swapon /dev/myvg/swap
```

Ваша собственная процедура резервирования, надеюсь, будет более элегантна, но эта наглядно иллюстрирует идею: снимок создается, используется для резервирования вместо работающей файловой системы, затем удаляется, а пространство возвращается группе томов. Что бы ни вносилось в **/var/log** или **/var/lib/mysql**, пока `tar`-архив пишется на ленту, оно будет проигнорировано — будут резервированы только файлы, имеющиеся в момент выполнения `lvcreate`.

Когда имена конфликтуют

Следует знать, что добавляя после загрузки диск, содержащий LVM-разделы, например USB-диск, вы должны сообщить системе, что они появились. Команды

```
vgscan
```

```
vgchange -a y
```

запускают сканирование групп томов и делают все найденные логические тома доступными.

Если новая группа томов имеет то же имя, что и существующая (совпадение имен томов внутри различных групп роли не играет), возникнет ошибка — это может случиться при подключении диска с другого компьютера. Решение — переименовать одну из групп:

```
vgrename oldvolgroup newvolgroup
```

Это довольно неудобно, поскольку дополнительный диск переименовать нельзя — конфликт имен с ходу прекратит распознавание. Нельзя также переименовать группу томов на вашем главном диске(-ах) при наличии смонтированной файловой системы. В любом случае, перед изменением потребуется отредактировать **/etc/fstab**, чтобы суметь после переименования заново смонтировать логические тома. Обычно наиболее удобный способ — использовать для изменения имен LiveCD, в котором ничего не смонтировано. Если вы дали группам томов осмысленные имена вместо `vgXX`, то шансы конфликтов значительно снижаются, спасая вас от возможных огорчений в будущем.

После присоединения нового диска и выполнения `vgscan/vgchange`, вы сможете просмотреть логические тома при помощи `lvdisplay` и смонтировать как любую другую файловую систему.

Более мощные опции

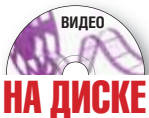
Я описал основные опции, используемые для создания и управления физическими томами, группами томов и логическими томами. Существует еще много других опций для настройки производительности под ваши нужды (более подробно см. **man lvm** и `man`-страницы отдельных команд). Однако предоставленная мною информация вполне достаточна для работы с логическими томами, созданными установщиком дистрибутива, или для создания своих собственных логических томов, и, как я надеюсь, хотя бы частично избавит вас от проблем разметки. **LXF**

СОВЕТЫ



- Не экспериментируйте с тем, что не хотите потерять! Используйте вместо этого переносные или `loop`-устройства.
- Не создавайте логические тома большего размера, чем нужно — при необходимости их легко увеличить.
- Используйте осмысленные имена для томов и групп.
- Добавьте строку фильтрации в `lvm.conf` для отключения сканирования ненужных устройств и снижения времени загрузки.
- Для быстрого просмотра ваших томов используйте `Webmin`.

XDMCP за 10 минут



Бойтесь сокращений длиннее трёх букв? Пол Хадсон познакомит вас с одним из тех, которые никак нельзя пропустить...

ЧТО НУЖНО:

- Fedora Core 5
- Xorg 7.0
- Домашняя сеть

ЧТО ПОЛУЧИТСЯ:

- Соединённые в сеть рабочие столы на каждом компьютере.
- Функционал тонких клиентов задаром.
- Доступ к рабочему столу Linux из Windows и Mac OS X.



Иметь в гостиной супермощный компьютер непрактично – разве что вы не представляете свою жизнь без постоянного шума компьютерной техники. Однако Linux уже довольно давно имеет решение этой проблемы: спрячьте ваш новейший двухпроцессорный мегасервер на антресоли и соедините его по локальной сети со скромной тихой системой в другой части дома.

Используя мощь XDMCP – X Display Manager Control Protocol – можно подключить один компьютер к другому «прозрачно» и использовать рабочий стол с другого компьютера, так, как если бы вы на нём и работали. Если вы когда-либо работали с VNC, то это покажется вам знакомым, однако XDMCP выводит этот принцип на новый уровень, отображая удалённый рабочий стол на весь экран, а не в окне. Дополнительным преимуществом XDMCP является возможность сканирования сети, поэтому вы можете использовать графический браузер и авторизоваться в любом доступном физическом компьютере.

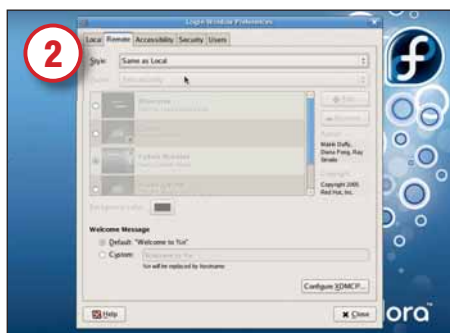
В этом десятиминутном руководстве мы настроим XDMCP на Fedora Core 5, который установлен на двух компьютерах: супермощном сервере и более скромном по производительности клиенте.

Впервые за всю историю *Linux Format* на нашем диске (директория **Magazine**) вы найдёте видеозапись, иллюстрирующую это руководство. Файл закодирован в формате Ogg Theora и должен проигрываться во всех современных дистрибутивах. Если файл у вас не проигрывается, найдите и установите пакет "ogg" для вашей системы.

НА СТАРТ, ВНИМАНИЕ... НАСТРОИТЬ КЛИЕНТ-СЕРВЕРНЫЙ ГРАФИЧЕСКИЙ ИНТЕРФЕЙС!



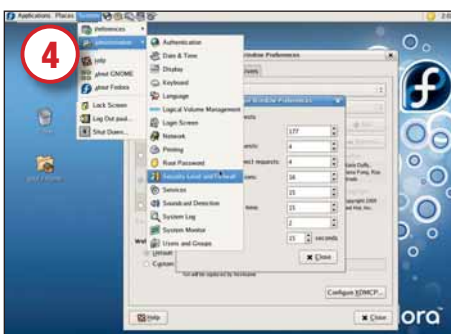
На сервере (у нас ему присвоено имя *hannah*) перейдите в System > Administration > Login Screen. В других дистрибутивах появившееся окно может называться по-другому – в Ubuntu это Login Window, например. Если в меню данного пункта нет, просто дайте команду **gdmsetup** от имени суперпользователя.



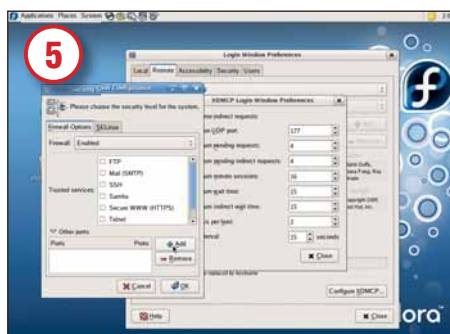
Перейдите на вкладку Remote и переключите пункт Style на значение Same as Local. Это задействует XDMCP и теперь удалённо подключившиеся пользователи увидят тот же экран приветствия, что и локальные пользователи.



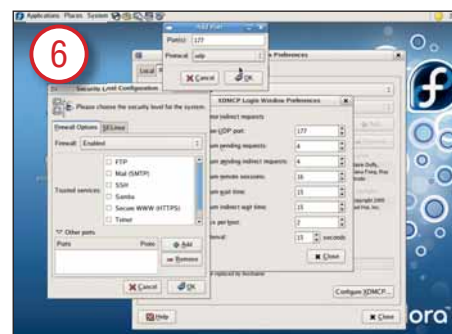
В нижнем правом углу окна нажмите кнопку Configure XDMCP. Теперь проверьте номер порта в пункте Listen on UDP Port – скорее всего, там указан номер 177. Нужно открыть доступ по этому порту в брандмауэре для того, чтобы XDMCP заработал.



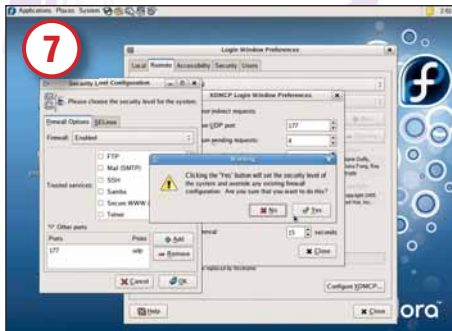
Вперед, на брандмауэр! В Fedora нужные нам настройки находятся в System > Administration > Security Level and Firewall. В других дистрибутивах для настройки брандмауэра вам может понадобиться специальный инструмент, вроде Lokkit.



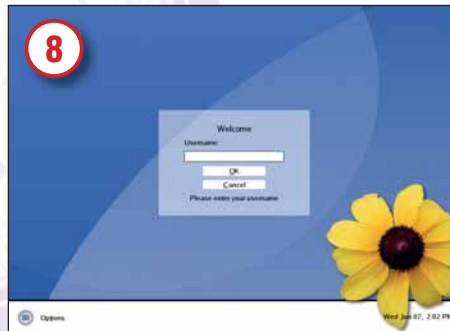
Теперь давайте выберем XDMCP. Fedora предоставляет стандартный список доверяемых сервисов, которые можно включать или выключать с помощью галочек. К сожалению, XDMCP в списке нет, поэтому вам нужно заглянуть в раздел Other Ports и добавить порт, нажав кнопку Add.



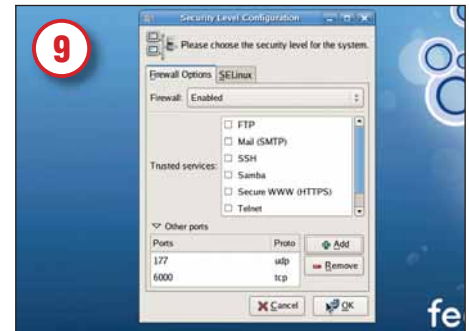
В появившемся окне нам необходимо вписать порт, который будет прослушиваться XDMCP. Запомнив, что Fedora сопоставила XDMCP порт 177, вписываем это число в наше окно и щёлкаем OK.



Возвращаемся к окну настройки уровня безопасности (Security Level Configuration). Нажмите Yes для того, чтобы закрыть диалог и ещё раз Yes, когда Fedora напомнит об изменении настроек брандмауэра.



Теперь мы пересаживаемся за второй компьютер, который на наше счастье тихо работает в гостиной. Машине присвоено имя *susannah*. Вы уже заметили, что мы специально поменяли тему оформления GDM для того, чтобы вам было легче отличать два компьютера. Войдите в систему как обычно.



На втором компьютере зайдите в настройки уровня безопасности и добавьте порт 177 для UDP и порт 6000 для TCP. Последний необходим для сетевого подключения к X-серверу.



На этом настройки компьютера *susannah* закончены. Вы можете выйти из системы обратно к экрану приветствия.



Здесь вам нужно найти пункт Options. Если его нигде не видно, поместите курсор в поле для ввода имени и нажмите F10. У вас должен появиться список опций, среди которых будет Remote Login via XDMCP.



На некоторое время (пока X-сервер перезагружается) ваш экран погаснет, а затем появится окно специального диалога, который сканирует сеть на предмет хостов XDMCP. Дождитесь, пока программа найдет ваш сервер.



Если предыдущее окно не нашло ваш хост (не волнуйтесь, это обычное явление!), подойдите к нему, откройте терминал и дайте от имени суперпользователя команду **ifconfig**. Вы увидите подробный вывод, где найдёте свой IP-адрес.



Возвращаемся к гостевому компьютеру. В окне добавления хостов (Add Host) вводим этот IP-адрес и добавляем его кнопкой Add. Теперь в верхней части окна появится наш хост с указанием точной версии ядра – для того, чтобы вы знали, куда подключились.



Когда вы нажмёте на кнопку Connect, X-сервер ещё раз перезагрузится, но на этот раз на экране появится приветствие удаленного хоста. Для входа используйте имя пользователя и пароль удаленного хоста, а не гостевого компьютера. Получилось! **LXF**



РАЗРАБОТКА 3D-ИГРЫ

Ogre Добавь движения и воды!

ЧАСТЬ 2 Что толку от изменяющегося ландшафта, если вы прикованы к месту? Пол Хадсон знает ответ.

**МЕСЯЦ
НАЗАД**



Мы создали землю и небо. Да, всего на четырех страницах!



В прошлый раз мы начали работать над игрой – кандидатом в самые продаваемые в 2007 году: *Висельник Чед*. Мы поместили нашего героя Чеда Холла на автоматически сгенерированный ландшафт под пустым небосводом. Он готов убежать от полицейских. Но из-за пространственных ограничений мы не дали ему возможности перемещаться по игровому полю – он может только осмотреться вокруг при помощи игрока, держащего мышь. Какой преступник не может убежать от полиции? Только совершенно бестолковый. Значит, надо сделать его подвижным: пусть носится по территории. Итак, ввод с клавиатуры будет преобразовываться в движение по горизонтали и по вертикали в соответствии с рельефом.

Театр Буфф(еризации)

Для начала добавим поддержку стандартных клавиш WASD. Нам самим необходимо перехватывать нажатия клавиш, а затем передавать их в *Ogre*, чтобы он соответственным образом переместил камеру (помните, что это стрелялка от первого лица, поэтому на самом деле мы Чеда не видим). В прошлый раз мы уже обработали клавишу *Escape*, поэтому вы уже примерно представляете, как написать новую часть кода. Чтобы было немного понятнее, я объединил классы *CChadFrameListner* и *CChadGame* в один класс (*CChadGame*). Теперь этот класс занимается и обработкой кадра, и вводом от клавиатуры, а в конце концов будет также обрабатывать механизмы игры – надеюсь, это сделает код более читабельным, ценой легкой объектной инкапсуляции!

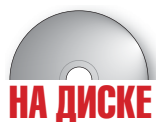
Для начала необходимо буферизовать ввод *Ogre*. Это значит, что если кто-то лупит по клавишам быстрее, чем *Ogre* успевае обрабатывать, то *Ogre* сможет отследить, сколько раз была нажата клавиша, а затем осуществить обработку нажатий как можно скорее – даже в случае, когда клавиша была позже отжата. Например, если некий особо продвинутый игрок путешествует по нашему игровому миру, совершая хитрые трюки и маневры, чтобы побить других игроков, он может превысить число вводов в единицу времени, поддающиеся обработке. Если мы будем буферизовать все, то игрок, может, пальнет и на 10 миллисекунд позже, чем планировал, но все-таки пальнет!

Достигнуть этого можно всего одной строкой кода *Ogre*. Добавьте следующее в файл **chad.h** в конструкторе, после остальных строк:

```
m_InputDevice->setBufferedInput(true, true);
```

Теперь займемся нажимаемыми клавишами – для них потребуется несколько вызовов метода *isKeyDown()*. Создадим вектор, задающий направление перехода (просто линия, соединяющая конечную и начальную точки); мы можем сказать *Ogre*, что хотим прибавить или вычесть смещение по X, Y и Z. Связав эту операцию с нажатием клавиш, мы и получим движение.

Если вы боитесь, что это сложно, расслабьтесь – я представлю вам восхитительные 6 строк, которые выполняют всю работу. Вставьте их перед `return true` в методе `frameStarted()` заголовка `chadframelistner.h`.



• Код примеров статьи

```

Vector3 translateVector = Vector3::ZERO;
if (m_InputReader->isKeyDown(KC_W)) translateVector.z =
-0.1f;
if (m_InputReader->isKeyDown(KC_S)) translateVector.z =
+0.1f;
if (m_InputReader->isKeyDown(KC_A)) translateVector.x =
-0.1f;
if (m_InputReader->isKeyDown(KC_D)) translateVector.x =
+0.1f;
m_Camera->moveRelative(translateVector);

```

Сохраните, наберите **make** и запустите игру – наслаждайтесь новообретенной свободой!

Птица высокого полета

Передвигаясь, вы столкнетесь с тремя большими проблемами:

- Так как ландшафт бугристый, кое-где вы можете вдруг провалиться сквозь землю.
- Направив мышь вверх и шагнув вперед (или направив ее вниз и шагнув назад), вы ни с того ни с сего взлетаете.
- Когда вы упираетесь в границу мира (или взлетаете в небо), ландшафт кончается, и вы видите черноту там, куда небо не достает.

Первые две проблемы можно решить, отслеживая позицию игрока относительно ландшафта и автоматически сопоставляя высоту камеры и высоту ландшафта. Не самым худшим будет чтение файла ландшафта для извлечения точной высоты нашей текущей позиции. Реализация этой идеи сопряжена с трудностями, потому что ландшафт описан как набор точек и их высот, которые интерполируются для создания гладкой поверхности: пусть точка (0,0) имеет высоту 0, а точка (0,1) высоту 1 – все работает гладко, если игрок решит перейти из точки (0,0) в (0,1). Но в стрелялке от первого лица передвижение измеряется не в целочисленных единицах – игрок может перейти в точку (0,0.1239), а для этого значения в файле данных нет.

Поэтому вместо этого мы прочертим линию или луч от места, где находится наша камера, вниз, затем найдем координату, где он пересекается с ландшафтом. Таким образом мы получим высоту ландшафта относительно нашей координатной системы, а затем надо изменить ее на высоту камеры относительно ландшафта (то есть рост игрока).

Это немного сложнее, но *Ogre* предоставляет некоторые необходимые ресурсы. Когда начинается кадр, вызывается метод `frameStarted` (который теперь находится в файле `chad.cpp`, т.к. довольно быстро разрастается!), и мы обрабатываем нажатия клавиш. Передвинув камеру, надо проверить расстояние до земли и сделать соответствующие поправки.

Ogre предоставляет класс `Ray`, экземпляр которого можно создать, задав начало луча и направление. Начало задается просто координатами X,Z нашей камеры (координата Y должна быть довольно большой, чтобы гарантировать, что мы находимся над землей), а направление – вниз. Затем мы создаем объект `RaySceneObject`, который и проведет проверку на пересечение с землей – мы просто передадим объекту луч, созданный ранее, и велит ему выполнить запрос.

По выполнении запроса мы беремся за пересечение (если оно есть) и устанавливаем высоту камеры соответственно высоте ландшафта. Наконец, надо освободить память, занимаемую запросом, с помощью метода `destroyQuery()`. Если вы думаете, что это не так легко, вы правы. Но по крайней мере этим не придется заниматься часто. Добавьте следующий код до выражения `return true` в методе `frameStarted()`:

```

Vector3 campos = m_Camera->getPosition();
Ray camray(Vector3(campos.x, 1000.0f, campos.z), Vector3::
NEGATIVE_UNIT_Y);
RaySceneQuery* query = m_SceneMgr->createRayQuery(
Ray() );
query->setRay(camray);
RaySceneQueryResult &result = query->execute();
RaySceneQueryResult::iterator iter = result.begin();
if (iter != result.end()) {

```

ПОЧЕМУ ВВОД OGRE ТАК НЕСНОСЕН

Позкспериментировав со вводом *Ogre*, вы заметите одну вещь: он капризный, непредсказуемый, простецкий и в целом невыносим. Как случилось, что лучший в мире инструментарий разработки игр имеет столь хромую систему ввода? Ответ прост: *Ogre* не претендует на роль инструментария для игр, занимаясь графикой, графикой и еще раз гра-

фикой – система ввода была дана разработчикам игр исключительно для создания кросс-платформенных демо. Будьте спокойны: как только понадобится более продвинутый вариант ввода, мы тут же оставим систему *Ogre*, но пока ее вполне достаточно для путешествий по пространству таким одиночным облаком.

```

float terrainheight = iter->worldFragment
->singleIntersection.y;
if ((terrainheight + 3.0f) != campos.y) m_Camera
->setPosition(campos.x, terrainheight + 3.0f, campos.z);
}
m_SceneMgr->destroyQuery(query);

```

Перекомпилируйте. Вы увидите, что теперь две первые проблемы решены! Ура!

Водный мир

Наша третья проблема состоит в том, что, добравшись до границы ландшафта, вы видите черную пустоту там, где кончается небосвод, а могли бы созерцать что-нибудь поприятнее. Ситуацию можно поправить, окружив наш остров океаном – пусть вы все еще выходите «за грань», но по крайней мере видите реалистичную воду.

Используем для нашего океана плоскость, на которую мы можем наложить любой материал (так же, как было с небом). Мы сделаем это за пять шагов, с каждым из которых наша вода будет становиться все привлекательнее.

Сначала необходимо создать плоскость, а затем наложить на нее одну из предустановленных в *Ogre* водных текстур. Для создания плоскости используем класс `Plane` и метод `createPlane()`, затем создадим объект *Ogre* и добавим плоскость к сцене. Вот этот код:

```

Entity* OceanEntity;
Plane OceanPlane;
OceanPlane.normal = Vector3::UNIT_Y;
MeshManager::getSingleton().createPlane(
"OceanPlane",
ResourceGroupManager::DEFAULT_RESOURCE_
GROUP_NAME,
OceanPlane,
16000, 16000,
10, 10,
true, 1,
50, 50,
Vector3::UNIT_Z);
OceanEntity = m_SceneMgr->createEntity("water",
"OceanPlane");
OceanEntity->setMaterialName("Examples/TextureEffect2");
OceanNode = m_SceneMgr->getRootSceneNode()->createChildScene
Node("OceanNode");
OceanNode->attachObject(OceanEntity)

```

Также вам понадобится добавить одну строчку в конец `CChadGame` в `chad.h`:

```
SceneNode* OceanNode;
```

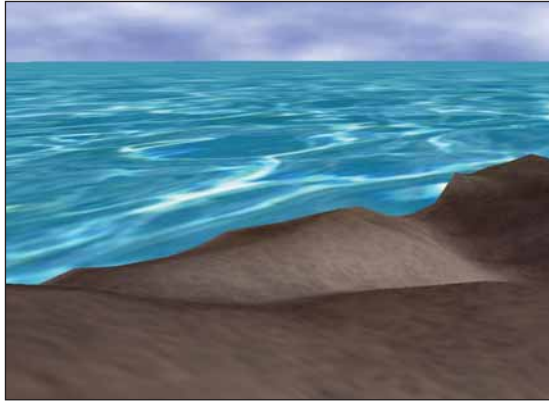
Подойдя к границе острова, вы увидите простирающийся океан, представленной простой текстурой (рис. 1). Для первой попытки неплохо, но пора создать нашу собственную настраиваемую текстуру воды. Создайте новый файл `chad.material` и введите

```

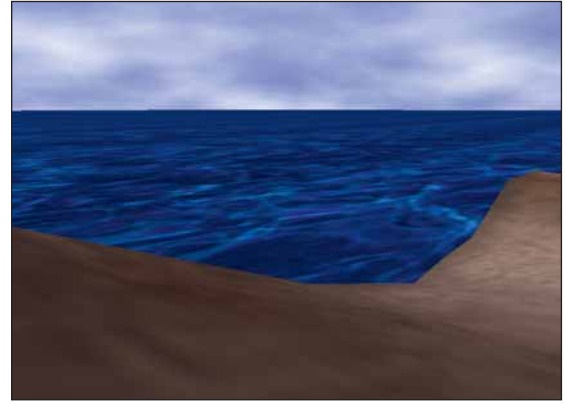
material Chad/Water
{
technique
{
pass
{
ambient 0.2 0.5 0.8

```





1) Пусть наша первая попытка сотворения океана выглядит жуткой фальшивкой, зато она маскирует дыру в небе!



2) По рисунку этого не заметно, но две независимо движущихся текстуры придают воде более естественный вид.



```
texture_unit
{
    texture Water02.jpg
    scroll_anim 0.02 0
}
}
```

Водная текстура та же, что и до этого, но взято меньшее значение `scroll_anim`, чтобы вода двигалась не так быстро. Также устанавливается синий цвет среды, чтобы вода стала более темной (Рис. 2). В `resources.cfg` проверьте, что за строкой `[General]` идет:

```
FileSystem=.
```

Это позволит нам загружать материалы и другие ресурсы из рабочего каталога нашей игры, поэтому теперь можно использовать `chad.material`. Вернемся к файлу `chad.cpp`: найдите строку

```
OceanEntity->setMaterialName("Examples/TextureEffect2");
```

и замените ее на

```
OceanEntity->setMaterialName("Chad/Water");
```

Перекомпилируйте и увидите, что океан стал малость (ну самую малость) реалистичнее.

Следующий шаг – использование материала из двух текстур, одна из которых будет двигаться по прямой, а вторая будет менять свое направление во время движения. Вместе они придают более естественный вид воде и больше реализма игре. В `chad.material` добавьте второй блок `texture_unit` сразу же после первого:

```
texture_unit
{
    texture Water02.jpg
    wave_xform scroll_y sine 0 0.1 0 0.1
}
}
```

Так как мы меняли только файл с материалом, запускать `make` необходимости нет: просто наберите `chad` – и увидите новую воду.

На следующем этапе, специально для любителей горных омутов, поднимем нашу воду над землей так, чтобы она затопила низины ландшафта и образовала бассейны (Рис. 3). Это очень просто – добавьте одну строчку в конец метода `createScene()`:

```
OceanNode->translate(2000, 20, 0);
```

И последнее изменение: добавим воде немного движения, напоминая легкий прилив/отлив. Реализацию можно осуществить множеством хороших, продвинутых и сложных способов – но мы этого делать не будем! На самом деле наша плоскость с водой будет подниматься и опускаться каждый кадр. Некоторые норовят написать для этого не меньше десяти строк кода, хотя на самом деле нужно всего три. Вот первая – добавьте ее под `CChadGame` в файле `chad.h`, но до определения `OceanNode`:

```
double m_OceanFlowNum;
```

Теперь добавьте следующие две строки в конец `frameStarted()` в файле `chad.cpp` перед выражением `return true;`:

```
OceanNode->translate(0, sin(m_OceanFlowNum) / 2000, 0);
m_OceanFlowNum += 0.001;
```

Теперь у нас есть вода, которая использует две независимо движущиеся текстуры, и она имеет определенную высоту, что позволяет образовать небольшие водоемы на острове. В качестве бонуса, мы добавили небольшой эффект текучести, придав сцене динамики. Пока этого более чем достаточно; перейдем к следующей теме...

Берем еще одну планку

Все выглядит неплохо, но на этом уроке я хочу реализовать еще одну вещь: простую инфраструктуру для нашего игрока. Развивать ее мы будем после, а сейчас я хочу представить вам базовый класс игрока, отслеживающий, бежим мы или идем. В данный момент наша камера по нажатию клавиш передвигается на фиксированное расстояние, поэтому надо немного поработать, чтобы класс игрока сам решал, как мы должны передвигаться, в зависимости от состояния (бежим, ползем, прыгает и так далее).

Заинтригованы? Отлично. Создайте новый файл `chadplayer.h` и запишите в него следующее:

```
class CChadPlayer {
public:
    CChadPlayer();
    float getSpeed();
    bool m_IsRunning;
    bool m_IsCrouching;
};
```

Это определение простого класса игрока; позже мы добавим и другие элементы. Сейчас он содержит только метод `getSpeed()`, который возвращает скорость, с которой надо передвинуть игрока, а также две булевы переменные, отслеживающие наше состояние – бежим или ползем. Реализация этого класса, `chadplayer.cpp`, выглядит так:

```
#include "chadplayer.h"
CChadPlayer::CChadPlayer() {
    m_IsRunning = false;
    m_IsCrouching = false;
}
float CChadPlayer::getSpeed() {
    if (m_IsCrouching) return 0.005f;
    if (m_IsRunning) return 0.2f;
    return 0.1f;
}
```

Пока ничего сложного, но, как я уже сказал, на следующих уроках мы расширим его возможности.

Следующим шагом создадим и удалим объект игрока внутри `chad.cpp`, поэтому добавьте эту строку после объявления `m_ViewPort` в `chad.h`...

```
CChadPlayer* m_Player;
```

...затем в `chad.cpp` добавьте строку в начало метода `run()`...

```
m_Player = new CChadPlayer();
...и, наконец, строку в деструктор ~CChadGame() в chad.cpp
delete m_Player;
```

Мы добавили код для игрока, но чтобы игра заново скомпилировалась, необходимо отредактировать Makefile так, чтобы **chadplayer.cpp** оказался в списке сразу после **chad.cpp** в цели **all**.

Теперь надо изменить код, отвечающий за передвижение игрока в методе **frameStarted()**, чтобы он использовал метод **getSpeed()**. Самым простым будет вызвать **getSpeed()** четыре раза, но гораздо лучше использовать промежуточную переменную, например:

```
float playerspeed = m_Player->getSpeed();
if (m_InputReader->isKeyDown(KC_W)) translateVector.z =
-playerspeed;
if (m_InputReader->isKeyDown(KC_S)) translateVector.z =
+playerspeed;
if (m_InputReader->isKeyDown(KC_A)) translateVector.x =
-playerspeed;
if (m_InputReader->isKeyDown(KC_D)) translateVector.x =
+playerspeed;
```

Теперь игра должна компилироваться и запускаться, хотя сейчас вы никакой разницы не заметите, потому что мы не устанавливали значения **m_IsRunning** и **m_IsCrouching**. Приступим к последнему заданию на сегодня.

Идем верным путем

Это сравнительно небольшое изменение, но оно действительно придаст игре достойный вид. Мы собираемся отслеживать нажатие левого **Ctrl** (ползти) и левого **Shift** (бежать). Придется немного поработать, потому что надо расширить класс **CChadGame**, чтобы реализовать класс **KeyListener** (а также классы **FrameListener**, **MouseListener**, **MouseMotionListener**).

Шаг первый: добавьте в проект файл **OgreKeyEvent.h**. Он позволит определять, какая клавиша была нажата. Добавьте следующий код рядом с другими выражениями **#include** в **chad.h**:

```
#include "OgreKeyEvent.h"
```

Ниже вы увидите строку:

```
class CChadGame : public FrameListener, public
MouseListener, public MouseMotionListener {
```

Добавьте после **MouseMotionListener** запятую, а затем **KeyListener**.

Идем дальше вниз – перед строкой **Root* m_Ogre** добавьте эти три строки:

```
void keyPressed(KeyEvent *e);
void keyReleased(KeyEvent *e);
void keyClicked(KeyEvent *e) { }
```

Эти три метода необходимы для реализации класса **KeyListener**. Мы собираемся реализовать первые два метода, когда клавиша нажата и когда отпущена соответственно. Последний метод мы оставим пустым, так как он нам не нужен – этот метод вызывается, когда клавиша сначала нажимается, затем отпускается.

Покончив с файлом **chad.h**, переходим к **chad.cpp**, в котором нам надо реализовать методы **keyPressed()** и **keyReleased()**:

```
void CChadGame::keyPressed(KeyEvent *e) {
printf("Key pressed!\n");
switch (e->getKey()) {
case KC_LCONTROL:
m_Player->m_IsCrouching = true;
break;
case KC_LSHIFT:
m_Player->m_IsRunning = true;
break;
}
}
void CChadGame::keyReleased(KeyEvent *e) {
switch (e->getKey()) {
case KC_LCONTROL:
m_Player->m_IsCrouching = false;
```

ТЕСТИРУЕМ ВОДУ

На каждом новом уроке мы будем увеличивать реализм игры, чтобы игрок все больше втягивался в игровой мир. Но, пытаясь быть реалистичным, вы столкнетесь с проблемой, похожей на ту, что мы получили с водой – что считать реализмом? Какого цвета должна быть вода – синего, зеленого, смешанного или просто прозрачной? Должны ли отдаленные предметы заволакиваться дымкой? Если да, то насколько?

Если вы живете в правильном месте, можете выйти наружу и посмотреть, какие цвета и настрой-

ки использовать для игры; увы, для большинства из нас это роскошь. Но не горюйте: иногда реализм не означает приближения к реальной жизни.

На самом деле необходимо выдерживать стереотип, который сложился у сотен людей, листавших каталоги путешествий. Сделайте вашу воду синей (или зеленой), на собственный вкус, лишь бы она казалась реальной лично вам и не мешала продолжать разработку – только это нам и важно.



Синий-синий, кобальтово-синий – это цвет моря в игре *Висельник Чед*.

```
break;
case KC_LSHIFT:
m_Player->m_IsRunning = false;
break;
}
}
```

Не думаю, что стоит комментировать этот код, но на всякий случай скажу, что **KC_LCONTROL** означает 'код клавиши левого **Ctrl**', а **KC_LSHIFT** означает 'код левого **Shift**'.

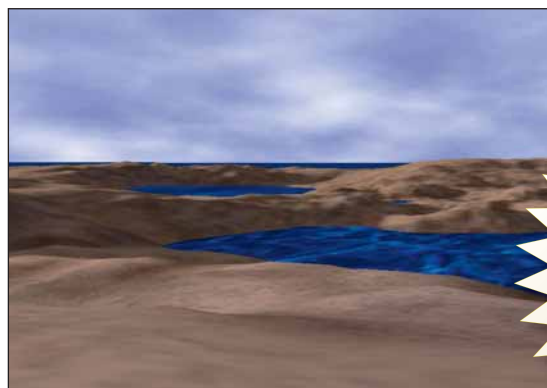
Далее найдите метод **addFrameListener()**. Сразу же за ним добавьте эту строку:

```
m_Ogre->addKeyListener(this);
```

Сохраните все файлы и запустите **make** – теперь вы играете в полноценную игру: можете и бегать, и на четвереньках ползти, наблюдать волны и даже наслаждаться простым обнаружением столкновений. Система ввода **Ogre** хотя и слабовата, но работает, и поэтому мы можем в следующем номере заняться чем-нибудь поинтереснее. Присоединяйтесь! **LXF**

БЛАГОДАРНОСТЬ

Видеокарта Nvidia GeForce 7800, используемая для разработки этого учебника, была любезно предоставлена MSI – спасибо, парни!



3) Затем стали... небольшие водоемы. Наш мир все еще построен как попало, но мы это исправим в *LXF84*!

ЧЕРЕЗ
МЕСЯЦ

Мы улучшим наш ландшафт, еще на шаг приблизившись к законченной игре.

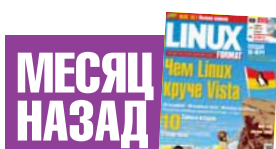


HARDCORE LINUX УЧЕБНИК ДЛЯ ПРОДВИНУТЫХ ПОЛЬЗОВАТЕЛЕЙ

PAM Аутентификация на заказ

Аутентификация пользователей – это не для новичков, но при грамотном применении PAM-модулей можно настроить довольно хитроумные схемы входа в систему – даже используя скромный USB-брелок.

Грэм Моррисон покажет – как.



Дэвид Кулсон рассматривал open source-способ интеграции voip в обычные телефонные системы: *Asterisk*.



Большинство из нас считает аутентификацию пользователя само собой разумеющейся: вводим имя пользователя и пароль, жмём **Enter**, и всё. Но всякий раз, когда вы входите в свою систему – хоть через обычную консоль, хоть с помощью графического менеджера вроде GDM, хоть удалённо, используя SSH – существует фоновая система, проверяющая ваши полномочия, и почти всегда этой системой является PAM, как сокращённо именуются подключаемые модули аутентификации (Pluggable Authentication Modules). Я сказал «почти», потому что Slackware по умолчанию не задействует PAM – насколько мне известно, он один такой среди крупных дистрибутивов. Пользователи же SUSE, Fedora, Ubuntu и Mandriva могут использовать PAM прямо сейчас.

Конкретно мы займемся на данном уроке вот чем: добавим новый модуль аутентификации, который позволит вам входить в систему, не используя ничего, кроме USB-брелка. Такой пример – лучший способ разобраться, как работает PAM и почему это такой замечательный инструмент. Несмотря на сложность этой темы и на тот факт, что вы все-таки читаете учебник для профессионалов, вы обнаружите, что настроить собственный модуль аутентификации с помощью PAM относительно

просто. Мы поместили этот учебник «для профессионалов», так как будем иметь дело с аутентификацией пользователей. С PAM может быть очень сложно работать, и если вы что-то сделаете неправильно, последствия могут быть катастрофическими.

Предостережение

Если вы соблюдаете обычную меру предосторожности, резервируя все редактируемые файлы, при испытании этого USB-проекта беспокоиться особо не о чем. Но всё будет немного иначе, если вы планируете использовать USB-аутентификацию в реальном окружении, как первую линию обороны от пользователей, пытающихся обольстить вашу систему с целью получения неавторизованного доступа. В этом случае единственное требование – убедитесь, что PAM настроен как надо. Нужно также ограничить физический доступ к машине и заблокировать все не в меру вольные сервисы, запущенные на вашей системе. Даже когда пользователи аутентифицируются исключительно при помощи USB-брелка, злоумышленник сможет получить административный доступ к вашей системе, просто нажав кнопку «Reset» и перегрузив ваш дистрибутив в безопасный режим (failsafe mode).

Если вы планируете использовать PAM, чтобы защитить свою систему таким способом, важно также иметь стратегию ограничения физического доступа к вашему компьютеру. Простое решение – запретить компьютер в защищённом помещении и подключить к USB-хабу, запортому в вашем столе. Так могут делать люди, желающие обезопасить свои драгоценные «USB-замки» для программной аутентификации, и вы можете это использовать для PAM-аутентификации. В любом случае, безопасность определяется самым слабым звеном в цепи.



НА ДИСКЕ

• Pam_usb 0.3.3

«ПОДОБНЫЙ ПРИМЕР – ЛУЧШИЙ СПОСОБ РАЗОБРАТЬСЯ, КАК РАБОТАЕТ PAM»

ШАГ 1 – ЗАЧЕМ НУЖЕН PAM

Прежде чем зарываться в настройку PAM, стоит рассмотреть это приложение как таковое. Программа была разработана в Sun Microsystems в 1996 году, с целью навести порядок в широком спектре механизмов аутентификации, использовавшихся в начале 90-х.

Проблема тогда заключалась в том, что не существовало единого унифицированного интерфейса, используемого при аутентификации пользователей. Тем, кто разрабатывал инструменты, требующие входа (login), приходилось вручную связывать свое приложение с необходимым методом (или методами) аутентификации. Хороший пример – Linux-команда login, генерирующая приглашение ввести имя пользователя на виртуальных терминалах. До PAM login нужно было связывать с каждым механизмом аутентификации, удобным автору, а это могли быть столь несходные методы, как шифрованные пароли, биометрия и всевозможные смарт-карты. Разработчикам было нелегко заниматься сопровождением в этой ситуации. Если администратор решал изменить механизм аутентификации, нужно было перенастраивать все приложения, которые его использовали.

Вот почему появился PAM. PAM размещается между тем, что называется сервисами (в основном, login и другие программы «системного входа»), нуждающимися в механизме аутентификации, и самим механизмом. В сервис нужно только встроить поддержку PAM – и употреблять любой метод аутентификации, предусмотренный в PAM. Пока login использует PAM, разработчику больше ни о чем думать не нужно: это забота PAM, и если вам нужно изменить механизм аутентификации для login, ftp или su, PAM – единственное место, где это нужно делать.

И аутентификация всей страны!

Как правило, файлы конфигурации PAM находятся в каталоге `/etc/pam.d` вашей файловой системы. Этот каталог содержит файлы конфигурации всех приложений с аутентификацией через PAM, обычно включающие записи для стандартных сервисов, например, `gdm`, `xdm`, `cups` и даже `screensaver` – почти всех, в которых требуется пароль.

Взглянув на любой конфигурационный файл PAM, вы увидите нечто подобное (взято из `gdm` в SUSE 10.1):

auth	include	common-auth
account	include	common-account
password	include	common-password
session	include	common-session

Каждый столбец этого конфигурационного файла представляет одну из четырёх директив. Это тип интерфейса модуля (первый столбец), флаг управления (второй столбец), путь к модулю (третий столбец) и любые аргументы модуля (в примере `gdm` их нет, поэтому столбец пустой). Интерфейс модуля и сам имеет четыре варианта, каждый из которых показан в нашем примере – **auth**, **account**, **password** и **session**. Модули **auth** будут аутентифицировать пользователей, что обычно означает проверку соответствия имени пользователя и пароля. Затем идут модули **account**, проверяющие правильность учётной записи. **Password**, естественно, проверяет пароль, а **session** обеспечивает выполнение всего, что полагается сделать перед предоставлением сервиса пользователю, например, монтирование пользовательского домашнего или почтового каталога.

Наиболее важный столбец – второй, флаг управления; его-то мы и будем менять в нашем новом модуле, чтобы изменить требования аутентификации, когда вставлен USB-брелок. В нашем куске кода из конфигурационного файла `gdm` все четыре поля установлены в **include**. Это особый случай, поскольку ключевое слово **include** на самом деле указывает PAM направление к другому конфигурационному файлу в

третьем поле (**common-auth**). Этот файл содержит общие правила, пригодные для различных сервисов. Содержимое файла **login** указывает туда же.

Открыв файл `/etc/pam.d/common-auth`, вы увидите нечто вроде следующего (если вы используете старую версию PAM, эта информация будет в файле `gdm`):

auth	required	pam_env.so
auth	required	pam_unix2.so

Когда пользователь входит в систему, модуль аутентификации передаёт сигнал к флагу управления практически тем же способом, как возвращают сигналы процессы и функции. Этот сигнал сообщает о том, что происходило, пока процес или функция исполнялась. Есть сигналы для «успеха» и «неудачи», а есть и сигналы для ошибок сервиса или оповещения, что учётная запись устарела. С новыми версиями PAM вы можете создавать собственные правила для любого из этих условий, это облегчает обработку исключений.

«МОЖНО АУТЕНТИФИЦИРОВАТЬСЯ ЧЕРЕЗ ОТДЕЛЬНЫЙ МОДУЛЬ PAM ИЛИ ЧЕРЕЗ СТЕК ИЗ НЕСКОЛЬКИХ».

Приведённый выше фрагмент включает более типичный флаг управления – **required**. Он говорит PAM, что модуль **auth** обязан завершиться успешно. Менее строгий флаг управления – **sufficient**, означающий, что если аутентификация для этого модуля успешна, системе PAM не нужно продолжать аутентификацию какими-нибудь другими модулями, чтобы предоставить доступ. Другой общий флаг управления – **optional**, который, как подсказывает название, не критичен для предоставления доступа. Причина, почему существуют меняющиеся степени «успешности», заключается в том, что вы можете объединить модули вместе и проектировать аутентификацию на основе «успешности» отдельного модуля или стека из нескольких – именно то, как мы будем настраивать нашу аутентификацию по USB.

>>



Используйте какой угодно USB-брелок для этого урока – только убедитесь, что ваш дистрибутив всегда монтирует это устройство, используя одну и ту же точку файловой системы.

ОБЗОР МОДУЛЯ

Каждый конфигурационный файл PAM использует одну строку из четырех столбцов:

Интерфейс модуля	Флаг управления	Путь к модулю	Опции
auth	include	common-auth	null

ШАГ 2 – УСТАНОВКА МОДУЛЯ USB

Модуль USB, который нам понадобится, разрабатывается отдельно от основного приложения PAM, и пока считается не завершенным, хотя на самом деле полностью функционален. Называется он Pam_usb, и его можно найти на нашем диске или на www.pamusb.org. Последняя версия выпущена в октябре 2005 года, но она работает с ядрами 2.6 и современными дистрибутивами.

Первый шаг – скачать исходный код с сайта Pam_usb. Сборка и инсталляция модуля очень просты: наберите **make**, затем – **make install** с правами **root**, а конфигурационного скрипта здесь нет. Будет установлена пара скриптов **hotplug** и создан каталог **/etc/pam_usb**. Кроме того, в **/usr/bin** будет установлен инструмент администрирования **usbadm**. Скрипт **hotplug** разработан для блокировки сеанса, когда извлекается USB-устройство, используемое для аутентификации.

Сидеть! Место!

Очень важно, чтобы ваше устройство USB всегда обнаруживалось в одном и том же месте. Горячее подключение USB-устройств печально известно в Linux как ненадежное, но многие из последних дистрибутивов, похоже, наконец-то заставляют его работать. Сложность заключается

достоверно находить ваше устройство USB в файловой системе, вам потребуется передать эту информацию в **usbadm**; она автоматически создаст ключи, необходимые для шифрования, чтобы защитить вашу информацию, и скопирует все файлы конфигурации на их законные места в файловой системе и на USB-брелок.

Отданная вами команда должна выглядеть следующим образом – только измените имя пользователя и размещение вашего USB-устройства:

```
usbadm keygen /media/KINGSTON graham 1024
```

В этом примере **/media/KINGSTON** – точка монтирования USB-брелка, **graham** – имя пользователя, а **1024** – размер DSA-ключа шифрования, который должен сгенерироваться. Фактически эта команда размещает открытый ключ в **.auth** в пользовательском каталоге, а закрытый ключ в **.auth** на USB-устройстве. Тем, кто знаком с шифрованием парным ключом, известно, что неважно, кому доступен публичный ключ – но каждый владелец копии закрытого ключа сможет пройти аутентификацию и войти на вашу машину. Другими словами, любая личность, присвоившая ваше USB-устройство...

Укрепляем USB

Конечно, вы наверняка уже обнаружили большую проблему. Запросто можно скопировать приватный ключ на другое устройство и использовать его для аутентификации. Это основное отличие между общим USB-устройством, используемым с модулем Pam_usb, и блокирующей проприетарной смарт-картой, которая запирает закрытый ключ на аппаратном уровне.

Есть два решения этой проблемы. Первая – держать ваш закрытый ключ на USB-устройстве с определенным серийным номером. Модуль Pam_usb может затем проверять этот серийный номер и убеждаться, что приватный ключ не был скопирован.

Если ваше USB-устройство снабжено встроенным серийным номером, вы можете запросить его и добавить его к модулю, набрав **usbadm addserial**, или добавить его вручную, введя **usbadm addserial 12345**. Серийный номер добавляется к **/etc/pam_usb/serials.conf**. Использовать серийные номера таким способом не совсем безопасно – отчасти потому, что программисту относительно легко раскрыть серийный номер, хранящийся в вашем устройстве, а отчасти потому, что так мало USB-брелков действительно снабжены серийным номером. Итак, подумаем над альтернативным решением.

Двойная защита

Каждый отдельный метод аутентификации называется токеном, и каждый имеет сильные и слабые стороны. В случае с приватным ключом, размещенным на USB-брелке, слабостью, очевидно, является безопасность ключа и серьезное неудобство в случае потери USB-брелка. Аналогично, основная слабость парольной аутентификации – атака подбором (brute force): взломщик может пройти по тысячам хорошо известных комбинаций символов в надежде угадать ваш пароль.

Решение, оно же моя альтернатива использованию серийного номера – двухкаскадная аутентификация, задействующая два различных метода одновременно. Здесь безопасность USB-брелка должна играть важную роль. Комбинация приватного ключа, размещенного на USB-брелке, и обычной аутентификации по паролю предоставляет сильный уровень системной безопасности и защищает от наиболее общей атаки на безопасность: подбора пароля. При использовании двух методов аутентификации, даже если пароль будет угадан, потенциальному злоумышленнику всё-таки не обойтись без закрытого ключа с вашего USB-брелка. В PAM это означает стековую аутентификацию, и на следующей странице я покажу вам, как её настроить.

«НАСТОЯТЕЛЬНО РЕКОМЕНДУЕТСЯ,
ЧТОБЫ УСТРОЙСТВА САМИ СЕБЯ
МОНТИРОВАЛИ, НЕ ТО ЗАВЯЗНЕТЕ.»

в том, что модулю Pam_usb необходимо находить устройство USB в одной и той же точке монтирования, когда бы вы ни подключили его к машине.

К тому же, настоятельно рекомендуется, чтобы устройства монтировали себя сами (используя **hotplug**): в противном случае вы завязнете, если не сможете войти на машину и разобраться с хулиганящим USB-оборудованием. Я обнаружил, что и SUSE 10.1, и Ubuntu 6.06 работают без каких-либо дополнительных ухищрений. Современные дистрибутивы монтируют эти устройства согласно их идентификаторам, назначенным изготовителем и определяемым скриптом **hotplug** при опросе USB (вы можете сделать это сами, набрав **lsusb**). Когда вы сможете

```

graham@degville:~$ usbadm keygen /media/KINGSTON graham 1024
[!] Directory /home/graham/.auth/ not found, creating one...
[!] Generating 1024 DSA key pair for graham@degville
[!] Extracting private key...
[+] Private key extracted.
[+] Private key successfully written.
[!] Writing public key...
[+] Public key successfully written.
graham@degville:~$

```

Создание ключа шифрования DSA с помощью **usbadm** очень похоже на генерацию ключа с помощью **GnuPG** – и не удивительно, поскольку оба используют один и тот же вид шифрования.

ШАГ 3 – ВАША КОНФИГУРАЦИЯ PAM

Теперь внимком в детали добавления нового метода PAM-аутентификации. Для начала, нам нужно выбрать сервис, который мы хотим запереть с помощью нашего USB-ключа. Вышеупомянутый *login* – хорошая стартовая точка, отчасти потому, что его легко использовать, а отчасти – потому что не будет большой беды, если вы что-то заблокируете (если, конечно, это не удалённая система, и *login* – не единственный способ подключиться к ней). Скрипт конфигурации PAM для *login* можно найти в **/etc/pam.d/login**; откройте его в своём любимом редакторе (или в нелюбимом, если вам нужно встряхнуться).

Нам нужно проверить, что *Pam_usb* работает, и простейший способ проверки – убедиться, что это единственный метод, требуемый для входа в систему, для чего мы прокомментируем строку, включающую **auth required pam_unix2.so**. Если такой строки нет, вероятно, найдётся строка с директивой **include**, указывающей на файл с именем **common-auth**. Просмотрите содержимое этого файла, и найдите **auth required pam_unix2.so** – но здесь это прокомментировать было бы глупо, потому что тогда любой другой сервис, включающий **common-auth**, не сможет выполнить аутентификацию и блокирует вашу систему.

Вместо этого скопируйте содержимое из **common-auth** в **login**, и прокомментируйте обе строки – **include**, указывающую на этот файл, и строку, заканчивающуюся на **pam_unix2.so**. Затем добавьте **auth require pam_usb.so**; получится нечто вроде

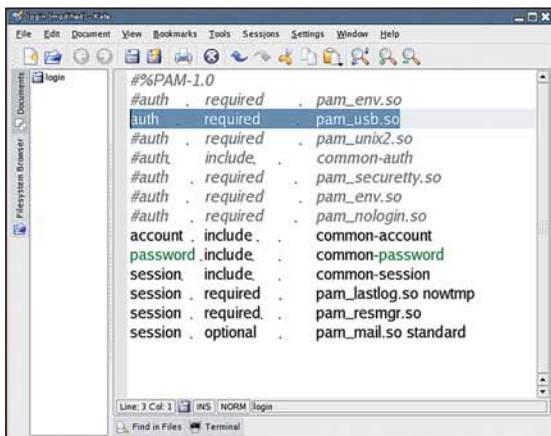
```
auth required pam_usb.so
auth required pam_env.so
#auth required pam_unix2.so
#auth include common-auth
```

Переключитесь в виртуальный терминал, убедитесь, что ваше USB-устройство подключено, и попробуйте войти в систему, используя учётную запись пользователя, которую вы настраивали с помощью пары открытого/закрытого ключей. Если всё пойдёт по плану, пароль у вас не спросят: вместо этого вы окажетесь в системе под вашей учётной записью. Вы даже можете заметить огонёк на вашем USB, вспыхивающий, когда *Pam_usb* считает публичный ключ. На экране вы должны увидеть следующее:

```
Authentication request for user graham from local console (/dev/tty2)
Access granted
```

Это сообщение отображается всякий раз, когда модуль *Pam_usb* выполняет свою работу. Если вы извлечёте ваше USB-устройство, то обнаружите, что обратно вас не пускают, и отображается предупреждение. Попробуйте подключить ваш USB-брелок и попытаться ещё раз.

Ура! Вы только что сумели заставить работать USB-аутентификацию. Но что произойдёт, если вы потеряете свой USB-брелок? Вы не сможете войти в систему. Это катастрофа! Решение – добавить ещё одно правило в стек в конфигурационном файле PAM. Просто прокомментируйте строку с **pam_unix2.so**, и измените директиву для *pam_usb* на **sufficient**:



Это – рабочий файл конфигурации PAM из SUSE 10.1, модифицированный добавкой модуля *Pam_usb*.

PAM ОГРАНИЧИВАЕТ УЧЁТНЫЕ ЗАПИСИ ПОЛЬЗОВАТЕЛЕЙ

PAM также можно использовать для ограничения прав пользователя в вашей системе. Это может быть ограничение на дисковое пространство или даже на допустимый расход ресурсов процессора. Чтобы установить всё это, потребуются отредактировать файл **/etc/security/limits.conf**. Простые правила могут выглядеть следующим образом:

```
* hard core 0
* hard rss 20000
@users hard nproc 100
@lusers hard cpu 10
@users hard fsize 100000
```

Первый столбец определяет круг пользователей, на которых распространяется данное правило. Вы можете использовать символ подстановки *****, чтобы указать всех пользователей, или определить идентификатор группы после символа **@**. В следующем

столбце указывается, жёсткое это ограничение (**hard**) или мягкое (**soft**). (Мягкие ограничения пользователь может менять, а жёсткие – нет). Третий столбец – суть ограничения. Первое ограничение, **core**, не даст пользователю создать никаких core-файлов, потому что параметр установлен в **0**. Второй, **rss**, – это объём резидентной памяти, предоставляемой пользователю для его приложений, тогда как **nproc** – число процессов, которые ему позволено запустить. Четвёртая категория, **cpu**, ограничивает процент тактов процессора, которые пользователь может использовать, а **fsize** – размер любого файла, создаваемого пользователем.

Вам также нужно будет убедиться, что в конфигурационный файл **login**, тот самый, который мы редактировали в основном тексте, добавлена следующая строка:

```
session required /lib/security/pam_limits.so
```

```
auth sufficient pam_usb.so
auth required pam_unix2.so
```

Как мы видели ранее, это означает, что если PAM сможет аутентифицировать пользователя только по USB-модулю, то так оно и будет. В противном случае процедура перекинется на нормальную аутентификацию по паролю с использованием **pam_unix2.so**. Старые версии PAM будут использовать **pam_unix.so**. (Это, вероятно, самое мудрое решение для тех, кто боится потерять свой USB-брелок).

Пароль про запас

Данный способ использования *Pam_usb*, несомненно, предлагает приемлемый уровень безопасности для домашней машины, и удобен тем, что не требует пароля – но если вы действительно бережёте свою учётную запись, вам нужен иной уровень безопасности. Простейший способ сделать это – настроить требование как пароля, так и USB-аутентификации одновременно, пресекая подбор вашего пароля и копирование ключа с USB-брелка.

Для этого просто измените **sufficient** для *pam_usb* снова на **required**, чтобы конфигурационный файл выглядел следующим образом:

```
auth required pam_usb.so
auth required pam_unix2.so
```

Поздравляем – у вас есть работающая система USB-аутентификации, использующая PAM. Просто подойдите к вашей машине и вставьте USB-брелок – правда, круто? Но зачем останавливаться на достигнутом? Для лучшей интеграции с вашим рабочим столом следующий шаг – взять ту же самую процедуру, которую мы использовали для *login*, и сделать такие же изменения с другими сервисами, которые ваша система использует для аутентификации пользователей – хотя бы с графическим менеджером входа в систему GDM. Запуск утилит с привилегиями администратора через *sudo* – тоже хорошее применение для ваших свежобретённых навыков. Просто сделайте такие же изменения в **/etc/pam.d/sudo**, как мы делали в **/etc/pam.d/login**, и пользователи вашей системы смогут использовать *sudo*, только если подключен USB-брелок.

У PAM куда больше возможностей, чем добавление нескольких дополнительных механизмов аутентификации. Теперь, когда вы вооружены идеями насчёт того, как создать собственный стек аутентификации и добавить произвольные модули, почему бы не построить супер-механизм аутентификации на базе той старой шляпы из фольги? **LXF**

ЧЕРЕЗ
МЕСЯЦ

Greasemonkey: расширение Firefox, которое позволит вам покуражиться над чужими сайтами.



PYTHON PRO



PYTHON ДЛЯ ПРОФЕССИОНАЛОВ

Работа с базами данных и web-программирование



НА ДИСКЕ

• Код примеров статьи

ЧАСТЬ 3 Что может быть мощнее связки «база данных + интернет»? А если к этому добавить еще и Python... Чтобы почувствовать все это на практике, погрузимся сегодня в пучины SQL-запросов и HTTP-ответов вместе с **Сергеем Супруновым**.

МЕСЯЦ НАЗАД

Мы разрабатывали клиент-серверные приложения.



Мы уже видели, что *Python* прекрасно подходит для работы с текстом. А что такое интернет-страницы, которые миллионы серверов *Apache* ежедневно миллиардами отдают на растерзание нашим браузерам? По сути, тот же текст, только немножко «гипер»... А значит, если нам нужно будет формировать html-страницу динамически, то *Python* прекрасно с этим справится. И никаких прелестей для разработки на нем CGI-сценариев не существует – web-серверу, по большому счету, безразлично, как именно выполняется скрипт и на каком языке он разработан: лишь бы он умел читать данные из потока ввода и переменных окружения да отдавать текст в стандартный выходной поток.

Впрочем, если вы жаждете скорости, то к вашим услугам *mod_Python*, да и в режиме *FastCGI Python* работать умеет. Но сейчас у нас разговор все же не о настройках CGI, а о *Python*, так что вернемся к тому, ради чего мы эту статью начали.

Постановка задачи

С любым вопросом лучше всего разбираться на практическом примере. Поэтому мы будем «плясать» вокруг несложного и, в общем-то, доста-

точно банального CGI-приложения: гостевой книги. Заодно разберемся с тем, как *Python* взаимодействует с базами данных, где представляется разумным хранить все наши сообщения.

Но прежде чем перейти к рассмотрению кода (вы найдете его целиком на нашем диске), полезно будет дать кое-какую вводную информацию.

Универсальное «междуморье» CGI

CGI (*Common Gateway Interface, общий шлюзовый интерфейс*) был разработан как средство взаимодействия HTTP-сервера с программами, которые могут запускаться в операционной системе. Если говорить упрощенно, то CGI, передавая управление такой программе (обычно их именуют cgi-сценариями, хотя это вполне может быть и двоичный файл, разработанный на C/C++), формирует для нее определенное окружение. В частности, параметры HTTP-запроса, полученного от клиента, могут помещаться в определенные переменные окружения или передаваться cgi-программе как аргументы или как входной поток (STDIN). В ответ HTTP-сервер ждет данные, которые cgi-программа должна выдать в стандартный выходной поток (STDOUT), и передает их клиенту.

Таким образом, все, что требуется от cgi-программы, это способность получать необходимую для работы информацию из формируемой HTTP-сервером среды и возвращать ответные данные, соответствующие протоколу HTTP, чтобы web-клиент знал, что с ними делать.

Учимся посылать

Начнем с формирования HTTP-ответа. Чтобы браузер клиента мог его правильно обработать, он должен состоять из заголовка и тела, разделенных пустой строкой. В заголовке передается необходимая служебная информация, например, тип содержимого, его кодировка, указание браузеру запросить другой ресурс (так называемое перенаправление), и т.д. Простейший cgi-сценарий на языке Python может выглядеть так:

```
#!/usr/bin/Python
# -*- coding: utf-8 -*-
print 'Content-Type: text/html\n'
print '<H3>Если вы это видите, значит все работает</H3>'
```

Первым оператором `print` мы формируем минимально необходимый заголовок – браузер клиента обязательно должен знать, каков тип пересылаемых ему данных (в нашем случае это простой текст, соответствующий формату HTML). Не забывайте о дополнительном переводе строки `\n`, необходимом для отделения заголовка от тела ответа. Ну и далее вы можете передавать любой HTML-код.

Аналогично могут передаваться любые объекты, поддерживаемые клиентом: изображения, звуковые файлы, css-таблицы и т.д. Главное, чтобы значение поля `Content-Type` (именуемое также MIME-типом) соответствовало содержимому.

Здесь играть, здесь не играть...

Однако какой смысл поручать формирование статических, по сути, страниц cgi-сценарию, если сам HTTP-сервер справится с этим намного лучше? В общем-то никакого. Разве что для общего развития... А вот в чем CGI по-настоящему силен, так это в формировании динамических страниц, содержимое которых зависит от информации, переданной пользователем.

Протокол HTTP предусматривает несколько способов передачи информации от клиента на сервер, называемых методами. Наиболее популярные из них – GET, POST, PUT и HEAD.

Метод GET позволяет вставлять информацию в URL, то есть в строку адреса запрашиваемого ресурса. Когда «Яндекс» вернет вам список искомых страниц, посмотрите на адресную строку в браузере – вот так данные и передаются методом GET. Кстати, обратите внимание на то, как все это кодируется, особенно если вы искали какое-то русское слово.

Если на сервер требуется передать большой объем информации, или ее желательно скрыть от любопытных глаз, используется другой метод – POST. В данном случае в заголовке передается лишь размер пользовательских данных, а сами данные пересылаются в теле запроса.

Метод PUT предназначается для размещения ресурсов на сервере и по соображениям безопасности практически не используется. Ну и, наконец, метод HEAD очень похож на GET, за тем исключением, что сервер в ответ на такой запрос возвращает не весь ресурс, а лишь информацию о нем, такую как дата последнего изменения, помещаемую в заголовке. Обычно используется прокси-серверами для определения «свежести» имеющихся у них данных – стоит ли запрашивать ресурс повторно или можно вернуть клиенту то, что есть в кэше.

Определенная сложность для разработчика cgi-сценария заключается в том, что данные, отправленные различными методами, передаются в сценарий по-разному. Так, информация, поступившая с помощью POST, подается на стандартный вход сценария и может быть считана оттуда, например, с помощью `sys.stdin.read(size)` или даже функцией `raw_input()` (хотя во втором случае сложнее контролировать объем принимаемых данных). Количество байт, которые требуется считать, можно получить из переменной окружения `CONTENT_LENGTH` (например, так: `size = os.environ['CONTENT_LENGTH']`).

НЕКОТОРЫЕ РАСПРОСТРАНЕННЫЕ MIME-ТИПЫ

MIME-тип	Описание
text/plain	Простой текст
text/html	HTML-страница
image/gif	Изображение GIF
video/mpeg	Видео-файл в формате MPEG
application/msword	Документ MS Word

Если клиент использует метод GET, то данные поступят в сценарий через переменную среды `QUERY_STRING`. Метод, которым данные переданы (нужно же как-то разобраться, где их искать) можно всегда получить из `REQUEST_METHOD`.

Есть еще один особый случай. Если данные передаются методом GET, но с использованием «индексного» формата, который формируется тегом `<ISINDEX>`, то в этом случае они кодируются не в виде «*переменная=значение&переменная=значение&...*», а как «*значение+значение+...*». И cgi-сценарию они будут переданы, помимо `QUERY_STRING`, через аргументы командной строки, как если бы сценарий вызывался такой командой:

```
script.cgi arg1 arg2 arg3
```

То есть, на этот раз пользовательские данные можно будет получить как `sys.argv[1]` и т.д.

Как видите, огромное число вариантов, предусмотренных CGI-интерфейсом, которые все должны быть учтены при разработке сценария, может вызвать нервный тик даже у опытных программистов, которые и во сне потихоньку набивают по подушке какой-то код. А если еще вспомнить, что данные передаются в закодированном виде (это англичанам хорошо – взял значение переменной и работай, а нам-то с вами это значение вернется в виде `%EC%E4%E0`), да еще и о проверке этих данных нужно позаботиться, чтобы какой-нибудь начинающий хакер не попытался заставить наш сервер работать по-своему... Нет, обо всем этом лучше и не вспоминать. Благо у нас есть модуль `cgi`, в котором все это уже сделано!

Но о нем – чуть позже. Сначала пару слов нужно сказать о HTML-формах.

Формируем формы

Чтобы вам было проще понять рассматриваемый сегодня пример, коротко скажу про то, как же клиент выполняет передачу данных нашему cgi-сценарию. Конечно, продвинутые пользователи могут набрать GET-запрос вручную в адресной строке браузера. Хотя что мелочиться – ведь можно же сформировать и POST-запрос, подключившись телнетом на 80-й порт! Впрочем, обычные пользователи предпочитают более понятные и «осязаемые» способы, например, формы.

Как они выглядят, думаю, каждый знает. Создаются они с помощью тега `<FORM>`, внутри которого добавляются такие элементы, как `<INPUT>` (поле ввода) или `<TEXTAREA>` (многострочный редактор). Этим элементам, если их данные должны быть переданы на сервер, присваиваются имена с помощью атрибута `name`. Начальное значение задается параметром `value` и в дальнейшем для «редактируемых» полей может быть изменено пользователем. Когда пользователь нажимает кнопку «Отправить» (надпись на ней, в принципе, можно изменить), то браузер объединяет все данные полей в пары `name=value`, разделяя их символом `&`. Затем полученная таким образом строка передается на сервер методом, указанным в атрибуте `method` тега `<FORM>`. Путь к сценарию, который будет заниматься ее обработкой, задается атрибутом `action` этого же тега. Если `action` не задан, то данные передаются файлу, сформировавшему текущую страничку.

Если что-то не совсем понятно, обратитесь к коду разрабатываемой гостевой книги, который приведен ниже.



« Наш спаситель – модуль cgi

Возвращаемся к обработке всего этого добра, которое сотни пользователей уже готовы обрушить на наш бедный сценарий. Мы решили воспользоваться стандартными средствами *Python*, и здесь все действительно очень просто – импортируйте модуль *cgi* и, создав объект класса *FieldStorage*, вы получите через него доступ ко всем данным, переданным пользователем, независимо от используемого метода:

```
import cgi
data = cgi.FieldStorage()
for entry in data.keys():
    print 'Переменная %s имеет значение %s' % (entry, data[entry].value)
```

Если вам нужно получить значение определенного поля, это делается так:

```
field = data['field'].value
```

Помимо пользовательских данных, объект класса *FieldStorage* содержит информацию и о полях заголовка (в нашем примере их можно получить из словаря *data.headers*). MIME-тип данных (передаваемый полем заголовка *Content-Type*) можно получить из атрибута *data.type*. Через этот же объект может быть выполнена и загрузка файла.

С помощью методов *keys()* и *has_key()* можно выполнять обработку полученных данных в цикле и проверять наличие той или иной переменной. Кстати говоря, проверять наличие переменной во входных данных, прежде чем приступать к их обработке, нужно непременно – ведь запрос формируется клиентом, а кто знает, что у него на уме?

Базируем данные

Итак, получать данные от клиента мы научились. Отправлять тоже умеем. Осталось придумать, как эти данные лучше всего хранить. Конечно, для несложной гостевой книги с небольшой нагрузкой вполне хватило бы и текстовых файлов. Правда, там есть свои сложности – если сразу пять человек захотят высказать свое мнение о вашей крутейшей домашней страничке, то сценарию придется каким-то образом регулировать доступ к файлу-хранилищу (как минимум, обрабатывать ситуацию, если файл уже открыт на запись другим экземпляром сценария). Но зачем нам все эти головные боли? Если мы так ловко отвернулись от необходимости вручную разбирать HTTP-запросы, то неужели не найдем что-то подходящее на этот раз?

Конечно, найдем! И это «что-то» называется системой управления базами данных (в просторечье – СУБД). Теперь наше дело – отправить запрос и получить ответ. Все остальное – уже не наша забота.

Для этого примера я выбрал в качестве «ответственного» за хранение данных сервер баз данных *PostgreSQL*. Поскольку мы пишем ну очень простую гостевую книгу, то и структура базы будет у нас элементарной – одна таблица с тремя полями: время публикации сообщения, имя автора и, собственно, само сообщение:

```
admin@toshiba:~$ psql
Welcome to psql 8.1.4, the PostgreSQL interactive terminal.

guestbook=# create user "www-data" ncreatedb ncreateuser;
CREATE ROLE
admin=# create database guestbook with owner "www-data";
CREATE DATABASE
admin=# \connect guestbook
Вы подключились к базе данных "guestbook".
guestbook=# create table guestbook (
guestbook(# datum timestamp, author varchar, message varchar);
CREATE TABLE
guestbook=# alter table guestbook owner to "www-data";
ALTER TABLE
guestbook=# \q
admin@toshiba:~$
```

Пожалуй, единственное, что здесь нужно пояснить, это почему базе и таблице мы назначили владельцем пользователя *www-data*. Просто к ним будет обращаться *cgi*-сценарий, работающий с правами HTTP-сервера *Apache*, который, в свою очередь, исполняется от имени данного пользователя [в вашем дистрибутиве он может называться

по-другому, – прим. ред.]. А *PostgreSQL* по умолчанию требует, чтобы имя пользователя в БД совпадало с его системным именем. Мне это кажется достаточно удобным, хотя вы, конечно, можете поступить по-своему.

DB API на страже унификации

Осталось разобраться, как же *Python* взаимодействует с базами данных. Для этого *Python* предоставляет DB API – специальный интерфейс, унифицирующий набор методов, которые будут одинаково работать независимо от того, с какой СУБД мы взаимодействуем. Для работы с *PostgreSQL* нам понадобится модуль *PyPgSQL* (в стандартной поставке его может не оказаться, но ваш менеджер пакетов наверняка будет в курсе, как его установить; кстати, это не единственный модуль – у вас, возможно, будет *PyGreSQL*, который работает ничуть ни хуже и с теми же самыми методами).

DB API определяет стандартные методы работы с базами данных, так что, какой бы модуль вы ни загрузили и с какой бы СУБД ни работали (будь то *MySQL*, *PostgreSQL*, *SQLite* или что-то еще), меняться будет только имя модуля. Главное, чтобы используемый модуль соответствовал DB API. Рассмотрим коротко основные методы:

```
conn = connect(dsn='localhost', user='admin', password='superparol',
database='mydb')
```

Так осуществляется подключение к базе. В зависимости от ситуации, вам может потребоваться указать только нужные параметры (например, имя хоста 'localhost' подразумевается по умолчанию).

```
cur = conn.cursor()
```

Курсоры поддерживаются далеко не всеми СУБД, но для общности в DB API они введены и, в случае необходимости, должны эмулировать модулями сопряжения искусственно. Так что не забывайте отправлять все ваши запросы через курсор.

```
cur.execute("SELECT * FROM mytable")
```

Так выполняется SQL-запрос. Если в строке запроса используются знакоместа *%s*, то вторым параметром передается список переменных значений, причем в SQL-запросе знакоместа не требуется окружать апострофами – модуль сделает это самостоятельно в зависимости от типа переменной.

```
cur.fetchall()
```

Возвращает двумерный список (строки – поля) полученных от СУБД данных. Существуют и другие методы, ознакомьтесь с которыми вы сможете в документации или с помощью знакомой вам функции *dir()* да пары-тройки несложных экспериментов.

Закрепляем на практике

Перейдем к рассмотрению нашего примера. Начнем стандартно – укажем кодировку, подключим нужные модули:

```
#!/usr/bin/Python
# -*- coding: utf-8 -*-

import PyPgSQL.PgSQL as pg
import cgi
```

Далее, определим две функции. Первая будет отвечать за добавление нового сообщения в базу:

```
def addMessage(author, message):
    db = pg.connect(database="guestbook")
    c = db.cursor()
    c.execute("""INSERT INTO guestbook (datum, author, message)
VALUES ('now', %s, %s);""", (author, message))
    c.close()
    db.commit()
    db.close()

print "Content-Type: text/html"
print "Location: ?#form\n"
```

Как видите, все очень даже логично: устанавливаем соединение с БД (поскольку в нашем случае подключение выполняется с именем текущего системного пользователя, то достаточно указать только имя

базы), создаем курсор (в PostgreSQL они не применяются, но они эмулируются каждым модулем, претендующим на соответствие DB API), выполняется запрос, закрывается курсор, фиксируются изменения (PostgreSQL использует транзакции, поэтому выполнение метода `commit()` обязательно, иначе ваши изменения не будут сохранены), и, наконец, закрываем само соединение с базой. В поле `datum` заносим значение встроенной переменной PostgreSQL – `now`, которая каждый раз заменяется текущим значением даты и времени.

Ну и печать заголовка «Location» выполняется для того, чтобы перенаправить пользователя на этот же сценарий, но уже без параметров – мы же должны показать клиенту, что он на самом деле ввел? (Якорь `#form` используется, чтобы автоматически прокрутить страничку на последнее сообщение).

Вторая функция будет отвечать за вывод на экран уже оставленных в книге записей, а также за форму, с помощью которой можно будет добавить и свое высказывание:

```
def showGB():
    db = pg.connect(database="guestbook")
    c = db.cursor()
    c.execute("""SELECT datum, author, message
              FROM guestbook ORDER BY datum;""")
    res = c.fetchall()
    c.close()
    db.close()
```

В этом фрагменте мы выбираем все строки из нашей таблицы данных, сортируя их по дате. Результат сохраняется в переменной `res`, с которой и будем работать. Теперь осталось лишь аккуратно разложить наши данные по табличкам и вывести их на экран:

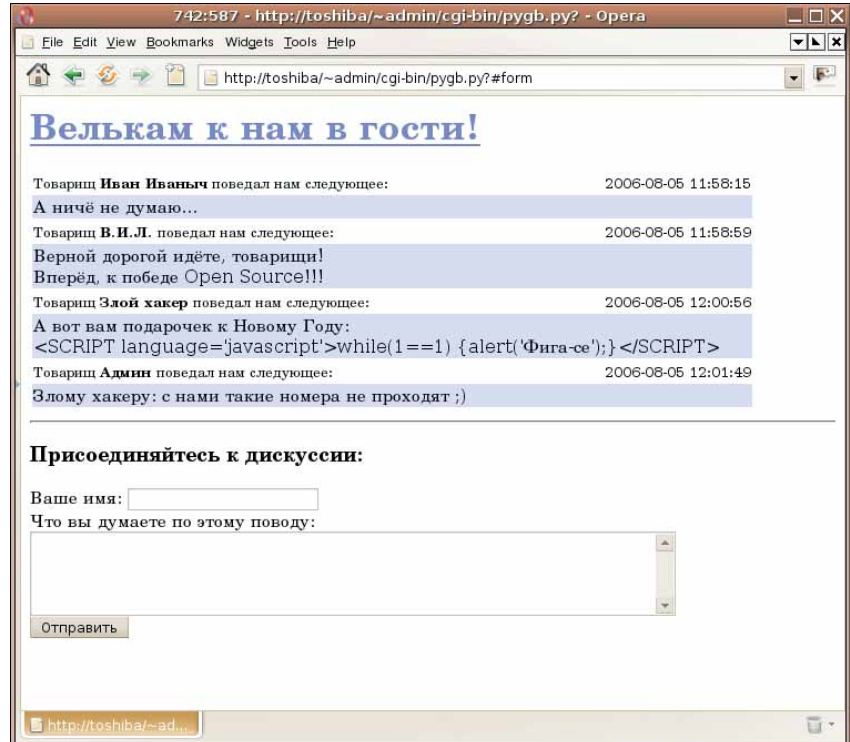
```
print "Content-Type: text/html\n"
print "<H1 style='color:#7777FF'><U>Велькам к нам в гости!</U></H1>"
for item in res:
    print ""<TABLE width='90%'>
        <TR><TD><SMALL>Товарищ <B>%s</B> поведал
        нам следующее:</SMALL>
        <TD align='right'><SMALL>%s</SMALL>
        <TR><TD style='background-color:#DDDDFF' colspan='2'>%s
    </TABLE>"" % (item[1], str(item[0]):19, item[2])
print "<HR><A name='form'><H3>Присоединяйтесь к дискуссии:</H3>"
print ""<FORM method='GET'>
    Ваше имя: <INPUT type='text' name='author'><BR>
    Что вы думаете по этому поводу:<BR>
    <TEXTAREA name='message' rows='5' cols='80'></TEXTAREA><BR>
    <INPUT type='submit' value='Отправить'>
</FORM>""
```

Смысл конструкции `str(item[0]):19` заключается в том, чтобы в строке времени отсечь ненужные нам миллисекунды, которые сохраняются в поле типа `timestamp`. После всех опубликованных сообщений выводим форму добавления нового, чтобы каждый мог присоединиться к нашей дискуссии. Кстати, в теге `<FORM>` мы не указали параметр `action`, поскольку данные будут передаваться на обработку этому же сценарию (благодаря чему имя сценарию можно присвоить любое).

Наконец, последний фрагмент:

```
form = cgi.FieldStorage()
if form.has_key("message") and form.has_key("author"):
    author = cgi.escape(form["author"].value)
    message = cgi.escape(form["message"].value)
    message = message.replace("\n", "<BR>")
    addMessage(author, message)
else:
    showGB()
```

Создаем `FieldStorage`-объект, и если в нем есть заполненные поля `message` и `author` (то есть запрос был сформирован из запол-



ненной пользователем формы), то, немножко их обработав (функция `cgi.escape()` заменяет все «неблагонадежные» символы – например, `<` – их стандартными SGML-сущностями, в данном случае – `<`), передаем функции `addMessage()`. Обработка нужна для того, чтобы злоумышленник не мог ввести в поле сообщения или имени автора что-нибудь такое:

```
<SCRIPT>alert('Да пошли вы все!');</SCRIPT>
```

К слову, пренебрегать проверкой введенных данных ни в коем случае нельзя. Зайдите как-нибудь на securitylab.ru и посмотрите, сколько уязвимостей типа «XSS» обнаруживается каждый месяц! Так что шутки шутками, но последствия могут быть очень серьезными.

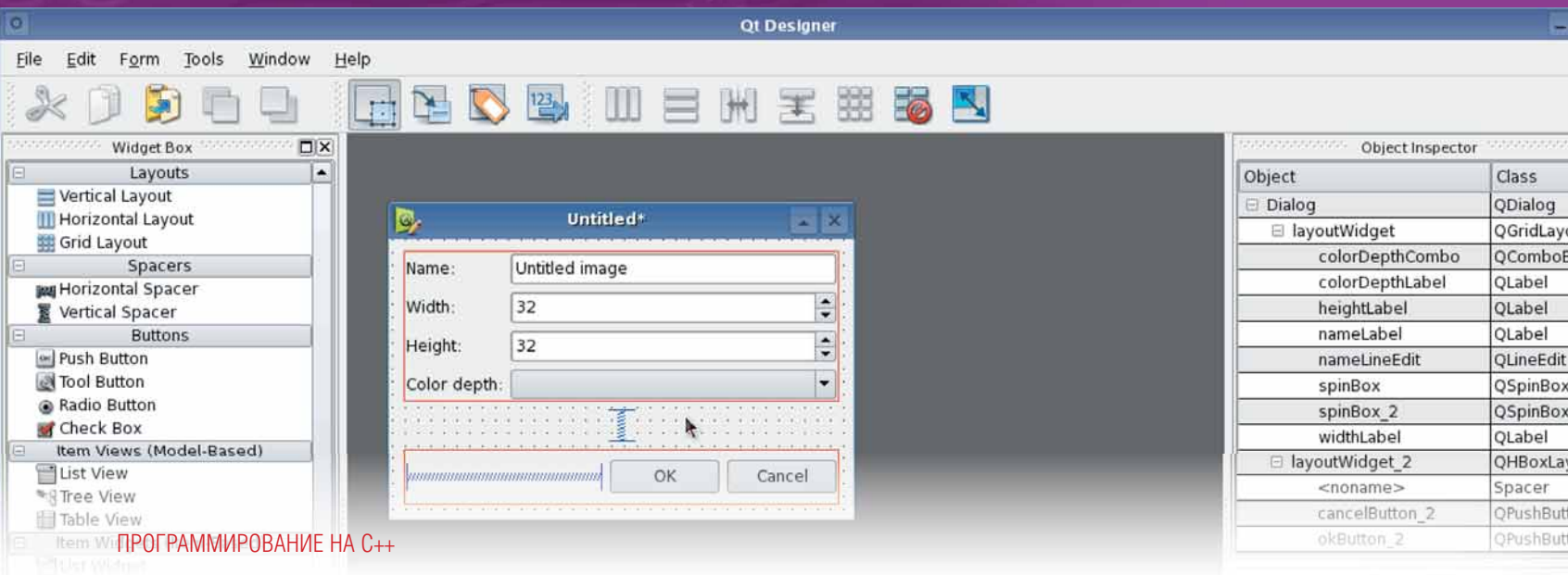
Куда же нам теперь идти?

Итак, что-то вполне работоспособное у нас есть (см. рисунок). Но как вы может догадаться, наша гостевая очень далека от совершенства. Что еще можно сделать? Ну, например, разбить на страницы. Пока сообщений в ней будет не больше дюжины, сойдет и так. А когда их число дойдет до сотни, то редкий пользователь дожидается окончания загрузки всех данных. Можно дать пользователям возможность использовать некоторые HTML-теги, чтобы их сообщения выглядели более красочно. Можно добавить смайликов... А можно даже сделать модуль администрирования, позволяющий редактировать или удалять сообщения, а также отвечать на них. Так что работы непочатый край. Держайте – не буду вам мешать. **LXF**

Ни смайликов, ни BB-кода, ни даже логотипа... Зато мы сделали эту гостевую за 10 минут!

ЧЕРЕЗ МЕСЯЦ

В следующий (теперь уже точно последний) раз мы посмотрим, что можно делать с изображениями с помощью библиотеки PIL. Ну и, может быть, немножко поподем...



Интегрированное совершенство

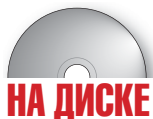
ЧАСТЬ 6 Созданное нами на прошлом уроке приложение было подобно любительским фотоснимкам: вроде бы все есть, но у профессионалов все равно получается лучше. Сегодня **Андрей Боровский** расскажет, что отличает мастера от ремесленника

Если уж я делаю машину времени, то почему бы не придать ей стильный вид?
Доктор Эмет Браун, «Назад в будущее!»

Структуры данных и алгоритмы могут работать вместе только потому, что они ничего не знают друг о друге.
Алекс Степанов (один из архитекторов стандартной библиотеки C++).

МЕСЯЦ НАЗАД

Мы создали «почти настоящее» KDE-приложение.



- Код примеров статьи



Присутствие стиля желательно не только для самодельной машины времени. В прошлый раз мы написали KDE-приложение "images", обладающее всеми основными признаками настоящей программы для малого или домашнего офиса. В этот раз мы займемся шлифовкой программы *images* и добавим в неё некоторые полезные возможности, которыми обладает каждая (ну или почти каждая) программа KDE. Вносить улучшения мы начнем с той области, в которой наличие или отсутствие стиля обычно сразу же бросается в глаза – с интерфейса программы.

В прошлый раз мы конструировали интерфейс программно, таким образом, что элементы интерфейса, соответствующие определенным в программе действиям, добавлялись с помощью создания соответствующих объектов явным образом с последующим вызовом метода `plug()`. В результате интерфейс нашей программы был жестко (хотя и с возможностью настройки пользователем) зашит в ее код. Windows-программисту такое положение дел могло бы показаться естественным, но в Unix этот подход не считается слишком удачным. При жестком «приваривании» интерфейса к коду программы нарушается один из основных архитектурных принципов Unix – принцип разделения «движка» и «оснастки». Согласно этому принципу все части программы, которые можно изменить, не нарушая принципов ее работы, следует вынести за пределы программы и настраивать отдельно. Примеры реализации принципа разделения движка и оснастки можно найти на всех уровнях Unix-системы. На фундаментальном уровне этот принцип выражается в строгом разделении алгоритмов и данных. Код программы должен содержать только алгоритмы, все настройки и данные следует вынести в отдельные файлы. Примером разделения на более высоком уровне является система *X Window*, которая содержит описание механизмов работы графического интерфейса, но не навязывает программам-клиентам какой-либо конкретный набор визуальных элементов (таких наборов существует довольно много и все они, в принципе, независимы от X-сервера). На уровне программирования интерфейса KDE-программы принцип разделения выражается в том, что описание интерфейса (оснастка) может храниться и настраиваться отдельно от самой програм-

мы (движок). Мы уже знаем, что пиктограммы кнопок могут храниться в собственных графических файлах. То же самое касается и других медиа-ресурсов, например, звуковых. Но это не все. Описание визуальных элементов-контейнеров, содержащих другие элементы интерфейса, также может храниться в отдельном файле. Примерами визуальных контейнеров, описание которых может храниться отдельно от двоичного файла программы, являются главное меню, контекстные меню и панели быстрого доступа.

Файлы, хранящие описание элементов интерфейса KDE, имеют расширение `rc`. Если вы откроете один из таких файлов (их можно найти, например, в директориях `$KDEDIR/share/apps/<имя_приложения>`), то увидите, что это файлы в формате XML (файлы `rc` являются частью специальной модели KDE, именуемой `XMLGUI`). Мы научимся создавать файлы описания интерфейса на примере программы *images*. Из метода `images::setup_Controls()` мы удалим все строки, связанные с созданием и отображением меню и панели быстрого доступа. Оставим только вызовы

```
KAction * command = new KAction(...);
```

и

```
command->setToolTip(...);
```

Далее откроем вкладку Automake Manager (по умолчанию она припаркована у правого края главного окна KDevelop), Раскроем группу «Данные в rc» и найдем файл `imagesui.rc` (как нетрудно догадаться, имя файла сгенерировано по шаблону `<имя_приложения>.ui.rc`). Этот файл создается автоматически в процессе генерации проекта, но по умолчанию он пустой и заполнить его мы должны сами. Далее следует текст файла `imagesui.rc`, который можно ввести с помощью текстового редактора:

```
<!DOCTYPE kpartgui SYSTEM "kpartgui.dtd">
<kpartgui name="images" version="1">
<MenuBar>
<Menu name="proc_menu"><text>&Processing</text>
<Action name="bw_command" />
<Action name="contrast_command" />
```

```

<Action name="intensity_command" />
</Menu>
</MenuBar>
<ToolBar>
<Action name="bw_command" />
<Action name="contrast_command" />
<Action name="intensity_command" />
</ToolBar>
</kpartgui>

```

Корневым тэгом документа является тэг `<kpartgui>`. Модификации подлежат атрибуты `name` и `version` этого тэга. Структура главного меню определяется парным тэгом `<MenuBar>`. Тэг `<name>` позволяет задать внутренне имя ниспадающего меню, а тэг `<text>` – название, которое пользователь увидит на экране. Сущность `&` (амперсанд) указывает, как и обычно, подчеркнутый символ для быстрого вызова команды с помощью `Alt`. Для того, чтобы наполнить ниспадающее меню Processing командами мы добавляем серию непарных тэгов `Action`, которые, как вы наверняка догадались, соответствуют объектам `KAction` (действиям) нашей программы. Атрибут `name` содержит имя объекта-действия, соответствующее его имени в коллекции `actionCollection`. Команды будут добавлены в меню в том же порядке, в котором перечислены тэги `Action`. Тэг `<ToolBar>` позволяет нам определить структуру панели инструментов. Формат у этого тэга такой же, как и у тэга `<Menu>`, за исключением того, что у панели инструментов нет видимого имени (а невидимое можно и не указывать, если мы не обращаемся к объекту панели).

На этом работа над описанием интерфейса нашей программы закончена. В прошлый раз нам не требовалось вызывать метод `plug()` для стандартных объектов-действий KDE. Не требуется указывать их и в файле `*ui.rc`. Файлы описания интерфейсов позволяют создавать и другие элементы меню и панелей инструментов – горячие клавиши, всплывающие подсказки, разделители, выравнивание и т.п. Более подробную информацию вы сможете найти на сайте KDE (см. врезку).

Для того, чтобы приложение могло воспользоваться файлом `*ui.rc`, этот файл должен быть размещен в одной из специальных директорий. Программа не увидит неправильно расположенный файл даже если он будет у нее под носом – в рабочем каталоге. Наше KDE-приложение ожидает найти файл `*ui.rc` либо в директории `$KDEDIR/share/apps/<имя_приложения>`, либо в локальной директории `~/.kde/share/apps/<имя_приложения>`. Файл `imagesui.rc` будет автоматически установлен в первую директорию, если вы воспользуетесь командой `KDevelop «Сборка/Установить (с правами root)»`.

Возможно, у вас возник вопрос – а зачем, собственно, все это нужно. В качестве ответа я предлагаю вам вспомнить, как легко можно изменить внешний вид приложения KDE с помощью менеджера тем. Менеджер тем может модифицировать файлы `*ui.rc`, например, припарковать меню Help у правого края окна, как это принято в *Motif*. Несмотря на то, что исходные тексты приложений KDE доступны, интерфейс программы будет проще настраивать, если его описание вынесено в отдельный файл простого формата. Мы можем доработать и метод установки пиктограмм для кнопок панели инструментов.

Программа *images* способна считывать файлы пиктограмм из своей рабочей директории (что она и делала в прошлом примере), однако в KDE существуют специальные правила расположения и наименования этих файлов. Следуя этим правилам, вы не только приобретете репутацию профессионального разработчика, но и позволите другим пользователям настраивать внешний вид панели быстрого доступа с помощью все того же менеджера тем KDE. Сравните простоту изменения облика KDE-приложений, в которых данные об интерфейсе вынесены в легко читаемые файлы, с решением аналогичной задачи для ОС Windows, где программам, модифицирующим интерфейс, приходится получать доступ к приложению различными недокументированными и несанкционированными путями.

Я надеюсь, что я вас убедил. Тогда приступим к редактированию иконок. Имя файла иконки, содержащейся в KDE-проекте, должно начинаться либо с префикса **hi**, либо с префикса **lo**, в зависимости

от количества цветов. За префиксом следует число, указывающее размер иконки – 16 для 16x16, 22 для 22x22 и т.д.

Далее через дефис следует название контекста иконки (смысловой группы, к которой принадлежит иконка) – **action** для иконки обозначающей действие, **app** – для иконки приложения, **mime** – для значка MIME, **device** – для значка устройства. Затем, снова через дефис, следует название самой иконки. Например, пиктограмма для приложения *images* в проекте KDE может носить имя **hi32-app-images.png** (наименование пиктограммы должно совпадать с именем приложения), а иконка для команды «Настроить контрастность» – **hi32-action-contrast.png** (чаще всего иконки хранятся в файлах PNG, но можно использовать и другие растровые форматы, поддерживаемые KDE). Во время установки приложения система модифицирует имя иконки. Например, иконка **hi32-action-contrast.png** переименовывается в **contrast.png** и помещается в директорию `$KDEDIR/share/icons/hicolor/32x32/actions`. Иконки также могут устанавливаться в директории `$KDEDIR/share/apps/<имя_приложения>`, `$KDEDIR/share/apps/<имя_приложения>/pics` и `~/.kde/share/apps/<имя_приложения>`. Для того, чтобы добавить иконку в наш проект, раскройте группу «Данные о пиктограммах» на вкладке Automake Manager (вы, наверное, уже поняли, что *Automake Manager* (рис. 1) – важнейший инструмент *KDevelop* после редактора исходных текстов) и выберите команду контекстного меню «Добавить пиктограмму...». Перед нами откроется диалоговое окно добавления пиктограммы (рис. 2) в котором можно выбрать размеры и тип файла. Если в поддиректории `src` директории проекта KDE не существует файла с выбранным именем, в этой поддиректории будет создан пустой графический файл.

В исходный текст *images* также нужно внести некоторые изменения. В новом варианте программы вызов конструктора объекта-действия `contrast_command` будет выглядеть так:

```

command = new KAction(i18n("&Contrast"),
    QIconSet(BalIcon("contrast")), 0, this,
    SLOT(procContrast()),
    actionCollection(), "contrast_command");

```

Для загрузки графического файла мы используем стандартную функцию `BalIcon()`, которой передаем только имя иконки. Функция сама будет искать соответствующий файл в одной из стандартных директорий (но не в рабочей директории программы). Теперь модифицировать пиктограммы нашего приложения стало не трудно, чем мы и воспользуемся (рис. 3).

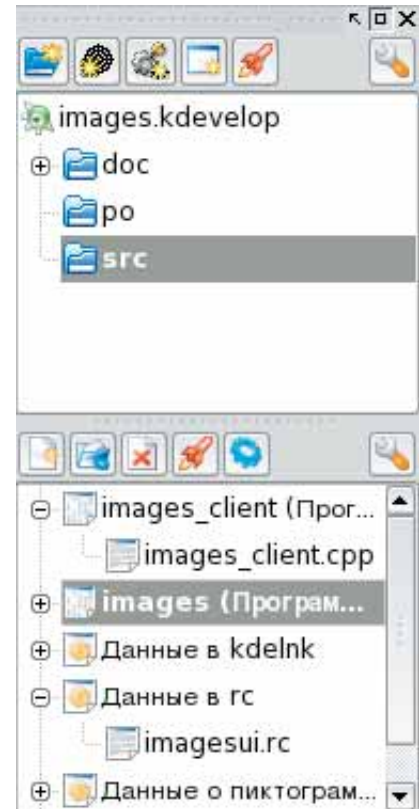


Рисунок 1. Automake manager

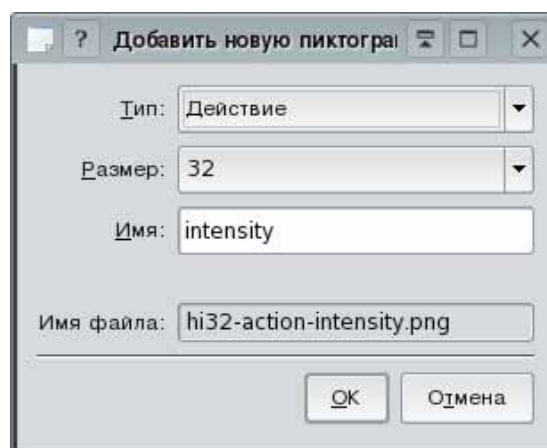


Рисунок 2. Окно добавления пиктограммы

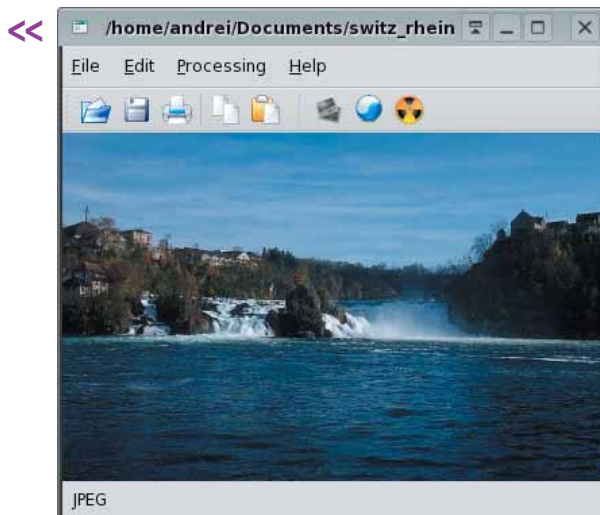


Рисунок 3. Обновленная программа "Images".

Обмен данными с помощью DCOP

Пора закончить издевательства над пользовательским интерфейсом и заняться расширением функциональности *images*. Разработчики объектно-ориентированной оболочки KDE не устояли перед искушением создать объектно-ориентированный механизм межпроцессного обмена данными и автоматизации. Протокол DCOP (Desktop COmmunication Protocol) построен на основе стандартного протокола межпроцессного взаимодействия X Window Inter Client Exchange. В клиент-серверной модели DCOP все программы, с которыми работает пользователь, являются клиентами. Роль сервера DCOP выполняет специальный демон, который осуществляет диспетчеризацию запросов и передачу данных между программами. С точки зрения разработчика модель DCOP напоминает распределенные объектные модели COM и CORBA. Приложение-поставщик сервисов DCOP экспортирует интерфейсы своих классов. Методы этих интерфейсов могут быть вызваны другим приложением посредством специального механизма.

В проекте приложения *images* уже содержатся элементы, необходимые для того, чтобы превратить это приложение в источник данных DCOP. Откройте файл **imagesiface.h**. В этом файле вы найдете объявление класса `imagesiface`, который является потоком класса **DCOPObject**.

```
class imagesiface : virtual public DCOPObject
{
    K_DCOP
public:
    k_dcop:
    virtual void show_Image(const QPixmap data) = 0;
};
```

Класс **DCOPObject** – базовый для всех, экспортирующих интерфейсы DCOP. Макрос **K_DCOP** указывает препроцессору KDE, что данный класс содержит описания интерфейса. Перед заголовками методов интерфейса следует ставить еще один макрос – **k_dcop**. Класс **Imagesiface** экспортирует один метод – `show_Image()`. Данный метод получает объект класса **QPixmap** и выводит переданное в нем изображение в окно программы. Благодаря методу `show_Image()` другие

программы смогут выводить изображение с помощью программы *images*. Поскольку **imagesiface** – всего лишь интерфейс, метод `show_Image()` объявлен в нем как виртуальный метод без реализации (pure virtual, в некоторых кругах, вероятно, сказали бы «чисто виртуальный метод»). Если теперь вы посмотрите на объявление класса **imagesView**, то увидите, что этот класс наследует **Imagesiface**. Именно в классе **imagesView** следует поместить реализацию метода `show_Image()`. Добавить реализацию просто. В объявлении класса **imagesView**, в разделе **public** напишите:

```
void show_Image(const QPixmap data);
```

Теперь щелкните правой кнопкой мыши по этому имени и в контекстном меню выберите команду «Генерировать член класса». В файле **imagesview.cpp** должна появиться заготовка метода, в которую мы впишем одну строчку кода:

```
void imagesView::show_Image(const QPixmap data)
{
    setPixmap(KPixmap(data));
}
```

Как видим, все очень просто. В конструктор класса **imagesView** следует добавить код, регистрирующий классы приложения на сервере DCOP (при этом регистрируются сразу все классы, в которых присутствует макрос **K_DCOP**):

```
DCOPClient * client = kapp->dcopClient();
client->attach();
client->registerAs("images");
```

Прежде всего, нам нужен объект класса **DCOPClient**. Этот объект предоставляется нам главным объектом приложения (объектом класса **KApplication**), к которому мы можем получить доступ из методов любых наших классов с помощью переменной **kapp**. Метод `attach()` объекта **DCOPClient** связывает наше приложение с сервером DCOP, а метод `registerAs()` регистрирует приложение (регистрировать нужно только те программы, которые экспортируют интерфейсы DCOP, но не те, которые только вызывают методы других объектов). Зарегистрированное имя *images* будет использоваться другими приложениями для доступа к нашей программе. Теперь наша программа *images* может быть источником сервисов DCOP.

Для того чтобы проверить механизм DCOP в работе, напишем программу, использующую экспортируемый метод **show_image()**. На диске вы найдете программу *iclient*, созданную в *KDevelop* на основе заготовки **Simple KDE Application**. Логика работы этой программы очень проста. Щелчок кнопки открывает диалоговое окно, в котором нужно выбрать графический файл. Выбранный файл считывается и по протоколу DCOP передается для отображения в предварительно запущенной программе *images*. В конструкторе главного и единственного визуального элемента, – кнопки **KPushButton**, мы связываем программу *iclient* с сервером DCOP:

```
DCOPClient *client = kapp->dcopClient();
client->attach();
```

Программа *iclient* не экспортирует никаких сервисов, так что метод **registerAs()** вызывать не нужно. В обработчике сигнала `clicked`, методе **button_Click()**, мы выполняем загрузку файла и вызываем метод **show_Image()** приложения *images*:

```
void iclient::button_Clicked()
{
    QString fn = KFileDialog::getImageOpenURL(NULL,
        (QWidget *) this, i18n("Open Image")).path();
    QPixmap pixmap;
    pixmap.load(fn);
    QByteArray param;
    QDataStream dataStream(param, IO_WriteOnly);
    dataStream << pixmap;
    DCOPClient * client = kapp->dcopClient();
    if (!client->send("images*", "imagesiface",
        "show_Image(QPixmap)", param))
        kdDebug() << "DCOP error\n";
}
```

ПОЛЕЗНЫЕ ССЫЛКИ

- Описание архитектуры XMLGUI можно найти по адресу: <http://developer.kde.org/documentation/library/kdeqt/kde3arch/xmlgui.html>
- Описание механизмов DCOP содержится по адресу: <http://developer.kde.org/documentation/library/kdeqt/kde3arch/dcop.html>
- Хорошее англоязычное руководство по основам KDE-программирования (с примером DCOP) вы найдете по ссылке: <http://developer.kde.org/~larrosa/tutorial/>

Разбор листинга мы начнем с конца. Метод `send()` класса `DCOPClient` осуществляет обращение к серверу DCOP. Первый параметр метода `send()` – имя приложения-поставщика сервисов DCOP (имя, под которым приложение зарегистрировалось на сервере DCOP). Поскольку сервер по умолчанию добавляет к имени приложения его PID (почему – мы увидим ниже), мы используем «звездочку», чтобы указать, что мы обращаемся к любому (точнее – к каждому) приложению, у которого зарегистрированное имя начинается с "images". Вторым параметром – имя интерфейса, к которому мы хотим обратиться (подчеркнем, что указывается имя интерфейса, а не класса реализации), третьим параметром – имя вызываемого метода, записанное так же, как записывается имя сигнала или слота при вызове `QObject::connect()`. Четвертый параметр представляет собой массив значений параметров вызываемого метода. Значением этого параметра должен быть объект класса `QByteArray`. Для того, чтобы записать значение параметров (в нашем случае – единственного параметра объекта класса `QPixmap`) в `QByteArray`, мы создаем объект класса `QDataStream` (поток данных), указываем этому объекту, что приемник данных – объект `QByteArray`, и с помощью оператора `<<` записываем значение объекта `pixmap` в этот поток.

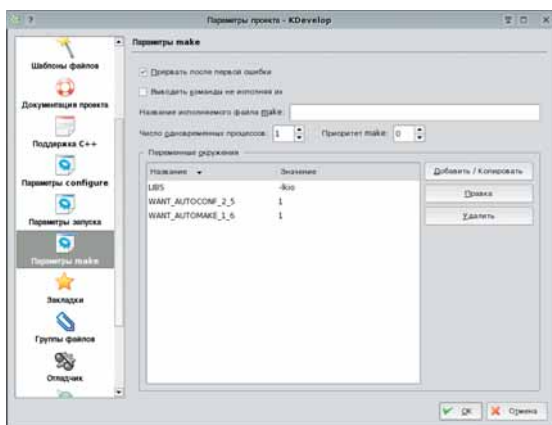


Рисунок 4. Окно «Параметры проекта».

Чтобы скомпилировать программу `iclient`, необходимо добавить в проект ссылку на библиотеку `libki0.so`. Эта библиотека является составной частью KDE, однако по умолчанию она не подключается к проекту Simple KDE Application. Для подключения библиотеки мы воспользуемся диалоговым окном «Параметры проекта» (команда «Проект/Параметры проекта»). В этом окне, на вкладке «Параметры make» (рис. 4), можно указать значения переменных для `make`-файла, в том числе значение переменной `LIBS`. Для подключения к проекту библиотеки `libki0.so` добавим значение

```
LIBS = -lki0
```

Теперь можно компилировать программу.

Что произойдет, если в системе уже запущено какое-то другое приложение с именем `images`? Ничего страшного не случится. Данные получит только тот экземпляр программы по имени `images`, в котором присутствует объект-потомок класса `imagesiface` и метод `showImage(QPixmap)`. Такое совпадение маловероятно, так что даже если на сервере DCOP одновременно зарегистрированы два приложения `images`, данные будут переданы только нашему приложению, но не другому. Сервер передает данные всем экземплярам приложения `images`, в которых существуют соответствующий объект и метод. Попробуйте запустить несколько экземпляров `images`, а затем передайте серверу DCOP изображение с помощью программы `iclient`. Вы увидите, что каждый экземпляр `images` получил копию картинки. Для того, чтобы данные направлялись только одному экземпляру, необходимо указывать полное DCOP-имя приложения, включающее PID. Впрочем, можно регистрировать приложение и не указывая PID. Для этого в тексте конструктора `imagesView` заменим вызов

```
client->registerAs("images");
```

на

```
client->registerAs("images", FALSE);
```

Теперь приложение будет зарегистрировано под именем "images".

Браузер DCOP

Отладка пар приложений, работающих совместно, представляет собой более сложный процесс, чем отладка самостоятельного приложения. При возникновении сбоя в обмене данными между разными приложениями трудно сразу определить, какая из программ виновна в ошибке. При отладке распределенных приложений очень полезно иметь инструмент, представляющий собой эталонную, заведомо верную реализацию клиентской или серверной части. Имея в своем распоряжении такой инструмент, мы можем добиться правильной работы по крайней мере одного компонента системы, а затем использовать этот компонент для отладки второго. В случае отладки приложений, обменивающихся данными с помощью DCOP, мы располагаем эталонным инструментом, позволяющим убедиться в том, что приложение, зарегистрировавшее свой интерфейс на сервере DCOP, работает правильно. Функции отладки нам предоставляет браузер DCOP (DCOP Browser), рис. 5.

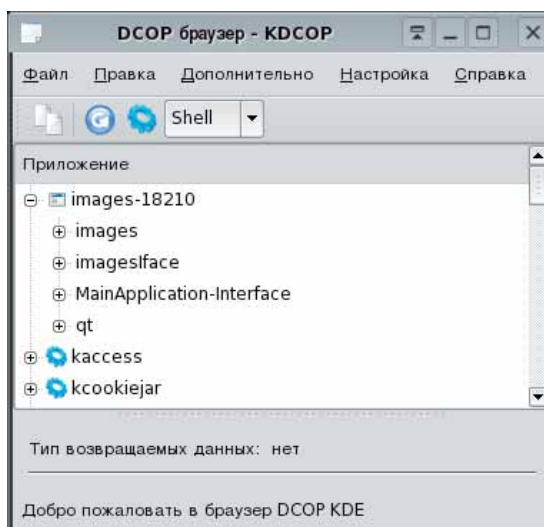


Рисунок 5. Браузер DCOP.

Программу-браузер можно запустить с консоли командой `kdcop`. В окне браузера мы сначала увидим список всех приложений пользователя, зарегистрированных на сервере DCOP. Щелкнув по значку приложения, мы получим список экспортируемых им DCOP-интерфейсов. Если мы запустили приложение `images` до запуска браузера, мы увидим в списке и его имя. Мы также можем запустить интересующее нас приложение после запуска браузера и воспользоваться кнопкой «Обновить». Щелкнув по имени интересующего нас интерфейса (в нашем случае – `imagesiface`), мы получим список его методов. Мы можем использовать эти методы для передачи данных экспортирующему приложению прямо из окна браузера. Если мы щелкнем по имени метода, откроется окно ввода данных для параметров вызываемого метода (число полей в окне ввода будет соответствовать числу параметров). В браузере DCOP мы можем передавать не только строковые и численные значения параметров, но и создавать более сложные структуры данных. Например, для того, чтобы создать экземпляр класса `QPixmap`, нам надо просто указать имя графического файла, из которого будет считано изображение. После того как мы заполним все поля ввода и щелкнем кнопку `OK`, будет вызван соответствующий метод DCOP-приложения с установленными нами значениями параметров. Таким образом, мы можем проверить, действительно ли приложение `images` обрабатывает запросы правильно. Если с `images` все в порядке, мы можем приступить к отладке `iclient`.

Механизм DCOP – не единственный инструмент интеграции приложений KDE. В следующей статье мы рассмотрим другой мощный инструмент интеграции – подключаемые модули. **LXF**

**ЧЕРЕЗ
МЕСЯЦ**

Расширения бывают не только у Firefox! Мы поговорим о том, как использовать подключаемые модули в приложениях KDE



Screenshots



СИМВОЛЬНЫЕ ВЫЧИСЛЕНИЯ

Maxima: укротитель выражений

ЧАСТЬ 3 Сегодня Тихон Тарнавский расскажет вам о возможностях Maxima по упрощению выражений, содержащих рациональные, иррациональные и тригонометрические функции.

**МЕСЯЦ
НАЗАД**

Мы изучали функции
и операторы Maxima



НА ДИСКЕ

- Дистрибутив Maxima и графические оболочки



В этот раз я расскажу о штатных возможностях Maxima по упрощению и прочим преобразованиям выражений. В частности, речь пойдет об автоматическом раскрытии скобок и вынесении за скобки; об упрощении как арифметических действий над некоторыми элементами, так и выражений с участием степенных, показательных и логарифмических функций; а также об обработке тригонометрических выражений. Все эти функции призваны облегчать читаемость математических формул и повышать простоту их восприятия, а посему стоит уделить этому уроку достаточно внимания: при верном использовании данные манипуляции позволят сэкономить в процессе работы значительное количество времени.

Выражаясь рационально...

Существенная часть интересующих нас сегодня функций предназначена для преобразования рациональных выражений. Напомним, рациональным называется выражение, состоящее только из арифметических операторов и возведения в натуральную степень; естественно, элементы такого выражения могут содержать и неарифметические и нестепенные функции – тогда такие элементы с точки зрения рационального выражения считаются атомарными, т.е. неделимыми и непреобразуемыми.

Функции, работающие с рациональными выражениями, описаны в разделе документации «Polynomials»; потому как рациональные функции с математической точки зрения рассматриваются как расширение многочленов (полиномов) – примерно так же, как рациональные числа считаются расширением целых (многочлены, кстати, тоже иногда называют целыми функциями; хотя общий математический смысл этого термина несколько шире).

Имена всех функций Maxima по обработке рациональных выражений содержат буквосочетание *rat*, но не от слова «крыса», а от слова «rational». И начнем мы знакомство с ними с функции, которая так и называется: *rat(выражение)*. Эта функция преобразовывает рациональное выражение к так называемой канонической форме (Canonical Rational Expression, CRE). То есть раскрывает все скобки, затем приводит все к общему знаменателю, суммирует и сокращает; кроме того, приводит все числа в конечной десятичной записи к рациональным.

$$(\%i1) \text{ rat}\left(\frac{(x-1)^2}{x^2+x} + \frac{1}{x+1} + 0.25\right)$$

'rat' replaced 0.25 by 1/4 = 0.25

$$(\%o1) \frac{5x^2 - 3x + 4}{4x^2 + 4x}$$

Тут надо заметить, что атомарные элементы, т.е. символы и числа, в канонической форме рационального выражения в Maxima имеют другое внутреннее представление. При работе в интерфейсах Maxima и xMaxima об этом напоминает приписка *!R* после имени ячейки вывода (в *wxMaxima* и *TeXmacs* такая приписка отсутствует). При этом внешне, на видимом пользователю уровне, каноническая форма ничем, кроме

этого обозначения, от общей не отличается. Но один достаточно интересный момент здесь есть: если каноническая форма рационального выражения используется в других рациональных выражениях, то последние также автоматически приводятся к канонической форме:

$$(\%i2) \% - \frac{1}{x}$$

$$(\%o2) \frac{5x - 7}{4x + 4}$$

Это может быть достаточно удобно, если вам нужно пошагово проделать большое количество рациональных преобразований: вы можете, один раз вызвав *rat()*, сослаться на предыдущие ячейки и благодаря этому далее автоматически видеть на каждом шаге итоговое выражение в канонической, а значит, достаточно компактной и удобной для восприятия, форме. Если на каком-то этапе такое поведение станет вам мешать, вы можете вернуть выражение из канонической к общей форме с помощью функции *ratdisrep(выражение)*. Кроме того, каноническая форма автоматически «отменяется» и в случае любых преобразований, не являющихся рациональными:

$$(\%i3) \text{ exp}(\%)$$

$$(\%o3) e^{\frac{5x-7}{4x+4}}$$

Здесь, хотя *%o2* было выражением в канонической форме, *%o3* – уже выражение общего вида, так как оно не является рациональным.

Скажем пару слов о приведении конечной десятичной записи чисел к рациональной. Конечная десятичная запись считается по определению приближенной, что и понятно, т.к. при вычислениях самой Maxima такая запись может возникнуть исключительно при применении приближенных методов либо при ручном указании о переводе числа в десятичную запись из математической, в результате чего результат тоже, вероятнее всего, окажется приближенным. Эта приближенность учитывается и при переводе в рациональные числа, а ее уровень, то есть мера, на которую рациональное число при переводе может отклониться от конечной десятичной записи, регулируется переменной *ratepsilon*, равной по умолчанию *2.0e-8*, т.е. *0.00000002*. Если такое положение вещей вас не устраивает, вы можете убедить Maxima оставлять десятичную запись чисел как есть, установив в *true* значение флага *keepfloat* (по умолчанию он равен *false*).

Следующая функция раскрывает скобки в рациональном выражении и называется *ratexpand()* (одно из значений слова *expand* и есть «раскрыть скобки»). Здесь также действует опция *keepfloat*. Кроме нее, есть еще одна опция – *ratdenomdivide*; по умолчанию она установле-

на в `true`, что приводит к тому, что каждая дробь, в которой числитель является суммой, распадается на сумму дробей с одинаковым знаменателем. Если же сбросить эту опцию в `false`, тогда все дроби с одинаковым знаменателем будут, напротив, объединены в одну дробь с числителем в виде суммы числителей изначальных дробей. То есть внешне результат будет в этом случае выглядеть почти так же, как и у функции `rat()`; к тому же единственная видимая пользователю разница проявляется только в рациональных выражениях от нескольких переменных (или различных иррациональных выражений). Заключается эта разница в том, что после `ratexpand()` и в числителе, и в знаменателе дроби все скобки будут раскрыты, в случае же `rat()` слагаемые, где присутствуют, скажем, две переменных, будут сгруппированы, и одна из них будет вынесена за скобки (в документации такая форма записи называется «рекурсивной» (recursive):

```
(%i1) ratexpand((x+1)^2*(%e^x-1)/(x+2)(2x+1), ratdenomdivide: false
```

```
(%o1) (x^2 e^x + 2 x e^x + e^x - x^2 - 2 x - 1) / (2 x^2 + 5 x + 2)
```

```
(%i2) rat(%)
```

```
(%o2) (x^2 + 2 x + 1) e^x - x^2 - 2 x - 1 / (2 x^2 + 5 x + 2)
```

```
(%i3) ratexpand(%)
```

```
(%o3) (x^2 e^x / (2 x^2 + 5 x + 2) + 2 x e^x / (2 x^2 + 5 x + 2) + e^x / (2 x^2 + 5 x + 2) - x^2 / (2 x^2 + 5 x + 2) - 2 x / (2 x^2 + 5 x + 2) - 1 / (2 x^2 + 5 x + 2))
```

Кроме того, разница, конечно, заключается и во внутреннем представлении: с точки зрения программы, после `ratexpand()` выражение будет по-прежнему общего вида. Соответственно и все результаты дальнейших рациональных действий с выражением не будут автоматически «канонизироваться». Я специально обращаю ваше внимание на схожесть между результатами этих двух различных функций, поскольку в документации эта схожесть никак не обозначена: в описании обеих функций и примерах к ним нет вообще никаких ссылок друг на друга.

Помимо флага `ratdenomdivide`, есть также функция, собирающая воедино дроби с одинаковыми знаменателями; зовут ее `combine()`:

```
(%i4) combine(%)
```

```
(%o4) (x^2 e^x + 2 x e^x + e^x - x^2 - 2 x - 1) / (2 x^2 + 5 x + 2)
```

В дополнение к функции `ratexpand()` есть также флаг `ratexpand`, который по умолчанию равен `false`, а будучи установлен в `true`, приводит к тому, что все рациональные выражения в канонической форме отображаются и преобразовываются к общему виду сразу же с раскрытыми скобками:

```
(%i5) ratexpand: true %o2
```

```
(%o6) (x^2 e^x / (2 x^2 + 5 x + 2) + 2 x e^x / (2 x^2 + 5 x + 2) + e^x / (2 x^2 + 5 x + 2) - x^2 / (2 x^2 + 5 x + 2) - 2 x / (2 x^2 + 5 x + 2) - 1 / (2 x^2 + 5 x + 2))
```

Обратите внимание, что при применении этого флага выражение сохраняет каноническую форму.

Действует в этом случае и флаг `ratdenomdivide` (напомню, что в строке `%i1` этот флаг был установлен локально, используя сокращенную запись функции `ev()`):

```
(%i7) ratdenomdivide: false %o2
```

```
(%o8) (x^2 e^x + 2 x e^x + e^x - x^2 - 2 x - 1) / (2 x^2 + 5 x + 2)
```

Иными словами, флаг `ratexpand` по своему действию аналогичен

одноименной функции, но действует он на все без исключения канонические рациональные выражения и при этом оставляет их в канонической внутренней записи и изменяет только внешнее отображение этой записи, сохраняя при этом и дальнейшую автоматическую «канонизацию».

...И не только рационально.

Помимо `ratexpand()` есть также и функция «просто» `expand()`. Различий между ними несколько, наиболее принципиальные таковы. Во-первых, `ratexpand()` раскрывает только рациональное выражение «верхнего уровня», все же подвыражения, не являющиеся рациональными, обрабатываются как атомарные, то есть внутрь них она не залезает; `expand()` же раскрывает скобки на всех уровнях вложенности:

```
(%i1) ratexpand((a+b)^(2-x)(x+2)+x^2)
```

```
(%o1) (b+a)^x^2+(2-x)(x+2)
```

```
(%i2) expand(%);
```

```
(%o2) b^4 + 4 a b^3 + 6 a^2 b^2 + 4 a^3 b + a^4
```

Во-вторых, `ratexpand()` приводит дроби-слагаемые к общему знаменателю, а `expand()` этого не делает; в-третьих, на функцию `expand` не действует переключатель `ratdenomdivide`:

```
(%i1) ratdenomdivide: false $ expand((x-2y)^4/(x^2-4y^2)^2+1)
```

```
(%o2) (16 y^4 / (16 y^4 - 8 x^2 y^2 + x^4) - 32 x y^3 / (16 y^4 - 8 x^2 y^2 + x^4) + 24 x^2 y^2 / (16 y^4 - 8 x^2 y^2 + x^4) - 8 x^3 y / (16 y^4 - 8 x^2 y^2 + x^4) + x^4 / (16 y^4 - 8 x^2 y^2 + x^4) + 1)
```

```
(%i3) ratexpand((x-2y)^4/(x^2-4y^2)^2+1)
```

```
(%o3) (8 y^2 + 2 x^2) / (4 y^2 + 4 x y + x^2)
```

И в-четвертых, `expand()` не преобразовывает к рациональным числам конечную десятичную запись – опять-таки, вне зависимости от флага `keepfloat`.

Функция `expand()`, в отличие от своего рационального сородича, имеет несколько вариаций – в виде отдельных функций с похожими названиями, которые раскрывают скобки несколько по-разному. Первую мы уже рассмотрели. Вторая называется `expandwrt(выражение, x, y, ..., v)`, где `wrt` расшифровывается как «with respect to...», то есть «относительно...». Она раскрывает скобки не везде, а только относительно тех символов, которые заданы в списке аргументов после выражения. Другими словами, только там, где из скобок можно вынести хотя бы один из перечисленных символов: >>>

```
(%i1) (x+y)(y+z)+x(v+w) $
```

```
(%i2) combine(expand(%))
```

```
(%o2) (y z + x z + y^2 + x y + x(w+v)) / (v z + v y + v x)
```

```
(%i3) combine(expandwrt(%o1, x))
```

Warning - you are redefining the Maxima function intersection

```
(%o3) (y(z+y)+x(z+y)+x(w+v)) / (v(z+y+x))
```

```
(%i4) combine(expandwrt(%o1, z))
```

```
(%o4) (y+x)z+y(y+x)+x(w+v) / (v(z+y+x))
```


« (На предупреждение, возникающее при первом вызове функций `expandwrt*`(), можете не обращать внимания – на функционале, о котором идет речь, оно никоим образом не отражается.)

Если в выражении встречаются дроби, то по умолчанию эта функция раскрывает скобки только в их числителях, оставляя знаменатели в покое. Изменить это поведение можно переключателем `expandwrt_denom`, установив его в `true` (по умолчанию он равен `false`):

(%i5) `combine(expandwrt(%o1,z),expandwrt_denom:true`

(%o5)
$$\frac{(y+x)z + y(y+x) + x(w+v)}{vz + v(y+x)}$$

И, наконец, последняя функция из этого семейства – `expandwrt_factored`(выражение, x, y, ..., v) – раскрывает скобки лишь в тех слагаемых, где упомянутые символы встречаются не в одном, а в каждом из сомножителей:

(%i1) `expandwrt((x+a)(b+y),x)`

Warning - you are redefining the Maxima function intersection

(%o1) `x(y+b)+a(y+b)`

(%i2) `expandwrt_factored((x+a)(b+y),x)`

(%o2) `(x+a)(y+b)`

(%i3) `expandwrt_factored((x+a)(b+y),x,b)`

(%o3) `xy+ay+bx+ab`

(%i4) `expandwrt((x+a)(b+y),x,b)`

(%o4) `xy+ay+bx+ab`

Раскрытием возведения в целую степень можно управлять как в контексте функции `expand()`, так и отдельно. В первом случае применяются переменные `maxpsex` и `maxnegex`, определяющие соответственно максимальные положительный и отрицательный показатель степени, которые будут раскрываться этой функцией. По умолчанию оба параметра равны `1000`. Переопределить их можно не только глобально, но и в контексте одного конкретного вызова функции `expand()` – в таком случае это делается с помощью дополнительных аргументов, задаваемых после выражения:

(%i1) `combine(expand((a+b)^2+(a+b)^4/(a-b)^6+(a+b)^3,3,3))`

(%o1)
$$\frac{(b+a)^4}{-b^3+3ab^2-3a^2b+a^3} + \frac{b^2+2ab+a^2}{(a-b)^6}$$

В противовес `maxpsex` и `maxnegex` можно задать максимальные положительную и отрицательную степени, которые будут раскрываться автоматически, без вызова функций группы `expand`. За это отвечают переменные `expor` и `expon`, и по умолчанию они равны нулю, то есть автоматически степени не раскрываются вообще.

Кроме самостоятельной функции `expand()`, существуют также флаги `expand` и `expand(p, n)` у функции `ev()`. Запись `выражение, expand` равносильна `expand(ev(выражение))`, а `выражение, expand(p, n)` – `expand(ev(выражение, p, n))`.

Возможности управлять раскрытием скобок на этом не заканчиваются. Еще одна функция – `distrib()` – представляет как бы облегченный вариант `expand()`. Она действует аналогично `expand()`, но только на один уровень в глубину:

(%i1) `((a+b)c+(b+c)d)(x+y)$`

(%i2) `distrib(%)`

(%o2) `(c+b)dy+(b+a)cy+(c+b)dx+(b+a)cx`

(%i3) `expand(%o1)`

(%o3) `cdy+bdy+bcy+acy+cdx+bdx+bcx+acx`

В противоположность функциям `*expand*`(), раскрывающим скобки, можно также и разложить выражение на множители, то есть макси-

мально выносить все за скобки. Делается это с помощью функции `factor()`:

(%i1) `factor(x^2-1)`

(%o1) `(x-1)(x+1)(x^2+1)(x^2-x+1)(x^2+x+1)(x^4+1)(x^4-x^2+1)(x^8-x^4+1)`

Если функции `factor()` передать целое число, она разложит его на простые множители; если же передать рациональное число – на множители будут разложены его числитель и знаменатель:

(%i1) `factor(2^100-1)`

(%o1) `3^5 11 31 41 101 251 601 1801 4051 8101 268501`

(%i2) `factor(123456789/987654321)`

(%o2)
$$\frac{3607 3803}{17^2 379721}$$

Если многочлен не может быть представлен в виде произведения нескольких сомножителей, его можно попытаться преобразовать в сумму таких произведений с помощью функции `factorsum()`:

(%i1) `factorsum(4yz+4xz+y^2+xy-vw-uw+tw)`

(%o1) `(y+x)(4z+y)-(v+u-t)w`

Функция `factorsum()` умеет раскладывать на множители только независимые слагаемые, то есть такие, которые не содержат одинаковых переменных. Если мы раскроем скобки в выражении, содержащем в двух разных местах один и тот же символ, то так как коэффициенты при этом символе после раскрытия сгруппируются, `factorsum()` не сможет понять, каким именно образом разгруппировать их обратно:

(%i1) `expand((x+y)^2+(x+z)^2)`

(%o1) `z^2+2xz+y^2+2xy+2x^2`

(%i2) `factorsum(%)`

(%o2) `z^2+2xz+y^2+2xy+2x^2`

Нужно заметить, что функции `factor()` и `factorsum()`, хотя и не имеют в имени приставки `rat`, все же ведут себя в смысле разбора передаваемых им выражений не как `expand()` и сопутствующие, а как `ratexpand()`; то есть на любой не-рациональной функции останавливаются и внутрь не идут:

(%i1) `factor(log(x^2+2x+1))`

(%o1) `log(x^2+2x+1)`

(%i2) `log(factor(x^2+2x+1))`

(%o2) `2log(x+1)`

Впрочем, об этом можно догадаться из документации, так как функции `factor*` описаны не в разделе `Simplification`, куда относятся `expand*`, а, так же, как и `rat*`, в разделе `Polynomials`.

Выносить за скобки, а также раскрывать эти скобки можно не только специальной функцией, но и дополнительным флагом ко все той же канонической форме рациональных выражений. Флаг этот зовут `ratfac`, и по умолчанию он равен `false`, то есть вынесение за скобки не происходит. Если же его установить в `true`, то в каждом рациональном выражении, приведенном к канонической форме, все будет максимально вынесено за скобки, но без вызова функции `factor()`; например, в примере ниже не произошло обратного свертывания $(x+1)^2$, хотя, будучи применен к первоначальному выражению, флаг `ratfac` сохранил и этот множитель нераскрытым (также можете сравнить этот

пример с аналогичным примером к функциям `ratexpand()` и `rat()`:

$$(\%i1) \text{ rat}\left(\frac{(x+1)^2 (\%e^x - 1)}{(x+2)(2x+1)}\right)$$

$$(\%o1) \frac{(x^2 + 2x + 1)e^x - x^2 - 2x - 1}{2x^2 + 5x + 2}$$

(%i2) `%,ratfac: true`

$$(\%o2) \frac{(e^x - 1)(x^2 + 2x + 1)}{2x^2 + 5x + 2}$$

$$(\%i3) \text{ rat}\left(\frac{(x+1)^2 (\%e^x - 1)}{(x+2)(2x+1)}\right), \text{ratfac: true}$$

$$(\%o3) \frac{(e^x - 1)(x+1)^2}{(2x+1)(x+2)}$$

Проще простого

Итак, о преобразованиях выражений мы уже говорили достаточно – теперь перейдем к их упрощению. Об элементарных упрощениях мы уже говорили в предыдущий раз: они могут производиться автоматически, на что влияет установленный флаг `simp`; и по умолчанию именно так и происходит.

Здесь тоже все начинается с рациональных выражений, которыми занимается функция `ratsimp(выражение)`. Она упрощает выражение за счет рациональных преобразований, но, в отличие от остальных функций по обработке рациональных выражений, работает в том числе и «вглубь», то есть иррациональные части выражения не рассматриваются как атомарные, а упрощаются, в том числе, и все рациональные элементы внутри них:

$$(\%i1) \text{ ratsimp}\left(\%e^{\frac{x^3+1}{x+1}}\right)$$

$$(\%o1) e^{x^2-x+1}$$

На `ratsimp()` действуют те же флаги, что и на `rat()`: и `ratexpand()`, и `keepfloat`, и `ratfac`. Но отличается она от `rat()` или `ratexpand()` не только умением работать «в глубину», но и некоторыми дополнительными рациональными преобразованиями, которые не поддерживаются этими двумя функциями:

$$(\%i1) \frac{\sqrt{(x-a)^3} - (x+a)\sqrt{x-a}}{\sqrt{(x-a)(x+a)}} \$$$

(%i2) `rat(%)`

$$(\%o2) \frac{\sqrt{x-a^3} + (-x-a)\sqrt{x-a}}{\sqrt{(x-a)(x+a)}}$$

(%i3) `ratexpand(%o1)`

$$(\%o3) \frac{(x-a)^{\frac{3}{2}} - x\sqrt{x-a} - a\sqrt{x-a}}{\sqrt{(x-a)(x+a)}}$$

(%i4) `ratsimp(%o1)`

$$(\%o4) -\frac{2a\sqrt{x-a}}{\sqrt{x^2-a^2}}$$

Кроме функции `ratsimp()`, есть еще и дополнительный переключатель – `ratsimpexpons`. По умолчанию он установлен в `false`; если же назначить ему значение `true` – это приведет к автоматическому упрощению показателей степени:

$$(\%o1) x^{\frac{a^2+a+\frac{1}{4}}{2a+1}}$$

(%i2) `%,ratsimpexpons: true`

$$(\%o2) x^{\frac{2a+1}{4}}$$

Функция `ratsimp()` – это уже достаточно мощный, и в то же время весьма быстрый, механизм упрощения; но, конечно, не достаточный: ведь те действия, которые можно упростить в разнообразных математических выражениях, не ограничиваются рациональными. Поэтому все же основной плюс этой функции – это скорость. А для более серьезных упрощений существует расширенный вариант – `fullratsimp(выражение)`. Эта функция последовательно применяет к переданному выражению функцию `ratsimp()`, а также некоторые не-рациональные преобразования – и повторяет эти действия в цикле до тех пор, пока выражение не перестанет в процессе них изменяться. За счет этого функция работает несколько медленнее, чем `ratsimp()`, зато дает более надежный результат – к некоторым выражениям, которые она может упростить с ходу, `ratsimp()` пришлось бы применять несколько раз, а иногда та и вообще не справилась бы с задачей.

$$(\%i1) \frac{(x^{\frac{a}{2}} - 1)^2 (x^{\frac{a}{2}} + 1)^2}{x^a - 1} \$$$

(%i4) `ratsimp(%)`

$$(\%o4) \frac{x^{2a} - 2x^a + 1}{x^a - 1}$$

(%i5) `ratsimp(%)`

$$(\%o5) x^a - 1$$

(%i6) `fullratsimp(%o1)`

$$(\%o6) x^a - 1$$

И третья основная функция упрощения выражений – уже никак с предыдущими двумя не соотносящаяся – `radcan(выражение)`. Если `ratsimp()` и `fullratsimp()` ориентированы на упрощение рациональных действий, то `radcan()` занимается упрощением логарифмических, экспоненциальных функций и степенных с нецелыми рациональными показателями, то есть корней (радикалов). Например, выражение из второго примера в этом разделе `radcan()` сможет упростить сильнее, чем `ratsimp()/fullratsimp()`:

$$(\%i1) \text{ ratsimp}\left(\frac{\sqrt{(x-a)^3} - (x+a)\sqrt{x-a}}{\sqrt{(x-a)(x+a)}}\right)$$

$$(\%o1) -\frac{2a\sqrt{x-a}}{\sqrt{x^2-a^2}}$$

(%i2) `fullratsimp(%)`

$$(\%o2) -\frac{2a\sqrt{x-a}}{\sqrt{x^2-a^2}}$$

(%i3) `radcan(%)`

$$(\%o3) -\frac{2a}{\sqrt{x+a}}$$

>>

(%i3) `trigreduce(%)`

(%o3)
$$\frac{\cos(2y+x)}{2} - \frac{\cos(2y-x)}{2} + \frac{\sin(y+2x)}{2} - \frac{\sin(y-2x)}{2} + \cos x \cos(2y) + \cos(2x) \sin y$$

(%i4) `trigreduce(%)`

(%o4)
$$\cos(2y+x) + \sin(y+2x)$$



В некоторых случаях наилучшего результата можно добиться, комбинируя `radcan()` с `ratsimp()` или `fullratsimp()`.

С функцией `radcan()` смежны по действию еще два управляющих ключа. Один из них называется `%e_to_numlog`. Влияет он не на саму функцию, а на автоматическое упрощение. Если выставить его в `true`, то выражения вида $e^{(r \cdot \log(\text{выражение}))}$, где r – рациональное число, будут автоматически раскрываться в `выражение'`. Функция `radcan()` делает такие преобразования независимо от значения ключа. Второй ключ – `radexpand` (от `radical`, не путать с `ratexpand`) – влияет на упрощение квадратного корня из четной степени какого-либо выражения. Он, в отличие от большинства переключателей, имеет не два, а три значения: при значении `all`, `sqrt(x^2)` будет раскрываться в x – как для действительных, так и для комплексных чисел; при значении `true` (по умолчанию), `sqrt(x^2)` для действительных чисел превращается в $|x|$, а для комплексных не преобразуется; а при значении `false`, `sqrt(x^2)` не будет упрощаться вообще.

Следующие две функции и один флаг относятся к упрощению факториалов. Функция `factcomb(выражение)` проводит упрощения вида $n!(n+1) = (n+1)!$ и тому подобные. Функция `minfactorial`, напротив, сокращает факториалы, то есть действует по принципу $n!(n-1)! = n$. И флаг `sumsplitfact`, который изначально установлен в `true`, находясь в состоянии `false`, приводит к тому, что после того, как отработает `factcomb`, `minfactorial` вызывается автоматически.

Вот под таким углом...

И напоследок поговорим о функциях для преобразования тригонометрических формул. Здесь так же, как и у рациональных функций, присутствует общая для всех приставка – `trig`; расшифровывать ее, думаю, особой нужды нет. Начнем по традиции с функции `trigexpand(выражение)`. Она, как нетрудно догадаться, раскрывает скобки в тригонометрических выражениях:

(%i1) `trigexpand(sin(2x+y)+cos(x+2y))`

(%o1)
$$-\sin x \sin(2y) + \cos x \cos(2y) + \cos(2x) \sin y + \sin(2x) \cos y$$

Здесь, как обычно, есть несколько управляющих флагов, первый из которых опять же является тезкой самой функции. Он приводит к повторному раскрытию всех синусов-косинусов, то есть фактически равнозначен повторному вызову самой функции:

(%i2) `%, trigexpand: true`

(%o2)
$$\cos x \left((\cos y)^2 - (\sin y)^2 \right) - 2 \sin x \cos y \sin y + \left((\cos x)^2 - (\sin x)^2 \right) \sin y + 2 \cos x \sin x \cos y$$

Второй флаг – `halfangles` – управляет раскрытием формул половинных углов. Оба эти флага по умолчанию сброшены. А следующие два флага – `trigexpandplus` и `trigexpandtimes` – отвечают соответственно за применение формул сумм углов и кратных углов. То есть в примере выше сначала сработал флаг `trigexpandplus`, а затем – `trigexpandtimes`. Эти флаги по умолчанию установлены, что и видно из примера.

Кроме всего уже упомянутого, есть еще флаги `trigsign` и `triginverses`. Первый принимает традиционные два значения (по умолчанию – `true`) и регулирует вынос знака за пределы тригонометрической функции, то есть, к примеру, `sin(-x)` упростится до `-sin(x)`, а `cos(-x)` – до `cos(x)`. Флаг `triginverses` – трехзначный, и умолчательное его значение

равно `all`. Он отвечает за обработку сочетаний вида `sin(asin(x))` или `atan(tan(x))`. Значение `all` позволяет раскрывать эти сочетания в обоих направлениях (напомню, что при этом часть корней будет теряться); значение `true` оставляет разрешенным раскрытие только вида `sin(asin(x))`, то есть блокирует вариант с потерями периодических значений; а случай `false` запрещает оба направления преобразований.

Функция, обратная `trigexpand()`, называется `trigreduce(выражение)` – здесь, в полном соответствии со значением слова `reduce`, действуют формулы понижения степени. Например, применив дважды эту функцию к результату предыдущего примера, мы получим его в исходном виде (см. рис. *наверху страницы*).

Эту функцию можно вызвать с более полным списком аргументов: `trigreduce(выражение, переменная)` – тогда формулы понижения степени будут применяться только по отношению к заданной переменной (переменная может быть, как и почти везде, не только отдельным символом, но и выражением).

Третья функция занимается уже упрощением, и зовут ее, соответственно, `trigsimp(выражение)`. Она старается упростить любое тригонометрическое выражение, используя известные формулы, такие как `sin^2(x)+cos^2(x)=1` и тому подобные. Для наилучшего результата ее можно комбинировать с `trigreduce()`, `ratsimp()/fullratsimp()` и `radcan()`.

Этим возможности *Maxima* по преобразованию и упрощению разнообразных выражений еще не совсем исчерпаны, но основные из них мы рассмотрели в полной мере. В следующий раз поговорим немного о применении некоторых встроенных функций, о работе с векторами, матрицами и множествами и, возможно, о работе с логикой, с уравнениями и неравенствами, а также их системами. **LF**



АНАЛИЗ ДАННЫХ

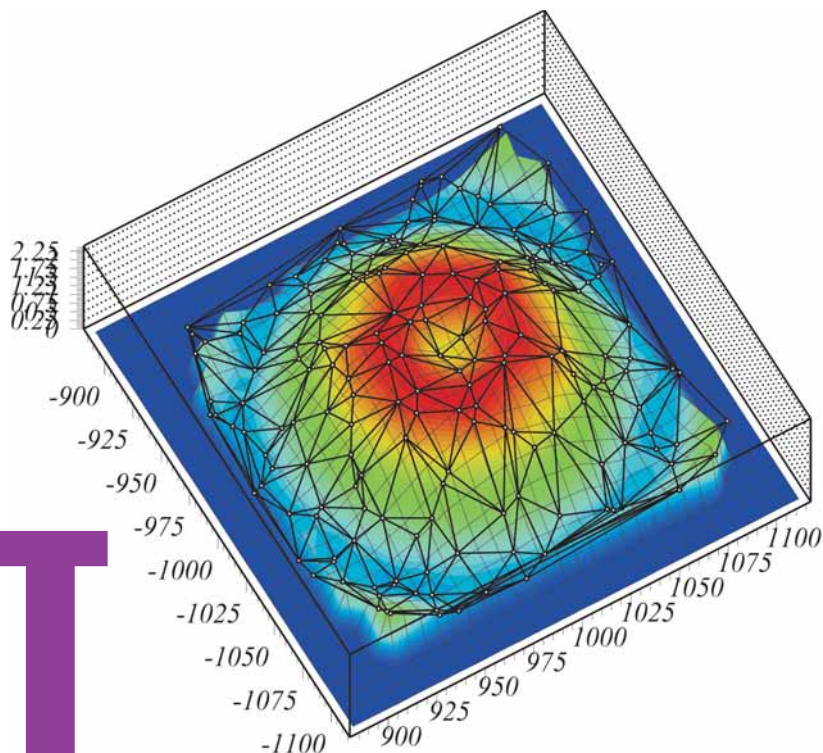
Продолжая традиции:

ROOT



НА ДИСКЕ

- Исходные тексты PAW
- Исходные тексты ROOT
- Документация
- Файлы данных, использовавшиеся в качестве примеров в статьях



ЧАСТЬ 3 Данные мало получить – надо ещё понять, а есть ли от них польза. Евгений Балдин представляет вашему вниманию «новинку» среди приложений для анализа данных.



Даже если данных много – их надо как-то проанализировать. Это может сделать только человек. Компьютер в этом деле только помощник. Выбор инструмента очень важен. *ROOT* – хороший инструмент. У него был достойный предок и он мог бы быть гораздо лучше. Но здесь и сейчас надо анализировать данные, фиксируя недостатки, дабы исправить их в будущем. Это возможно, потому что *ROOT* – это свободный продукт.

Примерно через десять лет после возникновения команде *PAW* (Physics Analysis Workstation) стало скучно, и ее лидер *PAW* Рене Брюн (René Brun) сотоварищи начал новый проект *ROOT* – An Object Oriented Data Analysis Framework¹.

Компьютеры стали много мощнее, но и поток данных увеличился. *ROOT* стал разрабатываться в рамках эксперимента NA49, где поток данных за один заход мог превышать 10 Тб².

С начала 2006 года *ROOT* (<http://root.cern.ch/>) стал выпускаться под лицензией GNU, и, возможно, скоро попадёт во все основные дистрибутивы GNU/Linux.

Сравнение с PAW

PAW является предком *ROOT*, если уж не в смысле кода, то уж в смысле реализации идей точно. Поэтому полезно понять, чем эти пакеты отличаются и в чём совпадают. Сравнительная таблица не претендует на фундаментальность, а просто отражает личные пристрастия автора.

Почему *PAW*? Если в вашем проекте *PAW* уже используется, особых причин для смены инструмента нет. Для стандартных операций анализа *ROOT* использовать значительно сложнее, чем *PAW*. Это плата за попытку объять необъятное.

¹ Почему *ROOT* так называется? У меня есть только догадки: OO — видимо, Object Oriented, а сам *ROOT* от английского «корень» или «источник» (root). Предполагается, что *ROOT* — это база для разработок, а не просто система анализа, то есть корень всех ...

² Таких заходов было чуть меньше 5 тысяч. Грубо, 1 Тб=103 Гб=106 Мб.

³ Я слышал такой вариант ответа: «потому что, в отличие от *PAW*, в графическом окне *ROOT* можно мышкой подправить экспериментальные данные» — очень надеюсь, что это была шутка.

Почему *ROOT*?³ C++ популярнее FORTRAN и KUIP. C++ привычнее и с его помощью проще решать задачи, которые являются вспомогательными к анализу — для всего используется один инструмент. *ROOT* активно поддерживается и развивается. У *ROOT* есть довольно мощное сообщество. На сайте <http://root.cern.ch> можно найти ответ почти на все вопросы, касающиеся пакета, в RootTalk (там же) можно задать вопрос любой сложности, на который вам с очень большой вероятностью ответят.

Запускаем ROOT

Так как *ROOT* получил лицензию LGPL совсем недавно, то, скорее всего, в вашем настольном дистрибутиве его нет. Поэтому запуск придётся отложить «на потом» после сборки и установки.

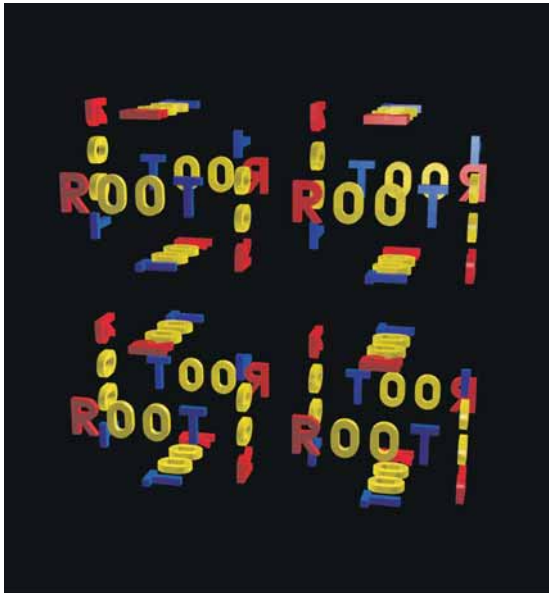
МЕСЯЦ НАЗАД

Мы рассмотрели работу с *PAW* на практическом примере



СРАВНЕНИЕ PAW И ROOT (ИМНО)

Признак	PAW	ROOT
Авторы	Рене Брюн и др.	Рене Брюн и др.
Возраст	20 лет	Чуть больше 10 лет
GNU	Начиная с 2000 года	С начала 2006 года
Интерпретатор	FORTRAN (COMIS)	C++ (CINT)
Командный процессор	KUIP	C++ (CINT)
Ускорение набора команд	Сокращение	TAB-completion
Кириллица?	Никак	Аналогично
Состояние	Матёрая, но немного устаревшая система	Надо повзрослеть, пора бы уже
Система помощи	Подробная официальная документация в почти 500 страниц, help в командной строке	Пухлое руководство пользователя, автодокументация по исходным текстам, но нет help.
Что бы сказал Брукс?	Старая школа	Все признаки «второй системы»
Что есть?	То, что нужно для анализа данных	То же, плюс много чего ещё лишнего и не очень



ROOT в действии – демонстрация.



Брать исходные тексты лучше всего с основного сайта: <http://root.cern.ch>. После распаковки дерева пакетов следует внимательно изучить инструкцию **README/INSTALL**. Сборка стандартная:

```
> ./configure --prefix=/usr/local ; make ; make install
```

make install необходимо делать от имени **root**.

Можно попробовать собрать *rpm*- или *deb*-пакет. Собрать *deb*-пакет под Debian 3.1 (Sarge) без дополнительных телодвижений не удаётся, так как отсутствует пакет, на который указывают зависимости. По-видимому, разработка ведётся для тестовой или нестабильной ветки дистрибутива.

После установки перед запуском необходимо установить переменные окружения. Для *bash* это будет выглядеть примерно так:

```
> export ROOTSYS=/usr/local/
> export PATH=$PATH:$ROOTSYS/bin
> export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ROOTSYS/lib/root
```

Установка **LD_LIBRARY_PATH** необходима из-за того, что почти весь функционал *ROOT* вынесен в разделяемые библиотеки, которые подгружаются во время работы программы. Аналогично можно оформить и свою библиотеку, расширив, таким образом, возможности *ROOT*.

Всё. Теперь открываем терминал и запускаем *ROOT*:

```
> root
*****
*                               *
*      WELCOME to ROOT          *
*                               *
*      Version   5.11/02   19 April 2006   *
*                               *
*      You are welcome to visit our Web site   *
*      http://root.cern.ch   *
*                               *
*****

FreeType Engine v2.1.9 used to render TrueType fonts.
Compiled on 19 May 2006 for linux with thread support.

CINT/ROOT C/C++ Interpreter version 5.16.11, April 14, 2006
Type ? for help. Commands must be C++ statements.
Enclose multiple statements between { }.
root [0]
```

Получив приглашение, можно приступать к работе. Сказать **“Hello World”** из *ROOT* можно следующим образом:

⁴ В руководстве пользователя в этом месте присутствуют ошибки.
⁵ Чем больше *q*, тем «сильнее» желание.

```
root [0] cout << "Hello World" << endl;
Hello World
```

При запуске *ROOT* считывается файл настроек **.rootrc** сначала в текущей директории, а, если здесь его нет, то в домашней; затем берётся системный файл **/etc/root/system.rootrc**. От версии к версии эта последовательность может меняться⁴.

По умолчанию есть ещё три файла, которые могут управлять поведением программы:

- **rootlogon.C** – выполняется при запуске,
- **rootalias.C** – загружается при запуске, но не выполняется,
- **rootlogoff.C** – выполняется при завершении сеанса.

ROOT можно запускать и не в интерактивном режиме. Для этого при запуске следует указать опцию **-b**. Полный список поддерживаемых опций можно получить при указании ключа **-h**.

Выйти из *ROOT* можно с помощью команды **.q**. Если в процессе анализа удалось зациклить программу, то желание выйти можно усилить с помощью команд **.qqq**, **.qqqqq** или **.qqqqqqq**⁵. **^C** так же может помочь в непредвиденных ситуациях.

«Командная логика»

В качестве командного процессора используется интерпретатор C++ CINT. Это означает, что интерактивная работа очень похожа написание обычной программы. Знание языка C/C++ при «общении» с *ROOT* является обязательным. Как и для *PAW*, напишем программу по вычислению чисел Фибоначчи:

```
root [0] {
end with '}', '@:abort > int a=0,b=1;
end with '}', '@:abort > cout << a << " " << b << " ";
end with '}', '@:abort > for (int i=2;i<=10;i++) {
end with '}', '@:abort > int x=a; a=b; b=x+b;
end with '}', '@:abort > cout << b << " ";
end with '}', '@:abort > }
end with '}', '@:abort > cout << endl;
end with '}', '@:abort > }
0 1 1 2 3 5 8 13 21 34 55
```

Команды группируются с помощью фигурных скобок. Этот же код можно сохранить в файл **fibonacci.cxx** и выполнить его как скрипт:

```
root [1] .x fibonacci.cxx
0 1 1 2 3 5 8 13 21 34 55
```

В случае C++ окончание команды отмечается «;». Если «;» опустить, то из *ROOT* получится неплохой калькулятор:

```
root [2] 2*sqrt(5)*sin(2*3.14*75/180)/3.14**2
(const double)2.27312089125660893e-01
root [3] 2**10
(const int)1024
root [4] 2.**1023
(const double)8.98846567431157954e+307
```

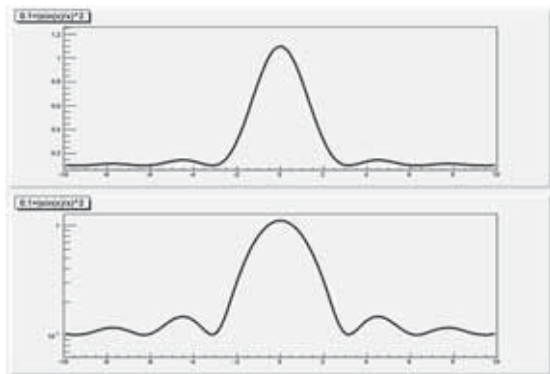
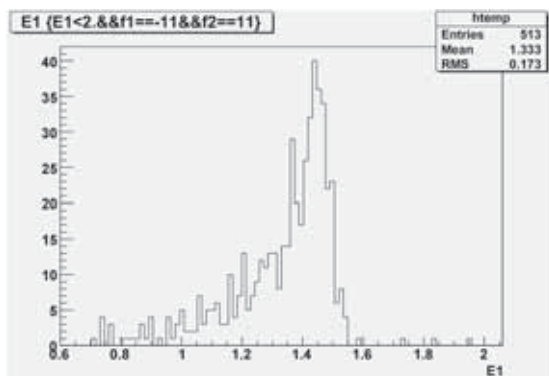
Все вспомогательные команды *ROOT* начинаются с точки (**.**). Для выполнения команд оболочки используется команда **!.!**, за которой следуют shell-инструкции:

```
root [5] !.ls *.cxx
fibonacci.cxx
```

Полный список вспомогательных команд можно получить с помощью инструкции **?.**

Все необходимые для анализа объекты представлены в виде классов. Класс **TFile** соответствует файлу, в который можно сохранять *ROOT*-структуры. Объект **TTree** представляет из себя более изощрённую реализацию идеи *ntuple*:

```
root [6] TFile *f=new TFile("ee-eng.root")
root [7] TTree *tree;
root [8] tree=(TTree *) f->Get("h1");
root [9] tree->Draw("«TAB»")
void Draw(Option_t* opt)
Long64_t Draw(const char* varexp, const TCut& selection, Option_t*
option = "", Long64_t nentries = 1000000000, Long64_t firstentry = 0)
Long64_t Draw(const char* varexp, const char* selection, Option_t*
```



Примеры графического представления гистограммы (канва E1) и функции (канва cfunc).»

```
option = "", Long64_t nentries = 1000000000, Long64_t firstentry = 0)
root [10] tree->Draw("E1", "E1<2.&&f1==11&&f2==11")
```

В строке [9] после скобки была нажата клавиша **Tab**, что привело к выводу подсказки по возможным командам. Отсутствие команды **help** восполняется автоматически создаваемой подсказкой.

Графический интерфейс

Графическое окно в *ROOT* называется «канвой» (объект *TCanvas*). Можно открыть сколько угодно таких окон:

```
//Создаём новую канву E1.
root [11] TCanvas *E1=new TCanvas("E1")
//Создаём новую канву cfunc.
root [12] TCanvas *cfunc=new TCanvas("func")
```

```
//Переходим в канву E1.
```

```
root [13] E1->cd();
```

```
//Рисуем гистограмму по параметру E1 с условием.
```

```
root [14] tree->Draw("E1", "E1<2.&&f1==11&&f2==11")
```

```
//Переходим в канву cfunc.
```

```
root [15] cfunc->cd();
```

```
//Делим канву cfunc на две части по Y.
```

```
root [16] cfunc->Divide(1,2)
```

```
//Создаём функцию.
```

```
root [17] TF1 f1("d1f", "0.1+(sin(x)/x)**2", -10,10)
```

```
//Переходим в верхнюю половину канвы cfunc.
```

```
root [18] cfunc->cd(1)
```

```
//Отображаем функцию.
```

```
root [19] f1->Draw()
```

```
//Переходим в нижнюю половину канвы cfunc.
```

```
root [20] cfunc->cd(2)
```

```
root [21] f1->Draw()
```

```
//Устанавливаем для нижней половины канвы cfunc
```

```
//логарифмический масштаб для оси Y.
```

```
root [22] cfunc->cd(2)->SetLogy()
```

```
//Из канвы cfunc создаём векторный eps-файл.
```

```
root [23] cfunc->Print("root-cfunc.eps")
```

```
//Из канвы E1 создаём растровый png-файл.
```

```
root [24] E1->Print("root-E1.png")
```

В отличие от своего предка *PAW*, *ROOT* позволяет интерактивно менять параметры картинки с помощью выпадающих меню. Тип меню зависит от того, на какой объект направлен указатель мыши. Также с помощью левой кнопки можно интерактивно изменять масштаб графика. Для возврата в исходное состояние в меню, относящемся к выбранной оси, следует выбрать команду **UnZoom**.

Не стоит этим увлекаться, так как кажущаяся простота увеличивает время, потраченное на создание картинок. В отличие от набранных команд, осмысленные движения и клики мыши сохранить для повторного использования невозможно.

Базовые объекты

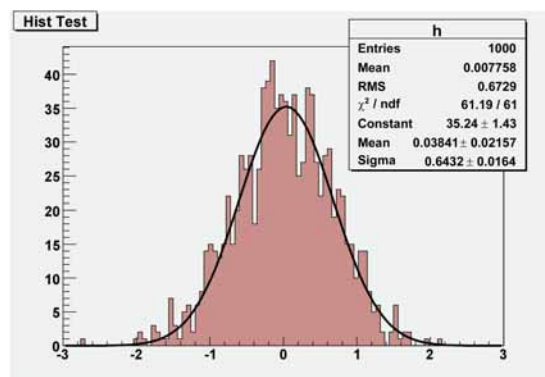
ROOT унаследовал все базовые объекты анализа, которые были в *PAW*. Но, в отличие от *PAW*, *ROOT* не ограничивается исключительно анализом. Примером такого подхода, например, служит включение в пакет операций для работы с матрицами (линейная алгебра) и базовых средств для манипуляции объектами OpenGL (отображение физических объёмов). *ROOT* претендует на нечто большее, чем быть просто пакетом анализа, но всё же в этом разделе будут перечислены только те объекты, которые могут пригодиться для представления данных.

Гистограммы

Гистограмма является одним из основных объектов анализа. По сравнению с *PAW*, в *ROOT* было добавлено больше типов гистограмм. Конструктор гистограмм имеет вид **TH1F**. Для двумерной гистограммы вместо **1** надо подставить **2**, а для трёхмерной (да, такие тоже есть, правда, непонятно, как их смотреть) – **3**. **F** означает, что на один бин используется *Float_t*, аналогично возможны и другие типы переменных для хранения значения в бине.

```
//Создаём новую канву.
root [25] TCanvas *ch=new TCanvas("Hist Test", "Hist")
//Создаём гистограмму в 100 бинов от -3. до 3.
root [26] TH1F *h = new TH1F("h", "Hist Test", 100, -3., 3.)
//Обычно гистограммы заполняются с помощью метода Fill.
root [27] h->Fill(«TAB»)
Int_t Fill(Double_t x)
Int_t Fill(Double_t x, Double_t w)
Int_t Fill(const char* name, Double_t w)
//Но мы сейчас идём другим путём:
// а) создаём функцию G,
root [28] TF1 *func = new TF1("G", "exp(-x**2)", -3,3)
// б) заполняем гистограмму случайным образом
// по форме функции G.
root [29] h->FillRandom("G", 1000)
//Меняем цвет гистограммы.
root [30] h->SetFillColor(45)
//Подгоняем гистограмму распределением Гаусса
root [31] h->Fit("gaus")
...
//Сохраняем полученную картинку.
root [32] ch->Print("root-histexample.eps")
```

Подгонкой «заведует» всё тот же *Minuit*, что был и в *PAW*, правда,



Пример подогнанной гистограммы.

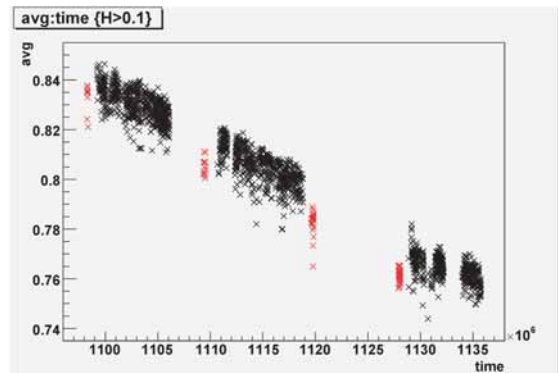


переписанный на С++. Алгоритмы не менялись.

Деревья

Деревья (tree) в ROOT – это логичное развитие идеи *ntuple*. *ntuple*, по сути дела, был таблицей со столбцами переменных типа *float*. В случае деревьев этого ограничения не существует, и в дереве можно сохранять любые объекты.

//Создаём файл на диске.

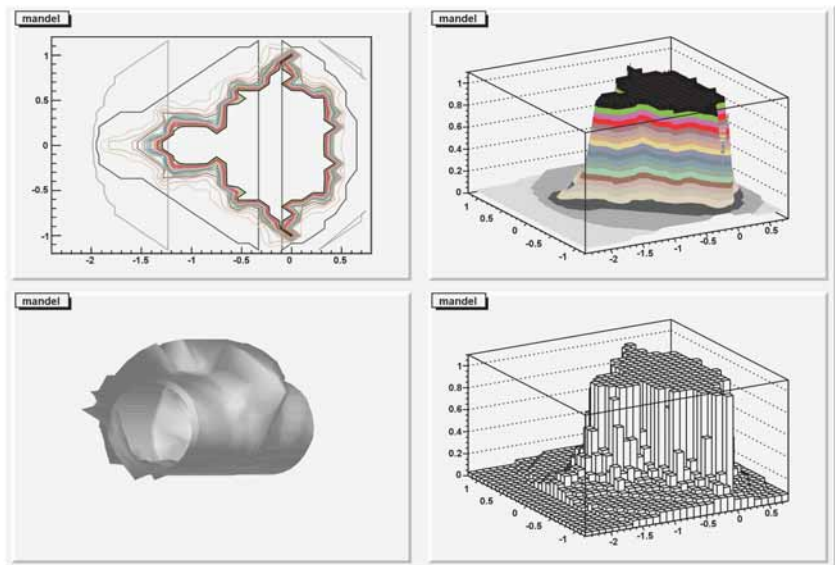


```

root [33] TFile *f = new TFile("lkravg.root","RECREATE")
//Заводим новое дерево
root [34] TTree *lkravg = new TTree("lkravg","LKr degrad")
//Считаем файл lkravg.dat - тот самый, что "мучили" в
//статье про PAW
root [35] Long64_t nlines = lkravg->ReadFile("lkravg.dat",
//список переменных
"time:run:avg:avg_er:P:H")
root [36] cout << "Number of lines: " << nlines << endl
//Рисуем картинку: чёрные маркеры - есть магнитное поле,
//красные маркеры - нет магнитного поля.
root [37] lkravg->SetMarkerStyle(5)
root [38] lkravg->Draw("avg:time","H>0.1")
root [39] lkravg->SetMarkerColor(kRed)
root [40] lkravg->Draw("avg:time","H<=0.1","same")
//Пишем дерево в файл и закрываем файл.
root [41] lkravg->Write();
root [42] f->Close();
//Теперь этот файл можно открыть
root [43] TFile *f2 = new TFile("lkravg2.root")
//и посмотреть что в нём есть - дерево сохранилось.
root [44] .ls
TFile** lkravg2.root
TFile* lkravg2.root
    
```

Графическое представление дерева *lkravg*. Видно, что красные точки в среднем ниже чёрных, что и требовалось доказать.

Примеры графического представления двумерной функции или гистограммы. Множество Мандельброта.



KEY: TTree lkravg;1 LKr degrad

В ROOT есть множество способов создать и заполнить дерево. Подробности лучше посмотреть в пользовательской документации.

Функции

Как и в PAW, в ROOT есть мощная поддержка функций как объектов. С помощью метода *Fit* можно подогнать гистограмму или график. Но до этого следует определить функцию, например, так:

```

//Файл mandel.cxx
//Множество Мандельброта
Double_t mandel(Double_t *XP,Double_t *par) {
const Int_t nmax=30;
Double_t xx=0.,yy=0.,tt,x,y;
x=XP[0];y=XP[1];
for (Int_t n=1;n<nmax;n++) {
tt=xx*xx-yy*yy+xx;
yy=2.*xx*yy+y;
xx=tt;
if (xx*xx+yy*yy>4.) break;
}
return Double_t(n)/Double_t(nmax);
}
    
```

Текст функции следует сохранить в файле **mandel.cxx**. После с ним можно работать из ROOT:

```

//Загружаем описание функции mandel.cxx.
//Теперь можно обращаться к функции.
root [45] .L mandel.cxx
root [46] TCanvas *cm=new TCanvas("mandelbrot","Mandelbrot")
//Создаём объект «двумерная функция» TF2
root [47] TF2 *Mandelbrot=new
TF2("Mandelbrot",mandel,-2.4,-8,-1.2,1.2,0)
root [48] cm->Divide(2,2)
root [49] cm->cd(1)
root [50] Mandelbrot->SetNpx(«TAB»
void SetNpx(Int_t npx = 100) // *MENU*
//Увеличиваем число шагов отображения.
//Как и в PAW функции отображаются через гистограммы.
root [51] Mandelbrot->SetNpx(200)
root [52] Mandelbrot->SetNpy(200)
//Контурное графическое представление.
root [53] Mandelbrot->Draw("cont")
root [54] cm->cd(2)
//Графическое представление в виде поверхность.
root [55] Mandelbrot->Draw("surf2")
root [56] cm->cd(3)
//Множество Мандельброта в цилиндрических координатах.
root [57] Mandelbrot->Draw("surf4cyl")
root [58] cm->cd(4)
//Графическое представление в стиле LEGO.
root [59] Mandelbrot->Draw("lego")
root [60] cm->Print("root-mandel.eps")
    
```

Интерпретатор С++ (CINT)

Интерпретатор С++ или CINT, который используется в ROOT, был независимым проектом. Сейчас он является составной частью ROOT, но его можно использовать и отдельно. Домашняя страничка CINT доступна по адресу <http://root.cern.ch/root/Cint.html>.

CINT охватывает примерно 95% конструкций ANSI C и 85% от С++. Следует понимать, что полное соответствие стандартам никогда не было основной целью CINT. Не следует писать больших программ, опираясь на интерпретатор, так как скорость выполнения команд уступает компилируемой версии программы примерно в десять раз. А где один порядок, там и два. Для небольших скриптов автоматизации анализа CINT вполне подходит, но для серьезных целей надо писать обычные программы. Благо, абсолютно всё, что доступно в ROOT интерактивно, доступно и через библиотечные вызовы. Так уж ROOT сделан.

Для внешних CINT-скриптов есть две полезные команды:

```
//Выполняем скрипт script.cxx
root [66] .x script.cxx
//Загружаем функции, описанные в lib.cxx
root [67] .L lib.cxx
```

Одной из отличительных особенностей *ROOT* является возможность делать функции из внешних библиотек доступными для выполнения в скриптах CINT или интерактивно. Ниже будет приведён пример, как подключить пользовательскую C-библиотеку.

Допустим, у вас есть C-библиотека, в которой есть функции `myfunc1()` и `myfunc2(char*)`, которые необходимо экспортировать в среду *ROOT*. Для этого нужно создать заголовочный файл `myfile.h` примерно следующего вида:

```
/*Файл myfile.h*/
#ifdef __cplusplus
extern "C" {
#endif
extern void myfunc1();
extern int myfunc2(char *);
#ifdef __cplusplus
}
#endif
```

Пока всё как обычно. Чтобы экспортировать функции в *ROOT*, необходимо создать ещё один заголовочный файл `myfileLinkDef.h` (к `myfile` добавляется `LinkDef`):

```
/*Файл myfileLinkDef.h */
#ifdef CINT_
#pragma link C++ function myfunc1();
#pragma link C++ function myfunc2(char*);
#endif
```

Так же можно экспортировать и структуры, подставив вместо слова `function` слово `struct`. После создания описанных заголовочных файлов необходимо создать «словарик»:

```
> rootcint -f myfileDict.cxx -c myfile.h myfileLinkDef.h
```

В результате будут созданы файлы `myfileDict.h` и `myfileDict.cxx`.

схх.

Далее нужно собрать саму библиотеку. Пусть для простоты вся библиотека представляет из себя один C-файл `myfile.c`:

```
# Компилируем myfile.c.
> gcc -c -fPIC myfile.c
# Компилируем словарь.
> g++ -c -fPIC `root-config --cflags` myfileDict.cxx
# Создаём разделяемую библиотеку.
> g++ -shared -o myfile.so myfile.o myfileDict.o
```

Теперь эту вновь созданную библиотеку можно загрузить в *ROOT* для интерактивной работы:

```
root [68] gSystem->Load("myfile")
root [69] myfunc1()
root [70] Int_t icount=myfunc2("string")
```

Это далеко не единственный способ подключить пользовательскую библиотеку к *ROOT*. Для компиляции скриптов можно использовать подсистему ACLiC.

P.S. Кроме CINT, в среде *ROOT* можно использовать скрипты, написанные на Python или Ruby. И наоборот: из этих языков можно общаться с библиотеками *ROOT*. К сожалению, описание этих механизмов выходит за рамки данной статьи.

Заключение

Эта статья – не описание *ROOT*, а всего лишь набор штрихов к его портрету. Для более подробного знакомства настоятельно рекомендуем посетить <http://root.cern.ch>.

ROOT – не просто инструмент анализа; это среда для генерации таких инструментов. Он, возможно, неповоротлив и избыточен, но гибок и очень легко расширяем. Это не идеал, но идеал, скорее всего, будет на него похож. **LXF**

**ЧЕРЕЗ
МЕСЯЦ**

Хотите узнать про анализ данных что-нибудь ещё?

Присылайте свои пожелания на адрес letters@linuxformat.ru

Делать больше с меньшими затратами.

Если Вам нужны мощные серверы, которые позволят делать больше работы и консолидировать вычислительные ресурсы, выбирайте серверы ASNORD® на базе двухъядерных процессоров Intel® Xeon®.



Компания INVICTUS специализируется в области решения задач по созданию систем обработки, хранения и защиты данных на базе кластеров, сетей хранения данных SAN, внешних RAID-массивов, NAS-серверов, ленточных накопителей. Подробнее о наших предложениях Вы можете узнать на нашем сайте по адресу www.invictus.ru. Ждем Ваши запросы по адресу invictus@invictus.ru и по телефонам...

(812) 449-8888 (495) 647-0080

Серверы ASNORD® - это работа 24 часа в сутки, высокие уровни реагирования и управляемости, 3 года стандартной гарантии, различные виды расширенной поддержки.

INVICTUS®
OPEN STORAGE TECHNOLOGIES

Обозначения Celeron, Celeron Inside, Centrino, Centrino logo, Core Inside, Intel, Intel Core, Intel logo, Intel Inside, Intel Inside logo, Intel SpeedStep, Intel Viiiv, Intel Xeon, Itanium, Itanium Inside, Pentium и Pentium Inside являются товарными знаками, либо зарегистрированными товарными знаками, права на которые принадлежат корпорации Intel или ее подразделениям на территории США и других стран.



socket programming linux

Поиск

[Расширенный поиск](#)
[Настройки](#)

Поиск в Интернете Поиск страниц на русском

Веб Результаты **1 - 10** из примерно **4 060 000** для **socket programming linux**. (0,25 секунд)

Совет: Ищите страницы только на [русском языке](#). Вы можете задать язык поиска в разделе [Настройки](#)

Unix - Linux Network Socket programming

PROГРАММИРОВАНИЕ ДЛЯ UNIX
This document is written as a tutorial, not a reference. It is probably at its best when read by individuals who are just starting out with **socket ...**

Сокеты

www.troubleshooting.com/cpp/articles/140101networkprogramming.shtml - 46k -
Сохранено в кэше - Похожие страницы

ЧАСТЬ 3 Сегодня ваши программы научатся общаться со внешним миром – следуйте за **Андреем Боровским!**

Socket Programming

A **socket programming** tutorial. ... This tutorial requires a **Linux** box. It hasn't been tested on other types of UNIX, but I think it might work. *«Сто лет тому назад дед нынешнего императора, в те времена еще наследный принц, за завтраком разбил яйцо с тупого конца и скорлупой порезал себе палец... С того времени все население Лилипутии разделилось на два лагеря – тупоконечников и остроконечников»*

Дж. Свифт, «Гулливер в стране лилипутов»

МЕСЯЦ НАЗАД

Мы говорили об очередях сообщений и семафорах



В конкурсе на лучшую компьютерную идею всех времен и народов сокеты, без сомнения, могли бы рассчитывать на призовое место. Как и другие средства межпроцессного взаимодействия, рассмотренные в этой серии статей, сокеты впервые были реализованы именно на платформе Unix (4.2BSD), однако концепция сокетов как универсального средства обмена данными между процессами оказалась настолько удачна, что все современные системы поддерживают, по крайней мере, некоторое подмножество сокетов. Причины успеха сокетов заключаются в их простоте и универсальности. Программы, обменивающиеся данными с помощью сокетов, могут работать в одной системе и в разных, используя для обмена данными как специальные объекты системы, так и сетевой стек. Как и каналы, сокеты используют простой интерфейс, основанный на «файловых» функциях read(2) и write(2) (открывая сокет, программа Unix получает дескриптор файла, благодаря которому можно работать с сокетами, используя файловые функции), но, в отличие от каналов, сокеты позволяют передавать данные в обоих направлениях, как в синхронном, так и в асинхронном режиме. Большинство программистов используют для работы с сокетами различные библиотеки высокого уровня, однако высокоуровневые библиотеки, как правило, не позволяют задействовать всю мощь и многообразие сокетов. Наглядный пример многообразия – файловые сокеты. Программистам Windows должны быть знакомы сетевые сокеты, которые обычно организуют обмен данными с помощью протоколов семейства TCP/IP, однако в Unix есть и другие типы сокетов, специально предназначенные для обмена данными между локальными процессами.

Сокеты в файловом пространстве имен

Сокеты в файловом пространстве имен (file namespace, их еще называют «сокеты Unix») используют в качестве адресов имена файлов специального типа. Важной особенностью этих сокетов является то, что соединение с их помощью локального и удаленного приложений невозможно, даже если файловая система, в которой создан сокет, доступна удаленной операционной системе. В следующем фрагменте кода мы создаем сокет и связываем его с файлом **socket.soc** (полный текст программы-сервера **fserver.c** можно найти на диске):

```
sock = socket(AF_UNIX, SOCK_DGRAM, 0);
if (sock < 0)
{
    perror("socket failed");
}
```

```
return EXIT_FAILURE;
}
srvr_name.sa_family = AF_UNIX;
strcpy(srvr_name.sa_data, "socket.soc");
if (bind(sock, &srvr_name, strlen(srvr_name.sa_data) +
    sizeof(srvr_name.sa_family)) < 0)
{
    perror("bind failed");
return EXIT_FAILURE;
}
```

Константы и функции, необходимые для работы с сокетами в файловом пространстве имен, объявлены в файлах **sys/types.h** и **sys/socket.h**. Как и файлы, сокеты в программах представлены дескрипторами. Дескриптор сокета можно получить с помощью функции **socket(2)**. Первый параметр этой функции – домен, к которому принадлежит сокет. Домен сокета обозначает тип соединения (а не доменное имя Интернета, как вы могли бы подумать). Домен, обозначенный константой **AF_UNIX**, соответствует сокетам в файловом пространстве имен. Второй параметр, **socket()**, определяет тип сокета. Значение **SOCK_DGRAM** указывает датаграммный сокет (я предпочитаю этот вариант написания используемому в [1] «дейтаграммный»). Датаграммные сокеты осуществляют ненадежные соединения при передаче данных по сети и допускают широковещательную передачу данных. Другой часто используемый тип сокетов – **SOCK_STREAM** соответствует потоковым сокетам, реализующим соединения «точка-точка» с надежной передачей данных. Впрочем, в пространстве файловых имен датаграммные сокеты так же надежны, как и потоковые сокеты. Третий параметр функции **socket()** позволяет указать протокол, используемый для передачи данных. Мы оставляем значение этого параметра равным нулю. В случае ошибки функция **socket()** возвращает **-1**.

После получения дескриптора сокета мы вызываем функцию **bind(2)**, которая связывает сокет с заданным адресом (связывать сокет с адресом необходимо в программе-сервере, но не в клиенте). Первым параметром функции является дескриптор, а вторым – указатель на структуру **sockaddr** (переменная **srvr_name**), содержащую адрес, на котором регистрируется сервер (третий параметр функции – длина структуры, содержащей адрес). Вместо общей структуры **sockaddr** для сокетов Unix (сокетов в файловом пространстве имен) можно использовать специализированную структуру **sockaddr_un**. Поле **sockaddr.sa_family** позволяет указать семейство адресов, которым мы будем



НА ДИСКЕ

• Код примеров статьи

пользоваться. В нашем случае это семейство адресов файловых сокетов Unix `AF_UNIX`. Сам адрес семейства `AF_UNIX` (поле `sa_data`) представляет собой обычное имя файла сокета. После вызова `bind()` наша программа-сервер становится доступна для соединения по заданному адресу (имени файла).

При обмене данными с датаграммными сокетами мы используем не функции `write()` и `read()`, а специальные функции `recvfrom(2)` и `sendto(2)`. Эти же функции могут применяться и при работе с потоковыми сокетами, но в соответствующем примере мы воспользуемся «сладкой парочкой» `read()/write()`. Для чтения данных из датаграммного сокета мы используем функцию `recvfrom(2)`, которая по умолчанию блокирует программу до тех пор, пока на входе не появятся новые данные.

```
bytes = recvfrom(sock, buf, sizeof(buf), 0, &rcvr_name, &namelen);
```

Вызывая функцию `recvfrom()`, мы передаем ей указатель на еще одну структуру типа `sockaddr`, в которой функция возвращает данные об адресе клиента, запросившего соединение (в случае файловых сокетов этот параметр не несет полезной информации). Последний параметр функции `recvfrom()` – указатель на переменную, в которой будет возвращена длина структуры с адресом. Если информация об адресе клиента нас не интересует, мы можем передать значения `NULL` в предпоследнем и последнем параметрах. По завершении работы с сокетом мы закрываем его с помощью «файловой» функции `close()`. Перед выходом из программы-сервера следует удалить файл сокета, созданный в результате вызова `socket()`, что мы и делаем с помощью функции `unlink()`.

Если программа-сервер показалась вам простой, то программа-клиент (**fsclient.c**) будет еще проще. Мы открываем сокет с помощью функции `socket()` и передаем данные (тестовую строку) серверу с помощью «напарника» `recvfrom()`, функции `sendto(2)`:

```
srvr_name.sa_family = AF_UNIX;
strcpy(srvr_name.sa_data, SOCK_NAME);
strcpy(buf, "Hello, Unix sockets!");
sendto(sock, buf, strlen(buf), 0, &srvr_name,
strlen(srvr_name.sa_data) + sizeof(srvr_name.sa_family));
```

Первый параметр функции `sendto()` – дескриптор сокета, второй и третий параметры позволяют указать адрес буфера для передачи данных и его длину. Четвертый параметр предназначен для передачи дополнительных флагов. Предпоследний и последний параметры несут информацию об адресе сервера и его длине, соответственно. Если при работе с датаграммными сокетами вызвать функцию `connect(2)` (см. ниже), то можно не указывать адрес назначения каждый раз (достаточно указать его один раз, как параметр функции `connect()`). Перед вызовом функции `sendto()` нам надо заполнить структуру `sockaddr` (переменную `srvr_name`) данными об адресе сервера. После окончания передачи данных мы закрываем сокет с помощью `close()`. Если вы запустите программу-сервер, а затем программу-клиент, то сервер распечатает тестовую строку, переданную клиентом.

Парные сокеты

Сокеты в файловом пространстве имен похожи на именованные каналы тем, что для идентификации сокетов используются файлы специального типа. В мире сокетов есть и аналог неименованных каналов – парные сокеты (`socket pairs`). Как и неименованные каналы, парные сокеты создаются парами и не имеют имен. Естественно, что область применения парных сокетов та же, что и у неименованных каналов, – взаимодействие между родительским и дочерним процессом. Так же как и в случае неименованного канала, первый из дескрипторов используется одним процессом, второй – другим. В качестве примера использования парных сокетов мы рассмотрим программу **sockpair.c**, создающую два процесса с помощью `fork()`. Дочерние процессы **sockpair.c** используют парные сокеты для обмена вежливым английским приветствием.

```
#include <sys/types.h>
#include <sys/socket.h>
#include <stdlib.h>
#include <stdio.h>
```

```
#include <errno.h>
```

```
#define STR1 "How are you?"
#define STR2 "I'm ok, thank you."
#define BUF_SIZE 1024
```

```
int main(int argc, char ** argv)
{
    int sockets[2];
    char buf[BUF_SIZE];
    int pid;
```

```
if (socketpair(AF_UNIX, SOCK_STREAM, 0, sockets) < 0)
```

```
{
    perror("socketpair() failed");
    return EXIT_FAILURE;
}
```

```
pid = fork();
```

```
if (pid != 0)
```

```
{
    close(sockets[1]);
    write(sockets[0], STR1, sizeof(STR1));
    read(sockets[0], buf, sizeof(buf));
    printf("%s\n", buf);
    close(sockets[0]);
}
```

```
else
```

```
{
    close(sockets[0]);
    read(sockets[1], buf, sizeof(buf));
    printf("%s\n", buf);
    write(sockets[1], STR2, sizeof(STR2));
    close(sockets[1]);
}
```

Парные сокеты создаются функцией `socketpair(2)`. У функции `socketpair()` четыре параметра. Первые три параметра функции те же, что и у `socket()`, а четвертым параметром является массив из двух переменных, в которых возвращаются дескрипторы. Дескрипторы сокетов, возвращенные `socketpair()`, уже готовы к передаче данных, так что мы сразу можем применять к ним функции `read()/write()`. После вызова `fork()` каждый процесс получает оба дескриптора, один из которых он должен закрыть. Для закрытия сокета мы используем функцию `close()`.

При взгляде на интерфейс программирования парных сокетов может возникнуть вопрос: а почему, собственно, эти функции относятся к сокетами? Ведь при работе с ними мы не используем ни адреса, ни модель «клиент-сервер». Это верно, но заметьте, что функции `socketpair()` передаются значения домена и типа сокета, так что и формально, и с точки зрения реализации в системе мы используем настоящие сокеты. Следует отметить, что указание домена в функции `socketpair()` выглядит явно излишне, поскольку для этой функции система поддерживает только сокеты в домене `AF_UNIX` (вполне логичное ограничение, если учесть, что парные сокеты не имеют имен и предназначены для обмена данными между родственными процессами).

Сетевые сокеты

Мы переходим к рассмотрению самого важного и универсального типа сокетов – сетевых сокетов. Думаю, что о значении, которое имеют сетевые сокеты в Unix-системах, распространяться не нужно. Даже если вы пишете систему приложений, предназначенных для работы на одном компьютере, рассмотрите возможность использования сетевых сокетов для обмена данными между этими приложениями. Возможно, в будущем ваш программный комплекс наберет мощь и возникнет необходимость распределить его компоненты на нескольких машинах. Использование сетевых сокетов сделает процесс масштабирования



«<< проекта безболезненным. Впрочем, у сетевых сокетов есть и недостатки. Даже если сокет используется для обмена данными на одной и той же машине, передаваемые данные должны пройти все уровни сетевого стека, что отрицательно сказывается на быстродействии и нагрузке на систему.

В качестве примера мы рассмотрим комплекс из двух приложений, клиента и сервера, использующих сетевые сокет для обмена данными. Текст программы сервера вы найдете в файле **netserver.c**, ниже мы приводим некоторые фрагменты. Прежде всего, мы должны получить дескриптор сокета:

```
sock = socket(AF_INET, SOCK_STREAM, 0);
if (socket < 0)
{
printf("socket() failed: %d\n", errno);
return EXIT_FAILURE;
}
```

В первом параметре функции **socket()** мы передаем константу **AF_INET**, указывающую на то, что открываемый сокет должен быть сетевым. Значение второго параметра требует, чтобы сокет был потоковым. Далее мы, как и в случае сокета в файловом пространстве имен, вызываем функцию **bind()**:

```
serv_addr.sin_family = AF_INET;
serv_addr.sin_addr.s_addr = INADDR_ANY;
serv_addr.sin_port = htons(port);
if (bind(sock, (struct sockaddr *) &serv_addr, sizeof(serv_addr)) < 0)
{
printf("bind() failed: %d\n", errno);
return EXIT_FAILURE;
}
```

Переменная **serv_addr**, — это структура типа **sockaddr_in**. Тип **sockaddr_in** специально предназначен для хранения адресов в формате Интернета. Самое главное отличие **sockaddr_in** от **sockaddr_un** — наличие параметра **sin_port**, предназначенного для хранения значения порта. Функция **htons()** переписывает двухбайтовое значение порта так, чтобы порядок байтов соответствовал принятому в Интернете (см. врезку). В качестве семейства адресов мы указываем **AF_INET** (семейство адресов Интернета), а в качестве самого адреса — специальную константу **INADDR_ANY**. Благодаря этой константе наша программа-сервер зарегистрируется на всех адресах той машины, на которой она выполняется.

Чтобы понять, что мы должны делать дальше, давайте вспомним, как работает сетевая подсистема Unix и, в данном случае, любой другой ОС. Сетевой сервер должен уметь выполнять запросы множества клиентов одновременно (наш сервер **netserver.c** фактически может обработать запрос только одного клиента, но речь сейчас идет об

общем случае). При этом в соединениях «точка-точка», например, при использовании потоковых сокетов, у сервера для каждого клиента должен быть открыт отдельный сокет. Из этого следует, что мы не должны устанавливать соединение с клиентом через сам сокет **sock**, предназначенный для прослушивания входящих запросов (обычно, при использовании сетевых сокетов мы и не можем этого сделать), иначе все другие попытки соединиться с сервером по указанному адресу и порту будут заблокированы. Вместо этого мы вызываем функцию **listen(2)**, которая переводит сервер в режим ожидания запроса на соединение:

```
listen(sock, 1);
```

Второй параметр **listen()** — максимальное число соединений, которые сервер может обрабатывать одновременно. Далее мы вызываем функцию **accept(2)**, которая устанавливает соединение в ответ на запрос клиента:

```
newsock = accept(sock, (struct sockaddr *) &cli_addr, &clen);
if (newsock < 0)
{
printf("accept() failed: %d\n", errno);
return EXIT_FAILURE;
}
```

Получив запрос на соединение, функция **accept()** возвращает новый сокет, открытый для обмена данными с клиентом, запрошившим соединение. Сервер как бы перенаправляет запрошенное соединение на другой сокет, оставляя сокет **sock** свободным для прослушивания запросов на установку соединения. Второй параметр функции **accept()** содержит сведения об адресе клиента, запросившего соединение, а третий параметр указывает размер второго. Так же, как и при вызове функции **recvfom()**, мы можем передать **NULL** в последнем и предпоследнем параметрах. Для чтения и записи данных сервер использует функции **read()** и **write()**, а для закрытия сокетов, естественно, **close()**.

В программе-клиенте (**netclient.c**) нам, прежде всего, нужно решить задачу, с которой мы не сталкивались при написании сервера, а именно: выполнить преобразование доменного имени сервера в его сетевой адрес. Разрешение доменных имен выполняет функция **gethostbyname()**:

```
server = gethostbyname(argv[1]);
if (server == NULL)
{
printf("Host not found\n");
return EXIT_FAILURE;
}
```

Функция получает указатель на строку с Интернет-именем сервера (например, **www.unix.com** или **192.168.1.16**) и возвращает указатель на структуру **hostent** (переменная **server**), которая содержит имя сервера в приемлемом для дальнейшего использования виде. При этом, если необходимо, выполняется разрешение доменного имени в сетевой адрес. Далее мы заполняем поля переменной **serv_addr** (структуры **sockaddr_in**) значениями адреса и порта:

```
serv_addr.sin_family = AF_INET;
strncpy((char *)&serv_addr.sin_addr.s_addr,
(char *)server->h_addr, server->h_length);
serv_addr.sin_port = htons(port);
```

Программа-клиент открывает новый сокет с помощью вызова функции **socket()** аналогично тому, как это делает сервер (дескриптор сокета, который возвращает **socket()** мы сохраним в переменной **sock**), и вызывает функцию **connect(2)** для установки соединения:

```
if (connect(sock, &serv_addr, sizeof(serv_addr)) < 0)
{
printf("connect() failed: %d", errno);
return EXIT_FAILURE;
}
```

Теперь сокет готов к передаче и приему данных. Программа-клиент считывает символы, вводимые пользователем в окне терминала. Когда пользователь нажимает **Enter**, программа передает данные серверу, ждет ответного сообщения сервера и распечатывает его.

ОСТРОКОНЕЧНИКИ И ТУПОКОНЕЧНИКИ

Именно так переводятся на русский язык термины «little-endian» и «big-endian». В компьютерной литературе эти термины обозначают порядок байтов, используемый процессором для представления простых многобайтовых типов (например, 32-битного целого). В оригинале же (то есть в сказочной повести Дж. Свифта «Гулливер в стране лилипутов») так именовались враждебные общественные течения, приверженцы которых придерживались противоположных взглядов на порядок очистки яйца от скорлупы. Разногласия между остроконечниками и тупоконечниками даже стали причиной войны между Лилипутией и враждебным государством Блефуску.

Впрочем, в компьютерном мире проблемы порядка байтов могут достигать совсем не лиллипутских размеров. На процессорах Intel порядок байтов остроконечный, а, например, в системах MacOS X — Power PC Sun SPARC — тупоконечный (если учесть,

что Apple отказалась от PowerPC, а Sun заменяет RISC-архитектуры на Opteron, получится, что остроконечники побеждают).

Однако протоколы Интернета используют «тупоконечный» порядок байтов. Для того чтобы избежать путаницы, во всех системах, включая «тупоконечников» рекомендуется использовать функцию **htons()**. Эта функция «знает» порядок байтов в системе и, если нужно, приводит его в соответствие с принятым в протоколах TCP/IP. В русскоязычном Интернете одно время кочевала статья, в которой утверждалась (впрочем, шутливо), что чуждый для Intel порядок байтов и вызванная этим необходимость в дополнительной операции перестановки являются результатом заговора со стороны гигантских софтверных компаний. В книге [1] используются термины «прямой порядок байтов» для little-endian и «обратный порядок байтов» для big-endian.

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА:

1. Стивенс У., UNIX: **Разработка сетевых приложений.** – СПб.: Питер, 2004
2. W. R. Stevens, S. A. Rago, **Advanced Programming in the UNIX® Environment: Second Edition**, Addison Wesley Professional, 2005

На протяжении этой статьи мы несколько раз упоминали не-блокирующие сокеты. Остановимся на них немного подробнее. О не-блокирующих сокетах вам нужно знать, прежде всего, то, что ими можно не пользоваться. Благодаря многопоточному программированию мы можем использовать блокирующие сокеты во всех ситуациях (и тогда, когда нам нужно обрабатывать несколько сокетов одновременно, и тогда, когда нам требуется возможность прервать операцию, выполняемую над сокетом). Рассмотрим, тем не менее, две функции, необходимые для работы с не-блокирующими сокетами. По умолчанию функция `socket()` создает блокирующий сокет. Чтобы сделать его не-блокирующим, мы используем функцию `fcntl(2)`:

```
sock = socket(PF_INET, SOCK_STREAM, 0);
fcntl(sock, F_SETFL, O_NONBLOCK);
```

Теперь любой вызов функции `read()` для сокета `sock` будет возвращать управление сразу же. Если на входе сокета нет данных для чтения, функция `read()` вернет значение `EAGAIN`. Для проверки состояния не-блокирующих сокетов можно воспользоваться функцией `select(2)`. Функция `select()` способна проверять состояние нескольких дескрипторов сокетов (или файлов) сразу. Первый параметр функции – количество проверяемых дескрипторов. Второй, третий и четвертый параметры функции представляют собой наборы дескрипторов, которые следует проверять, соответственно, на готовность к чтению, записи и на наличие исключительных ситуаций. Сама функция `select()` – блокирующая, она возвращает управление, если хотя бы один из проверяемых сокетов готов к выполнению соответствующей операции. В качестве последнего параметра функции `select()` можно указать интервал времени, по прошествии которого она вернет управление в любом случае (тайм-аут). Вызов `select()` для проверки наличия входящих данных на сокете `sock` может выглядеть так:

```
fd_set set;
struct timeval interval;
FD_SET(sock, &set);
tv.tv_sec = 1;
tv.tv_usec = 500000;
...
select(1, &set, NULL, NULL, &tv);
if (FD_ISSET(sock, &set)
{
// Есть данные для чтения
}
```

Все, что касается функции `select()` теперь объявляется в заголовочном файле `sys/select.h` (раньше объявления элементов функции `select()` были разбросаны по файлам `sys/types.h`, `sys/time.h` и `stdlib.h`). В приведенном фрагменте кода, `FD_SET` и `FD_ISSET` – макросы, предназначенные для работы с набором дескрипторов `fd_set`.

На этом мы закончим знакомство с увлекательным миром межпроцессного взаимодействия Linux. Следующая статья будет посвящена управлению процессами, сигналами и потокам. **LXF**

ЧЕРЕЗ МЕСЯЦ

Мы поговорим о сигналах Unix и потоках (которые, на самом деле, практически не отличаются от процессов).

SUPERMICRO®

РЕВОЛЮЦИЯ В СЕРВЕРОСТРОЕНИИ



Серверы TRINITY на базе платформ SUPERMICRO 2-Way Dual Core AMD Opteron (2-х процессорные двоядерные конфигурации)

Производительность двоядерных процессоров, превышает одноядерные процессоры на 70 - 90 %. Заказывая 2-х процессорную двоядерную конфигурацию Вы получаете производительность 4-х процессорного сервера по цене 2-х процессорного.

В начале июля компания ТРИНИТИ представила серверные системы на базе двоядерных процессоров AMD Opteron серии 200. На сегодня доступны двухпроцессорные системы на базе платформ Supermicro:

Trinity Revolution На базе Supermicro® H8DA8 # 17181



Case: Supermicro CSE-743S1-650w/ 8xHS HDD
CPU: 2 x AMD Opteron 265 Dual-Core
RAM: 2GB DDR PC3200 ECC REG
HDD: 2 x 73GB SCSI

Гарантия 3 года. Цена от: **\$ 4669**

Trinity Revolution На базе Supermicro® H8DAE # 17190



Case: Supermicro CSE-743S1-650w/ 8xHS HDD
CPU: 2 x AMD Opteron 265 Dual-Core
RAM: 2GB DDR PC3200 ECC REG
RAID: LSI MegaRAID 320-1+BBU
HDD: 3 x 73GB SCSI, RAID5

Гарантия 3 года. Цена от: **\$ 5289**

Trinity Revolution На базе Supermicro® H8DAE # 17191



Case: Supermicro CSE-743S2-760w/ 8xHS HDD
CPU: 2 x AMD Opteron 275 Dual-Core
RAM: 4GB DDR PC3200 ECC REG
RAID: LSI MegaRAID 320-2x+BBU
HDD: 6 x 73GB SCSI

Гарантия 3 года. Цена от: **\$ 8989**

Trinity Revolution На базе Supermicro® AS1020A-8 (H8DAR-8) # 17192



Case: Supermicro CS812S-420w/ 3xHS HDD
CPU: 2 x AMD Opteron 275 Dual-Core
RAM: 2GB DDR PC3200 ECC REG
RAID: LSI MegaRAID 320-1+BBU
HDD: 2 x 73GB SCSI

Гарантия 3 года. Цена от: **\$ 6619**

Trinity Revolution На базе Supermicro® AS1020A-T (H8DAR-T) # 17193



Case: Supermicro CS813T-500w/ 4xHS HDD
CPU: 2 x AMD Opteron 265 Dual-Core
RAM: 1GB DDR PC3200 ECC REG
HDD: 4 x 200GB SATA

Гарантия 3 года. Цена от: **\$ 4719**

Специальное предложение подписчикам **LINUX FORMAT** предъявите этот купон и Вы получите скидку

3%

TRINITY
CORPORATE IT PROJECTS

(812) 327-5960
(095) 232-9230
www.trinitygroup.ru

Любые вопросы по серверам и системам хранения данных на форуме: www.3nity.ru

КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ

LaTeX — КОМПЬЮТЕРНАЯ ТИПОГРАФИЯ

ЧАСТЬ 1 Евгений Балдин начинает новую серию статей, посвященных LaTeX — истинной TeXнологии. Создание книг еще никогда не было таким интересным.

Человеческая цивилизация зависит от книг. Передача знаний от поколения к поколению это то, что делает человека разумным. Написание книги всегда было/есть/будет одним из самых сложных видов деятельности. LaTeX берет на себя техническую часть по подготовке рукописи, оставляя человеку больше времени на творчество, и, в тоже время, позволяя ему контролировать весь процесс создания от начала и до конца.

«Если кто-то другой набирает ваше произведение, то у вас нет возможности контролировать появление ошибок; если же вы выполняете эту работу самостоятельно, то можете винить только себя.» — Дональд Э. Кнут.

Немного истории

Есть популярная идея по поводу того, что «история учит тому, что ничему не учит». Возможно, это так. Но чтобы понять логику текущих событий и явлений, все-таки необходимо знать как «оно» зародилось и почему «оно» до сих пор существует.

Доисторический период

Сначала вообще не было компьютеров, и люди писали все вручную. Но прогресс неумолим — и вслед за печатной машиной появилась коммерческая выгода от создания книг. Время шло, процесс печати удешевлялся — все были довольны, пока не случилось вот что...

Роль Человека в истории

Дональд Эрвин Кнут (Donald Ervin Knuth) является одним из немногих людей, благодаря которым информатика заслуженно носит звание научной дисциплины. Произведение, которое принесло ему широкую известность, это «пятитомник» «Искусство программирования»¹. После того, как в 1975 году был издан третий том «пятитомника», издатель окончательно избавился от печатной машины с металлическим набором типа «монотип» и заменил его на фотонаборное устройство. Результат превзошел все ожидания: получив оттиски, сделанные по новой технологии, Д.Э. Кнут, который как раз подготовил второе издание второго тома,

сильно загрустил. Сама мысль, что книги, написание которых он потратил свыше пятнадцати лет, будут так плохо выглядеть, не давала Кнуту покоя.

Новые машины были не аналоговыми, а дискретными. Буквы составлялись из точек — «это объект для компьютерной науки» — подумал Кнут и решил научить компьютер делать буквы из точек такими как надо, то есть красивыми. Поначалу задача казалась несложной. Кнут потом признавал,

что это был его личный рекорд по недооценке сложности проекта.

«Лучший способ разобраться до конца — это попробовать научиться этому компьютер.» — Д.Э. Кнут.

Примерно через десять лет после начала работы над проектом системы METAFONT (создание шрифтов) и TeX (лучшая программа разбиения абзацев на строки) были стабилизированы (версия 2.7 для METAFONT и 3.1 для TeX). Кнут отошел от активной разработки. В дальнейшем METAFONT и TeX модифицировались только для целей исправления ошибок. На текущий момент рабочая версия METAFONT равна 2.71828, а TeX — 3.141592. Кнут завещал, что после его смерти версии будут заморожены и равны числу e и числу π соответственно, а все оставшиеся неисправленные ошибки будут считаться особенностями реализации.

На текущий момент TeX, скорее всего, самая свободная от ошибок программа. Код TeX выпускался отдельной книгой «TeX: The Program» (ISBN: 0201134373), за обнаружение ошибки в своей программе Кнут выплачивает вознаграждение. TeX является примером свободной программы, которая возникла в академической среде задолго до наступления эпохи GPL.

«Математическая формула не может быть чьей-то «собственностью»! Она принадлежит Богу.» — Д.Э. Кнут.

Сегодня мастер на пенсии и все свое время посвящает написанию «пятитомника». На его домашней страничке можно заметить, что дело явно движется. Ждем результата с нетерпением.

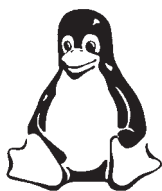
Техническое отступление

В основу TeX была заложена относительно простая идея. TeX работает только с боксами (box) и клеем (glue). Элементарные боксы — это буквы, которые объединяются в боксы-слова, которые в свою очередь сливаются в боксы-строки, боксы-абзацы и так далее. Между боксами «разлит» клей, который имеет ширину по умолчанию и степени увеличения/уменьшения этой ширины. Объединяясь в бокс более высокого порядка, элементарные боксы могут шевелиться, но как только найдено оптимальное решение, это состояние замораживается и полученный бокс выступает как единое целое. Наилучшее решение находится с помощью системы штрафов за то, что клея использовано больше или меньше оптимального значения, а также за разрывы абзаца в неподходящем месте. Чем меньше штрафа было получено, тем размещение «красивее». В зависимости от системы штрафов меняется форматирование.

Эмблемы TeX и METAFONT, созданные Дуайном Библи, взяты с домашней странички Д.Э.Кнута. Пингвина, судя по заголовку EPS, создал сотрудник Adobe Systems Inc. Нил Такер (Neal Tucker).



L^AT_EX
в России



¹ На сайте мастера <http://www-cs-faculty.stanford.edu/~knuth/> можно найти предварительные версии пока не опубликованных глав четвертого тома.

Первоначально Кнут предполагал, что у *TeX* будет множество модификаций, ибо каждая типография будет держать мастера-*TeX*ника для создания своей версии *TeX* под свои нужды. Вначале *TeX* не являлся в полном смысле языком программирования. Управляющие конструкции были добавлены в него позже, когда стало понятно, что развитие *TeX*нологии пошло совсем по другому пути.

А началось все с Лесли Лэмпорта, который в начале 80-х годов начал разработку издательской системы *LaTeX*, в основе которой лежал *TeX*. *LaTeX* представляет из себя набор макросов на языке *TeX*, позволяющих решить ту или иную задачу. Иными словами, это сборник рецептов. Чтобы выбрать сценарий стирки в автоматической стиральной машине, нет необходимости думать в терминах числа оборотов, уровня воды и количества порошка — достаточно просто выбрать готовое решение. Чтобы пользоваться системой *LaTeX*, не надо быть *TeX*ником — достаточно выбрать готовый стиль и использовать несколько простых команд в зависимости от того, что нужно сделать.

Дистрибутивы LaTeX

LaTeX, собственно говоря, как и Linux, не является монолитной программой. Он состоит из набора пакетов/программ, причем не фиксированного, что позволяет создавать дистрибутивы, преследующие ту или иную цель.

На сегодня все дистрибутивы *LaTeX* имеют общий корень, и этот корень носит название CTAN или The Comprehensive TeX Archive Network (<http://www.ctan.org>). CTAN — это репозиторий, в который стекаются все сколь-нибудь стоящие наработки в области *TeX*-строения. CTAN — это множество зеркалируемых серверов по всему миру. Модель была настолько успешна, что ее взяло на вооружение Perl-сообщество, организовав CPAN — The Comprehensive Perl Archive Network (<http://www.cpan.org>).

Наиболее известным в среде GNU/Linux является дистрибутив *teTeX*. (<http://www.tug.org/tetex/>). Если вы не знаете, что за дистрибутив стоит на вашем компьютере, то это, скорее всего, *teTeX*. Этот дистрибутив был создан Томасом Эшером (Thomas Esser). Первая публичная версия в 1994 влезала на три дискеты. В мае 2006 года Томас принял решение о прекращении поддержки своего детища в пользу настоящего флагамена *TeX*-сообщества (TUG — TeX User Groups, <http://www.tug.org/>) — дистрибутива *TeX Live* (<http://www.tug.org/texlive/>).

TeX Live создавался как дистрибутив, который можно было запускать прямо с CD. Базой для дистрибутива стал *teTeX*. Первая версия *TeX Live* была выпущена в 1996 году. Со временем дистрибутив рос, и к 2003 году *TeX Live* стал «влезать» только на DVD. Каждый год выпускается новая версия дистрибутива. Текущая нумерация идет по номеру года. *TeX Live* поддерживает все сколько-нибудь распространенные платформы. *TeX*, вообще, отличается исключительной переносимостью.

TeX Live — это свободное ПО. «Хранители» дистрибутива используют определение понятия свободы, принятое FSF (<http://www.gnu.org/philosophy/free-sw.html>) или сообществом Debian (http://www.debian.org/social_contract#guidelines). В случае конфликтов этих определений обычно принимается сторона FSF.

TeX Live был размещен на DVD, который прилагался к *LXF79*, так что этот дистрибутив вполне можно «потрогать руками» уже сейчас.

Запускаем LaTeX

Предполагается, что дистрибутив *LaTeX* на вашем компьютере уже установлен и настроен. Если это не так, то потревожьте по этому поводу суперпользователя. Обычно проблем с установкой не возникает. Возможные шероховатости с русским языком в современных дистрибутивах возникают по недоразумению. Основной дистрибутив GNU/Linux на сегодня это *teTeX*. В будущем, по возможности, рекомендуется переходить на *TeX Live*.

По традиции, для начала скажем «Здравствуй, мир!!!». Для этого в любом удобном для вас текстовом редакторе² создайте файл **helloworld.tex**, примерно следующего содержания:

```
%helloworld.tex
%выбор класса документа
```



```
\documentclass{article}
%Чтобы можно было использовать русские буквы в формулах
%но в случае использования предупреждать об этом
\usepackage{warn}[mathtext]
%выбор внутренней TeX-кодировки
%можно опустить - вызываю по привычке
\usepackage[T2A]{fontenc}
%выбор кодовой страницы документа
\usepackage[koi8-r]{inputenc}
%выбор языка документа
\usepackage[english,russian]{babel}
%начинать первый параграф с красной строки
\usepackage{indentfirst}
%конец преамбулы и начало текста
\begin{document}
%Поздравляем мир
\LARGE Здравствуй мир !!!
%конец текста
\end{document}
```

Комментарии, которые начинаются со знака % можно опускать. Все, что идет до `\begin{document}` называется преамбулой или «шапкой». Преамбула определяет вид итогового документа. Нет необходимости каждый раз набивать эти строчки с нуля. Для этого достаточно обучить текстовый редактор вставлять их автоматически при создании нового *tex*-файла. После создания текста его необходимо откомпилировать:

```
> latex helloworld.tex
This is pdfTeXk, Version 3.141592-1.30.3-2.2 (Web2C 7.5.5)
%&-line parsing enabled.
entering extended mode
(./helloworld.tex
LaTeX2e <2003/12/01>
...
(./helloworld.aux) [1] (./helloworld.aux)
Output written on helloworld.dvi (1 page, 240 bytes).
Transcript written on helloworld.log.
>
```

В качестве результата *LaTeX* выдает файл **helloworld.dvi**. Далее есть выбор:

- посмотреть результат с помощью *xdvi*:

Пример набора текста в текстовом редакторе — слева. Просмотр результата набора — справа.



² Лучше чтобы этим редактором оказался *emacs* — в каждой шутке есть только доля шутки.

```
<< > xdvi helloworld.dvi
• преобразовать dvi в PostScript3 и просмотреть его с помощью gv, а
потом распечатать на PostScript-принтере (если он есть, естественно):
> dvips helloworld
> gv helloworld.ps
> lpr helloworld.ps
• сделать PDF4, ну и, естественно, просмотреть его с помощью Acrobat
Reader:
> dvips helloworld
> ps2pdf helloworld.ps helloworld.pdf
> acroread helloworld.pdf
```

Во всех случаях вы увидите одно и то же:

Здравствуй мир !!!

Здравствуй, мир!!! от LaTeX.

Проблемы с компиляцией

Случается, что при наборе допускается ошибка, тогда при компиляции исходника LaTeX может затребовать дополнительную информацию.

Если просто запустить latex без каких-либо инструкций, то на экране появится приглашение:

```
> latex
This is pdfTeXk, Version 3.141592-1.30.3-2.2 (Web2C 7.5.5)
%&-line parsing enabled.
**
```

LaTeX ждет ввода имени текстового файла, чтобы начать его обработку. Можно прервать ожидание по ^C. Если же правильно задать файл при запуске latex, но при этом ошибиться в коде, то LaTeX выдаст сообщение об ошибке с номером строчки, где возникла проблема, и предложит сделать выбор:

```
! Undefined control sequence.
l.11 \errorinbody
? h
The control sequence at the end of the top line
of your error message was never \defed. If you have
misspelled it (e.g., \hobx), type `I' and the correct
spelling (e.g., \hbox'). Otherwise just continue,
and I'll forget about whatever was undefined.
? x
```

На запрос (?) можно ввести h, тогда будет выдана догадка о том, с чем может быть связана ошибка, или x, чтобы прервать компиляцию. В случае обычного перевода строки компиляция продолжится до следующей ошибки или до самого конца. Краткую информацию об управляющих командах можно получить введя ?.

Добрый совет: увидели ошибку – сразу же исправьте. Следующие предупреждения могут быть следствием предыдущей ошибки.

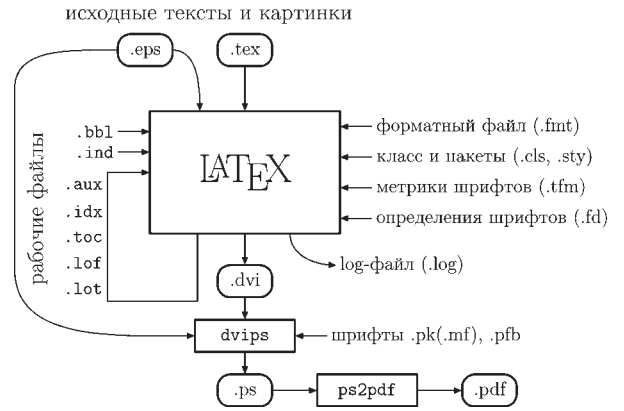
Интерактивный режим для работы с ошибками LaTeX – довольно мощный инструмент разбирательства, но на первых порах лучше следовать «Доброму совету». Текстовые редакторы, в которых предусмотрена поддержка редактирования файлов LaTeX, обычно позволяют локализовать ошибку на основании выданного сообщения.

LaTeX-конвейер

В процессе работы LaTeX читает и записывает несколько файлов. Полезно знать, что это за файлы, и зачем они нужны.

³ PostScript — язык описания страниц, разработан Джоном Уорноком и Чаком Гешке из Adobe Systems Inc.. Интерпретаторы PostScript, аппаратные или программные (ghostscript), широко используются при печати документов.

⁴ PDF — Portable Document Format. Этот формат, как и PostScript, создан фирмой Adobe Systems. Является стандартом для электронной документации.



LaTeX-конвейер.

На вход подается текстовый файл с LaTeX-разметкой. Традиционно файл имеет расширение **tex**.

В качестве результата на выходе получается файл с тем же самым именем, что и на входе, но с расширением **dvi**. dvi – device-independent (не зависящий от устройства) формат, который хранит информацию о форматировании текста и размещении всех его элементов на странице, но без самих букв и картинок. Программы, преобразующие dvi-файл в другое представление, называются dvi-драйверами.

xdvi преобразует dvi-файл в картинку на экране монитора. Это очень продвинутый драйвер. Он реалистично представляет вид напечатанной страницы, поддерживает гиперссылки и позволяет организовать обратную связь с текстом. Еще одним популярным dvi-драйвером является dvips. dvips производит качественный PostScript, который уже можно распечатать на принтере либо напрямую (если принтер поддерживает PostScript аппаратно), либо через программный интерпретатор ghostscript. Существуют и другие dvi-драйверы: например, dvi2tty пытается перевести dvi обратно в текст, dvi2j переводит dvi в инструкции PCL для принтеров HP, dvi2pdf транслирует dvi в pdf. Обилие dvi-драйверов позволяет рассмотреть/напечатать dvi-файлы практически в любой ситуации.

Свободный программный интерпретатор Ghostscript (gs) в свою очередь позволяет преобразовывать PostScript-файлы (.ps) в другие форматы. Обычно PDF получают именно из PostScript с помощью скрипта ps2pdf.

Графика в LaTeX добавляется через eps-файлы. EPS или Encapsulated PostScript – это векторный графический формат, который представляет собой инструкции на языке PostScript с некоторыми ограничениями. Одно из основных требований заключается в том, что в заголовке eps-файла обязательно должны быть указаны его размеры (BoundingBox). Пример заголовка eps, сделанного из примера «Здравствуй, мир!!!»:

```
%!PS-Adobe-2.0 EPSF-2.0
%%Creator: dvips(k) 5.95b Copyright 2005 Radical Eye Software
%%Title: helloworld.dvi
%%BoundingBox: 148 651 288 668
%%DocumentFonts: SFRM1728
%%EndComments
```

Перечислим теперь остальные файлы, который участвуют в этом круговороте:

1) Внешние файлы.

- **fmt** Форматный файл. Содержит, главным образом, все команды LaTeX'a в предварительно откомпилированной форме. Также содержит информацию о переносах. При изменении значений глобальных переменных TeX требуется пересборка форматного файла: **texconfig init**.
- **cls, sty** Определение макета и структуры документа. Класс документа (**.cls**) выбирается с помощью инструкции **\documentclass**. Дополнительные возможности и изменение поведения класса по умолчанию достигается с помощью пакетов (**.sty**) выбираемых посредством инструкции **\usepackage**.
- **tfm** Метрики шрифтов. Размеры и правила взаимодействия литер друг с другом.

- **fd** Приведение внешних названий шрифтов к виду, принятому в *LaTeX*.
- **pf** Векторные TrueType шрифты.
- **pk (mf)** Растровые рк-шрифты по мере необходимости создаются из векторных tf-шрифтов (METAFONT) с необходимым разрешением. При печати качество не страдает, но при просмотре из-за низкого разрешения дисплея предпочтительнее векторные шрифты.

2) Файлы, создаваемые в процессе LaTeX-конвейера. Обычно для получения итоговой копии документа требуется несколько раз запустить *LaTeX*. При первом проходе записывается некоторое число информационно-файлов, которые при последующих проходах используются для нумерации ссылок, составления оглавления и тому подобное.

- **log** Файл протокола. Сюда стекается вся информация, имеющая отношение к компиляции. Фактически дублирует стандартный вывод на экран.
- **aux** Информация о перекрестных ссылках.
- **toc** Файл оглавления (table of contents).
- **lof** Список иллюстраций (list of figures).
- **lot** Список таблиц (list of tables).
- **bbl** Список литературы, сформированный с помощью программы *BibTeX*. На первых порах можно обойтись без этой технологии, но для серьезных проектов управление списком литературы становится не менее серьезной задачей.
- **ind** Предметный указатель, сформированный программой *MakeIndex*. В каждой уважающей себя и несущей полезную информацию книге есть такой. Для составления используются idx-файлы. Организация указателя – это отдельная задача.

Выше описан «классический» LaTeX-конвейер. Программа *LaTeX* может быть заменена на *pdf_latex*, тогда на выходе сразу будет получаться **pdf**, а графическая информация должна быть представлена в форматах **png** или **pdf**. Возможны и другие вариации, но в целом структура остается той же.

Литература

LaTeX уже сорок лет. За время своего существования многие из его частей существенно переделывались и совершенствовались. База же, в виде *TeX*, до сих пор остается стабильной основой. Видимо, поэтому документация к *LaTeX* устаревает с очень большой неохотой.

Доступных в России книг по *LaTeX* относительно немного. С другой стороны, если вы сможете добыть хоть одну из перечисленных в этом разделе, то для обычного набора вам, скорее всего, будет ее достаточно. Логичная организация позволяет *LaTeX* расширяться без особого ущерба для совместимости.

Для более подробного ознакомления с конкретными пакетами следует обратиться к документации, сопровождающей пакет. Чего-чего, а описаний в дистрибутиве *LaTeX* хватает. Также для поиска того или иного решения можно обратиться к сайту CTAN (<http://www.ctan.org>) или TUG (<http://www.tug.org>).

Классика

- Д.Э. Кнут. «Все про TeX»: Пер. с англ. – М.: Вильямс, 2003, 560 с., ISBN 5-8459-0382-3 (рус.)

Библия *TeX*. Для обычного набора текстов в *LaTeX* информация, собранная в этой книге, не обязательна. *TeX*пертам и тем, кто таковыми себя считает, читать по несколько раз. Книга содержит сквозной разноразмерный по сложности материал. При прочтении можно выбирать свой уровень.

- Д.Э. Кнут. «Все про METAFONT»: Пер. с англ. – М.: Вильямс, 2003, 384 с., ISBN 5-8459-0442-0 (рус.)

Все, что сказано про библию *TeX*, относится и к этой библии *METAFONT*. Если вы создаете иллюстрации с использованием *MetaPost*, то эту книгу следует прочитать хотя бы для общего развития.

- Д.Э. Кнут. «Компьютерная типография»: Пер. с англ. – М.: Мир, 2003, 686 с., ISBN 5-03-3361-0

Сборник статей Д. Кнута, написанных в процессе создания *TeX* и *METAFONT*. В книге подробно разобраны алгоритмы, которые легли в основу *TeX*, и перечислены те проблемы, что необходимо решить в

процессе создания «компьютерной типографии». Книга интересна и исторически – в плане становления одного из самых успешных компьютерных проектов.

Учебники и справочники

- Г. Грэтцер. «Первые шаги в LaTeX» Пер. с англ. – М.: Мир, 2000, 172 с., ISBN 5-03-0033661-0

Новичкам посвящается. Компактный учебник, позволяющий быстро освоить базовые приемы. Упор на математику.

- М. Гуссенс, Ф. Миттельбах и А. Самарин. «Путеводитель по пакету LaTeX и его расширению LaTeX2ε»: Пер. с англ. – Мир, 1999, 606 с., ISBN 5-03-003325-4

Очень полное и исчерпывающее справочное руководство по основным пакетам и приемам *LaTeX*. На английском недавно вышло второе издание этой книги. Возможно со временем она доберется и до нас. Это моя основная «рабочая» книга по *LaTeX*.

- М. Гуссенс, С. Ратц и Ф. Миттельбах. «Путеводитель по пакету LaTeX и его графическим расширениям. Иллюстрирование документов при помощи TeX'a и PostScript'a»: Пер. с англ. – Мир, 2002, 621 с., ISBN 5-03-003388-2

При подготовке основного путеводителя авторы обнаружили, что описание систем для создания рисунков по объему начинает превосходить базовый текст. Пришлось выделить это в отдельную книгу. Здесь есть все: от шахмат, нот и электронных схем до трюков с PostScript и *MetaPost*. Очень полезный учебник-справочник для тех, кто самостоятельно делает иллюстрации.

- М. Гуссенс, С. Ратц. «Путеводитель по пакету LaTeX и его Web-приложениям»: Пер. с англ. – Мир, 2001, 604 с., ISBN 5-03-003387-4

Лучше бы авторы описали бы еще несколько пакетов *LaTeX*. На мой взгляд, эта книга – дань моде. С другой стороны, здесь расписано все, что связано с PDF и что с ним можно сделать. Любителям XML посвящается.

- «Не очень краткое введение LaTeX2ε. Или LaTeX2ε за 94 минуты» Перевод Бориса Тоботраса от 1999 г. Электронная версия и исходные тексты доступны здесь: <http://xtalk.msk.su/TeX/>.

В названии все сказано. Самый доступный и маленький из имеющихся на сегодня русскоязычных учебников. Из недостатков: отсутствует информация о кириллизации. Полезно для начального изучения.

LaTeX в России

- С.М. Львовский. «Набор и верстка в системе LaTeX. – 3-е издание» – М.: МЦНМО, 2003, 448 с.

Хороший переплет. Основной упор на математику. Классика жанра. Автор ориентируется на нестандартную русификацию, которая имеет свои преимущества, хотя и спорные. Есть свободная электронная версия, которую можно взять, например, здесь: <http://www.mccme.ru/free-books/>

- И.А. Котельников, П.З. Чеботаев. «LaTeX по-русски. – 3-е издание» – Новосибирск: Сибирский хронограф, 2004, 496 с., ISBN 5-87550-195-2

Очень качественный учебник. Предыдущее 2-е издание было моей настольной книгой. К сожалению, качество переплета не очень высокое, что не позволяет активно работать с этой книгой.

- А.И. Рожено. «Искусство верстки в LaTeX'e» – Новосибирск: Изд. ИВМиГ СО РАН, 2005. – 398 с., ISBN 5-901548-25-6

Краткий и достаточно исчерпывающий справочник с упором на разработанные автором макросы. В январском номере Linux Format была моя рецензия на эту книгу. Обсуждаются особенности русского стиля. Из недостатков: есть привязка к альтернативной операционной системе.

Кроме перечисленного, полезной может оказаться электронная документация, созданная В. Сюткиным, которую можно скачать здесь:

http://www.sbras.nsc.ru/win/docs/TeX/LaTeX2e/docs_koi.html 



СОЗДАНИЕ УЧЕБНЫХ ПОСОБИЙ

Создаем видеопособие с Wink

Замучились бегать через три этажа, чтобы объяснить «девочкам из бухгалтерии» как скопировать файл в Nautilus? Нет ничего проще – создайте видеокурс «Gnome для чайников», следуя советам Сергея Яремчука!

Народная поговорка гласит: «Лучше один раз увидеть, чем сто раз услышать». Простое описание продукта без сопровождающих его экранных снимков не сможет передать достоинства и недостатки интерфейса, его слабые и сильные стороны. Но статический образ не может передать особенности работы с самой программой. Возможно, поэтому сегодня так популярны различные видеопособия, рассказывающие о том, как проделать ту или иную операцию.

Действительно, такое пособие лучше продемонстрирует любые характеристики как простого, так и сложного программного обеспечения, покажет другим, как можно выполнить некую задачу в специфической программной среде, поможет ответить на любые вопросы поддержки, так как простая демонстрация способна заменить тысячи слов. Ранее, чтобы создать видеопособие в Linux, приходилось прибегать к виртуальным машинам либо запускать Windows-приложения для захвата действий, производимых на рабочем столе при помощи Wine. Теперь ситуация изменилась, так как появились удобные инструменты, позволяющие создавать готовые презентационные ролики, рассказывающие о работе программ. Среди них базирующаяся на GTK утилита Istanbul (см. «Полезные ссылки»), очень удобный инструмент, позволяющий сохранить результат в видеофайл в формате Ogg Theora, либо передать поток на Icecast-сервер для дальнейшей трансляции пользователям, что очень удобно при обучении. Ранние решения по созданию видеопособий использовали протокол VNC и были очень неудобными в настройке и использовании. Эти проблемы решены в проекте vnc2swf, при помощи которого можно захватить сеанс VNC и записать его в формате Macromedia Flash или VNCLog (последний можно затем воспроизвести при помощи vncrec). И, наконец, Wink – весьма удобная, мощная и простая в использовании программа, написанная с помощью wxWidgets.

Знакомьтесь – Wink

В отличие от упомянутых выше программ, Wink, к сожалению, не является открытым ПО, но при этом бесплатен для персонального и бизнес-использования. С его помощью можно делать снимки с экрана в ручном и автоматическом режиме, затем добавлять к ним комментарии, подписи и кнопки. Созданный с его помощью сжатый Flash-файл на порядок меньше по размеру, чем файл, созданный в Istanbul и закодированный Ogg Theora. Такой

Рис. 1. В первом окне мастера указываются частота и область захвата.



Рис. 2. Wink готов к записи.

файл можно затем смотреть в любом проигрывателе, понимающем Flash версии 3 и выше. Подобный ролик лучше всего подходит для схематического представления действий пользователя, тогда как видео Theora предпочтительнее в случае быстрых изменений картинки и сложных изображений. Кроме захвата снимка экрана, в качестве входящей информации могут использоваться графические файлы BMP/JPG/PNG/TIFF/GIF. На выходе, кроме Macromedia Flash, может быть создан HTML, который также удобен для представления в Web. Если вам нужна готовая презентация, то можно использовать скомпилированный exe-файл; если требуется распечатка, то подойдут форматы PDF и PostScript. Можно также создать шаблон, который затем будет использоваться многократно. Кроме Linux, Wink доступен для Windows 98/ME/2000/XP.

Наше первое пособие

Установка сложностей не вызывает, требуется распаковать архив и запустить скрипт installer.sh. При работе с Wink разрешение экрана желательно установить не менее 800x600, а лучше – 1024x768, тогда при редактировании вам будет доступна большая область. Для создания нового проекта выберите File → New, после чего запустится мастер (Wizard). На первом шаге (рис.1) укажите область захвата, воспользовавшись выпадающим списком. Это может быть весь экран (Screen), отдельное окно (Window), либо часть экрана (rectangle). Последний последний вариант позволяет произвольно выбрать размеры в строке Region, либо воспользоваться одним из предустановленных значений. Для указания окна или произвольной области экрана воспользуйтесь кнопкой Choose. После нажатия на нее очертите при помощи мыши нужную область, либо укажите на окно приложения, после чего щелчком левой кнопкой запишите результат. В строке Region при этом должны появиться данные новой области, которые можно подкорректировать вручную. Если окно Wink мешает выбрать нужную область, его можно скрыть. Для этого отметьте галочкой пункт Hide Wink Window. В окне Timed Capture Rate, выставляем количество снимков экрана, которое будет производить Wink в секунду, минуту или час. Этот параметр будет использоваться в режиме автоматического захвата.

Теперь нажмите OK и перейдите ко второму окну мастера (рис.2). После выбора Minimize To Tray Wink спрячется в трей (если конечно он есть в используемом оконном менеджере) и будет ждать команду на запуск процесса записи. По умолчанию нажатие на клавишу Pause приведет к захвату одиночного изображения, комбинация Shift + Pause запустит/остановит автоматический захват. После того как запись будет окончена, вызови-

те мастер двойным щелчком по иконке и нажмите Finish. Появится окно редактирования проекта, показанное на рис.3.

Редактируем проект

Окно редактирования разбито на три части. Посередине отображается кадр, над которым вы работаете в данный момент времени, внизу показаны уменьшенные копии изображений всех кадров проекта, а справа расположено окно свойств (Properties).

Команды редактирования можно вызвать из меню, при помощи окна свойств или воспользовавшись горячими клавишами. Над расположенными внизу кадрами можно производить любые действия: вырезать, копировать, удалять, вставлять кадры из буфера обмена, копии или пустые кадры. Также при помощи меню Project можно обрезать (Crop All Frames) либо изменить размер всех кадров проекта (Resize All Frames).

Особенностью Wink является то, что можно не только редактировать положение курсора, но и изменять его форму. Включите показ курсора, выбрав Cursor. Для выбора нового курсора нажмите **Ctrl+R**, либо пиктограмму возле галочки. В появившемся окне укажите вид курсора (рис.4), и нажмите ОК. Если курсор появился не на своем месте, щелкните по нему в окне текущего кадра и перетащите на положенное ему место.

После компиляции проекта время показа каждого кадра будет одним и тем же. Поставив необходимую цифру в поле Stay in this frame for time можно задержать показ кадра на указанное время, чтобы пользователь смог подробнее рассмотреть элементы управления, либо для более подробного пояснения ситуации.

Кроме того, каждый кадр может иметь подпись. Для того, чтобы создать ее, выберите Frame Titles и при помощи мыши выделите область, в которую будет помещена подпись. Здесь же указываются шрифт и выравнивание. Поясняющий текст вводится в каждом кадре. Если в каком-либо кадре подпись не нужна, то в окне подписи необходимо ввести пробел, иначе после компиляции в этих кадрах будет видна надпись "This frame's title will come here ...". Таким же образом можно добавить и текстовое поле. Активируйте пункт Textbox, выберите общий вид поля (Choose Callout), шрифт и выравнивание. При включении текстового поля автоматически активируются кнопки Back и Next: захватите их мышью и перетащите на свое место. В дальнейшем при просмотре презентации в местах, где расположены эти кнопки воспроизведение будет останавливаться, и ждать дальнейших действий пользователя. В других кадрах эти кнопки можно активировать, установив галочку напротив нужного пункта в окне свойств.

При ознакомлении с материалом очень часто приходится обращаться к другим кадрам для уточнения информации, представленной на экране. Wink осуществляет переход между кадрами при помощи кнопок Go. Для активации такой кнопки выберите Goto Button 1(2) и затем щелчком по расположенной рядом пиктограмме укажите номер кадра, к которому должен быть произведен переход. При необходимости вместо кнопок можно использовать изображение. Для этого выберите Project → Choose Goto Button Image 1(2) и указываем на файл в формате BMP, JPEG, PNG, CUR, ANI, ICO, PCX и PNM. В этом же пункте можно выбрать изображение для кнопок Back и Next, а также фоновое изображение кадров проекта.

Когда все кадры будут готовы, сохраните проект в файл с расширением **wnk**. Осталось только скомпилировать результат. Сначала нажмите Project → Setting (F3), укажите имя выходного файла, формат (**swf** или **exe**) и частоту кадров. Сохраните настройки нажатием кнопки ОК. Затем выберите Project → Render (F7) и дождитесь окончания процесса компи-

ляции. Требуемое время будет зависеть от производительности компьютера, количества и размера кадров, а также наличия дополнительных элементов управления. Если выбран формат Flash, то в результирующем каталоге обнаружите два файла с расширениями **swf** и **html**. Последний содержит код, необходимый для запуска ролика в web-браузере.

Возможности Wink позволяют создать действительно хороший материал, который покажет и объяснит любое действие. Единственным минусом является отсутствие звукового сопровождения. Wink не предоставляет такой возможности, но его легко добавить и самому. Для записи с микрофона будем использовать Audacity (см. LXF79). Запустите приложение, создайте новый проект Файл → Новый (Ctrl+N), затем выставите параметры выходного файла Файл → Настройки. Для Flash-ролика будет достаточно 16-битного моно-сигнала с небольшой частотой дискретизации.

Сохраните настройки нажатием ОК. Теперь на передней панели выберите источник сигнала Mic и нажмите на красную кнопку Rec. Затем запустите на воспроизведение ваш Flash-ролик и комментируйте происходящее в микрофон. После окончания, остановите запись и сохраните файл. Не забудьте вырезать «лишнюю тишину» в начале файла. Для этого выделите нужный фрагмент и нажмите Вырезать (Ctrl+X). Для добавления звука можно использовать скрипт **edit.py** из комплекта **vnc2swf**, принимающий в качестве входных параметров имена Flash и MP3-файла. Поэтому, выберите Файл – Экспортировать в mp3 и сохраните запись с микрофона в MP3-файл. Теперь соедините их вместе:

```
$ edit.py -a voice.mp3 first.swf -owith_sound.swf
```

Если результат удовлетворительным, переименуйте **with_sound.swf** обратно в **first.swf**, чтобы не править HTML-код.

```
$ mv with_sound.swf first.swf
```

Презентация готова! Можно опубликовать ее на сайте или отослать пользователям. Для экспорта в другие форматы загляните в меню File.

Wink является мощным и одновременно простым и понятным в использовании средством, позволяющим создавать презентации практически любой сложности. В отличие от других подобных инструментов, пользователю предоставляется возможность полноценного вмешательства в конечный результат, путем коррекции кадров, добавления поясняющего текста и прочей информации, делающей восприятие информации более простым. **LXF**

Рис. 3. Окно редактирования проекта.

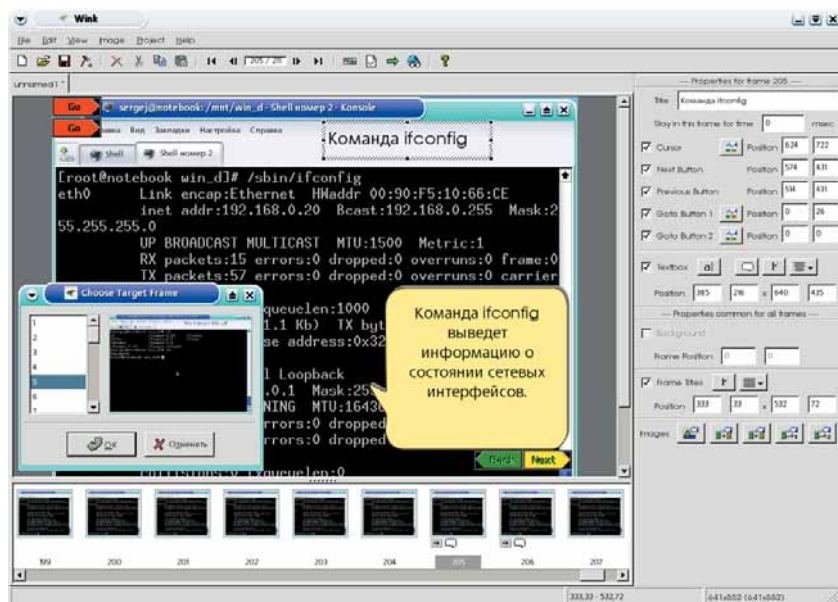
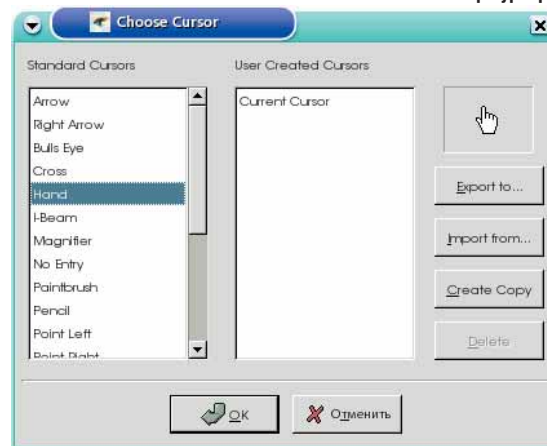


Рис. 4. Выбор курсора.



Вопрос? Ответ!

Если вы завязли в какой-то проблеме и чтение HOWTO не помогает, почему бы не написать нам? Наши эксперты помогут разобраться даже в самых сложных вопросах.

НАШИ ЭКСПЕРТЫ

Наши эксперты найдут ответ на самый трудный ваш вопрос. Если у вас проблемы с установкой, настройкой модема, сетью или еще чем-нибудь – просто напишите нам, а обо всем остальном мы позаботимся.

Владея Интернет-провайдером, а заодно подрабатывая редактором дисков для LXf, **Нейл Ботвик** скромно зовет себя мастером на все руки.



Для административных вопросов: **Кингс Коблер** является системным инженером и администратором Linux в Rackspace, и использует Linux 10 лет.



Адрес для писем с вопросами: email:lxf.answers@futurenet.co.uk. E-mail: answers@linuxformat.ru.

Вопрос по администрированию? Отправьте его Кингсу – sysadminqa@rackspace.co.uk.

Помогите нам помочь вам

- Для наилучшего ответа на ваш вопрос нам нужно знать как можно больше подробностей. Детально опишите конфигурацию системы. Если вы получили сообщение об ошибке, приведите текст сообщения и точно опишите вызвавшие его действия. Если у вас проблемы с оборудованием, то опишите его. Если Linux уже запущен, то выполните в root-терминале следующие команды и прикрепите к письму файл **system.txt**:

```
uname -a >>system.txt
lspci >>system.txt
lspci -vv >>system.txt
```

- Пожалуйста, помните, что сотрудники журнала НЕ являются авторами или разработчиками Linux, любых пакетов или дистрибутивов. Зачастую люди, отвечающие за приложения, выкладывают большую часть информации на web-сайты. Попробуйте почитать документацию!

Мы стараемся ответить на все вопросы. Если вы не нашли ответ на свой, это, возможно, потому, что мы уже ответили на похожий вопрос.

Fedora без сети

В Я установил Fedora Core 5, и все вроде прекрасно, кроме одного. Когда я пытаюсь выполнить Add/Remove Programs (Добавить/Удалить программы) или Package Updater (Обновление пакетов), у меня запрашивается пароль root. Затем я получаю сообщение об ошибке: «Не могу получить информацию о приложении» или «Не могу получить информацию об обновлении». Единственным отступлением от стандартной установки является отказ от Logical Volume Manager; вместо этого я вручную разбил диск на /boot (100 МБ), / (38 ГБ) и swap (1024 МБ). Да еще мой ПК не подключен к Интернету. Я пытался установить и с CD, и с вашего DVD приложения. Но получаю те же ошибки.

Альфред Мифсуд [Alfred Mifsud]

О Отсутствие Интернет-соединения и есть причина этих сообщений. Обе программы при работе пытаются прочитать информацию из online-репозитория программ. В случае Software Updater это неизбежно: по самой своей природе обновления новее пакетов на установочных носителях, так что без соединения с Интернетом не обойтись.

Во избежание этой ошибки с Add/Remove Programs, вам нужно отредактировать файлы репозитория: отключить все online-источники и добавить DVD. Делается это из-под root. Загрузите **/etc/yum.repos.d/fedora-core.repo** в ваш любимый текстовый редактор, найдите начало раздела **[core]** и закоментируйте строки **baseurl** и **mirrorlist**, поместив в начало каждой строки **#**. Затем добавьте новую строку вида

```
baseurl=file:///media/disk
```

Создается новый репозиторий на **/media/disk**, где смонтирован DVD. Затем вам следует отредактировать другие **.repo**-файлы и изменить все вхождения **enabled=1** на **enabled=0**. Теперь единственный активный репозиторий – это DVD, и запуск Add/Remove Software должен позволить вам установить приложения с диска. **НБ**



Отредактируйте файл конфигурации репозитория Fedora Core, чтобы менеджер пакетов работал без соединения с Интернетом.

Черепашья почта

В Мой почтовый сервер на основе Qmail отправляет почту с большой задержкой. Я пытался ускорить его, но ничего не вышло. Не могли бы вы перечислить, что может вызывать задержку?

Т.Мэтьюс [T Matthews]

О Наиболее вероятная причина задержек – запросы DNS. Первое и главное, убедитесь, что имя сервера нормально разрешается. Кроме того, для IP-адреса, используемого сервером для отправки почты, должна быть создана PTR-запись. Вы также можете ускорить запросы, запустив собственный локальный кэширующий сервер имен.

Также можно отключить DNS-запросы вообще. Если вы запускаете **qmail-smtpd** через **tcpserver**, то добавьте в его опции флаг **-H**, чтобы он не искал имена хостов в DNS, и удалите переменную среды **\$TCPREMOTEHOST**. Во избежание циклического использования этой опции для серверов по TCP-порту 53. Если это не так, то вы, вероятно, запускаете его через **inetd/xinetd**; и можете добавить в конфигурацию флаг **-Rt0**, в пункте **server_args** конфигурационного файла вашего **inetd/xinetd**. Это запретит Qmail выполнять запросы **ident** при установлении SMTP-соединения. Прделанные действия проявляют себя в задержке между установлением TCP-соед-

нения и появлением приглашения.

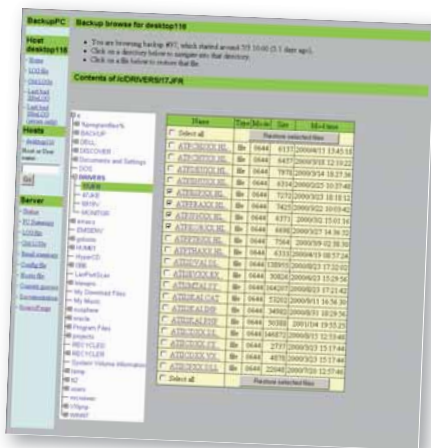
А если ваша очередь постоянно переполняется, можете добавить файл **/var/qmail/control/queuelifetime** и установить в нем срок менее семи дней (срок по умолчанию, означающий, что попытки отправки всех поступающих писем будут повторяться в течение недели). Одного или двух дней вполне хватит. Эти шаги должны снизить время, необходимое Qmail для отображения приглашения. **КК**

Реаниматор

В Мой друг владеет маленькой юридической фирмой (шесть пользователей, использующих Windows XP Pro, Linux proxy/mail сервер). Они часто портят компьютеры, и он зовет меня чинить их. Обычно это означает резервирование всех документов, сохраненных где попало, и переустановку Windows.

Поэтому я пытаюсь найти и установить полностью автоматизированную систему на базе Linux для восстановления после сбоев, которая бы производила резервирование после каждой установки (типа Ghost или G4L) и еженежно резервировала бы каждую рабочую станцию, чтобы, если кто-то навредил, нужно было только загрузиться с другого компьютера в сети или с CD и восстановить систему из образов с backup-сервера.

Кристос Иоанну [Christos Ioannou]



Web-интерфейс BackupPC делает восстановление файлов проще даже для пользователей Windows!

Здесь две отдельных проблемы. Первая — это создание резервной копии на CD или в сети, чтобы полностью переустановить систему в серьезном случае. Вторая — регулярное резервирование данных.

Вы не ошибетесь, если для первого пути выберите *Partition Image* — www.partimage.org. Это Linux-программа, имеющая клиент-серверные функции. Вы можете запустить сервер на вашем Linux-компьютере и использовать Live CD для создания образов дисков каждой Windows-машины. Вам понадобится Live-CD дистрибутив, который можно будет использовать для восстановления диска из файл-образа на сервере. RIP (Recovery Is Possible [«восстановление возможно»; также Rest In Peace — покойся с миром — надпись на надгробье; см. стр. 48 — прим. ред.]) вполне подойдет (www.tux.org/pub/people/kent-robotti/looplinox/rip). Документация содержит детальные инструкции изменения CD-образа под ваши нужды; вы можете добавить небольшой скрипт командной оболочки и вызвать его из `/etc/rc.d/rc.local`, чтобы автоматизи-

ровать полное восстановление системы при загрузке с CD.

Для ночного инкрементного резервирования хорошим выбором будет BackupPC (<http://backuppc.sourceforge.net>). Она запускается на Linux сервере и не требует специального ПО, установленного на Windows ПК, т.к. доступ к нему осуществляет Samba. Нужно только настроить на ПК совместное использование, чтобы BackupPC смог загрузить файлы. Вся работа выполняется на Linux-машине, так что ночным резервированием сможет управлять простое задание Cron. BackupPC обладает web-интерфейсом, и пользователям не придется учить загадочные команды для восстановления файлов из резервной копии.

Эта программа особенно хороша при резервировании некоторого числа схожих ПК, поскольку хранит одну копию файла, находящегося на нескольких компьютерах. Вместе со сжатием это значительно уменьшает требуемое пространство для резервирования узлов сети. **HB**

Захват экрана

Ищу приложение, которое бы записывало все, что я делаю на машине, в небольшой фильм. Можете ли вы что-то порекомендовать?

Мак-Скрафф [McScruff], с форума LXF

Для этого существует множество решений, в зависимости от предназначения фильма. Если вы хотите опубликовать его в Интернете, лучшим выбором будет *Vnc2swf*. Она записывает VNC-сессию как Flash-анимацию. Вам нужно будет установить VNC (или *Tightvnc* с www.tightvnc.com). VNC создан для удаленного рабочего стола,

но вы также можете использовать его на одном компьютере. Запустите VNC-сессию `vncserver -depth 16 -geometry 800x600` и увидите строку вида:

```
New 'X' desktop is yourhostname:N
```

Последняя часть — имя хоста (`yourhostname`) и номер дисплея. Если ваш компьютер не подключен к сети, можете использовать `localhost`. Теперь начните сессию командой

```
vnc2swf -startrecording -geometry 800x600 -depth 16 -framerate 5 demo.swf yourhostname:N.0
```

Убедитесь, что разрешение (`geometry`), глубина цвета (`depth`), имя хоста (`yourhostname`) и дисплей соответствуют только что запущенному вами VNC-серверу. `0` в конце обязателен. Появится новое окно, содержащее рабочий стол VNC-сессии, и все, что вы будете делать в нем, запишется в `demo.swf`. Завершите запись, закрыв это окно. Программа выведет некоторый HTML-код для просмотра Flash-анимации в веб-браузере, который вы при желании можете перенаправить в файл. Размер и количество кадров определяется веб-браузером, но для вывода демонстрации локально на монитор или проектор вы можете захотеть увеличить и то и другое.

Для генерации фильма можете использовать *Vncrec*. Работает она по той же схеме, что и *Vnc2swf*, но создает файл в своем собственном формате, который вы можете перекодировать в AVI или MPEG при помощи *transcode*.

```
vncrec -record demo.vnc
transcode -x vnc -use_rgb -y xvidth - k -dvd_access_delay 5 -f 10 -i demo.vnc -o demo.avi
```

Используемое разрешение опять-таки должно совпадать с разрешением, в котором запущен сервер. Ключ `-f` устанавливает количество кадров в секунду для видео. Получившийся файл может быть просмотрен в любом видеопроигрывателе, например, *MPlayer* или *Xine*.

Какое бы записывающее приложение вы ни выбрали, если вы хотите, чтобы в начале записи запускалась некоторая программа, то пропишите ее в `~/vnc/xstartup`:

```
oaimpress sample.pps
```

Другая возможность — *Istanbul* с <http://live.gnome.org/Istanbul>. Это программа для Gnome, но работает и на других рабочих столах. Она помещает иконку на панель: нажмите ее для начала записи, а повторное нажатие остановит запись. Результат сохраняется под именем `~/desktop-recording.ogg`, в формате Theora. Она может показаться примитивной по сравнению с альтернативами, но настраивается быстро и легко. **HB**

Лишенец прав!

Я недавно перешел с Xandros на Fedora Core 5. Я перекачал 3 Гб данных и обнаружил, что все файлы в моем домашнем каталоге имеют владельца и принадлежат к группе root. Есть ли скрипт, который поможет мне изменить права доступа на мое имя пользователя?

Д.Ригли [D Wrigley]

Если вы скопировали свой домашний каталог (к примеру, `/home/dave`) с одной машины на другую, наилегчайший способ восстановить владельца в этом каталоге — рекурсивно выполнить *chown* от имени root в `/home/dave` с корректными именами и группой владельца. Это можно безопасно сделать в вашем домашнем каталоге, поскольку он обычно содержит только файлы и каталоги, принадлежащие пользователю и группе пользователя, к которому относится этот каталог.

```
chown -R maccaddy:maccaddy /home/maccaddy
```

Если у вас много файлов и каталогов, принадлежащих другим пользователям и группам, необходимо выполнить поиск с заменой. Так что если пользователь 'dave' >>>

КРАТКАЯ СПРАВКА: КОМПИЛЯЦИЯ ПРИЛОЖЕНИЙ

Большинство дистрибутивов имеют большие репозитории программ, содержащие готовые пакеты практически на все случаи жизни, но иногда необходимо собрать пакет из исходных текстов. Обычная причина — отсутствие обновленного (или полное отсутствие) пакета для вашего дистрибутива, или необходимость исправить исходный код, добавив функцию или убрал ошибку. Поддерживать систему в актуальном состоянии помогает менеджер пакетов вашего дистрибутива, но если вам необходимо собрать пакет из исходных текстов, то процесс милосердно прост.

Во-первых, распакуйте архив с исходными текстами (tarball) одной из двух команд: `tar xvzf foo-1.2.3.tar.gz` или `tar xvjf foo-1.2.3.tar.bz2`

Поздние версии tar могут определить чем был создан архив (**gzip** или **bzip2**), так что аргументы `z` или `j` можно опустить. Исходные тексты обычно распаковываются в папку с именем, совпадающим с названием архива; после выполнения команд введите `cd foo-1.2.3`. Найдите файлы с именами **README** или **INSTALL** и прочтите их — обычно там находится инструкция по установке. Стандартная процедура такова:

```
./configure
make
su -c "make install"
```

Первая команда проверяет вашу систему, убеждаясь в соблюдении всех зависимостей и установке оптимальных параметров для программы. Неплохо сначала выполнить команду `./configure --help`, чтобы увидеть имеющиеся опции.

Вторая команда компилирует приложение, помещая созданные файлы в текущий каталог. Для конфигурирования и компиляции права root не нужны, но третья стадия — это копирование скомпилированных

файлов в системные подкаталоги, а потому требует прав root. Вот почему мы используем `su` для выполнения только этой команды от имени root. Пользователи Ubuntu должны использовать вместо нее команду `sudo make install`

Если вы не указывали для `./configure` путь, то по умолчанию скомпилированная программа устанавливается в `/usr/local/bin`. Если вы работаете в RPM-системе, а `configure` сообщает, что не найдена библиотека, хотя вы точно знаете, что она есть — установите соответствующий библиотеке пакет `-devel`, т.е. `libbar-devel`.

ЧАСТО ЗАДАВАЕМЫЕ ВОПРОСЫ О WPA

FAQ Что такое WPA?
WPA – потомок WEP с улучшенной безопасностью.

FAQ Спасибо, конечно, но что это?
WPA (Wi-Fi Protected Access – Защищенный Wi-Fi доступ) – система обеспечения защиты передачи данных в беспроводных сетях.

FAQ А оно мне нужно?
Почти наверняка – да. Беспроводные сети передают ваши данные на сотни метров во всех направлениях. Это означает, что кто угодно с ноутбуком и беспроводной картой может прочесть ваши данные так же легко, как при подключении к вашему сетевому хабу.

FAQ Мой ноутбук даже не дотягивает до всех комнат моего дома, так что я в безопасности, разве нет?
Отнюдь! Хотя доступ вашего ноутбука ко внешним устройствам ограничен неким расстоянием, у другого оборудования это расстояние может быть куда больше. Добавив к вашей беспроводной карте антенну, вы охватите значительно большую площадь. Соответственно улучшенные системы могут перекрывать огромные области. Вам нечего бояться, если вы используете Wi-Fi только дома для выхода в сеть с вашего ноутбука,

но если ваша фирма использует слабозащищенные (или не защищенные) беспроводные сети, то ваши конфиденциальные файлы могут читать прямо из машины на офисной парковке.

FAQ Раз уж вы упомянули: что такое WEP?
WEP – Wired Equivalent Privacy (Эквивалент проводной безопасности), предыдущая попытка защитить беспроводные сети. WEP лучше, чем ничего, но взломать его нетрудно. Он может не допустить вашего соседа к вашему компьютеру, если тот случайно подключится к вашей точке доступа вместо своей (так я и узнал, что мой сосед использует небезопасную сеть), но не сможет остановить взломщика.

FAQ Какое оборудование использует WPA?
Все новые беспроводные сетевые карты и точки доступа поддерживают как WPA, так и WEP. Если у вас старое устройство, может оказаться, что оно поддерживает только WEP.

FAQ А как насчет программ?
Вам необходима двухуровневая поддержка WPA на уровне драйвера устройства (большинство драйверов ее уже имеют) и программа, реализующая WPA. Наиболее популярна

wpa_supplicant, она должна поставляться с вашим дистрибутивом и доступна по адресу http://hostap.epitest.fi/wpa_supplicant. Это программа, управляющая согласованием ключей и аутентификацией.

FAQ Я полагаю, чтобы использовать WPA в Linux, я должен прочесть man-страницы и HOWTO, а также отредактировать файлы конфигурации?
Вы, конечно, можете сделать это, если

ладите с Vim. Однако существуют графические инструменты настройки, облегчающие жизнь. Самый продвинутый из них – вероятно, *NetworkManager* из Gnome (www.gnome.org/projects/NetworkManager). Он включен в последние дистрибутивы, например, Fedora Core 5 и SUSE 10.1. Пользователи KDE могут предпочесть *KWlan* (www.kde-apps.org/content/show.php?content=37041) (или *KNetworkManager*: <http://en.opensuse.org/Projects/KNetworkManager>, – прим. ред.).



Новый *NetworkManager* от Gnome (чтобы им пользоваться, Gnome запускать не нужно!) очень облегчает настройку зашифрованного соединения с сетью.

имеет множество файлов и каталогов в `/var/www/html` и вы хотите сменить владельца этих файлов на пользователя и группу 'bigmac', следует применить `chown`

`-R` для смены владельца каталогов. Беда в том, что заодно меняют владельца и те файлы, которые вам трогать не хотелось. Чтобы каталоги, не принадлежащие Dave,

остались без изменений, для поиска и смены владельца используйте команду `find`:

```
find /var/www/html -user dave -group dave -exec chown bigmac: bigmac {} \;
```

Эта команда найдет любой каталог или файл в `/var/www/html`, принадлежащий пользователю и группе dave, а затем сменит владельца на bigmac. {} указано для замены найденных файлов, соответствующих критериям `-user` и `-group`, а \; необходимо во избежание интерпретации; командной оболочкой, и чтобы сообщить `find`, что список параметров закончился.

Итак, для стандартных домашних каталогов наилегчайший способ сменить владельца одновременно – использовать команду `chown -R`. Помните, что этот метод применим не во всех частях файловой системы! **KK**

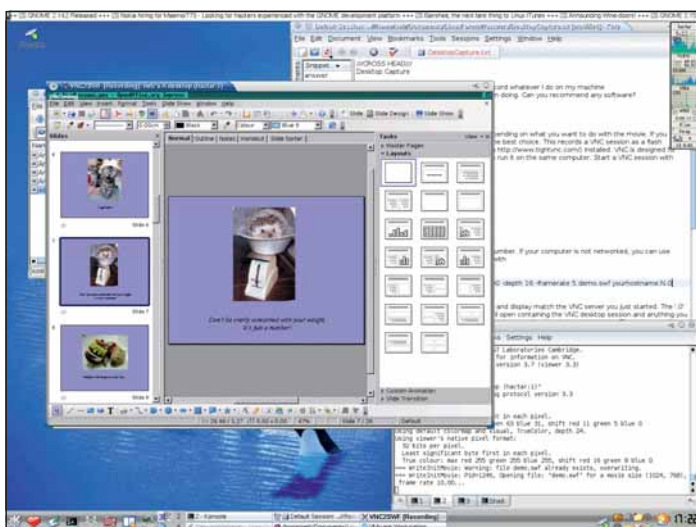
Пропал Thunderbird

Щелкая на email-адресе в KDE, я получаю ошибку: `'Kdeinit can not start /usr/share/application/Thunderbird/Thunderbird'`.

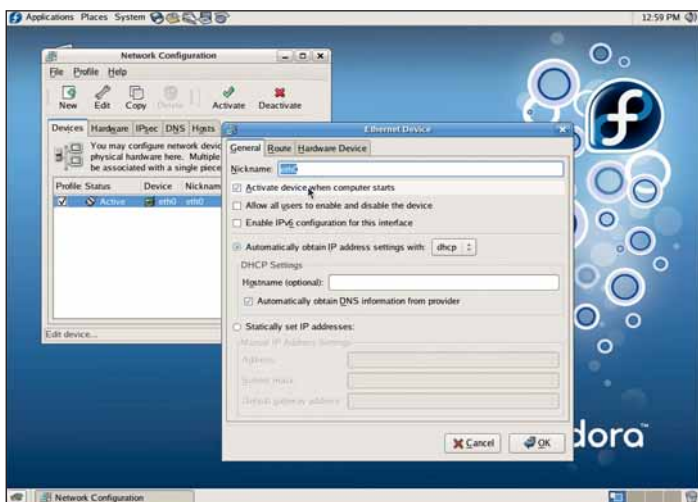
Thunderbird установлен в `/opt/Thunderbird`. Был у меня SUSE, но в сейчас я использую Gentoo, и когда я переносил каталог `/home`, вероятно, что-то затер в *Kdeinit*, но не могу понять что. Не могли бы вы сказать мне, как сделать, чтобы *Kdeinit* искал в нужном месте?

Стьюарт [Stewart]

Похоже, что KDE не там ищет *Thunderbird*. Как и большинство опций KDE, вы можете изменить это в Центре управления KDE; хотя найти нужное место в Центре Управления трудно – там столько всего, и не всегда там, где вы ожидаете увидеть. Центр Управления содержит функцию поиска, которая обычно помогает, но не в этом случае (по крайней мере не в KDE 3.5.3). Нужная вам опция находится в Компоненты > Выбор Компонентов > Email-клиент. Выберите радиокнопку Использовать другой Email-клиент, затем щелкните на маленькой иконке справа от строки ввода чтобы открыть окно выбора приложения. Выбрав программу >>>



Vnc2swf и *Vncrec* позволяют записывать содержимое VNC-сессии.



Если две сетевые карты смущают вашу систему, отключите одну из них!

таким образом, вы установите верный путь. *Thunderbird* откроется, но без адреса получателя или любой другой информации. Чтобы исправить это, добавьте следующее в строку запуска *Thunderbird*:

```
-compose "mailto:%?subject=%s& body=%B"
Задержите курсор мыши над строкой ввода, чтобы увидеть доступные опции. НБ
```

Спутанные карты

Вот только что установил Fedora Core 5, и слегка в недоумении: какая из моих сетевых карт работает. У меня их две, и при предыдущих установках по умолчанию использовалась eth0. Вот результат выполнения *ifconfig*:

```
eth0 Link encap:Ethernet HWaddr 00:30:18:58:4A:A3
inet addr:192.168.1.101 Bcast:192.168.1.255
Mask:255.255.255.0
UP BROADCAST MULTICAST MTU:1500
Metric:1
RX packets:0 errors:0 dropped:0 overruns:0
frame:0
TX packets:0 errors:0 dropped:0 overruns:0
carrier:0
eth1 Link encap:Ethernet HWaddr 00:50:BA:B3:B1:A5
inet addr:192.168.1.152 Bcast:192.168.1.255
Mask:255.255.255.0
inet6 addr: fe80::250:baff:feb3: b1a5/64
Scope:Link
UP BROADCAST RUNNING MULTICAST
MTU:1500 Metric:1
RX packets:258479 errors:0 dropped:0
overruns:0 frame:0
TX packets:264885 errors:0
dropped:0 overruns:13
carrier:0
```

Мне кажется, что при работе в сети весь трафик идет через eth1. Можете ли вы пролить свет на это?

pk fox

Вы используете DHCP на обеих сетевых картах? Если да, то вот что скорее всего происходит:

- 1) Обнаруживается первая сетевая карта (NIC) и загружается модуль.
- 2) Поднимается её интерфейс и используется DHCP для назначения IP адреса, а также настройки DNS и маршрутизации.
- 3) Определяется вторая NIC и загружается модуль.
- 4) Поднимается её интерфейс и используется DHCP для назначения IP адреса, а также настройки DNS и маршрутизации.

Четвертый шаг переписывает шлюз по умолчанию в таблице маршрутизации, забывая установки, сделанные на втором шаге. Вы можете проверить это, выполнив

```
route -n
```

Строка, показывающая расположение 0.0.0.0, завершается интерфейсом, используемым по умолчанию. Есть ли причина, по которой вы используете две NIC? Если да, то вам, видимо, придется настроить шлюз по умолчанию вручную. Или же отключите опцию *activate device when computer starts* для одной из NIC в программе настройки сети. **НБ**

Сам себе браузер?

В Я управляю сервисом хостинга, на нем более 100 доменов. Последние несколько дней стало казаться, что наш сервер тормозит. Я выполнил несколько предварительных тестов (используя *netstat*) и обнаружил множество соединений с моего сервера по TCP-порту 80 к моему серверу по временным портам. Из полученных результатов я понял, что

имеются соединения от *Apache* по порту 80 к другим портам моего сервера. Но почему? Как мой сервер может просматривать мои собственные web-сайты? Используется *Apache 2* под Red Hat Enterprise Linux 3.

Джон [John]

Очень похоже, что это поведение спровоцировал один из ваших новых web-сайтов. Из описанного вами отчета *netstat* я делаю вывод, что некий код вызывает соединения с вашим сервером. Проанализировав деятельность сервера *Apache*, мы можем сравнить число *netstat*-соединений с обслуживаемыми *Apache*. Для этого поместим следующее в конфигурационный файл *Apache* (*/etc/httpd/conf/httpd.conf*):

```
ExtendedStatus on
<Location /server-status>
SetHandler server-status
</Location>
```

Если вы откроете страницу www.domain.com/server-status?refresh=5, то статус вашего сервера будет обновляться раз в пять секунд. Уделите особое внимание загруженности процессора и числу секунд с начала нового запроса (SS). Также, сравнивая число соединений выдаваемых *netstat* с числом соединений на отдельном виртуальном хосте, вы быстро найдете виновника! **КК**

Саботаж Sendmail

В Я создаю сервер для размещения управляющего ПО нашей лаборатории (я написал его на PHP, благодаря урокам в LXF!). Установил сервер Kubuntu с *Sendmail* – и начались проблемы.

Я выполнил простую настройку *Sendmail*, оставив все как есть, поскольку почти ничего не понимаю в настройке *Sendmail*! Он отправляет почту людям в локальной сети (например, john@localnet.co.uk), но во внешний мир ничего не исходит (например, на john@hotmail.com). Был бы очень благодарен за совет – а может, и за учебник по установке почтового сервера?

Джон Клейтон [John Clayton]

О *Sendmail* – не лучший выбор для вас. Это несомненно мощный почтовый сервер, но, увы, трудно настраиваемый. *Postfix* или *Exim* больше подойдут вам, они оба доступны через репозиторий Ubuntu (*Postfix* – стандартный почтовый сервер и находится на установочных дисках Ubuntu). Эти сервера отлично документированы, файлы настроек имеют обычный текстовый формат, и изучить их настройки проще, чем продираяться сквозь множество опций настройки *Sendmail*.

Какой бы сервер вы ни выбрали, вам следует настраивать его через *Webmin*, он предоставляет опции в дружелюбной гра-

фической оболочке и позволяет избежать ошибок конфигурирования сервера, приводящих к потере почты или снижению безопасности. При желании вы также сможете читать или подстраивать файлы настроек вручную – *Webmin* помогает изучать опции настройки, а не прячет их.

Для любого выбранного сервера, файлы журналов (log-файлы) предоставят причину сбоя. Выполните

```
tail -f /path/to/logfile
```

и попытайтесь отправить почту во внешний мир. Вы увидите сообщение об ошибке, соответствующее сбою: варианты – от сбоя DNS (хотя это маловероятно, т.к. Интернет вообще-то работает) до блокирования исходящего SMTP-трафика вашим провайдером. Многие провайдеры делают это в качестве противоспамной меры: или перенаправляют весь SMTP-трафик на собственный почтовый сервер, или блокируют вообще. Если это так, вам необходимо настроить ваш почтовый сервер на использование сервера вашего провайдера как 'smarthost', тогда вся почта не для вашей локальной сети будет отсылаться через тот сервер. Для этого в *Sendmail* поместите в *Sendmail.cf* следующее:

```
DSmail.isp.com
заместив mail.isp.com SMTP-сервером провайдера. Для Postfix такая строка relayhost = mail.isp.com
```

Если вы используете *Webmin*, это первая опция в модуле *Sendmail* и четвертая в модуле *Postfix*. **НБ**

Damn-ГРУБиян

В На моем ПК много разделов, и установлено несколько дистрибутивов Linux, хочу поиграть с ними, прежде чем остановиться на одном. Мой загрузчик – *XOSL*, и со многими дистрибутивами он работает прекрасно, включая штуку из Редмонда! Но он серьезно слетел после дистрибутива Damn Small Linux с *LXF80*. Скрипт установки DSL на жесткий диск не предоставляет выбора места размещения загрузчика *Lilo* или *Grub* – он всегда пишет его прямо в главную загрузочную запись (MBR), туда же, где размещается *XOSL*! Поэтому, когда я восстанавливаю *XOSL*, он находит все остальные дистрибутивы, но не DSL. Или ПК загружается только в DSL. Они не работают вместе.

Не могли бы вы объяснить на уровне начинающего: как настроить загрузку (*Lilo* или *Grub*) на разделе, где установлен DSL, чтобы *XOSL* смог найти и запустить его?

Джефф [Jeff]

О Установить *Grub* на раздел вместо MBR просто, и позор, что DSL не предоставляет такой возможности. Предположим для примера, что DSL установлен на */dev/hda5*. Загрузитесь в DSL, откройте root-терми-



★ ВОПРОС-ПОБЕДИТЕЛЬ!

Счастливец и победитель этого номера по мнению английской редакции – Мэтью.

Эволюция для почты

В Пытаюсь исправить проблемы с аутентификацией на моем почтовом сервере, и единственный обнаруженный способ – его настройка в Evolution. Есть ли другие пути, обходящиеся без создания записи в Evolution?

Мэтью Саид [Matthew Saeed]

О Один из лучших способов протестировать множество различных сервисов, включая SMTP SAUTH – использовать Telnet. Конечно, я не стал бы рекомендовать Telnet для обычного входа, но в качестве теста сервисов он не оценим. Для исправления вашей проблемы надо соединиться с почтовым сервером по порту 25 и зарегистрироваться, используя кодировку BASE64 (читайте об этом на <http://en.wikipedia.org/wiki/>

Base64). Для начала расшифровка нескольких полезных строк, закодированных при помощи www.dillfrog.com/tools/base-64_encode:

- 'VXNlcm5hbWU6' расшифровывается как 'Username:'
- 'UGFzc3dvcmQ6' расшифровывается как 'Password:'
- 'dGVzdF9seGZAcmV6ZC5jby51aWw=' расшифровывается как 'test_lxf@rezd.co.uk'
- 'Zm9vYmFy' расшифровывается как 'foobar'

Следующие строки являются проверкой регистрации на сервере. Мы используем BASE64 кодирование для нескольких строк описанных выше.

Первое, соединимся при помощи Telnet к домену/IP адресу почтового сервера (т.е. mail.rezd.co.uk или 10.0.0.1) по порту 25:

```
telnet 10.0.0.1 25
```

Сервер отвечает SMTP-приглашением (баннером):

```
Trying 10.0.0.1...
Connected to mail.rezd.co.uk (10.0.0.1).
Escape character is '^'.
220 mail.rezd.co.uk ESMTP
Даем команду EHLO:
EHLO other.domain.rezd.org.uk
Далее сервер сообщает, что именно он поддерживает (в этом почтовые сервера могут различаться).
250-mail.rezd.co.uk Hello other.domain.rezd.org.uk [192.168.0.1], pleased to meet you
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-8BITMIME
250-AUTH DIGEST-MD5 CRAM-MD5 LOGIN PLAIN
250 HELP
```

Регистрируемся на почтовом сервере:

```
AUTH LOGIN
```

Он выдает запрос имени пользователя (username):

```
334 VXNlcm5hbWU6
```

Теперь введем имя пользователя, под которым хотим зарегистрироваться, например, test_lxf@rezd.co.uk:

```
dGVzdF9seGZAcmV6ZC5jby20=
```

Затем запрашивается пароль:

```
334 UGFzc3dvcmQ6
```

Мы отвечаем:

```
Zm9vYmFy
```

И, наконец, он говорит «да» (yes), то есть регистрация (authentication) работает:

```
235 2.0.0 OK Authenticated
```

Если мы получаем следующее, то понимаем, что с регистрацией есть проблемы:

```
535 5.7.0 authentication failed
```

Этого достаточно для тестирования регистрации; если мы хотим протестировать отправку писем, то мы должны продолжить SMTP-диалог. **КК**

на и выполните **Grub**. Вы перенесетесь в командную оболочку **Grub**, где следует набрать

```
root (hd0,4)
setup (hd0,4)
quit
```

Grub начинает счет с нуля, поэтому первый диск, пятый раздел (**hda5** в терминологии Linux) – это **hd0,4**. Теперь загрузчик **DSL** установлен в разделе, и вы можете велеть **XOSL** загружаться с этого раздела. Когда **XOSL** загрузит **DSL**, вы перенесетесь в меню **Grub** – возможно, и зря, поскольку вы уже выбрали, какую ОС загружать. Избавиться от этого можно, отредактировав **/boot/grub/menu.lst** и изменив время ожидания с 15 на 0. Если вы хотите сохранить выбор опций **DSL**,

представленных в меню **Grub**, измените время на меньшее, но ненулевое значение. **НБ**

Медленно и узко

В Пытаюсь найти широкополосного провайдера с разумными ценами, имеющего дело с Linux. Поискал высокоскоростное телефонное соединение под Linux, но не достиг успеха. Поэтому я исследовал статистику загрузки под различными дистрибутивами.

Fedora Core 5 оказалась в самом низу с пиком в 1,8 КБ/с и средним около 0,7. Fedora 4 и SUSE достигли максимума 3 КБ/с при средней скорости загрузки 1,5 КБ/с. Knoppix 4 не многим лучше. В Xandros 3 получилось 4 КБ при среднем около 2 КБ.

Лучше всех Mandriva 10.1 (использовались *Mozilla* и *Epiphany*), с пиком около 13 КБ/с и средним около 6 КБ/с.

Данные получены для различных соединений в различное время, но результаты вполне согласуются – они все работают плохо около 19:00 и 10:00 и все показали лучший результат в воскресенье утром. Я использую внешний COM-модем 56K. Есть идеи, как получить двузначную цифру скорости?

Крис Марсден [Chris Marsden]

PS А также, идеи о сети из двух Linux-компьютеров с двумя разными дистрибутивами.

О В Великобритании есть два провайдера, специализирующихся на пользователях Linux: UKLinux.net и UK Free Software Network (www.ukfsn.org). Оба предоставляют как ADSL, так и телефонное соединение. Ваши проблемы со скоростью действительно выглядят слегка странными, но сделать выводы трудно, поскольку вы сообщили очень мало информации – даже не указали марку модема.

Было бы интересно сравнить настройки модема в каждом из дистрибутивов. Использование браузера для измерения скорости закачки – не самый надежный тест, на него влияет множество факторов, включая прокси-сервер провайдера. Лучшим тестом будет попытка загрузить файл через *wget*. Попробуйте в каждом дистрибутиве команду:

```
wget ftp://ftp.mirror-service.org/sites/ftp.kde.org/pub/kde/stable/3.5.2/src/kdeaddons-3.5.2.tar.bz2
```

Любой файл с FTP-сервера в Великобритании сойдет для теста. Вы не сможете получить двузначную цифру с 56K-модемом, если только не загружаете сжимаемые данные, например, группы новостей или web-страницы (но не картинки). Лучшее, на что вы можете рассчитывать со сжатыми данными вроде указанного выше файла или картинки – это около 7 КБ/с. Сжатые файлы дают истинную оценку качества связи.

Указанные вами времена интересны; 19-00 – это пик использования Интернета в Великобритании (сами решайте, web или *Emmerdale*), тогда как утром в воскресенье отмечается спад. Стоит также попросить ВТ протестировать вашу линию. Даже если они скажут, что все в порядке, сам факт тестирования часто способствует улучшению.

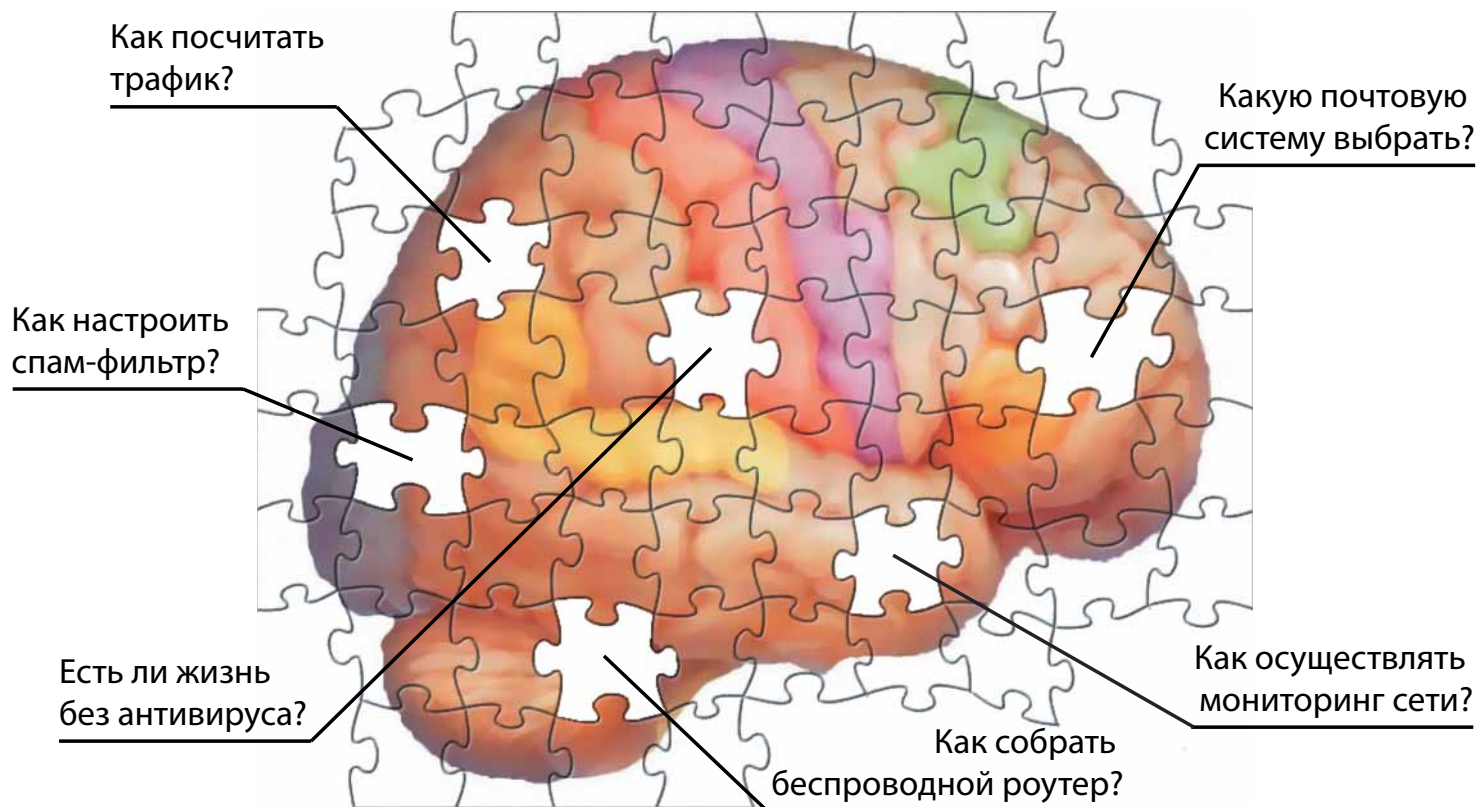
При использовании провайдера с поддержкой Linux вы вправе получить такую поддержку. Я бы предложил использовать телефонный доступ через UKFSN (это окупается) и попросить обоих провайдеров помочь со скоростью соединения. Кто окажется полезнее, того и делайте вашим широкополосным партнером.

Что до вашего вопроса о соединении в сеть двух компьютеров с разными дистрибутивами, то это все равно, что два компьютера с одним и тем же вариантом Linux. Хотя инструменты настройки могут различаться, большинство дистрибутивов в принципе схожи. NFS, HTTP, *Samba*, практически все, что вы хотите использовать для организации сети, работает одинаково во всех дистрибутивах. **НБ**

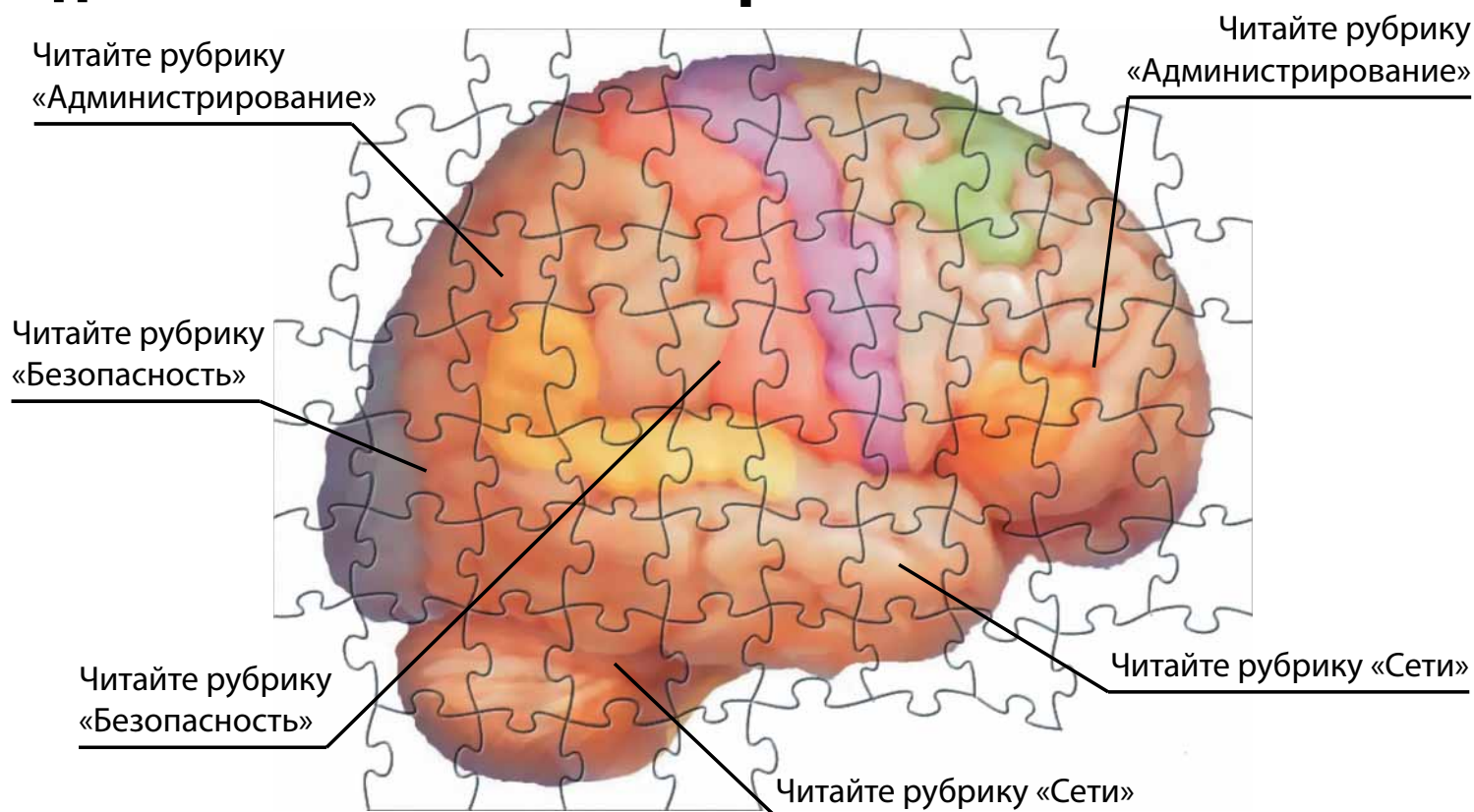


XOSL будет только счастлив если каждый дистрибутив будет хранить свой загрузчик на своем разделе – и значит **DSL** нужно слегка подправить.

Какими мыслями занят системный администратор, который не читает свой журнал?



Остальные системные администраторы знают, где искать ответы на вопросы!



Журнал «Системный администратор»
www.samag.ru

Диск Linux Format

Ubuntu, Mandriva и тонны ПО на одном DVD!

ПАТЕНТОВАННОЕ БЕЗУМИЕ



Майк Сондерс — машина для кодирования на макаронном топливе.

Как я уже говорил, Брюс Пиренс [Bruce Perens] только что поместил сообщение на Techprocrat со скверными новостями: держатели патентов на ПО начинают атаковать разработчиков открытых проектов. Он использовал два примера, иллюстрирующие, на что замахнулась эта угроза: на Hibernite, ныне в собственности Red Hat, купившей JBoss за 350 миллионов долларов, и на JMRI —

«ДАЖЕ ОДНА УГРОЗА СУДА СПОСОБНА ПОВРЕДИТЬ ПРОГРЕССУ.»

Java-утилиты, контролирующую действия модели железной дороги. Один проект радует людей с железнодорожным хобби, а второй — серьезный игрок на рынке корпоративных приложений, но оба используют идеологию открытого кода, и оба вовлечены в судебные разбирательства.

Сообщество Linux уже давно рисовало апокалиптическую картину патентования ПО и его потенциальных последствий. Поскольку большая часть открытого кода написана добротами-хакерами, у которых нет денег на оплату юристов, даже одна угроза подобной разборки может испортить все дело. Конечно, на нашей стороне есть и большие фирмы вроде IBM, и они очень хорошие участники сообщества, но если проекты с открытым кодом втянут их в рулетку судебных процессов, крупный бизнес может призадуматься: а стоит ли ввязываться?

Если сбудутся худшие предсказания, Linux переместится в подполье. Мы сможем скачивать операционную систему по Интернету, но никто не захочет поддерживать ее или вкладываться в ее разработку. Конечно, это наихудший возможный сценарий, но, увы, реальный, и все мы должны это понимать. Что ж, извините за негатив — наслаждайтесь нашим DVD!

mike.saunders@futurenet.co.uk

ДИСТРИБУТИВЫ UBUNTU 6.06

Процесс выпуска Ubuntu Dapper Drake занял на шесть недель больше, чем планировалось (задержка была вызвана необходимостью доработок по части надежности системы). Ubuntu за последние два года стал чрезвычайно популярен, претерпев метаморфозу от любопытного потомка Debian до настольной ОС, полюбившейся многим пользователям, а релиз 6.06 означает особенный этап. По мнению разработчиков, это первый релиз Ubuntu, готовый к широкому использованию на корпоративных рабочих станциях и серверах, а также первый дистрибутив с пятилетним циклом поддержки обновлений безопасности для серверного варианта (для настольного — три года).

Рабочий стол в Ubuntu — Gnome 2.14.1, но поклонники KDE или Xfce могут выбрать вариации на базе Ubuntu: Kubuntu

и Xubuntu. Благодаря тому, что дистрибутив основан на Debian, вы можете устанавливать тысячи дополнительных программ из онлайн-репозитариев.

Так-то так, но мы тут в Башне LXF ищем легких путей и не стали просто сливать стандартный Ubuntu на наш DVD. Мы постарались свести Ubuntu, Kubuntu и Xubuntu в один дистрибутив и добавить дополнительное ПО, не включенное в поставку по умолчанию (например, Mono, Scribus и Emacs).

Три вариации

Не будем вдаваться в подробности процесса установки (см. стр. 32) но у вас теперь есть три разновидности Ubuntu — и много больше. По умолчанию используется Gnome, но вы можете переключиться в KDE или Xfce, отключившись от системы и выбрав другой

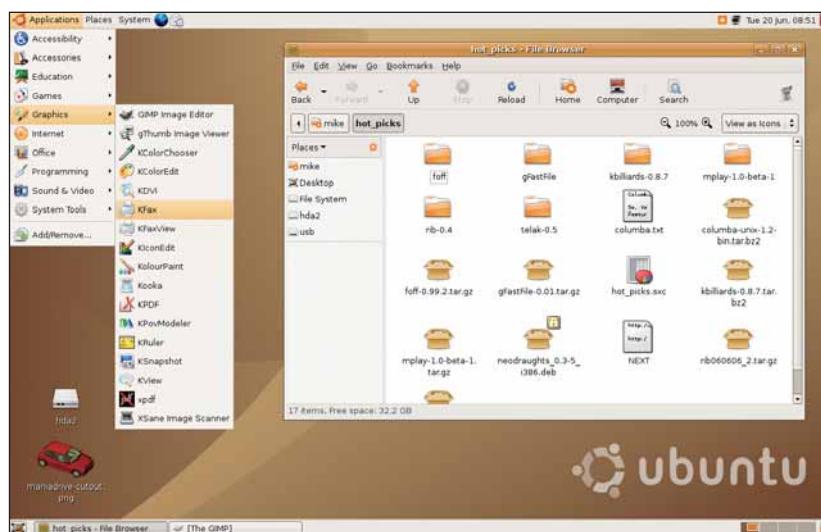


Xubuntu использует быстрый Xfce вместо Gnome — превосходно для старых ПК.

тип сессии на экране входа в систему.

Запустите Live-версию Ubuntu, загрузив ваш ПК с DVD, и если она вам понравится (а мы в этом уверены), установите Ubuntu на жесткий диск, два раза кликнув по кнопке Install на рабочем столе. Чтобы инсталлятор работал с хорошей скоростью, рекомендуется

ПК с 512 МБ ОЗУ, но на случай проблем мы добавили на диск дополнительную опцию: альтернативный CD Ubuntu. Его установка выполняется в текстовом режиме, как это было в предыдущих релизах Ubuntu, и памяти требуется гораздо меньше (128 МБ вполне хватит). Если загружаемая с DVD Live-версия почему-либо не заработает, запишите образ из каталога **Distros/Ubuntu** на CD-R и загрузитесь с него.



Ubuntu: одним он кажется мрачно-бурым, другим — глянцево-рыжим.



НЕТ DVD-ПРИВОДА?

Вы можете сделать CD-ROM-версии дистрибутивов, размещенных на нашем диске. Для Ubuntu мы включили альтернативный ISO-образ 6.06, который загружается прямо в текстовый инсталлятор. Вы найдете этот ISO-файл в каталоге **Distros/Ubuntu** —

запишите его на CD-R и загрузитесь.

Для Mandriva One мы включили систему Jigdo, позволяющую конвертировать DVD в набор CD-образов. Откройте index.html в корневом каталоге диска и прочтите, как запускать Jigdo для изготовления CD-версии. В большинстве случаев достаточно будет одной команды, но

стоит также посмотреть и на дополнительные опции.

Есть и еще пара приятных вещей для машин без DVD-привода: RIP полезен для исправления неисправностей, да еще мы сумели втиснуть Damn Small Linux 3.0. Читайте 115 страницы.

Информация о диске

Внимательно прочтите это перед тем, как использовать DVD-диск.

ЧТО-ТО ПОТЕРЯЛИ?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials» (Главное) на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, следует заглянуть именно туда.

ФОРМАТЫ ПАКЕТОВ

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любые другие. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными бинарными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы смогли собрать его самостоятельно.

ДОКУМЕНТАЦИЯ

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

ЧТО ЭТО ЗА ФАЙЛЫ?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux, различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

имя_программы-1.0.1.i386.rpm – вероятно, это бинарный пакет RPM, предназначенный для работы на системах x86;
имя_программы-1.0.1.i386.deb – такой же пакет, но уже для Debian;
имя_программы-1.0.1.tar.gz – обычно это исходный код;
имя_программы-1.0.1.tar.bz2 – тот же файл, что и выше по списку: «tgz» – это сокращение от «tar.gz»;
имя_программы-1.0.1.tar.bz2 – тот же файл, но сжатый bzip2 вместо обычного gzip;
имя_программы-1.0.1.src.rpm – также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;
имя_программы-1.0.1.i386.fc4.rpm – бинарный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;
имя_программы-1.0.1.ppc.suse9.rpm – бинарный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;
имя_программы-devel-1.0.1.i386.rpm – версия для разработчиков.

Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте:

disks@linuxformat.ru

ЛИНУКС ФОРМАТ В ГИГАБАЙТЕ DVD

2 DISTROS

ЛИНУКС
ФОРМАТ

Икскаре 0.44 – великолепный векторный редактор с улучшенным экспортом в PDF
 RIP – скорая помощь на CD на базе Slackware Linux
 Компиляторы – последние достижения мира Open Source
 Документация – свежий срез Linux Documentation Project
 Slam Soccer – переиграйте Чемпионат Мира на свой лад!

Также на диске



Специальная мегапак-редакция
 самого популярного дистрибутива!

UBUNTU 6.06

- Включает Ubuntu, Kubuntu и Xubuntu
- Тысячи пакетов в Сети
- Стабильность Debian плюс новейшие приложения
- Запускается как LiveDVD с возможностью установки



Журнал
3D games - Файлы из учебника Ogre
Inkscape - Анимация из учебника Inkscape
Oo Basic - Дополнительный код из учебника
XDMCP - Видео из учебника

Рабочий стол
Basket - Информационный менеджер KDE
ClamTk - Интерфейс к антивирусу
Grsync - Интерфейс к Rsync
HomeBank - Персональный финансовый менеджер
TnyuFSK - Редактор и файловый менеджер
Xconfig - Помощник в настройке X

Разработка
Binutils - Утилиты разработчика
Compilers - Компиляторы с открытым кодом
Protoeditor - Текстовый редактор KDE
Ruby - Язык программирования
Ruby on Rails - Среда web-разработки
Rubygems - Менеджер пакетов Ruby

Дистрибутивы
Damn Small Linux - Ультра-лёгкий настольный дистрибутив
Mandriva One - Live-версия Mandriva
RIP - Дистрибутив для восстановления системы
Ubuntu - Супер-дружелюбный дистрибутив

Игры
Balazar Brothers - Трёхмерные платформы
Nexiz - Трёхмерная многопользовательская стрелялка
Pyamb - Головоломка
RedShift - Авиасимулятор
Slam Soccer - Футбол

Графика
GEGE - Библиотека обработки изображений
Inkscape - Векторный графический редактор
Open Clip Art - Свободная галерея векторных изображений
Rawstudio - Конвертер raw-графики

Помощь
LDP - Руководства из Linux Doc Project
HotPicks
DOSBox - Эмулятор DOS и ПК
Foff - Клиент FTP с GTK-интерфейсом
Gfast File - Файловый менеджер
KBilliards - Бильярд
Lighttpd - Web-сервер
MPlay - Медиа-проигрыватель
NeoDaemons - Настольная игра
RiB - Пейджер рабочего стола
Telak - Просмотр изображений

Интернет
Flail - Почтовый клиент
Kopete - Обмен мгновенными сообщениями для KDE
Liferea - RSS-ридер
Willow - Фильмующий прокси-сервер

Офис
AbiWord - Текстовый процессор
Morla - Редактор PDF-документов
Textmax - Редактор WYSIWYG

Сервер
Cloudwiki - Wiki-система
FinScribe - Web-аудиоинг
Rep - Балансирующий нагрузки
Sshproxy - SSH-прокси для Python
Wefis - Файловый сервер с web-интерфейсом

Звук
Mesa - Музыкальный проигрыватель
MMA - Генератор музыкального аккомпанемента
Serpentine - Запись музыкальных CD-дисков
Система
AppArmor - Система безопасности
Gambad - Браузер Samba
RAM-USB - RAM-модуль для USB
Video drivers - Драйверы ATI и Nvidia

Комментарий? Присылайте ваши мысли и предложения по электронной почте: info@linuxformat.ru
Пожалуйста, ознакомьтесь с опубликованной в журнале инструкцией перед использованием данного диска.

Настоящий диск тщательно тестировался и проверялся на всех стадиях производства, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать антивирусный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. К сожалению, редакция Linux Format не может принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые могут повлечь за собой использование этого DVD, представленных на нем программ или данных. Перед тем, как устанавливать какое-либо ПО на компьютер, подключенный к сети, проконсультируйтесь с сетевым администратором.

Дефектные диски. В маловероятном случае обнаружения дефектов на данном диске, пожалуйста, обращайтесь по адресу: disks@linuxformat.ru

Тираж изготовлен ООО «Иармедиа», Россия, Санкт-Петербург, 196006 ул. Цетковича д. 7, тел. +7 (812) 388-8290. Лицензия ИИПР России ВАР № 77-68

Поставляется вместе с журналом LINUXFORMAT номер 9(83) Сентябрь 2006



СОЗДАНИЕ УСТАНОВОЧНЫХ ДИСКОВ ПРИ ПОМОЩИ CDRECORD

Самый быстрый способ записать ISO-образ на чистую матрицу – это *cdrecord*. Для всех перечисленных ниже действий потребуются права root. Для начала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке, например, 0,3,0. Теперь вы можете с легкостью записать образ на диск:

```
cdrecord dev=0,3,0 -v /путь к образу/image.iso
```

Чтобы упростить дальнейшее использование *cdrecord*, сохраните некоторые настройки в файле `/etc/default/cdrecord`. Добавьте по одной строке для каждого устройства записи (вероятно, в вашей системе присутствует всего одно такое устройство):

```
Plextor=0,3,0 12 16M
```

Первое слово в этой строке – это метка, затем, после адреса SCSI-устройства вы должны указать скорость и размер буфера. Теперь вы можете заменить SCSI-адрес в командной строке на выбранную вами метку. Все будет еще проще, если вы добавите следующее:

```
CDR_DEVICE=Plextor
```

Все, что вам теперь нужно для записи ISO-образа – это набрать команду

```
cdrecord -v /path/to/image.iso
```

Если вы не из числа любителей командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее из-под root, выберите вкладку «Burn» и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажимайте на «Combust!». Пока ваш образ пишется на диск, можете выпить чашечку кофе.

Другая ОС?

Вам не обязательно использовать Linux для записи компакт-диска. Все необходимые файлы уже включены в ISO-образ. Программы вроде *cdrecord* просто переносят данные на чистую матрицу. Если у вас нет устройства для записи дисков, можно найти того, у кого оно есть, и записать диск на его компьютере. На нем может стоять Windows, Mac OS X, AmigaOS, или любая другая ОС.

Нет устройства для записи дисков?

А что если у вас нет устройства, с помощью которого можно было записать образ на диск? Вы знаете кого-либо с таким устройством? Вам не придется использовать Linux для записи дисков, подойдет любая операционная система, способная распознать привод записи дисков (см. выше).

Некоторые дистрибутивы умеют монтировать образы дисков и выполнять сетевую установку или даже установку с раздела жесткого диска. Конкретные методы, конечно, зависят от дистрибутива. За дополнительной информацией обращайтесь на web-сайт его разработчика.

LXF

СОДЕРЖИМОЕ DVD-ДИСКА

Журнал

3D games	Файлы из учебника Ogre
Inkscape	Анимация из учебника KDE
KDE	Файлы из учебника KDE
Linapi	Файлы из учебника Unix
Maxima	Символьные вычисления
OOo Basic	Дополнительный код из учебника
PAW	Исходные тексты и данные для анализа
Python	Файлы из учебника
ROOT	Исходные тексты
XDMCP	Видео из учебника

Рабочий стол

Basket	Информационный менеджер KDE
ClamTk	Интерфейс к антивирусу
Grsync	Интерфейс к Rsync
HomeBank	Персональный финансовый менеджер
TinyFCK	Редактор и файловый менеджер
Xconfig	Помощник в настройке X

Разработка

Binutils	Утилиты разработчика
Compilers	Компиляторы с открытым кодом
Protoeditor	Текстовый редактор KDE
Ruby	Язык программирования
Ruby on Rails	Среда web-разработки
RubyGems	Менеджер пакетов Ruby

Дистрибутивы

Damn Small Linux	Ультралегкий настольный дистрибутив. Live-версия Mandriva.
Mandriva One	Live-версия Mandriva.
RIP	Дистрибутив для восстановления системы.
Ubuntu	Супер-дружелюбный дистрибутив.

Игры

Balazar Brothers	Трёхмерные платформы
Nexuiz	Трёхмерная многопользовательская стрелялка
Pyamb	Головоломка
RedShift	Авиасимулятор
Slam Soccer	Футбол

Графика

GEGE	Библиотека обработки изображений
Inkscape	Векторный графический редактор
Open Clip Art	Свободная галерея векторных изображений
Rawstudio	Конвертер raw-графики



При объеме 50 МБ, Damn Small Linux богат программами.

Справка

LDP	Руководства из Linux Doc Project.
LaTeX	С.М. Львовский. «Набор и верстка в системе LaTeX.»

HotPicks

Celtx	Инструмент сценариста
DOSBox	Эмулятор DOS и ПК
Foff	Клиент FTP с GTK-интерфейсом
GFast File	Файловый менеджер
KBilliards	Бильярд
Lighttpd	Web-сервер
Mplay	Медиа-проигрыватель
NeoDraughts	Настольная игра
RIB	Лейджер рабочего стола
Telak	Просмотр изображений

Интернет

Flail	Почтовый клиент
Kopete	Обмен мгновенными сообщениями для
KDE	
Liferea	RSS-ридер
Willow	Фильтрующий прокси-сервер

Офис

AbiWord	Текстовый процессор
Morla	Редактор PDF-документов
Texmacs	Редактор WYSIWYG

Система

AppArmor	Система безопасности
Gsambad	Браузер Samba
PAM-USB	PAM-модули для USB
Video drivers	Драйверы ATI и Nvidia

Сервер

Cloudwiki	Wiki-система
FinScribe	Web-ауторинг
Pen	Балансировщик нагрузки
Sshproxy	SSH-прокси для Python
Wefis	Файловый сервер с web-интерфейсом

Звук

Mesik	Музыкальный проигрыватель
MMA	Генератор музыкального аккомпанемента
Serpentine	Запись музыкальных CD-дисков

Главное

Avifile	Библиотека чтения/записи AVI файлов
CheckInstall	Создание двоичных пакетов
Coreutils	Утилиты командной строки
CSV	Индекс диска LXF
Glib	Низкоуровневая библиотека
Glibc	Библиотека GNU C
GTK	Инструментарий разработчика пользовательского интерфейса
Jigdo	Утилита создания ISO-образов
Kernel	Последняя версия ядра Linux
libsigc	Система обратных вызовов с проверкой типов
libXML	Анализатор XML и инструментарий
ncurses	Язык программирования
Python	Запись образов на дискеты
Rawrite	Не зависящий от системы менеджер загрузки
SBM	Мультимедийная библиотека
SDL	

DISTROS

МИНИ-ДИСТРИБУТИВЫ

В этом месяце компанию мини-дистрибутивов украшает Damn Small Linux 3.0: миниатюрный дистрибутив, позволяющий преспокойно работать на Pentium с 32 МБ ОЗУ; он взобрался на вершину списка нашего сравнения Live-дистрибутивов в LXF80, получив 9 очков из 10 (тогда это была версия 2.2).

В версии 3.0 добавились ACPI, Fuse и поддержка SSHFS, а также файловая система UnionFS, позволяющая вам объединять несколько файловых систем в одно дерево каталогов. Damn Small Linux работает как Live-дистрибутив (но его можно установить и на жесткий диск). Чтобы запустить его, запишите образ dsl-3.0.1.iso (из каталога **Distros/DamnSmall** нашего DVD) на CD-R и загрузитесь с него. Если при загрузке возникнут проблемы, попробуйте вариант образа с syslinux.

Также в разделе Дистрибутивы имеется RIP (или Recovery Is Possible, Восстановление Возможно) – небольшой Live-дистрибутив, помогающий исправлять проблемы в операционной системе. Он содержит утилиты для монтирования различных файловых систем, редактирования конфигурационных файлов и доступа к сети. Вам стоит записать образ **RIPLinux-1.7.grub.iso** на CD-R, чтобы, если что-то пойдет не так в вашем основном дистрибутиве Linux, вы смогли быстро загрузиться с компакт-диска и выяснить, в чем проблема.

ДИСТРИБУТИВЫ

MANDRIVA ONE 2006

Ubuntu, возможно, и самый популярный дистрибутив в данный момент, но Mandriva тоже очень популярен на рабочих столах, и заслуженно – это отполированный и дружелюбный к пользователю продукт. Теперь он доступен в виде Live-дистрибутива под именем Mandriva One на нашем DVD. Загрузите с него ваш ПК, и когда появится меню загрузки, выберите Start Mandriva One. Во время загрузки вам зададут несколько вопросов, касающихся настройки системы, а затем вы увидите рабочий стол KDE, работающий так же, как и в случае обычного дистрибутива Mandriva. Это отличный способ познакомиться с возможностями Mandriva или показать его друзьям, либо остаться в Linux, даже если вы вынуждены пользоваться ПК, на котором установлена Windows.

Mandriva One создан на базе Mandriva 2006, с рабочим столом KDE 3.4 и *OpenOffice.org* в качестве офисного пакета. Mandriva рекомендует ПК как минимум с 512 МБ ОЗУ и процессором 1-ГГц, но дистрибутив можно запустить и на 256-МБ памяти, если вас не бесит некоторое замедление в работе. Имейте в виду, что



На вид вылитая Mandriva, но это Live-дистрибутив!



Великолепный Центр управления Mandriva – ателье, где мгновенно кроют конфигурации системы!

независимо от скорости вашего компьютера Live-дистрибутив всегда будет работать гораздо медленнее, чем установленный на жесткий диск, просто потому, что он вынужден читать все с DVD!

Если Mandriva с виду вам понравилась и вы хотите ее установить, то вам повезло. Mandriva One включает

чрезвычайно простой инсталлятор, который проведет вас через процесс установки за несколько минут – вся сложная работа будет проделана за вас. Для запуска инсталлятора кликните два раза на иконку Live Install на рабочем столе. На следующей странице мы пройдемся по процедуре установки Mandriva.

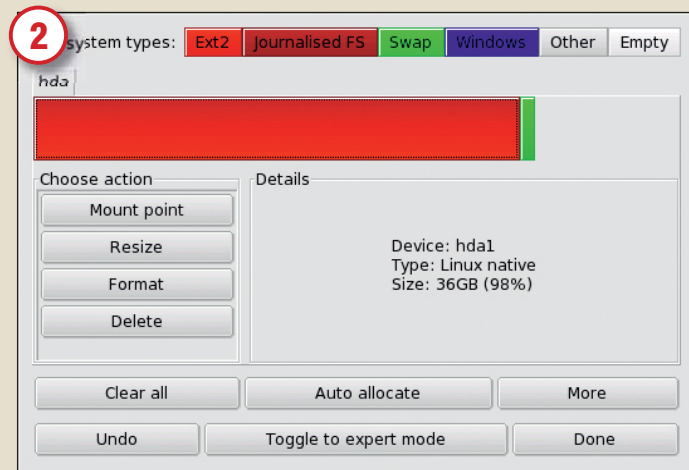


УСТАНОВКА MANDRIVA ONE 2006

Следуйте нашим инструкциям по установке Mandriva на ваш жесткий диск. Возникли проблемы? Обращайтесь на наш форум: www.linuxformat.ru.



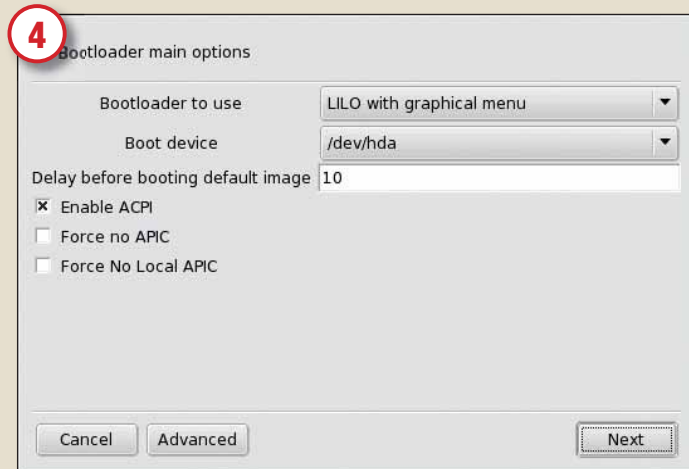
Кликните два раза на иконку Live Install на рабочем столе – появится инсталлятор. Нажмите Next, чтобы пропустить первый экран, и вы окажетесь в интерфейсе менеджера разделов. Можете выбрать установку на весь жесткий диск, либо разбить его на разделы.



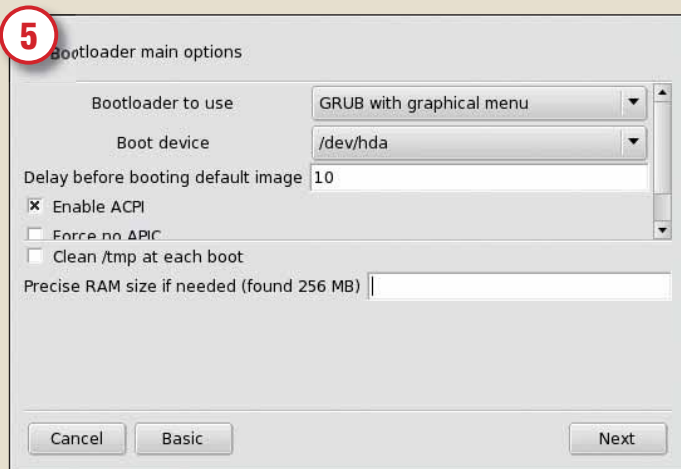
Если вы выбрали вариант с разбивкой, то учтите, что вам понадобится 5 Гб для Linux с файловой системой ext3, точкой монтирования в / и разделом подкачки (512 Мб). Кликните Done в подтверждение того, что вы согласны отформатировать разделы.



Инсталлятор начнет копировать файлы Mandriva One на ваш жесткий диск. Это не займет много времени – обычно до 20 минут, кроме того, вы будете видеть, как идет процесс.



После завершения копирования вам предложат установить загрузчик, чтобы ваш ПК смог загружать Linux (а не только Windows). Если вы не знакомы с процессом установки Linux, примите значения по умолчанию.



Если вы хотите настроить загрузчик, можете переключиться с загрузчика по умолчанию Lilo на Grub и выбрать между текстовым и графическим режимом. Вы также можете указать значение таймута и дополнительные опции, во избежание проблем при загрузке.



Процесс завершен! Кликните на кнопку Finish и перезагрузитесь, не забыв предварительно вынуть DVD-диск. Mandriva One теперь будет запускаться с вашего жесткого диска.

ДРУГИЕ ПРОГРАММЫ НОВЫЕ РЕЛИЗЫ

В разделе **Graphics** помещен релиз этого месяца – векторный графический редактор *Inkscape* 0.44, который неуклонно хорошеет. Несмотря на небольшой номер версии, *Inkscape* способен выполнять реальные задачи, что и продемонстрировано в нашей постоянной серии учебников. Новыми возможностями данного релиза стали диалог слоев, поддержка вырезания и масок, улучшенный экспорт в PDF и режим эскиза. Здорово, что команда *Inkscape* продолжает делать независимые от дистрибутива пакеты, поэтому процесс установки длится секунды – достаточно запустить файл **Inkscape-0.44.x86.package** (возможно, вам понадобится сделать его исполняемым, либо наберите **sh Inkscape-0.44.x86.package** в оболочке).

После установки *Inkscape* вам понадобятся картинки для редактирования (если только вы не планируете создать их с нуля сами!). Поэтому мы включили на диск отличный открытый архив с изображениями, в который входят 6900 изображений в SVG-формате. В архив вошли обычные картинки с животными, машинами, едой и так далее – некоторые отлично прорисованы, другие похуже, но они вполне подходят как основа для создания своих изображений.

Траты, траты, траты

Если вы собираетесь привести в порядок свои финансы, взгляните на программу *HomeBank* из раздела **Desktop**. Это приложение с открытым кодом включает поддержку нескольких учетных записей, детальную статистику и графики, отображающие ваш финансовый статус. Разработка

началась на Amiga в 1995 г., а затем перенесена в мир Linux. В планах есть и версия для Windows. Для запуска программы распакуйте файл: **homebank-3.2a1-binary.tar.gz**, перейдите в получившийся каталог и запустите **homebank** (к счастью, единственная зависимость – GTK, так что на большинстве компьютеров все пройдет успешно).

На диске еще очень много достойного исследования – ищите то, что вам нужно, открыв файл **index.html** из корневого каталога DVD и походив по категориям,

Галерея Open Clip Art содержит тысячи свободных SVG-картинок – среди них этот странный пингвин...



где есть скриншоты и описания к каждому приложению. Дайте шанс *Serpentine*, если вам нужна новая утилита записи аудио-CD, или *Gsambad (System)* в качестве умного интерфейса к *Samba*. Как всегда, если у вас есть рекомендации по содержимому нашего диска, киньте мне письмо на адрес: mike.saunders@futurenet.co.uk.



«Пытаешься меня поджечь? Это неспортивно, старина!»

ИГРЫ И НАКОНЕЦ...

Вдоволь наисследовавшись Ubuntu и Mandriva, почему бы не расслабиться за игрой? В этом месяце появилось множество новых и обновленных версий игр, и как всегда, все сливки вы найдете в разделе **Games** на нашем DVD. Вернемся в прошлое, к номеру *LXF71* – возможно, вы вспомните трехмерную стрелялку *Nexuiz*. Этот бластер, построенный на движке *DarkPlaces* (который в свою очередь базируется на *Quake*), был спроектирован для сетевой многопользовательской игры, но в новой версии 2.0 появился желанный многими однопользовательский режим – кампания.

Чтобы играть в эту игру, потребуется 3D-акселератор, 200 МБ жесткого диска и по меньшей мере 1.5-ГГц процессор. Распакуйте файл **nexuiz-20.zip** в ваш домашний каталог – придется подождать, поскольку будут распаковываться большие файлы. Перейдите в получившийся каталог **Nexuiz** и запустите двоичный файл **nexuiz-linux-686-glx**. После загрузки вы увидите главное меню, позволяющее выбрать различные режимы игры. Если вы просто хотите немного пострелять, кликните на пункт *Instant Action* в нижнем левом углу и нажмите кнопку *Space* – игра и начнется.

Если такие игры не для вас, попытайтесь предотвратить позор Великобритании на Кубке Мира по футболу, поиграв в *Slam Soccer*. Этот футбольный симулятор выполнен в мультяшном стиле и включает 80 команд, 20 стадионов и различные игровые режимы. Имейте в виду, что этой игре требуется Java (на диске она упоминается как *Bolzplatz*). Наконец, поклонники платформных игр должны попробовать *Balazar Brothers* – можете даже наблюдать видео этой игры ([balazar_brother.mpeg](#)). **LXF**

Statistics Report							
Display							
For:	Category						
View:	Expense						
<input checked="" type="checkbox"/> By amount							
Date filter							
From:	03.10.2003						
To:	21.06.2006						
Range:	All date						
Month:	----						
Year:	2006						
Expense: -4 025,62 € Income: 9 980,88 € Balance: 5 955,26 €							
Result	Expense	%	Income	%	Balance	%	
Invoices:Housing	-1 683,00 €	(-41,81)			-1 683,00 €	(28,26)	
(None)	-577,62 €	(-14,35)	18,00 €	(0,18)	-559,62 €	(10,00)	
Car	-500,00 €	(-12,42)			-500,00 €	(8,40)	
Computer	-263,00 €	(-6,53)			-263,00 €	(4,42)	
Invoices:in line service/Internet	-210,00 €	(-5,22)			-210,00 €	(3,53)	
Food:Grocer	-159,00 €	(-3,95)			-159,00 €	(2,67)	
Food:Restaurant	-150,00 €	(-3,73)			-150,00 €	(2,52)	
Taxes:Income taxes	-149,00 €	(-3,70)			-149,00 €	(2,50)	
Withdrawal of cash	-89,00 €	(-2,21)			-89,00 €	(1,49)	
State	Date	Info	Payee	Wording	Expense	Income	Category
✓	03.10.2003	VISA	Free	Internet DSL	-30,00 €		In line service/Internet
✓	03.11.2003	VISA	Free	Internet DSL	-30,00 €		In line service/Internet
✓	03.12.2003	VISA	Free	Internet DSL	-30,00 €		In line service/Internet
✓	03.01.2004	VISA	Free	Internet DSL	-30,00 €		In line service/Internet
✓	03.02.2004	VISA	Free	Internet DSL	-30,00 €		In line service/Internet
✓	03.03.2004	VISA	Free	Internet DSL	-30,00 €		In line service/Internet
✓	03.04.2004	VISA	Free	Internet DSL	-30,00 €		In line service/Internet

HomeBank: Бесплатный и простой учет средств, говорят разработчики. Достойная альтернатива *GnuCash*, скажем мы.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ:

ГЛАВНЫЙ РЕДАКТОР

Валентин Сеницын info@linuxformat.ru

Литературные редакторы

Елена Толстякова, Александр Толстой, Иван Мищенко

Переводчики

Александр Бижмев, Сергей Захаров, Светлана Кривошеина, Александр Кузьменков, Алексей Опарин, Сергей Супрунов, Александр Толстой, Александр Черных, Юлия Шабунио, Павел Шер

Допечатная подготовка

Мария Пучкова
Родион Водейко

Креативный директор

Станислав Медведев

Технический директор

Денис Филиппов

Директор по рекламе

Денис Игнатов +7 812 965 7236 advert@linuxformat.ru

Заместитель генерального директора

Софья Виниченко

Генеральный директор

Павел Фролов

УЧРЕДИТЕЛИ

частные лица

ИЗДАТЕЛИ

Павел Фролов, Станислав Медведев

Отпечатано в типографии «Текст»

000 «ППК «Текст»

188680, Ленинградская область,

Всеволожский район, Колтуши, д.32

Заказ _____

Пре-пресс: [drive-group](http://drive-group.ru)

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ:

Редактор Ник Вейч (Nick Veitch) nick.veitch@futurenet.co.uk

Заместитель редактора Пол Хадсон (Paul Hudson) paul.hudson@futurenet.co.uk

Художественный редактор Эфрайн Эрнандес-Мендоза

(Efrain Hernandez-Mendoza) efrain.hernandez-mendoza@futurenet.co.uk

Новостной редактор Майк Сондерс (Mike Saunders) mike.saunders@futurenet.co.uk

Литературный редактор

Ребекка Смолли (Rebecca Smalley) rebecca.smalley@futurenet.co.uk

Штатный автор

Грам Моррисон (Graham Morrison) graham.morrison@futurenet.co.uk

Ассистент по выпуску

Эндрю Грегори (Andrew Gregory) andrew.gregory@futurenet.co.uk

АВТОРЫ

Марк Бейн (Mark Bain), Ладислав Боднар (Ladislav Bodnar), Нейл Ботвик (Neil Bothwick), Энди Ченел (Andy Channelle), Кингс Кобблер (Kings Cobbler), Кристин Коуэн (Crispin Cowan), Майкл Дж. Хэммел (Michael J Hammel), Евгений Балдин, Андрей Боровский, Дмитрий Кирсанов, Владимир Попов, Сергей Супрунов, Тихон Тарнавский, Алексей Федорчук, Сергей Яремчук.

ХУДОЖЕСТВЕННЫЙ ОТДЕЛ

Художники: Йон Блекшау (Jon Blackshaw), Эмиг Петел (Amit Patel)

Фотографии: Corbis UK Ltd, Photodisc, Joby Sessions, SuperStock

Иллюстрации: Шейн Коллинж (Shane Collinge), Крис Винн (Chris Winn),

Elly Walton Illustrations

Создание диска: Майк Сондерс (Mike Saunders)

КОНТАКТНАЯ ИНФОРМАЦИЯ

England: Linux Format, 30 Monmouth Street, Bath BA1 2BW

Tel 01225 442244 Email linuxformat@futurenet.co.uk

Россия: Санкт-Петербург, ул. Гончарная, 23, офис 54, телефон: (812) 717-00-37

Email: info@linuxformat.ru

Web: www.linuxformat.ru

Авторские права. Статьи, переведенные из английского издания Linux Format, являются собственностью или лицензией Future Publishing Ltd (Future plc group company). Все права зарегистрированы. Никакая часть данного журнала не может быть повторно опубликована без письменного разрешения издателя.

Все письма, независимо от способа отправки, считаются предназначенными для публикации, если иное не указано явно. Редакция оставляет за собой право корректировать присланные письма и другие материалы. Редакция Linux Format получает неэксклюзивное право на публикацию и лицензирование всех присланных материалов, если не было оговорено иное. Linux Format стремится оставлять уведомление об авторских правах всюду, где это возможно. Свяжитесь с нами, если мы не упомянули вас как автора предложенных вами материалов и мы постараемся исправить эту ошибку. Редакция Linux Format не несет ответственности за опечатки.

Все присланные материалы могут быть помещены на CD или DVD-диск, поставляемые вместе с журналом, если не было оговорено иное.

Ограничение ответственности: используйте все советы на свой страх и риск. Ни при каких условиях редакция Linux Format не несет ответственность за повреждение или ущерб, нанесенные вашему компьютеру и периферии вследствие использования тех или иных советов.

За содержание рекламных материалов редакция ответственности не несет.

Linux-зарегистрированная торговая марка Линуса Торвальдса (Linus Torvalds). Название «GNU/Linux» заменяется на «Linux» в целях сокращения. Остальные торговые марки являются собственностью их законных владельцев.

Linux Format является торговой маркой Future Publishing Ltd (Future plc group company).

За информацией о журналах, издаваемых Future plc group company, обращайтесь <http://www.futureplc.com>



© Linux Format 2005

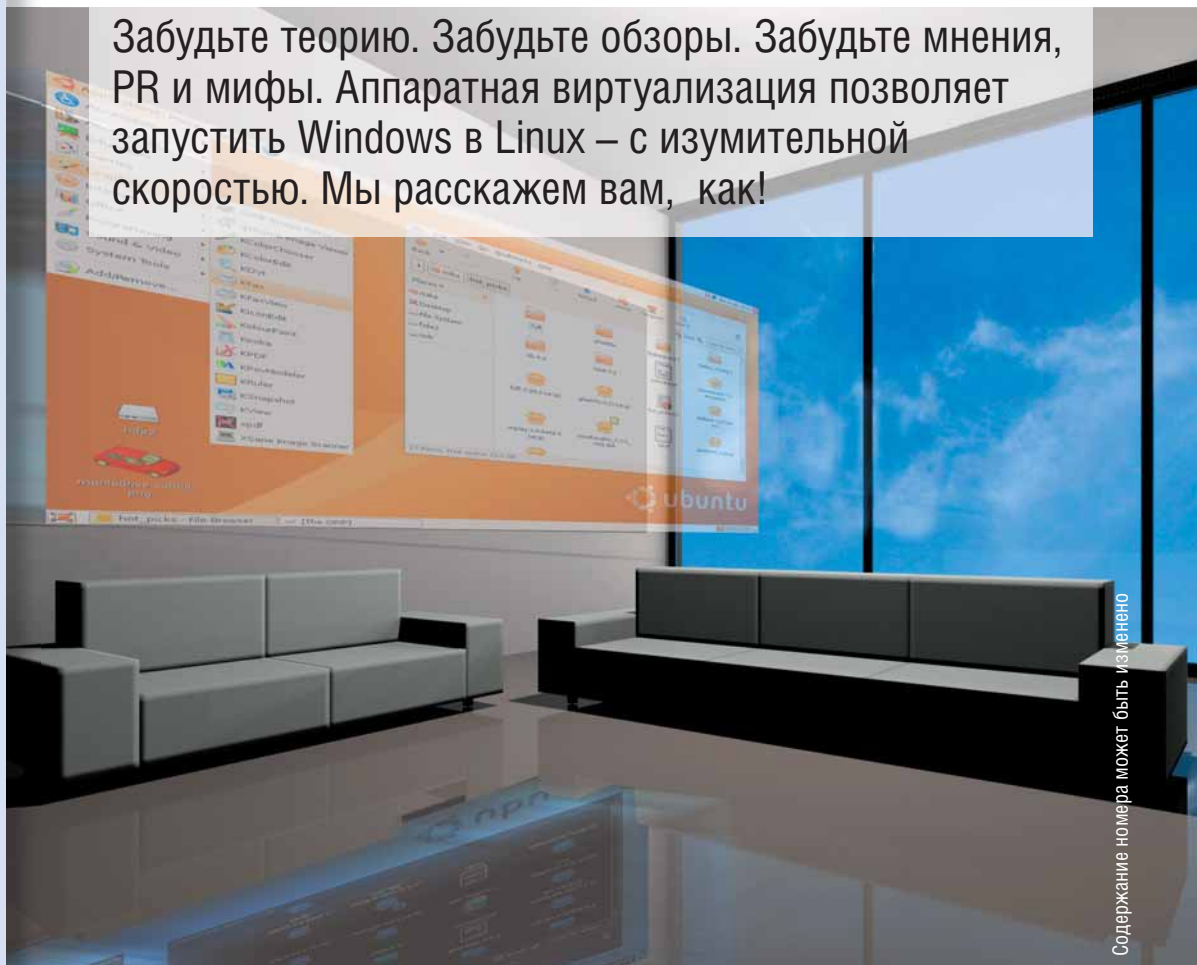
© Future Publishing Ltd 2005

В следующем месяце

LINUX FORMAT 10(84) ОКТЯБРЬ 2006

ВИРТУАЛЬНАЯ РЕАЛЬНОСТЬ

Забудьте теорию. Забудьте обзоры. Забудьте мнения, PR и мифы. Аппаратная виртуализация позволяет запустить Windows в Linux – с изумительной скоростью. Мы расскажем вам, как!



Открыто для всех?

Как OSDL помогает свободному ПО

Google Earth

Понравилась Picasa? Попробуйте это!

RAID-массивы

Пусть ваши диски запоют в унисон

ПОДПИСКА НА LINUX FORMAT

ПОДПИСКА В ЛИНУКСЦЕНТРЕ

Сколько стоит подписка?

Подписка на журнал «Linux Format» **12 номеров** (январь, февраль, март, апрель, май, июнь, июль, август, сентябрь, октябрь, ноябрь, декабрь) стоит **1800 рублей**

Подписка на журнал «Linux Format» **6 номеров** (июль, август, сентябрь, октябрь, ноябрь, декабрь 2006 года) стоит **900 рублей**

Как оформить подписку?

Чтобы оформить подписку на журнал «Linux Format», необходимо зарегистрироваться в интернет-магазине Linuxcenter.Ru, указав ФИО и подробный почтовый адрес подписчика, заказать товар «Подписка на журнал «Linux Format» 12 номеров 2006 года», или товар «Подписка на журнал «Linux Format» второе полугодие 2006 года», получить от системы квитанцию для оплаты в любом отделении Сбербанка (для физических лиц) или счет для оплаты по безналичному расчету (для юридических лиц)

Как оплатить подписку?

- по выставленному счету (для юридических лиц)
- по квитанции в любом отделении Сбербанка

Плюсы подписки

- подписка дешевле!
- гарантированное получение нового номера журнала!

ПОДПИСКА - 2006!
ПОДПИСКА ПО КАТАЛОГАМ РФ

Каталог агентства «РОСПЕЧАТЬ» – подписной индекс **20882**

Каталог «ПРЕССА РОССИИ» – подписной индекс **87974**

Ф. СП-1

Министерство связи РФ

АБОНЕМЕНТ на газету-журнал **Linux Format** (индекс издания)

(наименование издания) Количество комплектов:

на 200_ год по месяцам

1	2	3	4	5	6	7	8	9	10	11	12
---	---	---	---	---	---	---	---	---	----	----	----

Куда (почтовый индекс) (адрес)

Кому (фамилия, инициалы)

ДОСТАВОЧНАЯ КАРТОЧКА

ПВ место литер на газету-журнал **Linux Format** (индекс издания)

(наименование издания)

Подписная цена руб. коп. Количество комплектов:

на 200_ год по месяцам

1	2	3	4	5	6	7	8	9	10	11	12
---	---	---	---	---	---	---	---	---	----	----	----

Куда (почтовый индекс) (адрес)

Кому (фамилия, инициалы)

ПОДПИСКА НА LINUX FORMAT

ПОДПИСКА ПО КАТАЛОГАМ СНГ И БЛИЖНЕГО ЗАРУБЕЖЬЯ

Каталог «Российская Пресса»- совместный проект Государственного предприятия «Казпочта», Агентства «Книга-Сервис» и АРЗИ.

Блок изданий АРЗИ в национальных Каталогах Украины и Беларуси. В Азербайджане, Армении, Грузии, Киргизии, Узбекистане и Молдове - по изданиям, включенным в Объединенный каталог, распространяемые через АРЗИ.

Азербайджан

- по Объединенному каталогу российских изданий через Предприятие по распространению печати «Гасид» (370102, г. Баку, ул. Джавадхана, 21);

Армения

- по списку номенклатуры «АРЗИ» через ГЗАО «Армпечать» (375005, г.Ереван, пл.Сасунци Давида, д.2) и ЗАО «Контакт-Мамул» (375002, Г.Ереван, ул.Сарьяна, 22);

Белоруссия

- по Каталогу изданий стран СНГ через РГО «Белпочта» (220050, г.Минск, пр-т Ф.Скорины, 10);

Грузия

- по списку номенклатуры «АРЗИ» через АО «Сакпресса» (380019, г.Тбилиси, ул.Хошараульская, 29) и АО «Мацне» (380060, г.Тбилиси, пр-т Гамсахурдия, 42);

Казахстан

- по Каталогу «Российская Пресса» через ОАО «Казпочта» и ЗАО «Евразия пресс»;

Молдавия

- по каталогу через ГП «Пошта Молдавей» (МД-2012, г.Кишинев, бул. Штефан чел Маре, 134);
- по списку через ГУП «Почта Приднестровья» (МД-3300, г.Тирасполь, ул.Ленина, 17);
- по прайс-листу через ООО Агентство «Editil Periodice» (2012, г.Кишинев, бул. Штефан чел Маре, 134).

Узбекистан

- по Каталогу «Davriy nashrlar» российские издания через Агентство по распространению печати «Davriy nashrlar» (7000029, Ташкент, пл.Мустакиллик, 5/3, офис 33);

Украина

- Киевский главпочтамт.
- Подписное агентство «KSS» Телефон/факс (044)270-62-20, 270-62-22



АЛЬТЕРНАТИВНЫЕ АГЕНСТВА РФ

Агентство «Интер-Почта»
(095) 500-00-60, курьерская доставка по Москве.

Агентство «Вся Пресса»
(095) 787-34-47

Агентство «УралПресс»

- Екатеринбург, Березовский, В. Пышма, Первоуральск тел. (343) 375-80-71, 375-84-93, 375-84-39, факс 375-62-74, info@ural-press.ru
- Нижний Тагил тел. (3435) 411448, 417709, ntagil@ural-press.ru
- Челябинск тел. (351) 262-90-03, 262-90-05, pochta@chel.surnet.ru
- Пермь тел. (3422) 60-24-40, 60-22-95, 60-35-42, parma-press@permonline.ru