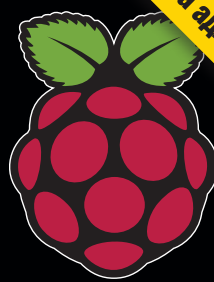


4 ТОП-ДИСТРИБУТИВА!
Fedora 17, Mint 13, Mageia 2
ArchBang, книга админа Debian и более того!

LINUX FORMAT



» **Raspberry Pi:**
Превратите свою в файл-сервер

Главное в мире Linux

Август 2012 № 8 (160)

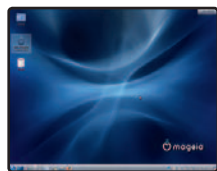
Как сделать миллиард долларов способом Red Hat

Сделайте свободу функцией » Дайте клиентам то, что им нужно
» Продавайте стабильность » Станьте Linux-пюре » И все это — не будучи злом!



Хакинг от Хокни

» Преобразование фотографий
в шедевры поп-арта с. 66



Mageia 2

» Есть ли жизнь после Mandriva?
А передайте-ка мне Камамбер с. 17



Бьорн Балаж

« Прелесть открытого
ПО — каждый может
встать на плечи гигантов »

Бьорн Балаж: как быть гражданином в ПО с. 46

Также в номере...

Город открытых данных

» Данные желают свободы. Тем
более что за все уплачено с. 50



Firefox ускоряет темп

» Заново откройте для себя
мощь браузера от Mozilla с. 54



Сборка суперкомпьютера

» Кластер из ваших машин забудет IBM с. 88

ПЛЮС!
Углубимся
в Android
с. 24



Управление диском
Идеальные разделы
» В ваше хранилище —
больше дистрибутивов!

Тщеславие в командах
Кастинг оболочки
» Раструбите про свое мастер-
ство по всем интернетам

Открытые социальные сети
Говорить свободно
» Шлите твиты, повсюду суйте
свой нос и всех доставайте

ПОДПИСНЫЕ ИНДЕКСЫ В КАТАЛОГАХ
Агентство «Роспечать» — 36343
«Почта России» — 11932, «Пресса России» — 90959



Новое поколение средств защиты

Межсетевые экраны ССПТ, не имеющие IP-адреса

ССПТ-2 — это сертифицированное ФСТЭК, ФСБ и ГАЗПРОМСЕРТ средство защиты информации нового поколения, реализующее функции межсетевого экрана, но при этом остающееся «невидимым» для любых протоколов и тестовых воздействий, что достигается за счет отсутствия физических и логических адресов на его фильтрующих интерфейсах. ССПТ-2 невозможно обнаружить никакими известными средствами удаленного мониторинга сети.

Скрытность функционирования межсетевого экрана повышает надежность системы защиты в целом и существенно упрощает процедуру установки ССПТ-2 в компьютерные сети и функционирующие на их основе информационные и телематические системы.

Защита для высокоскоростных корпоративных сетей Ethernet 100/1000 Мбит/с

Сертифицированы ФСТЭК и ФСБ (3-й класс защиты)

На базе процессоров с 64-разрядной многоядерной архитектурой



Назначение устройства

Основное средство защиты для реализации различных политик информационной безопасности с помощью:

- фильтрации пакетов на канальном, сетевом, транспортном и прикладном уровнях;
- управления транспортными соединениями между отдельными узлами ЛВС или виртуальной ЛВС (VLAN);
- контроля контента данных на прикладном уровне с учетом направления, времени и типа протоколов передачи трафика.

Дополнительное устройство защиты для:

- обеспечения безопасности функционирования ранее установленных в компьютерной сети средств защиты и устройств маршрутизации;
- мониторинга трафика с возможностью анализа данных регистрации пакетов по различным критериям и интеграции с IDS;
- обеспечения функционирования сетевых распределенных телематических приложений и GRID-ресурсов.

Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

Linux-эксперт для вашего бизнеса. www.linuxcenter.ru

Linux  center

1С:ПРЕДПРИЯТИЕ 8

Новое поколение систем автоматизации управления и учета

- 1С:Управление производственным предприятием
- 1С:Бухгалтерия
- 1С:Зарплата и управление персоналом
- 1С:Управление торговлей
- Отраслевые и специализированные решения

Теперь работает в Linux!

Для запуска «1С:Предприятия» в Linux необходимо наличие WINE@Etersoft, который входит в состав дистрибутива Mandriva Linux 2011 Powerpack



Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

Linux-эксперт для вашего бизнеса. www.linuxcenter.ru





Нордавинд Северо-Запад

Охранные системы на базе GNU/Linux

Компания «Нордавинд» существует с 2004 года и является пионером в области разработки и производства компонентов для охранных систем на базе высоконадежной операционной системы GNU/Linux

Наши услуги и продукты

- Проведение НИР
- Разработка и изготовление систем охранного телевидения и сопутствующих компонентов
- Проектирование сложных систем и комплексов
- Широкий ассортимент готовой продукции

Наши системы видеонаблюдения предназначены для:

Паркингов
Зданий
Бизнес-центров

Торговых комплексов и складов
Открытых площадок
Жилых домов

Объектов повышенной секретности
Режимных предприятий

Сертификаты ФСТЭК и ФСБ

4 канальная система видеонаблюдения стоимостью **40 тыс. руб.** —
в 2 раза дешевле, чем предыдущие технологии.



ТелеВизард

Многофункциональный сервер общего назначения.



ТелеВизард Авто

Самая бюджетная и высокоэффективная система распознавания автомобильных номеров.



Источники видеосигнала

Комплекс eyeSense, в состав которого входят видеочамера и объектив. Дополнительное оборудование — ИК-прожектор и климат-защита.

«Нордавинд Северо-Запад»

Научно-производственная компания

Тел. +7 (812)

Тел./факс +7 (812)

309 0686
640 4990

<http://spb.nordavind.ru>

e-mail: spb@nordavind.ru

Приглашаем к партнерству дилеров и проектировщиков систем

Что мы делаем

» Мы поддерживаем открытое сообщество, предоставляя источник информации и площадку для обмена мнениями.

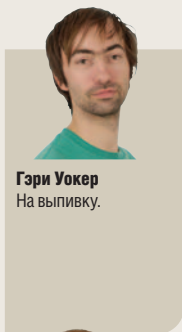
» Мы помогаем всем читателям получить от Linux максимум пользы, публикуя статьи в разделе «Учебники»: здесь каждый найдет что-то по своему вкусу.

» Мы выпускаем весь код, появляющийся на страницах раздела «Учебники», по лицензии GNU GPLv3.

» Мы стремимся предоставлять точные, актуальные и непредвзятые сведения обо всем, что касается Linux и свободного ПО.

Кто мы

Red Hat сделала миллиард по доходам за прошлый финансовый год. Будь этот миллиард у вас, на что бы вы его потратили?



Гэри Уокер
На выпивку.



Эндрю Грегори
Права на весь каталог записей Моррисси... и весь его гардероб.



Эфраин Эрнандес-Мендоса
Выстроил бы развилку моста через Северн поперек Атлантики и до Мексики.



Бен Эверард
На биробота руки для работы с ПО распознавания речи. Обещаю быть очень аккуратным, веля ей разорвать пакет с орешками.



Маянк Шарма
Купил бы яхту Кима Доткома и разъезжал бы по всему свету, занимаясь кроважидным пиратством.



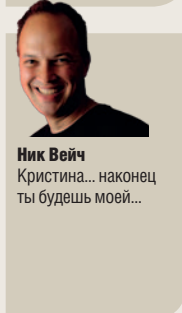
Джонатан Робертс
Скупил бы все молочные фермы в Девоне и в одночасье заморозил производство чая со сливками. Покоряй изнутри!



Майк Сондерс
Я построю свой мир в *Minecraft* и заживу в нем... но без жутких монстров. Они до сих пор сняты мне в кошмарах.



Валентин Синицын
Собрал бы свой собственный дистрибутив, с играми и платной поддержкой.



Ник Вейч
Кристина... наконец ты будешь моей...



Сюзан Линтон
На собственную фабричку печенья. М-м-м, эти шоколадные кружочки с джемом, которые всегда меньше, чем вы помнили...



Шашанк Шарма
Два миллиарда акций Facebook. Они уже достаточно подешевеют, когда вы будете читать это. А на сдачу куплю шоколадку.



Нейл Ботвик
40000000 штук Raspberry Pi. Потом построил бы из них суперкомпьютер и захватил весь мир.

Политэкономия Red Hat

» Если верить Википедии, предметом политэкономии являются «социально-экономические явления, которые складываются в определенных сферах». Историю и достижения компании Red Hat без всяких сомнений можно назвать именно таким явлением.

Если рассматривать только экономику, то перед нами – самая крупная и успешная (по критериям оценки бизнеса инвесторами) компания, занимающаяся свободным ПО.

А вот причины этого успеха, на мой взгляд, находятся в аспекте социальном. Компании впервые в индустрии удалось объединить ценности двух, казалось бы, несовместимых культур – сообщества разработчиков свободного ПО и традиционной корпоративной Америки. Причем от обеих было взято все лучшее. Энтузиазм и постоянное стремление к совершенству кода дополняется надежностью компании и гарантией (закрепленной юридически в виде договоров) высокого качества предоставляемых услуг, таких как долговременная поддержка, обучение, сертификация специалистов и другие. Именно это сочетание культур принесло Linux в мир крупных корпораций и государственных структур США. Но при этом не забыто и сообщество. Код RHEL используется во многих самых разных проектах, от CentOS до MC BC.

Главное, что удалось доказать Red Hat – вполне возможно зарабатывать самому и при этом помочь заработать другим. В том, собственно, и вся политэкономия...

Кирилл Степанов

Главный редактор

» info@linuxformat.ru

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

Вопросы распространения: sales@linuxformat.ru

Web-сайт: www.linuxformat.ru

» Адрес редакции: Россия, Санкт-Петербург, Лиговский пр., 50, корп. 15

» Телефон редакции: (812) 309-06-86. Дополнительная информация на с. 112

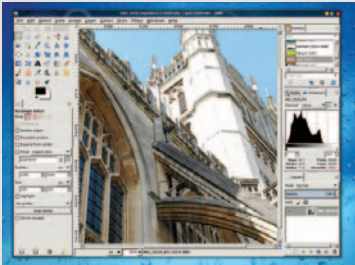
Содержание

Если нас послушают, все скоро перейдут на Linux-совместимую читалку PDF.

Обзоры

GIMP 2.8 14

Вмонтируйте себя в фотографию Кристины Агилеры и наврите всем, что вы дружны, не платя Adobe 650 ф. ст. за это удовольствие.



› Батский камень чересчур бликует — абердинский гранит получился бы лучше.

Miro 5 16

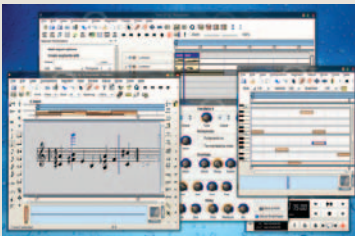
Это недооцененный медиа-плеер типа все-в-одном заслуживает вашего внимания, если у вас смешанная подборка материала для просмотра/прослушки.

Mageia 2 17

Мы очень хотели его полюбить, но второй релиз Mageia особого восторга не вызвал. Как вам понравится окрошка из Unity/KDE?

Pianoteq 4 18

Выпустите на волю Ференца Листа внутри вас ради этого блестяще-заумного синтезатора пианино — о, дивный Стейнвей-прототип...



› Как выяснилось, роялизм даром не дается.

Rosegarden 12.04 19

Ференц Лист не одобрил бы *Rosegarden*: у него слишком длинные пальцы для обычной клавиатуры Qwerty.

Runtu Gnome 12.04 20

Отечественный убунтоид ориентирован на русскоязычных пользователей и нетребователен к аппаратной части.



Шляпы долой перед Red Hat: Как сделать миллиард долларов с. 38



Сравнение: Открытые социальные сети с. 32



Что за штука — OpenCL? с. 58

Люди говорят



“ Это затеяли, потому что пара берлинцев решили обеспечить удобство ”

Бьорн Балаж из OpenUsability.org с. 46

На вашем бесплатном DVD



Fedora 17

» Передовые программы с улыбкой

Mint 13

» Пробуйте рабочие столы Mate и Cinnamon

ПЛЮС: Горячие новинки и коды к учебникам... **с. 106**

Ищите в этом номере



Города открытых данных 50

Данные – ключ к будущему человечества.

Firefox 54

Как Mozilla сломала монополию браузеров.

Сисадминам 60

Безопасность нужна и серверам, и пользователям.



Пропустили номер?

Узнайте на с. 108, как получить его прямо сейчас!



Учебники

Редактирование изображений Коллажи а-ля Хокни 66

Стоит ли париться с искусством? Проще всех надуть и вдесятеро быстрее изготовить все в GIMP.

Хакерство Raspberry Pi 70

Поиграйте в игры, послушайте музыку, постройте NAS – а потом можно и покушать.

Жесткие диски Установка нового диска 74

Нейл Ботвик избавит вас от мучений с разбиением на разделы – вот такой он добрый.

Командная строка Поделитесь опытом 76

Вы гениальны, и мир должен в этом убедиться. Майк Сондерс научит вас добиваться славы с помощью *Shelr*.

Электроника Arduino взволнован 80

Шагнем дальше: превратим Arduino в генератор замысловатых колебаний.

Языки программирования Erlang 84

Основа распределенной системы – механизм взаимодействия ее компонентов.

Суперкомпьютеры Кластеры Beowulf 88

Не путать с мифологическими чудовищами и клистиром. Влейтесь во всемирные расчеты.

Командная строка Текстовый браузер 92

Ellink уберезит вас от излишеств разных нехороших, присущих некоторым сайтоделам.

Постоянные рубрики

Новости 4

Вышел первый российский мультиклет, новый министр угрожает НПП, Mozilla представила мобильную ОС, а Линус Торвалдс пристыдил NVIDIA.

Android 24

Роман Ярыженко вникает в архитектуру ОС, а Джульетта Кемп покажет, как использовать HTTP-запросы и JSON, чтобы ваши приложения говорили с удаленными базами данных.

Сравнение 32

Найдите лучшую социальную сеть с открытым исходным кодом – здесь никто не употребит ваши конфиденциальные данные во зло.

Интервью LXF 46

Для вовлечения пользователей в тестирование ПО Бьорн Балаж применил когнитивный диссонанс.

Что за штука 58

OpenCL – открытый стандарт параллельного программирования для графических процессоров.

Рубрика сисадмина 60

Почему пароль "Password" необходимо срочно сменить...

Ответы 96

ПРОБЛЕМЫ LINUX РЕШЕНЫ! Преобразование винила, EXIF-данные MP3 в KDE и загрузка Flash.

Hotpicks 100

Отведайте горяченького: лучшие в мире новинки свободного ПО.

Диск Linux Format 106

Содержимое двустороннего DVD этого месяца.

Пропустили номер? ... 108

Купите парочку поскорее – иначе не видать Эффи отпусковых на отдых в Канкуне.

Школа LXF ---

А у нас каникулы! Лето все-таки...

Через месяц 112

Соберите идеальную Linux-машину и настройте домашний сервер с NAS.



ГЛАВНОЕ Первый отечественный мультиклет » НПП могут отменить
» Евронаступление » Огненная ОС » Торвальдс отжигает

ИНТЕРЕСНАЯ РАЗРАБОТКА

Мультиклеточный числодробитель

Выпущена первая опытно-промышленная партия российского микропроцессора MultiClet.

В начале июля 2012 г. вышла первая опытно-промышленная партия российского мультиклеточного микропроцессора MultiClet MCp0411100101 на четырех клетках. Он построен по уникальной пост-неймановской архитектуре и предназначен для задач управления и цифровой обработки сигналов. Также процессор способен выполнять программы общего назначения, распараллеливая «предложения» из RISC-подобных команд на клетки. Микропроцессор развивает идею роста производительности не за счет гонки мегагерц, а за счет оптимизирующей вычисления архитектуры.

Краткие характеристики:

- » номинальная частота: 100 МГц;
- » разрядность процессора: 32/64 бит;
- » память данных (ПД): 128 КБ;
- » память программ (ПП): 128 КБ;
- » производительность: 2,4 ГФлопс;
- » потребляемая мощность: 45 мВт

Цена микропроцессора в партии до 100 шт. – 462 руб. Цена отладочного комплекта с платой периферии – 38200 руб.

Области его применения – системы промавтоматики; универсальные навигационные приемники ГЛОНАСС/GPS/Galileo/COMPASS/IRNSS/QZSS; мобильные телефоны; видеотехника 3D; автомобильная электроника для бортовых систем, контролирующая дорожную обстановку; системы безопасности, распознавание «свой/чужой». Компания «Мультиклет» уже начала поставки на предприятия электронной промышленности России.

Выхода первого в мире мультиклеточного процессора с нетерпением ждали не только создатели продукта, но и многие российские компании, заключившие договора поставки еще на этапе научно-исследовательской работы над проектом.

«Мы получили с фабрики первую партию разработанных нами мультиклеточных процессоров – это поистине знаменательное событие не только для нашей компании, но и для всей электронной промышленности страны, – отметил генеральный директор ОАО «Мультиклет» Борис Зырянов. – Процессор предназначен для

» **Борис Зырянов** – д. т. н., генеральный директор ОАО «Мультиклет».



решения широкого круга задач управления и цифровой обработки сигналов в приложениях, требующих минимального энергопотребления и высокой производительности. Работа с мультиклеточным процессором для пользователей не отличается от обычной; специалистами разработан набор утилит, позволяющий писать программы для мультиклета. В его составе: ассемблер, редактор связей (компоновщик), компилятор Си, функциональная модель, а также в разработке – ОС RTOS».

Разработчики мультиклеточных процессоров завершили этап тестирования продукта, специалисты компании начали отгрузку процессоров и отладочных комплектов на предприятия страны 10 июля 2012 года. Промышленным концернам, сотрудничающим с компанией «Мультиклет», еще предстоит большая, серьезная работа по апробации полученных образцов.

О компании и технологии



ОАО «Мультиклет» – российская компания, занимающаяся разработкой, производством и выводом на рынок высокопроизводительных и дефектоустойчивых процессорных ядер и процессоров с низким энергопотреблением, спроектированных на базе мультиклеточной архитектуры. MultiClet (читается «мультиклёт») – термин, обозначающий процессорное ядро или процессор с мультиклеточной архитектурой.

Мультиклеточное процессорное ядро – первое процессорное ядро с принципиально новой (пост-неймановской) мультиклеточной архитектурой. Предназначено для решения задач управления и циф-

ровой обработки сигналов в приложениях, требующих минимального энергопотребления и высокой производительности. От фон-неймановской модели мультиклеточная архитектура отличается непосредственным указанием информационных связей между операциями и, соответственно, снятием требования упорядоченного размещения описаний операций в программе. Эта неупорядоченность делает ненужными все методы (суперскалярность, широкое командное слово, суперконвейер, предсказание переходов и т. п.), резко усложнявшие проектирование процессора и инструментальных программных средств.



» Рубрику готовил
АРТЕМ ЗОРИН

RIW//12

RUSSIAN
INTERNET
WEEK

РАЭК

17-19
ОКТАБРЯ
2012 ГОДА

WWW.RUSSIANINTERNETWEEK.RU

НОВАЯ МЕТЛА...

Судьба НПП под вопросом

Министр связи РФ намерен отказаться от национальной программной платформы на базе Linux?

С приходом нового министра Николая Никифорова политика Минкомсвязи может сильно поменяться: не исключено, что государство откажется от создания национальной программной платформы в пользу продуктов Microsoft.

Илья Массух, заместитель министра связи РФ, сообщил газете «Ведомости» о решении покинуть свой пост из-за разногласий с новым министром связи, который намерен отказаться от создания национальной программной платформы на базе Linux и свободного ПО и заключить глобальное соглашение с Microsoft для всех органов исполнительной власти, поставив таким образом все органы государственного управления в зависимость от одной иностранной компании. Мотивом заклю-

чения соглашения с Microsoft называется повышение показателей эффективности госзакупок, из-за возможности получения специальной цены, существенно ниже розничной стоимости продажи Windows.

Такое сотрудничество лишь позволит решить ряд текущих проблем, но может завести в тупик в долгосрочной перспективе. Массух поясняет: «Глобальное соглашение с вендорами для госорганов – это быстрая победа, которая ни к чему не ведет. Сейчас Россия – одна из немногих стран в мире, способных делать собственные операционные системы. А заключение таких контрактов с иностранными компаниями ставит крест на отечественной индустрии, ведь такие гиганты, как Google и Microsoft, не заинтересованы в развитии российских программистов».

► Илья Массух – один из инициаторов создания НПП.



Николай Никифоров, новый министр связи, отреагировал на статью сообщением в Twitter, что не стоит буквально верить всему, что пишут газеты. На вопрос, нет ли в планах сворачивания НПП, министр ответил: «Нет, но нужно очень четко понять, чего же мы все хотим от НПП, на что и зачем тратим деньги налогоплательщиков».

В ЗАЩИТУ СВОБОДЫ

Новости Старого Света

Официальные органы Евросоюза активизировались.

За последние несколько недель различные ведомства единой Европы успели сделать немало хорошего.

Во-первых, Суд Европейского союза вынес решение в деле Used Soft GmbH против Oracle International Corp, подтвердившее правомерность перепродажи лицензий на неиспользуемое ПО. Разобрав претензию компании Oracle, считавшей незаконным бизнес немецкой компании Used Soft GmbH (скупка лицензий на неиспользуемое ПО у клиентов Oracle и дальнейшая их перепродажа по сниженным ценам), суд принял сторону ответчика и постановил, что лицензию на ПО нельзя рассматривать в привязке к установленной копии продукта, поэтому «подержанная» лицензия может передаваться в другие руки и применяться для приложений, копия которых была загружена из Интернета. Примечательно, что решение вынесено, несмотря на явный запрет в лицензионном соглашении Oracle передачи лицензии третьим лицам и разрешение применения ПО только для внутренних нужд клиента.

Во-вторых, Парламент Европейского союза отклонил законопроект АСТА

(Международное торговое соглашение по борьбе с контрафакцией), грозящий ущемлением гражданских свобод и права свободного доступа к информации. Против принятия АСТА проголосовало 478 человек, за – 39, и 146 воздержались.

Вопрос о совместимости законопроекта с законами Европейского Союза пока открыт, но отклонение его парламентариями означает, что сторонникам АСТА придется заняться написанием нового закона. Ранее соглашение было ратифицировано Еврокомиссией и парламентами стран – участниц Евросоюза. Для вступления закона в силу оставалось утвердить проект в Европарламенте, который, к счастью, прислушался к пожеланиям общественности. Помимо Евросоюза, указанный проект активно продвигается в США, Канаде, Японии, Южной Корее, Австралии, и включает план создания подобной системы в развивающихся странах.

В-третьих, Европейский Фонд СПО сообщил о решении Верховного суда Евросоюза по наложению на компанию Microsoft штрафа в размере 860 млн евро за использование своего положения

на рынке ПК, близкого к монопольному, для подавления конкуренции в области серверных систем. Изначально штраф был назначен Еврокомиссией еще четыре года назад; нынешнее решение суда отвергло апелляцию и настояло на исполнении ранее вынесенного решения. Однако размер штрафа, присужденного в 2008 году, был снижен на 39 млн евро.

После штрафов в 2004 и 2006 годах и серии неудачных апелляций, Microsoft обеспечила документацию, но стала распространять ее по непомерно высокой цене, что в 2008 году привело к наложению еще одного штрафа. Компания Microsoft подала апелляцию, суть которой сводится к заявлению, что так как информация по обеспечению совместимости с релизами подпадает под множество патентов и является инновационной, для доступа к информации компания имела право взимать с конкурентов значительный размер денежных средств. В итоге суд определил, что Microsoft продолжает отказывать разработчикам свободного ПО в полном доступе к информации для оперативного обеспечения совместимости.

softline®

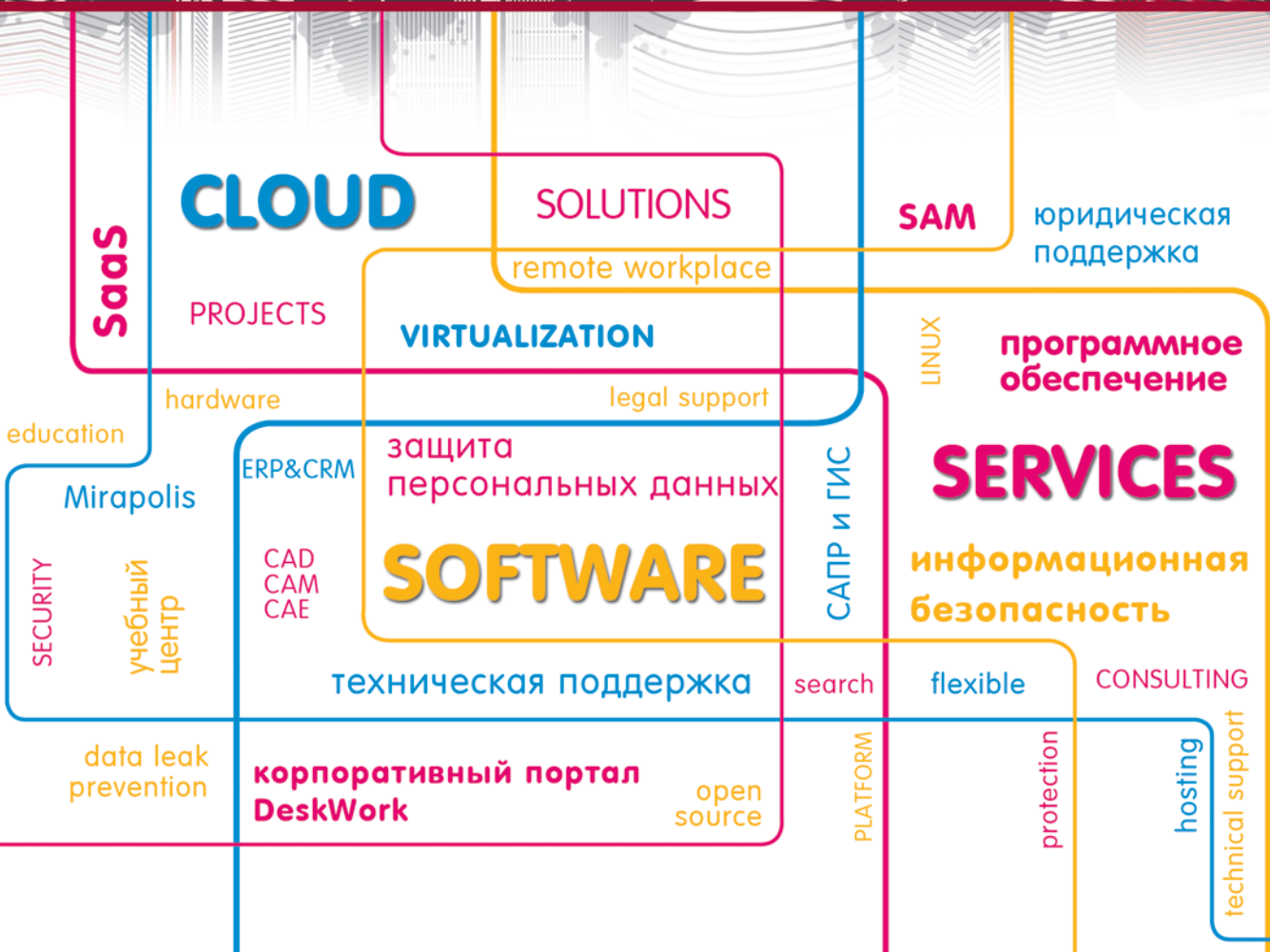


Services

Software

Cloud

ИТ-архитектура вашего бизнеса



+7 (495) 232-00-23

www.softline.ru

info@softline.ru

ЛЕД ТРОНУАСЯ

Новая мобильная платформа

Проект Mozilla представил мобильную ОС Firefox OS и первые телефоны на ее основе.

Мozilla Foundation представил новую мобильную операционную систему Firefox OS. Изначально проект развивался под именем Boot to Gecko (B2G), но в июле было заявлено, что новая ОС будет поставляться под узнаваемым брендом Firefox, с целью привлечь пользователей к смартфонам – новичкам на рынке. ОС целиком базируется на открытых web-стандартах, предоставляя разработчикам приложений основанные на HTML5 компоненты для задействования всех возможностей аппаратных устройств.

Первые модели смартфонов, укомплектованных новой ОС, выпустят компании TCL Communication Technology (бывшая Alcatel) и ZTE, на аппаратной платформе Qualcomm Snapdragon, на базе которой производятся многие современные смартфоны с Android. Первые смартфоны с Firefox OS поступят в продажу в начале 2013 года под брендом Vivo, принадлежащим испанской компании Telefónica. Точные данные о стоимости не сообщаются, но упоминается, что цена, с учетом намерения форсировать выход на рынок новичка, будет более чем привлекательной.

Mozilla гарантирует полную открытость проекта и независимость разработки от отдельных производителей. Разработчики проекта намерены передать эталонную реализацию Web API, разработанного в процессе создания Firefox OS, в организацию W3C для ут-

верждения в качестве web-стандарта. При этом Web API по возможности базируется на уже принятых стандартах, расширяя их в необходимых направлениях. Платформа изначально оптимизирована для начального сегмента смартфонов и не имеет излишних промежуточных прослоек, что позволяет мобильным операторам подготовить продукты, предоставляющие богатый набор функций по цене дешевых телефонов начального ценового уровня.

Инициатива по созданию новой ОС и формированию связанной с ней полностью открытой экосистемы разработки мобильных приложений на базе технологий HTML5 получила скорый отклик у пред-

«Mozilla гарантирует полную открытость проекта.»

ставителей индустрии – о своем участии в продвижении и развитии Firefox OS заявили такие известные операторы связи, как Deutsche Telekom, Etisalat, Smart, Sprint, Telecom Italia, Telefónica и Telenor. Они будут способствовать продвижению нового участника рынка мобильных систем, предоставляя для этого свои ресурсы.

Развиваемая в рамках проекта Firefox OS мобильная платформа базируется на идее использования браузерного окружения вместо рабочего стола. В отли-

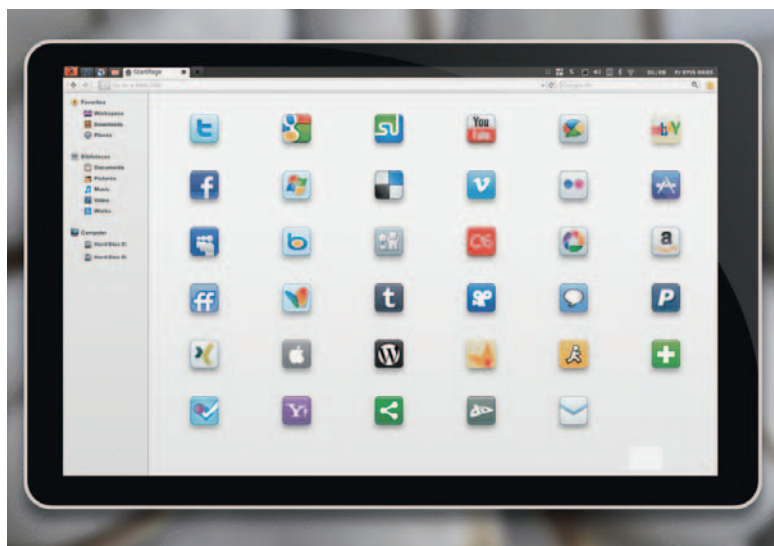
чие от ChromeOS, платформа Firefox OS ориентирована прежде всего на мобильные устройства и предоставляет расширенный Web API для создания специализированных мобильных web-приложений, использующих все аппаратные возможности современных телефонов. В качестве основы используется ядро Linux и низкоуровневые компоненты из платформы Android. Вместо виртуальной машины Dalvik для запуска приложений задействован web-стек Mozilla.

Для распространения обновлений будут применяться уже опробованные проектом Firefox технологии. Для решения вопросов противодействия попыткам распространения вредоносного ПО будет использован опыт поддержки каталога дополнений для Firefox. Приложения, как бесплатные, так и платные, будут распространяться через магазин приложений Mozilla Marketplace. Для организации идентификации пользователей будет задействован сервис Mozilla Persona, основанный на технологии BrowserID.

Пользовательский интерфейс платформы будет сформирован из набора web-приложений Gaia. В его состав войдут web-браузер, калькулятор, календарь-планировщик, приложение для работы с web-камерой, адресная книга, интерфейс для осуществления телефонных звонков, клиент электронной почты, система поиска, музыкальный плеер, программа для просмотра видео, интерфейс для SMS/MMS, конфигуратор, менеджер фотографий, рабочий стол и менеджер приложений с поддержкой нескольких режимов отображения элементов (cards и grid).

Вместо предоставления доступа к реальной файловой системе, программы будут ограничены внутри виртуальной ФС, изолированной от основной системы.

Созданные с использованием Web API программы смогут работать не только в окружении Firefox OS, но в любом поддерживающем стандарты web-стека. В итоге, будет сформирован набор стандартов для создания универсальных мобильных web-приложений, способных обеспечить функциональность, свойственную обособленным мобильным стекам, как правило, контролируемым отдельными производителями (Android, Apple iOS и Windows Phone).



➤ Демонстрация работы Firefox OS на Samsung Galaxy S 2.

Red Hat Enterprise Linux

предоставляет вам производительность,
масштабируемость, безопасность и надежность,
ранее доступные только на очень дорогих платформах

Самая популярная в мире
Linux платформа для бизнеса

Обеспечивает высокую
производительность, надежность,
масштабируемость и безопасность

Сертифицирована ведущими
производителями оборудования
и разработчиками ПО



Совместима с широким спектром
оборудования от рабочих станций
до серверов и мэйнфреймов

Обеспечивает одинаковые
условия работы приложений
при использовании в физической,
виртуальной и облачной средах

Пользователи RHEL экономят на оборудовании,
лицензиях на программное обеспечение
и эксплуатационных расходах



ГНУ/Линуксцентр — Linux-эксперт для вашего бизнеса

- Premier Business Partner компании Red Hat
- 12 специалистов по разработке и внедрению, сертифицированных компанией Red Hat
- Более 100 клиентов, использующих Red Hat
- 10 лет на рынке

Red Hat — ведущий серверный дистрибутив Linux

- Более 15 лет промышленного использования
- Свыше 80% рынка корпоративного Linux по данным CIO Insight
- 5 лет среди лучших вендоров
- Выгодная совокупная стоимость владения (TCO)
- Поддержка в течение 10 лет

Специальное предложение!

Закажите Red Hat Enterprise Linux в ГНУ/Линуксцентре и получите в подарок книгу «Полное руководство пользователя Red Hat Enterprise Linux»



Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

Linux-эксперт для вашего бизнеса. www.linuxcenter.ru

Linux  center

ЛИНУС РАЗБУШЕВАЛСЯ

Пристыдили производителя видеокарт № 1

Линус Торвалдс резко раскритиковал отношение компании NVIDIA к Linux-сообществу.

Выступая в Хельсинкском университете, Линус Торвалдс [Linus Torvalds] крайне негативно высказался насчет компании NVIDIA, в ответ на просьбу студентки прокомментировать ситуацию с поддержкой в Linux технологии NVIDIA Optimus (возможность на лету переключаться между встроенной энергоэффективной видеокартой на базе GPU Intel и дискретной картой NVIDIA). Его эмоциональное заявление сопровождалось не совсем приличным жестом в камеру. Линус назвал NVIDIA худшей компанией из производителей оборудования, с которой он имел дело, пояснив, что NVIDIA даже не пытается сотрудничать с Linux-сообществом или хоть как-то помочь разработчикам открытых драйверов, держа в секрете спецификации и отстраняясь от участия в создании открытых драйверов. Линус также пояснил, что по своему отношению к Linux-сообществу и неприятию открытого кода NVIDIA является исключением из правил: остальные компании охотно идут на сотрудничество.

Жест создателя Linux не остался без внимания компании NVIDIA. Длинный пресс-релиз производителя видеокарт сводится к следующему: «Предоставление качественной поддержки всех наших видеоакселераторов на многих платформах для всех наших потребителей является одной из наших приоритетных задач. Мы всегда будем поддерживать сообщество Linux». Параллельно с этим Стивен Уоррен [Stephen Warren], разработчик



► Темпераментный жест создателя Linux при трансляции несколько завуалировали.

ядра Linux из компании NVIDIA, развернул в списке рассылки Kernel Summit дискуссию о том, что компании следует сделать для улучшения поддержки ядра Linux. Стивен указал, что в своем комментарии Линус Торвалдс подчеркнул, что вклад NVIDIA в разработку ядра в основном связан с обеспечением поддержки

«Остальные компании охотно идут на сотрудничество.»

Tegra SoC. В связи с этим, сотрудникам компании интересно, в каких еще областях NVIDIA может участвовать в разработке ядра, за исключением большого вопроса поддержки связанного с графикой оборудования, код для которого компания пока не может открыть. **LXF**

даже невзирая на шансы получения столь немалого дохода. К тому же портирование двоичного драйвера на MIPS потребовало бы затрат в миллионы долларов (в данный момент драйвер поддерживает только x86 и x86_64). В итоге, NVIDIA лишилась крупного заказа. Это наводит на мысли о том, что слова Линуса Торвалдса были справедливы...

Новости короткой строкой

► Финские исследователи обнаружили вредоносное ПО, поддерживающее атаку на системы с Windows, Mac OS X и Linux. Источник: www.f-secure.com

► Компания Google собирается опубликовать исходные тексты недавно выпущенной платформы Android 4.1. Источник: <https://groups.google.com>

► 8 июля в возрасте 31 год погиб в Бразилии Евгений Додонов – талантливый разработчик и замечательный человек, работавший в последнее время в компании Intel над разработкой драйверов видеокарт для Linux. Источник: www.saocarlosagora.com.br

► Группа из шестерых бывших сотрудников компании Nokia основала стартап Jolla, нацеленный на разработку и продвижение на рынок новых смартфонов на базе Linux-платформы MeeGo, от которой Nokia отказалась. Источник: <http://venturebeat.com>

► Ришар Фонтана [Richard Fontana] основал проект GPL.next, с целью предпринять попытку создания ответвления лицензии GPLv3, более совместимой с либеральными лицензиями по типу лицензии Apache. Источник: www.infoworld.com

► Компания Mozilla объявила о своих намерениях прекратить официальное участие в разработке почтового клиента Mozilla Thunderbird. Источник: <http://pastebin.com>

► Компания 1C приступила к тестированию ознакомительной версии платформы для автоматизации бухгалтерского учета и деятельности предприятия «1С:Предприятие» 8.3 для Linux. Источник: www.1c.ru

► Фонд свободного ПО отметил пятилетие с момента выхода лицензии GPLv3, внесшей такие новшества, как средства противодействия тивоизации (привязки программного обеспечения к оборудованию) и заключению дискриминационных патентных соглашений. Источник: www.fsf.org

► Группа организаций, отстаивающих гражданские свободы, а также компании – разработчики ПО и популярные сайты инициировали новый совместный проект под названием «Лига защиты Интернета». Источник: www.internetdefenseleague.org

Линус, ты прав

Китайская компания, разработавшая MIPS-процессоры Loongson, изучала возможность по приобретению очень крупной партии GPU, порядка 10 миллионов устройств.

В случае с NVIDIA проблема была связана с тем, что двоичный драйвер не поддерживает архитектуру MIPS. Отмечается, что компания NVIDIA наотрез отказалась предоставить исходные тексты драйвера,



АЛЕКСЕЙ ФЕДОРЧУК
Тэг «сарказм»
по умолчанию,
смайлики по вкусу.

Нативная ZFS — Linux'у!

Ровно 2 года назад (LXF133) я писал о победе инженерной мысли (воплощенной Брайаном Беллендорфом) над юридическим крючкотворством сочинителей лицензий — методе прикручивания поддержки ZFS на уровне ядра Linux, получившем развитие в проекте ZFS on Linux. Но «мейнстримовые» дистрибутивы не торопились включать в свои системы поддержку нативной ZFS даже опционально. Пионером выступил Sabayon — не очень распространенный и довольно «хулиганский» отпрыск Gentoo: в его 9-й релиз включена возможность задействовать ZFS на стадии инсталляции (разумеется, не по умолчанию).

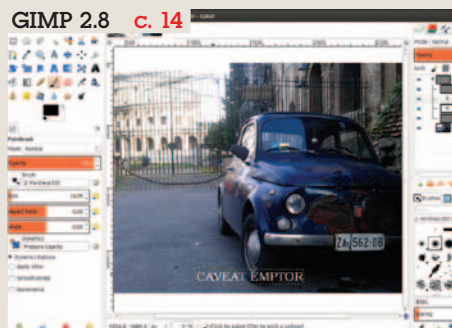
В принципе, никто не обделен: разработчики ZFS on Linux предлагают его пользователям Ubuntu в виде пакета, помещенного в PPA-репозиторий и готового к употреблению. А в качестве заботы о прочих предоставляют, кроме неперменных исходников, также подробные инструкции по сборке пакетов Deb- и RPM-форматов.

Не знаю, воспользовались ли этой любезностью майнтейнеры иных фронтальных дистрибутивов, но вот майнтейнеры openSUSE — воспользовались. И ныне пакет поддержки ZFS со всем сопутствующим инвентарем пользователи этого дистрибутива могут легко отыскать в системе Open Build Service и установить «в один клик». Не без подводных камней, заставляющих вспомнить родственников сочинителей лицензий, но в принципе такая возможность есть. А что они за это получают — поговорим в другой раз.

alv@posix.ru

Сегодня мы рассматриваем:

- GIMP 2.8** 14
Один из краеугольных камней настольного Linux наконец-то вернулся на эти страницы после нескольких лет отсутствия — теперь с улучшенной обработкой текста, упрощенным интерфейсом и кучей правок за кадром. Откройте бесплатную альтернативу *Photoshop* уже сегодня!
- Pianoteq 4** 18
Некоторые люди любят варьировать количество соударений молоточка со струнами синтезированного пианино, чтобы услышать, как это влияет на тон. Подобным типам *Pianoteq* понравится. Кстати о тоне: за эталон звучания был принят берлинский концертный рояль.
- Miro 5** 16
Все ваши медиа-файлы, будь то в Интернете, локальном хранилище или вашем личном облаке, воспроизводятся с помощью одного приложения... звучит как мечты безумца, но Долевой фонд культуры воплотил их в жизнь. Мы предпочитаем Кандинского, но и *Miro* дадим шанс.
- Rosegarden 12.04** 19
Если вам нужна рабочая станция цифрового аудио, да чтобы и имя у нее благоухало, выбирайте *Rosegarden*, и не ошибетесь — невзирая на слегка суетливый интерфейс. Где розы, там и шипы... зато падающая программа стала значительно реже и подружилась со звуковым сервером.
- Mageia 2** 17
Ответвление от Mandriva, разрабатываемое сообществом, ждет новый этап конкуренции, поскольку Mandriva теперь тоже ответвление от Mandriva, разрабатываемое сообществом. А вот разработчики достигнут больших высот.
- Runtu Gnome 12.04** 20
Разработчики отечественного варианта Ubuntu предпочитают не отклоняться от курса: русифицировав интерфейс окружения Gnome, на KDE они распылять силы не стали. Учите английский, господа пользователи!



Создайте в **GIMP** дивное царство воображения. Или хотя бы уберите красные глаза с фоток.



Мы все еще не уверены, как произносится **Mageia**, но зато выучились писать это слово без ошибок.

Сравнение: Открытые социальные сети

Diaspora



Friendica



Identi.ca



Qaiku



Pidder



При помощи этого Сравнения хронофагов вы сможете раззвонить на весь мир, что вы ели на завтрак, не дав при этом Марку Цукербергу приторговать вашей жизнью, чтобы справиться очередную курточку.

GIMP 2.8

GIMP – один из самых знаменитых открытых проектов, но развивается он со скоростью подвижки пород. **Ник Вейч** изучает плод четырехлетних трудов...

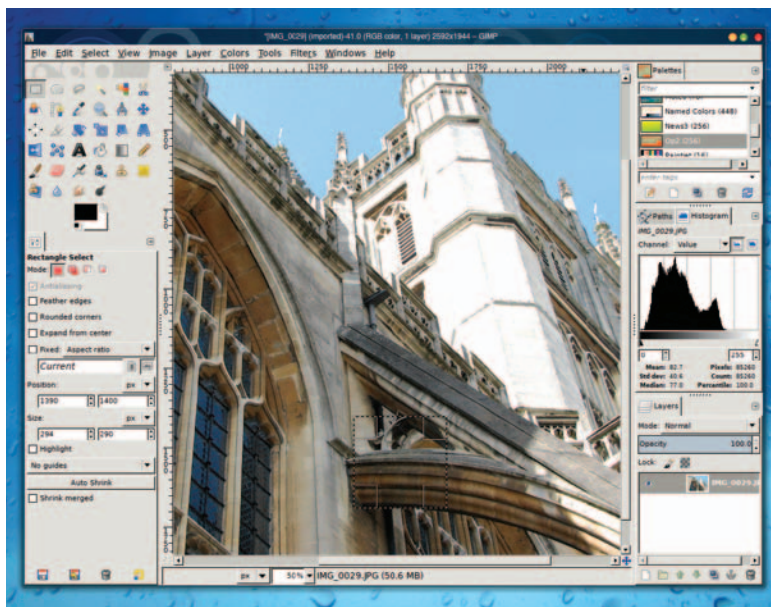
Вкратце

» Полнофункциональный растровый графический редактор, подобный *Photoshop*. См. также *Krita* (www.krita.org), но тот ориентирован скорее на рисование, чем на редактирование.

Может показаться, что *GIMP 2.6* у нас уже лет сто, хотя на самом деле – чуть больше четырех. Последний стабильный релиз принес ему мощь движка GEGL – абсолютно новой графической подсистемы, обещавшей, наконец, освободить ПО от одного из самых досадных ограничений (глубины цвета и цветовых моделей), одновременно переведя основные графические преобразования на более разумную базу.

Поскольку главный капремонт производился внутри, версия 2.6 не смотрелась прорывом (а прорыв был!), и надежд на столь затянувшийся выход версии 2.8 возлагали немало.

И вот она наконец-то обретена; однако некоторые реформы уже кажутся довольно противоречивыми. Если вы просто успешно обновились с 2.6 до 2.8, более всего вас озадачит новый механизм сохранения. Принято решение ограничить обе опции, Сохранить и Сохранить как,



» Главное новшество версии 2.8 – однооконный интерфейс, объединивший все панели и вкладки.

«Надежд на затянувшийся выход версии 2.8 возлагали немало.»

«родным» для *GIMP* форматом *.xcf*, чтобы избежать потерь при случайном сохранении в другом формате. При этом, как ни странно, диалоговые формы по-прежнему

предлагают «выбрать тип файла по расширению», хотя выбора и в помине нет. Оно, конечно, неплохо, если изображение создается прямо в *GIMP*; но если вы, например, загружаете файл *.png*, то, скорее всего, захотите, слегка подкорректировав его, нажать **Ctrl+S** и получить обратно в первоизданном виде.

Вместо этого приходится обращаться либо к функции Экспортировать, ли-

бо к Перезаписать файл в главном меню. Что выглядит непродуманным: ведь большинство пользователей работают с файлами других форматов, и заставлять их скакать в *.xcf*, а затем обратно – не лучшая идея. Готовы поспорить, что пользователи скоро запросят возврата к старому.

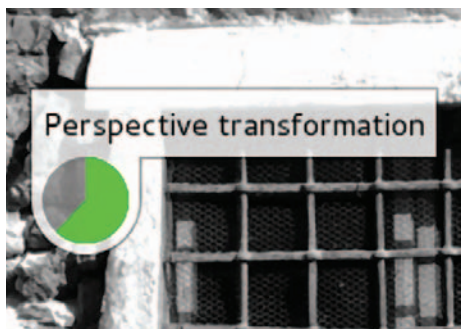
Но в целом, новшеств много. Возможно, не настолько сногсшибательных, чтобы вострубить о победе функционала *GIMP* над *Photoshop* (а ведь такое бывало), но порой именно мелочи, делающие комфортнее повседневную работу с приложением, радуют гораздо больше.

Свойства навскидку



Трансформация в рамке

Поможет преобразовать фрагменты неправильной формы – нечто вроде гибрида инструмента Перспектива и фильтра искривления.



Cairo

Новый прогресс-индикатор на экране, как и многое другое, появился благодаря графической библиотеке Cairo.

Поговорим о тексте

Отдельной похвалы здесь достоин новый способ редактирования текста. Раньше текст приходилось вводить в дурацком маленьком окошке, по ходу мучительно представляя, как это потом впишется в рисунок. Теперь же появился инструмент Текст, и работа перешла прямо на холст. По-прежнему есть небольшое диалоговое окно Стиль, где задаются атрибуты текста (совет: сначала напишите, а затем форматировать – настроив стили, тяжело потом активировать набор в прежнем формате).

Этим улучшением (действительно серьезным) *GIMP* обязан спонсорской под-

держке Summer of Code – проекта, спонсируемого Google, и деньги потрачены не зря! Щелкните ради праздника на какой-нибудь рекламе за это...

Самым желанным новшеством стал, конечно же, однооконный интерфейс (ранее доступный только через специальные расширения), вместо прежних разбросанных по экрану фрагментов. Причем реализовано это очень разумно, простым переключателем в меню Окна, и будьте уверены, ваш выбор сохранится до следующего раза. Это отражает общий характер современных реализаций настольных систем – с лишними окнами тяжело управляться в «модерновых» рабочих столах типа Unity и Gnome 3, где все неактивные окна максимально сворачиваются.

Единое окно – также шаг навстречу эпизодическим пользователям и новичкам, поскольку позволяет просто повозиться с бесчисленными палитрами и эффектами, особо не ломая головы.

Тем же, у кого интерес более глубокий, в особенности любителям работы с многослойными изображениями, наверняка понравится новая возможность группировки слоев. Теперь можно создать целое дерево слоев (помещая группы внутри других групп), что значительно облегчает управление этими сложными конструкциями. Представьте, что

«GIMP, без сомнения, незаменимый компонент настольного ПО.»

вам нужно наложить ряд различных объектов – и если каждый находится в составе группы, включающей соответствующие слои (картинку, эффекты, текстуры), разобраться с ними будет гораздо проще; да хотя бы размер окна Слои останется управляемым.

Кстати о группировке: для кистей, шаблонов и текстур теперь поддерживается режим «тэгов». Суть в том, что можно пометить кисти любым количеством тэгов, таких как «натуральные» и/или «текстура», а затем при помощи тэгов фильтровать их отображение; и когда вам понадо-

Анонс 2.9

Естественно, выход 2.8 знаменует начало работы над следующим релизом. Так и есть, скоро уже можно будет приложить руки к созданию кода для новой версии, в которой предполагается использование разной глубины изображения, а также расширение поддержки разных форматов файлов и потоков операций. Но не ждите, что этот релиз сразу же будет стабильным! <http://wiki.gimp.org/index.php/Roadmap>

Исследуем интерфейс GIMP

Добавление текста стало удобнее: это теперь можно делать прямо на холсте, не мучаясь с неуклюжими всплывающими окошками, как в прошлых версиях.

Однооконный режим

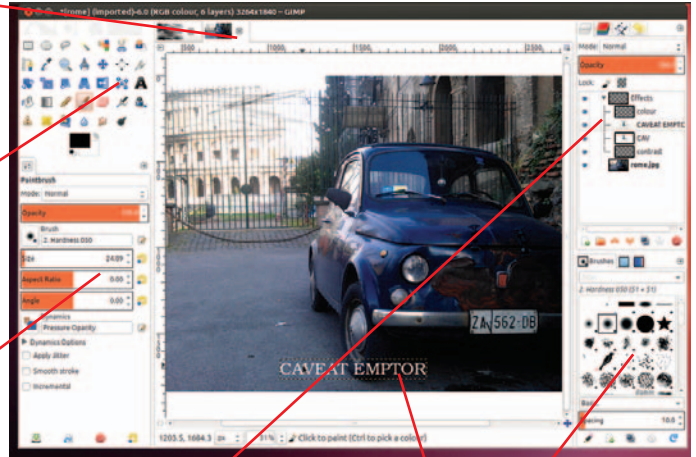
В этом режиме все изображения открываются в виде вкладок.

Трансформация в рамке

Этот новый инструмент поможет преобразовать элементы изображений неправильной формы.

Свойства инструмента

В новом переработанном виде, с ними стало проще работать на планшетах, да и визуальное представление настроек улучшилось.



Группировка слоев

Помогает работать с составными изображениями, оперируя с группами слоев.

Инструмент Текст

Благодаря его усовершенствованию, делать надписи на рисунках стало проще.

Тэги

Добавляйте их к своим кистям и структурам, и их будет легче найти.

бится именно «натуральная» кисть, поиск не затянется. Это поможет справиться с многообразием инструментов для рисования: не перебарщивать с тэгами не стоит, иначе взаимодействие с ними самими будет трудоемким (кстати, приписывание тэгов возможно только при отображении объектов в виде списка).

Не все программные улучшения в этой версии связаны с организацией работ: есть и совершенно новый инструмент! Трансформация в рамке [Cage Transform] – это некий гибрид инструмента Перспектива и фильтра искривления. Как и в первом, вы можете менять форму фрагмента изображения, перетаскивая рукоятки углов, но разница в том, что фрагмент позволяет выделять произвольным многоугольником.

Последний штрих

GIMP, без сомнения, незаменимый компонент настольного ПО. Не будь его, многие люди не пустили бы Linux на свои компьютеры, и трудно представить, что кто-то ни разу им не пользовался. И не так уж важно, захотите ли вы обновляться до новой версии или нет – скорее всего, захотите.

Все нововведения себя оправдывают, да и побаловаться с новыми инструментами тоже интересно. А всякие глобальные перемены, в виде поддержки разной глубины цвета и цветовых моделей, могут подождать и до следующего раза. Де-

ло же не в отсутствии энтузиазма или преданности разработчиков, ведь некоторые из них – как Митч Наттерер [Mitch Natterer] и Ойвинд Колас [Óyvind Kolás] – потратили на проект годы.

Просто, в отличие от других программ, вроде разных инструментов виртуализации, баз данных или web-браузеров, проект GIMP не имеет постоянного источника доходов или какой-либо спонсорской помощи (помимо усилий Google в Summer of Code), поэтому вряд ли станет развиваться быстрее. Но это и вправду потрясающая программа, и если вы хотите сделать GIMP еще лучше, возможно, стоит задуматься о том, чем можете помочь именно вы. **LXF**

LINUX FORMAT Вердикт

GIMP 2.8

Разработчик: Команда GIMP
Сайт: www.gimp.org
Цена: Свободно под GPLv3

Функциональность	8/10
Производительность	9/10
Простота использования	8/10
Документация	7/10

» Явное достижение, достойное аплодисментов, но работы впереди еще много.

Рейтинг 8/10

Miro 5

Завидуете своим друзьям, использующим Windows и огромную медиатеку от iTunes? **Джонатан Робертс** предлагает альтернативу – *Miro*.

Вкратце

» Медиаплеер-плюс-TV Times. См. также: *Banshee*, *Rhythmbox*, *Amarok*.

Вам надоело, что к своим любимым медиа-файлам вы вынуждены обращаться через сотню разных приложений, и у каждого свои хитрости? Разработчикам *Miro*, по всей видимости, да, потому что то, что они создали, можно назвать iTunes на стероидах.

В основе своей, это вполне приличный локальный медиа-плеер. При первом запуске *Miro* вы можете парой щелчков добавить в библиотеку всю музыку и видео, хранящуюся у вас на диске. Тогда при входе в раздел Музыка вы будете иметь доступ к файлам в обычном формате, а вот в разделе Видео есть более интересные решения.

Вместо перечня маленьких заголовков перед вами открывается длинный список с большими значками, краткими описаниями и быстрым доступом к кнопке Удалить. И в этом один из ключевых принципов применения *Miro*: для подпис-

«Возможен доступ к медиа из любого источника, имеющего сайт.»

чиков видеоподкастов. А поскольку век подкастов короток, кнопка удаления для этих больших устаревших файлов просто незаменима.

Продолжая тему поддержки функции видеоподкастов, отметим, что здесь *Miro* просто нет равных. Сразу после запуска перед вами открывается Miro Guide. Он снабжен приятным интерфейсом и сообщает вам о недавно добавленных, рекомендуемых и «лучших» видео.

Сильный упор делается на материалы из независимых и уникальных источников (доступ к ним предоставляется за счет Participatory Culture Foundation), и здесь можно обнаружить даже такое, о существовании чего вы и не подозревали. Мы сами видели в топах видео от NASA, Bernerd Center for Research on Women и Psychedelic Salon.

Причем изрядная часть этого было в HD-качестве – если вы как раз искали, на чем бы испытать ваш новый телевизор, *Miro* просто обязан быть среди самых первых ваших источников. Любой из роликов в *Miro* скачивается при помощи встроен-



» *Miro* ориентирован на независимые и уникальные медиа-ресурсы, а значит, у вас всегда есть шанс наткнуться на что-то интересное и необычное.

ного клиента на основе *Bittorrent*. Тех, кто сомневается в надежности данной функции, спешим заверить, что у нас все загрузилось очень быстро; это, пожалуй, самый интуитивный клиент *Bittorrent*, который нам доводилось использовать – и с огромным запасом абсолютно легального контента!

Внешние ресурсы

Через *Miro* возможен доступ к медиа из любого другого источника, имеющего сайт. По умолчанию в нем уже есть YouTube, Hulu (работающий только в США) и MP3-магазин Amazon, но вы легко можете добавить и свои.

Проверка этих возможностей показала, что это немногим более чем встроенные web-страницы, так что при попытке установить iPlayer мы с изумлением прочли «Извините, данный тип устройства не поддерживается» – видимо, происки BBC, решили мы. Также мы потерпели неудачу с 4oD, который потребовал обновить Flash-плеер.

Все эти ресурсы связаны единой функцией поиска, осуществляемого на YouTube, Daily Motion, Metavid, Mininova, Bliр и Google Video. Идея великолепная, но реализация ее вышла медленной и не слишком результативной. Широкий функционал *Miro* дополняется также поддержкой практически любых портативных медиа-устройств и мобильных телефонов на Android, позволяя вам копировать ме-

диа на ваш любимый носитель или с него. Возможность скачивать приложения из магазина Google и синхронизировать их с вашим телефоном весьма приятна, но нас значительно больше впечатлило то, что *Miro* автоматически конвертирует видео в нужный формат. И не нужно больше гадать, какие параметры задать в *ffmpeg*, чтобы потом это нормально воспроизводилось телефоном.

Короче, *Miro* – хороший медиа-плеер, а Miro Guide и надежная аппаратная поддержка – его главные козыри. Ну, а многие другие функции суть хорошие идеи, пока что требующие доработки, поскольку они загромождают интерфейс и ухудшают общее впечатление. **LXF**

LINUX FORMAT **Вердикт**

Miro 5

Разработчик: Participatory Culture Foundation
 Сайт: <http://getmiro.com>
 Цена: GPL v2 или позже

Функциональность	8/10
Производительность	7/10
Удобство использования	7/10
Документация	8/10

» Менее известный, чем его конкуренты, Miro заслуживает внимания хотя бы потому, что у него столько интересных функций.

Рейтинг 7/10

Mageia 2



Второй по счету релиз этого дистрибутива выглядит многообещающим, но не идеальным, считает **Бен Эверард**.

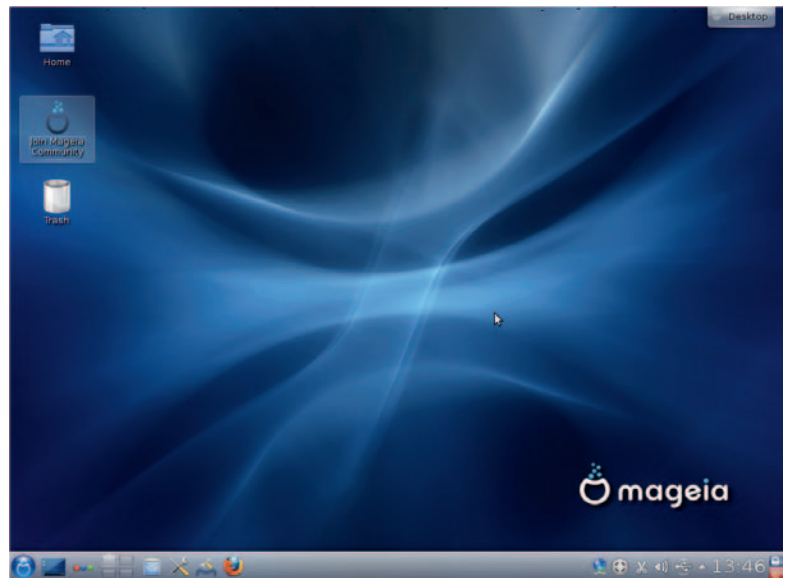
Вкратце

» Mageia стремится сделать процесс настройки простым и без всякой командной строки. См. также PCLinuxOS, Rosa и Ubuntu.

Мandriva был первым дистрибутивом, открывшим дорогу в Linux непрофессионалам, и как таковой стал идейным вдохновителем Ubuntu, Mint и всех остальных, что само по себе примечательно. Так что когда ядро разработчиков в 2010 году отделилось от Mandriva, чтобы начать работы над новым дистрибутивом, мы все с нетерпением ждали, что же у них получится. И вот перед нами второй релиз проекта, Mageia 2, и это здорово.

Окружение рабочего стола выглядит как плод союза KDE и Unity. В отличие от последнего, здесь нет Dash и HUD, но есть панель, которая выглядит компромиссом между обоими. Вместо названия окна, на панели задач отображается только иконка, а число открытых окон обозначается треугольничками. Это весьма помогло, когда мы тестировали дистрибутив на виртуальной машине, поскольку сэкономило место на экране. Правда, после установки системы такая необходимость отпала, и экономия показалась излишней.

Можно переключиться на традиционный менеджер задач KDE, щелкнув на опции Разблокировать виджеты, затем правой кнопкой – на панели, и выбрав Удалить только этот значок – в Менеджере задач. Затем снова щелкнуть правой кнопкой на панели, и далее Настроить панель > Добавить виджеты и выбрать Менеджер задач. Еще одно отличие от стандартного KDE – классическое меню. Если же оно вам не по нраву, можете изменить стиль меню, правой кнопкой щелкнув по Переключиться на... Стиль.



» Обновиться к Mageia – естественный выбор для тех пользователей Mandriva, кто беспокоится о будущем дистрибутива.

В Mageia Control Center много и других рабочих столов, но ни один из них не имеет столь тонкой настройки.

Под контролем

Mageia Control Center разбит на восемь разделов: Программы, Оборудование, Сеть и Интернет, Система, Совместное использование сети, Локальные диски, Безопасность и Загрузка. Каждый из них предоставляет простейшие функции администрирования, годные для настройки одной машины или небольшой сети. Но при нетипичной установке или для особо тонкой настройки все же дубовато.

Менеджер пакетов прост и понятен, как и сам рабочий стол, но до лучших образцов пока не дотянул. Здесь нет, например, способа оценить программу или написать отзыв, да и поиск довольно примитивен.

Одна из самых полезных функций Control Center – Родительский контроль. Его можно установить, задав либо черный список (блокируемые сайты), либо белый список (те, что будут доступны) Но несмотря на то, что регулярно обновляемых черных списков существует много, Mageia Control Center позволяет вводить адреса только по одному, отчего эта функция становится практически бесполезной.

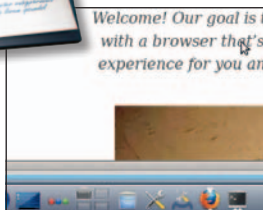
В наборе программ сюрпризов немного. Есть офисный набор LibreOffice 3.5, му-

зыкальный плеер *Amarok*; разработчики умудрились втиснуть даже последнюю версию *GIMP* (2.8).

Mageia – открытый дистрибутив, и приятно видеть на рабочем столе иконку с приглашением присоединиться к сообществу. Нажав на нее, вы попадаете на сайт Mageia, где вам предлагаются разные роли. Более всего востребована нетехническая аудитория – для написания документации, юридической помощи, перевода. В настоящий момент ситуация с документацией несколько удручает, но мы надеемся, что вскоре все выправится. **LXF**



Свойства навскидку



Менеджер задач

Менеджер задач использует только иконки, экономя место на небольших экранах.



Control Center

Поможет настроить вашу систему, но на многое рассчитывать не приходится.

LINUX FORMAT Вердикт

Mageia 2

Разработчик: Команда Mageia

Сайт: www.mageia.org

Цена: Бесплатно

Функциональность	8/10
Производительность	8/10
Удобство использования	8/10
Документация	6/10

» Mageia хорошо обособилась от Mandriva, но чтобы стать первоклассным дистрибутивом, ей пока не хватает лоска.

Рейтинг 7/10

Pianoteq 4 Standard

Грэм Моррисон решил тряхнуть стариной и обнаружил, что пианино не везде звучат так же, как на его Casiotone MT-500.

Вкратце

» Коммерческий эмулятор пианино, превращающий Linux в концертный рояль. См. также: свободные звуковые банки и FluidSynth.

Pianoteq – это программный эмулятор фортепиано. Подцепляете MIDI-клавиатуру, запускаете инструмент, и играйте себе. Если подключить наушники и колонки, создается «ощущение объемного живого звука», как мы писали в обзоре версии 3.0 в сентябре 2009. И это вовсе не значит, что Pianoteq просто воспроизводит сэмплы, записанные с реального инструмента. О нет. Эффект достигается математическим моделированием сложных связей между внутренними частями пианино и корпусом и их совместного звучания. Результат получается выразительным, динамичным и удивительно гибким в настройке, причем этот инструмент занимает мизерное место на диске и не дает большой нагрузки на ЦПУ – что и объясняет наше ликование при выходе версии 4.

Установить и настроить Pianoteq немногим сложнее, чем уссесться за пианино. Единственная версия для загрузки под Linux идет в 32- и 64-битном вариан-

«Никто и не заподозрит, что вы играете не на живом инструменте.»

тах, а сам пакет представляет собой архив 7z, что в последнее время стало редкостью. Файл README содержит полезные советы, как минимизировать задержку звука в вашей системе, чем мы и воспользовались, обновив настройки воспроизведения аудио в реальном времени на нашем



» В Pianoteq спасу нет от всевозможных регуляторов и скрытых меню.

Arch. Имеется автономный исполняемый файл, а также собственный VST-пакет, способный работать на любом VST-совместимом хосте; автономная версия может подключаться к аудиосистемам Jack и ALSA, но мы предпочли модуль VST под Renoise: интеграция со средой композиции как-то элегантнее.

Пианомания

Главное новшество этого релиза – звуковой движок: прототипом ему послужил концертный рояль Steinway D из Берлина. Качество звука – вещь субъективная, и надо признать, что нам довелось лишь раз поиграть на Steinway D, прежде чем охрана любезно препроводила нас из вестибюля отеля Peachtree Plaza в Атланте, 20 лет назад. Но нам эта новая модель показалась великолепной. Даже не потому, что она звучит лучше нашей сэмпл-версии (которая все равно не запустится на Linux): просто при игре на приличной клавиатуре создается иллюзия живого инструмента, что сэмплу не по зубам. И обертоны глубже, и отклик реалистичнее.

Если придираться, звук даже слишком чистый. Можно сделать более произвольным Момент атаки [в живом инструменте – момент удара молоточка по струне, – прим. пер.]; хорошо бы в будущем такую рандомизацию ввели и в другие параметры, пусть это и нарушит идеал. Ведь каждый технарь-пианист мечтает открыть новые возможности инструмента, настро-

ить его «под себя», и идеальность порой не главная цель. Мастер калибровки выжал все, что можно, из нашей увесистой клавиатуры (чтобы насладиться всем великолепием звука, нужна, конечно, хорошая). Понравился и новый инструмент реверберации, дающий ощущение объемности звука. Версии Standard и Pro позволяют даже загружать импульсные характеристики для имитации акустики ваших любимых залов; надеемся, что появится возможность загрузки стороннего аудио в Pianoteq с целью аналогичной обработки.

Ну, а если вы наденете смокинг и спрячете свой Linux под сценой, никто и не заподозрит, что вы играете не на живом инструменте. LXF



Свойства новскидку



Новые эффекты

В этой версии впервые появились Усилитель и Ревербератор...



Твердость молоточков

...а также регулировка силы удара молоточков по струнам.

LINUX FORMAT Вердикт

Pianoteq 4 Standard

Разработчик: Modartt SAS
Сайт: www.pianoteq.com
Цена: €99–399

Функциональность	9/10
Производительность	10/10
Удобство использования	9/10
Оправданность цены	7/10

» Фортепиано, которое не потребует настройки после очередного непогожего так называемого лета в Англии.

Рейтинг 9/10

Rosegarden 12.04

Грэм Моррисон вкладывает новообретенный талант пианиста в тестирование одного из лучших бесплатных MIDI-секвенсоров.

Вкратце

» MIDI- и аудио-секвенсор, может использоваться для создания партитур, написания и проигрывания музыки. См. также: *Ardour*.

Мы обошли стороной несколько релизов программы, часто называемой одной из самых мощных цифровых аудиостудий под Linux. Дело в том, что по характеру они были скорее итеративными, чем революционными, добавляя лишь рост производительности или улучшения интерфейса. Но за пять лет накопилось столько обновлений, что, увидев *Rosegarden* сейчас, испытываешь смешанные чувства ностальгии и удивления.

Главной проблемой *Rosegarden* некогда была привязка к звуковому серверу Jack. Jack, конечно, отличное средство заставить аудиоприложения взаимодействовать так, как физические устройства работают с аудио в студии звукозаписи. Но и запустить, и поддерживать его было сплошной мукой. Если «падал» Jack, то и *Rosegarden* вслед за ним. А теперь программа легко подключается к Jack, вне зависимости, запущен он или нет, да и сам Jack запускается быстрее.

Колючка в бок

За некоторым исключением – в частности, новой черно-белой цветовой схемы (под названием Thorn [колючка] Style), которая, возможно, неплохо смотрелась бы в темной студии, но не в нашем офисе – интерфейс пользователя улучшился. Теперь после перезагрузки вы услышите ту же мелодию, без потери параметров программных синтезаторов или эффектов; также не возникло проблем с изменением параметров, сохранением проекта и нахо-



» Отличная среда для MIDI-редактирования, особенно если вы читаете ноты, но в целом *Rosegarden* приотстал от жизни.

ждением прежней конфигурации при следующей загрузке. Интеграция *Rosegarden* с форматом DSSI для виртуальных инструментов превосходна. Мы легко добавляли новые, вставляли их в проекты, и все они работали как положено. Непонятно, зачем было добавлять ко всем подряд модулям кнопку Редактировать, есть у них встроенный редактор или нет, но зато нас порадовал новый автоматизированный подход к распределению MIDI-каналов. Теперь инструменты динамически забирают себе первый свободный канал, и при перемещении отрезков на временной шкале не путают данные контроллера.

Не одни только розы

Серьезные проблемы все же остались. Нам не понравилось, что в новом проекте так много готовых каналов – ведь их добавление должно быть делом пользователя, который может сделать свой шаблон сам. В плане написания музыки, довольно утомительно постоянно переключаться между Добавлением и Правкой. Было бы гораздо проще иметь для этого единый инструмент, как в *Ableton Live* или штейнберговском *Cubase*. Аудиомикшер тоже не мешает переделать. Здесь можно создавать под-шины, но нельзя использовать их как ах-посылы для создания шины эффектов, потому что нет суммы посыла; и неплохо бы, чтобы у каждого канала были функции «соло» и «без звука». Менять эффекты и параметры параллельно с про-

игрыванием файла получается, но не всегда. Хотя критиковать *Rosegarden* все же не совсем честно, ведь это по-прежнему самый полнофункциональный набор для редактирования MIDI под Linux, и к этому релизу в коде было много исправлений.

За несколько дней тестирования программа упала лишь однажды, и хотя матричный редактор, скажем так, не первоклассный, зато редактор партитуры на высоте. Но хотя *Rosegarden* и расцветает потихоньку, его коммерческие аналоги двигались за последние 10 лет вдвое быстрее. В результате он выглядит и ощущается немного архаичным и неудобным – а это не совсем то, что нужно, когда на вас нахлынет вдохновение. **LXF**



Свойства навскидку



Микширование в реальном времени

Аудиомикшер с показом сразу 16-ти каналов немедленно отображает всевозможные ваши действия.



Редактирование в реальном времени

Параметры синхронизации и эффектов редактируются в реальном времени, а значения сохраняются в проекте.

LINUX FORMAT Вердикт

Rosegarden 12.04

Разработчик: Chris Cannam и др.
Сайт: www.rosegardenmusic.com
Цена: Свободно под GPL

Функциональность	7/10
Производительность	6/10
Удобство использования	6/10
Документация	6/10

» Как и GIMP, *Rosegarden* не мешало бы подумать об удобстве использования.

Рейтинг 6/10

Runtu Gnome 12.04

Игорь Штомпель представляет новую версию российского дистрибутива, основанного на Ubuntu.

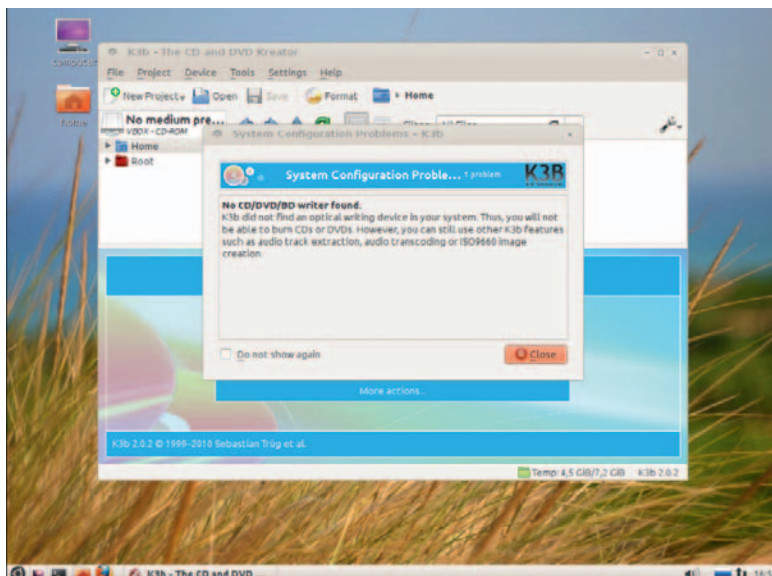
Вкратце

» Runtu — удобство Ubuntu для русскоязычных пользователей. См. также: Ubuntu, Linux Mint.

Разработка Linux носит ярко выраженный интернациональный характер. Разработчики со всего мира улучшают эту ОС, поддерживают уже существующие или создают свои дистрибутивы. И в нашей стране есть ряд проектов, предлагающих свои решения. Среди них заметное место занимает проект Runtu, который за время своего существования выпускал сборки на базе различных рабочих окружений (Gnome, LXDE, OpenBox). На этот раз новая версия дистрибутива вышла на базе Gnome.

Проект Runtu

Согласно Википедии (<http://ru.wikipedia.org/wiki/Runtu>), проект Runtu основали Алексей Черноморченко и Александр Бехер 14 мая 2007 года, одновременно с выпуском первой версии своего дистрибутива под названием Ubuntu Full Power. Правда, это название прожило недолго. Canonical изменила условия использования своих торговых марок, объявив: «Canonical принадлежит ряд торговых марок, в том числе Ubuntu, Kubuntu, Edubuntu и Xubuntu. Торговые марки зарегистрированы в двойной совместной форме: слово и логотип. Любой торговый знак, заканчивающийся буквами Ubuntu и Vuntu, будет похож на один или несколько товарных знаков, поэтому необходимо будет получать разрешение на его использование» (<http://www.ubuntu.com/aboutus/trademarkpolicy>). Пришлось подыскивать



» Рис. 1. KDE-приложения не имеют поддержки русского языка по умолчанию.

для проекта новое имя, и Александр Бешкенадзе предложил Runtu, что и было принято в июне 2007 года.

Проект продолжает свое развитие, и последняя версия дистрибутива – Runtu Gnome 12.04 – выпущена в конце июня сего года.

Получение и установка

На странице анонса выхода дистрибутива (<http://runtu.org/news-project/runtu-gnome-1204.html>) есть ссылка для полу-

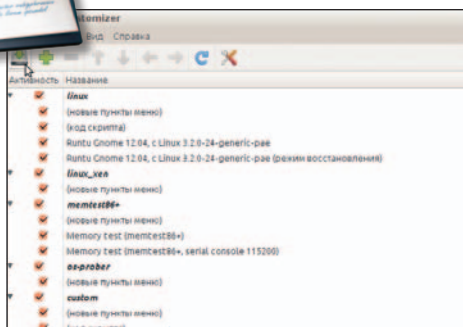
чения подробной информации. Пройдя по ней, вы попадете на страницу форума проекта Runtu (<http://forum.runtu.org/index.php/topic,2712.0.html>), содержащую описание новшеств и особенностей Runtu Gnome 12.04, а также ссылку на страницу сервиса Яндекс.Диск, с которой можно загрузить образ ОС. Размер его – 571 МБ.

Пользователи, справившиеся с Ubuntu 12.04, легко установят и Runtu. Есть небольшие отличия в процессе установки, но они не относятся к осложнениям. Так, при знакомстве с Runtu Gnome 12.04 и загрузке с LiveCD, если вы захотите установить ОС и щелкнете на значке Установить Runtu 12.04, появится окно Установка, где русский язык выбран по умолчанию (в отличие от Ubuntu). В процессе установки Ubuntu 12.04 после ввода имени пользователя, названия компьютера и т.д., начинается показ слайдов, а в нижней части окна можно наблюдать за самим процессом. В Runtu 12.04 вы слайдов не увидите, а окно установки будет «обрезано» – там отображается только информация о процессе установки. И, естественно, все упоминания Ubuntu заменены на Runtu.

Фишки Runtu Gnome

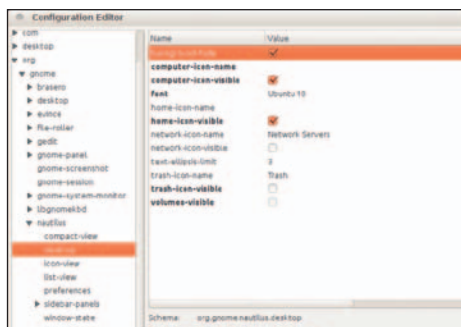
Как и следовало ожидать, Runtu в первую очередь ориентирован на русскоязычных пользователей и предоставляет русскую

Свойства навскидку



Восстановите Grub

Используйте *Grub Customizer* для восстановления системного загрузчика.



Dconf-editor

Тонкая настройка рабочего окружения и его скрытых параметров делается в Dconf-editor.

локализацию по умолчанию во время и после установки.

Однако у нас реализация локализации в Runtu Gnome 12.04 вызвала недоуменные вопросы. По умолчанию после установки русифицирован интерфейс рабочего окружения Gnome. А пакеты с поддержкой русского языка для KDE не установлены (*kde-l10n-ru, language-pack-kde-ru, language-pack-kde-ru-base*); также, несмотря на установку по умолчанию поддержки русского языка для LibreOffice (*libreoffice-l10n-ru*), не работала проверка орфографии.

С орфографией можно разобраться, установив пакет *myspell-ru*. Что же касается поддержки локализации KDE-приложений по умолчанию... Понятно, что дистрибутив ориентирован на Gnome, и KDE его заботит меньше. Но многие пользователи устанавливают в среде Gnome также и приложения KDE. И после установки *K3b* в Runtu Gnome 12.04, как показано на рис. 1, пользователь увидит интерфейс на английском языке. На наш взгляд, это серьезное упущение для дистрибутива, позиционируемого как российский.

Что до аппаратных архитектур, то Runtu выпускается только для платформы x86 на базе ядра версии 3.2.0-24-generic-pae. Минимальные системные требования выглядят следующим образом: процес-



Рис. 2. Рабочий стол Runtu Gnome 12.04. Русское поле...

и т.д.). Соответственно, переключаться между режимами можно с помощью Завершения сеанса, выбрав в окне менеджера дисплеев (LightDM) необходимый сеанс.

Разработчики Runtu 12.04 действовали как репозитории Ubuntu 12.04, так и репозитории с Launchpad. С перечнем последних можно ознакомиться, например, в Synaptic (Настройки > Репозитории > Другое ПО).

Теперь о составе программ, установленных в Runtu Gnome 12.04 по умолчанию. Для работы с мультимедиа предусмотрены *VLC, Audacity, RecordMyDesktop, Brasero*, для работы с графикой – *mtPaint*, а для работы в Интернет – *Pidgin*. Среди программ, не доступных по умолчанию в Ubuntu 12.04, можно отметить *GtkHash, Midnight Commander* (задействует встроенный редактор по умолчанию), *Grub Customizer, Htop*.

Особого внимания заслуживает *Ubuntu Tweak* – приложение, позволяющее настроить внешний вид интерфейса рабочего стола (темы, шрифты, параметры экрана входа в систему и т.д.). Но это не все его возможности. Вы также можете получить информацию о компьютере (аппаратная часть: процессор, память), рабочем столе (имя компьютера, платформа, дистрибутив, среда рабочего стола), выполнить ряд административных задач (подключить сторонние репозитории ПО, установить последние из них), произвести очистку системы (кэш *apt*, неиспользуемые пакеты, устаревшие ядра и т.д.) и кэша *Mozilla Firefox*.

Для реализации соединений VPN с помощью протокола L2TP разработчики добавили специальный модуль *network-manager-l2tp* (<http://seriyvs.ru/blog/2012/01/31/plugin-l2tp-dlya-networkmanager/>). Для этого был подключен соответствующий репозиторий с Launchpad (<https://launchpad.net/~seriy-pr+archive/network-manager-l2tp>).

Также в установку по умолчанию разработчики добавили модуль расширения

от Google, ориентированный на поддержку голосового и видеочата с помощью веб-интерфейса. Соответственно, для его использования необходимо войти в свою учетную запись Google.

Что касается документации и руководства, то вам стоит обратить внимание на специальную ветку форума проекта Runtu, которая называется «Руководства и инструкции» (<http://forum.runtu.org/index.php/board.6.0.html>).

Вы найдете здесь тему «Документация к Runtu Gnome 12.04» с краткими инструкциями по установке дистрибутива, установке/удалению программного обеспечения (*Synaptic, gdebi*), восстановлению загрузчика и др.

Заключение

Runtu – интересное решение, представляющее собой отечественное видение того, как должен выглядеть и функционировать персональный компьютер на базе Ubuntu. Рабочее окружение Gnome-Fallback выбрано явно с оглядкой на жизненные реалии: шикарные эффекты для не особенно авангардного машинного парка – неуместная роскошь.

Выбор включенного разработчиками по умолчанию программного обеспечения довольно удачен и отвечает потребностям современного пользователя настольной системы. Для любителей расширенной настройки *Ubuntu Tweak* предоставляет удобный плацдарм, откуда до всего можно дотянуться, так сказать, «не отходя от кассы». Досадно, конечно, отсутствие локализации KDE-приложений по умолчанию. Но, как всегда, *Synaptic* или *apt* помогут исправить положение.

Runtu Gnome 12.04 нетребователен к аппаратной части и должен работать даже на достаточно старых компьютерах. Радует и размер образа диска – всего 571 МБ. Определенно, данная версия Runtu – неплохое решение. Разработчики постарались для нас. **LXF**

«С орфографией можно разобраться, установив пакет *myspell-ru*.»

сор – Pentium 3 700 МГц, оперативная память – 256 МБ, требуемое пространство на жестком диске – 3 Гб. Разработчики подчеркивают, что при использовании компьютера с оперативной памятью до 512 МБ установку желательно начать из загрузочного меню LiveCD.

Не прельстившись ультрамодернистским Unity, Runtu Gnome 12.04 по умолчанию использует рабочее окружение Gnome-Fallback – это режим работы Gnome, предназначенный для компьютеров, не имеющих поддержки 3D-акселерации (см. рис. 2). По словам разработчиков, он «выглядит аналогично “классическому” интерфейсу Gnome 2 в однопанельном режиме».

Коли речь зашла о режимах Gnome – по умолчанию работает сеанс Gnome Classic (No effects), который использует оконный менеджер *Metacity*. Альтернативный режим, Gnome Classic, использует оконный менеджер *Compiz*. Режим с *Compiz* предоставляет расширенные возможности настройки интерфейса рабочего стола (поведение окон, эффекты

LINUX **Вердикт**
FORMAT

Runtu Gnome 12.04

Разработчик: Сообщество проекта Runtu.org
Сайт: <http://runtu.org/>
Цена: Бесплатно

Функциональность	9/10
Производительность	9/10
Удобство использования	9/10
Документация	9/10

» Нетребовательный к аппаратуре российский дистрибутив. Огорчает, правда, «недолокализация».

Рейтинг **9/10**



ТЯЖБА С ORACLE ЗАВЕРШЕНА

Поисковый гигант выиграл

Google не заплатит ни цента в рамках судебного разбирательства с Oracle.

Завершилась заключительная, третья, фаза судебного разбирательства, рассматривавшего обвинение в нарушении связанной с Java интеллектуальной собственности компании Oracle в платформе Android. Состоявшееся заседание имело значение не только для вовлеченных в разбирательство отдельных компаний и экосистемы Android, но и для всей индустрии, так как на данном заседании судья должен был принять решение, являются ли нет программные интерфейсы (API) объектом авторского права. В итоге решение вынесено в пользу Google – суд постано-

» Адвокат Oracle Дэвид Бойе (David Boies) задает вопросы по существу вице-президенту Google Энди Рубину (Andy Rubin) (слева).



вах на одну версию кода и при этом запретить другим создавать свои собственные варианты реализации, повторяющие все или некоторые функции изначального варианта. Формирующее API дерево имен, конечно, содержит в себе элементы творчества, но оно также является частью структуры команд – набора символов, связанного с определенной функцией. Подобный набор команд трактуется законом об авторском праве как не подпадающий под действие копирайта, так как дублирование структуры команд является непременным условием обеспечения совместимости и переносимости. Поэтому идентичность строк с декларациями и заголовочными описаниями методов не имеет значения: для реализации ана-

логичной функциональности они и должны совпадать, даже если сама функциональность реализована по-другому. Так как существует только один способ выражения идеи или функции, то каждый должен использовать идентичные декларации, и никто не может монополизировать такие выражения.

Напомним, что в начале мая 2012 года Суд Европейского союза вынес аналогичное решение в деле «SAS против WPL», постановив, что API, языки программирования и поведение программ не могут быть отнесены к категории интеллектуальной собственности и не подпадают под защиту копирайта. Спустя несколько дней в США завершилась первая фаза разбирательства «Oracle против Google», когда присяжные признали нарушение прав Oracle по 2 пунктам из 5 (заимствование 9 строк кода, что может вызвать штраф в несколько тысяч долларов), но не достигли консенсуса по вопросам, можно ли считать API интеллектуальной собственностью и выходит ли использование API Java в Android за рамки условий «добросовестного использования [fair use]». На второй фазе разбирательства присяжные отвергли все обвинения в нарушении патентов Oracle в платформе Android.

«Признано, что API не подпадает под защиту копирайта.»

вил, что структура, последовательность и организация 37 фигурирующих в деле API Java не является интеллектуальной собственностью компании Oracle.

Признание, что API не подпадает под защиту копирайта, подтверждает легальность создания совместимых функциональных аналогов программных платформ и невозможность монополизации определенных программных интерфейсов. Решение также подтверждает правомерность использования в сторонних продуктах API без выплаты лицензионных отчислений. Декларируя свое решение, судья заявил, что так как код, используемый для непосредственной реализации методов, отличается, в соответствии с законом об авторском праве любой может написать свой вариант кода, выполняющий аналогичные функции или действия, описанные в Java API.

Признание доводов компании Oracle позволило бы любому заявить о своих пра-

Так почему же «ни цента»?

Несмотря на победу Google по всем фронтам в деле об использовании связанной с Java-технологиями интеллектуальной собственности Oracle в платформе Android, компанию Google все же ожидала выплата возмещения ущерба в размере нескольких тысяч долларов, так как присяжными было признано заимствование 9 строк кода с реализацией функции `rangeCheck()`. Но компания Oracle сочла данное нарушение незначительным и согласилась признать размер ущерба в данном эпизоде нулевым.

Таким образом, компания Google не только ничего не выплатит Oracle, но и имеет возможность в течение двух недель взыскать с Oracle компенсацию судебных издержек, так как иск можно считать необоснованным. Размер подобных издержек может исчисляться миллионами долларов.

Со своей стороны, Oracle готовится подать апелляцию и планирует обжаловать решения, касающиеся нарушения патентов, вынесенные в ходе судебного разбирательства.

ТЯЖБА С APPLE НАЧИНАЕТСЯ

Злоключения смартфонов в США

Прошлый месяц принес поклонникам Android не только радости, но и огорчения.

Компания Apple, выполняющая роль второго «запятого друга» Google и Android после Microsoft, добилась права на запрет продаж в США нового флагманского смартфона компании Samsung – Galaxy Nexus. Окружной судья Люси Ко [Lucy Koh] вынесла вердикт в пользу Apple. Apple согласилась депонировать 95 миллионов долларов на возмещение возможного ущерба Samsung, если данное решение суда будет пересмотрено. Постановлено, что компания Samsung нарушила следующие технологии, защищенные патентами Apple на ПО:

- » метод обнаружения и выделения таких данных, как номера телефонов или адреса электронной почты, а затем инициирование звонка или создания e-mail, когда пользователь щелкнул по этим данным;
- » метод одновременного поиска данных в нескольких СУБД и источниках данных;
- » разблокирование сотового телефона проведением пальцем по его экрану;
- » функция автозавершения набора слов, позволяющая пользователю принять или отвергнуть предложенные варианты.

Все это применяется во всех устройствах на базе Android; так что Apple торпедирует продажи всех смартфонов и планшетов на базе данной ОС в США. Интересно,

что в деле фигурирует и смартфон Galaxy Nexus – продукт Samsung, продаваемый от лица Google.

Заседание 2 июля 2012 года дало Samsung и Google время на подачу апелляции, отсрочив запрет. Впрочем, судебные решения только предоставили Apple возможность требовать запрета продажи

Около 70% компаний, продающих в США устройства на базе Android, шлют дань Microsoft. За закрытыми дверями, без разглашения финансовых деталей и списка нарушенных патентов, подписаны подобные сделки с Samsung, Acer, ViewSonic, HTC, LG, Quanta Computer, Velocity Micro, General Dynamics, Onkyo, Wistron Corp и Compal Electronics. Из крупных производителей не заключили соглашения с Microsoft только Motorola, Huawei и Sony Ericsson.

Представители Nokia заявили о нарушении патентов компании в представленном недавно планшете Google Nexus 7, поставки которого начались в середине июля этого года. Nokia утверждает, что ни компания ASUS – изготовитель данного планшета, ни Google, от лица которого будет продаваться устройство, не лицензировали ряд патентов Nokia, связанных с беспроводной передачей данных.

По сообщению Nokia, уже около 40 производителей мобильных устройств осуществили лицензирование данных патентов и выплачивают отчисления, но среди них нет ASUS и Google. Указанные патенты являются стандартообразующими, т.е. их нарушает любое устройство, поддерживающее IEEE 802.11 Wi-Fi.

«Суд дал Samsung и Google время на подачу апелляции.»

устройств у регулирующих органов США; воспользуется ли этим Apple, пока не ясно.

Компания Microsoft заключила с компаниями Aluratek и Coby Electronics соглашение, признающее наличие в Android и Chrome OS технологий, нарушающих патенты Microsoft, и назначена выплата отчислений за все проданные потребительские устройства на их базе. Сумма отчислений остается тайной, но по неофициальной информации это 10–15 долларов с устройства. Coby и Aluratek производят дешевые (\$100–200) планшеты, и даже малое подорожание продуктов заметно скажется на их конкурентоспособности.

С ANDROID НА БОРТУ

«Игрозапускалка» за \$100

Проект OUYA намерен создать полноценную игровую приставку на базе Android.

На сайте общественного финансирования творческих проектов Kickstarter появилось сообщение о разработке новой игровой консоли OUYA (читается как «Ойя») на платформе Android. За первый же день сбора средств скопилось более двух миллионов долларов, при плане в 960 тысяч. Проект будет готов к весне следующего года. Все разработки планируется распространять под открытыми лицензиями; разработчикам игр подготовят открытый и бесплатный SDK.

Устройство базируется на платформе NVIDIA Tegra 3, сочетающей 4-ядерный CPU ARM Cortex-A9 MPCore, добавочное пятое ядро с низким энергопотреблением и 12-ядерный GPU Nvidia, и управляется



» Стильный дизайн Ouya так и приглашает в нее поиграть!..

тачпадом и беспроводным контроллером (2 джойстика, d-pad и 8 кнопок). На борту OUYA 1 Гб, Flash 8 Гб, HDMI (1080p HD), Wi-Fi 802.11 b/g/n, Bluetooth LE 4.0, порт USB 2.0.

Планируемая цена приставки – \$99. Приложения для нее будут поставлять-

ся по модели “free to play”, т.е. с бесплатной демо-версией для оценки игры до покупки.

Подразумевается, что разработчики будут поставлять базовый костяк игры, а плату взимать за подписку на новые версии, дополнительные уровни и расширенные возможности. Версию для приставки OUYA уже готовят разработчики *Minecraft*. Привлечен Ив Беар [Yves Béhar], известный по проектам Jawbone Jambox и One Laptop Per Child. Консультанты – Эд Фрис [Ed Fries] из команды Xbox и Жюли Урман [Julie Uhrman] из IGN, основатель и исполнительный директор Ouya, а также Маффи Гадиали [Muffi Ghadiali] из команды Amazon Kindle. **LXF**



Android: А что у нас внутри?



Наш
эксперт

Романа Ярыженко хлебом не корми – дай попробовать какую-нибудь экзотическую (и не очень) ОС. Желательно открытую.

Роман Ярыженко решил попристальнее приглядеться к отличительным особенностям платформы Android.

О платформе Android не слышал нынче, кажется, только ленивый. Это неудивительно – по многим причинам; некоторые из них будут рассматриваться дальше. Пока же можно отметить беспрецедентную открытость по сравнению с другими популярными платформами (конечно, на момент написания статьи вслед за Google исходные коды начали открывать и другие фирмы, но Android был первой ласточкой). На этом вступление можно закончить и перейти к предмету написания статьи.

История успеха

Google купил стартап Android Inc в 2005... Нет, пожалуй, надо сначала сделать небольшой экскурс в прошлое. Прежде всего отметим, что Linux стал использоваться в сотовых телефонах за 4 года до выпуска первой версии Apple iOS, то есть в 2003 году. И первым смартфоном (не прототипом), в котором официально использовался Linux, был Motorola A760. В этом же году свой смартфон на базе Linux, SCH-i519, представил Samsung. Предлагаю рассмотреть первый – хотя бы ради интереса.

Каким же он был, первый смартфон на основе Linux? MontaVista CEE 3.0 на основе ядра 2.4.20, Intel Xscale PXA262, 16 МБ ОЗУ, 32 ПЗУ... слабоват по современным меркам. И несмотря на то, что там стоял Linux с QT, на нем без особых телодвижений... нельзя было запускать ничего, кроме J2ME-приложений! Ладно. Смотрим, скажем, в 2005 год. В конце года был анонсирован китайский смартфон Haier N60 (первый официальный Linux-смартфон на российском рынке). Характеристики – процессор Intel Xscale PXA271, 312 МГц, 59 МБ ПЗУ... и опять же невозможность запускать что-либо, помимо J2ME.

Картина ясна: несмотря на наличие Linux в смартфонах того времени, они были почти такими же закрытыми, как и те, что были основаны на проприетарных платформах (OSE, Nucleus...).

В июле 2007 года фирмой OpenMoko был выпущен смартфон Neo1973. Смартфон был полностью (за исключением, быть может, стека GSM) открытым. Но из-за того, что целевой аудиторией были техна-

ром удачной маркетинговой политики), он не получил широкого распространения.

В том же 2007 году Google был основан OHA – Open Handset Alliance. День его основания и считается началом Android-эры.

Первым устройством на основе Android стал HTC Dream, он же T-Mobile G1. Сей гаджет имел на борту 192 МБ ОЗУ, 256 МБ ПЗУ, процессор Qualcomm 7201 528 МГц и QWERTY-клавиатуру.

Возникает резонный вопрос: почему именно Android, а не какой-нибудь Mizi Linux? Помимо закулисных интриг Google (без которых, как мне кажется, действительно не обошлось) и прочих конспиративных теорий заговора, имеется и вполне резонный ответ на этот вопрос. Производителей привлекает низкая стоимость разработки. Ну еще бы – за ядро платить не надо, за интерфейс – тоже. А также, что немаловажно, большая часть фреймворка Android кроссплатформенна, что, впрочем, больше касается разработчиков ПО. Конечным же пользователям нравится централизованный Android Market – своего рода репозиторий, идея которого была подсмотрена у Apple, ну а Apple же, по всей видимости, взяла эту идею у дистрибьюторов Linux.

...устройства на основе Android штампуются десятками тысяч в день, и не только с именитыми брендами (Motorola, Alcatel), но и безвестными фирмами. Давайте же перейдем к описанию архитектуры и возможностей этой платформы.

Внутреннее устройство

Начнем с очевидного. В основе Android лежит ядро Linux. Нет, исключения, конечно же, имеются – поскольку, строго говоря, то, что большинство пользователей называет словом «Android», теоретически может запускаться даже на Windows CE. Но здесь и далее мы не будем рассматривать подобные случаи (точнее говоря, мы будем рассматривать Android 2.2 с ядром 2.6.39, но, думаю, все что будет описано, применимо и к старшим версиям, а частично – и к младшим).

Архитектура ядра, в общем-то, такая же, как и везде (за исключением, может быть, IPC – но IPC частично относится к пользовательскому режиму), поэтому мы не будем на ней останавливаться, а поднимемся выше. Однако, если это будет необходимо, мы опишем механизмы ядра – тот же IPC.

Выше у нас *libc*. В качестве такового используется не *glibc*, а *Bionic*, построенный на базе *libc* BSD – Google посчитала, что *Glibc* не подходит из-за тяжеловесности и лицензионных ограничений. Но, поскольку *Bionic* создавался с учетом встраиваемых систем, разработчикам пришлось поступиться некоторыми функ-



» Смартфон OpenMoko NEO1973.

циями «старшего брата». Так, он не поддерживает исключений C++, у него собственная реализация pthread, вместо использования, к примеру, `.etc/resolv.conf` используется область разделяемой памяти, многие функции, типичные для «настольных» дистрибутивов, такие как `getrrent()`, просто-напросто отсутствуют из-за специфики архитектуры.

Что же дальше? А дальше идет слой HAL – набор библиотек, написанный на C/C++, служит для обеспечения прозрачности работы с оборудованием. И на этом же примерно уровне остальные Native-библиотеки (мы не будем их подробно рассматривать, но перечислим некоторые) – WebKit, SSL, OpenGL ES...

Поднимаемся выше – на уровень Native-приложений. К такому относятся, например, `install` – демон, через который проходит установка пакетов, и `rild` – демон-прослойка между аппаратно-специфичным и платформо-независимым ПО GSM.

Едем дальше... что мы видим? Видим фреймворк и виртуальную машину, очень похожие на JRE, но формально им не являющиеся. Тем не менее, набор базовых классов и синтаксис основного языка программирования ясно дают понять, что в основе фреймворка лежит творение Sun.

Но как же выполняется деление на процессы и выполняется ли оно вообще? Для ответа на первый вопрос необходимо ответить на второй. Да, выполняется. Но зачем? Неужели нельзя положить на фреймворк? Дело в том, что, если на обычном компьютере процесс виртуальной машины Java «упадет», с системой ничего не случится. А если он упадет на Android, где на Java написан практически весь средний и верхний уровень, то последствия будут фатальными. Таким образом, разделение на процессы оказывается необходимым. Теперь ответим на вопрос «как?». В JRE 5 была спецификация Isolates, и кое-где они применяются до сих пор (JNode), но в JRE 6 их убрали, а в Android используются обычные Linux-процессы. Каждый новый процесс Android порождается ответвлением процесса Zygote (исключение составляет `system_server`, в котором выполняется, например, графическая подсистема) и имеет название вида `com.android.Launcher`. Фактически, каждый процесс запускается в собственной VM, которая носит название Dalvik VM. Особенности Dalvik VM таковы: собственный формат хранения классов DEX, собственный байт-код, регистровая архитектура, а начиная с версии Android 2.2 – еще и JIT-компиляция.

Рассмотрим некоторые части фреймворка, а также некоторые специфичные понятия Android.

Приложения Android строятся на основе уже имеющихся классов. То есть, к примеру, если потребуется разработать собственный редактор, для этого берется уже готовый класс и расширяется/дополняется до необходимого. В этом (да и не только в этом) смысле платформа Android более похожа на Web-приложения и фреймворки.

У приложения всегда есть манифест (нет-нет, это ни в коем разе не манифест Коммунистической партии! Это всего-навсего конфигурационный файл формата XML, который при сборке приложения преобразуется в двоичный вид). Он предназначен для описания структуры приложения, его привилегий, требований к железу... в общем, много для чего.

Приложение может быть многоуровневым. Рассмотрим список основных классов фреймворка Android, на основе которых строятся большинство приложений (этот список далеко не полный; сюда включены лишь самые основные классы):

» Activity – грубо говоря, логика GUI; сам графический интерфейс описывается в XML-файлах слоев. Activity доступен только



» HTC Dream, первый коммуникатор на основе Android.

на переднем плане, во всех остальных случаях он уничтожается – исключение составляет разве что случай, когда теряется фокус, в этом случае Activity приостанавливается.

В Android не реализован многооконный интерфейс – по той простой причине, что экран не позволяет вместить большое количество окон. Вместо этого используется модель стека: на вершине находится тот Activity, который сейчас на переднем плане. При нажатии аппаратной кнопки «Back» текущий Activity уничтожается, а тот, который находился ниже в стеке, создается вновь.

» ContentProvider – если Activity можно считать клиентом, то ContentProvider в терминах web-фреймворков выполняется на стороне «сервера» (на самом деле, все несколько сложнее, но рассмотрение шаблона проектирования MVVM выходит за рамки данной статьи). ContentProvider предоставляет Activity данные и в большинстве случаев является оберткой вокруг SQLite. Выполняется он в контексте требующего данных Activity, поэтому не может работать сам по себе.

» BroadcastReceiver является некой «прослойкой», которая перехватывает широковещательные сообщения (Intents; о них чуть позже) и решает, что делать в ответ. Иначе говоря, это обработчик событий, которые создаются путем отправки сообщения методом `sendBroadcast` (либо `sendOrderedBroadcast`) соответствующего класса. Необходимо отметить, что экземпляры класса BroadcastReceiver «одноразовы», т.е. их необходимо либо регистрировать в манифесте, либо после срабатывания регистрировать заново (в случае, если BroadcastReceiver использовался в контексте Activity).

» Service... думается, что с этим все более-менее понятно. Это компонент приложения, работающий в фоновом режиме и не имеющий графического интерфейса. Примером может служить MP3-плеер или, допустим, служба предупреждения о низком заряде аккумулятора. В отличие от Activities, которые уничтожаются, если пользователь запустил другое приложение, службы уничтожаются либо сами, либо в случае, если имеет место истечение памяти.

Теперь, кажется, можно перейти к следующему разделу данной статьи.

IPC и безопасность

IPC в Android довольно своеобразна. Помимо стандартных средств Unix IPC (каналы, сигналы, сокеты) в платформе от Google появилось новое средство – BINDER. Но, как водится, «Новое – это

«Приложения Android строятся на основе имеющихся классов.»

»

➤ Первый телефон на Linux.



хорошо забытое старое», поэтому совершим краткий исторический экскурс.

Предтеча BINDER, OpenBinder, был разработан в ныне уже не существующей Be Inc. в пику попыткам разработки объектно-ориентированной ОС. Большинство из этих попыток предполагали поддержку OOP ядром. OpenBinder же, напротив, мог базироваться на существующих платформах. Практически, аналогом может служить технология COM от Microsoft или (возможно, в меньшей степени) CORBA.

Почему именно BINDER, а не SysV IPC? Дело в том, что SysV IPC подвержен утечкам ресурсов. А одним из требований к архитектуре BINDER было как раз требование не допускать утечек.

Посмотрим, как устроен BINDER в Android. В самом низу лежит модуль ядра, написанный на C. Что делает этот модуль? Он предоставляет базовые операции-примитивы над передаваемыми ресурсами. Файл устройства `/dev/binder` поддерживает такие операции, как `open`, `mmap`, `release`, `poll`, а сам драйвер поддерживает `ioctl`. Файловые операции мы рассматривать не будем, а рассмотрим самую «вкусную» – `ioctl`. Аргументы системного вызова – код команды и буфер данных. Коды команд могут быть такими:

➤ `BINDER_WRITE_READ` – наиболее часто используемая команда, передает пакет(ы) данных драйверу.

➤ `BINDER_SET_MAX_THREADS` – используется для задания максимального количества потоков для каждого процесса, которые должны отвечать на запросы BINDER. Иначе говоря, это ограничивает число соединений с каждым процессом.

➤ `BINDER_SET_CONTEXT_MGR` – задает Context Manager (см. далее), может быть выполнена только один раз при загрузке Android.

➤ `BINDER_THREAD_EXIT` – используется, если связанный поток завершился.

➤ `BINDER_VERSION` – ну... тут без комментариев.

У читателя может возникнуть резонный вопрос: зачем

реализовывать BINDER в режиме ядра? Если не вдаваться в подробности (а мы не будем этого делать, ибо размер статьи ограничен), здесь есть два ответа: во-первых, это дает выигрыш в производительности, а во-вторых – по историческим причинам.

Выше модуля ядра лежит часть фреймворка BINDER, реализованная в режиме пользователя и написанная на C++. Здесь не имеется ничего такого, на что можно было бы обратить внимание, разве что тот момент, что программы на «чистом» C++ могут использовать фреймворк напрямую, но при этом придется заново описывать абстракции более высокого уровня, которые на Java уже реализованы.

Переходим на уровень Java. Тут появляются новые термины:

➤ Binder – в англоязычных источниках так называется как архитектура BINDER, так и конкретная реализация интерфейса. Самое время вспомнить, что «Binder» в переводе с английского означает... в общем, вещество, которое *связывает*. Далее мы будем именовать саму идею, архитектуру «связывания» прописными буквами – «BINDER», конкретную реализацию – строчными, а в случае, когда более подходящим будет русский термин, будем использовать его.

➤ Интерфейс IBinder [IBinder interface] – строго определенный набор методов, свойств и событий, который может поддерживать конкретная реализация Binder.

➤ Объект Binder [Binder Object] – экземпляр класса, который принимает Binder. Может реализовывать несколько связываний.

➤ Маркер Binder [Binder Token] – уникальный идентификатор Binder.

➤ Объекты Intent – объекты, инкапсулирующие сообщения. Эти объекты могут использоваться как внутри одного процесса, так и для межпроцессного взаимодействия. Необходимо отметить, что сообщения асинхронны. Очевидно, они могут использоваться и для оповещения о каком-то событии.

➤ AIDL (Android Interface Definition Language) – используется для описания преобразования объектов Java в объекты, которые могут передаваться между процессами. Передавать можно только примитивы, строки, списки и объекты, реализующие интерфейс Parcelable. AIDL на практике на этапе компиляции преобразуется в код Java, который «запаковывает» и «распаковывает» соответственно передаваемые методу параметры и возвращаемые значения в объекты Parcel.

➤ Parcel – образно, упаковка для передачи более сложных объектов, чем примитивы Java. Класс должен реализовывать «запаковку» и «распаковку» этих объектов.

Читателю, скорее всего, уже надоели все эти перечисления. Тем не менее, код мы рассматривать не будем, но и излишне теоретизировать тоже не будем. Вместо этого ответим на вопрос: как же это все работает? И так...

При запуске Android диспетчер служб [Service Manager] регистрируется в модуле ядра Binder в качестве Context Manager путем отправки соответствующего кода команды, и получает идентификатор Binder, равный нулю. Это делается из соображений безопасности, так как «сферический клиент в вакууме» не должен заранее знать маркер связывания сервера. Все операции по связыванию проходят через диспетчер служб.

Когда сервер (в роли которого обычно выступает служба) запускается, он вызывает метод `onBind` и таким образом регистрируется в системе. Если клиент хочет связаться с сервером, он вы-

зывает метод `bindService`, среди параметров которого имеется, в том числе, и Intent, в котором находится имя службы. После этого, если не вдаваться в технические подробности, клиенту диспетчером

«Сотовые телефоны нынче — предметы личной гигиены.»

служб выдается маркер Binder, с помощью которого и выполняются дальнейшие операции. Если сервер находится в одном процессе с клиентом, модуль ядра не используется.

В данном описании все очень упрощено (к примеру, не рассматривается, что должно получиться в результате компиляции AIDL-файла).

Упомянем еще один механизм IPC Android – `ashmem`, механизм разделяемой памяти. Основное отличие его от `SysV shm` – то, что имеется счетчик ссылок сегментов; таким образом уменьшается вероятность утечки памяти.

Теперь о принципах безопасности. Безопасность в Android обеспечивается в основном при посредстве трех следующих механизмов:

1 Поскольку устройства Android по идее однопользовательские (в самом деле, не будет же среднестатистический пользователь сотового телефона разрешать звонить с него другим лицам? Сотовые телефоны нынче – что-то наподобие предмета личной гигиены), Google сделал «финт ушами» – он не стал изобретать велосипед, а использовал для изоляции каждого приложения систему UID и GID. Таким образом, если какое-либо приложение попытается удалить или изменить данные другого, у него этого не получится.

2 Механизм привилегированных API. Доступ к некоторым ресурсам Android, таким, к примеру, как телефония и SMS, ограничен. Это необходимо, поскольку Android – система многозадачная, а некорректное использование данных ресурсов может навредить системе, пользователю или его кошельку. Чтобы ис-



➤ Пример главного экрана Android (версия 2.0).

пользовать какие-либо привилегии, при разработке приложения необходимо их прописать в манифесте. Пользователь при установке приложения может либо согласиться, либо не согласиться со всем набором привилегий полностью; выбирать их по отдельности он не может. Но в некоторых модифицированных прошивках (таких, например, как SuapogenMod) подобная возможность предусмотрена.

И, наконец, каждое Android-приложение должно быть подписано цифровым сертификатом (проверяется только при установке). Это позволяет уникально идентифицировать автора и, в теории, должно препятствовать распространению вредоносных программ. На практике же абсолютное большинство приложений является самоподписанными, в отличие от аналогичной системы, например, в Symbian, где подписанными приложения могут быть только с использованием сертификата от центра сертификации, привязанного к IMEI и к тому же предоставляющего ограниченный набор привилегий (за расширенный набор уже надо платить и, возможно, отправлять подписанное приложение на проверку). Автор не нашел в исходных кодах Android возможности отключить самоподписанные сертификаты; судя по всему, таковой там и не имеется.

Но пришла пора закруглять статью. Скажем лишь, что в одной статье трудно описать все возможности того или иного программного проекта. Это в полной мере относится и к Android. Можно лишь с тем или иным успехом выделить отличительные особенности платформы, что мы и попытались сделать.

Итоги

В заключение следует отметить, что Google сейчас задает направления в области развития высоких технологий, и мобильных платформ это тоже касается. Но что ждет Android в будущем? Не прекратят ли его разработку в результате патентных козней «злых дядей» из Редмонда?.. На этот вопрос можно будет дать ответ лет через 5; пока же шествие Android по планете выглядит триумфальным. **LXP**



Академическая программа для учебных заведений

Mandriva Linux

Mandriva.Ru предоставляет учебным заведениям лицензию, дающую право на неограниченное по числу рабочих станций использование дистрибутива Mandriva Linux на всех компьютерах в образовательном учреждении, всех компьютерах преподавателей и всех компьютерах учащихся, в том числе и домашних.

По этой программе учебное заведение получает:

- ★ свежие версии дистрибутива Mandriva Linux (дважды в год)
- ★ доступ к обновлениям системы
- ★ техническую поддержку

Комплект поставки:

Mandriva Linux Powerpack 2009.1 Spring — 32- и 64-битные версии (2 DVD), а также печатное руководство

- ★ Mandriva Free 2009.1 Spring
- ★ Mandriva One 2009.1 Spring
- ★ Репозиторий Mandriva 2009 — бинарные пакеты для платформы x86 (4 DVD)
- ★ EduMandriva (1 DVD) — дополнительное ПО для образования
- ★ Академическая лицензия

www.mandriva.ru

Тел.: (812) 309-06-86, (499) 271-49-55
info@mandriva.ru



Android: Базы данных в облаке



Наш эксперт

Дзюльетта Кемп – писатель-фрилансер; сейчас она думает над тем, что еще ей удастся сделать с помощью HTTP-подключения своего телефона к серверу.

Дзюльетта Кемп показывает, как подключиться к облаку и заставить свое приложение взаимодействовать с сервером.

В прошлом месяце мы работали с локальной базой данных Android. Ну, а если нужно связаться с удаленной базой данных? Есть несколько способов это сделать, но на нашем уроке мы рассмотрим вариант, при котором для общения с базой данных используются HTTP-запросы – это, пожалуй, самый простой способ. Этот вариант также является расширяемым: мы получаем информацию из базы данных, но с помощью тех же

методов и API можно получить все, что готов предоставить веб-сервер. Мы будем обрабатывать данные с помощью JSON и посмотрим, как делать это в фоне, чтобы основное приложение не зависало при ожидании подключения от сервера.

Этот код основан на коде для работы с локальной базой данных из предыдущей статьи; если вы ее не читали, возьмите полный код приложения с нашего диска.

Базовый запрос HTTP

Прежде чем начать писать код HTTP-запроса, приложению нужно предоставить права доступа к сети. Добавьте в манифест строку `<uses-permission android:name="android.permission.INTERNET" />`

Для выполнения HTTP-запроса существует два варианта API: **HTTPClient** и **HttpURLConnection**. Начиная с Android 2.3, Google советует использовать **HttpURLConnection**, но на практике многие еще предпочитают **HTTPClient**, и с ним легче получить советы или помощь. Чтобы продемонстрировать оба класса в действии, я воспользуюсь **HTTPClient** для написания простого метода получения данных с сервера, а **HttpURLConnection** – для простого метода отправки данных на сервер.

Разбор JSON

Для отправки данных туда и обратно вам понадобится формат хранения данных, который смогут понять и телефон, и веб-сервер. Хороший вариант, которым мы здесь и воспользуемся – JSON, сокращение от “JavaScript Object Notation [Формат описания объектов JavaScript]”. Это легкий формат обмена данными, который понятен и для чтения/записи человеку, и для разбора/генерации компьютеру. Существует множество других форматов обмена данными, но JSON – хороший базовый формат для строковых данных, которые и хранятся в нашей базе.

Один **JSONObject** будет напоминать описание одного из заданий из таблицы заданий приложения **ToDo**:

```
{ "_id": "1", "task": "test task", "duedate": "2011-12-24", "create-date": "1324660453756", "category_link": "2" }
```

Прежде чем вступать во взаимодействие с настоящим сервером **MySQL**, рассмотрим простой пример на PHP, который воспроизводит эту строку вручную:

```
<?php  
$output=array(
```

```
    "_id" => 1,  
    "category" => 'work'  
); print(json_encode($output));  
?>
```

Сохраните этот код на сервере в файле **test.php**, и теперь он готов ответить на запрос вашего телефона. Затем создайте новый класс Android в своем проекте **RemoteToDoSync.java**:

```
public class RemoteToDoSync extends Activity {  
    public static final String WEBADDRESS = "http://yourserver/  
test.php";  
    private static final String TAG = "RemoteToDoSync";  
    public void onCreate(Bundle savedInstanceState)  
    {  
        super.onCreate(savedInstanceState);  
        retrieveData();  
        Intent i = new Intent();  
        setResult(RESULT_OK, i);  
        finish();  
    }  
}
```

Добавьте его в **AndroidManifest.xml**, затем добавьте такие строки в **ToDo.java** для вызова Действия и обработки результата:

```
public void onCreate(Bundle savedInstanceState) {  
    ...  
    startActivityForResult(new Intent(this.getContext(),  
        RemoteToDoSync.class), SERVER_UPDATE_ID);  
    ...  
}  
@Override protected void onActivityResult(int requestCode, int  
resultCode, Intent i) {  
    super.onActivityResult(requestCode, resultCode, i);  
    switch (requestCode) {
```

```

case SERVER_UPDATE_ID:
    populateList();
default:
    // do nothing; no other result expected
}
}

```

Метод `retrieveData()` класса `RemoteTodoSync` получит нашу тестовую строку (представляющую объект `JSONObject`) из файла `test.php`:

```

private void retrieveData() {
    InputStream is = null;
    try {
        HttpClient httpClient = new DefaultHttpClient();
        HttpPost httpPost = new HttpPost(WEBADDRESS);
        HttpResponse response = httpClient.execute(httpPost);
        HttpEntity entity = response.getEntity();
        is = entity.getContent();
    } catch (IOException e) {
        Log.e(TAG, "IOException " + e.toString());
        e.printStackTrace();
    }
    try {
        BufferedReader reader = new BufferedReader
            (new InputStreamReader(is, "iso-8859-1"), 8);
        String line = null;
        while ((line = reader.readLine()) != null) {
            JSONObject jo = new JSONObject(line);
            Log.v(TAG, "id is: " + jo.getInt(TodoDatabaseProvider.ID));
            Log.v(TAG, "task is: " + jo.getString(TodoDatabaseProvider.TASK));
        }
        is.close();
    } catch (Exception e) {
        Log.e(TAG, "Exception: " + e.toString());
        e.printStackTrace();
    }
}

```

В первом блоке `try` создается `HttpClient`, отправляется запрос на URL, заданный в `WEBADDRESS`, и результат этого запроса сохраняется в `InputStream`. Во втором блоке производится разбор этого результата; сейчас результат просто выводится в лог-файл без дополнительных действий.

Посмотрев на код, вы увидите, что на самом деле считывается строка (`String`), которая затем вновь превращается в `JSONObject`. Так ее проще разобрать, поскольку можно воспользоваться методами разбора `JSONObject`, а не заниматься этим вручную. Как иллюстрируют приведенные здесь примеры получения целых чисел и строк, методами `getX()` из объекта `JSONObject` можно получить значения любых типов.

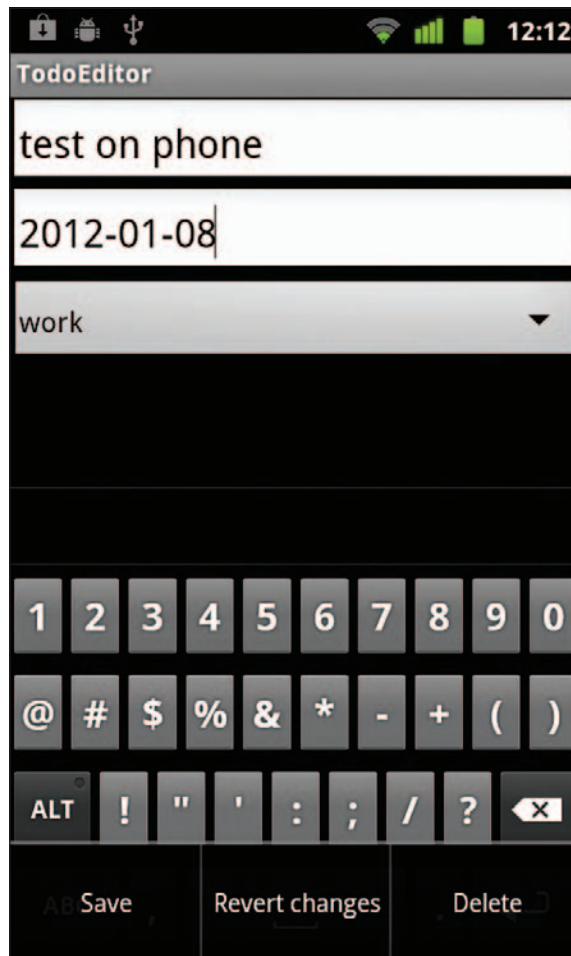
`InputStreamReader` преобразует байты входного потока (`is`) в символы с помощью преобразователя символов. `BufferedReader` буферизует результат в группах по восемь символов. С помощью этого примера можно посмотреть, что происходит (и убедиться, что все классы инициализированы правильно). Однако данные, полученные от настоящего сервера `MySQL`, будут представлены в виде `JSONArray`, который требует несколько большего разбора и выглядит так (квадратные скобки ограничивают массив):

```

[{"_id":1,"task":"test task", "duedate":"2011-12-24", "create-date":1324660453756, "category_link":2}
{"_id":2,"task":"new work task", "duedate":"2011-12-31", "create-date":1324661242417, "category_link":1}]

```

Вот простой пример PHP-кода для запуска на web-сервере (обратите внимание, здесь подразумевается, что база `MySQL` уже создана и заполнена данными, иначе никаких данных из запроса вы не получите):



▶ Приложение на реальном устройстве.

```

<?php mysql_connect("host", "user", "password");
mysql_select_db("todo");
$output;
$sql = mysql_query("select * from tasks"); while($e = mysql_fetch_assoc($sql) {
    $output[] = $e;
} print(json_encode($output)); mysql_close();
?>

```

Новый улучшенный вариант разбора JSON выглядит так (первый блок `try HttpClient` остался без изменений):

```

try {
    BufferedReader reader = new BufferedReader
        (new InputStreamReader(is, "iso-8859-1"), 8);
    String line = null;
    while ((line = reader.readLine()) != null) {
        JSONArray resultArray = new JSONArray(line);
        for (int i = 0; i < resultArray.length(); i++) {
            JSONObject jo = (JSONObject) resultArray.get(i);
            Log.v(TAG, "id is: " + jo.getInt(TodoDatabaseProvider.ID));
            Log.v(TAG, "task is: " + jo.getString(TodoDatabaseProvider.TASK));
        }
    }
    is.close();
}

```

Сервер вернет одну строку для всего массива (хотя если он вернет несколько строк, наш код сумеет справиться с их обработкой), которую мы разбираем сначала в отдельные объекты `JSONObject`, а затем в необходимые компоненты. Опять же, возвращается строка, которая преобразуется в `JSONArray` — это упрощает разбор.

»

Скорая помощь

Это не лучший способ проверки дубликатов. В идеальном случае нужно пользоваться уникальными идентификаторами, но они должны быть уникальными и на локальном, и на удаленном серверах. Здесь мы пользуемся простейшим вариантом, чисто чтобы проиллюстрировать идею.

Работа с полученными данными

Конечно, в том чтобы просто вывести данные в лог-файл, проку мало. Нам нужна какая-то форма синхронизации с базой данных Android. На этом уроке я не буду говорить о синхронизации (это довольно сложная тема – логика ее работы и проблемы, которые нужно обсудить, заняли бы большую часть статьи). Вместо этого мы выполним простую проверку на дубликаты и затем скопируем данные с сервера на устройство Android. В следующем разделе мы займемся отправкой данных с устройства Android на сервер, но по тем же причинам экономии места и внимания мы не будем касаться серверного кода, необходимого для их сброса в *MySQL* на стороне сервера.

При изменении кода локальной базы данных для работы с сервером метод `retrieveData()` во многом останется прежним. Однако `JSONArray todoDataFromServer` станет переменной класса, и мы не будем разбирать ее в компоненты `JSONObject`. Вместо

этого мы добавим вызов `addTodosFromServer()` в `onCreate()` после вызова `retrieveData()`. Это большой фрагмент кода, и мы поместим его на DVD в Листинге 1.

`addTodosFromServer()` просто перебирает объекты `JSONObject`, возвращенные сервером в массиве `JSONArray`, проверяет, есть ли объект в локальной базе данных, методом `checkIfTaskExists()` и добавляет его в базу данных, если его там нет.

`checkIfTaskExists()` получает строку «задания» из `JSONObject` и выполняет SQL-запрос, который ищет это значение в локальной базе данных. Если запрос не возвращает результатов, задания не существует локально, и его нужно добавить.

Наконец, метод `insertTask()` добавляет новую строку в базу данных, получает новые значения из `JSONObject` и затем обновляет новый URI этими значениями. Используемые здесь методы `insert()` и `update()` рассматривались в предыдущей статье (если вы ее пропустили, их код можно найти на DVD).

Отправка данных на сервер

Мы узнали, как получить данные с сервера, но надо и знать, как их отправить. На стороне сервера мы просто запишем эти данные в файл. В реальном приложении их нужно будет добавить в базу данных *MySQL*, а также подумать о синхронизации данных, а не просто записать все данные в базу. На этот раз мы воспользуемся `URLConnection` вместо `HttpClient`.

Вот пример кода, который нужно разместить на сервере:

```
<?php
$myfile = "jsonout.txt";
$fh = fopen($myfile, 'w') or die("can't open file");
$json = file_get_contents('php://input');
$obj = json_decode($json); fwrite($fh, "JSON var: " . $json);
fclose($fh);
?>
```

Этот код выводит значения в указанный файл. Код метода `addTodosToServer()` для отправки данных на сервер слишком длинен, мы поместили его на LXF DVD. В первой части мы просто получаем текущие задания из локальной базы данных и преобразуем их в объекты `JSONObject`. Затем они помещаются в массив `JSONArray taskArray`, который будет отправлен на сервер. В последнем блоке `try` делается вся трудная работа: здесь мы создаем объект для взаимодействия с сервером `URLConnection`. При этом особенно важен метод `setDoOutput()` – без него `URLConnection` не сможет ничего отправить. Метод `setChunkedStreamingMode()` позволяет отправлять данные пор-

циями, а не все сразу. С небольшим объемом данных это не играет особой роли, но с большими может произойти переполнение, и такую практику стоит ввести в привычку. Мы также задаем метод запроса (здесь делать это не обязательно, так как по умолчанию используется метод `GET`, также поддерживаются `POST`, `PUT` и др.) и тип содержимого заголовка запроса [`content-type`], который определяет, какое содержимое мы передаем.

Для отправки данных нужно подключить `DataOutputStream` соединения `URLConnection` и записать в него данные. После завершения записи важно очистить и закрыть его – и для того, чтобы гарантировать запись данных в поток, и для того, чтобы удалить все ненужные объекты. Наконец, обычно стоит получить статус запроса (это код возврата HTTP – 200 означает, что все прошло успешно, 404 – «страница не найдена» и т.д.) и записать его в лог-файл или обработать иным способом. После этого отключите `URLConnection`.

Теперь наше приложение может связываться и обмениваться данными с удаленным web-сервером в обоих направлениях. Как вы убедились, два способа реализации такого взаимодействия, `HttpClient` и `URLConnection`, похожи; можно выбрать любой, но в долгосрочной перспективе стоит предпочесть `URLConnection`, поскольку, как мы упоминали выше, `HttpClient` устарел. Наш код будет запускаться автоматически при запуске приложения. На DVD можно найти код, в котором эти же действия привязаны к пункту меню (это удобно при тестировании).

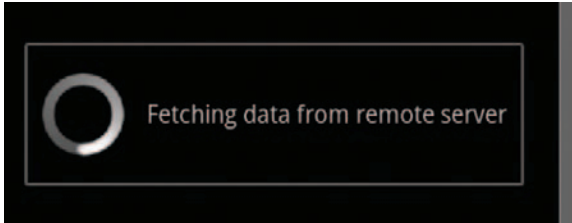
Асинхронные задачи

При запуске приложения Android система создает «главную» нить выполнения, которая отвечает за взаимодействие с виджетами и другими компонентами пользовательского интерфейса Android. Все компоненты в этом процессе запускаются в одной и той же нити, и если не делать ничего особенного по части нити, все остальное тоже будет выполняться в этой нити. Недостаток такого подхода состоит в том, что если взаимодействие занимает долгое время, пользователь может счесть приложение зависшим.

Все, что мы делаем здесь, выполняется в главной нити пользовательского интерфейса Действия. Это несложно с точки зрения программирования, но не слишком хорошо с точки зрения пользователя. Если приложению нужно много времени на подключение к серверу, оно затормозит. В лучшем случае это просто нервничает пользователя; в худшем, если приложение будет молчать достаточно долго, пользователь может не дожидаться своего счастья и принудительно закрыть приложение. Чтобы обойти

эту проблему, нам нужна другая нить для выполнения операций по работе с базой данных. Говоря о нитях, важно помнить, что все операции с графическим интерфейсом должны выполняться только в главной нити пользовательского интерфейса, так как инструментарий пользовательского интерфейса Android не является `thread-safe` (то есть не гарантирует безопасной работы при одновременном выполнении в нескольких нитях). И вся работа с кодом инструментария должна производиться в «главной» нити пользовательского интерфейса, и любые изменения в графическом интерфейсе (например, повторное заполнение списка после синхронизации с базой данных) должны выполняться в главной нити, а не в нашей второй нити.

В нашем приложении вся работа с базой данных будет осуществляться во второй нити, а затем мы вернемся в первую нить для обновления списка заданий. Самый простой вариант здесь – воспользоваться классом `AsyncTask`, предназначенным для уп-



► Не оставляйте своих пользователей во мраке — приятный индикатор выполнения процесса сообщит им, что делает приложение.

рошения подобной работы с нитями. Чтобы воспользоваться **AsyncTask**, нужно создать класс, унаследованный от него. Можно было бы создать внутренний класс в **RemoteToDoSync**, но это означало бы передачу выполнения нити из главного класса **ToDo** в **RemoteToDoSync**, что в свою очередь прервало бы взаимодействие с пользователем. Мы оставим **RemoteToDoSync** и создадим частный внутренний класс в **ToDo.java**, переместив все приватные методы **RemoteToDoSync**, которые и делают всю работу, в этот класс:

```
private class SyncDataTask extends AsyncTask<String, Void, Integer> {
    @Override
    protected void onPreExecute() {
        // not doing anything here yet
    }
    @Override
    protected Integer doInBackground(String... address) {
        getAddress = address[0];
        putAddress = address[1];
        getData();
        addTodosFromServer(taskArray);
        updateIdsOnServer();
        getLocalTasks();
        sendTasksToRemoteDb();
        return RESULT_OK;
    }
    @Override
    protected void onPostExecute(Integer result) {
        if (result == RESULT_OK) {
            populateList();
        }
    }
}
```

В приведенном примере показаны три главных метода, которые вы, возможно, захотите реализовать для своего **AsyncTask**: **onPreExecute()**, **doInBackground()** и **onPostExecute()**. Основные действия выполняются в методе **doInBackground()**.

В первой строке класса нужно определить шаблонные типы **AsyncTask**: здесь это **<String, Void, Integer>**. Первый аргумент — параметры, передаваемые **AsyncTask** при его вызове (см. код ниже), второй — тип единиц измерения хода процесса при выполнении в фоновом режиме, а третий — тип результата, возвращаемого методом **doInBackground()**. Это возвращаемое значение будет передано в метод **onPostExecute()**.

В данном случае метод **doInBackground()** получает адреса наших двух серверов: одного для получения данных, второго — для отправки, и вызывает уже написанные нами методы для взаимодействия с сервером. Все эти методы были перемещены в класс **SyncDataTask**, но они почти не изменились с прошлого раза (при необходимости загляните в код на DVD). Завершив свою работу, метод передает глобальную константу **RESULT_OK** методу **onPostExecute()**, который прибирается за задачей и возвращает соответствующее **Намерение** исходному вызывающему классу **ToDo** (после получения **Намерения** список будет заполнен новыми задачами, полученными от сервера). Для инициализации

AsyncTask мы добавляем две константы в верхнюю часть класса **ToDo** и строку в **onCreate()** для вызова задачи:

```
public static final String GETADDRESS = "http://yourserver/test.php";
public static final String PUTADDRESS = "http://10.0.0.4/test2.php";
public void onCreate(Bundle savedInstanceState) {
    ...
    new SyncDataTask().execute(GETADDRESS,PUTADDRESS);
    ...
}
```

Обратите внимание на две строки, которые передаются в **SyncDataTask**, и на то, что для вызова трех методов по порядку вы пользуетесь **SyncDataTask().execute()**.

Теперь все взаимодействие с сервером осуществляется в отдельной нити, а приложение может сосредоточиться на взаимодействии с пользователем.

Индикатор выполнения процесса

У вас может возникнуть желание сообщать пользователю о том, что приложение делает в фоновом режиме. В **AsyncTask** есть удобный метод для отправки информации о ходе процесса обратно в главную нить — **onProgressUpdate()**. Чтобы им воспользоваться, сначала нужно убедиться, что второй аргумент шаблона типа **AsyncTask** был задан верно. Мы воспользуемся **CharSequence** для отображения в окне хода выполнения процесса. Метод **onProgressUpdate()** выглядит так:

```
private final ProgressDialog dialog = new
    ProgressDialog(ToDo.this);
@Override protected void onProgressUpdate(CharSequence...
    message) {
    dialog.setMessage(message[0]);
    dialog.show();
}
```

Чтобы воспользоваться этим, добавьте в любом месте класса, где должно появиться сообщение, вызов **publishProgress("сообщение")**; Диалог также нужно будет закрыть методом **dialog.dismiss()** в **onPostExecute()**.

Однако при запуске этой версии вы заметите одну проблему, связанную с этим окном: оно будет мешать задаче полностью работать в фоне, так как фокус перейдет на окно сообщения. Это может быть удобно, если вы не хотите, чтобы пользователи работали с приложением при выполнении задачи в фоне — так они видят, что приложение не зависло.

Если вы хотите, чтобы пользователи могли работать с приложением, есть другой вариант — показать индикатор выполнения процесса в заголовке окна. Это просто. Для начала добавьте эту строку в метод **onCreate()** в **ToDo.java** (перед вызовом **setContentView()**):

```
requestWindowFeature(Window.
    FEATURE_INDETERMINATE_PROGRESS);
setContentView(R.layout.main);
```

Получите настройки диалогового окна в **SyncDataTask** и добавьте в **onPreExecute()** строку

```
ToDo.this.setProgressIndicatorIndeterminateVisibility(true);
```

а в **onPostExecute()** — строку

```
ToDo.this.setProgressIndicatorIndeterminateVisibility(false);
```

(здесь не используется метод **onProgressUpdate()**, так как мы не обновляем информацию о ходе выполнения процесса, а просто показываем, что дела идут). Готово. Скомпилируйте и запустите программу, и вы увидите индикатор выполнения процесса в правом верхнем углу. Следующее, о чем нужно подумать, если вы хотите продолжить разработку приложения — что произойдет, если действие будет завершено или перезапущено в процессе выполнения? Это особенно актуально с **AsyncTask**, так как теперь у нас несколько нитей. **LXF**

Скорая помощь

Обязательно вызовите метод **dialog.dismiss()** в **onPostExecute()**, чтобы убрать окно — в противном случае вы получите ошибку "Window Leaked". Трек возникновения этой ошибки выглядит так, как будто она возникла гораздо раньше в классе, поэтому отследить ее трудно.

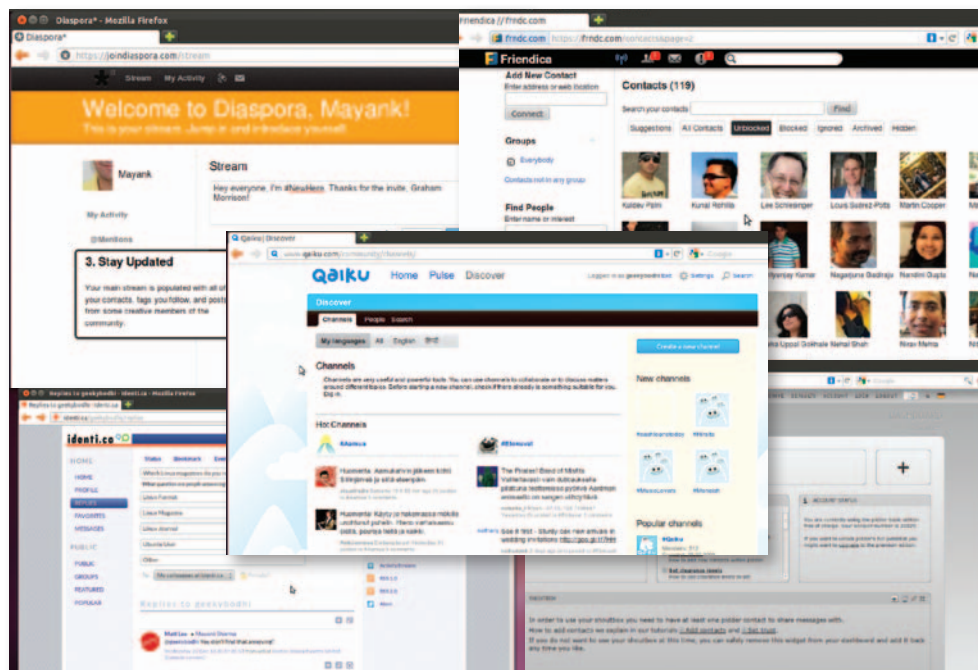
Сравнение

»» Каждый месяц мы сравниваем тысячи программ — а вы можете отдыхать!

Открытые сети общения



Забота о конфиденциальности отпугивает вас от популярных социальных сетей? **Маянк Шарма** нашел альтернативы, открытые более чем в одном аспекте.



Про наш тест...

Сравнивать эти сети друг с другом трудно — не все социальные сети одинаковы. Некоторые богаче функциями, и они позволяют сделать намного больше, чем разместить обновление, состоящее из 140 слов. Вместо этого мы сделали акцент на тестировании функций, выделяющих эти сети. Мы учитывали гибкость их управления конфиденциальностью, их способность позволять нам портировать наши данные и применение ими таких открытых стандартов, как OpenID, OStatus и пр.

И, наконец, мы протестировали удобство этих сервисов по сравнению с проприетарными сетями наподобие Facebook, Google+ и Twitter. Мы сравнивали их интерфейсы пользователя и управление индивидуальной настройкой, а также их мобильные версии и приложения.

Наша подборка

- »» Diaspora
- »» Friendica
- »» Identi.ca
- »» Pidder
- »» Qaiku

Вы не найдете Ричарда Столлмана [Richard Stallman] или Эбена Моглена [Eben Moglen] в Facebook, Google+ или Twitter. Но эти борцы за свободное ПО избегают таких сетей не просто потому, что они построены на проприетарном ПО с закрытым кодом. У них есть более серьезная проблема: их конфиденциальность, или ее отсутствие.

Немало было нападков на социальные сети из-за их политики конфиденциальности, позволяющей компаниям, стоящим за этими сетями, беззастенчиво эксплуатировать данные их пользователей. Сто-

ронники конфиденциальности долго призывали к созданию децентрализованных и распределенных социальных сетей, которые относились бы к пользовательским данным с уважением.

Рады сообщить, что такие сети стали появляться. Созданные на ПО с открытым кодом, эти сети придерживаются различных открытых стандартов переносимости данных, делая акцент на возможность взаимодействия и увеличивая степень интеграции друг с другом.

В нашем Сравнении мы рассмотрим пять социальных сетей, построенных на компонентах с открытым кодом. Фак-

тически, некоторые из них даже позволяют вам использовать эти компоненты для создания и размещения собственной сети, предоставляя вам полный контроль за вашими данными.

Однако помните, что эти сервисы лишь недавно появились на свет, и ни один из них не может похвастаться таким же количеством пользователей, как их популярные проприетарные альтернативы. Некоторые из этих сервисов, например, Diaspora, все еще находятся на ранней стадии разработки, и ограничивают регистрацию, продолжая отлаживать сервис для работы с широкой общественностью.

Функции

Что предлагают сети?

Diaspora – одна из наиболее известных альтернативных социальных сетей. Она еще находится на стадии разработки, но предлагает множество функций, полагающихся социальной сети. Она позволяет размещать текст и изображения и встраивать видео с YouTube и Vimeo, но не разрешает предварительного просмотра ссылок, хотя те и конвертируются в посты.

Пользователи могут отмечать свои посты как публичные или ограничивать их определенными группами пользователей, именуемыми Aspects. Вы также можете направлять и просматривать свои новости в Aspect. Также можно следить за другими пользователями, не добавляя их в друзья, как и в Google+ и Facebook.

Выделяет Diaspora среди других сетей то, что это распределенная сеть. Другими словами, одного сайта Diaspora в природе нет – его роль играет штат серверов Diaspora, которые называются “pods”. В качестве пользователя вы вовсе не обязаны знать о наличии pods: вы можете зарегистрироваться на любом из них и общаться с пользователями на других pods при пол-

ной прозрачности. Friendica – тоже децентрализованная сеть, подобно Diaspora, но она развивает эту концепцию еще дальше. Кроме объединения пользователей на разных серверах Friendica, она также может переносить контент и контакты из Facebook, Twitter, Identi.ca, и даже лент RSS и сайтов. А главное, можно комментировать, оценивать или делиться постами ваших друзей с Facebook изнутри Friendica.

Сервис не налагает лимита символов на обновления статуса, который, подобно посту в блоге, тоже может иметь заголовок и размещаться в разных категориях. Помимо обычной способности загружать фотографии и ссылки с YouTube и Vimeo, посты поддерживают форматирование текста и снабжены встроенными плеерами для воспроизведения аудио- и видеоссылок. Вы можете делиться и редактировать свои посты, и кроме отметки «нравится [like]», можно поставить и отметку «не нравится [dislike]».

Сервис микроблогов Identi.ca работает на сервере с открытым кодом StatusNet. Он поддерживает хэш-тэги [hashtag] и может автоматически сокращать URL.

Можно прикреплять к вашим обновлениям статуса изображения видео и аудио, а также встраивать контент



с YouTube, Flickr и других сайтов. Главное преимущество Identi.ca перед Twitter в том, что, помимо обновлений статуса, сервис позволяет поделиться закладками, назначить мероприятие, задать вопросы или провести опрос.

Зато в Qaiku, еще одном сервисе микроблогов, основное внимание уделено многоязычности. Сервер вмещает множество каналов, что отлично помогает найти других пользователей с интересами, подобными вашим.

И, наконец, Pidder отличается как от популярных проприетарных сетей, так и от соперников по Сравнению. В нем нет обычных для социальной сети функций, но он создан для защиты вашей конфиденциальности – и подходит к этому со всей серьезностью. Он шифрует все, чем вы делитесь, и в нем есть возможность расширенного управления конфиденциальностью.

Pidder также единственный сервис в Сравнении, совмещающий бесплатные учетные записи и учетные записи премиум-класса.

» Friendica снабжена всеми функциями, полагающимися социальной сети.

«Выделяет Diaspora среди других то, что это распределенная сеть.»

Вердикт

Friendica ★★★★★
 Diaspora ★★★★★
 Pidder ★★★★★
 Identi.ca ★★★★★
 Qaiku ★★★★★
 » Friendica богаче всех функциями.

Объединение данных

Как они ладят с другими сервисами?

Tрадиционные проприетарные социальные сети ведут себя, как изолированные склады информации. Более того, если вам надо соединиться и поделиться чем-то, скажем, изображениями, с вашими друзьями в другой

сети, единственный способ сделать это – зарегистрироваться в их социальной сети и продублировать данные в ней.

Хорошая открытая социальная сеть должна обладать способностью объединять данные с другими сетями. Diaspora решает часть этой проблемы, разрешая пользователям на разных pods соединяться и делиться своими данными друг с другом. При регистрации Diaspora разрешает разместить основную информацию профиля вашей учетной записи в Facebook.

Объединение данных – одна из сильных сторон Friendica, разработанной, чтобы визуализировать социальную сеть, подобно электронной почте, и давать пользователю выбор, какую сеть он хочет использовать. Friendica может объединять данные

не только с другими социальными сетями, такими, как Facebook, Twitter и Diaspora, но также брать их с сайтов и блогов.

У Identi.ca также есть интеграция с Facebook, и она может зеркально отображать обновления с Twitter, а также импортировать сообщения с лент RSS и Atom в вашу ленту сообщений. Вдобавок сервис также позволяет вам делиться новостями с вашими друзьями в Twitter.

Qaiku позволяет входить в сервис с учетной записи Facebook, и для этого сервиса не нужно создавать отдельную учетную запись. Также он умеет брать контакты из Facebook, Twitter и Gmail. Можно добавить свою учетную запись Flickr и свой блог в свой профиль Qaiku, а тот затем будет отсылать им все ваши обновления.



» Модуль Diaspora для Drupal позволяет CMS соединиться с Diaspora, как с pod'a.

Вердикт

Friendica ★★★★★
 Qaiku ★★★★★
 Diaspora ★★★★★
 Identi.ca ★★★★★
 Pidder ★★★★★
 » Friendica умеет делиться информацией со множеством сетей.

Удобство пользования

Свобода – хорошо и все такое, но готовы ли эти сети к популярности?

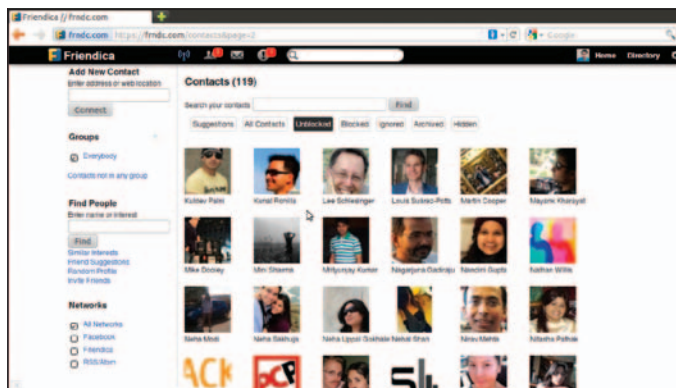
Для успеха социальной сети привлечения внимания сообщества открытого кода мало. Все сети в Сравнении должны ориентироваться не только на ученых любителей приватности, но на куда большую аудиторию людей – не-технарей.

Технологическое превосходство этих сетей привлечет пользователей с уровнем знаний и навыков, позволяющим оценить эти инновации. Но для более широкой аудитории удобство сети в ис-

пользовании играет не менее важную роль. Чтобы протестировать удобство сетей, мы подвергли их внутреннему ненаучному тестированию на бабушке. Мы показали ей эти сети и попросили ее с ними ознакомиться. Уж если бабуля освоит эти сети, то, без сомнения, и любой другой тоже сумеет! Но с другой стороны, если ей этого не удастся, никакие восхваления высоких технологий не уведут ее с любимой проприетарной социальной сети.

Diaspora ★★★★★

Diaspora и внешне, и по ощущениям очень похожа на Google+, только термины здесь другие: так, группы называются Aspects, а не Circles. Но в ней также есть хэш-теги, ставшие популярными благодаря сервисам микроблогов вроде Twitter, что очень удобно на старте. При вашей регистрации Diaspora просит указать около пяти областей интересов в виде хэш-тегов, и далее использует их для заполнения вашей начальной ленты. Есть хэш-теги для приветствия новичков и для того, чтобы помочь им с проблемами при работе в сети, а также с отчетами об ошибках и запросами на функции. Поиск людей и добавление их в Aspects несложны. Почувствовав себя одиноко, отправьте друзьям по электронной почте приглашение пообщаться. Размещая обновления статуса, можно прикрепить к ним изображение и ограничить доступ отдельными Aspects. Настройки системы [System settings] также легко доступны.



Friendica ★★★★★

Так же, как и у Diaspora, интерфейс Friendica не покажется вам незнакомым, если вы побывали хоть в одной из популярных социальных сетей. Friendica предлагает ссылку на «Советы новичкам [Tips for New members]», с указаниями на документацию, которая поможет новому пользователю разобраться с терминологией сети. Также она поясняет свои компоненты, сравнивая их с эквивалентами в Facebook и Diaspora. Если вы свяжете Friendica с Facebook, она заполнит ваш профиль информацией из профиля в Facebook. Опционально вы также можете перенести из Facebook ваши контакты и сообщения. Как и в Diaspora, можно описать свои интересы с помощью ключевых слов – Friendica потом применит их для предложений дружбы. Довольно просто объединить друзей в группы частного общения, с максимумом конфиденциальности, и настроить учетную запись, придав ей личный характер.

Клиенты рабочего стола

Оффлайн-клиенты, которые взаимодействуют с сервисом.

В популярных социальных сетях сервисов есть множество приложений для взаимодействия с друзьями. Открытые социальные сети обладают подобными же приложениями и для обычных компьютеров и для мобильных платформ. Принимая во внимание статус Diaspora «пре-бета», и тот факт, что ее API пока не готов, мы немало удивились, что здесь таких приложений уже несколько. Клиент *Dianara*, подобно сервису, который он поддерживает, пока что в стадии интенсивной разработки, и может отображать только около дюжины постов.

Есть также web-клиент Diaspora для устройств на Android, который пока не работает (на всех устройствах, включая наше).

Сеть с наибольшим количеством приложений – *Identi.ca*. Помимо нескольких специальных кросс-платформенных приложений, вроде клиента рабочего стола *StatusNet*, вы можете использовать популярные приложения, которые работают с Twitter – *Adium*, *Choqok* и *Gwibber* – для размещения постов в *Identi.ca*.

Приложения, разработанные для *Identi.ca*, годятся также для размещения сообщений во *Friendica* – например, *FrenTcl*

и клиент *Xpostulate* (кросс-платформенный); но вы не найдете их в репозитории вашего дистрибутива. Есть также простое приложение пересылки уведомлений, которое выводит на ваш рабочий стол сообщения от *Friendica*.

В *Qaiku* имеется мобильное приложение *Mauku* для Nokia N900, хотя пользователям других мобильных телефонов придется довольствоваться ее интерфейсом, оптимизированным под мобильные устройства, на m.qaiku.com. На рабочем столе можете воспользоваться популярным *Gwibber* или инструментами *Ping.fm*.

Вердикт

- Identi.ca* ★★★★★
- Qaiku* ★★★★★
- Friendica* ★★★★★
- Diaspora* ★★★★★
- Pidder* ★★★★★

» *Identi.ca* – явный победитель в этой номинации.

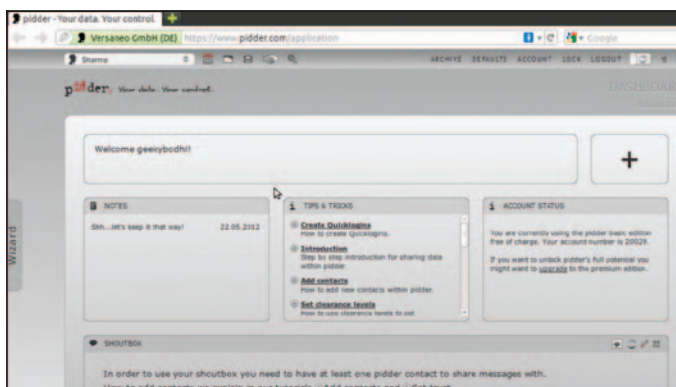


Identi.ca ★★★★★

В отличие от двух предыдущих, Identi.ca – сервис микроблогов, и на первый взгляд ничем не отличается от Twitter. Она позволяет делать все, что обычно – просматривать сообщения в контексте, отвечать на них, повторять их и отмечать как избранные. Также позволяет помечать свои заметки как конфиденциальные, или делиться ими со всеми остальными в Интернет или только в Identi.ca. Интерфейс облегчает поиск групп по интересам или создание собственных групп. Но сервис также позволяет вам делать намного больше, чем просто размещать посты длиной в 140 слов. Несложный интерфейс помогает создавать и делиться событиями и закладками со своими друзьями. Сервис располагает и более прямыми способами узнать мнение окружающих, например, инструментами для простого опроса с пятью градациями оценок.

Qaiku ★★★★★

Еще один сервис микроблогов, Qaiku, обладает интерфейсом, подходящим и для рабочего стола, и для мобильных устройств. Его главное преимущество – упор на многоязычность. Здесь имеются настройки, позволяющие менять язык интерфейса и указывать известные вам языки, что, впрочем, не вполне понятно по его стартовой странице. В сервисе нет никакой документации, хотя это не особенная помеха: большинство функций не требуют пояснения. Qaiku также позволяет легко просматривать обсуждения, в которых вы принимали участие. Можно также сортировать публичные сообщения по известным вам языкам. Сервис позволяет без проблем загружать фотографии, добавлять тэги к вашим заметкам или размещать ссылки с сайтов вроде YouTube, Slideshare и прочих. Поиск и соединение с другими пользователями или каналами в нем тоже не отличаются сложностью.



Pidder ★★★★★

Как будет ясно из оставшейся части данной статьи, Pidder – не повседневная социальная сеть и подходит далеко не всем. Повторим, с помощью Pidder можно упорядочить и зашифровать свои важные данные, и поделиться с другими онлайн-сервисами. Также он хранит информацию, которую вы передали внешним сторонам, под вашей зашифрованной учетной записью на сервисе. Однако прекрасное управление конфиденциальностью приносит в жертву удобство: у Pidder самый сложный интерфейс из всех социальных сетей в нашем Сравнении. Отчасти спасает хорошая документация и удобный мастер, минимизированный в левой стороне интерфейса. Вам не удастся воспользоваться этим сервисом, не потратив времени на изучение его обширной документации или просмотр видео, чтобы разобраться с основными задачами – как делиться данными или добавлять контакты, создавать quicklogins и т.д.

Параметры и настройки

Тонкая настройка служб.

Хотя все эти сервисы полностью готовы к работе, они предоставляют возможность индивидуальной настройки различных аспектов под ваши потребности. Diaspora позволяет настроить список действий, о которых вы бы хотели получать уведомления, а также соединяться или разрывая соединение со сторонними сервисами типа Facebook, Twitter и Tumblr.

Поскольку команда пока штурмует разработку ядра сервиса, опции для оптимизации интерфейса пользователя отсутствуют. А вот во Friendica 13 тем с на-

страиваемыми элементами, например, размером шрифта для постов и комментариев. Можно также скрыть некоторые элементы, такие, как страницы сообщества, последние отметки “like”, и т.д. Помимо этого, Friendica обладает расширенными настройками безопасности и конфиденциальности, и чтобы уменьшить спам, можно ограничить число ежедневных частных сообщений от неизвестных вам людей.

Конфиденциальность – сильная сторона Pidder. Сервис позволяет менять пароль и затем заново шифровать все свои данные уже с новым паролем. Identi.ca

использует собственный сервис сокращения URL – ur1.ca, но его можно изменить в Settings. Учитывая внимание, уделяемое многоязычию в Qaiku, вы можете выбрать язык для своего пользовательского интерфейса, и основной язык, который будет в тэгах ваших обновлений статуса. В Qaiku можно создать тему с собственным фоновым изображением и цветами, выбранными вами лично. Более того, его настройки конфиденциальности позволяют отключить функцию Location [Местоположение], которая показывает на карте ваши перемещения людям, читающим ваши новости.

Вердикт

- Friendica ★★★★★
- Diaspora ★★★★★
- Identi.ca ★★★★★
- Pidder ★★★★★
- Qaiku ★★★★★

» Вообще-то в настройке особой нужды нет.

Управление конфиденциальностью

Может ли Большой Брат подслушать?

Эти сети существуют, чтобы защитить ваши данные от любопытных глаз желающих воспользоваться вашими данными для низких целей. Благодаря распределенному дизайну Diaspora, здесь нет централизованного управления, способного отслеживать ваши учетные записи или указывать вам, чем вы должны поделиться.

Friendica защищает пользователей шифрованием данных при передаче от сервера к серверу. Все пункты в ее сети управляются списками доступа, которые запрещают или разрешают доступ к ним отдельным личностям или определенным группам пользователей.

В большинстве сетей также имеются группы для частных бесед, которые ограничивают общение только своими членами. Friendica зашла еще дальше, позволив настраивать «срок хранения» для контента, по истечению которого он удаляется со всех серверов Friendica, где только может храниться его копия.

Большинство сетей также поддерживают собственную обмена частными сообщениями от одного пользователя к другому. Во Friendica пользователи могут отправлять сообщения любому пользователю Friendica или даже Diaspora, включая и тех, кто находится на других сайтах.

Некоторых может удивить, что в Identi.ca нет защищенного режима, но эта сеть

не ограничивает доступ к контенту. Фактически, здесь открыто заявляется, что все посты в сети публикуются под лицензией Creative Commons Attribution licence.

Если вас заботит приватность, советуем прочитать политику конфиденциальности Identi.ca, где ясно указано, что она может использовать некоторые из ваших частных данных для публикации сводных показателей (например, «30 % наших пользователей зарегистрированы на адреса Hotmail») и обязана предоставлять вашу частную информацию по судебному ордеру. Qaiku же предлагает простые и понятные функции конфиденциальности, позволяющие выбрать, кто будет видеть ваши посты, а кто нет. Опций, связанных с конфиденциальностью, две: одна показывает или скрывает ваше местоположение, а другая помогает вам решить, хотите ли вы показать свои сообщения всем в Интернете, или только пользователям Qaiku, или только вашим друзьям, или тем, кто указан в ваших контактах.

Более того, этот сервис дает вам расширенный контроль над вашими контактами – и в Qaiku, и в других сетях, например, в Twitter и Facebook. Настройки конфиденциальности в обсуждении, начатом кем-то другим, отображаются в боковой панели рядом с лентой. Можете использовать эту информацию, чтобы выяснить, кто сможет увидеть ваши комментарии, и затем



решить, хотите вы принять участие в этом обсуждении или нет.

Обеспечение конфиденциальности – главная задача Pidder. Он использует собственную внутреннюю библиотеку pidCrypt с открытым кодом, написанную на JavaScript, для шифрования ваших данных, которые переносятся на серверы через HTTPS. В отличие от других сетей, управление идентификацией пользователей является центральным пунктом для Pidder, и он позволяет создавать учетные записи и псевдонимы для работы с данными на ваше усмотрение.

Вы также можете настроить шесть разных уровней допуска для своих контактов в сети. В сочетании с функциями управления идентификацией это дает превосходные возможности контролировать, какой пользователь какой именно информацией и с кем именно делится.

» В своей политике приватности Pidder заявляет, что его сервисы могут применяться при полной анонимности.

Вердикт

Pidder	★★★★★
Friendica	★★★★★
Diaspora	★★★★★
Qaiku	★★★★★
Identi.ca	★★★★★

» Конфиденциальность – высшее для Pidder.

Переносимость данных

Кто владеет вашими данными?

Помимо управления конфиденциальностью, возможность распоряжаться своими данными и своей учетной записью является еще одним козырем открытых сетей.

В частности, Diaspora позволяет пользователям скачивать свои текстовые данные и фотографии. Также имеется очень доступная ссылка для закрытия учетной записи. Помните, что, в соответствии с дизайном Diaspora, данные, которыми вы делитесь с кем-то на другом роде, хранятся именно на нем, вместе с копией вашего профиля. Поскольку они будут доступны администратору базы данных этого рода, выбирайте род Diaspora тщательно. Однако при удалении поста или данных профиля они удаляются с вашего рода и с лю-

бых других родов, на которых они ранее хранились. Кроме того, команда Diaspora работает над функцией, которая позволит экспортировать ваши данные с одного рода и импортировать их на другой.

Friendica тоже позволяет скачивать свои личные данные и удалять свою учетную запись. В Identi.ca можно скачивать свои заметки и файлы в форматах, пригодных для использования другими сервисами, вместе с вашей информацией профиля.

В Qaiku, как это ни странно, нет инструментов для скачивания ваших данных или удаления учетной записи! Pidder можно использовать также в качестве онлайн-менеджера паролей, так что он позволяет импортировать и экспортировать данные в нескольких форматах, напри-

мер, pidder-XML, CSV, KeePassX, и т.д. Если вы удалите свою учетную запись, сервис вычистит все ваши данные по истечении двух недель.



» Identi.ca экспериментирует со своей функцией резервного копирования, и не копирует частную информацию.

Вердикт

Identi.ca	★★★★★
Diaspora	★★★★★
Friendica	★★★★★
Pidder	★★★★★
Qaiku	★★★★★

» Странно, что в Qaiku нет функции экспорта данных.

Открытые социальные сети

Вердикт

О Diaspora, безусловно, чаще всего говорят, как о замене Facebook с открытым кодом, которая уважает вашу конфиденциальность. Эта сеть произвела на нас сильное впечатление, несмотря на альфа-стадию ее разработки, и мы рады бы присудить победу Diaspora.

Но – не присудим. Сервису не хватает жизненно важных функций, таких, как IM-чат, чтобы вы могли общаться с друзьями прямо в браузере. Разработчики постоянно добавляют новые функции, так что ждем эту функцию в будущей версии.

Затем есть еще и тот факт, что регистрация на главном роуд, где размещаются разработчики, покамест ограничена. Но вы можете зарегистрироваться на других роуд'ах, или настроить собственный.

Итак, победитель Сравнения – Friendica. Это, вероятно, лучшая объединенная социальная сеть из неизвестных вам. Она не столь шумная, как Diaspora, но довольно популярна, и, кажется, в ней не меньше пользователей, хотя у нас не было возможности подсчитать их точ-

ное количество. Во Friendica имеются все функции, которые есть в вашей любимой проприетарной социальной сети, и у нее достаточно интуитивный интерфейс. ПО Friendica поддерживает Facebook-подобный чат в браузере через XMPP с модулем расширения. Но модуль должен активироваться администратором вашего сайта Friendica. К сожалению, на данный момент он не активируется на frndc.com.

Хотя Friendica годится в замену любой социальной сети, которой вы в данный момент пользуетесь, для узких целей она может быть перебором. Если вам всего и нужно, что сервис микроблогов, обратитесь к [Identi.ca](http://identi.ca) и не ищите ничего другого.

Это полная замена Twitter, и она даже позволяет размещать собственный сервис в контролируемой среде – скажем, транснациональной корпорации или образовательном учреждении. И хотя многоязычие не является коньком [Identi.ca](http://identi.ca) в той же степени, как [Qaiku](http://qaiku.com), она поддерживает заметки на языках



и кроме английского. И ее инструменты удобны для изменения темпа обсуждения.

А вот Pidder – статья особая. Мы не можем рекомендовать его в качестве повседневной социальной сети, да он и не для этого. Он предназначен тем, кто ценит свою приватность, и лучше всего подходит для обмена действительно важной информацией, которую вы не хотите предавать гласности.

➤ Механизмы защиты конфиденциальности у Friendica даже лучше, чем у Diaspora.

«Friendica — это, вероятно, лучшая открытая сеть из неизвестных вам.»

I Friendica ★★★★★
 Сайт: www.frndc.com
 » Ваша новая любимая социальная сеть.

IV Qaiku ★★★★★
 Сайт: www.qaiku.com
 » Хороший сервис микроблогов, особенно для полиглотов.

II Diaspora ★★★★★
 Сайт: www.joindiaspora.com
 » Пока не полная замена вашей любимой сети, но почти.

V Pidder ★★★★★
 Сайт: www.pidder.com
 » Идеален для параноидов. Для остальных чересчур сложен.

III Identi.ca ★★★★★
 Сайт: www.identi.ca
 » Лучший открытый сервис микроблогов. Точка.

Обратная связь
 Вы пользовались этими сетями? По вашему мнению, они готовы расцвести? Поделитесь своими мыслями с нами: ixf.letters@futurenet.co.uk

Рассмотрите также...

Вы удивитесь, но программ с открытым кодом для создания собственной социальной сети очень много: это Elgg, LovdByLess и т.д. На них, а также на движках сетей из нашего Сравнения основано немало нишевых сетей. Так, Benticio применяет StatusNet, движок [Identi.ca](http://identi.ca). Модули расширения превращают в социальные сети ряд популярных систем управления контентом – такие, как Yagert для Xoops. Можно создать свою сеть на базе открытого ПО типа

RetroShare и Newebe и сотрудничать с друзьями, не выдавая никакой информации чужим. Newebe (пока на бета-стадии) имеет клиент микроблогов и позволяет делиться изображениями и заметками, а RetroShare – кросс-платформенная, и все обсуждения шифруются GPG и OpenSSL. А инициативы AppleSeed Project, OneSocialWeb и Anahita разрабатывают стандарты и инструменты для открытых социальных сетей. И есть еще проект FreedomBox от FSF. **LXF**



Шляпы долой перед

Red Hat

Джонатан Робертс выясняет, как крупнейшая в мире компания открытого ПО заработала миллиард, сделав свободу предметом первой необходимости.

Red Hat достиг важного рубежа: на конец прошлого финансового года они получили доход, превысивший \$1,000,000,000 (то есть миллиард долларов). Эта цифра сама по себе шок, но еще интереснее, как умудриться заработать такие деньги?

Многие из успешных компаний, работающих в индустрии ПО, смогли занять и удерживать свою долю рынка благодаря действиям, которые в лучшем случае были сомнительными, а в худшем — незаконными. Сокрытие технологий; плановый моральный износ; а также пресловутое «затянуть, захватить, загасить» [embrace, extend, extinguish — фраза, которой корпорация Microsoft отражала свою политику, — прим. пер.] — вот

стратегии, применявшиеся разными компаниями, которые также замедляли темп инноваций, увеличивали себестоимость и урезали свободу пользователя. Red Hat, с момента ее появления в начале 1990-х, воротила нос от подобных практик. Вместо этого, как заявил во вступительной части Отчета за 2003 год Мэтью Шулик [Matthew Szulik], бывший председатель и руководитель компании, Red Hat позиционирует себя «в центре отношений, связывающих потребителя, сообщество и партне-

«Red Hat ведет бизнес иначе. По крайней мере, такое ощущение есть.»

ров», чтобы все оставались в выигрыше. Red Hat ведет свой бизнес иначе.

По крайней мере, такое ощущение есть. Мы попробуем разобраться, что же это значит на самом деле — как Red Hat зарабатывает деньги на свободном ПО, какая часть ее репутации заслуженна, как она добилась успеха и где окажутся компания и движение открытого кода в будущем. Мы побеседовали с Джимом Уайтхерстом [Jim Whitehurst], руководителем компании с 2008 года, и Майклом Тиманном [Michael Tiemann], вице-президентом по делам открытого кода и основателем Cygnus Support, компании, занимающейся свободным ПО, которая зарабатывала миллионы еще до того, как появился Linux.

Происхождение

Начнем мы с происхождения Red Hat, поскольку это отнюдь не история внезапного успеха. Как и у всех свободных программ, ее корни уходят в 1980-е годы, к Ричарду Столлмену [Richard Stallman], GNU Project, Free Software Foundation, и к Линусу Торвалдсу и ядру Linux.

После создания этих основных ингредиентов многие начали создавать и распространять свободное ПО, чтобы работать с ними. При наличии времени и навыков можно было собрать все составляющие и создать свободную UNIX-подобную операционную систему для обычных ПК. Сейчас мы воспринимаем это как должное, но в конце 1980-х большинство корпораций работало на системах UNIX, которые стоили более 10 000 дол-

пускать все, что создавали, как программы с открытым кодом, также распространяя их свободно. Первой задачей Юинга после слияния двух компаний стала переделка Red Hat, чтобы упростить ее загрузку с сайтов FTP по всей сети.

Прагматичный идеализм

Здесь не было никакой идеологической подоплеки, мотивированной глубокой верой в то, что ПО должно быть свободным, как у Столлмена, когда он запустил GNU Project. Юинг не планировал заниматься свободными программами, а Янг начал, распространяя и свободные, и коммерческие программы – лишь бы заработать денег.

При этом важность свобод, гарантированных лицензиями GNU, мимо них не прошла: последние пару десятилетий оба говорили о воздействии, оказанном идеологией свободного ПО на их ранние усилия.

На поверхностный взгляд, без свободного ПО их предприятие просто бы не состоялось, из-

за дороговизны проприетарного UNIX. Но если копнуть глубже, оба сознавали, что происходит нечто большее.

С технической точки зрения, Юинг оценил преимущества методологии открытого кода. В том же интервью для [Salon.com](#) он сказал: «...как программист, имеющий академическую подготовку и работавший с программами с открытым кодом... я знал, что это намного более эффективный и производительный способ разработки ПО».

Если вы работаете над проприетарной программой и вам нужна помощь, то вы явно влип-

ли – ведь нельзя взять исходный код и показать кому-то, чтобы попросить помощи в решении проблемы, не обложившись как минимум несколькими юристами. С открытым кодом вы можете просто выйти в Интернет и показать код кому угодно. Вы быстро получите и полезный и точный ответ, и удовольствие от сотрудничества с другими людьми.

Бизнес-модель

Янг осознавал важность открытого кода, выбирая бизнес-модель, которая позволила бы им зарабатывать деньги на свободном ПО. В книге *Open Sources: Голоса революции открытого кода* он объяснял, что «надо фокусироваться на размере рынка, а не на его доле... Чем больше пользователей Linux в целом, тем больше пользователей будет у Red Hat для своей продукции».

Это наблюдение означает, что предпосылка успеха Red Hat – успех соперничающих с ним дистрибутивов Linux, и лицензии на ПО с открытым кодом помогли быстрее внедрить их в жизнь. Когда Red Hat улучшала какую-то программу, все ее соперники-дистрибутивы, в том числе и основанные на ее продуктах, получали от этого преимущество; когда другие дистрибутивы улучшали какие-то программы, Red Hat и ее клиенты тоже получали от этого преимущество. Как сказал Янг: «Производитель программ с открытым кодом начинается с качественного продукта. Фокус в том, чтобы найти эффективный способ зарабатывать деньги, предлагая своим клиентам преимущества открытого кода». Этим трюком Red Hat овладела в совершенстве, и Уайтхерст объяснил нам, как она использует его сегодня.

«Тогда работа в Linux не была такой гладкой, как сегодня.»

ларов. Наличие свободной альтернативы стало истинным освобождением.

По крайней мере, так считал Марк Юинг [Marc Ewing]. Окончив университет, он основал софтверную компанию под названием Red Hat. Сперва он начал разрабатывать приложения для UNIX, но не мог себе позволить покупку терминала; как раз в это время появился Linux, причем бесплатный, так что он этим воспользовался для создания собственного приложения.

Однако в те времена Linux был не так отлажен и отнюдь не готов к работе с ходу, как сейчас. В интервью для [Salon.com](#) в 1999 году Юинг сказал, что «за несколько последующих месяцев... я обнаружил, что трачу свое время не на работу над собственным проектом, а на отладку Linux... Я вдруг осознал, что на самом деле должен собрать дистрибутив Linux лучшего качества. Так я и поступил».

Приблизительно тогда же Боб Янг [Bob Young], еще один предприниматель в области технологии, основал ACC Corporation. Он ушел со своей работы и начал работать в швейном закутке жены, где и публиковал информационный бюллетень *New York UNIX* и каталог ПО под названием *ACC PC Unix and Linux Catalog*.

Когда пошли разговоры об октябрьском – «Halloween» – бета-релизе Red Hat, дистрибутива Юинга, Янг позвонил ему, чтобы договориться о внесении этого дистрибутива в каталог. Они быстро выяснили, что их таланты – технические способности Юинга и знания Янга в финансах и маркетинге – идеально сочетаются. В 1995 году Янг купил компанию Юинга, объединил ее с собственной, и они оба засели за работу под брендом Red Hat Software. Поскольку изначально их бизнес строился на свободно распространяемых программах с открытым кодом, они продолжили вы-



» Джим Уайтхерст возглавил Red Hat в качестве CEO в 2008, сменив Мэтью Шулика.

«Сила открытого кода как модели разработки в том, что он разбивает проблему на более мелкие части, и вы получаете итеративные и быстрые улучшения. Это превосходно для устранения ошибок и добавления функций и функциональности, но является просто катастрофой для работы в центре обработки данных, потому что изменения в большом количестве – это последнее, что им нужно... Если взглянуть на обычные дистрибутивы Linux, они иногда изменяются ежедневно.

Что мы делаем – и это одна из основных ценностей, предоставляемых нами нашим клиентам – каждый два года мы замораживаем спецификацию, и создаем версию Red Hat Enterprise Linux, и принимаем на себя обязательство по его поддержке без нарушений двоичной совместимости в течение 10 лет».

Это вовсе не простая задача. Каждый релиз Red Hat Enterprise Linux ведет команда из более чем 100 инженеров, сопровождающих этот релиз на протяжении всего его 10-летнего жизненного

цикла. Они портируют исправления ошибок и обновления системы безопасности из новых программ назад в старые, изначально бывшие в по-ставке, не нарушая двоичной совместимости.

Естественно, сюда же входит и прочая работа, включая техподдержку, обеспечение документации и т.п., но в центре бизнеса Red Hat – обеспечение традиционным предприятиям доступности всей силы и мощи открытого кода, работа над его стабильностью, чтобы в него оправданно было делать солидные вложения.

Катализатор

Хотя вышеописанный сценарий представляет основной бизнес Red Hat, все же, согласно утверждению Уайтхерста, некоторые клиенты ценят именно ее тесную связь с сообществом открытого кода.

«Самые яркие примеры – когда мы работали с ВМФ, встраивая ядро реального времени в предшествующие проекты, или с Агентством национальной безопасности, встраивая SELinux в их ПО. В этих примерах мы работали непосредственно с компаниями над приоритетами, которые имелись у них раньше».

«Они портируют исправления ошибок из новых программ в старые.»

► Майкл Тиманн основал Cugulus Solutions, разработал часть C++ для GCC и сейчас является вице-президентом Red Hat по делам открытого кода.



У этих компаний нет необходимых знаний или опыта для эффективной работы с ранее вышедшими проектами. Они не понимают, как работает общественный контроль над исходником или отчеты об ошибках, как работают списки рассылки и каковы стандарты кода в каждом сообществе. А вот у Red Hat есть и подобный опыт, и знания. Более того, Red Hat является такой же компанией, как и их компании, и она умеет разговаривать с ними на их же языке и стать мостом, соединяющим эти два мира.

Это важно не просто потому что обеспечивает их еще одной клиентской базой – теми, кому нужно интегрировать новые функции – но и потому, что делает их бренд сильнее в глазах тех, кто заинтересован только в использовании программ с открытым кодом. В изначальном описании бизнес-модели Янг сказал, что больше всего их работа напоминает производство потребительских товаров, например, бутилированной воды или кетчупа. В том смысле, что любой может создать свой собственный кетчуп из имеющихся под рукой ингредиентов, не нарушая при этом никаких законов, однако никто такого не делает.

В чем-то это удобство – кому охота самому замешивать кетчуп? Но это не объясняет, почему Heinz захватил такую долю рынка. Heinz стал



► Открытый код и открытые репозитории произвели революцию в разработке программ и сделали конкурентоспособной относительно небольшую компанию Red Hat.

С днем рождения, RHEL!

Следуя за Red Hat Linux, Red Hat Enterprise Linux стал основным продуктом. В этом месяце исполняется десять лет со дня его первого релиза.

Основная разница между этими двумя продуктами в том, что скомпилированный и готовый двоичный дистрибутив RHEL не раздается бесплатно (хотя его исходный код по-прежнему находится в свободном доступе). Более того, компания начала сокращать частоту релизов, но увеличивать период поддержки своих продуктов. В самой компании изменения восприняли без энтузиазма. Тим Берк [Tim Burke], вице-президент Linux engineering, сказал, что «в компании хватало тех, кто говорил:

“Вы, ребята, нас убиваете, мы уходим из бизнеса, вы просто не ведаете, что творите”, и в культурном смысле это был истинный прыжок в неведомое». Некоторые инженеры ушли из компании в связи с этим решением, потому что они действительно хотели продолжать работать над Red Hat Linux.

Решение было смелым, но оно себя оправдало. Благодаря более долгому циклу релизов и прицельному вниманию Red Hat сумела привлечь множество крупных и влиятельных клиентов. Тиманн рассказал нам, что в 2003 году, году перехода, Red Hat была в состоянии делать «потрясающие вещи на Уолл-стрит, которые в итоге привели к тому, что Linux

стал самой надежной платформой для корпоративных приложений. Мы действительно смогли стать сплывающим фактором». В том же году восемь из десяти ведущих инвестиционных банков работали на системах на базе Red Hat.

Хотя этот переход мог снизить привлекательность Red Hat как настольного Linux, он не лишил других производителей возможности использовать плоды своего упорного труда. CentOS и Scientific Linux, популярные проекты, управляемые сообществом, перекомпилируют исходные коды Red Hat Enterprise, давая преимущество предприятиям малого бизнеса и научным учреждениям.

доминировать на рынке, где каждый потенциально способен создать конкурентоспособную альтернативу, потому что смог создать сильный бренд – Heinz для многих синоним кетчупа.

Янг был убежден, что Red Hat может доминировать на рынке, где каждый может скопировать ее продукт, создав бренд, равный по силе бренду Heinz. Чего и добились поставкой качественного кода, подкрепленного 10-летней поддержкой; но не в меньшей степени – созданием веских оснований для претензии на роль идейного лидера в мире открытого кода, который в курсе всего происходящего и в деталях знает весь процесс разработки.

Конечно, сообщества, с которыми работает Red Hat, являются независимыми организациями с собственными целями и желаниями – и собственными функциями, которые они хотят

добавлять. Не обходится без ситуаций, когда цели сообществ вступают в конфликт с целями Red Hat и ее партнеров; и мы спросили у Тиманна, что в таких случаях происходит.

«Мы делаем шаг назад и задаемся вопросом, что послужило причиной подобного противоречия; и нельзя ли создать новые условия, в которых этого конфликта бы не возникло? Это чаще срабатывало, чем нет.

И мы всегда смотрим на искомый компонент – а это может быть технология, способ организации пакетов, тестирование или решение, принятое руководством; но мы всегда смотрим на Fedora, как на единое целое, вплоть до самого кода, и спрашиваем: «Где узкое место на данный момент?», и расширяем его».

Значительные изменения

В случае Fedora можно видеть, что все эти разнообразные компоненты сообщества претерпели солидные изменения с момента ее появления. Самое примечательное в том, что еще за год до этого существовали отдельные репозитории кода,

куда каждый мог вносить свой вклад, и даже тогда это было в форме Extras, существовавших отдельно от основного дистрибутива.

С течением времени Extras слились с Core, и теперь любой может вносить посильный вклад в любую часть операционной системы. В самом начале это стало источником трений между Red Hat и сообществом (чтобы разобраться в причинах, загляните на <http://lwn.net/Articles/83360>), но эта проблема уже улажена.

Эмоциональная победа

Когда слушаешь Тиманна, Уайтхерста и прочих из Red Hat, легко поддаться их энтузиазму относи-



► Red Hat — главный спонсор Fedora Project, дистрибутива сообщества, находящегося на переднем крае свободного ПО.

тельно компании. Свои деяния они описывают так эмоционально и захватывающе, что невольно понимаешь, что их миссия – сделать этот мир лучше.

Вот что Тиманн говорит о ситуациях, приведших Red Hat к самым большим успехам:

«У нас не просто была солидная рациональная база – а лучшая производительность за более низкую цену является весьма рациональной базой – но мы также смогли создать эмоциональный настрой, способный заставить всех осознать преимущества для своей карьеры, для своих друзей, для своих сверстников... открытый код – это не просто аргумент, это не просто значит быть умнее других. Это значит привлечь других и помочь им добиться успеха в своей деятельности.

Когда я пришел в Red Hat, среди прочего меня впечатлила философия Red Hat: даже в ситуации

«Любой может вносить посильный вклад в любую часть ОС.»



► Fedora Core 3, Gnome 2.8 и KDE 3.3.

Майкл Тиманн о возможностях

Во время беседы с Тиманном мы спросили его, с какими трудностями может столкнуться открытый код в следующие пять – десять лет. И первым делом он сказал, что оборотной стороной трудности является возможность.

«Последнее, чему я стараюсь учить всех приходящих в Red Hat, это цитата из Ганди. Он сказал: «Пусть вы делаете неважную работу – важно, что вы ее делаете». Я осознал, что открытый код дает механизм принятия содействия от каждого, независимо от его значимости и величины, делая каждого крайне важным.

В мире открытого кода человек, работающий над J2BE, может перейти на Apache и исправить там что-то, и наоборот. А следствие таково, что десятки тысяч незаметных, крошечных инноваций ведут к созданию более понятной, более стабильной, быстрее адаптируемой платформы...

Так что насчет трудностей, с которыми сталкивается открытый код, я бы в первую очередь сказал, что благодаря этой развивающейся инновационной модели возможности настолько огромны, что подлинная битва – это даже не битва – подлинный контекст, в котором мы находимся, тот, что открытый код – единственный настоящий практический способ не отстать от быстро меняющегося мира».

Говоря о самих трудностях, Тиманн, как ни странно, не упомянул среди них патенты на ПО или иные законодательные реформы. Он сказал, что это – общие трудности, поскольку они «оказывают давление на то, что Бен Франклин, Джон Адамс и, черт возьми, Адам Смит изначально описывали, как свободный рынок». Настоящая проблема в понимании Тиманна, в том, чтобы создать ПО, отвечающее требованиям 2012 года, «когда каждый триллион долларов имеет значение»

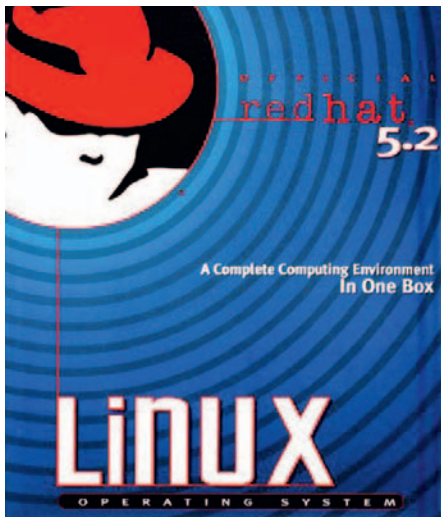
и когда БРИК (Бразилия, Россия, Индия и Китай) имеют тенденцию к росту вплоть до экономического превосходства.

Он подсчитал, что 500 миллиардов долларов из суммарных затрат всех предприятий на ИТ ежегодно тратятся впустую из-за огромного количества приложений, отвергнутых до внедрения, или тех, которые запоздали, или не обладают нужными функциями, или имеют дефекты, снижающие производительность.

Поскольку открытый код «создает лучшие программы, которые быстрее совершенствуются и имеют более низкую стоимость, мы можем обеспечить эффективность на каждый потраченный доллар, которая будет в четыре – десять раз выше... нужно только дать людям понять, что технология – превосходная вещь, а не просто грабеж 30 центов с каждого доллара, на нее потраченного».

конкурентных продаж мы не рассматривали ее как игру с нулевым исходом... мы изо всех сил стараемся делиться лучшими достижениями, ведь чем больше людей это понимают, тем больше наш успех, которого мы добились как часть просто на дрожжах растущего пирога».

История вклада Red Hat в сообщество открытого кода добавляет этим словам вес. Уайтхерст сказал нам, что «У нас есть 40 человек, работающих в команде настольных систем, но на самом деле мы не продаем потребительских настольных систем, их доля в нашем бизнесе очень мала. Эта доля определенно не оправдывает работу аж 40 человек. Мы занимаемся этим только по единственной причине: это важно для сообщества



Хотите — верьте, хотите — нет, но Red Hat раньше продавала программы в коробках.

открытого кода, это важно для разработчиков, занимающихся другими значительными компонентами Linux».

Собрав все эти факты вместе, легко вообразить, что у деятельности Red Hat гораздо больше общего с благотворительностью, чем с традиционным бизнесом.

Red Hat == Microsoft

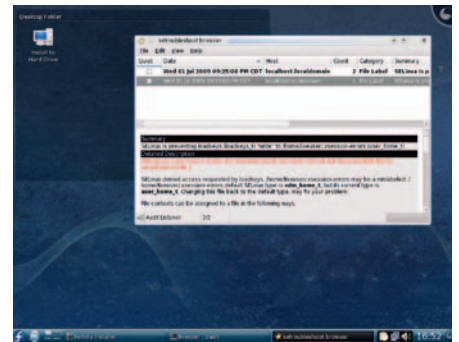
Но если вы почитаете финансовые документы Red Hat или рассмотрите ее историю, станет очевидна и другая сторона компании. В случае финансовых документов, мелькание фраз типа «использование сообщества с открытым кодом» мигот отгоняет мысли о филантропии и навевает идею о нещадной эксплуатации добровольцев ради зарабатывания кругленьких сумм для Red Hat и ее акционеров.

Да и в истории Red Hat два случая отлично иллюстрируют, почему некоторые столь скептически настроены по отношению к ней.

В первом случае Red Hat версии 7 вышла с GCC 2.96, этапом разработки на пути к GCC 3.0. Именно эта версия компилировала код, несовместимый с более ранними версиями и с финальной версией, когда та вышла.

Многих пользователей это раздосадовало, поскольку привело к появлению множества ошибок в операционной системе. Код, предназначенный для более старых версий GCC, не компили-

«Пакеты, к которым люди привыкли, не работали.»



Федора 11 (2009) отличается 20-секундной загрузкой, ext4 и Firefox 3.5.

ровался, и пакеты, к которым люди привыкли, не работали с новой ОС.

Многие усмотрели в этом происки темных сил. За два года до этого Red Hat трудилась, не покладая рук, привлекая партнеров — Oracle, SAP, Novell, HP, Compaq и Dell все согласились оказать поддержку Red Hat Linux, новому компилятору и всему прочему.

Желая разделить успех Red Hat, другие разработчики тоже должны бы были переключиться на новый компилятор, против воли команды разработки GCC. Это, конечно, был отнюдь не конец света — другие дистрибутивы могли использовать тот же компилятор, поскольку и он сам и весь его код был открытым — но этот шаг вносил раздор, и выглядело это очень похоже на тактику Microsoft по защите своих рынков. Более того, у Red Hat наверняка имелись ресурсы для работы со всеми этими программами, которые не компилировались, а у других дистрибутивов их не было.

Джим Уайтхерст об облачных вычислениях

Уже некоторое время темп роста Red Hat составляет более 20 % в год. Мы поинтересовались у Джима Уайтхерста, долго ли это, по его мнению, будет продолжаться и чем будет обусловлен их рост в будущем.

«По поводу перспектив нашего роста я настроен оптимистически. Я полагаю, наш рост опирается на два столпа. Один — то, что мы по-прежнему находимся на ранней стадии этой парадигмы в компьютерных технологиях. Когда я говорю о сдвиге парадигмы, я подразумеваю этот переход с клиента-сервера, «толстых» клиентов, на децентрализацию крупных центров обработки данных и быстрый рост числа небольших устройств, будь то мобильные устройства, или основанные на браузерах, или еще бесчисленное множество других типов устройств.

В этой парадигме Linux повсеместно выбирается в качестве операционной системы. Почти все облака, кроме Azure, работают на Linux, используют виртуализацию с открытым кодом и программы-

посредники с открытым кодом. Мир идет к нам. Так что я думаю, в более широком смысле... я бы озвучил это так: мы хотим, чтобы открытый код стал выбором по умолчанию для следующего поколения компьютерной инфраструктуры, так же, как Microsoft был выбором по умолчанию в прошлом.

...и поскольку эта парадигма продолжает развиваться, я полагаю, что Red Hat обладает огромным потенциалом среди существующих продуктов для быстрого роста. Нам еще далеко не конец».

Нас удивило, что глава компании с открытым кодом так свободно говорит об облаке. В конце концов, Эбен Моуглен [Eben Moglen] все последние годы говорил о вреде облака для пользовательской свободы и безопасности данных — во многом оно кажется антитезой программ с открытым кодом. Уайтхерст отверг это предположение: «Централизация вычислений — одна из лучших вещей, происшедших с открытым кодом... возьмите простую вещь, Hadoop. Все, что происходит в мире больших данных, теперь сначала происходит в открытом

коде — а не в лабораториях IBM или Oracle. Причина в том, что у компаний типа Google, Amazon, Facebook, Yahoo есть технические проблемы, решаемые с помощью открытого кода.

У меня меньше проблем с конфиденциальностью, поскольку здесь нет ничего общего с конфигурацией и с централизацией. Куда больше общего здесь с контрактами — на которые люди соглашались — на использование некоторых из этих сервисов. По сути, нет ничего страшного в том, что ваши данные размещаются в облачном сервисе, то есть что другие смогут увидеть эти данные.

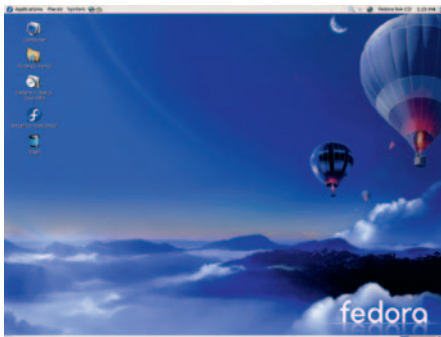
Сама по себе архитектура, по мнению некоторых, могла бы быть безопаснее, потому что можно найти людей, которые сделали бы шифрование и обеспечили безопасность на военном уровне. Так что я понимаю точку зрения Эбена, но для меня это — не проблема архитектуры, эта проблема бизнес-модели: люди решаются променять свою конфиденциальность на бесплатное использование этих сервисов».

Скрытые заплатки

Менее давний инцидент произошел в 2011 году, когда Red Hat изменила свою политику относительно заплат ядра. В прошлом все заплатки, которые она портировала в свои поддерживаемые релизы, предоставлялись в виде отдельных файлов, как делали все остальные.

Любой, кто использовал ту же версию ядра, мог взглянуть на все эти заплатки и увидеть, есть ли среди них пригодные и для его дистрибутива. Если они там были, их можно было применять выборочно, игнорируя другие, несовместимые. Таким образом можно было легко позволять всем пользоваться преимуществом открытого кода: делиться друг с другом улучшениями. В 2011 году Red Hat начала поставлять исходник ядра в виде одного большого tar-архива со всеми заплатками, уже примененными. И хотя все по-прежнему могли видеть код, стало намного сложнее определять, какие части были заплатками, созданными Red Hat, и какие проблемы они решали.

«Что бы они ни говорили, ясно, что это разозлило сообщество.»



► Fedora 7 (2007) включила версии live, USB и DVD, а также KVM в ядре.

Red Hat настаивает на том, что это неважно, поскольку ее версии ядра настолько отличаются от тех, которые используются в других дистрибутивах, по функциям, внедренным из более новых версий ядра в процессе разработки, что большинство ее заплат не соответствуют другим. Это, и тот факт, что она продолжает предоставлять всю свою работу над ядром и любые соответствующие заплатки непосредственно другим производителям — и есть политика Red Hat.

Но что бы они ни говорили, ясно, что это разозлило часть сообщества. Разработчик ядра Debian, Максимилиан Аттемс [Maximilian Attems], сказал, что «Red Hat следовало бы откатиться и не делать таких глупых с точки зрения управления шагов. Рядом с ними даже полуразработанная ветвь Oracle, «Unbreakable» 2.6.32, смотрится лучше».

Хотя мотивация, стоящая за всем этим, кажется вполне разумной — чтобы другие компании, занимающиеся Linux уровня предприятия, не предлагали клиентам Red Hat поддержку Red Hat Enterprise Linux за более низкую цену: благодаря запутанной системе заплат третьим сторонам становится намного сложнее предлагать тот же уровень поддержки, что и сама Red Hat; все же многих в сообществе разочаровывают подобные поступки.

Как и прежде, это вовсе не мировая катастрофа. Запатки Red Hat по-прежнему находятся в открытом доступе, как видно из поста <http://lwn.net/Articles/430600> в списке рассылки; это просто затруднило работу всех остальных и огорчило членов сообщества открытого кода.



Симбиоз

Оба этих примера, а есть и другие, дают некоторым повод предположить, что Red Hat ничем не отличается от других компаний. Пусть она и сотрудничала с сообществом, но когда в игру вступают интересы акционеров и растет заинтересованность менеджеров в добычании все больших сумм денег, она начинает мешать движению открытого кода, чтобы обезопасить свои преимущества. Справедливо ли рассматривать ситуацию в таком ключе?

Первое, что можно сказать в ответ, что это лишь отдельные инциденты за 20-летнюю историю. Если сравнить каждый из них с общим вкладом Red Hat в дело открытого кода, который иллюстрируют и ее работа над рабочим столом, и положение ведущего корпоративного помощника и участника разработки ядра, трудно счесть любой из них частью плана подрыва самого сообщества и конкуренции внутри открытого кода. Более того, наш разговор с Уайтхерстом объяснил, почему Red Hat никак и никогда не может целенаправленно подрывать движение открытого кода, даже если иногда и допускает неверные шаги в отношениях с широким сообществом.

«Многие думают так: «О, ведь есть же это общество, дай-ка я им воспользуюсь». Когда

OpenShift

Говоря о включении OpenShift (не OpenShift) в Fedora 17, мы разъяснили, что это инфраструктура, как проект Service. То есть она облегчает возможность быстро создавать новые виртуальные машины по мере роста вашего бизнеса и удалять их по мере его сокращения.

Проблема с этой моделью в том, что она создает множество дополнительной нагрузки, требующей управления. Каждая виртуальная машина управляется тем же способом, что и традиционный сервер, с программами, которые надо устанавливать, настраивать и поддерживать.

В некоторых обстоятельствах это неплохо, но если ваш бизнес состоит в разработке программ, вы не захотите заниматься системным администрированием.

Вы просто захотите разрабатывать свой код, синхронизировать его с репозиторием *Git* и затем смотреть, как ваши труды выходят в жизнь.

Именно это предоставляют компании в виде Platform as a Service [Платформа как сервис, PaaS], такие, как Heroku. Разработчики могут пользоваться всеми преимуществами облака и перебросить возню с задачами системного администрирования на людей, оказывающих им выбранную услугу.

Недавно Red Hat запустила собственную PaaS под названием OpenShift (не OpenShift — ну да, мы знаем, что перепутать легко!). Она работает на Red Hat Enterprise Linux и использует технологии, которые мы все знаем и любим, типа SELinux (ну ладно, пускай хоть и не любим), чтобы обес-

печить безопасное хранение данных и кода каждого приложения. Каждое работает в собственной «песочнице» без необходимости снабжать каждую собственную виртуальную машину.

OpenShift может работать с приложениями, разработанными на Node.js, Ruby, Python, PHP, Perl и Java, но это могут и другие PaaS. Интересным OpenShift делает то, что он с открытым кодом — и неудивительно, раз уж его предлагает Red Hat. А стало быть, вы можете работать с ним на любой хост-системе, какую пожелаете — будь то собственная хост-версия от Red Hat, или вы установите свою собственную на Amazon Web Services, или возьмете ремикс Fedora, чтобы запустить его на оборудовании в вашем собственном офисе.

мы об этом говорим, часто проводим аналогию с Томом Сойером, когда ему надо было покрасить забор, а он все повернул так, что это за него сделали другие. Можно убедить людей поступить так один раз, но если они не видят симбиоза во взаимоотношениях, когда вы тоже что-то делаете, они не попадутся на ту же удочку снова.

В большой степени это экосистема – я думаю, мы это поняли. Это основной источник нашего конкурентного преимущества, потому что другим компаниям так тяжело понять, что это значит, и вводить вместе с сообществом инновации – не в роли руководителя и не в роли эксплуататора, а именно буквально, *вместе с сообществом*».

Red Hat, конечно, использует сообщество открытого кода. Отдельные результаты труда всех этих разработчиков со всего мира – именно то, что дает компании ее основное преимущество. Это – бизнес для получения прибыли, и она зарабатывает деньги на результатах труда других; было бы ошибкой считать Red Hat таким филантропом.

Мудрая бизнес-модель

Но не меньшей ошибкой было бы считать ее не интереснее, чем другие компании. По словам Уайтхерста, Red Hat не может просто «употреблять» сообщество. Идти в ногу с быстрым темпом разработок, в частности, в области ядра, практически невозможно, если не работать с предыдущими. Грег Кроа-Хартман [Greg Kroah-Hartman], один из главных разработчиков ядра, отметил следующее об Android: «Не один год сообщество по разработке ядра предлагает этим компаниям поддерживать единство кода, чтобы быстрее выпускать исправления системы безопасности и легче переходить на новые API... но вот сейчас у них проблемы. Компании, чьи программы привязаны к различным платформам для Android или проприетарным



► Fedora 16 (2011) включает GNOME 3.2 и новую парадигму рабочего стола GNOME Shell.

драйверам, не могут использовать ранее наработанный код [upstream], что приводит к значительному увеличению времени разработки и усложнению поддержки».

Red Hat вынуждена поддерживать сбалансированные, симбиотические взаимоотношения, выгодные обоим партнерам; без этого стоимость включения в ядро новых функций и их поддержки значительно вырастает. Возможно, ей и удастся перенести некоторое увеличение стоимости, но, по словам Тиманна, Red Hat не может «взвалить на свои плечи бремя самостоятельного изменения мира» – на это уйдет куда больше, чем миллиард долларов. Это означает, что бизнес-модель, которую Янг и Юинг изначально создали для Red Hat, всегда будет гарантией того, что Red Hat останется хорошим партне-

ром для сообщества, даже если и будет время от времени совершать ошибки.

Еще более примечательно в этой модели то, что она приносит куда большую выгоду тем, кто находится за пределами Red Hat, чем любая традиционная модель, но в то же время является самодостаточной, в отличие от любого филантропического поступка.

Здесь, в LXF, мы считаем, что это стоит похвалить, и мы полагаем, что продолжающийся рост и успех Red Hat и в дальнейшем сможет приносить сообществу открытого кода только благо.



«Было бы ошибкой считать Red Hat филантропом.»

Уникальная культура

Вера Red Hat в модель открытого кода простирается далеко за пределы разработки ее программ. Ею пропитана вся корпоративная культура. Уайтхерст сказал нам: «Мы считаем, что создание во всей организации культуры, которая более открыта и свободна и является собой практически саморазвивающееся сообщество – наилучший способ управлять нашей компанией, в финансах или где бы то ни было еще».

Это приводит к ряду проблем, когда компания нанимает или интегрирует приобретенные внешние компании. «То, что мы считаем сплоченным и дружелюбным сообществом и шансом использования

наилучших идей, многим видится иначе: «это полный хаос в чистом виде, и я так работать не могу»».

Чтобы решить эту проблему, Red Hat создала партнерскую программу, согласно которой новых работников для приема в штат рекомендуют сотрудники компании (в этом году на работу принято более 1000 человек). Однако при интеграции приобретенных компаний нужен более активный подход.

«Когда мы купили JBoss – это было пять лет назад – это было очень, очень круто, потому что это было первое крупное приобретение, и это была совершенно иная культура. Мы извлекли из этого

урок, и сегодня мы очень усердно трудимся над тем, чтобы не просто прийти и сказать: «Добро пожаловать на борт», но и объяснить все «как» и «почему» нашей культуры. Так что мы очень много общаемся.

Возьмем, например, как мы купили Qumranet... когда мы завершили эту сделку, я встречался с нашими отделами продаж и отделами технологии и производства, и я привел целую армию, и встречался со всеми. Мы устраивали пикники и затевали множество всяких вещей вместе с командой, просто давая им понять, кто мы и почему мы делаем то, что мы делаем, именно так».

Fedora 17 «Beefy Miracle»

Проект Fedora Project был основан, когда Red Hat решила прекратить выпуск своего свободного дистрибутива Red Hat Linux, чтобы сфокусироваться на версии Enterprise. Сегодня это проект, ведомый сообществом, где инженеры из Red Hat сотрудничают с людьми из других компаний и проектов над разработкой и внедрением самого современного и превосходного ПО.

За последние девять лет он стал одним из самых популярных дистрибутивов и заработал немалое уважение за тесное сотрудничество с предлагающими разработчиками и интеграцию самых передовых технологий. Если вас интересует будущее компьютерных технологий уровня предприятия, Fedora – именно тот дистрибутив, который стоит попробовать. На диске этого месяца вы найдете версию Fedora 17, известную также как «Beefy Miracle», ее самый последний релиз. Многие новые функции – результат интеграции в дистрибутив самых последних разработок, и там есть от чего прийти в восторг.

Фишки рабочего стола

Поместите его в DVD-привод, и, когда он загрузится, вы увидите рабочий стол на основе Gnome Shell, дополненный фирменной маркой Fedora. Рабочий стол – одна из основных новых функций в этом релизе, теперь он основан на самой свежей версии, Gnome 3.4. Это крупное обновление: в нем

содержатся не только отладки и исправления ошибок, но и совершенно новые приложения. Самое примечательное – *Boxes*, простой инструмент для создания и управления новыми виртуальными машинами KVM и удаленными соединениями VNC с физическими машинами.

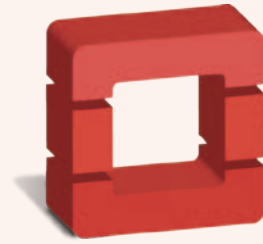
Еще одна вещь на рабочем столе, на которую стоит обратить внимание – это интеграция файлов в результатах поиска режима Overview, вызов видео в *Empathy* и появление *GIMP 2.8* – первого релиза, предоставляющего поддержку режима с единым окном для холста и панелей инструментов.

Открытые облака

Помимо рабочего стола, мы счастливы видеть последнюю версию *OpenStack* под названием Essex. Это – конкурент с открытым кодом Amazon Web Services (AWS), изначально разработанный NASA и Rackspace.

Его можно использовать, чтобы запустить собственное частное или общественное облако, позволяющее реплицировать Amazon Elastic Compute Cloud and Simple Storage Service. Его API совместимы с API от Amazon, так что замечательные инструменты AWS будут с ним работать.

Вы как-то не уверены, что вам нужны подобные программы – в конце концов, почему не использовать Amazon или традиционное резервное копирование и обслуживание файлов? Ну, во-первых, традиционное решение по резервному копированию даже близко не может сравниться по масштабу или обеспечить такую высокую скорость чтения и записи данных. *OpenStack* означает, что при необходимости вы сможете легко и быстро добавить больше данных для хранения.



openstack™
CLOUD SOFTWARE

Облако – большая часть будущего Red Hat, и вы сможете попробовать самую свежую итерацию OpenStack в Fedora 17.

Относительно Amazon, сайт *OpenStack* ссылается на Центр суперкомпьютеров Сан-Диего [San Diego Supercomputer Centre], который считает провайдеров проприетарных облаков более дорогостоящими для долгосрочной работы. Также у них вызывает беспокойство их поддержка решений по альтернативному хранению – что, если в будущем они решат перейти на другую платформу, или даже использовать одновременно несколько платформ для снижения риска? Поскольку API у *OpenStack* открытые, эти риски значительно меньше.

Динамические брандмауэры

Более того, если вам надоела необходимость циклов в модулях сетевых фильтров вашего ядра для быстрого (или даже временного) изменения правил брандмауэра, вы будете рады узнать, что теперь брандмауэром по умолчанию в Fedora является *firewalld*. Поскольку брандмауэр стал демоном, пользователи могут настраивать правила, активировать сервисы и закрывать порты, даже не прикасаясь к модулю ядра.

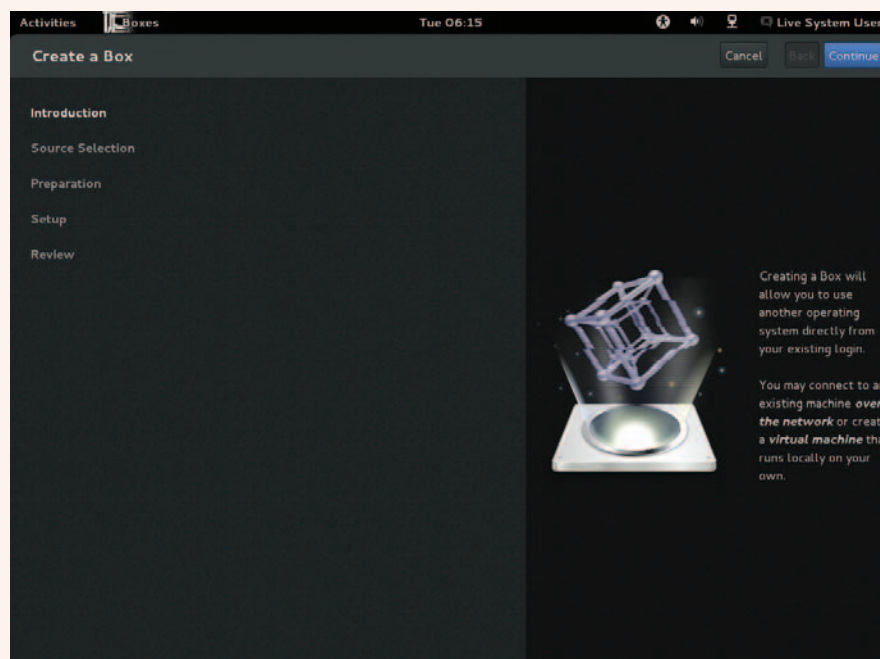
Будущее предприятие

Red Hat анонсировал выход следующего релиза Red Hat Enterprise Linux в 2013 году, так что можете рассчитывать увидеть некоторые из описанных функций в этом релизе.

Конечно, учитывая собранный на текущий момент объем данных и количество компаний, заинтересованных в безопасности облака и возможности содержать его на имеющемся у них оборудовании, стоит ожидать, что Red Hat воспользуется *OpenStack*. Это развивающийся стандарт, и Canonical начинает его поддержку вместо *Eucalyptus*, и уже более 100 других компаний принимают участие в проекте.

Все это, вкуче с усовершенствованиями ext4, позволяющих поддерживать файловую систему размером более 16 терабайт, делают Fedora и будущие релизы Red Hat Enterprise Linux крайне привлекательными для пользователей и поставщиков крупной инфраструктуры. **Linux**

«Мы счастливы видеть самую свежую версию OpenStack.»



Fedora 17 включает некоторые новые версии приложений Gnome, в том числе и Boxes – простой менеджер виртуальных машин.

Бьорн Балаж



Бьорн Балаж

OpenUsability.org пытается сделать вашу Linux-систему удобнее. Мы встретились с одним из его сооснователей.



Свободное ПО финансирует идеями: хорошие — приживаются, плохие — отмирают, так и должно быть. Однако

в погоне за крутыми новинками разработчики порой забывают о том, что работать с ними придется простым смертным. Обладая ресурсами Sapocal, можно прибегнуть к масштабному тестированию эргономичности, что и сделала компания с целью улучшить рабочий стол Unity. Но такие роскоши доступны далеко не всем программам в вашем репозитории.

Поэтому есть потребность в своеобразной системе сбора пользовательских отзывов, с передачей их разработчикам. И хорошо бы она была универсальной, дабы не расплывать время и силы на сотни разных версий. Воплощением этой блестящей идеи стал проект **OpenUsability.org**. Мы побеседовали с Бьорном Балажем [Bjorn Balazs], одним из его основателей, о когнитивных диссонансах, Джимми Уэйлсе и о том, почему ему больше не нужно покупать футболки KDE.

LXF: Расскажите немного о **OpenUsability.org**...

Бьорн Балаж: Мы основали этот проект в начале 2000-х. Точную дату появления перво-

го сайта я не назову, но эта инициатива шла от двух берлинцев — профессионалов в сфере эргономики и пользователей свободного ПО: нас обоих не устраивал неуклюжий интерфейс программ, с которыми мы работали, и мы независимо друг от друга занялись этой проблемой.

Мы считали свободное ПО достойным усилий, и связывались с лидерами проектов, предлагая помощь. Однако результат был всегда один. Нам отвечали: «Ну да, это не здорово, но мы не знаем, как будет правильно, мы ж технари», и чувствовалось, что они просто хотят послать нас подальше.

Они делали свое дело, а нам оставалось только досадовать, что мы не можем помочь проекту, в котором заинтересованы... не придется ведь к разработчикам просто так, заявляя: «Привет, я-то вам и нужен» — это уже наглость.

Тогда мы встретились и решили: мы это все перевернем. Пора сказать: «У нас к вам есть предложение. Мы хотим сделать свободное

ПО удобнее в использовании, и ваш проект в том числе».

И так мы сдвинулись с мертвой точки. Мы заявили о себе, разработчики могли в любой момент обратиться к нам за советом, и это оказалось востребовано. Для меня все началось именно так.

У второго основателя проекта другая история: он пользовался Windows, его девушка — Linux, а его фирма занималась эргономикой; они поспорили, что же из двух все-таки лучше. И однажды он сказал: «Я, как владелец соответствующей фирмы, в состоянии выдать оценку».

Итак, где-то в 2001–2002 году, они действительно провели серию тестов практичности для Linux и KDE против Windows. Они брали неопытных пользователей, без специальных знаний о компьютерах, и давали им одинаковые задания. Результат оказался поразительным: победителя не нашлось. В чем-то лучше была Windows, в чем-то — Linux или KDE. Так что в конечном счете, можно сказать, победила дружба.

Очевидно, что обычный пользователь может и не уметь смонтировать CD-ROM, ведь в те времена их еще нужно было монтировать, а если вам неизвестно, что это такое, то как вы это сде-

О том, как все началось

«Мы считали свободное ПО достойным усилий, и связывались с лидерами проектов.»

лаете? Вот такими разными дорогами мы двое пришли к тому, чтобы искать какое-то новое решение.

Мы дали ему юридический статус. Теперь мы «добровольное сообщество», как KDE eV – это такая немецкая форма организации, позволяющая иметь некоторые фонды и т. д.

LXF: То есть вы стали этим заниматься, потому что вы программист и хотите, чтобы ваши программы было легче использовать?

ББ: Я – психолог. Будь я программистом, я вряд ли бы в это ввязался: программистам как раз проще попасть в любой проект, там вся инфраструктура под них подстроена. Они умеют пользоваться баг-трекерами, а я, хоть и занимаюсь множеством проектов, баг-трекеров до сих пор избегаю: мне кажется, что это не для людей. Как бы объяснить... конечно, программисты – тоже люди, но чтобы понимать, как это работает, нужно иметь особый склад ума.

Лично я и не вникаю, какая версия программы у меня стоит – я просто ею пользуюсь, и мне повезло в том, что в случае проблемы я просто могу прийти к разработчику и сказать: «Вот тут у меня ошибка, исправьте ее».

LXF: А в какой программе?

ББ: Я участвую в проекте *LibreOffice*, и меня весьма волнуют открытые форматы. Работая с этими ребятами, узнаешь много нового и начинаешь разбираться в проблеме форматов.

LXF: Это не в Берлине власти официально перешли на открытые форматы? Вот в Мюнхене точно связались с Linux.

ББ: В Мюнхене свой Linux-дистрибутив, по имени Mupix или что-то вроде, и он добился довольно большого успеха. В смысле, было много и проваильных проектов, но Мюнхен – история успеха.

LXF: Вы когда-то говорили о когнитивном диссонансе – в связи с чем, не напомните?

ББ: В контексте вопроса «Что мною движет сейчас». Мы как бы идем по тропе эволюции. Мы выступили с инициативой, проделали массу экспертной работы по проектам, а помощь требовалась многим; привлекли пару специалистов по эргономике. Со временем обнаружилось, что най-

ОБ УДОБСТВЕ ИСПОЛЬЗОВАНИЯ:

«Ведь оценивать удобство должны сами пользователи, а вовсе не эксперты.»

ти таковых непросто, так же как привлечь самих пользователей для продолжения тестирования. Ведь оценивать удобство должны сами пользователи, а вовсе не эксперты, которые «знают, что всем надо». А выходит, что экспертов найти гораздо проще. И все острее вставал вопрос: «Как привлечь пользователей? Как включить их в процесс разработки?» Код им писать не надо. Уже хорошо, если кто-нибудь вовремя скажет: «Эй, вы что, все еще на *OpenOffice*? Вообще-то давно есть *LibreOffice*». Вот так вы станете частью сообщества, ведь это значит, что вас волнует судьба проекта. И мы вновь и вновь упирались в вовлечение пользователей. Понятно, что все читали вопрос «Будете ли вы регистрироваться?» во время установки системы. Есть и другие давние способы пригласить пользователей к участию, но ведь они не работают. В основном, нет.

LXF: Ну, это объяснимо: какой смысл регистрироваться, если ты еще не опробовал программу? Не лучше ли сначала ее испытать?

ББ: Вот именно. Пусть я хочу пользоваться программой. Если она хорошая, нравится мне и помогает в моей работе, может, я и зарегистрируюсь... Поэтому когда мы работаем над программой, выявляем недостатки, мы пробуем всякое – скажем, в одном из наших проектов мы на стадии регистрации предлагаем пользователям поучаствовать в периодическом опросе, и где-то раз в два месяца выдаем ссылку, пройдя по которой, они могут нам помочь.

Но даже так обратной связи было очень и очень мало. Примерно от 1% пользователей, а то и меньше, конечно, не база для выводов. Выборка пользователей не репрезентативна. И тут сработало мое образование психолога: мы решили прибегнуть к этому самому когнитивному диссонансу. Это такая ситуация, когда ваши представления расходятся с тем, что вы делаете, и у вас возникает дискомфорт.

Обычно, если ситуацию нельзя изменить, человек меняет свое отношение к ней, свое мнение. Проводилось множество замечательных психологических экспериментов, доказывающих действенность этой теории: например, вы ставляете людей написать эссе на тему «Я люблю ядерное оружие», и после этого их мнение действительно меняется. А если им хорошо заплатить, то они смогут сказать себе: «Я просто сделал это за деньги, и вовсе не обязан так думать».

Вот и мы теперь пытаемся заманить народ что-то сделать по мелочи, как бы говоря «это же всего минутка, просто возьми и сделай» и не заставляя их решать, хотят ли они помогать сообществу. Даже в социуме, стоит ли ожидать от другого человека чего-то сверх общепринятого или диктуемого обществом? Поэтому мы и просим самую малость, а потом уже человек, дабы избежать когнитивного диссонанса, начинает думать: «А мне нравится этот проект. Я уже в нем участвую». И уже гораздо проще просить его о чем-то большем.

Хорошим примером может быть проект KDE «Вступай в игру». По факту, это способ регулярно получать деньги – порядка €100 в год, причем платить надо поквартально, то есть деньги регулярно снимаются с вашего банковского счета. И – это сюрприз? – у них не так-то много подписчиков. Причина ясна: представьте, что людям нужно сделать 14 дел, а это будет 15-м, а им нужно сделать сперва свое, и в результате мы остаемся с теми, кто и так был в сообществе. Я сам вступил просто потому, что работаю с KDE и рад, что мне больше незачем покупать их футболки в поддержку проекта – у меня их и без того уже куча.

LXF: Это в стиле Джимми Уэйлса, да?

ББ: Да, точно [Jimmy Wales – основатель Википедии, – прим. ред.]. Как только у них какая-нибудь новая кампания, он уже на всех web-страницах, и это начинает раздражать, а что может быть хуже, чем раздражать пользователей, от которых





ты хочешь получить деньги? А мы стараемся найти ситуацию, когда уместно попросить их о помощи, причем так, чтобы им было интересно, пробудить их любопытство... вот тогда это сработает. И мы все еще экспериментируем. Это лишь начало пути, и у нас куча идей, которые еще надо проверить.

LXF: Вы упомянули, что пробовали вести такую работу в *LibreOffice*?

ББ: Я поработал со многими проектами, и надеюсь, что удастся убедить их запустить пилотный проект – видимо, в *LibreOffice* и в менеджере персональных данных [PIM] KDE, *Kontakt*. Повторюсь, все это пока в теории, решений в самом ПО еще нет, но мы ведем переговоры с обоими проектами и с них хотим начать.

Нам безусловно нужен выход на дистрибутивы, поскольку они не должны убирать то, что встроили мы, и надо работать глобально, на уровне всей системы; мы, например, сотрудничаем с OpenSUSE, и на данном этапе их позиция такова: «да, мы не будем это убирать, нам нравятся ваши идеи и мы вас поддержим» (то есть они идут на это не в пик Ubuntu, а в своих интересах, ради своей же системы).

Опять-таки, мы имеем дело со свободным ПО, то есть коллективным знанием, и это хорошо, что все больше людей вовлекается в этот процесс, ведь мы можем учиться друг у друга. Цель не в том, чтобы лично моя идея или программа всех победила, а в том, чтобы в итоге победило свободное ПО. Какое именно, мне безразлично.

LXF: Постановка такой цели не может не радовать, и я знаю, что вы много общаетесь с пользователями, да?

ББ: Тут можно упомянуть нашу деятельность в OpenUsability, которая направлена на создание программной инфраструктуры, доступной как для свободного, так и для проприетарного ПО – ну, понятно, что свободное ПО может использовать кто угодно – свободным проектам она пригодится, поскольку для привлечения новых членов сообществу требуются две вещи: техническая инфраструктура и люди, точнее, своего рода управляющий сообществом, знающий, что людям нужно. Потому мы этим и занимаемся, и нам это не легко дается, ведь мы в основном не технари. И мы ищем людей в помощь с собственно технической ча-

стью – пользовательским интерфейсом и прочим; это очень, очень приветствуется. Если есть желающие продвинуть наш проект в этом направлении, мы будем более чем счастливы с ними связаться.

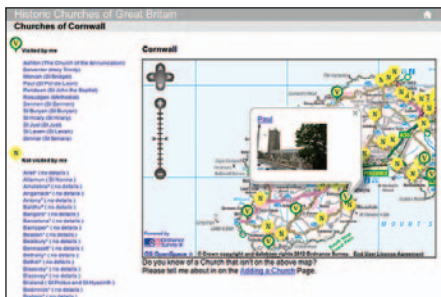
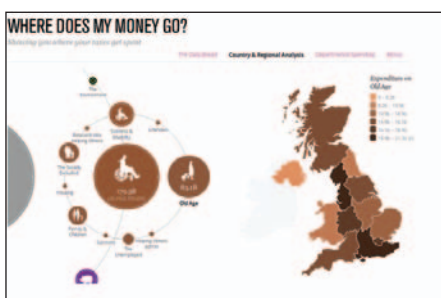
LXF: А вы придерживаетесь конкретных языков программирования или платформ?

ББ: На данный момент мы работаем на Wicket, это сетевая архитектура Java. Менять пока не собираемся, но, опять же, я не технарь и не знаю, как это взаимодействует со всем остальным, хотя полагаю, что такое возможно. Я хочу сказать, ведь в этом же вся прелесть свободного ПО, верно? Каждый может встать на плечи гигантов. Но как это работает, я не очень понимаю, и проблема в том, что ПО, о котором я говорю, изначально было закрытым, так что пока там нет никаких связей, однако в ближайшие один-два месяца это уже будет открытый код.

Главным образом, для меня это эксперимент: мне еще не доводилось руководить свободным проектом, я всегда был в роли консультанта. Но мне очень любопытно узнать, что получится. **LXF**

Открытые данные могут спасти мир

Первая в Англии конференция Городов открытых данных доказала огромную политическую поддержку доступности общественных наборов данных. Пользуйтесь ими, чтобы их не утратить, говорит **Адам Оксфорд**.



Человечество переживает кризис экзистенциальности. Наши природные ресурсы и политические структуры сталкиваются с проблемами почти непреодолимой сложности, которые грозят просто раздавить их. Стоимость жизни растет, а наше население быстро растет и быстро старится.

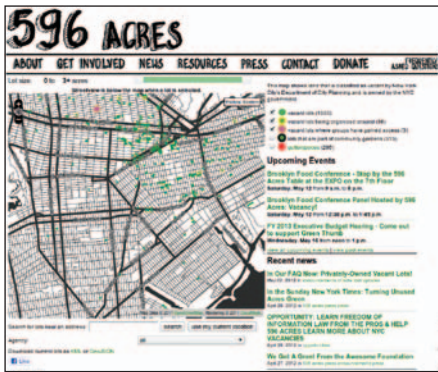
К 2071 году ожидается, что количество британцев, доживших до ста лет, увеличится в сто раз. Продолжительность жизни в развивающихся странах растет на пять часов ежедневно. Плюс к тому, технология развивается столь стремительно, что нам, похоже, придется учить наших детей профессиям, которых пока даже не существует, иначе

мы рискуем быть вытесненными формирующейся рыночной экономикой, которая ничего не потеряет, избавившись от устаревшего образа мысли.

Если, по-вашему, гибель западной цивилизации маловероятна, побеседуйте-ка о неизбежности цивилизации с представителем народа майя. Столь драматичный прогноз было бы легко оставить без внимания, будь он дан изрыгающим пену безумцем или вычитан в сценарии очередной се-

«Придется учить детей профессиям, которых пока не существует.»

рии Горца. Но когда эта перспектива хладнокровно озвучена директором Local Government Information Unit [информационной службы органов местного самоуправления] Джонатона Карр-Уэста, есть о чем задуматься, особенно если предупреждение звучит не на базарном углу, а в солидной и рациональной атмосфере первой в Великобритании конференции Городов открытых данных [Open-data Cities], недавно прошедшей в Брайтоне [Brighton]. Карр-Уэст вовсе не фанат конспирологии, а бывший заместитель директора программ в RSA, и с его послужным списком нельзя не считаться. Он имеет степень доктора философии [PhD, высшая ученая степень, — прим. пер.] по культур-



► 596 Acres перечисляет общественные территории вокруг Бруклина, свободные для аренды.

ной идентификации в Западной Африке и работы по водным и санитарным проблемам и экономике миграций.

«Общества, отказывающиеся от инноваций и от адаптации к сложным проблемам, стоящим перед ними, могут просто исчезнуть, — подчеркивает Карр-Уэст. — Ничто не гарантирует дальнейшее существование мира, в котором мы живем — если мы намерены продолжать поддерживать определенный уровень жизни, нам необходимо найти инновационные способы решения этих проблем».

Объединенное местничество

Решением является своего рода открытость и сотрудничество, которое уже долгое время практикуется в сообществе FOSS, но на муниципальном уровне. Карр-Уэст выдвигает идею «объединенного местничества [connected localism]» как способа использования мощи Интернета для решения глобальных проблем на местном уровне, усилив ее принципом «открытых данных [open data]».

Суть открытых данных в том, чтобы сделать огромный объем знаний, накопленных местными и национальными правительствами — за счет налогоплательщиков — доступными общественности. Это не сложнее, чем загрузка огромных таблиц CSV в сеть, как сделал Уайтхолл [Whitehall] с релизом своей базы данных по правительственным расходам COINS вскоре после прихода к власти коалиции, и не сложнее, чем сводные карты мест преступлений, ежемесячно публикуемые на police.org.uk. Важно, что «открытые» здесь также означает «читаемые компьютером и легкодоступные».

Начиная с утраты веры в демократию и заикающаяся энергоэффективностью, проблемы, стоящие

перед человечеством — и особенно перед городами — слишком серьезны, чтобы их решали политики. Пора дать шанс людям попытаться сделать это, выдав им нужную информацию и необходимые API, чтобы они могли заняться инновациями.

«Исторически город функционировал как место, где люди собирались и делились своими идеями — не только товарами и услугами, но идеями и новшествами, инновациями: на форуме, в кофейне, в пабе или на рынке, — говорит Карр-Уэст. — И подобно тому, как город являлся местом для инноваций и общения, открытые данные начинают создавать такое же пространство».

Города — главная цель движения открытых данных, потому что большинство из нас более способно создавать изменения в своем непосредственном окружении, нежели на национальном или глобальном уровне. И сейчас самое время стать ориентированным на сообщество кодером, жаждущим что-то изменить. Благодаря прозрачности и свободе информации у вас есть доступ к большому, чем когда бы то ни было, объему данных о том месте, где вы живете, в реальном времени. Хотите создать приложение или сайт про расписание местных автобусов, голосование или вывоз мусорных ящиков? Есть вероятность, что эта информация уже имеется на домашней странице органов местного самоуправления, или станет доступнее, если вы попросите ее предоставить.

Города важны еще и потому, что это — место, где мы живем. Согласно данным Всемирного банка, к середине нашего века число горожан практически сравняется с количеством ныне живущих людей. Рост населения в целом может сильно замедлиться, но урбанизация — никогда.

Как заявила Лин Дуди, партнер фирмы Arup, занимающейся инжинирингом и дизайном, и еще один докладчик на конференции, этого никто не ожидал. «Десять лет назад ИТ и телекоммуникации должны были прикончить города, — сказала Дуди. — Повсеместное распространение широкополосной связи должно было отменить местоположение как таковое; предполагалось, что мы все будем сидеть в своих телекоттеджах, таращась в мониторы и набирая текст».

Дуди много лет исследовала «умные города», определяемые ею как применение ИТ-стратегии к городам, относительно трех аспектов: экономического развития, превращения городов в привлекательное место для жизни и экологически безопасного управления ростом.

По словам Дуди, одно из главных препятствий на инфраструктурном уровне то, что собранная информация — поступает ли она с сетевых сенсоров или от визуального наблюдения — как правило, разрознена и хранится разобченно. Открытые данные в той же мере касаются необходимости обеспечить базам данных возможность общения между собой, в какой они касаются необходимости свободного использования их гражданами. В качестве примера она привела Городской совет Бирмингема [Birmingham City Council], где Arup помогает объединить данные транспортного и цифрового отделов и отдела устойчивого развития.

«Мы думаем о том, как их лучше всего объединить и как расширить некоторые из этих проектов, чтобы помочь им начать мыслить более интегрированно».

Данные для людей

Том Стайнберг кое-что знает об открытых данных. Будучи основателем MySociety.org, он помог убедить правительство Великобритании открыть информацию и дать тем, кто не является программистами, возможность ее использовать. Он начал свою презентацию с шуточного признания: с целью настройки riteToThem, онлайн-формы создания писем депутатам Парламента, он украл данные адресной книги и справочника почтовых кодов Палаты Общин, потому что в 2003 году эта простейшая информация не была доступна в электронной форме на общественном домене.

«Открытые данные — это как свобода слова, — говорит Стайнберг. — Преимущества оценить трудно, но стоит оказаться в стране, где этого нет, как разница станет очевидна».

Как и Дуди, Стайнберг подчеркивает важность открытых данных как способа повышения эффективности работы правительственных учреждений. Годы споров с политиками научили его, что луч-



► Том Стайнберг [Tom Steinberg]

Основное приложение для открытых данных

Закрывая конференцию Open-data Cities, Тим Ходсон [Tim Hodson] от фирмы Talis, занимающейся связанными данными, продемонстрировал прототип приложения, которое может стать основным приложением для открытых данных. Digital City соединяет как можно больше общественно доступных наборов данных на платформе от его компании Kasabi и немного напоминает смесь Google Maps. Все, от данных земельного реестра до списка стан-

ций ТО и музейных экспозиций, можно узнать по запросу через простой поисковый инструмент, базирующийся на справочнике почтовых индексов.

Ходсон объясняет, что идея заключается в предоставлении пользователям возможности вносить дополнения в базу данных и разрешении компаниям извлекать из нее доходы. Его коллега, Ли Доддс [Leigh Dodds], говорит о данных как о ресурсе с определенным доступом, и считает, что их ценность

кроется в увязывании и предоставлении людям возможности для эксперимента. «Интегрируя данные с Сетью, мы обретем новое употребление общественных инструментов, созданных для контента и приложений, — объясняет он. — Данные приобретают интерес, когда они контекстуальны. Мы должны создать цифровые граффити».

<http://talis-systems.com/2012/04/open-data-cities-demonstrator/>

ший способ убедить их открыть информацию – демонстрация, насколько это облегчит им жизнь. Онлайн-версия парламентского справочника Hansard, www.theyworkforyou.com, с возможностью поиска, была моментально принята членами Парламента, ставшими самыми активными пользователями этого сервиса.

Еще одно преимущество открытых данных – они позволяют не-членам правительства создавать инструменты и сервисы для применения рядовыми гражданами при поиске ответов на свои вопросы, что снижает административную нагрузку по отработке запросов, касающихся Свободы Информации, и ответов на ежедневные звонки. «Опубликуйте это, и ваш телефон перестанет звонить» – вот мощный аргумент, если нужно раздобыть данные, до сих пор не попавшие в доступ.

Однако активисты открытых данных отнюдь не занимаются анти-истеблишментом. В адаптацию открытых API существующих баз данных для открытого использования и изыскания возможности их публикации в сети вовлекается немало следств. и многие официальные лица испы-

«Правительственные закупки должны контролироваться обществом.»

тывают известное нежелание делиться знаниями, хранителями которых они традиционно являлись. Таких убеждают в ценности открытия данных и доверия «коллективной мудрости».

(Не)Свобода (данных)

«Открытые данные не бесплатны, – говорит Стайнберг. – ПО следует приобретать на конкурсной основе. Правительственные закупки должны контролироваться обществом.»

Это перекликается с высказываниями доктора Лоры Джеймс из Open Knowledge Foundation [OKF, Фонд открытых знаний] – организации, которая выступает в том числе и за применение программ с открытым кодом в правительственных системах.

«Открытые данные пока стоят дорого, поскольку мы их еще осваиваем, – говорит Джеймс. – Для правительства они по-прежнему остаются нововведением, а представители бизнеса едва начали над ними задумываться... с открытыми данными мы сейчас на той же стадии, на какой были с Сетью, когда вручную писали на HTML.»

По словам Джеймс, концептуальный скачок для правительственных агентств в том, что им следует осознать: самое лучшее применение для данных, которые они складывают, в том, что над ними начнут размышлять другие люди, способные прояснить их и додуматься до чего-то невообразимого ранее. Подобно Стайнбергу, она обнаружила, что среди работников правительственных учреждений весьма популярны демократические проекты, подобные сайтам OKF Open Spending (openspending.org): манипуляция сложными данными здесь проще, чем с помощью их собственных внутренних инструментов.

► Несмотря на изобилие наборов данных, 75% посетителей уходят с первой же страницы data.gov.uk.

Одна из проблем, знакомых большинству делегатов, заключается в привлечении людей к опубликованным данным и к их использованию. Согласно недавнему отчету Национального бюро аудита, в декабре 2011 года на data.gov.uk было 7865 наборов данных, но 75% посетителей ушли с первой же страницы, а 84% экономической выгоды пришлось на данные Метеорологической службы. Посещаемость тоже была низкой: за первые два года было 1,75 млн посещений, по сравнению с 47 млн посещений полицейского сайта мест преступлений с февраля по декабрь 2011. С одной стороны, пока что рано выносить суждения об успехе открытых данных. С другой – американский эквивалент data.gov был в прошлом году закрыт, в немалой степени из-за отсутствия интереса.

Тем не менее, большинство выступающих на конференции были убеждены в том, что публикация данных – правильный путь, даже если придерживаться PDF или иных нечитаемых форматов.

«Начните с малого, – советует Эме Коулмен, заместитель директора по сотрудничеству в области цифровых технологий Правительственной Службы Цифровых Технологий Великобритании [UK

Government Digital Service]. – Не позволяйте стремлению к лучшему стать врагом хорошего».

Хитрость в том, чтобы, опубликовав имеющиеся данные, продолжать прислушиваться к тому, что люди хотят с ними делать, и сотрудничать с разработчиками, чтобы отладить наборы данных и улучшить их внятность.

Равные условия

На своей предыдущей должности директора по цифровым проектам в Администрации Большого Лондона, Коулмен взяла на себя инициативу по созданию London Data Store [Склад данных о Лондоне] (data.london.gov.uk). Публикуемая там информация была очень выгодна как академическим учреждениям, так и предприятиям общественного и частного сектора. Она использовалась для создания транспортных карт города в реальном времени, «тепловых» карт самых популярных маршрутов в центрах проката велосипедов и простой в использовании экспликации распределения административных расходов специалистами Openlylocal.

«[Открытые данные – это] город, который общается сам с собой, – говорит Коулмен. – Для London Data Store данные никогда не были главным. Главное – совершенно иной подход к управлению... речь идет о том, чтобы уравнивать шансы и вести

Бибишники желают открыться

Не только городские советы хотят сделать данные свободными. Билл Томпсон – глава отдела партнерского развития и архивов в BBC. Его последним заданием было разобраться с обширным национальным архивом киноматериалов BBC, чтобы оцифровать их и обеспечить поиск по ним.

Основой проекта является Redux – Томпсон описывает его как «iPlayer на стероидах». Благодаря Redux, создатели программ получают доступ к архивам, задействуя различные метаданные для поиска материалов и пленок. Главной частью его разработки было сканирование всех материалов программы *Radio Times* с 1923 года, так как это была лучшая запись, сделанная BBC – список, ранее

не существовавший в цифровом формате. Когда с помощью Redux из архива берется пленка, создателям программ предлагается вернуть ее, дополнив метаданными и тэгами. Томпсон надеется, что постепенно создастся ресурс, которым смогут пользоваться создатели местных новостных программ, библиотеки, школы, историки. По завершении цифровизации, его потенциал как части национальных архивов будет просто безграничен.

Томпсон уподобляет текущее состояние открытых данных строительству Венеции.

«Появятся дворцы изумительной красоты, – говорит он, – сияющие вокруг. Но пока мы только вбиваем сваи в болото.»

► Билл Томпсон
[Bill Thompson]



взвешенный диалог с гражданами, когда обе стороны располагают одинаковой информацией».

Грег Хэдфилд [Greg Hadfield], который организовал конференцию Open-data Cities, страстно верит, что данные помогут людям построить новый мир, в корне отличающийся от старого, в котором управление происходит сверху вниз. Будучи основателем Schoolsnet, одного из первых сайтов открытых данных, где публиковались таблицы с результатами экзаменов, он знает, насколько это мощная сила.

«У всего этого есть демократический мотив, и я сказал “демократический” вместо “правительственный” сознательно», говорит Хэдфилд.

С одной стороны, демократия означает, что открытые данные можно использовать для увеличения прозрачности и отчетности политиков (как в том случае, когда разразился скандал, связанный с расходами членов Парламента) и чтобы обязать администрацию публиковать информацию о любых расходах, превышающих 500 ф. ст. Но также она связывает людей с тем местом, где они живут. До 1990-х годов Хэдфилд был журналистом с Флит-стрит по расследованиям, так что он остро чувствует всю силу данных для подкрепления повествования.

«Ведется серьезная работа по подготовке к 2014 году, столетию Первой мировой войны, ко-

торая – надеюсь – объединит данные по именам с позабытых военных мемориалов и судьбы реальных людей в реальных местах в истории».

Для Хэдфилда города с открытыми данными означают «города, которые мыслят, как сеть», с перекрестными ссылками, адаптируемые, стойкие. «Для меня вопрос заключается не в визуализации, а в более крупномасштабной картине, в которой мы существуем на планете уже 160 000 лет, и живем в городах уже 6 000 лет, – говорит он. – А Сеть существует только последние 50 лет... Я верю не в технологии и бизнес, а в своих сограждан».

На страже приватности

Конечно, есть один вопрос, касающийся любого проекта по открытым данным или умным городам, и это – вопрос конфиденциальности. Анонимность данных – это очень хорошо, но стоит начать собирать и публиковать их, как появятся непредвиденные последствия: от внезапного обнаружения возможности взлома «умных домов», когда их хозяев нет дома, до непреднамеренного привлечения внимания к жертвам преступлений.

Нравится нам это или нет, но мы живем в эпоху больших данных. Мы можем, подобно ткачам-людям [противники механизации в Англии в конце XVIII – начале XIX в., – прим. пер.], отправиться с гигантским метафорическим молотом на сер-

веры Facebook и настроить Anonymous в центрах управления умным городом в IBM, чтобы затормозить возможности технологии, позволяющие узнать что-то о нас. Или мы можем работать совместно, чтобы управлять нашими данными и быть уверенными в том, что будущее наших городов связано с развитием местного самоуправления. Мы имеем уникальную возможность создавать цифровые инфраструктуры, которые будут совершенствовать демократию, поддерживать слабых и привлекать к ответственности власть предрержащих, сохраняя в то же время возможность людям оставаться хозяевами собственной конфиденциальности.

«Facebook, Groupm и прочие навязываемые нам сервисы не являются единственной возможностью, – говорит Хэдфилд. – Мы должны взять управление в свои руки, чтобы создать такие города, которые нам нужны».

Но для этого мы должны воспользоваться имеющейся у нас сейчас возможностью. Общественный сектор проявляет добрую волю, и у нас есть поддержка от высших эшелонов власти, чтобы открыть хранилища с данными. Не воспользовавшись выданными нам инструментами как разработчики и энтузиасты открытого кода, мы их утратим. Если все это вам кажется крайне преувеличенным, подумайте вот о чем: в своей автобиографии Тони Блэр [Tony Blair] сказал, что единственное, о чем он жалел все время пребывания на посту премьер-министра – это Закон о свободе информации [Freedom of Information Act], который, по его мнению, связывал членов правительства, имевших более важные дела, и – в основном – был немногим больше, чем этакий мазохистский подарок от политиков прессе «большой дубинки», чтобы ее впечатлить. Сегодняшняя мода на открытые данные отнюдь не означает, что она продлится вечно, и даже ее величайшие достижения можно развернуть вспять.

На Даунинг-стрит уже забормотали, что Дэвид Кэмерон [David Cameron] начал склоняться к точке зрения Блэра. «Мы пока не достигли стадии зрелости в этом вопросе, – говорит Эме Коулмен. – Вы обольщаетесь, что строите структуры, разрушить которые будет непросто, но кто на самом деле знает, как все обернется?» **LXF**

Пять примеров работы открытых данных

1 Куда деваются деньги?

www.wheredoesmymoneygo.org

Визуализация данных, может, и устарела немного, но OKF демонстрирует ее мощь с помощью Where Does My Money Go? Национальные и местные бюджеты для лучшего усвоения разделены на мелкие части.

2 596 Acres

www.596acres.org

Много ли мы знаем о своем городе? Где располагаются границы избирательных участков, мусорные баки и зеленые зоны? 596 Acres перечисляет в базе данных пустые и свободные участки, находящиеся в общественном владении, вокруг Бруклина, Нью-

Йорк, чтобы жители могли брать их в аренду или использовать в качестве мест для встреч.

3 Исторические церкви Великобритании

www.peterandsusan.co.uk/church

Картографическое управление [Ordnance Survey] – одна из первых организаций Британии, открывших свои данные – тесно сотрудничает с потенциальными разработчиками по вопросам их применения. Исторические церкви Великобритании – отличный пример такого применения: Питер и Сьюзан наделены миссией посетить и сфотографировать каждую из 22 000 церквей в стране, руководствуясь данными Управления.

4 (Не)Сфотографированная Испания

<http://senseable.mit.edu/worldseyes/visuals.html>

Один из нескольких проектов MIT Senseable City Lab, который использует данные Flickr, чтобы выяснить, какие туристы посещают Испанию, как они перемещаются и что их интересует.

5 Информационный Сервис Брайтона и Хоува

www.bhils.org

Хотя этот ресурс и не слишком хорошо разрекламирован, но он феноменален. BHLS [Brighton and Hove Information Service] – сложное хранилище данных со встроенными инструментами визуализации, постоянно дополняемое новой информацией.

Турбо-заряд

Firefox

Самый популярный в мире браузер с открытым кодом становится все мощнее. Майк Сондерс отслеживает его прогресс и показывает, как усилить его мощь, добавив функций.

Firefox, бесспорно, самый важный проект свободного ПО в истории. Конечно, Linux хорош, и на нем работает немалая часть Интернета, но по-прежнему не слишком известен вне кругов компьютерных умников. А вот Firefox используется повсеместно, его марка отлично распознаваема, в его поддержку проводятся превосходные маркетинговые кампании, и он вернул надежду Всемирной паутине, освободив нас от тирании Internet Explorer. Был момент в 2002 году, когда на долю браузера от Microsoft приходилось целых 95% рынка браузеров, и ситуация казалась устрашающей и безнадежной.

Mozilla держалась на остатках поддержки от старых пользователей Netscape, но остро был нужен новый, быстрый, современный, привлекательный многоплатформенный браузер, продвинутый на рынок, чтобы отвоевать Сеть. Firefox (ранее известный как Phoenix и Firebird) все это предоставил – а главное, ему удалось оттянуть пользователей из мира Windows. Он не явился на рынок браузеров завоевателем, но это даже хорошо – чем больше, тем веселее.

Но последние годы были для Firefox довольно трудными. Internet Explorer из полного хлама пре-

вратился в нечто вполне приемлемое, а Google Chrome дал уже существующим браузерам весьма чувствительного пинка в тыльную часть, благодаря своей нереально быстрой производительности и отличному интерфейсу.

Чтобы отвоевать упущенное, команда Firefox решила похоронить старый цикл релизов, когда основные релизы выходили раз в год-два, а между ними шли мелкие обновления функций, выбрав куда более быстрый цикл выпуска новых основных релизов ежемесячно. Это было весьма спорно, и пользователи поделились на два лагеря: те, кто за, и те, кто против.

За: Новые функции доходят до пользователя намного быстрее, поскольку релизы выходят в соответствии с более строгим графиком, а не по прихоти команды разработки. Это позволяет Firefox конкурировать с Chrome, который постоянно обновляется где-то в тени и удивляет пользователей все новыми и новыми прелестями.

«Был момент в 2002 году, когда на долю Microsoft приходилось 95% рынка.»

Против: Добавки слишком часто нарушают совместимость, и это – логистический кошмар для отделов техподдержки крупных компаний. Они месяцами тестируют новые релизы на предмет взаимодействия с внутренним ПО, а к моменту их готовности уже успело пройти пять новых релизов с изменениями в UI, и старый больше не поддерживается. Последняя проблема отчасти смягчена в Firefox 10, который являет собой ESR, или Extended Support Release – Релиз с продленной поддержкой; иными словами, он будет получать исправления ошибок и отладки системы безопасности почти в течение года, так что компании могут использовать его некоторое время, сохраняя уверенность в том, что их сайты и расширения не рухнут абсолютно внезапно.

В любом случае, учитывая темп развития в сообществе Firefox, кажется, что уже целая вечность прошла с тех пор, как мы подробно о нем писали – о Firefox 4 в номере 145, если быть точным. Теперь он уже на стадии версии 12, и за это время много чего произошло. Мы рассмотрим, что же изменилось, и покажем, как установить последний релиз и приправить его лучшими дополнениями.

Что нового?

После настоящего шквала новых релизов за последние 12 месяцев, столько всего изменилось со времен релиза 4.0! Прогуляемся же по версии 12.0.

В погоне за *Google Chrome*, и в меньшей степени – за *Internet Explorer*, команда *Firefox* так и сыплет новыми основными релизами. Однако спросите человека с улицы, что нового в *Firefox 9*, и он, скорее всего, пожмет плечами в ответ. Вероятно, повышенная частота релизов несколько охладила интерес публики: вместо нового *Firefox*, который появляется ежегодно под звуки фанфар, и о новых функциях размышлял журналы и сайты, мы теперь ждем новую версию через каждые несколько месяцев, и как-то не рвемся узнать, что же в ней нового. Однако с версии 4.0 у браузера наблюдается солидный прогресс:

5 С этим релизом Сеть стала намного симпатичнее, поскольку включили поддержку анимации CSS. Она не только позволяет сайтам выглядеть более интерактивно, но и во многих случаях устраняет потребность во Flash. Ура! *Firefox 5* начал также серьезно продвигать настройку Do Not Track (она была включена в предыдущий релиз), которая сообщает сайтам, что вы хотите отказаться от целевой рекламы. А реализация холста JavaScript и HTML5 улучшили производительность.

6 Большинство изменений были «под капотом», однако интерфейс немного приблизился к *Chrome*, с поправками панели URL. Теперь адреса отображаются серым, а доменное имя – черным, выделения ради. Заодно команда реализовала JavaScript «Scratchpad» – способ быстрой проверки кода JavaScript – и взбодрила интерфейс Web Console (который применяется разработчиками сайтов).

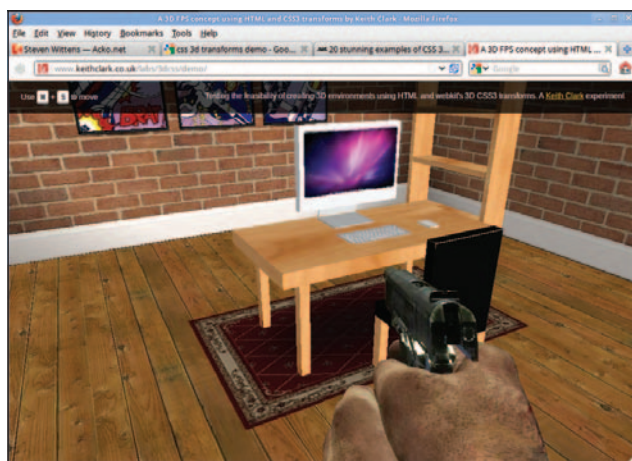
7 Здесь главным изменением, видимым пользователям, стало опущение в адресной строке 'http://'. Но команда *Firefox* на славу поработала

с печально известными проблемами браузера с памятью, заявляя, что теперь он использует на 50 % меньше ОЗУ, чем версия 4. Также была улучшена поддержка MathML.

8 Самое радостное, что эта версия принялась за решение проблемы «просачивания дополнений»: она автоматически отключает дополнения, устанавливаемые сторонними программами (обычно со всякой рекламной ерундой), и тех, кто обновил браузер с версии 7, она просит выбрать только нужные им дополнения. Мы знаем, что дополнения бывают превосходными, но многие пользователи, не имеющие технической подготовки, приходят к тому, что у них накапливаются горы совершенно ненужных им дополнений, так что это был разумный шаг.

9 Здесь основными изменениями стали улучшенная производительность JavaScript и усовершенствованное соответствие стандартам в HTML5 – ничего примечательного для пользователей Linux.

10 Этот релиз стал более гибким благодаря дополнениям, приняв почти все, написанные для версии 4 и выше. По части интерфейса, кнопка Вперед перестала отображаться, пока не появится сайт, на который можно с ее помощью перейти; также включена поддержка трансформации CSS 3D, позволившая разработчикам размещать на страницах 3D-элементы.



► Немного дергается, но демо FPS от Кейта Кларка [Keith Clark] на <http://tinyurl.com/6wprkeot> показывает возможности трансформаций CSS 3D.

11 Здесь началась борьба с *Google Chrome* за отвоеванные позиции, и появилась возможность переноса ваших сохраненных данных из этого браузера в *Firefox*. Стало возможным синхронизировать дополнения на нескольких компьютерах с *Firefox*, а для разработчиков добавлен редактор стилей CSS.

12 Теперь пользователи могут вставлять URL в менеджер загрузок, а опция View Source теперь имеет номера строк. Изрядно поправлены инструменты разработки (например, Web Console и Page Inspector).

Итак, если пользовательский интерфейс во всех релизах подвергался некоторым отладкам и обновлениям, велась также большая работа над начинкой, благодаря которой браузер готов к работе с новыми web-технологиями и может облегчить их использование для разработчиков.

Переверните страницу, и мы научим вас, как сделать из *Firefox 12* нечто особенное...

Как получить свежую версию?

Плотный график выхода релизов *Firefox* становится головной болью для поставщиков дистрибутивов. Подумайте, например, о Debian: этот дистрибутив крайне консервативен, и только очень важные исправления ошибок и заплатки системы безопасности могут рассчитывать на включение в стабильные релизы. Команда Debian должна поддерживать программы, включенные в дистрибутив, а когда вокруг скачет *Firefox* с кардинальными изменениями каждые пару месяцев, это серьезно осложняет им жизнь.

К счастью, во многих дистрибутивах есть возможность загрузки *Firefox* с сайта браузера. Зайдите на www.getfirefox.com, и вы увидите кнопку Free Download. Через несколько мгновений у вас в домашней директории появится файл **.tar.bz2** – в этом примере у нас был **firefox-12.0.tar.bz2**. Это сжатый архивный файл, так что распакуйте его командой

```
tar xfv firefox-12.0.tar.bz2
```

Создастся новая директория, со всей прямотой названная **firefox**, а запускаемая программа внутри

именуется (довольно логично) 'firefox'. Так что можно просто ввести

```
firefox/firefox
```

Теперь вы можете удалить файл **.tar.bz2**. Если все нормально, запустится браузер, и затем вы сможете добавить значок быстрого запуска на рабочий стол или в менеджер окон. Помните, что это – локальная установка *Firefox* ручной работы, поэтому ваш менеджер пакетов ее обновлять не будет. Однако вы можете постоянно заглядывать на сайт *Firefox* и загружать более новые версии тем же манером.

Путь к просмотру небес

Вот наша подборка самых полезных трюков, настроек и расширений, с которыми вы превратите *Firefox* в идеальный браузер.

Мы всегда восхищались командой *Firefox* за то, что они не пытаются делать ничего чрезмерного. Соблазнительно было бы наштамповать браузер невероятным количеством дополнительных функций и приспособлений, но все же основной его приоритет – корректное отображение страниц. Поэтому нас всегда радовала применяемая в браузере система дополнений (за исключением проблем с совместимостью), которая позволяет прикрутить к основе добавочные функции. Вы их получите, зайдя на <http://addons.mozilla.org> – и сейчас мы перечислим наших фаворитов среди модулей расширения.

DownThemAll

Стандартное окно загрузок [Downloads] в *Firefox* довольно рудиментарно, и данное дополнение было разработано для обеспечения большей гибкости. Вы можете загрузить все ссылки на странице, или все изображения, или все видео, или ввести собственные постоянные выражения для поиска совпадения – практически все, что хотите. Это крайне полезно, если у вас на сайте есть галерея изображений – например, фотоальбом – и вы хотите сохранить их все одним махом. *DownThemAll* отличается исключительной настраиваемостью и поддерживает параллельные загрузки, немного увеличивающие скорость. После установки оно будет находиться в меню Tools.

«Web Of Trust объединяет опыт тысяч пользователей.»

NoScript

Как только вы его установите, JavaScript, Java и Flash на большинстве сайтов немедленно заблокируются. Зачем это надо? Ну, их активация подставляет вас под атаки типа кросс-сайт скриптинг. *NoScript* делает ваш браузер не только более безопасным, но и более быстрым, поскольку JavaScript по большей части затрагивает всякую презентационную ерунду, а не содержание сайтов. Вы можете включить использование скриптов на сайтах из белого списка, и в *NoScript* уже предусмотрен белый список для наиболее популярных адресов сети. Это особенно рекомендуется, если вы работаете со своими банковскими данными онлайн.

Web Of Trust

А это особенно интересно для родителей, желающих защитить своих детей от вредного контента, или тех, кто не хочет случайно угодить на сайт с порнухой или вредоносным содержанием. *Web Of Trust (WOT)* объединяет опыт тысяч пользователей, позволяя вам увидеть рейтинги сайтов по их надежности, конфиденциальности и безопасности для детей. Его можно на-



► *Web Of Trust* собирает обзоры сайтов, помогая вам избегать скользких мест.

строить на блокировку определенных сайтов или на показ рейтинга сайтов при получении результатов от поисковой машины. Инструментов подобного рода немало, но *WOT* так хорошо интегрируется с браузером, что мы считаем его лучшим в своем классе.

FireFTP

Если вы соединяетесь через *Firefox* с сервером FTP, перед вами предстает список файлов и директорий, и ничего более. *FireFTP* превращает браузер в полноценный клиент FTP с возможностью загрузки локальных файлов, проведения проверок целостности и синхронизации директорий. Он также поддерживает SFTP, SSL-шифрование и редактирование удаленных файлов. Есть множество отличных клиентов FTP, но очень полезно, когда именно этот находится всего в паре

Укрепите свою безопасность

С настройками по умолчанию *Firefox* является достаточно безопасным и сохраняющим конфиденциальность браузером. Если, например, вы использовали *Google Chrome*, то вы знаете, что он постоянно обменивается информацией с серверами *Google* для обеспечения дополнительных функций, таких, как автозаполнение адресной строки и предупреждения о потенциально вредоносном содержании сайтов (конечно, все эти функции можно отключить). *Firefox* не настолько «разговорчив», но для его улучшения можно кое-что сделать.

Сначала зайдите в Edit > Preferences, затем в панель Privacy, и поставьте флажок в окне Tell websites I do not want to be tracked [Сообщать сайтам, что я не желаю быть отслеженным]. Таким

образом сайты получают информацию, что они не должны устанавливать вам cookies для отслеживания ваших действий в Сети – например, какие еще сайты вы посетили, какие слова для поиска ввели, и т.д. К сожалению, это только предложение – юридически оно не является обязательным к исполнению, и поскольку на онлайн-рекламе делаются немалые деньги, отнюдь не все сайты его используют. Но лиха беда начало.

В выпадающем окне History выберите Custom Settings и уберите галочку из окна Accept thirdparty cookies [Принимать сторонние cookies]. Это означает, что только тот сайт, где вы в данный момент находитесь, сможет хранить в браузере cookies, а с других сайтов (например, рекламных) они появ-

ляться не могут. В большинстве случаев это неплохо работает, но на некоторых сайтах эту функцию надо сперва активировать.

Если вы хотите выполнить в Сети определенную задачу, и вам не нужно, чтобы браузер сохранил следы вашей работы, перейдите в меню Tools и выберите Start Private Browsing. Начиная отсюда и до момента закрытия браузера или возврата в меню Tools, никакие ваши действия (кроме добавления в закладки) не будут записываться браузером. Это полезно, если надо связаться со своим банком онлайн и в ваши планы совершенно не входит предоставление вредоносным сайтам шансов злонамеренно воспользоваться вашими cookies или иной информацией.

щелчков мыши от вас. Разработчик также создал *FireSSH*, клиент SSH, который сидит в окне браузера и написан полностью на JavaScript.

Greasemonkey

Это называется расширением, но по сути является платформой для тысяч скриптов, написанных пользователями со всей Сети. Эти скрипты на ходу преобразуют web-страницы, изменяя их внешний вид и поведение. Вам ненавистна панель Recommended Videos на YouTube? Есть скрипт *Greasemonkey*, который избавит вас от нее. Мечтаете о чем-то вроде опции форматирования в сообщениях *Gmail*? Ба, кто-то уже написал скрипт *Greasemonkey*, чтобы ее активировать. На www.userscripts.org их множество, и найдя те, которые вам нужны, вы уже не сможете жить без этого расширения.

Firebug

Firebug добавляет к *Firefox* множество инструментов web-разработки, позволяя редактировать и отлаживать HTML, CSS, JavaScript и много чего еще изнутри web-браузера. Вы можете проделывать это также в режиме реального времени, просматривая содержание страницы в верхней панели и редактируя ее в нижней. Здесь есть весьма богатый функциями отладчик JavaScript, а также возможность отслеживать сетевую активность между вашим браузером и сервером, выявляя узкие места. Если вы пишете что-то Ajax'овое, можно даже на ходу отслеживать XMLHttpRequests. Среди разработчиков он весьма популярен, и наглядно демонстрирует, на что способно расширение API.

RightToClick

Вы же знаете эти нудные сайты, которые используют JavaScript для замещения контекстного меню, выводимого правым щелчком мыши, не дают вам выделять текст и делают еще много всяких глупостей? Похоже, они захватывают Сеть, но *RightToClick* борется с ними. С этим дополнением вы можете выбирать, какие функции JavaScript разрешено активировать странице, при-



Хотите, чтобы *Firefox* вел себя, как *Vim*? Установите *Pentadactyl* и дайте мышам отпуск.

чем на очень впечатляющем уровне. Вы можете, например, отключить любые таймеры JavaScript на странице, что поможет избавиться от ежесекундно появляющихся назойливых всплывающих окон с «напоминанием» о необходимости зарегистрироваться. Или не позволять сайту захватить управление клавиатурой. Это своего рода болеутоляющее для работы в Сети.

Pentadactyl

И, наконец, для поклонников *Vim*, или для тех, кто не шибко ладит с мышью и желает, чтобы *Firefox* больше опирался на работу с клавиатурой, *Pentadactyl* предоставляет возможность управления браузером с клавиатуры в стиле *Vim*, так что вы можете набрать `:open` следом за URL и перейти на сайт, или `:q`, чтобы закрыть браузер. Вы можете также просматривать страницы с помощью клавиш H, J, K и L, и нажимать на D, чтобы закрыть вкладку. В нашем



Что день грядущий нам готовит?

Firefox 13 щегольнет солидными изменениями в интерфейсе пользователя, например, новым видом функции Most Visited при открытии новой вкладки. Как и аналоги в *Chrome* и *Opera*, она отображает таблицу миниатюр ваших любимых сайтов, и для перехода на сайт достаточно нажать на одну из миниатюр.

Мы не вполне уверены в том, что это такое уж хорошее добавление – ну да, выглядит оно мило, но миниатюры слишком малы, чтобы различить текст, а вид любимых сайтов вам и без того знаком. Но если чуточка красотой поможет переманить пользователей других браузеров, вы не услышите от нас ни слова жалобы.

Также будет новая начальная страница, предоставляющая доступ к закладкам, закладкам, журналу и т.д. через большие значки, вместе с обычной строкой поиска Google. Эти функции помогут сделать *Firefox* более удобным для работы на устройствах с сенсорным экраном, поскольку нажимать на крупные значки намного удобнее, чем шарить по пунктам меню.

Еще одно улучшение интерфейса – мягкая прокрутка, которая раньше была опцией, а теперь будет включена по умолчанию. Когда вы медленно перемещаетесь по странице с помощью мыши или пальцами на сенсорной площадке ноутбука, страница будет прокручиваться очень мягко, приращи-

вая пиксели, а не рывками на много пикселей за раз. Это чуть больше нагружает ЦПУ, зато смотрится отлично. И, наконец, *Firefox 13* будет поддерживать протокол SPDY, альтернативу HTTP от Google, для более быстрого просмотра страниц.

Определение состава функций *Firefox 14* и *15* пока не завершено, но есть планы по фоновому обновлению в немом режиме (таком же, как в *Google Chrome*) и размещению панели Download по нижней части экрана, вместо обычного окна. Будет также проделана работа по решению текущих проблем с браузером, и поскольку команда *Firefox* получает солидную обратную связь, они смогут исправить ошибки и улучшить время отклика. **LXF**



Что за штука...

OpenCL

Марко Фиоретти представляет открытый стандарт для процессоров по обработке графики, с которым многие приложения существенно ускорятся.

В Начнем с основного вопроса: так что же такое OpenCL?

О Дословно: Open Computing Language (www.khronos.org/opencl) – открытый стандарт для параллельного программирования на GPU.

В А нельзя ли для начала попроще? Что такое параллельное программирование?

О На самом деле, это проще, чем кажется (если, конечно, не вам этим заниматься). Параллельное программирование – такой способ написания программ, при котором много схожих операций выполняются параллельно (если это поддерживается оборудованием) – то есть одно-

временно, а не последовательно, когда одна операция должна закончиться, чтобы началась другая.

В То есть параллельное программирование заставляет ПО работать быстрее. И насколько?

О Многие сложные программы будут работать быстрее. Насколько? Однозначно не скажешь. Части обычной программы, использующей для обработки данных вложенные циклы, могут ускориться вдвое. Программы, состоящие преимущественно из математических вычислений – значительно быстрее.

В Так, с первой частью более-менее понятно, теперь про GPU, пожалуйста.

О GPU [Graphic Processor Units] – это специализированные чипы, выполняющие всю тяжелую работу по обработке графики, от цифрового ТВ до 3D-анимации и супер-реалистичных видеоигр. Обычные процессоры, на которых выполняются операционные системы, не оптимизированы для подобных существенно параллельных задач. А GPU, наоборот, предназначены именно для этого.

В каком-то смысле, OpenCL – это открытый аналог Compute Unified Device Architecture (Cuda) от Nvidia, которая тоже ориентирована на более

эффективное исполнение кода с использованием графических процессоров. Удивительно умная штукавина.

В Так или иначе, OpenCL был создан для ускорения обработки 3D-графики на настольных компьютерах, так?

О Вовсе нет! Дело в красоте OpenCL и его значимости (наряду с открытостью, конечно). Судя по последним тенденциям, наличие такого стандарта просто необходимо.

В О каких, собственно, тенденциях идет речь?

О Их несколько. Первая – рост производительности GPU для обычных настольных компьютеров. Они стали настолько мощными, что использовать их только для 3D-игр и видео просто стыдно.

В Да, в этом смысле у параллельного программирования действительно неплохие перспективы. А что еще?

О Повсеместность GPU. Сегодня специальные чипы для параллельного программирования вшиты в каждый смартфон или планшет.

Но, как правило, они не задействованы для работы с текстами, электронными таблицами или сетевым обозревателем. Упростив для сложного интерактивного ПО доступ к этой вычислительной мощности в реальном времени, мы по производительности приблизим мобильные устройства к полноценным настольным системам.

В И много ли нынче ПО, работающего таким образом?

О Пока что ускорение GPU используется лишь в некоторых программах. Даже *Photoshop*, *Camtasia* и *Matlab* только начинают внедрять эти функции.

В Но если параллельное программирование так замечательно, почему же его так мало где используют?

О Отчасти потому, что сама идея применения ускорения GPU для не-графического ПО вышла из научной компьютерной среды сравнительно недавно. Другая причина в том, что процессоры GPU на портативных устройствах тоже не так давно стали настолько мощными и распространенными, чтобы оправдать введение специального стандарта. То же касается облачных вычислений.

В Облачных вычислений? А вы что имеете в виду?

О Облачные вычисления – это средства удаленного доступа к вашим файлам и программам, с любого компьютера, планшета или смартфона. Вы же слышали об этом, правда?

В Еще бы. И уже пользуюсь этим с Gmail, Google Docs и Dropbox. И что?

О Параллельное программирование на основе GPU добавляет ко всему этому аппаратное ускорение. Благодаря виртуализации, удаленный

сервер, оборудованный самыми современными GPU, способен, например, обрабатывать видеопотоки параллельно для нескольких пользователей.

В Здорово! Теперь вы меня убедили, что открытый стандарт для кросс-платформенного параллельного программирования на основе GPU необходим. Насколько с этим справляется OpenCL?

О OpenCL должен существенно упростить написание эффективного, портируемого кода, работающего на многих комбинациях многоядерных CPU, GPU и других параллельных процессоров.

В А почему вы употребили выражение «на комбинациях»?

О Потому что именно это от OpenCL и требуется. Обычные программы не могут выполнять много параллельных задач просто потому, что работают на традиционных (то есть последовательных) CPU. А задача OpenCL в том, чтобы максимально перенести эту нагрузку на GPU, чтобы все выполнялось параллельно.

В Если я правильно понимаю, это значит, что программы, использующие OpenCL, тоже обязаны иметь смешанную структуру. Так?

О Именно! OpenCL вызывает центральный процессор, где запускается ядро обычного приложения. GPU называются вычислительными устройствами. Каждое такое устройство включает одну или более вычислительных единиц, которые, в свою очередь, содержат один или более элементов обработки.

«Язык OpenCL – это своего рода специальная версия C.»

В Какая роль в этой ситуации отведена хосту?

О Исполняемый им код контролирует взаимодействие с пользователем, управление памятью и все задачи OpenCL, включая передачу данных на GPU и его запуск. Когда все устройства заканчивают работу, хост получает данные-результаты и переходит к следующему шагу. OpenCL предоставляет стандартные API для распознавания устройств и обработки данных и команд.

В А какова роль программиста в данном процессе?

О Разработчики должны явно указать в коде, какая его часть исполняется на главном процессоре, а какая – на устройствах, пользуясь спецификацией OpenGL.

В То есть параллельный код исполняется только на устройствах?

О Да, это так. Иногда говорят, что волшебство OpenCL вершится на устройствах. Именно

там исполняется код на языке OpenCL – это своего рода специальная версия C.

В Но ведь API – общий программный интерфейс. Как же хост сообщает устройствам, что именно им делать?

О Для этого есть особые функции, называемые ядрами вычислений. Это своего рода точки входа для устройств OpenCL, которые их активируют и сообщают им задания.

В А что начинает происходить потом?

О Выделенный ядром фрагмент кода, который на языке OpenCL называется рабочим элементом, исполняется параллельно много раз, с новыми данными. Такие элементы обычно организуются в группы и диапазоны, определяющие порядок запуска.

В Что требуется для совместимости с OpenCL?

О В основной спецификации перечислены все требования, которым должна соответствовать полноценная программная/аппаратная среда для совместимости с OpenCL. В официальном профиле конфигурации для портативных и встроенных устройств указаны необходимые условия для этих двух случаев. Ну и, конечно, стандарт также определяет язык, упомянутый нами как «специальная версия C». Наконец, есть расширения, то есть необязательные функции. Они включены в стандарт потому, что кажутся достаточно интересными, чтобы в будущем также войти в основную спецификацию OpenCL.

В Каков в настоящий момент статус OpenCL?

О OpenCL 1.2, имеющий обратную совместимость с версиями 1.0 и 1.1, но поддерживающий большее число устройств и ядер, вышел в ноябре 2011. В него были добавлены средства взаимодействия с OpenGL и OpenGL ES, улучшены характеристики производительности и поддержка DirectX 9 и 11.

В А есть ли хотя бы один пример реализации OpenCL в свободном ПО?

О Есть. Присмотритесь к Portable OpenCL (<http://pocl.sourceforge.net>) – это проект, целью которого является «улучшить переносимость программ на OpenCL», что значит обеспечить более-менее стабильную производительность для любых хостов и устройств, минимизируя необходимость ручной оптимизации. Дополнительную информацию вы можете получить на <http://opencl.codeplex.com/wikipage?title=OpenCL+%20Tutorials+%20-%20%201. LXF>



По рецептам доктора Брауна

Д-р Крис Браун

Доктор обучает, пишет и консультирует по Linux. Ученая степень по физике элементарных частиц ему в этом совсем не помогает.

Мелким шрифтом

Мы все это делаем. Мы покупаем билет через интернет-сайт железнодорожной компании. Мы загружаем закрытые драйверы для видеокарты или устанавливаем модуль поддержки Flash для браузера. И нам неизбежно встречается эта штука под названием «Соглашение».

Иногда оно встроено в вашу навигацию, то есть его приходится просматривать в нескольких окнах, идущих друг за другом. Но нередко, чтобы его увидеть, надо щелкнуть по ссылке и перебраться в отпечатавшееся окно.

Сперва проверьте

Итак, мой вопрос: вы когда-нибудь, когда-нибудь, КОГДА-НИБУДЬ читаете то, что написано мелким шрифтом в этих так называемых «сворачиваемых по щелчку» соглашениях? Или просто хотите побыстрее покончить с этим и отматываете изысканно-запутанный юридический текст страницу за страницей, не читая, после чего беззастенчиво ставите галочку «Принять»?

А что произойдет, если на странице 27 между абзацами о недопустимости обратного инжиниринга и создания копий затесано обязательство отказаться от своего первенца? Имеет ли установка галочки реальную юридическую силу?

В «Венецианском купце» Порция смогла противостоять Шейлоку и защитить Антонио от того, чтобы у него вырезали кусок мяса за неуплату по векселю, но кто оборонит меня, например, от юристов Microsoft?

Напоследок признаюсь. Я только что установил на свой компьютер Ubuntu 12.04. Так как я устанавливаю только релизы «с долгосрочной поддержкой», то рабочий стол Unity увидел впервые. И знаете что? Я вправду думаю, что он не так уж плох. Да, мне известно, что он пользуется дурной репутацией, и сейчас я, несомненно, потерял 30 % своих читателей... прощайте...

chris.linuxformat@gmail.com

Эзотерическое системное администрирование из причудливых заворотов кишок серверной



Защита человека

Обезопасить серверы недостаточно. Нужно позаботиться и о пользователях.

Итак, вы ответственно относитесь к защите своих серверов: регулярно выполняете обновления безопасности, настроили брандмауэр, отключили вход в систему от имени root и задали сложные пароли.

Но закручивание гаек на серверах не гарантирует безопасности корпоративных данных, если сотрудники вашей компании текут как решето. В детстве мама не велела вам брать конфеты у незнакомцев. Но многие организации не дают этого простого совета, чтобы обезопасить своих сотрудников в киберпространстве. Если у вашей компании нет программы обучения основам безопасности, не убедить ли босса начать ее?

Институт SANS (www.sans.org) – частная американская компания, и у нее немало ресурсов вам в помощь. В верхней части диапазона – большое портфолио курсов (и с преподавателем, и онлайн) для профессионалов в сфере безопасности, обычно стоимостью в 4000 долларов. Но у этого сайта есть родич (www.securingthehuman.org), где мас-

са бесплатных материалов: например, опрос для сотрудников по теме безопасности из 25 пунктов (типичный вопрос: «Как часто вы берете информацию из офиса и работаете с ней на домашнем ПК?»), или примеры презентаций, вводящих в курс обучения пользователей основам безопасности (в т.ч. программа для детей), и удобная таблица параметров для оценки, есть ли от программы прок – часто это простые вопросы вроде «Сколько сотрудников запирают свой компьютер перед уходом с работы?» А если вам нужно напугать свое начальство, чтобы оно поддержало вас, на сайте SANS есть ссылки на страшилки из новостей.

Наконец, на сайте www.sans.org/tip_of_the_day.php вы получите совет безопасности дня; на 1 мая 2012 года он был «Закрывайте доступ к компьютеру перед тем, как контракт с работником заканчивается или вы говорите ему, что он уволен».

Ох!

С целью осведомления пользователей о безопасности, SANS также делает бесплатную ежемесячную рассылку OUCH!, доступную через сайт securingthehuman.org. Последние темы: безопасная утилизация старого мобильного; стирание метаданных из изображений и офисных документов; защищенность и конфиденциальность браузера; безопасная работа в сети; что можно и чего нельзя делать с электронной почтой. Это простые советы для не технической, но знакомой с компьютером аудитории.



➤ Сайт осведомления о безопасности – лишь один из девяти сайтов, посвященных вопросам безопасности, которые поддерживаются институтом SANS.

WWW. SITE CONF. RU

КОНФЕРЕНЦИЯ
«САЙТ-2012. СОЗДАНИЕ,
РАЗВИТИЕ И ПОДДЕРЖКА
ИНТЕРНЕТ-ПРОЕКТОВ»

27—28 сентября. Москва, Digital October

Организаторы



Пароли и политики

Во многих организациях используются странные ритуалы управления паролями и политиками. Вот несколько простых практических советов.

Сколько у вас паролей? Один на домашнем ноутбуке, другой на рабочем компьютере. Затем – учетная запись вашей почты и на корпоративном почтовом сервере. Пару учетных записей в социальных сетях, Amazon, ebay, PayPal, интернет-банке и этой штуке под названием “verified by Visa”. Это по меньшей мере дюжина, даже не считая все серверы, которыми вы управляете. Общее количество вполне может быть в два раза больше. Для многих из этих учетных записей в качестве логина используется почтовый адрес, который не является секретом. Поэтому от забвения вас отделяет только пароль.

Выбираем надежные пароли

Выбор хорошего, надежного пароля – пожалуй, единственная важная вещь, которую можно сделать, чтобы сохранить учетную запись в безопасности. Хороший пароль должен быть:

- » Достаточно длинным и сложным, чтобы противостоять и словарным атакам, и атакам методом грубого перебора
- » Запоминаемым
- » Таким, что его можно правильно набрать с первой попытки

Удовлествить всем трем пунктам довольно сложно. Некоторые практические советы вы можете найти во врезке «Четыре шага к хорошему паролю». Надежность пароля оценивается по его длине и употреблению символов различных групп. Возьмем средний пароль из восьми символов. Если он состоит только из строчных букв, комбинаций будет около 200 миллиардов. Если используются буквы в верхнем и нижнем регистрах, цифры и специальные символы (всего около 96 символов), это 7 миллиардов комбинаций. На сайте www.passwordmeter.com находится приложение, которое оценивает прочность произвольного пароля.

Самые отъявленные параноики из нас, конечно, дважды подумают, прежде чем отправить свой надежный пароль недо-

Test Your Password

Password:

Hide:

Score: 87%

Complexity: Very Strong

Minimum Requirements

- Minimum 8 characters in length
- Contains 3/4 of the following items:
 - Uppercase Letters
 - Lowercase Letters
 - Numbers
 - Symbols

Additions	Type	Rate	Count	Bonus
Number of Characters	Flat	+(n*4)	12	+ 48
Uppercase Letters	Cond/Incr	+((len-n)*2)	2	+ 20
Lowercase Letters	Cond/Incr	+((len-n)*2)	8	+ 8
Numbers	Cond	+(n*4)	2	+ 8
Symbols	Flat	+(n*6)	0	8
Middle Numbers or Symbols	Flat	+(n*2)	2	+ 4
Requirements	Flat	+(n*2)	4	+ 8

» Простая метрика для оценки сложности пароля на основе длины пароля и количества групп используемых символов.

ренному сайту, особенно такому, который не использует HTTPS. В Linux есть модули PAM, которые помогут увеличить надежность пароля. *pam_cracklib* обнаружит пароли, поддающиеся словарным атакам, а *pam_passwdqc* задаст минимальную длину пароля на основе количества различных групп символов, используемых паролем.

Политики

Есть мнение, что следует взять один надежный пароль и пользоваться им везде. Я бы не назвал это хорошей идеей. Есть сайты, которым вы доверяете и которые заботятся о сохранении пароля в секрете (например, интернет-банк). Но создавать учетные записи приходится и на многих других сайтах. Где гарантия ответственности их подхода, и умеют ли они обеспечивать защиту от SQL-инъекции и других атак? Наверняка у вас есть учетные записи, в которых в качестве логина используется адрес электронной почты. Если одни и те же логин и пароль используются на некритичном сайте и их крадут, потому что вы подключились со взломанного клиента или взломали их сервер, это компрометирует все остальные учетные записи.

Меня также слегка нервируют сайты, предлагающие задать контрольные вопросы для восстановления пароля. Это полезная возможность, но, насколько я знаю, сложность вопроса не проверяется. Поэтому в общем надежнее иметь как минимум два пароля: один для всякой ерунды, и второй – для действительно важных сайтов.

К сожалению, выбором двух сложных паролей история не заканчивается. Во многих организациях применяется политика возраста пароля, и пользователей заставляют периодически менять пароли. Логика изначально состояла в том, чтобы менять пароли чаще, чем их можно было бы взломать методом грубого перебора. Может, 30 лет назад в мире мейнфреймов это и имело смысл. Но с тех пор компьютеры стали мощнее, и время атаки методом грубого перебора значительно сократилось. В любом случае, есть и другие технологии, с которыми плохие парни могут узнать ваш пароль, например, фишинг. Мне кажется, что если вы будете менять пароль каждые 30 дней, а взламывают его три недели, тот факт, что у злодеев останется всего неделя на разграбление

Четыре шага к хорошему паролю

Рискуя причинить вам обиду за ваш интеллект, приведу четыре способа выбора надежного пароля.

» Первый способ. Возьмите два не связанных друг с другом слова, например, “goat” и “cleaner”. Вставьте между ними какой-нибудь спецсимвол, например, “goat=cleaner”. Смените пару символов на заглавные, например, “Goat=cleaneR”, и, наконец, превратите одну букву в цифру, например, “G0at=cleaneR”.

» Другой способ – взять выражение и «сжать» его, часть букв сделав заглавными, а часть превратив в цифры. Например, “I wish that I was out in the sunshine” можно превратить в “lwsht1w0itSs”.

» Третий способ – перемешать слово с последовательностью цифр. Например, из “treacle” и “654321” можно получить “t6r5e4a3c2l1e”.

» Наконец, для тех, у кого не хватает фантазии ни на один из этих способов, предусмотрены разные команды, генерирующие пароли. Обычно нужно что-то проделать с выводом `/dev/urandom` – например, команда:

```
tr -dc a-zA-Z0-9 < /dev/urandom | head -c 10
сгенерирует пароль из символов из набора a – zA-Z0-9. Он будет длиной 10 символов и до ужаса случайным, так что его наверняка придется записать.
```

Есть более сложные утилиты, например *apg*, которые умеряют свой пыл до чего-то удобопроизносимого, если, конечно, вы сможете произнести вслух “ghavTi9graf”. И, разумеется, существуют такие сайты, как freepasswordgenerator.com, которые сгенерируют для вас пароль указанной длины из предложенного набора символов.

26 character set (e.g. all lower-case letters)			
Length	Combinations	Case A	Case B
6	309 x 10 ⁶	30 sec	< 1 sec
8	200 x 10 ⁹	348 min	3.5 min
10	141 x 10 ¹²	163 days	39 hours

62 character set (lower case, upper case, digits)			
Length	Combinations	Case A	Case B
6	57 x 10 ⁹	1.5 hours	56 sec
8	218 x 10 ¹²	253 days	60 hours
10	839 x 10 ¹⁵	2,600 years	26 years

Case A: Fast PC testing 10,000 passwords/sec
Case B: Cluster or supercomputer testing 1,000,000 passwords/sec

► **Время на взлом пароля методом грубого перебора зависит от длины пароля и от набора составляющих его символов.**

вашей жизни, прежде чем вы снова смените пароль – слабое утешение. Если вы действительно хотите реализовать механизм возраста пароля в Linux, можно воспользоваться полями из `/etc/shadow` для задания срока службы пароля и абсолютной даты истечения учетной записи. Лучший способ задать эти параметры – команда `chage`; детали можно узнать на странице руководства. Впрочем, знайте, что самая популярная среди сотрудников реакция на требование смены пароля – добавление цифры в конец пароля и увеличение этой цифры.

Грубый перебор...

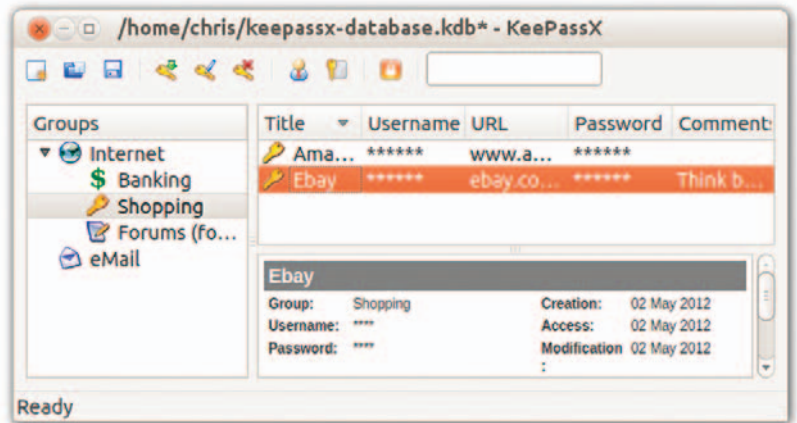
Если у злоумышленника есть доступ к свертке пароля (в Linux это доступ к `/etc/shadow`), то он сможет выполнить атаку методом грубого перебора в офлайн-режиме. Можете оценить, сколько времени понадобится на взлом пароля по его длине, длине набора символов, из которых он составлен, и вычислительной мощности в распоряжении взломщика.

Чтобы дать вам представление о цифрах, для пароля из восьми символов нижнего и верхнего регистра существует 53 триллиона комбинаций, и на быстром компьютере его можно взломать за 62 дня, а на суперкомпьютере – за 15 часов. Подробности имеются на сайте www.lockdown.co.uk/?pg=combi.

Конечно, вряд ли кто-то разорится на аренду 15 часов суперкомпьютера, чтобы взломать ваш пароль в Facebook. Более вероятно, что пароли будут украдены с помощью мониторинга нажатий клавиш на скомпрометированном компьютере или перехвачены со взломанного сайта. Проведение такой атаки путем попыток входа в систему вряд ли завершится успешно, особенно по сети. С одной стороны, это гораздо медленнее, чем сгенерировать пароль, получить свертку и сравнить ее со сверткой в `/etc/shadow`. С другой стороны, это оставляет обширный след в файлах системных журналов. А в Linux есть утилиты вроде `fail2ban` – с их помощью можно заблокировать IP-адреса, откуда производятся неудачные повторные попытки входа в систему.

...и грубое невежество

В реальности люди часто выбирают в качестве паролей клички собак, и становятся легкой добычей злоумышленников, пользующихся словарными атаками. Программы, выполняющие эту работу, становятся все умнее, обычно используя сочетания “root” (словарного слова) и короткого суффикса, такого как “!” или “123”. Некоторые также проверяют замену букв на цифры, с которой



► **Одна из многих утилит управления паролями *keepassx* причесывает все ваши пароли за одним главным паролем. Вы уж постарайтесь его не забыть.**

“hotel” становится “h0t3l!”. А некоторые пытаются получить записи из адресной книги, почтового архива или другие текстовые строки с жесткого диска.

Например, Брюс Шнайер [Bruce Schneier] проанализовал 100 000 паролей, полученных фишинговым сайтом MySpace. Он обнаружил, что 12 % паролей состояли из словарных слов с цифрой – самым популярным был “password1” (см. www.schneier.com/essay-144.html). Это своего рода прогресс: десять лет назад самым популярным паролем был “password”. Шнайер написал о надежности паролей, пожалуй, больше чем кто-либо другой. В 2006 году он сказал, что любой пароль, достаточно простой для запоминания, можно взломать на современных компьютерах.

Но взглянем правде в глаза: выбрать и запомнить разные сложные пароли для каждого сайта почти невозможно, особенно если вы будете менять их каждый месяц (кто-нибудь на самом деле это делает?). Одно из решений – воспользоваться «бумажником паролей», программой, которая хранит все пароли в зашифрованном виде под одним главным паролем.

Несколько лет назад меня всерьез обеспокоило то, что данные всех моих учетных записей хранятся в маленькой черной книжечке, и я написал для этого небольшую утилиту. По сути, это скрипт-обертка `gpg` и `grep`, но ей я хотя бы доверяю. Собственную программу писать не пришлось, потому что графических менеджеров паролей Linux больше, чем вы можете представить.

Например, *KeePassx* (www.keepassx.org) может хранить имена пользователей, URL-адреса, вложения и комментарии в базе данных, зашифрованной 256-битным TwoFish или AES. Для доступа к базе данных требуется просто указать главный пароль; вы можете настроить *KeePassx* так, что для аутентификации, кроме пароля, потребуется и файл. Файл сгодится любой, но если создать файл с длинной случайной строкой и записать его на USB-брелок, то фактически вы обзаведетесь двухфакторной аутентификацией.

Двухфакторная аутентификация

В схеме двухфакторной аутентификации нужно представить два независимых удостоверения своей личности. Чаще всего первое – «то, что вы знаете» (пароль или PIN-код), а второе – «то, что у вас есть» (банковская карта или ключ на USB-брелке). Реже одним из доказательств выступает «то, чем вы являетесь»

(например, отпечаток пальца или даже изображение сетчатки глаза). Скажем, в банке используется двухфакторная аутентификация – вам нужно вставить карту и ввести PIN-код.

Обратите внимание, что запрос двух паролей НЕ является двухфакторной аутентификацией!

Сетевая файловая система

NFS – протокол общего доступа к файлам в системах UNIX и Linux. Он зрелый, стабильный, простой в администрировании и невероятно полезный.

Пропустя говоря, NFS позволяет смонтировать часть файловой системы сервера в файловую систему компьютера клиента. При должной настройке он прозрачен, то есть конечные пользователи не ощущают разницы в доступе к локальным и удаленным файлам. NFS был одной из первых вещей в сетях UNIX (они на много лет опередили сети Linux), с которыми я столкнулся, запуская сеть рабочих станций Sun в Университете Шеффилда. Мы пользовались им для централизации домашних каталогов пользователей на нескольких серверах, и я сомневаюсь, что хоть один из этих пользователей знал, что происходит.

Чтобы превратить компьютер с Linux в файловый сервер NFS, нужно установить демоны *mountd*, *nfsd* и *portmapper*. В моей тестовой системе Fedora 15 все они находились в пакете *nfs-utils* (он может быть установлен по умолчанию). На сервере файл */etc/exports* определяет, какие части файловой системы экспортировать и каким клиентам. Несколько строк из этого файла с пояснениями показаны на рисунке. Если вы пришли из Windows, то эти экспортированные каталоги аналогичны «каталогам с общим доступом», но обратите внимание, что в Windows общему каталогу дается имя (оно может отличаться от имени самого каталога), а в NFS этого дополнительного уровня именования нет. Имя экспортируемого каталога в файловой системе сервера – это имя, под которым его видит клиент.

Болезнь .d

NFS недавно подхватил болезнь *.d*, общий симптом современных дистрибутивов Linux, в которых один файл настройки (*/etc/exports*) превращается в каталог таких файлов (в */etc/exports.d*); но если вы хотите поместить все в файл «верхнего уровня» (*/etc/exports*), такая возможность еще есть.

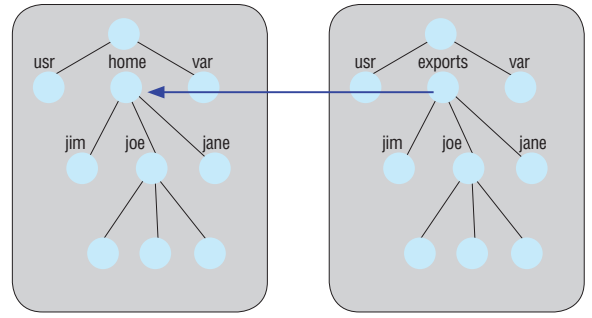
Специальной клиентской программы (вроде *Thunderbird* для почтового сервера POP3) на стороне клиента нет; настоящая клиентская часть NFS находится на уровне виртуальной файловой системы, что позволяет смонтировать экспортированную файловую систему NFS в файловую систему клиента. Для монтирования вручную (нужны права root) скомандуйте

```
# mount -t nfs pluto.example.com:/exports /home
Сделайте монтирование «постоянным», добавив в /etc/fstab строку
pluto.example.com:/exports /home nfs hard,intr 0 0
```

➤ В файле */etc/exports* (на сервере) задаются экспортируемые файловые системы. Для более подробной информации запустите `man exports`.

Да вы кем себя воображаете?

При выполнении запроса серверу NFS числовые идентификаторы пользователя (UID) и группы (GID) клиента передаются серверу, где они используются для проверки прав доступа. Это работает прекрасно, если вы используете LDAP (или даже старые добрые «Желтые страницы»), потому что идентификаторы пользова-



Файловая система клиента Сервер **pluto.example.com**

➤ Результат запуска команды `mount -t nfs pluto.example.com:/exports/home` (на клиенте).

теля будут одинаковы на всех компьютерах. Но если нет, может возникнуть ситуация, при которой UID Джона на клиенте будет совпадать с UID Мэри на сервере, и тогда Джон завладеет файлами Мэри. А это нехорошо.

Особый случай – пользователь root. Если я работаю на клиенте от имени root, мой UID на сервере изменяется так, что на сервере я выступаю как пользователь “nfsnobody” (или “nobody”, в зависимости от дистрибутива). Обычно это непривилегированная учетная запись, потому что (простите за злоупотребление отрицаниями) никому (nobody) не принадлежат никакие файлы. И хотя на клиенте я root, на сервере я получу права доступа для всех файлов как «остальные». Отключить этот механизм можно, добавив параметр *no_root_squash* в */etc/exports* (но только если вы полностью доверяете клиентским компьютерам). Добавив параметр *all_squash*, вы включите этот механизм для всех пользователей.

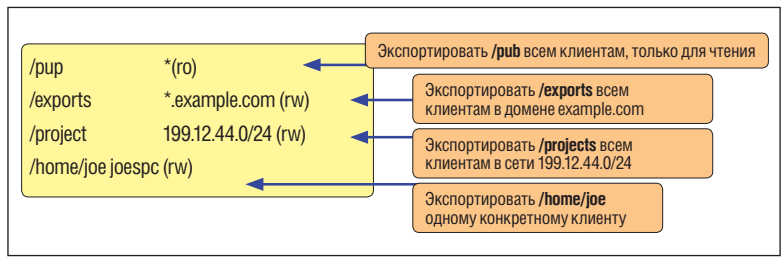
Узнать больше

На map-странице *exports* синтаксис файла */etc/exports* описан более подробно, чем я показал на рисунке. На map-странице для NFS описаны многие опции, которые можно использовать с командой *mount* или добавить в файл *fstab*. В частности, можно исследовать параметры “hard” и “soft”, которые определяют поведение (на стороне клиента), если он пытается получить доступ к удаленному файлу, когда сервер выключен. Параметр “hard” заставит клиента ждать до тех пор, пока сервер не включится. С параметром “soft” клиенту будет отправлена ошибка по истечении таймаута операции чтения или записи.

Статей по безопасности и защите файлов очень много. Они сложноваты, но скрасят вам дождливый день... наверное. **LXF**

УТИЛИТЫ ПОЛЬЗОВАТЕЛЯ

На сервере NFS вам понадобятся несколько утилит командной строки. Первая – *exports*, которая передает настройки из файла */etc/exports* демону *mountd*. Если вы измените файл */etc/exports*, нужно запустить *exportfs -a*. Вторая – диагностическая утилита *showmount*, которая показывает, какие точки монтирования NFS смонтированы на заданном сервере. С параметром *-e* она также покажет таблицу экспорта сервера.



Наши эксперты помогут вам с любым приложением Linux!



ЕВГЕНИЙ БАЛДИН
Упомянул PostgreSQL на защите кандидатской диссертации по физике элементарных частиц.

Linux в телевизоре

Нужно расставить приоритеты: что для нас важнее – попасть в телевизор или какое-то дурацкое убийство? «Южный парк», сезон 1, серия 2

В какое время мы живем! Человечество в среднем тупеет или как минимум всячески сопротивляется обратному процессу, а бытовые приборы умнеют. Никому уже не удивишь «умным телефоном»: почти любой его носит в заднем кармане штанов/ дамской сумочке; непонятно, что для аппарата опасней. Настает время «умных телевизоров».

В случае телефонов уже примерно ясно: это либо «айфон», либо «Андроид». С телевизорами же не все так просто: пока еще всяк лепит кто во что горазд. Объединяет эти попытки одно: кроме маргинальных направлений, в основе почти всех разработок лежит ядро Linux, причем обычно «зативоизированное» по самые уши. Столлмен с его GPL 3 был прав, но сейчас разговор не о нём, а о том, что же получит пользователь от поумневшего ТВ-ящичка?

Пока не много, но уже сейчас очевидно, что видеоманитоны и аналоговые антенны уходят в прошлое: их сменяют просмотр фильмов по Сети и Gismeteo с прогнозом погоды. Умные холодильники, говорят, уже тоже есть. Очевидно, скоро придет время умных чайников. Не менее очевидно, какая там будет ОС.

P.S. Осенью будущего года в продажу поступит новая приставка OUYA (звучит как Oooya) на базе Android. Подозреваю, что это лишь одна из первых ласточек, и на консольном рынке грядет смена предпочтений.

E.m.Baldin@inp.nsk.su

В этом месяце вы научитесь...



Подделывать шедевры 66
Изготовим убедительные коллажи в стиле Хокни при помощи вашего фото, **Ника Вейча** и последней версии **GIMP**.



Продвигаться с Pi 70
Грэм Моррисон устанавливает пакеты приложений, играет в игры S64 и превращает Raspberry Pi в сетевое хранилище.



Добавлять диск 74
Добудьте больше места на жестком диске и установите больше дистрибутивов, следуя отличному руководству **Нейла Ботвика**.



Публиковать оболочку 76
Несравненный **Майк Сондерс** осваивает утилиту для выгрузки в Сеть сессий вашей оболочки – в названиее другом.



Устраивать волнения 80
Arduino как генератор колебаний: **Никит Вейчу** захотелось поглубже вникнуть в счетчик тактов крошечного компьютера.



Вникать в Erlang 84
Узлы – это всего лишь именованные экземпляры среды выполнения Erlang, говорит **Андрей Ушаков**. Связать их просто.



Сворачивать белок 88
Давид Хейворд превращает несколько завалившихся машин в мощную распределенную вычислительную сеть. # winning.



Серфить без графики 92
По убеждению **Тихона Тарнавского**, основное в Интернете – текст. Ну и зачем вам всякие там Firefox'ы, когда есть **ellink**?

Пропустили номер?

Узнайте на с. 108, как получить его прямо сейчас!



GIMP: Подделаем

Фотолюбитель **Ник Вейч** показывает, как превратить обыкновенное цифровое фото в убедительный шедевр поп-арта.



» После: Итоговое изображение; наверное, можно было сделать и лучше, но зато проиллюстрирован эффект.

» До: Эта фотография старенького «фиата» на фоне Колизея прекрасно нам подходит — даже после искажений автомобиль все равно узнаваем!



Наш эксперт

Когда **LXF** только появился, его держали на плаву только скрипты *Bash* от **Ника Вейча**. Потом их заменили «людьми», и это, по мнению Ника, стало шагом назад...

» Еще один пример того, что можно сделать, если не спешить и делать все правильно.

Поклонникам искусства Дэвид Хокни [английский художник, заметный представитель поп-арта 1960-х, — прим. пер.] знаком не только своими великолепными картинами, но и любопытнейшими фотографиями. Одна из его техник, которой часто подражают — фотоколлаж, в котором одна «сцена» состоит из десятков, а то и сотен отдельных изображений, наложенных одно на другое. Истинная визуальная привлекательность таких коллажей (скажу без лишней претенциозности и не отклоняясь от темы) в том, что путем нарушения деталей и элементов сцена абстрагируется от обычной, приземленной реальности. Черт, все-таки получилось претенциозно. Важно помнить, что хотя эту технику можно применить к любой фотографии, больше всего она подходит для изображений с несколькими



явными акцентами — объектами, легко узнаваемыми, даже если немного размыть их границы или изменить цвета. Поэтому среди примеров лучших работ самого Хокни в этой области — известные предметы и вещи, такие как деревья, здания, дороги и, достаточно нередко, текст.

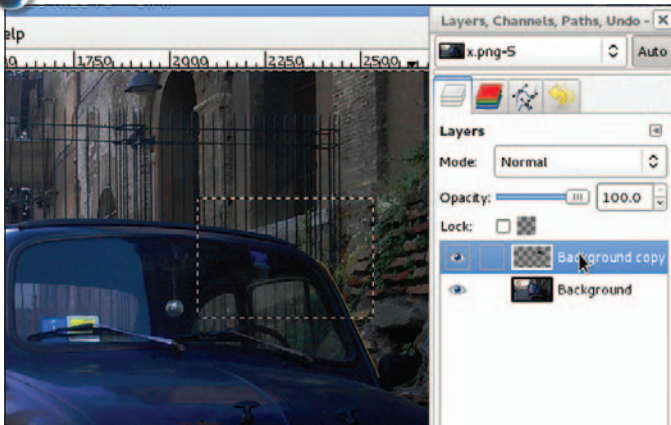
Итак, для получения лучшего результата выберите фотографию получше. При «правильном» подходе делаются тысячи кадров, обычно пленочной камерой; варьирование параметров съемки и положения фотографа, а также времени суток, погоды, света и движения предметов в кадре позволяет создать разнообразные реплики отдельных изображений. Однако мы тут пойдем на маленькую хитрость и с самого начала примем за основу единственное изображение: хотя цифровой камерой и несложно отснять хоть 500 фотографий, все равно пришлось бы менять те или иные фрагменты.

Начнем с выбора изображения и слоев, с помощью которых мы расчленим фотографию на фрагменты. Затем мы пройдем по нескольким этапам «изменения» отдельных изображений и, наконец, соберем все вместе в финальный вариант нашего поп-арт-шедевра.

ПСЕВДО-ХОКНИ

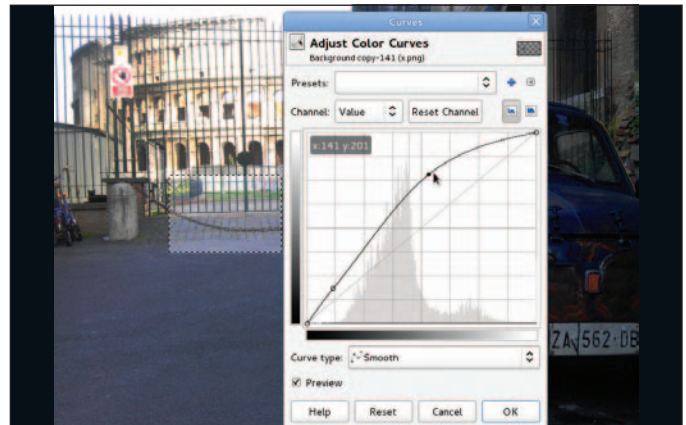


Шаг за шагом: Создаем коллаж



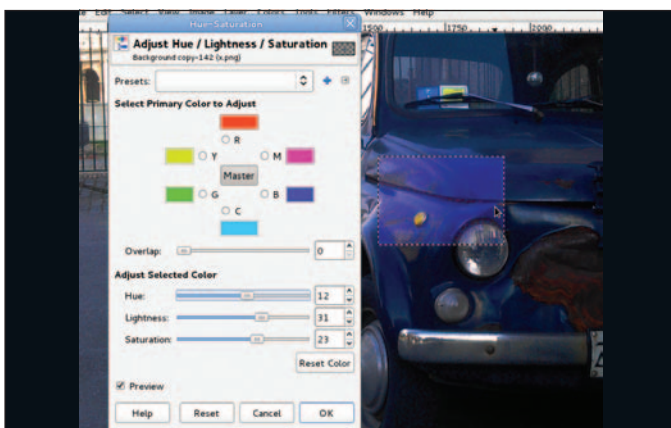
1 Создаем слои

Загрузив изображение, добавьте к нему альфа-канал, так как по умолчанию его нет (Layer > Transparency > Add Alpha Channel [Слой > Прозрачность > Добавить альфа-канал]). Идея состоит в том, чтобы пользоваться множеством слоев. Откройте окно Layers [Слой] (Control+L). Щелкните правой кнопкой мыши на существующем слое и выберите Duplicate Layer [Создать дубликат слоя] в открывшемся меню. Будет создан и выбран новый слой. С помощью инструмента Selection [Выбор] задайте прямоугольную область. Нажмите Ctrl+I, чтобы инвертировать выделение, и Ctrl+X – для очистки всего остального на этом слое. Вот с этими кусочками мы и порезвимся.



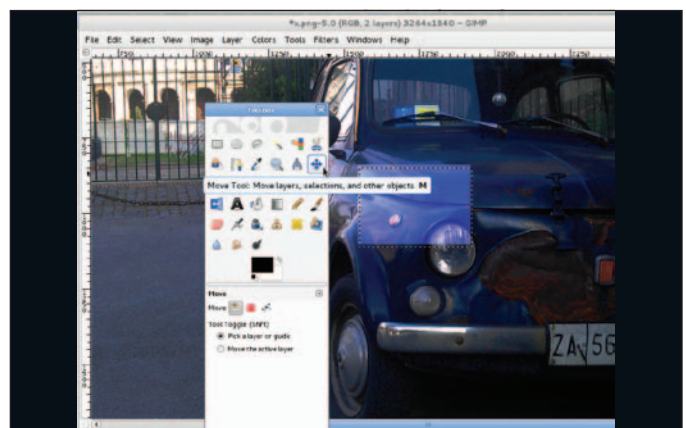
2 Изменяем кривые

Изменение контраста и яркости изображения с помощью соответствующих инструментов – это процесс проб и ошибок. Лучше применить Colour > Levels [Цвета > Уровни] (в этом случае мы получаем более тонкий контроль) или Colour > Curves [Цвета > Кривые]. Просто переместите кривую в нужное положение – если поместить ее над центральной линией, то соответствующие участки изображения станут светлее, а если под ней – темнее. Разумеется, не стоит регулировать их, «причесывая всех под одну гребенку»: одним участкам требуется больше контраста, другим – меньше.



3 Экспериментируем с цветом

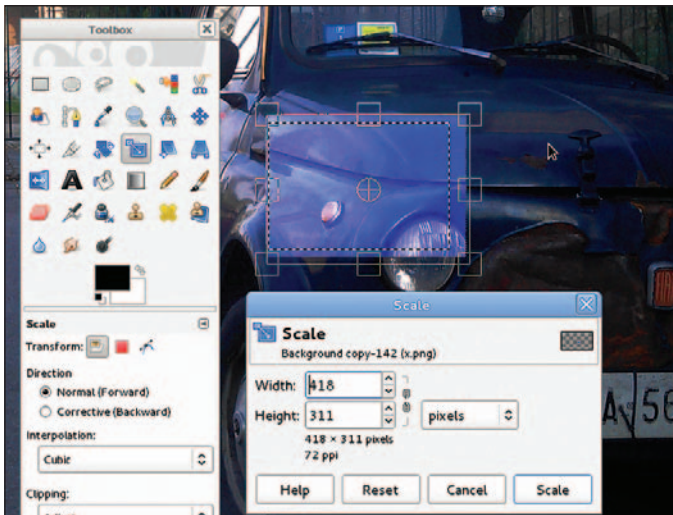
Кроме яркости и контраста изображения, можно поиграть и со значениями цвета. Это бывает особенно эффективно, когда нужно подсветить определенные объекты – как в нашем изображении, где, как вы видите, мы во многих местах усилили голубоватые оттенки автомобиля. Для этого выберите Colours > Curves и меняйте нужные каналы индивидуально. Также стоит поэкспериментировать с Colours > Hue-Saturation [Цвета > Оттенок-Насыщенность] и увеличить насыщенность – но слишком усердствовать с этим не надо, иначе изображение будет выглядеть ненатурально.



4 Все перетасуем

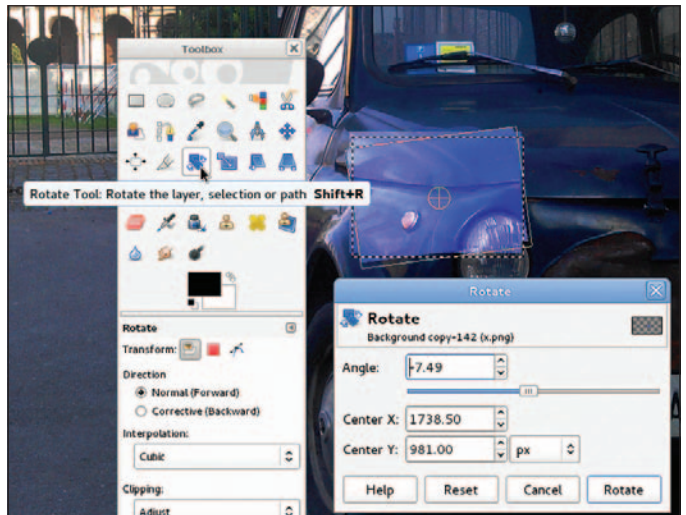
Идея подобных коллажей состоит также и в том, чтобы разрушить формы и контуры объектов сцены – это процесс удаления «изображения» и замены его «имитацией». Тут пригодны многие способы, но самый эффективный, безусловно, изменение положения фрагментов. Как и в большинстве случаев, здесь требуется некоторое искусство, но, быть может, стоит сместить все, хотя бы ненамного. Для этого нам, как вы понимаете, понадобится инструмент Repositioning [Изменение положения].

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)



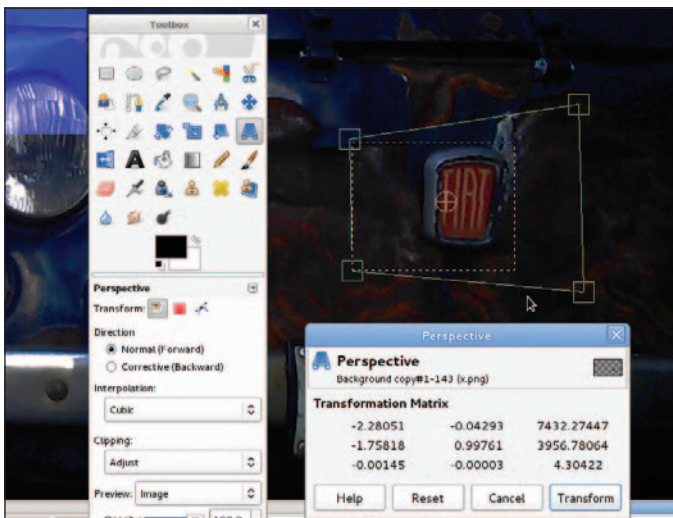
5 Изменяем масштаб

Масштабирование отдельных частей изображения не только помогает создать иллюзию, что они взяты не из одной и той же картинки, но и позволяет подчеркнуть отдельные части изображения, чтобы изобразить объекты сцены в карикатурном виде. Это часто встречается в работах Хокни, и сделать такое нетрудно. Выбрав нужную область, нажмите на инструмент Scale [Масштаб] и, удерживая клавишу Ctrl, потяните за уголок. При удерживании клавиши пропорции сохраняются, поэтому картинка при масштабировании не искажается. После масштабирования обязательно сместите изображение.



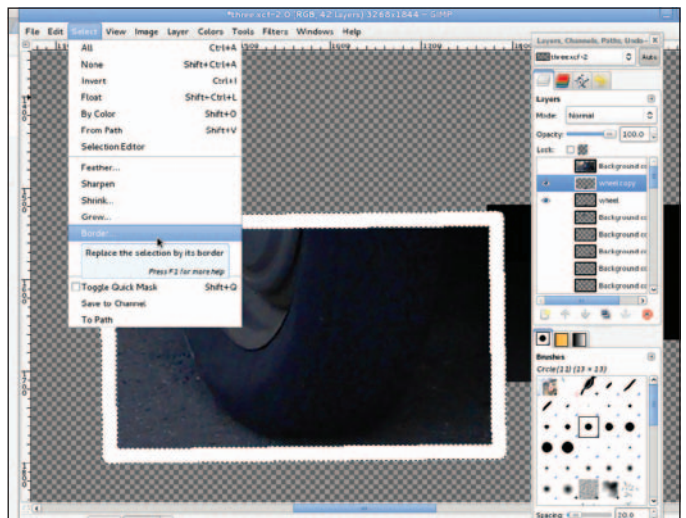
6 Поворот фрагмента

Повороту отдельных частей изображения нужно уделить большое внимание. Он позволяет разделить изображение на отдельные части и разрушить контуры объектов, и в целом это штука хорошая. Если прямоугольник выбран, лучше воспользоваться инструментом Rotation [Поворот] из панели инструментов. Он автоматически определит центр выделенной области, и с помощью ползунка вы ее повернете – вид изменения будет отображаться в окне предпросмотра. Не переборщите: даже поворота на несколько градусов достаточно для заметной разницы; поворот на большой угол может исказить изображение до неузнаваемости.



7 Перспектива

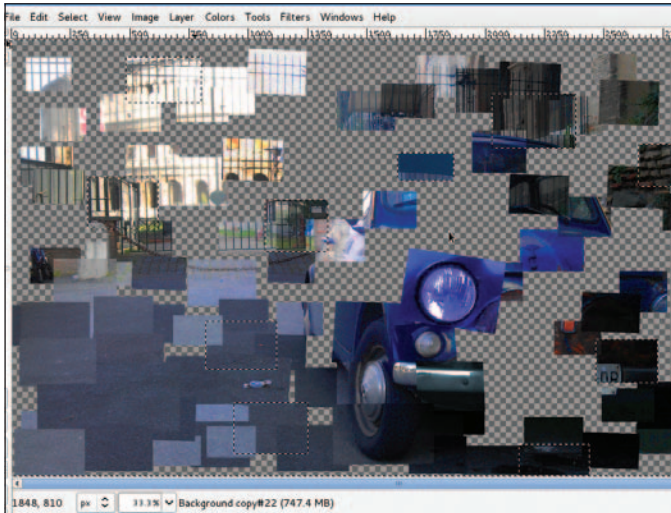
При работе с одним изображением сложнее всего имитировать то, что исходные изображения были сняты с разных точек. С помощью инструмента Perspective [Перспектива] можно сделать это довольно достоверно, но делайте это аккуратно, чтобы не возникло искажений. Выделите прямоугольную область, нажмите на инструмент Perspective и потяните за уголки. Лучше всего сдвигать только одну сторону – изображение не должно чересчур перекручиваться. При работе с этим инструментом изначально выбирайте область большего размера, чем это необходимо – в конце ее придется подрезать.



8 Обводка

В некоторых Хокни-подобных коллажах каждый фрагмент изображения обведен по контуру. Опыт показывает, что иногда это оправдано, но в большинстве случаев немного не к месту. На самом деле это работает, только когда все прямоугольники приблизительно одного размера. Так или иначе, добавить обводку довольно легко. Выберите прямоугольную область, затем выберите Selection > Border [Выделение > Обводка] и введите подходящее значение ширины обводки в пикселях, согласно изображению. Затем выберите Edit > Fill with BG Colour [Редактирование > Залить цветом фона], чтобы залить его белым цветом (или любым другим на ваш выбор).

» Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.



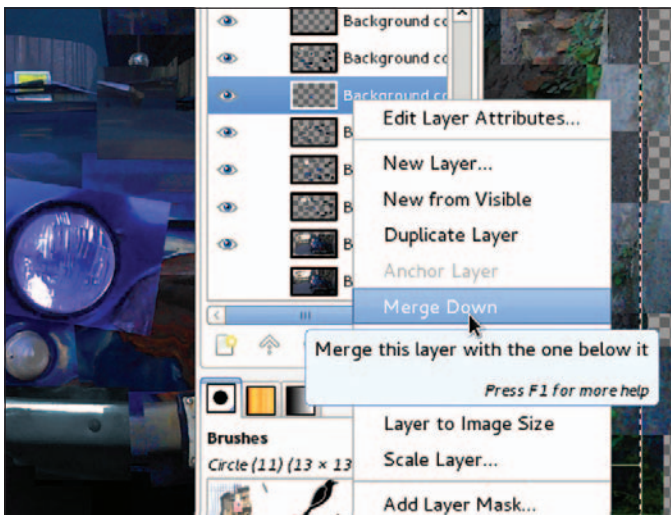
9 Схитрим!

Скоро вы поймете, что делать десятки слоев из мелких прямоугольников долго. Даже если разбираться с каждым из них довольно быстро, на всю работу с изображением все равно может уйти несколько часов. Поэтому нам нужна хитрость. Ничто не мешает нам использовать эти приемы на нескольких областях изображения одновременно (можно даже поворачивать их, если делать это аккуратно). Вместо выбора только одной области изображения выберите несколько. Удерживая клавишу Shift, измените размер нескольких прямоугольников. Выбираемые области должны иметь похожие параметры, которые вы меняете (например, цвет), и, конечно, они не должны быть слишком близко друг к другу, иначе эффект будет испорчен.



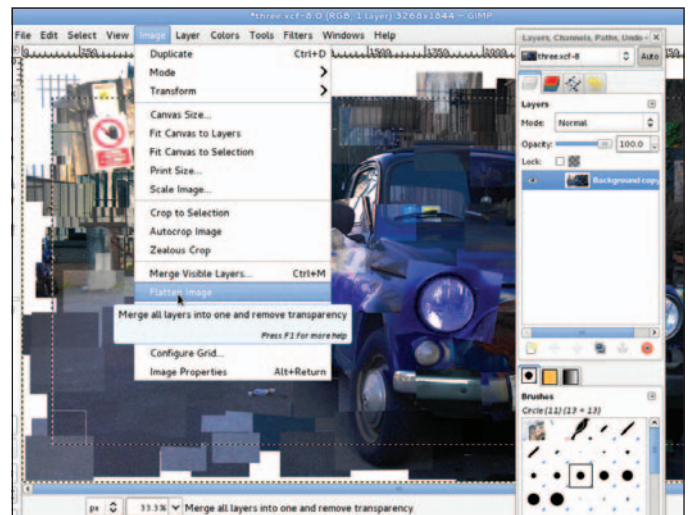
10 Опять схитрим!

Даже с предыдущим приемом, состоящим в обработке нескольких областей одновременно, обработка всего изображения может затянуться. Только подумайте о бедаогах, которые делают все это с настоящими фотографиями! Так или иначе, есть еще одна удобная хитрость: сохранить исходное фоновое изображение и заполнить им все пустые места. Самые проницательные из вас поймут, что это уничтожит иллюзию неправильных краев, но еще более сообразительные также поймут, что их можно просто отрезать. Инструментом Selection выберите внутреннюю область изображения, которой нужно заполнить пустые места – нажмите Ctrl+I для инвертирования и Ctrl+X для удаления краев.



11 Сохраняйте изображение по мере работы

Разумеется, стоит регулярно сохранять изображение, лучше в нескольких разных версиях. В зависимости от размера изображения, работа с несколькими слоями может потребовать довольно большого количества памяти. Например, одна из сохраненных версий нашего изображения лишь немного не дотягивала до 1 ГБ, а это, согласитесь, немало. GIMP и Linux очень стабильны, и система не «упадет» только из-за обработки больших объемов памяти, но ваша работа может заметно тормозиться. Одно из решений проблемы – объединять слои (Layer > Merge Down [Слой > Объединить]) для тех участков изображения, с которыми вы покончили.



12 Объединяем слои

Осталось только объединить слои. Перед этим у вас может возникнуть желание переместить некоторые слои. Принцип работы со слоями означает, что изображение будет своего рода «перевернутым», так как новые слои, создаваемые из фона, помещаются поверх предыдущих – и созданные вами слои будут наверху. Перемещайте их в окне Layer [Слой], пока не удовлетворитесь результатом. Закончив (сначала сохраните версию файла!), выберите Image > Flatten Image [Изображение > Разгладить изображение] для объединения всех слоев – а затем наносите финальные штрихи. **LXF**

Raspberry Pi.

В нашем втором учебнике по RPi Грэм Моррисон проясняет для вас основы управления пакетами, эмуляцию и сетевые хранилища данных.



Наш эксперт

Когда Грэм Моррисон не ломает голову над тем, что поставить в *Linux Format*, и не возится со старыми синтезаторами, он скорее всего ищет новые способы повысить продуктивность KDE 4.

Месяц назад мы открыли коробку с Raspberry Pi, запустили операционную систему и подключили периферийные устройства, а также изменили разделы на SD-карте, чтобы освободить место (если вы пропустили первую часть, обратитесь к PDF-версии на DVD). Теперь мы нырнем с этой точки глубже, погрузившись в то, что Linux может предложить Pi, от удаленного управления и установки пакетов до прослушивания музыки и использования его в качестве сервера файлов и контента для компьютеров сети. Pi можно превратить в очень дешевое и невероятно гибкое сетевое хранилище данных, потратив сумму, равную стоимости внешнего USB-диска.

Но для начала мы включим SSH. SSH – «безопасная оболочка [secure shell]», с помощью которой мы получим доступ к командной строке Pi через сеть. Для Pi это особенно важно, потому что место перед телевизором – не самое удачное для работы. С SSH можно работать из любой другой системы Linux, Windows или OS X, и хотя он был предусмотрен по умолчанию в дистрибутиве Debian, который мы установили, но не запущен. Для его запуска войдите в систему на своем Pi (по умолчанию используются имя пользователя «pi» и пароль «raspberrypi») и введите команду:

```
sudo service ssh start
```

Новичкам в Linux эта строка может показаться китайской грамотой, но если вдуматься, то понять ее несложно. Отдельные слова в командной строке обычно запускают одноименную утилиту или дают этой утилите дополнительные указания о том, что ей делать. В приведенной выше команде, **sudo** – утилита, позволяющая запускать другие команды с привилегиями администратора, **service** – утилита для запуска скрипта из каталога `/etc/init.d`, а **ssh**

start – дополнительная информация, необходимая для запуска **service**: сначала указывается сам сервис – **ssh**, затем действие, которое нужно выполнить – **start** [запустить]. Также возможны действия **stop** [остановить] или **restart** [перезапустить]. Массу дополнительной информации о каждой команде можно получить, набрав **man** и имя команды. Например, чтобы разузнать о команде **sudo**, наберите **man sudo**. Мы также хотим, чтобы SSH запускалась при загрузке Pi, и для этого нам нужна другая команда:

```
sudo insserv ssh
```

man insserv покажет, что это команда автоматического запуска скрипта при загрузке системы, а раз SSH настроен и запущен, можно даже не подключать к Pi клавиатуру, мышь и монитор. Пока Pi подключен к сети, с помощью SSH им можно управлять удаленно. Но для этого нужно знать IP-адрес устройства. Raspberry Pi сообщает свой IP-адрес в конце журнала загрузки, но его можно определить и командой **ifconfig**, посмотрев число справа от **inet addr** в разделе **eth0**. **ifconfig** – мощная команда: с ее помощью можно изменить множество сетевых настроек, а также посмотреть текущие. Устройство 'lo' в порту Ethernet – это виртуальное устройство обратной петли, удобное при доступе к сервисам, запущенным на Pi, с самого Pi.

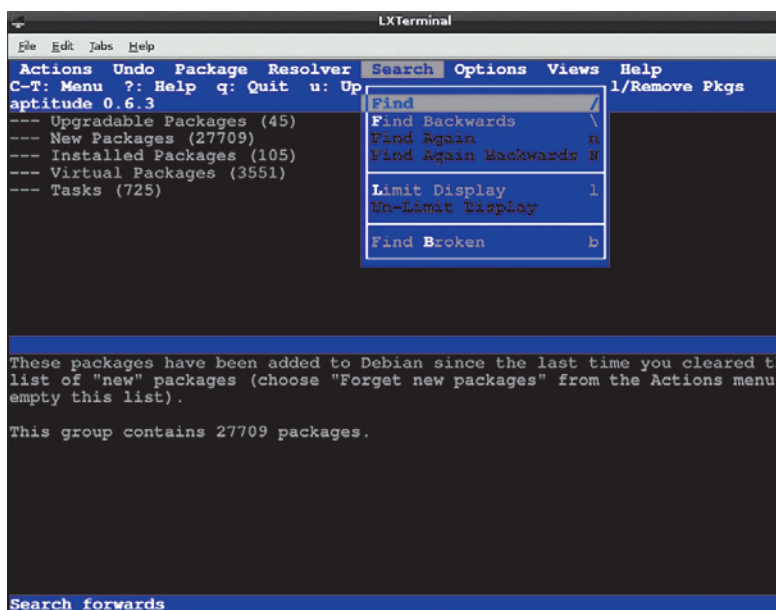
Теперь у нас есть IP-адрес Pi. С другого компьютера той же сети выполните команду **ssh pi@192.168.1.25** – только замените IP-адрес на свой. У вас спросят пароль, после чего вы окажетесь в точно такой же командной строке, как на самом компьютере, и сможете делать все то же самое через SSH-соединение. Добро пожаловать в мир удаленной работы!

Управление пакетами

Даже в системе, установленной по умолчанию, можно сделать многое. Но рано или поздно возникает желание установить дополнительное ПО. В Linux это делается с помощью так называемого менеджера пакетов, который по сути похож на магазин приложений. Вы выбираете нужные программы, и они загружаются автоматически. В Linux загружаются и все зависимости – это все библиотеки и приложения, необходимые для запуска программы, которую вы хотите установить. Так как операционная система нашего Raspberry Pi – Debian, а менеджеры пакетов могут зависеть от дистрибутива, мы будем пользоваться менеджером пакетов Debian. Для взаимодействия с ним используется утилита **apt** (advanced package tool – продвинутая утилита управления пакетами). Так как Debian лежит в основе Ubuntu, то в ней используется та же схема управления пакетами (но пакеты другие), поэтому вы скорее всего уже имели с ней дело, хотя, наверное, не из командной строки. Для установки пакета достаточно ввести

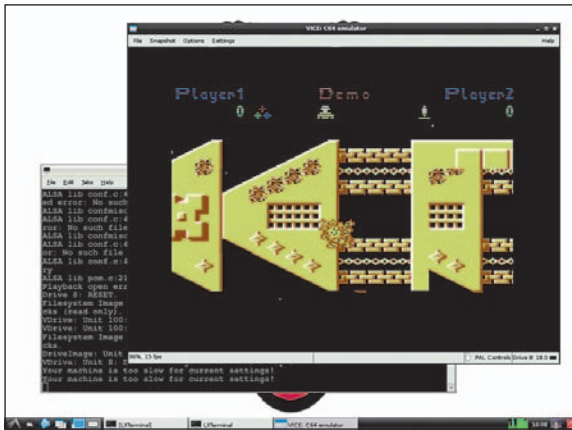
```
apt-get install packagename
```

Есть, однако, и гораздо более простая альтернатива: утилита под названием **aptitude**. Ею можно пользоваться в командной строке для поиска и установки пакетов (**aptitude search <имя пакета>**) и **aptitude install <имя пакета>**), но в ней есть и примитивный интерфейс командной строки. Для его появления наберите **aptitude**, и вы убедитесь, что работать в нем гораздо проще,



» В отсутствие графического менеджера пакетов, лучшей его альтернатива для Raspberry Pi – **aptitude** в командной строке.

Строим NAS



» Эмуляторы старинных компьютеров, вроде Commodore 64 и ZX Spectrum, работают на Pi вполне прилично.

чем в командной строке. После запуска утилиты обновляется кэш пакетов, и окно делится на три части. Навверху находится меню, под ним – список пакетов, по которому можно перемещаться с помощью клавиш управления курсором, а под списком – результаты ваших действий. Для поиска пакета перейдите в верхнее меню. Если подключена мышь, должен сгодиться и щелчок, даже при работе через SSH, но можно переключить фокус ввода в меню и вручную, нажав **Ctrl+T**. Список найденных пакетов обновляется в реальном времени, и по нему можно перемещаться клавишами управления курсором или клавишами **PageUp** и **PageDown**.

Эти пакеты загружаются из так называемого «репозитория». Репозиторий по умолчанию в Debian довольно строг: в нем нет ничего, кроме компонентов операционной системы Debian. Но это ограничение можно обойти, добавив новые репозитории, что мы сейчас и сделаем. Список репозитория находится в текстовом файле `/etc/apt/sources.list`, и его нужно открыть с правами системного администратора. Для этого в командной строке наберите `sudo nano /etc/apt/sources.list`. Когда файл откроется, добавьте слово **contrib** в конец первой строки (оканчивающейся на **squeeze main**). Архив **contrib** содержит свободные пакеты, которые зависят от пакетов в несвободном архиве, репозитории, который уже должен быть в файле **sources.list**.

Добавив новый репозиторий, обновите список пакетов: просто введите команду

```
sudo apt-get update
```

Теперь вам будут доступны для установки множество пакетов, ранее отсутствовавших. Например, мы установили *Vice*, эмулятор Commodore 64, тем самым связав Pi с его предками – домашними компьютерами из 1980-х. Чтобы сделать это на своем Pi, набери-

те `apt-get install vice` (раньше эта команда бы не работала). По завершении установки скачайте прошивки (ROMs), необходимые для нормальной работы эмулятора. Они входят в версию *Vice* для Windows, но их сомнительная легальность не позволяет включить их в пакеты Debian. Если вы имеете право на эти действия, распакуйте образы из версии Windows и поместите их в нужный каталог следующими командами:

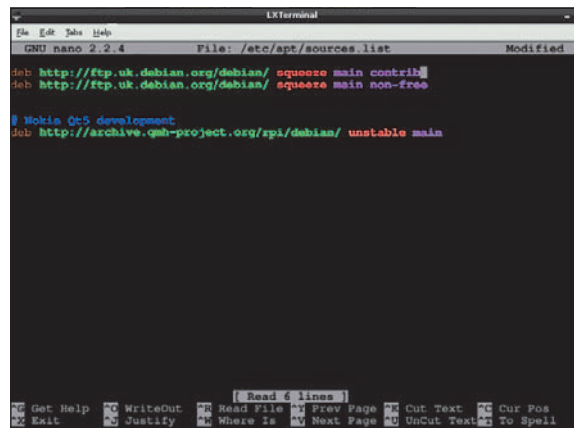
```
tar xvf vice-1.5-roms.tar.gz
sudo cp -a vice-1.5-roms/data/* /usr/lib/vice/
```

Теперь можно запустить эмулятор, набрав `x64`. Эмулятором можно пользоваться только при подключенном мониторе, а не через SSH, но благодаря простому интерфейсу *Vice* достаточно щелкнуть в меню файлов, чтобы открыть нужный образ и запустить игру. Рекомендуем добавить параметр **auto-start**: благодаря ему вам не придется набирать **LOAD "*"8,1** и **RUN** для запуска игры по умолчанию на любом образе диска. Нам удалось достичь хорошей производительности эмулятора, и в большинстве игр мы поиграли на почти стопроцентной скорости. Даже когда частота кадров падала ниже 10 %, игры шли все еще терпимо.

Добьемся звучания

Причина, по которой не работает звук, по крайней мере в ранней версии Debian, которой мы пользуемся – не загружен драйвер для поддержки звуковых устройств. Драйверы Linux отличаются от драйверов Windows точно так же, как пакеты отличаются от загружаемых исполняемых файлов, и драйверы обычно являются частью ядра, а не отдельными файлами, которые нужно загружать и устанавливать. Поэтому так много устройств работает без ручного вмешательства – ядро обнаруживает их и автоматически загружает нужный драйвер. Драйверы, динамически подгружаемые таким образом, называются «модулями», и набрав в командой

«Ядро обнаруживает и автоматически загружает драйверы.»



» Для добавления пакетов сторонних разработчиков в Pi достаточно изменить файл настройки и обновить кэш пакетов.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

строке **sudo lsmod**, вы увидите список загруженных модулей. Однако, поскольку дистрибутивы для Raspberry Pi точно «знают», с какими устройствами работают, им не нужны большинство модульных драйверов. Кроме звукового. В нашей системе загружен всего один модуль – fuse; он позволяет динамически монтировать удаленные файловые системы. Звуковой модуль должен входить в состав вашего дистрибутива, и для его загрузки нужно выполнить команду

```
sudo modprobe snd_bcm2835
```

Снова набрав **lsmod**, вы увидите, что этот модуль был загружен вместе со всеми зависимостями. Эти модули необходимы для работы звукового драйвера. Теперь можно проверить, что звук работает, вставив наушники или колонки в аудиовход Pi и запустив приложение Music Player из меню Sound & Video [Звук и видео]. В установке Debian есть даже тестовый фрагмент музыки для проверки. Теперь звук работает, и нужно снова включить его в эмуляторе *Vice*. Так как это приблизит нагрузку процессора к предельной, мы также рекомендуем снизить частоту кадров, увеличить размеры буфера и фрагмента, а также отключить эмуляцию фильтра в меню настройки *Vice*.

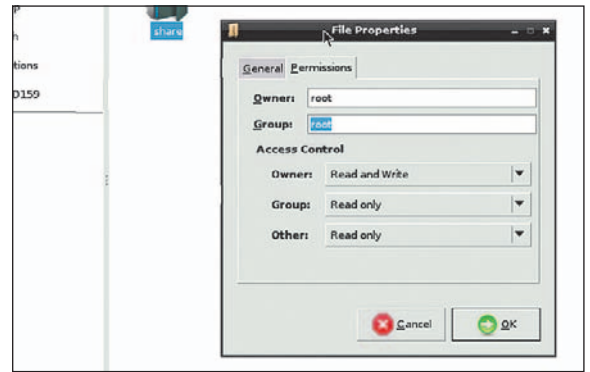
Звуковой модуль в Raspberry Pi мы сочли довольно шатким: аудиоприложения иногда зависали. Возможно, поэтому он и не включен по умолчанию; но если вы хотите, чтобы модуль загружался при загрузке системы, просто добавьте в текстовом редакторе имя модуля (**snd_bcm2835**) в **/etc/modules**. Для изменения этого файла вам потребуются права системного администратора, но после перезапуска Pi модуль будет загружен автоматически. Чтобы модуль больше не загружался, удалите строку из этого файла, а чтобы выгрузить модуль из системы, скачайте **rmmmod snd_bcm2835**.

Разделение файлов

Напоследок мы превратим Raspberry Pi в сетевое хранилище данных (NAS). Это позволит подключить к нему внешний USB-диск и получать общий доступ к этим файлам по сети. Добавьте сюда возможность воспроизведения музыки, и Pi станет центром медиа-хаба – с общим доступом к файлам и воспроизведением музыки через удаленное соединение. В центре этого функционала – программа под названием *Samba*, стабильная и широко используемая реализация сетевого протокола Microsoft.

Так как этот протокол используется для удаленного доступа к файлам в Windows, OS X и Linux, то с ним лучше всего организо-

«Звуковой модуль в Raspberry Pi мы сочли довольно шатким.»



Желая сделать файлы и каталоги доступными другим пользователям и в других сетях, измените их свойства.

вать файловый сервер, доступ к которому возможен из любой операционной системы. Для установки *Samba* достаточно одной команды:

```
sudo apt-get install samba
```

По завершении установки будут запущены два фоновых процесса. Такие процессы, как и сервер SSH, который мы запустили до этого, обычно называются «демонами», и они будут автоматически обрабатывать любые запросы, проходящие по сети. Те-

перь нужно определить, где в файловой системе мы будем хранить файлы. Так как мы будем пользоваться внешним диском, подключенным через USB, нужно создать соответствующую точку в фай-

ловой системе и сделать ее постоянной. После подключения USB-диска наберите **dmesg**. Эта команда отображает вывод системных журналов, и в конце вы должны увидеть сообщения, относящиеся к подключению USB-диска. Поищите нечто вроде **sda: sda1**; будет либо именно так, либо это может быть **sdb**. В любом случае, это ваше устройство в виртуальной файловой системе, используемой для управления устройствами в Linux. В **/dev/** находится огромный список доступных устройств.

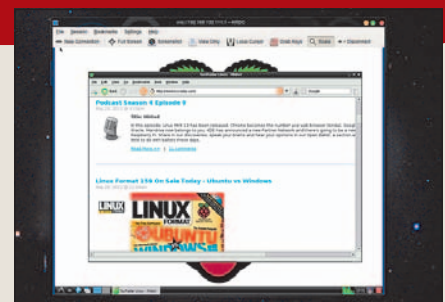
Чтобы добавить устройство в систему, сначала создайте новую папку, которая будет выступать в качестве корневой для этого устройства. Мы создали подкаталог **share** в существующем каталоге **/mnt**, командой **mkdir /mnt/share**. Чтобы этот каталог монтировался при загрузке системы, добавьте следующую строку в **/etc/fstab**:

```
/dev/sda1 /mnt/share vfat defaults 0 0
```

Удаленные рабочие столы

Работать на Raspberry Pi, усевшись с клавиатурой и мышью перед телевизором, неудобно, поэтому мы установили сервер SSH для удаленного управления. Однако можно пойти дальше и получить полноценный доступ к рабочему столу через сеть. Это значит, что вы сможете настраивать свой Pi с дивана или даже из офиса. Идея состоит в установке сервера VNC. VNC – общий протокол доступа к рабочему столу, который «упаковывает» ваш рабочий стол в серию изображений формата JPEG и воссоздает его на клиенте, обрабатывая ввод с клавиатуры

и мыши. Для установки сервера введите команду **apt-get install tightvncserver**, а для запуска нового сеанса сервера – команду **vncserver:1**. У вас запросят пароль, который нужен клиенту для подключения. Советуем также запускать сервер на Pi без X: это экономит ценную память. На удаленном компьютере нужно установить клиент VNC – впрочем, и в Gnome, и в KDE клиенты встроены. Введите IP-адрес своего Pi с портом 5901, затем пароль, и через несколько секунд вы получите полноценный Pi на своем компьютере/планшете/телефоне.

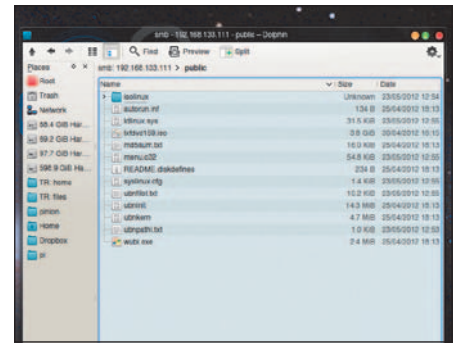
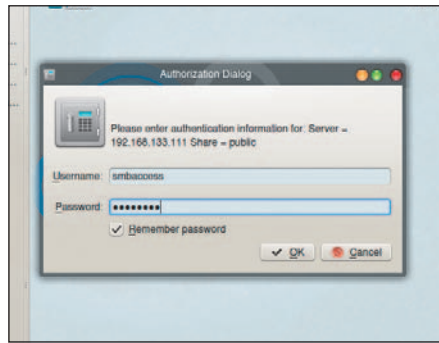
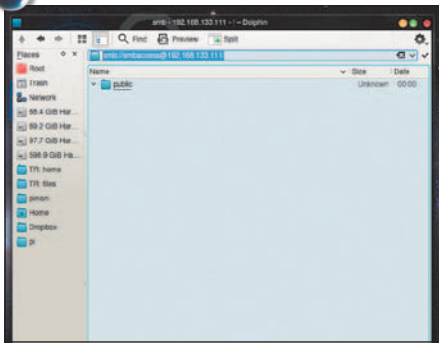


Наш любимый клиент VNC – KRDC: он автоматически масштабирует рабочий стол.

» Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.



Шаг за шагом: Доступ к файлам на Pi



1 Запустите файловый менеджер

В файловом менеджере или на рабочем столе откройте Pi, указав следующий адрес: `smb://ip_address`.

2 Введите пароль

Введите имя пользователя и пароль, созданные в предыдущей статье.

3 Работайте с файлами

Теперь вы можете читать и записывать файлы в разделяемый каталог на Raspberry Pi.

Эта строка состоит из источника (это наше USB-устройство с именем устройства, взятым из вывода `dmesg`), места назначения (это только что созданный каталог), типа файловой системы, ключевого слова 'defaults' и двух нулей. Сомнения вызывает только тип файловой системы, так как диск, вероятно, будет отформатирован как в Microsoft (тип `vfat`, на языке файловых систем).

Гораздо лучше реформатировать устройство точно так же, как мы реформатировали карты в прошлой статье, и воспользоваться «родной» файловой системой Linux, такой как `ext3`. Если вы это сделаете, укажите соответствующий тип файловой системы. Сохраните изменения в `fstab` и проверьте, что все работает, набрав `sudo mount /mnt/share`. Если все прошло успешно, то для просмотра содержимого USB-устройства достаточно открыть в браузере или в командной строке каталог `/mnt/share`.

Чтение и запись

Если что-то не работает, посмотрите на новые сообщения в `dmesg`, чтобы понять, в чем проблема. В противном случае USB-диск должен автоматически монтироваться при каждом запуске Pi. Теперь мы хотим сделать так, чтобы каждый мог читать и записывать файлы в общем каталоге. Это можно сделать из командной строки двумя следующими командами:

```
sudo chown -R root:users /mnt/share
sudo chmod -R ug+rwX /mnt/share
```

Первая изменяет владельца всех файлов на диске (аргумент `-R` означает, что изменения будут применяться рекурсивно). Фрагмент `root:users` делает владельцем пользователя `root` (администратора системы), но также и группу `users`. В терминах Linux это означает, что у любого пользователя из группы `users` тоже будет доступ к этим файлам. Наберите в командной строке `groups`, и вы увидите список групп, членом которых является текущий пользователь. Если вы работаете как пользователь `pi`, в списке групп будет и `users`. Вторая команда делает файлы доступными для чтения (`r`), записи (`w`) и исполнения (`x`) пользователю и группе, связанным с файлами. Теперь мы создадим пользователя, от имени которого будем работать с общим каталогом. Для этого сначала создадим новую учетную запись в системе, а затем свяжем ее с `Samba`. Имя пользователя и пароль, создаваемые на данном этапе, вы будете вводить каждый раз для доступа к каталогу `Samba` с удаленного компьютера. Сделать это можно двумя следующими командами:

```
sudo useradd smbaccess -m -G users
```

```
sudo passwd smbaccess
```

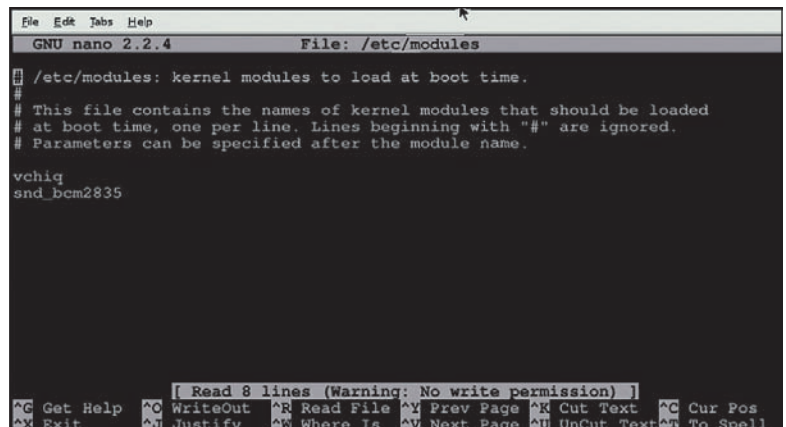
```
sudo smbpasswd -a smbaccess
```

Сначала мы добавили пользователя `smbaccess` и сделали его членом группы `users`. Затем задали пароль для этой учетной записи перед добавлением его в `Samba`. Затем мы просим `Samba` организовать общий доступ к указанной точке монтирования по сети, но для этого нужно добавить следующие параметры в конец `/etc/samba/smb.conf`.

```
[public]
comment = Public
path = /mnt/share
valid users = @users
force group = users
create mask = 0660
directory mask = 0771
read only = no
```

Последний этап – перезапустить `Samba`, и для этого мы вернемся к первой команде этой статьи: `sudo service samba restart`. Теперь у нас есть доступ к файлам с любого компьютера в сети. Для этого укажите протокол `smb` и IP-адрес Raspberry Pi. Например, в KDE можно набрать `smb://smbuser@192.168.133.111` в командной строке или в файловом менеджере.

Вы увидите свои файлы и каталоги и сможете читать и записывать их с любого компьютера в сети. **LXF**



Если вы хотите, чтобы модули загружались при загрузке системы, добавьте их в `/etc/modules`.



Добавляем НОВЫЙ

Неважно, собрались вы добавить 2-терабайтный колосс или крошечный SSD: Нейл Ботвик поможет сделать все правильно.



Наш эксперт

У Нейла Ботвика по компьютеру в каждой комнате, но по соображениям безопасности он ни за что не скажет, где находится центральный сервер.

Когда-то добавить новый жесткий диск было просто: требовалось лишь подключить его к компьютеру и создать разделы любимым инструментом командной строки (*fdisk*, *cfdisk*) или графическим (*Gparted*). Если вы меняли системный жесткий диск, можно было загрузиться с дистрибутивного CD, и установщик системы выполнял разметку за вас. Теперь все стало несколько сложнее: с виду разные накопители, как выясняется, демонстрируют сходные проблемы.

Большие жесткие диски, порядка 2 ТБ, стали стоить вполне разумно, и вы можете захотеть просто воткнуть один такой в компьютер и начать им пользоваться – а в результате окажется, что производительность ниже, чем у вашего старого диска. Всею виной схема организации данных. Многие годы информация хранилась блоками по 512 байт (этот блок не связан с размером блока файловой системы). Диски становились больше, число блоков росло, и при 2 ТБ превысило максимальное значение, которое можно сохранить в 32-битном целом.

Тогда производители дисков перешли на 4-КБ блоки, в восемь раз больше; таким образом, проблема отложена до времен, когда один диск будет вмещать 16 ТБ. Это также позволило эффективнее хранить большие файлы, особенно с учетом того, что многие файловые системы, включая ext2/3/4, также используют блоки по 4 КБ. Это все звучит неплохо до тех пор, пока вы не осознаете, что жесткие диски вынуждены обеспечивать обратную совместимость с некой популярной ОС, притворяясь, что они по-прежнему оперируют блоками по 512 байт. Что в этом плохого? А то, что если вы начнете файловую систему с блока, номер которого не кратен 8 (скажем, 63, что издревле

было значением по умолчанию в Linux), каждый 4-КБ блок файловой системы будет храниться в двух физических блоках на диске, и каждое его чтение или запись превратится фактически в две операции ввода-вывода, что крайне негативно скажется на производительности (по ряду причин, увеличение объема работы в два раза вызывает более чем двукратное падение скорости).

Твердотельные диски

Выравнивание блоков еще более важно на другом конце шкалы емкостей – в сегменте SSD-дисков. Они используют флэш-память, а не магнитный носитель, что дает ряд преимуществ, особенно для мобильных устройств (почему большинство твердотельных дисков имеют форм-фактор 2,5"). Отсутствие движущихся частей делает такие диски более тихими и ударопрочными, и у них более экономное энергопотребление. Но важнее то, что SSD-диски куда быстрее своих магнитных аналогов; фактически, они могут превзойти пропускную способность интерфейса SATA. Поэтому для достижения максимальной производительности наиболее продвинутые (и дорогие) SSD-диски подклю-

чаются через карты PCIe; но в нашей статье мы будем говорить о SATA-вариантах. Несмотря на использование флэш-памяти, SSD-диски имеют мало общего с USB-флешками. SSD-диски используют технологию равномерного износа [wear-levelling], чтобы предотвратить быстрый выход из строя из-за постоянной записи на одно и то же место (например, в журнал файловой системы). Это налагает ограничения на разметку. Хотя размер блока может оставаться равным 512 байт, как и у магнитного диска

«Отсутствие движущихся частей делает SSD-диски ударопрочными.»

чаются через карты PCIe; но в нашей статье мы будем говорить о SATA-вариантах. Несмотря на использование флэш-памяти, SSD-диски имеют мало общего с USB-флешками. SSD-диски используют технологию равномерного износа [wear-levelling], чтобы предотвратить быстрый выход из строя из-за постоянной записи на одно и то же место (например, в журнал файловой системы). Это налагает ограничения на разметку. Хотя размер блока может оставаться равным 512 байт, как и у магнитного диска

CHS или LBA?

На первых жестких дисках положение блока определялось номером «блина» (или головки), дорожки (цилиндра) на «блине» и сектора в цилиндре. Такая схема адресации известна как CHS (Cylinder, Head, Sector). На смену ей пришла схема LBA (Logical Block Addressing, Адресация логических блоков), в рамках которой номера блоков на диске составляют простую последовательность, начинающуюся с нуля и продолжающуюся до физической емкости диска.

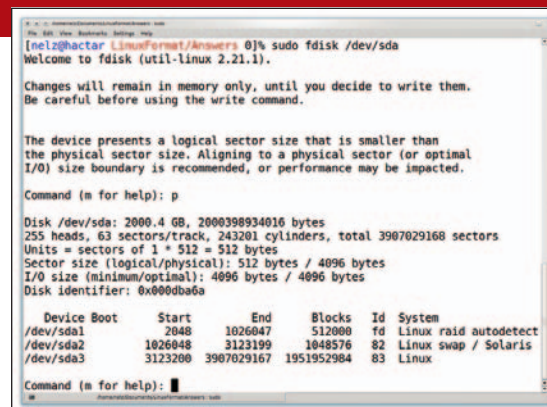
Однако традиционные инструменты разметки, вроде *fdisk*, продолжают использовать CHS-адресацию. Вот, например, вывод *fdisk -l* на 2-ТБ диске:

```
% fdisk -l /dev/sde
Disk /dev/sde: 2000.4 GB,
```

```
2000398934016 bytes
255 heads, 63 sectors/track, 243201
cylinders, total 3907029168 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes /512
bytes
I/O size (minimum/optimal): 512 bytes /512
bytes
Disk identifier: 0x0008f6db

Device Boot Start End Blocks Id System
Blocks Id System
/dev/sde1 2048 3907028991 1953513472 83 Linux
```

Здесь просматривается как CHS-адресация, так и то, что 2-ТБ диск использует 512-байтные блоки, хотя на сайте изготовителя сказано, что размер блока – 4 КБ.



Современные выпуски *fdisk*, а также все версии *gdisk* автоматически выравнивают границы устройств для максимальной производительности, даже если устройство скрывает истинный размер блока.

ВИНЧЕСТЕР

аналогичной емкости, эти блоки группируются в так называемые блоки стирания [erase block]. Таким образом, затирание и перезапись памяти происходит более крупными кусками, и эффективность работы повышается.

Размер блока стирания может варьироваться (обычно между 128 КБ и 512 КБ). Любая схема разметки и файловая система, не выровненная по границам этих значений, будет испытывать те же проблемы с производительностью, что и на больших жестких дисках – если не хуже.

Выравнивание блоков

Итак, мы столкнулись с одной и той же проблемой и одним и тем же решением для двух дисков совершенно разных типов. Как обеспечить, чтобы разделы создавались в «правильных» местах? Можно подумать, что достаточно сделать размеры всех разделов кратными 4 КБ, но эта идея не сработает, поскольку первый раздел начинается с блока под номером 63 – это многолетнее соглашение – т.е. на один блок раньше, чем вы хотели бы. Результат – проблемы с производительностью.

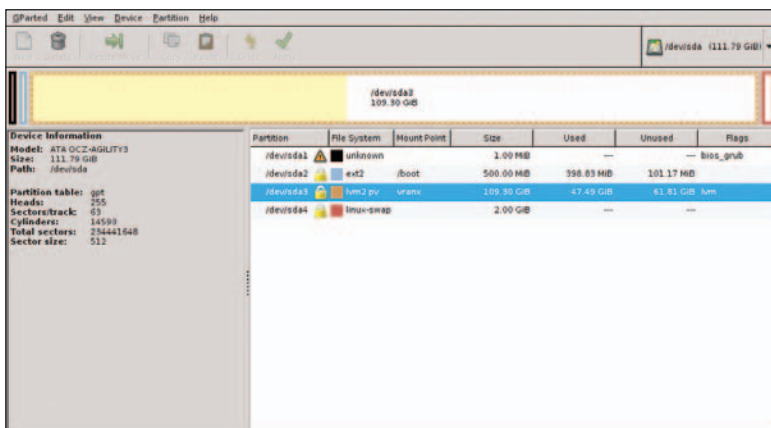
Если вы размечаете жесткий диск свежей версией *fdisk*, по умолчанию все разделы будут кратными 2048 секторам, независимо от их размера; при 512-байтных блоках это будет обозначать выравнивание по границе 1 МБ. Таким образом, будут удовлетворены требования и SSD, и больших жестких дисков. Если вы и впредь будете следить за тем, чтобы все разделы были выровнены таким образом, проблем не будет. Наилучший способ достичь этого – задавать размеры раздела в МБ (или ГБ), а не секторах – тогда *fdisk* сделает всю работу за вас. Если у вас старая версия *fdisk*, которая пытается начать с 63-го сектора, и вы по какой-то причине не можете обновить ее, начните раздел с блока 2048 и выбирайте размеры разделов кратными 1 МБ; это самое простое решение (более сложный вариант – задавать разные CHS в командной строке, указывая ключи **-C**, **-h** и **-S** при старте *fdisk*. Это тоже работает, но малейшая ошибка приведет к некорректному выравниванию). В любом случае, не связывайтесь с *cdisk*: ранее достойный, сейчас он не может обеспечить должную степень контроля над происходящим.

BIOS или GPT?

Есть и еще один способ: отказаться от таблицы разделов BIOS тридцатилетней давности. Помимо навязывания CHS-адресации, она создает ограничение в четыре первичных раздела. Его, конечно, можно обойти при помощи логических разделов, но это работает, только если у вас еще нет четырех разделов, а на многих ноутбуках Windows и разделы восстановления занимают их все.

К тому же логические разделы остаются хаком – в таблице по-прежнему имеется четыре слота, а логические разделы ссылаются друг на друга. Это усложняет создание резервных копий таблицы разделов, к тому же повреждение в любом блоке, содержащем ссылку, сделает все последующие разделы нечитаемыми.

GPT (GUID Partition Table, таблица разделов на основе GUID-идентификаторов) приходит на замену таблице BIOS и может использоваться практически любым современным оборудованием. Инструменты для работы с GPT-дисками обычно включены в пакет *gptfdisk*. Сюда входят *gdisk* и *cgdisk*, заменяющие *fdisk* и *cdisk*,



соответственно. *Parted* (и его графические оболочки) уже поддерживает GPT – никакая новая программа не требуется. Программы работают в основном привычным образом, но автоматически выравнивают все по 2048 блокам, что позволяет избежать описанных выше проблем.

Наилучшей рекомендацией для сохранения обратной совместимости с существующими загрузчиками является создание специального загрузочного раздела в начале диска. Запустите *gdisk* от имени root, укажите имя своего диска и нажмите **n** для создания раздела. Значения номера раздела и стартового сектора можно оставить как по умолчанию (для чистого диска это 1 и 2048), а для конечного сектора введите **+1 M** (что означает 1 МБ от начала раздела). На вопрос о типе раздела введите **EF02**, чтобы создать загрузочный раздел BIOS. Разметьте остальные разделы стандартным образом.

Процедура полностью аналогична *fdisk*, разве что вас не спрашивают, хотите вы создать логический или основной раздел. Другое отличие – тип раздела теперь указывается двухбайтовым кодом: так, для Linux-раздела это 8300, а не 83. Обратите внимание, что загрузочный раздел BIOS – не то же самое, что раздел **/boot** в Linux: здесь не содержится никаких файлов операционной системы. Это просто первый раздел жесткого диска, а «полезные» разделы начинаются с **/dev/sda2**.

Альтернативные инструменты

В примере выше мы использовали *gdisk*; *cgdisk* аналогичен, но использует текстовый *ncurses*-интерфейс в противовес командной строке *gdisk*. Есть также *parted*: он очень мощный и имеет параметр командной строки, включающий оптимальное выравнивание. Однако его интерфейс даже менее интуитивен, чем у *fdisk*. Есть и графические оболочки – например, популярный *Gparted*, а также инсталляторы большинства дистрибутивов; однако *Gparted* не предлагает способа управления выравниванием, так что его можно использовать на больших дисках и SSD, только если вы создаете GPT-разделы.

Если при установке на новый жесткий диск инсталлятор вашего дистрибутива предлагает использовать GPT – соглашайтесь; в противном случае вы можете решить предварительно разметить диск вручную. **LXF**

► Если вам никак не расстаться с мышью, *Gparted* послужит вам не хуже *parted*, хотя гибкостью придется пожертвовать.

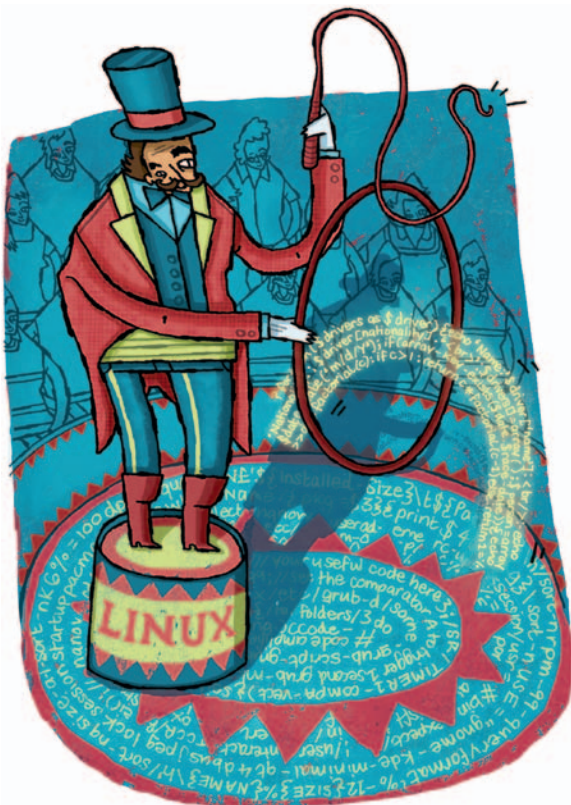
Shelr. Покажите

Майк Сондерс рассказывает о классном сайте и программе, которая поможет вам поделиться приемами работы в командной строке со всем миром.



Наш эксперт

Майк Сондерс сжился с командной строкой, и когда его вынуждают работать в этом новомодном «графическом» интерфейсе, его бросает в дрожь.



С виду командная строка дико скучна. Конечно, ее можно приукрасить, придав ей цвета и разнообразия шрифты, но почти все картинки из Bash выглядят одинаково (вы представляете, как страдает Эффи, наш дизайнер, когда мы что ни месяц даем ему стопку экранных снимков командной строки и просим расположить их красиво? Буйные испанские ругательства неминуемы).

Но решение есть, и это *Shelr*. Эта программа поможет записать ваши действия в командной строке, загрузить их в Интернет и поделиться ссылкой с кем угодно. Зачем это нужно? Ну, скажем, вы хотите показать кому-то, как переделать обычную систему Debian в ультра-защищенный web-сервер. Можно записать несколько команд и выложить их список в Интернете, но это ведь не очень вдохновляет, правда?

Shelr позволяет создать «псевдофильм», иллюстрирующий шаги, необходимые для запуска web-сервера. Гораздо интереснее видеть процесс в действии, чем читать листинг команд. К тому же благодаря такой схеме можно остановить воспроизведение, замедлить или ускорить его и скопировать текст с экрана. Это просто потрясающе, и мы считаем, что *Shelr* революционизирует онлайн-руководства. Рассмотрим ее подробнее...

Shelr написана на Ruby, а ее сайт – <http://shelr.tv>; щелкнув по ссылке “What’s up?” вверху, вы найдете установочные паке-

ты для различных дистрибутивов. Если вы пользуетесь Ubuntu 12.04, возьмите PPA для своего дистрибутива; есть также сборка *PKGBUILD* для Arch Linux и *ebuilds* для Gentoo. Если для вашего дистрибутива ничего найти не удастся, попробуйте через менеджер пакетов установить пакет *rubygems* (уж он-то должен быть в репозиториях) и выполнить следующую команду от имени суперпользователя-root:

```
gem install shelr
```

Если вы пользуетесь более ранней версией Ubuntu или Debian 6.0 или меньше, выполните следующую команду перед запуском программы:

```
export PATH=/var/lib/gems/1.8/bin:$PATH
```

Добавьте ее в конец файла *.bashrc* (в домашнем каталоге), и ее не придется набирать вручную перед каждым запуском *Shelr*.

Запись сеанса

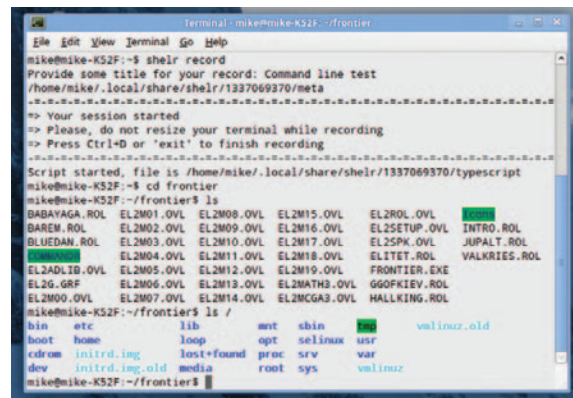
Итак, программа установлена – посмотрим ее в действии. Введите следующую команду для запуска сеанса записи *Shelr*:

```
shelr record
```

Прежде всего надо ввести название ролика. Это кажется слегка излишним, но если у вас есть коллекция роликов, то так их проще идентифицировать. Поэтому введите название, отражающее смысл ваших действий, например, «Как скомпилировать Apache». Нажмите Enter, и в окне терминала появится сообщение, говорящее о том, что сеанс записи начался и что не нужно менять размер окна терминала. Почему? Потому что это вызовет большие проблемы при трансляции записей в Сеть. В выводе наиболее важна следующая строка:

```
Script started, file is /home/mike/.local/share/shelr/1336992918/typescript
```

Расположение и номер в вашей системе будут отличаться (разве что вас тоже зовут Майк и имеет место редчайшее совпадение). Номер здесь представляет собой уникальный идентификатор ролика, которым можно воспользоваться, чтобы воспроизвести его позже.



» Типичный сеанс *Shelr* в действии в командной строке.

ЗНАНИЕ ОБОЛОЧКИ

Итак, за работу: введите несколько команд, просмотрите несколько текстовых файлов, очистите экран командой **clear** и т.д. Закончив, наберите **exit** (или нажмите Ctrl+d), и вы увидите сообщение “session finished [сеанс окончен]”. Все ваши действия с учетом скорости набора текста и остановок уже записаны *Shelr* для последующего воспроизведения.

Повторить последний ролик можно командой

```
shelr play last
```

Вы должны увидеть точный повтор своих действий. А если роликов несколько, наберите **shelr play**, и вы увидите следующий список:

```
Missing id for shellcast
```

```
1336992666 : Disk space
```

```
1336992918 : How to compile Apache
```

```
1336993272 : Log file check
```

Здесь мы и осознаем всю пользу заголовков. Для воспроизведения конкретного ролика воспользуйтесь его идентификатором, например, **shelr play 1336993272**. Учтите, что с программами типа текстовых редакторов на основе *ncurses*, которые проделывают с экраном нестандартные вещи, воспроизведение местами может выглядеть немного странно. Но для нашего очередного раздела это не проблема...

Делимся роликом со всем миром

Лучшее свойство *Shelr* – вышеупомянутая возможность публикации роликов в Интернете. Она реализована действительно хорошо – не нужно возиться с формами, загружать файлы и т.д. Вместо этого можно публиковать ролики с командной строки. Например, допустим, вы создали ролик с идентификатором 1337018264. Для его публикации скомандуйте:

```
shelr push 1337018264
```

Вас известят о типе лицензии на ваш ролик – это лицензия Creative Commons Attribution-ShareAlike, и она означает, что другим людям позволено менять вашу работу и делиться ею, но при этом они обязаны сослаться на ваше участие; кроме того, им нельзя сменить лицензию на совершенно другую. Но это вас вряд ли обеспокоит, если только вы не создали трехчасовой фильм с анимированными картинками из символов ASCII.

Затем вас попросят ввести описание ролика. Пишите здесь что хотите, но выбирайте самые точные ключевые слова, чтобы облегчить другим поиск этого ролика на сайте. Для той же цели вас попросят выбрать теги. Наконец, вам предложат вставить ключ API или просто нажать Enter для анонимной публикации – выбирайте второй вариант, так как к первому варианту мы вскоре вернемся.

После небольшой паузы на выгрузку данных вы увидите две строки:

```
=> Record published!
```

```
=> http://shelr.tv/records/4fb148fe96608047be000106
```

Вставьте ссылку в браузер. Вуаля, ваши победы в командной строке теперь доступны всему миру! В нижней части проигрывателя вы увидите регулятор скорости воспроизведения. По умолчанию видео воспроизводится со скоростью, в 2,5 раза большей скорости записи. Это прекрасно, если вас интересует только вывод, но если вы хотите увидеть ролик в «реальном времени», перетяните ползунок на скорость 1x.

Затем нажмите зеленую кнопку воспроизведения в левой нижней части проигрывателя. В нем особенно мило то, что это не встроенный апплет на Flash, который показывает последовательность картинок. Это сложный JavaScript, эмулирующий окно терминала, так что можно нажать на паузу и скопировать или вставить текст в окно терминала. Под проигрывателем вы увидите окошко Embed [Встроить], где находится HTML-код для встраивания видео на ваш сайт.

Другая приятная возможность – проигрывание сеансов в самом терминале. Нажмите на шестеренку в проигрывателе, »

Скорая помощь

Хотите избавиться от ролика? Узнайте его идентификатор и найдите соответствующий подкаталог **.local/share/shelr/** в вашем домашнем каталоге. Удалите этот подкаталог (например, командой **rm -rf**). Теперь по команде **shelr play** ролик больше не будет появляться в выводе.

Приоткрывая завесу

В каталоге **.local/share/shelr/** в вашем домашнем каталоге вы найдете файлы данных для роликов. Они находятся в подкаталогах с числовыми идентификаторами, поэтому просто зайдите в один из них, и вы увидите три файла: **meta**, **timing** и **typescript**. В первом хранятся параметры

ролика – заголовок, размер окна терминала, используемая оболочка (например, Bash) и др. Эта информация используется при воспроизведении, а также при публикации ролика.

В файле **timing** хранятся задержки между нажатиями клавиш – это важно для макси-

мально точного воспроизведения сеанса. Файл **typescript** содержит вывод терминала.

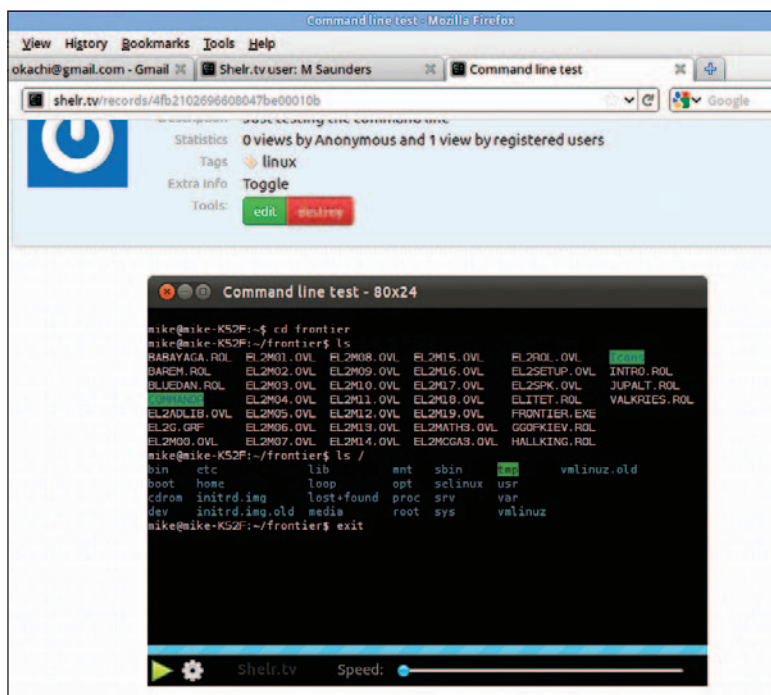
В этих файлах стоит покопаться, чтобы понять, как работает *Shelr*. Однако редактировать их вручную рискованно – не стоит выводить из строя серверы *Shelr* загрузкой бессмысленных данных!

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

и откроется окошко с командой, которую можно скопировать и вставить в окно терминала (она очень похожа на URL записи, но с `.json` на конце). Весь сеанс прокрутится в вашем терминале, но не стоит пугаться, заведя разрушительные команды вроде `rm -rf *` – это всего лишь текст, и на файлы он не влияет.

Продвинутые функции

У *Shelr* есть ряд продвинутых функций, достойных исследования. Пока что все сделанные нами ролики публиковались под именем Anonymouse с аватаркой внушительного ниндзя. Но можно и персонализировать свои загрузки, создав учетную запись пользователя. Зайдите на сайт и нажмите на кнопку Login [Войти] справа вверху. Вам предложат несколько способов аутентификации – через GitHub, Twitter, Google и OpenID. Скорее всего, одна из таких учетных записей у вас уже есть, поэтому выберите подходящую. Мы воспользовались Google, который сказал нам, что *Shelr* хочет знать адрес нашей почты – так что беспокоиться не о чем.



► Наш сеанс *Shelr* воспроизводится в web-интерфейсе. Даже рамочки в *Shelr* сделаны в стиле Ubuntu.

После этого вам предложат создать профиль. Задайте имя пользователя, адрес сайта и описание (последние два поля необязательны). Обратите особое внимание на ключ API – с его помощью в командной строке *Shelr* ваша учетная запись связывается с роликами. Рядом с ключом API указана команда вида: `shelr setup a9d0f926...`

Вставьте ее в терминал и запустите. Никакого ответа не последует, но ключ сохранится в файлах настройки *Shelr*. Вернитесь в браузер и нажмите Save [Сохранить], чтобы сохранить свою учетную запись, и вы войдете в систему. Теперь можно попробовать публиковать ролики от своего имени. Запишите сеанс *Shelr* как обычно, а затем опубликуйте его командой `shelr push`, указав идентификатор записи. После ввода тэгов и описания вы увидите URL – зайдите на него. На сей раз ролик будет опубликован не под именем Anonymouse, а под вашим собственным.

Вы обрели власть

Вы заметите, что теперь у вас появились дополнительные возможности, например, возможность изменить описание и тэги ролика после публикации. Также можно нажать Destroy [Удалить], чтобы удалить ролик, но запроса на подтверждение не последует – ролик удалится сразу. Также можно отметить видео как частное, и никто другой его не увидит; эта опция находится в окне Edit [Изменить]. Разработчики *Shelr* шутят, что с этой опцией их сайт «вышел на промышленный уровень», но мы думаем, что она очень полезна команде разработчиков с общей учетной записью, для обмена хитростями и приемами работы в командной строке.

Набрав просто `shelr`, вы увидите и другие параметры, о которых мы не упомянули. Они предназначены в основном для разработчиков, но знать о них стоит. Командой `shelr dump` с идентификатором ролика можно получить файл JSON этого ролика. JSON – формат хранения данных, обычно используемый приложениями JavaScript, и через этот формат программа *Shelr* взаимодействует с сайтом. Вы даже можете воспроизвести такие файлы, командой `shelr play` с именем файла. Это удобно, если вы хотите сохранить запись локально в одном файле без необходимости копаться в каталогах с данными (см. врезку «Приоткрывая завесу»). Также есть опция 'backend', определяющая, какую программу *Shelr* использует для воспроизведения ролика. Обычно это `script`, ее мы рассмотрели далее во врезке «Сохраняем вывод командной строки». Если с ней есть проблемы, можно изменить ее на альтернативный вариант – `tyrec`. По умолчанию в большинство дистрибутивов она не входит, но ее можно найти в репозиториях. Она также пригодится вам в следующем разделе нашего руководства...

Хорошая вставка

Если нужно разместить в Интернете некоторый текст не в форме видео, есть несколько вариантов это сделать. Лучший из них, безусловно, *Pastebin* (www.pastebin.com), бесплатный сервис, позволяющий скопировать текст из буфера обмена на web-страницу и получить ссылку на этот текст, которой можно поделиться с другими.

Это особенно удобно, когда на форуме или IRC нужно вставить содержимое файла настройки. Загромождать форум или IRC-канал полным текстом файла неэтично; отправив же текст в *Pastebin*, вы укоротите

свое сообщение, а к ссылке обратятся только те люди, которым это действительно необходимо.

Наша любимая возможность *Pastebin* – подсветка синтаксиса. Если вставить фрагмент кода (на C, Perl, Python, Bash, HTML и многих других языках), ключевые слова и переменные будут выделены цветом, это облегчит чтение кода. А если вы радуете об экологии и не хотите напрасно занимать место на серверах *Pastebin*, создайте ссылку только на заданное время, по истечении которого она исчезнет.



► *Pastebin* расцветывает исходный код дополнительными красками.

» Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

Альтернатива: Playterm

В мире Linux мы избалованы выбором, и мы расскажем вам об альтернативе, которую можно предпочесть, если *Shelr* у вас не работает или вам не по душе название сайта. *Playterm* (www.playterm.org) похож на *Shelr*, хотя и не настолько «отшлифован», но у него есть преимущество – для записи сеансов ему не нужна специальная утилита. Он использует вышеупомянутую *tttyrec*, для установки которой в Ubuntu и дистрибутивах на основе Debian достаточно выполнить **apt-get**. *tttyrec* похожа на программу *script*, но обладает рядом добавочных возможностей и более точным таймированием. Если ее нет в пакетах вашего дистрибутива, загрузите исходный код с <http://0xcc.net/tttyrec/index.html.en>.

Для запуска, наберите **tttyrec** в окне терминала. Никакого ответа не последует, но вы вернетесь в командную строку – с тем исключением, что на сей раз все происходящее на экране будет записываться. Закончив, нажмите **Ctrl+d** для выхода из сеанса *tttyrec*, и хотя вы снова ничего не увидите, в текущем каталоге создается файл **tttyrecord**. Попытка открыть его в текстовом редакторе выдаст бессвязную мешанину символов; но файл можно воспроизвести командой

```
tttyplay tttyrecord
```

Теперь заходите на сайт www.playterm.org/?upload=now, и вы увидите форму. Нажмите **Browse** [Обзор] и выберите созданный файл **tttyrecord**. Далее выбирается размер терминала – таковых предусмотрено два: 80 символов × 24 строки (стандартный размер для XTerm/Konsole/Gnome Terminal) и 120×35 (для истинно больших сеансов оболочки). Если создать файл **tttyrecord** другого размера, результаты могут выглядеть немного странно, но если вы изготавляете ответственный ролик, обеспечьте задание размера терминала как один из этих вариантов.

Нажмите на кнопку **Upload** [Выгрузить], и через несколько секунд появится окно подтверждения, где можно проверить ролик перед его загрузкой на сайт. Если вас все устраивает, жмите **Yes** [Да]. Затем вас спросят, сделать ли ролик публичным или приватным; в последнем случае его можно будет встроить в свой сайт, и через некоторое время он исчезнет. А если сделать его публичным, к нему можно добавить еще информацию – например, описание – и разрешить комментировать ваш шедевр. В итоге появится HTML-код для вставки на ваш сайт, а если ролик публичный – то и ссылка для просмотра на сайте *Playterm*. *Playterm* сделан вполне добротнo, но ему не хватает всех функций *Shelr*: например, нельзя менять скорость воспроизведения ролика. С другой стороны, мы обнаружили, что он лучше ладит с определенными программами (в частности, с теми, что используют *ncurses*), и если у вас возникли проблемы с одним из сервисов, стоит попробовать другой.

«Если у вас проблемы с одним сервисом, попробуйте другой.»



На сайте *Playterm* нам необычайно нравится еще кое-что – и это «живой» Linux, с которым можно поиграть. Зайдя на www.playterm.org/live, вы увидите черное окно с мигающим курсором. Оно выглядит не слишком выразительно, пока внезапно не появятся знакомые загрузочные сообщения... и вы попадете в командную строку. Да, это настоящий Linux, хотя и урезанный, и здесь можно попрактиковаться в выполнении различных команд. Впечатляет и способ реализации: это эмулятор ПК,

написанный на JavaScript, о чем несколько лет назад и мечтать было нельзя; но движки JavaScript в современных браузерах достаточно быстры, чтобы это сделать. Автор эмулятора – Фабрис Белляр [Fabrice Bellard], которого вы, возможно, помните по другим прекрасным эмуляторам ПК, таким как *QEMU*. И если вы завязли на компьютере с занудной Windows и нуждаетесь в дозе кислорода командной строки Linux, вы уж знаете, куда податься... **LXF**

► Создайте учетную запись, чтобы публиковать ролики от своего имени и делать их приватными.

Обратная связь

Вы создали блестящий ролик своих героических подвигов в командной строке? Или показали, как улучшить Gnome 3 парой команд **gsettings**, или придумали способ перекомпиляции *LibreOffice* без взрыва мозгов? В любом случае отправьте свои ссылки нам, по адресу linuxformat@futurenet.com

Сохраняем вывод командой *script*

Другая возможность сохранения скриптов оболочки – команда **script**. Она весьма неприязнительна, и ей далеко до *Shelr*, зато установлена по умолчанию почти во всех дистрибутивах Linux, и ею легко воспользоваться. Набрав **script**, вы увидите строку

```
Script started, file is typescript
```

Теперь приступайте к делу – вводите команды и просматривайте результаты. Все происходящее в окне терминала также будет сохранено в текстовом файле **typescript**. Закончив, наберите **exit**

(или нажмите **Ctrl+d**) для завершения сеанса, и вы сможете просмотреть файл командой **less typescript** (или открыть его в своем любимом текстовом редакторе).

Команда хорошо работает со стандартными командами оболочки, когда вы что-то вводите и получаете результат на экране, и похуже – с интерактивными программами, наподобие текстовых редакторов. Например, если во время сеанса запустить *Nano*, в результирующем файле ждите абракадабры.

Arduino: Гоним

Хитроумно манипулируя с таймерами, **Ник Вейч** превращает Arduino в генератор колебаний.



Наш эксперт

Когда **LXF** только появился, его держали на плаву исключительно скрипты *Bash* от **Ника Вейча**. Потом их заменили «люди», и это, по мнению Ника, стало шагом назад...

Что нам надо

- » По Arduino версии 1.0.
- » Arduino Duemilaprove или Uno.
- » Могут пригодиться динамик и подходящий конденсатор.

В прошлый раз мы говорили о... времени, а именно – в подробностях рассмотрели работу со счетчиком `Timer1` и узнали, как с помощью кое-каких низкоуровневых манипуляций сделать то, что нам нужно: запустить несколько надежных процедур прерывания типа реле времени. Но, как это ни было прекрасно, возможности таймера гораздо шире, и сегодня мы рассмотрим еще несколько примеров его использования. Можете не тревожить свой ящик с деталями, хотя динамик, пожалуй, не повредит.

Впрочем, перед тем как перейти к самим таймерам, выясним, как они используются в самом Arduino. Вы когда-нибудь интересовались тем, как чисто цифровое устройство формирует аналоговый выход? Этот специальный режим выходного сигнала можно включить на выводах 3, 5, 6, 9, 10 и 11 Arduino на базе `Atmega168` и `Atmega368`. При записи, например, значения 128 в один из этих выводов значение выходного сигнала оказывается равным (в данном случае) около 2,5 В.

Происходит следующее: на короткий интервал времени вывод включается и затем снова выключается. Он генерирует прямоугольное колебание, которое в среднем образует сигнал напряжением 2,5 В. В реальности это лишь очень быстрое переключение между 5 В и 0 В. Значение, которое вы записываете, определяет соотношение времени в состоянии ВКЛ ко времени в состоянии ВЫКЛ (коэффициент заполнения, часто выражаемый в процентах). Как и при ловкости рук, скорость здесь – важный фактор. Хотя среднее напряжение, независимо от того, включать и выключать его дважды или двести раз в секунду (при том же соотношении продолжительности ВКЛ/ВЫКЛ), будет одинаково,

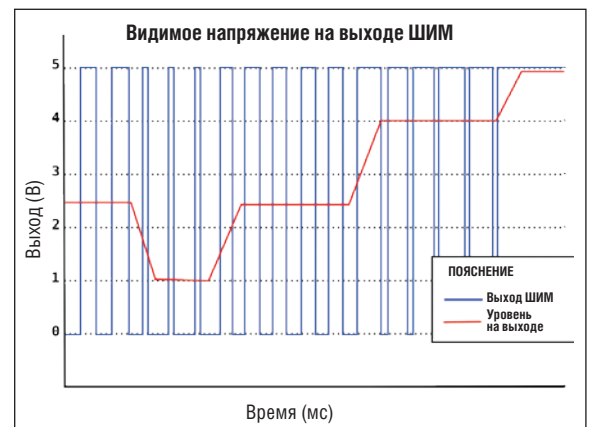
в первом случае вы получите мигающий свет, а во втором свет окажется тусклым (на самом деле он просто мигает очень быстро). Обычно частота прямоугольных колебаний составляет около 500 Гц при вызове функции `digitalWrite()`, но можно изменить сами часы, чтобы получить другие значения.

Может иметь значение основная частота широтно-импульсной модуляции (ШИМ). Настоящие электронные компоненты существуют в реальном мире, и хотя время нарастания сигнала (требуемое для изменения выходного сигнала с нуля [LOW] до единицы [HIGH]) в Arduino измеряется в наносекундах, время, необходимое для включения или выключения электромагнита, легко может быть в 1000 раз больше. Поэтому иногда основную частоту ШИМ нужно изменить. Для выводов 11 и 3, использующих встроенный `Timer2`, это не повлияет на остальную часть вашей программы. А вот для выводов 5 и 6 изменение частоты (т.е. изменение значения предварительного делителя частоты) имеет побочный эффект. С помощью импульсов этого таймера работают функции Arduino `millis()` и `delay()`, поэтому уменьшение частоты может увеличить задержки.

Основы ШИМ

Таймер управляет режимом ШИМ вполне очевидным образом. Таймер подсчитывает периоды сигнала ВКЛ до своего предела в 255 (для `Timer2`) и затем переполняется, снова начиная отсчет с нуля. Можно настроить регистр сравнения на любое из значений от 0 до 255, и когда счетчик достигнет выбранного значения, в зависимости от заданного режима могут быть выполнены различные действия, обычно включающие автоматическую установку значения определенного вывода.

Как всегда на микросхемах `Atmega`, есть несколько различных вариантов возможных действий. В самом простом случае регистр сравнения переключает выход в «единицу» и сбрасывается снова



» При быстром включении и отключении вывода, особенно если он направлен на каскад фильтров, его выход выглядит аналоговым, а не цифровым.

» **Месяц назад** Мы манипулировали встроенным таймером Arduino.

ВОЛНУ ТАЙМЕРОМ

при перезапуске счетчика. Тогда высокое значение регистра сравнения означает больше времени «выключено» и меньше «включено». Также можно использовать один из регистров сравнения в качестве верхнего предела счетчика, таким образом более точно отрегулировав его частоту (а также вычисления коэффициента заполнения).

Мы будем пользоваться ШИМ с коррекцией фазы, полагая, что сигнал достигает высшей точки ровно посреди временного интервала. Это означает и то, что сгенерированный сигнал имеет самый низкий уровень в начале и в конце этого временного интервала. Для генерации сигналов правильной формы, например, для управления двигателем или даже воспроизведения звука, это жизненно важно.

В режиме коррекции фазы счетчик ШИМ ведет себя иначе – сначала он досчитывает до 255, а потом вместо сброса начинает считать обратно до нуля. Если регистр сравнения настроен на изменение выходного сигнала в обоих направлениях, то в каждом направлении на состояния ВКЛ и ВЫКЛ приходится одинаковые интервалы времени, и коэффициент заполнения сохраняется. Явный недостаток этого режима в том, что эффективная частота уменьшается вдвое. Впрочем, это не должно стать проблемой для нас, так как при частоте процессора 16 МГц полоса остается приемлемой для аудио- и других ценных сигналов.

Больше шума

Мы можем воспользоваться возможностями ШИМ Timer2 для генерации звука. В предыдущих экспериментах со звуком мы полагались только на изрыгающую биты встроенную функцию `tone()`, которая формирует приемлемые, разве что звучащие сердито, прямоугольные колебания на заданном выводе. Но с возможностями ШИМ и небольшим хитрым кодом мы можем создавать звук любой формы и размера. Конечно, хотя мы считаем этот сигнал звуковым, фактически мы создаем сигналы, пригодные в самых разных ситуациях – как для тестирования другого оборудования, так и для непосредственного управления двигателями.

Для генерации колебаний мы можем имитировать аналоговые значения, используя ШИМ на высокой частоте для включения и выключения вывода. Но нужно задать это значение надежно. Тут нам поможет тот же самый таймер. Как и у `Timer1`, которым мы пользовались в предыдущей статье, у `Timer2` есть вектор прерываний. В режиме ШИМ с коррекцией фазы его можно заставить срабатывать по завершению полного цикла счетчика (т. е. когда счетчик досчитал до 255 и вернулся обратно к 0). Связав с этим процедуру обработки прерывания, при необходимости можно менять значение вывода ШИМ на каждом цикле вывода.

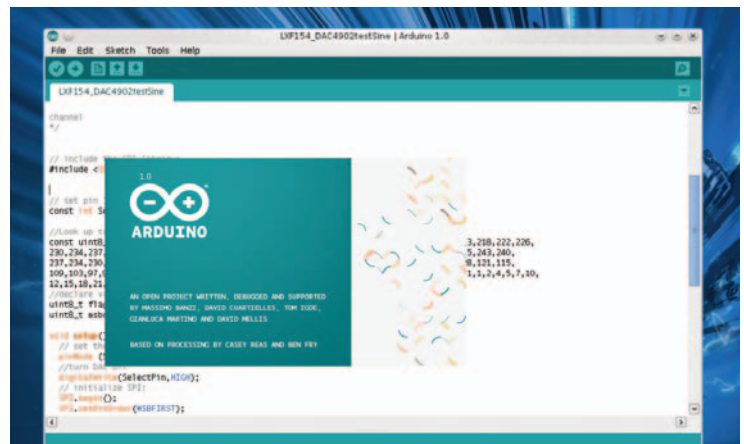
Полный цикл таймера составляет 510 тактов (конечные значения считаются только один раз), и это дает нам $16 \text{ МГц} / 510 = 31,372 \text{ кГц}$ – более чем достаточно для создания аудиосигнала. Как и в прошлый раз, для задания корректных значений таймера необходимо немного поработать со специальными регистрами микросхемы Atmega. Вот код для установки таймера:

```
TCCR2A = _BV(COM2B1) | _BV(WGM20);
TCCR2B = _BV(CS20);
```

Arduino 1.0!

Да, после нескольких лет пребывания в версии «ноль точка», команда Arduino, наконец, сочла ПО достойным присвоения ему номера 1.0. Если вы еще не обновились до этой версии, поспешите. Одна из мелких, но важных реформ – смена расширения по умолчанию для проектов с исходным кодом с `.pde` на `.ino`, чтобы исключить возможные конфликты расширений.

Формат файла не изменился, но вы обнаружите, что старые версии программ не распознают новое расширение, и чтобы открыть файл в новой версии программы, понадобится копирование и вставка. В новой версии можно сделать так, чтобы тип файла автоматически обновлялся для более старых проектов при их повторном сохранении, и это, пожалуй, будет удобно.



» В новой версии Arduino есть немного косметических изменений, а также несколько долгожданных реформ в библиотеках.

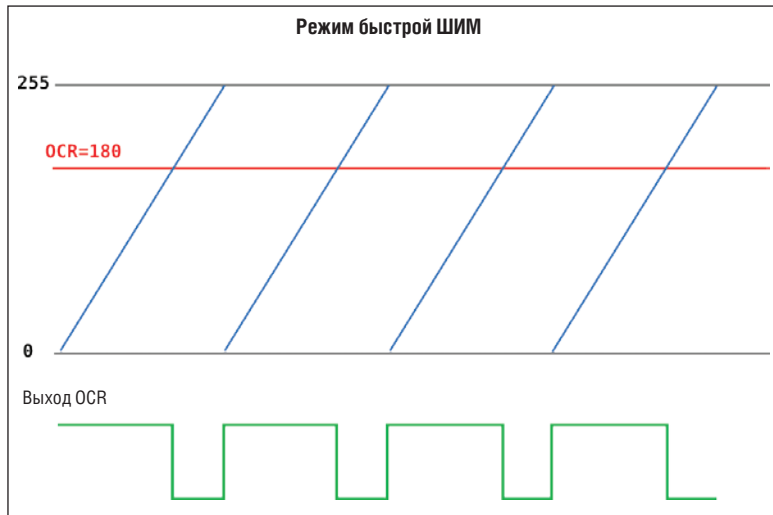
```
TIMSK2 = _BV(TOIE2);
pinMode(3,OUTPUT);
```

Если вы не читали прошлой статьи, то вряд ли что-то поняли. Мы не будем возвращаться к побитовой арифметике (то, что делает символ `|`), но немного пояснить происходящее все же стоит. `TCCR2A` и `TCCR2B` – два 8-битовых регистра (т. е. две специальные области памяти), которые содержат значения, управляющие работой таймера. Среди прочего они определяют, какой режим ШИМ используется и как быстро он работает. Каждый бит или набор битов задает определенный аспект работы таймера. Установленный бит `COM2B1` говорит компаратору `B`, что в режиме ШИМ с коррекцией фазы нужно сбросить выход в ноль при достижении верхнего предела и установить его в значение верхнего предела при достижении нуля (обратный режим). Бит `WGM20` переводит таймер в режим ШИМ с коррекцией фазы. Три менее важных бита в `TCCR2B` устанавливают значение делителя частоты таймера (сколько тиков системных часов нужно ждать для обновления значения таймера). Если установить только бит `CS20`, таймер будет работать с той же частотой, что и системные часы (а сброс всех битов в ноль отключает таймер). `TIMSK2` – другой регистр, который на сей раз содержит маску прерываний. При возникновении определенного

Скорая помощь

Хранить таблицы справочников в памяти расточительно – они занимают столько места! А у вас всего 2 КБ. Подготовьтесь к следующей части заранее, познакомившись с `PROGMEM` (<http://www.arduino.cc/en/Reference/PROGMEM>).

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)



С быстрой ШИМ счетчик сбрасывается в ноль при достижении верхнего значения. Это позволяет удвоить частоту режима с коррекцией фазы, но подходит не для всех случаев.

события (например, достижения таймером конца цикла) процессор считывает значение этого регистра, чтобы посмотреть, назначено ли прерывание для этого события. Если да, он обращается к вектору прерываний, другой специальной области памяти, в которой хранится адрес процедуры, которую нужно запустить.

Наконец, мы назначаем выходной сигнал на цифровой вывод 3. Почему на него? А он напрямую управляется регистром сравнения В таймера Timer2. Желая воспользоваться выводом 11, можно взять вместо него регистр сравнения А таймера Timer2. Только два этих вывода управляются напрямую с таймера.

Прерывание

Чтобы корректно задать прерывание, воспользуемся специальным макросом `ISR()`. Дело в том, что адрес прерывания нужно загружать в вектор прерывания, но пока мы не скомпилировали программу, этот адрес неизвестен. Этот волшебный макрос позаботится обо всем; нужно лишь объявить следующую процедуру:

```
ISR(TIM2_OVF_vect){
}

Указанный бит говорит макросу, с каким вектором мы хотим связать эту процедуру. Теперь нужно только заполнить ее. Раз уж мы возжелали иметь несколько различных типов колебаний, есть смысл воспользоваться конструкцией switch ... case. В языке C она позволяет выполнять различные блоки кода в зависимости от значения одной переменной. Она работает так:

switch(variable) {
case 1:
// что-нибудь делаем
break;
case 2:
// что-нибудь делаем
break;
...
default:
// остальное не подошло, сделаем это
}
```

Если значение переменной равно 1, выполняется первый блок кода, и т. д. Если это значение не соответствует ни одному из блоков, выполняется блок `default`, но он не обязателен. Каждый блок следует завершать оператором `break`, по которому происходит выход из всей структуры, в противном случае переменная продолжит сравниваться с каждым блоком (и в итоге выполнится код по умолчанию). Поэтому все, что нам нужно – переменная, кото-

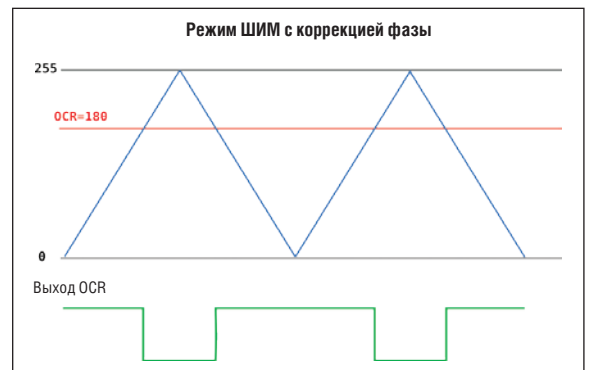
рая скажет нам, какой сигнал мы генерируем. Переменная должна быть глобальной и задаваться вне этой конструкции, чтобы можно было установить ее в основном коде. Назовем ее `wave.waveform`.

Да, и нужно сделать кое-что еще. Хотя код вызовется точно в нужный момент времени по прерыванию, на его выполнение может потребоваться разное время, в зависимости от того, какой блок кода будет выполнен. Чтобы избежать дрожаний выходного сигнала, мы должны подать значение регистра сравнения (т. е. нашу величину от 0 до 255) на выход с самого начала. Затем мы можем воспользоваться оставшейся частью прерывания для генерации выходного значения для следующего раза. У компаратора есть специальный адрес, но, к счастью, есть и условное значение, в которое его можно установить. Выходное значение будет еще одной из этих нудных глобальных переменных, поэтому назовем его `wave.output`. Начало нашей функции теперь выглядит так:

```
ISR(TIM2_OVF_vect){
OCR2B = wave.output;
switch(wave.waveform) {
case 1: // прямоугольная волна
```

Да, мы начинаем с прямоугольных колебаний! Ну, да, да, мы могли сделать это и раньше, но не на тех частотах, которые теперь у нас в распоряжении! Прямоугольные колебания довольно просты: полпериода волна проводит вверх, другие полпериода – вниз. Чтобы определить, когда ее включать и выключать, нужно знать частоту воспроизводимой ноты (`wave.frequency`) и количество маленьких порций времени, которые уже прошли (м-мм, `wave.f_counter`). После этого генерация волны сводится к проверке, в какой временной доле мы находимся, определении, не во второй ли мы половине полного цикла, и включении или выключении сигнала. Как мы знаем, каждая временная доля составляет 1/31372,5 секунды, и дело лишь за небольшой арифметикой:

```
case 1: // прямоугольная волна
wave.f_counter++;
if (wave.f_counter < (15686/wave.frequency)){
// первая половина цикла – выход максимален
wave.output=wave.volume;
}
else{
// вторая половина цикла – выход = 0
wave.output=0;
}
if (wave.f_counter > (31372.5/wave.frequency)){
// цикл завершен, сборос счетчика
wave.f_counter=0;
}
break;
```



В режиме с коррекцией фазы счетчик доходит до верхнего значения и начинает обратный отсчет; полный цикл равен 510 шагов.

» Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

Да, мы вбросили туда и `wave.volume` – почему бы нет? Тогда мы сможем управлять еще и амплитудой прямоугольных колебаний. Может быть, сейчас вас беспокоит, откуда будут браться эти значения и как мы будем ими пользоваться. Не волнуйтесь, именно так и стоит писать код. Теперь у нас есть все необходимые параметры, и можно создать специальную переменную, где все они будут храниться. В предыдущих руководствах мы рассматривали структуры, и они довольно просты. Это лишь способ задать составной тип данных – нечто вроде класса, но без методов.

```
struct generator{
    bool active;
    uint8_t waveform;
    uint16_t frequency;
    uint16_t f_counter;
    uint16_t duration;
    uint16_t table_value;
    double table_inc;
    uint8_t output;
    uint8_t volume;
} wave;
```

Можно порезвиться с некоторыми переменными и типами, но в целом это очень похоже на готовое управление звуком (к табличным данным перейдем через минуту). Как видите, в конец объявления структуры мы поместили имя `wave`. Это значит, что будет создан один экземпляр структуры с именем `wave`, и для доступа к ее элементам мы можем использовать конструкции `wave.waveform`, `wave.frequency`... весьма удобно, потому что мы уже написали код, который делает именно это.

Можно работать с частями этой переменной по отдельности, но лучше создать вспомогательную функцию для задания выходного значения. Она может принимать один аргумент (частоту) и устанавливать все необходимые значения следующим образом:

```
void playSquare (uint16_t f){
    wave.waveform=1;
    wave.frequency=f;
    wave.f_counter=0;
    wave.active=true;
    wave.volume=255;
};
```

Здесь мы присваиваем булевское значение переменной `wave.active`. Ее можно использовать в качестве ключа в начале процедуры прерывания – если волна выключена, то никаких действий предпринимать не нужно. Это поможет сэкономить несколько тактов процессора для других действий.

Синусит времен

Конечно, есть и более натуральные волны – хотя бы треугольные и синусоидальные волны. Синусоидальная волна лучше всего генерируется с помощью таблицы значений (см. врезку). Математические расчеты для вычисления значений на лету слишком затратны по времени. Помните, наши временные доли равны 1/31372,5 секунды – а на выполнение процедуры отводится не более данного интервала, и нужно экономить! Определить положение в таблице, найти значение и инкрементировать несколько счетчиков – это всего несколько тактов процессора (из возможных 510, которые у нас есть), так что это простой подход.

Как определить наше положение в таблице? Нам известен размер таблицы и частота воспроизводимой ноты – достаточно выполнить умножение, деление и посмотреть на остаток. Нам не нужно считать временные доли – все сложные вычисления можно выполнить заранее при задании параметров волны; у нас должны быть только счетчик индекса и переменная, содержащая

Гляньте-ка в таблицу!

До появления калькуляторов у каждого школьника (и, наверное, школьницы, хотя я не ходил в такую школу) была книжка с таблицей логарифмов. Каждый раз, когда вам нужен был логарифм числа, вы заглядывали в эту таблицу с мелкими цифрами, ища максимально близкое.

Ныне нам не нужны даже калькуляторы – у нас есть компьютеры, и большинство людей и не упомнят, что такое логарифм. Но по иронии судьбы поиск чисел в таблице, невероятно скучный для нас, для компьютера часто гораздо быстрее (в зависимости от требуемых вычислений). Расчет логарифмических и тригонометрических значений затретен, и, что иногда важно, для его выполнения может требоваться разное время. Взять значение в таблице надежно, и время процессора на это всегда требуется одинаковое.

Создание таблицы с результатами и поиск в ней очень эффективен, особенно когда дело доходит до микроконтроллеров, где нет сопроцессора для работы с плавающей точкой. Единственный недостаток – нужно сгенерировать и хранить таблицу с данными, и если вам нужна очень высокая точность или у вас нет запасной оперативной памяти, возможно, стоит выполнять вычисления на лету.

Во всех остальных случаях без таблиц соответствия не обойтись.

Объяснения еще нужны? Хорошо. Таблица, по сути, массив чисел, а индекс – значение, для которого вы ищете результат. Конечно, роль играют несколько факторов – число значений в массиве, или, если угодно, размер шага, зависит от требуемой точности данных (нечто вроде битрейта в музыке). Точность можно слегка увеличить, делая интерполяцию между двумя значениями из таблицы, но для этого нужны добавочные вычислительные ресурсы.

Единственное ограничение для таблицы соответствия – в том, что индекс должен быть целым числом: ведь нельзя же получить значение 3,82-го элемента в списке. Но это может быть и благом! Хотя точность при этом снижается, хранение чисел без плавающей точки делает их (обычно) меньше, и работать с ними становится определенно быстрее. Если вам хватает двух знаков после запятой, зачем вам четыре или пять, которые могут дать числа с плавающей точкой? Умножьте все на 100 и пользуйтесь целыми числами! Или, для более эффективного использования пространства, используйте числа до 255, которые аккуратно уместятся в один байт.

количество битов индекса, которые нужно пропускать с каждой долей. Мы делим значение счетчика на 256 и используем остаток для получения следующего значения. Если необходима большая точность, можно сделать эти значения действительными (с плавающей точкой) и работать с долями значений индекса (получить два и интерполировать), но размер таблицы в любом случае должен быть невелик – для хранения большей таблицы элементарно не хватит памяти. Поэтому теперь сложные расчеты выполняются во вспомогательной функции:

```
void playSine (uint16_t f){
    wave.waveform=3;
    wave.frequency=f;
    wave.f_counter=0;
    wave.table_value=0;
    wave.table_inc=(255.0/(31372.54/wave.frequency));
    wave.active=true;
}
```

Но реализовать ее в прерывании просто:

```
case 3: // синусоидальная волна
    wave.output=sinLUT[wave.table_value%256];
    wave.table_value += wave.table_inc;
    break;
```

Ничего не забыли? Ах да, сама таблица. Ну, это просто стопка чисел – ее можно найти на DVD. Треугольная волна также реализована в таблице соответствия – лениво, да? Но хотя значения для треугольной волны достаточно просто рассчитать, работать с таблицей все равно быстрее! Если вы хотите послушать некоторые звуковые частоты, подключите динамик между выводом 3 и землей, желательно с конденсатором. **LXF**

» **Через месяц** Можете отдохнуть от таймеров – займемся сетями!

Erlang: Распреде-

Андрей Ушаков осуществляет связь между узлами, где бы они ни были: и на одном компьютере, и в разных подсетях.



Наш эксперт

Андрей Ушаков активно приближает тот день, когда функциональные языки станут мейнстримом.

В двух прошлых номерах мы говорили о многозадачности и об устойчивости к ошибкам в языке Erlang. Сегодня мы продолжим разговор о средствах построения промышленных приложений и поговорим о таком важном понятии, как распределенные системы.

Основными элементами для построения распределенных систем на языке Erlang являются узлы. Узел – это всего лишь именованный экземпляр среды выполнения Erlang. Чтобы сделать экземпляр среды выполнения узлом, достаточно при ее запуске задать один из параметров `-name` или `-sname` и имя, которое узел получит. Параметр `-sname` задает короткое имя; это имя может использоваться, если создаваемые узлы находятся либо на одном компьютере, либо на разных компьютерах в одной подсети. Кроме того, если сервис DNS не доступен для создаваемого узла, то создавать узел можно только с параметром `-sname`. Параметр `-name` позволяет задать длинное имя; это имя может использоваться во всех тех же случаях, что и короткое, а также в случаях, когда узлы располагаются на разных компьютерах в разных подсетях.

Пусть имя компьютера является полностью определенным доменным именем – например, `stdstring.example.com`. В этом случае, если мы создаем узел с ключом `-sname test`, имя узла будет `test@stdstring`, а с ключом `-name test` – `test@stdstring.example.com`. Теперь предположим, что имя компьютера является обычным именем (а не полностью определенным доменным именем), например, `stdstring`. Тогда, если мы создаем узел с ключом `-sname test`, имя узла будет `test@stdstring`. Если же попытаться создать узел с ключом `-name test`, мы получим ошибку при создании узла (при запуске экземпляра среды выполнения Erlang). Решение в этом случае достаточно простое: при создании узла с помощью ключа `-name` следует указать полное имя узла. Если, например, создавать узел мы будем с ключом `-name test@stdstring.cpp`, то аргумент ключа `test@stdstring.cpp` и будет именем созданного узла. В дальнейшем, когда мы будем использовать имя узла, его следует задавать как атом. Если атомы в своем составе содержат символы, отличные от строчных латинских

букв и символа `'_'`, то тело атома должно быть заключено в одинарные кавычки. Так, например, для узла `test@stdstring.cpp` его имя мы будем задавать следующим образом: `'test@stdstring.cpp'`.

При назначении узлам имен следует учитывать следующее соображение: узлы с разными типами имен (т.е. с длинными и короткими) не могут взаимодействовать друг с другом через стандартный механизм обмена сообщениями. Если такое взаимодействие необходимо, задавайте создаваемым узлам имена одного типа. Отметим также, что если экземпляр среды выполнения Erlang не является узлом, то имя у такого экземпляра все равно есть: это атом `nonode@nohost`.

Помимо имени, при создании узла необходимо задавать также и некоторое «магическое» значение, называемое `magic cookie`. Данное значение является атомом и используется для процесса аутентификации узлов в момент установления соединения между ними. При этом происходит следующее: когда один узел пытается

установить соединение с другим узлом, среда выполнения Erlang сравнивает «магические» значения этих двух узлов; если они равны, то связь между узлами устанавливается, если же нет, связь меж-

«Узел — это именованный экземпляр среды выполнения Erlang.»

ду узлами установлена не будет. После установления соединения между узлами их «магические» значения более не используются. А значит, после установления соединения с каким-либо узлом с одним «магическим» значением, мы, изменив «магическое» значение на текущем узле, можем установить соединение с другим узлом с другим «магическим» значением. Пусть, например, у нас есть два узла – `node1@comp1` и `node2@comp2`, для которых «магические» значения установлены в **A** и **B** соответственно. Перед тем, как установить соединение с узлом `node1@comp1`, мы устанавливаем «магическое» значение на текущем узле в **A**, после чего, желая установить соединение с узлом `node2@comp2`, мы устанавливаем «магическое» значение на текущем узле в **B**. «Магическое» значение на узле можно установить тремя способами, причем два из них позволяют установить «магическое» значение узла только при его создании. Самый простой способ – ничего не делать; при этом среда выполнения Erlang (при запуске ее как узла) сама прочитает значение из файла `$HOME/.erlang.cookie` (а если такого файла нет, то среда выполнения Erlang его создаст). Чтобы при таком способе несколько узлов, расположенные на разных компьютерах, могли установить соединение друг с другом, на всех таких компьютерах содержимое файла `$HOME/.erlang.cookie` должно быть одинаковым. Если же мы создаем несколько узлов на одном компьютере, то нам беспокоиться не о чем: у всех узлов, создаваемых таким способом, будет одно и то же «магическое» значение. Другой способ, позволяющий задать «магическое» значение при создании узла – использовать опцию командной строки `-setcookie` и желаемое «магическое» значение. Например, если при создании узла с коротким именем `test` (на компьютере с именем `stdstring`) мы хотим, чтобы «магическое» значение было `some_value`, то сделать это мы можем сле-

Замечание о примерах

В данной статье, когда мы обсуждаем примеры, очень часто встречаются инструкции наподобие: на узле с именем `nodeName` введите команду `Cmd` (например – на узле `n1@stdstring` введите команду `nodes()`). По этому поводу заметим следующее. Узел является именованным экземпляром среды выполнения Erlang. Создавая узел, мы запускаем в консоли среду выполнения

Erlang с ключами. Среда выполнения Erlang – интерактивное консольное приложение; значит, после ее запуска мы работаем уже в консоли Erlang. А при вводе команды в консоли Erlang каждую команду необходимо завершать символом `“.”`; команда отправляется на выполнение после нажатия клавиши `Enter`. Так что не забывайте ставить точку.

ЛЕННЫЕ СИСТЕМЫ

дующим образом: `erl -sname test -setcookie some_value`. И наконец, последний способ установить «магическое» значение – это использовать функцию `erlang:set_cookie(Node, Cookie)` модуля `erlang`. Она позволяет установить «магическое» значение `Cookie` на узле `Node`. Например, чтобы установить «магическое» значение `SomeValue` на узле `node1@stdstring`, мы делаем следующий вызов: `erlang:set_cookie('node1@stdstring', 'SomeValue')`. Желая получить установленное «магическое» значение на текущем узле, мы должны использовать функцию `erlang:get_cookie/0` модуля `erlang`. Вполне понятна причина, по которой мы можем получить установленное «магическое» значение только на текущем узле; если бы мы могли получить его на любом узле, это сводило бы на нет систему аутентификации между узлами.

Проведем несколько экспериментов по установлению соединения между узлами. Для этого нам потребуется функция (BIF), о которой мы еще не говорили: `nodes/1`. Она позволяет получить список имен узлов определенного типа, так или иначе связанных с текущим узлом. Если в качестве аргумента передать атом `visible`, мы получим список видимых узлов, т.е. узлов, установивших обычное соединение с текущим узлом; если атом `hidden`, мы получим список невидимых узлов, т.е. узлов, установивших «невидимое» соединение с текущим узлом (мы поговорим об этом чуть позже); атом `connected` – список всех узлов, установивших соединение с текущим узлом; атом `this` – список, содержащий только имя текущего узла. И, наконец, если в качестве аргумента передать атом `known`, мы получим список всех узлов, известных текущему узлу, т.е. тех, которые когда-либо устанавливали с ним соединение (также этот список будет содержать и имя текущего узла). Кроме того, мы можем передать приведенные выше атомы в качестве списка аргументов; тогда мы получим список имен узлов, который является объединением списков имен узлов, полученных для каждого аргумента. Следует также сказать, что существует функция (BIF) `nodes/0`, являющаяся аналогом вызова `nodes(visible)`.

После этого небольшого вступления перейдем к собственно экспериментам. Для начала создадим три узла с короткими именами (на компьютере `stdstring`): `n1@stdstring`, `n2@stdstring` и `n3@stdstring`. Узлы в распределенной среде Erlang являются слабосвязанными; это означает, что пока к узлу не было обращения со стороны другого узла, соединение между этими узлами отсутствует. Если мы на узле `n1@stdstring` введем `nodes(visible)`, то получим пустой список; это означает, что узлы при создании не устанавливают соединения с уже созданными узлами. Для установления соединения между удаленным и текущим узлами необходимо использовать имя удаленного узла в одной из функций, обращающихся к заданному узлу. К таким функциям относятся, например, семейство функций `spawn/2,4` (о них мы поговорим чуть позже). Простейшей же функцией, которая устанавливает соединение между удаленным и текущим узлами, является функция `net_adm:ping/1` модуля `net_adm`. Эта функция проверяет, доступен ли удаленный узел для установления соединения (что ясно из названия этой функции); если да, то соединение между удаленным и текущим узлами устанавливается и возвраща-

ется атом `pong`, иначе возвращается атом `pang`. Давайте на узле `n1@stdstring` введем `net_adm:ping('n2@stdstring')`; после этого на узлах `n1@stdstring` и `n2@stdstring` введем `nodes(visible)`. На узле `n1@stdstring` мы получим список `[n2@stdstring]`, а на узле `n2@stdstring` – список `[n1@stdstring]`. Это означает, что соединение между этими узлами было установлено. Давайте на узле `n1@stdstring` введем `net_adm:ping('n3@stdstring')`; после этого на узле `n1@stdstring` введем `nodes(visible)` – и получим список `['n2@stdstring', 'n3@stdstring']`, что ожидаемо. Теперь на узлах `n2@stdstring` и `n3@stdstring` введем `nodes(visible)`; мы получим списки `['n1@stdstring', 'n3@stdstring']` и `['n1@stdstring', 'n2@stdstring']` соответственно. Этот несколько неожиданный для нас результат означает, что установление соединения – процесс транзитивный. Т.е. когда узел `A` устанавливает соединение с узлом `B`, уже имеющий установленное соединение с узлом `C`, то узел `A` попытается установить соединение и с узлом `C`.

Транзитивность процесса установления соединения является поведением по умолчанию; однако его можно изменить, создавая узел с ключом `-connect_all false`. Если такой узел участвует в процессе установления соединения, установится только одно соединение: между инициатором и явно указанным узлом (если вдруг «магические» значения у этих узлов не совпадут, то никакого соединения не будет). Предположим, что при создании узла `n3@stdstring` мы указали флаг `-connect_all false`, после чего на узле `n1@stdstring` последовательно сделали два вызова `net_adm:ping('n2@stdstring')` и `net_adm:ping('n3@stdstring')`. В этом случае, вызов `nodes(visible)` на узле `n1@stdstring` вернет `['n2@stdstring', 'n3@stdstring']`, вызов `nodes(visible)` на узле `n2@stdstring` вернет `['n1@stdstring']`, то же самое вернет и вызов `nodes(visible)` на узле `n3@stdstring`.

Подобное поведение при установлении соединения (отсутствие транзитивности) можно получить, если при создании узла указать, что это узел должен быть невидимым. Для этого при создании узла необходимо указать ключ `-hidden`. При установлении соединения между узлами, если хотя бы один из узлов является невидимым, будет установлено т.н. невидимое соединение. Это означает, что в результатах вызова `nodes()` или `nodes(visible)` узлы, соединенные с текущим узлом невидимым соединением, будут отсутствовать. Чтобы получить узлы, соединенные с данным узлом при помощи невидимых соединений, необходимо использовать вызов `nodes(hidden)`; чтобы получить все узлы, соединенные с данным узлом – вызов `nodes(connected)`. Для чего нужны невидимые узлы? Для того, чтобы вести проверку некоторых частей системы (располагающихся на одном или нескольких узлах), не влияя на всю систему целиком (не устанавливая соединения со всеми остальными узлами).

Давайте вернемся к нашему примеру. Предположим, что при создании узла `n3@stdstring` мы указали флаг `-hidden`, после чего на узле `n1@stdstring` последовательно сделали два вызова `net_adm:ping('n2@stdstring')` и `net_adm:ping('n3@stdstring')`. В этом случае, вызов `nodes(visible)` на узле `n1@stdstring` вернет `['n2@stdstring']`, вызов `nodes(hidden)` вернет `['n3@stdstring']`, а вызов `nodes(connected)` вернет `['n2@stdstring', 'n3@stdstring']`, а вызов `nodes(connected)` вернет `['n2@stdstring', 'n3@stdstring']` »

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

На темной стороне силы (на Microsoft Windows)

Когда мы создаем узел (именованный экземпляр среды выполнения Erlang) и не указываем (при помощи ключа `-setcookie`) «магическое» значение, оно берется из файла `$HOME/.erlang.cookie`, а если такого файла нет, то среда выполнения Erlang создает его. Пусть мы хотим создать несколько узлов, располагающиеся на разных компьютерах, которые должны быть соединены друг с другом, и не хотим при этом задавать «магическое» значение при помощи ключа `-setcookie` или вызова `erlang:set_cookie/2`. Тогда надо обеспечить, чтобы

содержимое файла `$HOME/.erlang.cookie` на всех компьютерах было одинаковым. Как мы знаем, язык и среда Erlang являются кроссплатформенными, и возможна ситуация, что часть узлов в такой сети будут создаваться на компьютерах под управлением ОС Microsoft Windows. Возникает вполне логичный вопрос, где в таком случае располагается файл, хранящий «магическое» значение по умолчанию. В Microsoft Windows этот файл точно так же называется `.erlang.cookie`, и располагается он в корне пользовательской директории. Найти расположение

этой директории нам помогут две переменные окружения: `%HOMEDRIVE %` и `%HOMEPATH %`. Первая возвращает букву диска, на котором располагается эта директория (в Microsoft Windows не существует единого дерева каталогов, как в Linux), а вторая – путь до пользовательской директории (без буквы диска). Поэтому полный путь до файла с «магическим» значением по умолчанию в ОС Microsoft Windows будет `%HOMEDRIVE %\%HOMEPATH %\.erlang.cookie`. Да пребудет с вами сила, и не обратитесь вы на ее темную сторону.

`stdstring`']. На узле `n2@stdstring` вызовы `nodes()`, `nodes(visible)`, `nodes(connected)` вернут `['n1@stdstring']`, а вызов `nodes(hidden)` вернет пустой список. На узле `n3@stdstring` вызовы `nodes()`, `nodes(visible)` вернут пустой список, вызовы `nodes(hidden)`, `nodes(connected)` вернут `['n1@stdstring']`.

Разорвать соединение между узлами можно двумя способами. Во-первых, можно завершить работу узла средствами операционной системы. Во-вторых, можно разорвать соединение между текущим и некоторым другим узлами при помощи функции (BIF) `disconnect_node/1`, которая в качестве аргумента принимает имя узла на другой стороне установленного соединения. В контексте разговора о разрыве соединений, поговорим также об известных узлах (узлах, которые мы можем получить вызовом `nodes(known)`). К известным относительно текущего узла относятся узлы, когда-либо устанавливавшие соединение с текущим узлом. Это означает, что, если соединение между текущим и каким-либо другим узлом уже разорвано, то этого узла в списке, который возвращает вызов `nodes(connected)`, уже не будет, а в списке, который возвращает вызов `nodes(known)`, он будет присутствовать. Если же соединение между текущим и каким-либо другим узлом не разорвано, то этот узел будет присутствовать как в списке, возвращаемом вызовом `nodes(known)`, так и в списке, возвращаемом вызовом `nodes(connected)`.

Опять вернемся к примеру. Пусть мы создали три узла `n1@stdstring`, `n2@stdstring` и `n3@stdstring` обычным способом (без всяких дополнительных флагов). Затем на узле `n1@stdstring` мы последовательно сделали три вызова `net_adm:ping('n2@stdstring')`, `net_adm:ping('n3@stdstring')`, `disconnect_node('n3@stdstring')`. Вызов функции `nodes(connected)` на узле `n1@stdstring` вернет список `['n2@stdstring']`, а вызов функции `nodes(known)` вернет список `['n1@stdstring', 'n2@stdstring', 'n3@stdstring']` (мы помним, что список известных узлов содержит также и имя текущего узла). На узле `n3@stdstring` все будет аналогично: вызов функции `nodes(connected)` вернет список `['n2@stdstring']`, а вызов функции `nodes(known)` вернет список `['n1@stdstring', 'n2@stdstring', 'n3@stdstring']`. И, наконец, на узле `n2@stdstring` (с которым никто из узлов соединение не разрывал) вызов функции `nodes(connected)` вернет список `['n1@stdstring', 'n3@stdstring']`, а вызов функции `nodes(known)` вернет список `['n1@stdstring', 'n2@stdstring', 'n3@stdstring']`.

Мы достаточно много говорили об узлах, но главный вопрос пока остался за кадром: как использовать узлы для создания

распределенных приложений. Ответ на этот вопрос достаточно прост: имя узла нам нужно только для создания процесса одной из функций из семейств `spawn/2,4`, `spawn_link/2,4`, `spawn_opt/3,5`, которые принимают имя узла в качестве первого параметра. И все. Эти функции возвращают идентификатор созданного процесса, для работы с которым имя узла не нужно. Для взаимодействия процессов опять же используются их идентификаторы, и знать узлы, на которых эти процессы располагаются, не нужно. Таким образом, видно, что при написании многозадачных приложений, выполняющихся в одном экземпляре среды выполнения Erlang и распределенных приложений, все отличие будет только при создании процессов. Это замечательный факт, и мы к нему еще вернемся при реализации практических задач.

Рассмотрим теперь базовый набор функций, предназначенный для работы с узлами (большую часть из этих функций мы уже видели и использовали).

«Разорвать соединение между узлами можно двумя способами.»

Для управления «магическими» значениями у нас есть две функции: `erlang:get_cookie/0` и `erlang:set_cookie/2`. Первая возвращает «магическое» значение текущего узла, вторая позволяет задать «магическое» значение произвольному

узлу. Чтобы разорвать соединение между текущим и заданным узлами, используется функция (BIF) `disconnect_node/1`. Функция (BIF) `is_alive/0` позволяет определить, может ли текущий узел (а точнее, экземпляр среды времени выполнения Erlang) быть частью распределенной системы узлов. Функции `monitor_node/2` и `erlang:monitor_node/3` позволяют включать, выключать и устанавливать опции мониторинга жизненного цикла узла. Если узел прекратит свою работу (или если узел не существует), текущему узлу будет послано сообщение `{nodedown, Node}`, где `Node` – имя узла, за которым осуществлялся мониторинг. Функция `node/0` позволяет получить имя текущего узла; функция `node/1` – имя узла, на котором расположен объект (процесс, порт, ссылка). Функция `nodes/1` возвращает список узлов, которые так или иначе связаны с текущим узлом и удовлетворяют определенным критериям (передаваемым в качестве аргумента); функция `nodes/0` является аналогом вызова `nodes(visible)`. И, наконец, семейства функций `spawn/2,4`, `spawn_link/2,4`, `spawn_opt/3,5` позволяют создать процесс на узле, имя которого идет первым аргументом этих функций.

Следующая тема – регистрация имен процессов. Чтобы взаимодействовать с каким-либо процессом (послать ему сообщение и, возможно, получить ответ), необходимо знать его идентификатор. Если один процесс создал другой процесс, то проблем

» Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

с взаимодействием нет: родительский процесс, зная идентификатор дочернего процесса, может послать ему в сообщении свой идентификатор, на что дочерний процесс может ответить родительскому, и т. д. Однако возможна и другая ситуация – когда мы создаем некоторый сервисный процесс обособленным способом, после чего этот сервисный процесс является точкой доступа для получения какой-либо услуги. Обычно, когда некоторый сервис предоставляет свои услуги, доступ к нему осуществляется при помощи некоторого стабильного имени. Идентификатор процесса для этого не подходит, т. к. не является стабильной величиной (это означает, что каждый раз, создавая такой сервисный процесс, я буду получать разные идентификаторы этого сервисного процесса).

Именно для этого и нужна регистрация имен процессов. Мы можем зарегистрировать некоторое имя (которое является атомом) и связать его с определенным процессом. В дальнейшем, при взаимодействии с этим процессом, мы можем использовать зарегистрированное имя вместо идентификатора процесса. Для регистрации имени процесса служит функция (BIF) `register(RegName, Pid)`, где `RegName` – регистрируемое имя процесса, `Pid` – идентификатор соответствующего процесса. Помимо регистрации имени процесса, мы можем эту регистрацию убрать (при помощи функции `unregister/1`), получить список всех зарегистрированных имен (при помощи функции `registered/0`) и получить по зарегистрированному имени процесса его идентификатор (при помощи функции `whereis/1`). У имен процессов, регистрируемых подобным образом, есть только один, но существенный минус: они регистрируются только на одном определенном узле (т. е. на двух разных узлах могут оказаться два процесса с одинаковыми зарегистрированными именами). Этот недостаток можно обойти и обращаться к процессу на узле `Node` по его зарегистрированному имени `Name` при помощи следующего синтаксиса: `{Name, Node}`.

Для примера создадим простой модуль (с именем `simple_message_handler`), который содержит только одну экспортируемую функцию `message_loop/0`. Эта функция представляет собой просто бесконечный цикл обработки сообщений, для каждого полученного сообщения выводящий его на экран. Вот тело этой функции:

```
message_loop() ->
receive
Message -> io:format("Message: ~p~n", [Message]),
message_loop()
end.
```

Откомпилируем этот модуль и создадим два узла: `n1@stdstring` и `n2@stdstring`. На узле `n1@stdstring` создадим процесс, рабочей функцией которого будет функция `simple_message_handler:message_loop/0` (бесконечный цикл обработки сообщений): `ProcessPid = spawn(fun simple_message_handler:message_loop/0)`. После этого зарегистрируем имя для вновь созданного процесса: `register(simple_service, ProcessPid)`. Теперь мы (все еще находясь на узле `n1@stdstring`) можем послать сообщение вновь созданному процессу при помощи зарегистрированного имени: `simple_service!{simple_message, message_data}`. В результате мы увидим следующий вывод на консоль (на узле `n1@stdstring`):

```
Message: {simple_message, message_data}.
```

Это означает, что сообщение созданным нами процессом получено и обработано. Теперь попробуем на узле `n2@stdstring` обратиться к созданному процессу по зарегистрированному имени: `simple_service!{simple_message, message_data}`. В результате мы получим ошибку времени выполнения, т. к. зарегистрированные имена являются локальными относительно узла. Попробуем обратиться к созданному процессу по зарегистрированному имени еще раз, но используя специальный синтаксис: `{simple_service,`

`n1@stdstring}`!{simple_message, message_data}. В результате на узле `n2@stdstring` выполнение этого выражения будет успешно, а в консоли на узле `n1@stdstring` мы увидим следующий вывод:

```
Message: {simple_message, message_data}.
```

Как мы увидели, несмотря на то, что регистрация имен локальна для каждого узла, мы можем использовать зарегистрированные имена на любых узлах при помощи специального синтаксиса. Но, во-первых, этот синтаксис не очень удобен, т. к. вынуждает задавать два имени вместо одного, а во-вторых (и это достаточно серьезное ограничение), этот синтаксис раскрывает детали внутренней реализации системы: на каком узле и с каким именем располагается сервис. Чтобы преодолеть эти недостатки, нам нужен механизм, позволяющий регистрировать имена глобально (на уровне используемой сети из узлов, образующих приложение). Для этих целей служит модуль `global`; более подробно о функциях из этого модуля мы поговорим в следующем номере, а пока ограничимся лишь глобальной регистрацией имен. В модуле `global` определены следующие функции (полный аналог функций, регистрирующих имена процессов локально): `global:register_name/2`, `global:unregister_name/1`, `global:whereis_name/1`. Но при использовании этих функций существуют и некоторые различия: во-первых, узел, желающий воспользоваться глобальным зарегистрированным именем, д. б. известен узлу, который эту регистрацию осуществил; во-вторых, для отсылки сообщения при помощи глобального зарегистрированного имени следует использовать функцию `global:send/2`. Обычная функция (BIF) `send/2` и оператор `!` с глобальными зарегистрированными именами не работают. Наш пример в этом случае будет выглядеть так. На узле `n1@stdstring` мы вводим команды

```
ProcessPid = spawn(fun simple_message_handler:
message_loop/0).
global:register_name(simple_service, ProcessPid).
global:send(simple_service, {simple_message, message_data}).
```

А на узле `n2@stdstring` мы вводим команды

```
net_adm:ping('n1@stdstring').
global:send(simple_service, {simple_message, message_data}).
```

В результате на узле `n1@stdstring` на консоль дважды выведется следующая строка:

```
Message: {simple_message, message_data}.
```

Сегодня мы рассмотрели средства для построения распределенных приложений на языке Erlang. Мы увидели, что специфика построения распределенных приложений в основном касается инфраструктуры; основной код приложения при этом особо не меняется, по сравнению с обычным многозадачным кодом. Как следствие, это упрощает написание распределенных приложений (что мы еще увидим на практике). Мы пока не прощаемся с темой, посвященной многозадачности и распределенным приложениям: в следующий раз мы поговорим о том, какие у нас есть библиотечные средства для построения многозадачных и распределенных приложений. [LXF](#)

Ошибка в предыдущем номере

В предыдущем номере в заметках я привел несколько примеров решения проблемы с очисткой ресурсов на языке C. Второй пример содержит досадную ошибку: везде вместо сравнения на неравенство `“!=”` написано сравнение на равенство `“==”`. Ниже приводится правильный вариант этого примера:

```
int descr1, descr2;
```

```
descr1 = open("file1.dat", O_RDWR);
if (-1!= descr1) {
    descr2 = open("file2.dat", O_RDWR);
    if (-1!= descr2) {
        some_task(descr1, descr2);
        close(descr2);
    }
    close(descr1);
}
```

Создаем свой

Мы в LXF любим суперкомпьютеры и свертывание белка, так почему бы не объединить это вместе в супер-руководстве от Дэвида Хейворда?



Наш эксперт

Дэвида Хейворда вырастили тюлени, природные враги пингвинов, но это отнюдь не мешало ему любить Linux.

Что нам надо

- » Как минимум два компьютера.
- » Два экземпляра BCCD на диске или на USB-брелке. ISO-образ есть на нашем DVD.
- » Локальная проводная сеть с доступом к Интернету и DHCP.
- » Номер вашей любимой команды Folding@Home, либо – милости просим в команду LXF, 217805.

Что такое суперкомпьютер? Не монолитное ли это чудовище, гудящее в потайной серверной с кондиционером? Или, когда вы слышите это слово, на ум приходят компьютеры из «Военных игр» или «Супермена 3»? Ныне термин «суперкомпьютер» означает просто группу процессоров, параллельно работающих над выполнением любой поставленной задачи. Современный суперкомпьютер – просто набор определенным образом настроенных компьютеров (узлов кластера), иногда со специализированной операционной системой или набором программ для параллельной обработки данных.

Кластеризация для всех

Конечно, на самом деле все немного сложнее, но заберись мы в дебри суперкомпьютеров и кластеризации, на рассказ ушел бы весь день; поэтому не будем. Обычно мы представляем себе суперкомпьютеры (или кластеры, как мы будем называть их далее) как что-то недоступное нам, простым смертным, и принадлежащее полубогам IT, чья работа – обитать в темных уголках серверных комнат, спрятанных глубоко во внутренних помещениях огромных корпораций, кормить кластеры и удовлетворять их потребности.

На самом деле все совсем не так, по крайней мере, в основном, но до самых недавних пор кластеризация была недоступна обычным людям. С появлением Linux и бесплатных открытых операционных систем каждый, у кого есть немного знаний и по меньшей мере пара компьютеров, может успешно создать кластер и пользоваться им почти так же, как это делают в NASA. Кластеры можно применять для решения многих повседневных задач,

стоит только захотеть. В кластерах высокой доступности HA (High Availability) есть избыточные узлы, вводимые в действие, если другой узел выходит из строя. Они используются в основном для рабочих целей, но вы можете пользоваться ими и дома, например, для запуска web-хостинга.

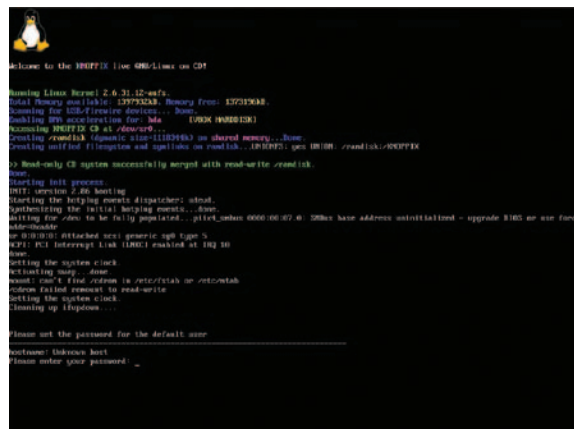
У вас может быть кластер хранения файлов, который предоставляет дисковое пространство для использования несколькими компьютерами через сеть – дома его можно использовать как сервер мультимедиа. Эта идея лежит в основе многих популярных облачных сервисов.

У нас может быть кластер обработки данных, или вычислительный кластер. Он использует циклы простоя процессоров на каждом узле для выполнения вычислений или моделирования. Последним мы и займемся далее, запустив его поверх нашей локальной сети и используя доступные нам ресурсы. Подобная архитектура также известна как кластер Беовульф.

Folding@Home

Folding@Home, или FAH, существует несколько лет. С помощью распределенных вычислений проект помогает понять, как молекулы белка свертываются, ренатурируют и вызывают заболевания. Участие в этом проекте позволяет ученым прояснить течение синдромов Альцгеймера, Хантингтона и Паркинсона и, будем надеяться, определить механизм лечения. Если вам интересны подробности, зайдите на <http://folding.stanford.edu/English/HomePage> и прочтите текст о науке, стоящей за этим ПО.

Конечно, кроме гуманного аспекта здесь есть и игровой элемент. Команды, «свернувшие» наибольшее количество белков, попадают в чарты. Мы с нашим кластером не стремимся попасть на вершину чартов, мы просто хотим немного порезвиться с суперкомпьютерами и помочь проекту FAH. Для этого мы создали Команду свертывания белков Linux Format (Linux Format Folding Team), или LXF, под номером 217805, и ею мы воспользуемся на нашем уроке.



» Рис. 1. Для загрузки BCCD достаточно нажать Enter после пустого пароля.

суперкомпьютер

Первые шаги к успеху кластера

Загляните во врезку «Что нам надо»; мы предполагаем, что у вас уже есть два компьютера и локальная сеть. Без них создание кластера может усложниться. Мы также уточнили, что сеть должна быть проводной, так как не уверены, что операционная система BCCD поддержит все функции всех беспроводных устройств, которые могут у вас оказаться. Хотя большинство дистрибутивов Linux хорошо поддерживают беспроводные сети, лучше воспользоваться проводной, чем остаться без кластера.

Следующий пункт в списке – экземпляр ISO BCCD. BCCD, сокращение от Bootable Cluster CD [Загрузочный диск кластера]; это образ Knoppix Linux, поставляемый с готовыми системами распределенных вычислений.

Возьмите ISO-образ с нашего диска или загрузите его с bit.ly/o0tbc0. Мы воспользовались 32-битным образом версии 3.x. Запишите образ на несколько дисков с помощью своей любимой программы или на несколько USB-брелков с помощью *Unetbootin*. Почему на несколько? Ну, вы ведь хотите запускать это не на одном компьютере, правда?

Покончив с записью (мы воспользовались брелками), включите свой первый компьютер, или главный узел, как мы будем его называть, откройте меню загрузки ПК, обычно клавишей F12 или похожей, и выберите носитель, с которого нужно загружаться. Это Live-образ, и он не повлияет на данные, хранимые на жестком диске – но, как и в большинстве таких случаев, лучше сделать резервную копию.

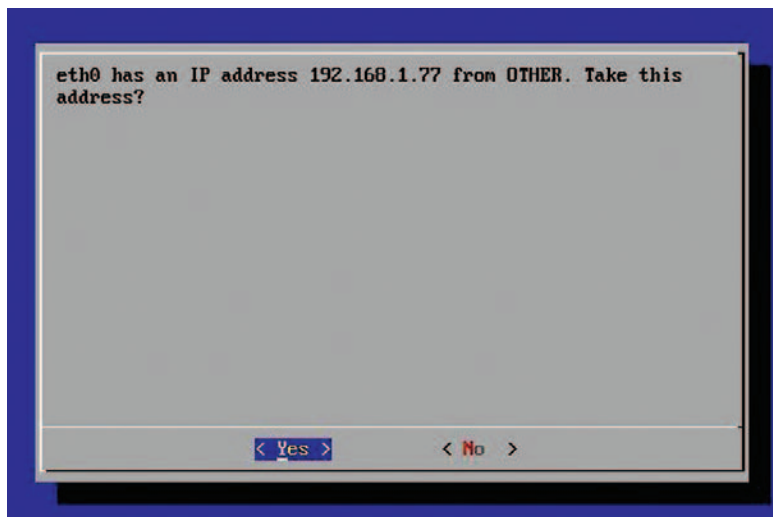
Загрузка главного узла

Во время загрузки BCCD задаст несколько вопросов; первым из них будет предложение нажать Enter для загрузки. Если при появлении заставки BCCD этого не происходит, не волнуйтесь, это загрузчик *Unetbootin* (или что-то похожее) перехватывает управление. Если вверху экрана есть пингвины и образ загружается, все хорошо.

Затем будет запрошен пароль (имя хоста заполняется автоматически). Просто нажимаем Enter, чтобы оставить его пустым (мы сделали так на обоих компьютерах); последнее, что нам нужно – чтобы пароли не совпали из-за неверно напечатанного слова, поэтому нажмите Enter для задания пустого пароля и еще раз – для подтверждения (см. рис. 1).

Сеть

Если система BCCD распознает сетевую карту в вашей системе, следующий вопрос будет простым: «Это ваш сетевой адрес от DHCP (обычно, от маршрутизатора), принять? [this is your network address from the DHCP (usually your router), do you accept?].» Ответ в общем случае – да, так как здесь мы не хотим слишком много возиться с сетевыми настройками BCCD. Просто нажмите Enter на ответе «да [yes]», и загрузка продолжится. См. рис. 2.



► Рис. 2. Настройка сети по умолчанию в BCCD лучше не менять.

Рабочий стол BCCD

Вслед за этим должен появиться рабочий стол BCCD. Если этого не происходит и вы остаетесь в текстовом режиме, причина, возможно, в том, что не удалось загрузить X-сервер; в этом случае наберите `startx` и подождите, пока рабочий стол загрузится. Если все пройдет успешно, вы увидите на экране нечто вроде рис. 3.

Прополощем, постираем, повторим

Отлично, первый барьер преодолен. Оставьте первый компьютер и перейдите ко второму, он же – узел 2. Повторите те же действия: пустой пароль и т.д. Сетевая карта должна получить другой IP-адрес (если нет, перезагрузитесь и попробуйте снова), и через минуту или две вы должны смотреть на два одинаковых рабочих стола BCCD.

Создание кластера

Прежде чем начать смеяться злобным смехом и гладить белую кошку, наберем несколько команд, с целью убедиться, что два компьютера могут связаться друг с другом и кластер будет

Кластер по имени...

BCCD – не единственная кластерная система. На поиски в сети можно потратить целый день, но чтобы вы не слишком напрягали глаза, вот еще несколько:

- Clusterknoppix
- Quantian

► Pelican HPC

► Parallel Knoppix

Одни кластеры лучше, другие хуже, но обязательно прогуляйтесь на их сайты и убедитесь, что проекты еще живы и развиваются.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

работать. На главном узле в верхнем окне Терминал [Terminal] RX-VT (с черным фоном) введите следующие команды:

```
bccd-allowall
```

Эта команда соберет ключи SSH хостов, что позволит обмениваться данными между двумя хостами без ввода паролей. Теперь введите команду:

```
bccd-snarfhosts
```

Она создаст или перезапишет файл **machines**, используемый программой **MPI** – подробнее о ней чуть позже. Сделайте то же самое на узле 2; хотя в теории этого не требуется, мы обнаружили, что **BCCD** лучше работает с одинаковыми файлами **machines**. Если вам интересно, что в этом файле, наберите в окне терминала главного узла команду:

```
vi ~/machines
```

Откроется текстовый файл, состоящий из нескольких строк. Первой будет нечто вроде **node000.bccd.net slots=1**. Это главный узел – так **BCCD** идентифицирует его в сети. Во второй строке будет узел 2, как **BCCD** идентифицирует его в сети – **node009.bccd.net slots=1**. Если в файлах **machines** одно и то же, отлично, вы почти у цели. Для выхода из *Vi* наберите

```
:quit
```

и нажмите Enter.

Разворачиваем FAH

Освоить **FAH** нетрудно. Несмотря на то, что последняя версия – 7, мы воспользуемся версией 6, так как она уж точно работает с **BCCD**. На главном узле введите следующую команду в окне терминала:

```
mkdir -p ~/folding
```

Создается каталог **folding** с родительскими каталогами (при необходимости). Затем выполните команду

```
cd ~/folding
```

Вы перейдете в новый каталог **folding**. Команда

```
wget http://www.stanford.edu/group/pandegroup/folding/release/FAH6.02-Linux.tgz
```

загрузит клиент **FAH** в 32-битной версии с внешнего сервера. Команда

```
tar xzf FAH6.02-Linux.tgz /
```

распакует архив и извлечет необходимые файлы.

Теперь у нас есть клиент, и его нужно настроить для работы в **BCCD**. Для этого введите следующую команду:

```
chmod +x fah6
```

Она сделает программу **fah6** исполняемой. Наконец, нужно сделать так, чтобы клиент работал с должной командой участников и именем пользователя. Введите команду:

```
./fah6 -configonly
```

Будет запущена утилита настройки клиента **FAH**. У вас запросят имя пользователя (не важно, что вы введете, если у вас его пока нет) и команду бойцов, за которую вы хотите «сворачивать» белки – можно выбрать свою собственную или присоединиться к команде **Linux Format** с номером 217805. Остальные параметры лучше не менять, поэтому просто нажмите Enter. Если вы осознаете, что делаете с **FAH**, задайте те значения, которые лучше всего соответствуют вашим потребностям.

MPI: Ввод и вывод

MPI (*Message Passing Interface* – интерфейс передачи сообщений) – программа, которая позволяет процессам взаимодействовать друг с другом путем отправки и приема сообщений через сеть. Она используется для программирования параллельных вычислений и выступает в качестве посредника между многими узлами, которые потенциально могут общаться друг с другом.

Другими словами, она нужна нам затем, чтобы **FAH** использовал процессоры на обоих компьютерах так, как если бы он рабо-

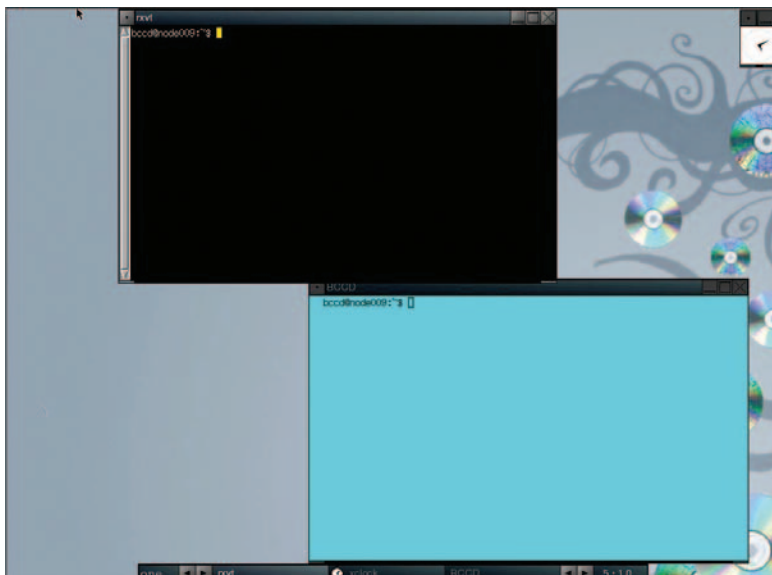


Рис. 3. Рабочий стол BCCD во всей своей серой, черной и бирюзовой красе.

Распределенные вычисления

Если вам почему-либо не нравится **FAH** или вы не согласны с ним, вот краткий список других всемирных проектов параллельных вычислений:

- **SETI@Home** – поиск внеземных цивилизаций с SETI.
- **Docking@home** – модель докинга (стыковки) молекул белка.
- **GPUGRIP.net** – проводит полное атомно-молекулярное биомоделирование с картами на базе CUDA [разработанная компанией NVIDIA программно-аппаратная архитектура, позволяющая производить вычисления с применением графических процессоров NVIDIA и технологии GPGPU

(произвольных вычислений на видеокартах), – прим. пер].

- **Climateprediction.net** – предпринимает попытку сузить область неопределенности климатических моделей.
- **AQUA@home** – использует Quantum Monte Cairo для предсказания производительности сверхпроводниковых адиабатических квантовых компьютеров. Кхе...
- **Einstein@home** – к сожалению, не модель самого гения в натуральной величину, а поиск гравитационных волн.

Проектами буквально полнится Интернет – уверен, вам удастся найти что-нибудь для себя.

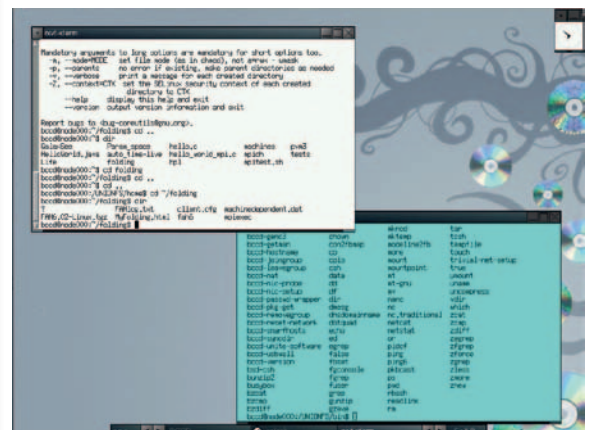


Рис. 4. Сворачивание в кластере или завихрение клякс? Право, я начинаю путаться.

➤ Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

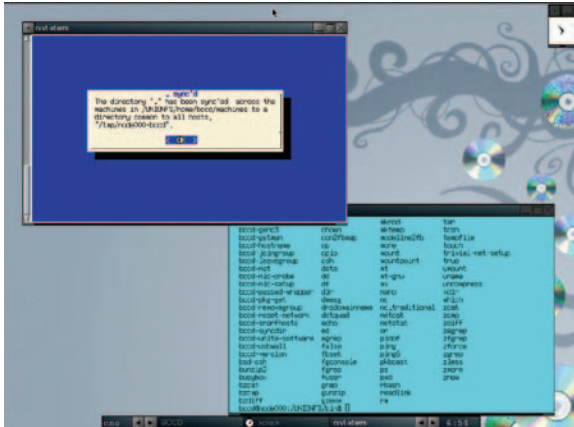


Рис. 5. Запишите каталог TMP, который упоминает BCCD.

тал на одном компьютере. На самом деле это очень сложно. MPI управляет вводом и выводом данных и гарантирует, что они верны и в них нет ошибок; это незаменимо для программы вроде FAH.

Работай, MPI, работай

Если просто запустить FAH на главном узле, то клиент будет принимать пакеты и обрабатывать их, но без помощи второго узла.

Теперь нам нужно сообщить главному узлу, что он будет запускать клиент FAH в кластере из двух уже идентифицированных узлов (самого себя и узла 2) в соответствии с файлом `machines`, так что ему будут доступны вычислительная мощность и результаты обработки с обоих компьютеров. Не спешите пугаться – это значительно проще, чем кажется.

Первая задача – синхронизировать каталог `folding` с обоими узлами. Для этого перейдите во вновь созданный каталог `folding` (на главном узле), набрав в терминале: `cd ~/folding`. Текущий каталог можно определить по строке приглашения. Она должна быть `bccd@node000:~/folding$`, как на рис. 4. Когда вы окажетесь в каталоге FAH, введите следующую команду (только на главном узле): `bccd-syncdir ~/machines`.

Она создаст синхронизируемую папку на основе информации из файла `machines`, это каталог `~/folding`, а идентифицированные в кластере компьютеры, которые получают доступ к этому файлу – `node000` и `node009` (главный узел и узел 2). Через пару минут на экране должно появиться окно, показанное на рис. 5.

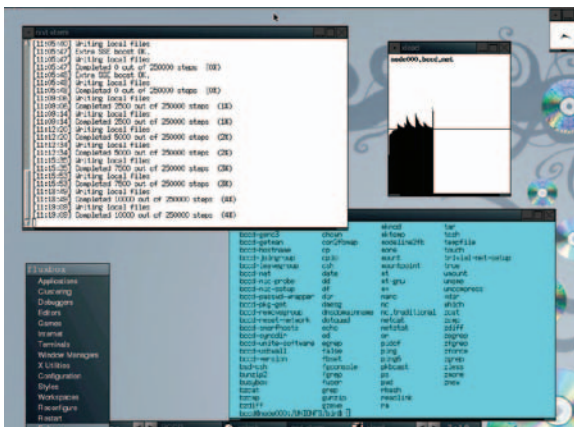


Рис. 6. FAH в кластере. Следующая остановка – мировое господство.



Запишите временный каталог, использованный при синхронизации, в данном случае это `/tmp/node000-bccd`. Он понадобится нам на следующем шаге. Затем нажмите OK для возврата в терминал. Синхронизация настроена; осталось вызвать

А вот этот мы создали раньше. Для одного вечера неплохо.

«На ум приходят компьютеры из “Военных игр” или “Супермена”?»

MPI для запуска клиента FAH в синхронизируемом каталоге, доступ к которому имеют оба наших узла.

Чтобы запустить FAH в кластере, введите следующую команду в окно терминала:

```
mpirun -machinefile ~/machines -np 6 /tmp/node000-bccd/./fah6 -verbosity 9
```

Эта команда просит MPI запускаться и заглянуть в файлы `machines` для определения активных узлов. Ключ `-np 6` сообщает MPI, что исполняемая команда должна использовать шесть процессоров. Каталог `/tmp` – временный, он используется при синхронизации. Часть `./fah6 -verbosity` – исполняемый файл, который запускает FAH. Нажмите Enter, и вы должны увидеть окно, показанное на рис. 6.

Кластер благодетей

Ну вот и все – поздравляю, вы только что создали свой первый суперкомпьютер, или кластер. И послужили на благо людям. Если вы ждали завывания сирен и десантников в масках, врывающихся в комнату через окно – простите, что разочаровал. Щелкнув правой кнопкой мыши на рабочем столе каждого узла, вы можете открыть утилиты X и в них `Xload`; так вы получите базовую информацию по загрузке процессора.

Если вам понадобятся графические клиенты для сбора статистики, поищите сами и выберите наилучший для вас. Но для начала щелкните правой кнопкой мыши на рабочем столе, выберите Clustering [Кластеры] и затем XPM. Кликните на Hosts [Хосты] и добавьте другой хост (`node000` или `node 009`), затем кликните на Tasks > Spawn [Задания > Размножить] и в командной строке наберите:

```
/tmp/node000-bccd/fah6
```

Посмотрите на различные представления – они должны дать вам приблизительное понятие о происходящих там процессах.

Остальное – за вами: как далеко вы намерены зайти в этом проекте? Есть ли у вас другие компьютеры, которые можно добавить в кластер? Что бы вы ни решили, помните, что здесь мы только коснулись поверхности, поэтому двигайтесь дальше и учреждайте кластеры во славу *Linux Format*. LXF



Командная строка Оказывается, что не так она страшна, как ее малюют

Elinks: Текстовый

Аскетические наклонности Тихона Тарнавского внушают ему легкую неприязнь к графическим наворотам – в том числе и у web-браузеров.



Наш эксперт

Тихон Тарнавский работает в Linux-консоли больше восьми лет и точно знает, как сделать эту работу удобной.



Сколько бы Интернет ни развивался в сторону интерактивности, графики и видео, а все же значительную его часть по-прежнему составляет и будет составлять текстовая информация. И при работе с этой текстовой информацией распространенные ныне полнофункциональные и тяжеловесные браузеры часто оказываются избыточными. И дело даже не только в тяжеловесности как таковой, хотя по требовательности к ресурсам компьютера, особенно при множестве открытых вкладок, такие браузеры растут примерно с такой же скоростью, что и мощность самих компьютеров. Зачастую лишняя информация в виде картинок, флэш-роликов или боковых меню сайта оказывается действительно лишней и мешает воспринимать информацию основную.

В таких случаях и бывает полезно воспользоваться браузером текстовым, который всю эту лишнюю информацию сам по себе тщательно скрывает. Последнее время такие браузеры почти забыты и считаются отсталыми и недоразвитыми. И совершенно незаслуженно, так как есть среди них один, некоторыми своими возможностями даже превосходящий ряд своих графических собратьев. Именно эти его возможности мы сейчас и рассмотрим подробнее. Называется он *elinks*. Название отсылает нас к друго-

му консольному браузеру – *links*, от которого герой этой статьи и происходит (подробнее см. врезку «Родословная героя»). Изначально оно происходило от «experimental links». Со временем расшифровка была переименована в «extended/enhanced links», то есть «расширенный/улучшенный links».

Мы расскажем об основных принципах работы с *elinks*, а как именно эту работу организовать – уже дело личное; как и то, в каких случаях использовать его, а в каких – полновесный графический браузер. Обратите внимание: хотя стабильной считается версия *elinks 0.11*, использовать нужно версию *0.12pre1* и выше (актуальная версия – *0.12pre5*), так как только в ней появилась поддержка локали UTF-8, которая используется по умолчанию практически во всех современных дистрибутивах. На практике эта версия ведет себя вполне стабильно.

Главное — начать

А начать нужно с открытия некоего адреса, будь то страница в Интернете или локальный файл. В начале работы, когда у вас еще нет собственноручно созданных закладок и сокращений для адресов, простейший способ – воспользоваться окном ввода адреса. В стандартной конфигурации это окно открывается автоматически при запуске браузера или открытии новой вкладки. Если в вашем дистрибутиве умолчания изменены, его можно вызвать клавишей **g** (от «go» – «пойти» или «перейти») и в нем ввести нужный адрес. Можно также задать адрес в виде параметра командной строки при запуске *elinks*, как и для многих других браузеров. Зайдя на некую web-страницу, логично будет перейти к базовой навигации: прокрутке страницы и переходу по ссылкам. Сначала только скажем пару слов в целом о концепции управления *elinks*.

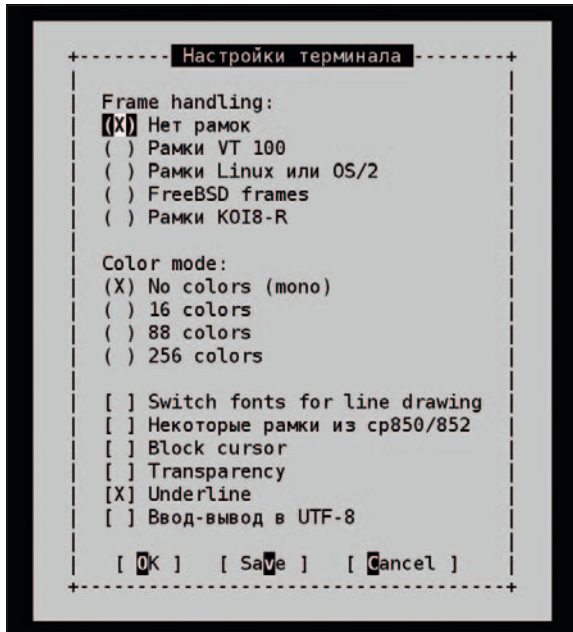
Обычный режим работы с консольным браузером подразумевает управление исключительно с клавиатуры. После получения минимальных навыков это очень удобно, но поначалу скорее всего окажется не слишком привычным. Поэтому сразу пара советов. Первый: в отличие от других консольных браузеров, *elinks* пол-

«Бывает полезно воспользоваться браузером текстовым.»

ностью поддерживает управление мышью. Правую кнопку скорее всего перехватит эмулятор терминала, но она и не нужна: прокрутка доступна по вращению колесика мыши, а все остальное управление осуществляется «одной левой». Если же вам понадобится использовать мышь для выделения текста или вставки его нажатием средней кнопки (колесика), удерживайте Shift: это временно отключит «мышинное» управление браузером и переведет мышь в «обычный» режим.

Впрочем, управление с клавиатуры на практике оказывается куда удобнее и быстрее, так что перейдем ко второму совету: не старайтесь заучить умолчательное расположение управляющих клавиш – особенно если они вам не покажутся удобными. *Elinks* – добротнo проработанная программа в этом смысле, и полностью следует разумной концепции: компьютер – всего лишь инструмент, он должен уметь подстраиваться под человека, вместо

Интернет



► Рис. 1. Окно настроек терминала (монохромный режим).

того, чтобы насильно переучивать его под себя. Практически все поведение программы поддается тонкой настройке, и в том числе управляющие клавиши. Более того, настроить их, как и любые другие опции, можно двумя различными способами на выбор: не только традиционным для GNU/Linux текстовым конфигурационным файлом, но и через систему меню в визуальном интерфейсе браузера. Впрочем, к настройкам мы вернемся позже; сейчас скажу лишь, что в интерфейсе браузера все они доступны из меню, по умолчанию вызываемого клавишей Esc. А сейчас поговорим о базовой навигации.

Для выбора ссылок и перехода по ним в *elinks* используется так называемая навигация в стиле *lynx* [*lynx-style navigation*], полностью завязанная на стрелки (клавиши управления курсором). Стрелками вниз и вверх курсор перемещается к следующей и предыдущей ссылке соответственно. Если на экране больше нет ссылок в этом направлении, страница проматывается на один экран вверх либо вниз (как по клавишам PageUp и PageDown), чтобы отобразить соседнюю ссылку, на которую и устанавливается курсор. Стрелка вправо (как, впрочем, и клавиша ввода) используется для перехода по ссылке (то есть «вперед»). А стрелка влево – для возврата на предыдущую страницу («назад»). Это как раз те привязки, от которых я вам не советую отказываться, даже если поначалу они покажутся неудобными: при длительном (читай: привычном) использовании что-либо более сподручное придумать непросто. Можно также отредактировать вручную URL

текущей страницы по клавише G (Shift+g) или выбранной ссылке – E (Shift+e).

Прокрутка по чуть-чуть тоже доступна, по умолчанию она назначена сразу на две пары клавиш: Ins/Del и Ctrl-p/Ctrl-n. Здесь задействована удобная особенность настройки клавиш в *elinks*: каждое действие может быть привязано одновременно к любому количеству управляющих клавиш. При желании вы можете как назначить новые клавиши для короткой прокрутки, так и изменить количество строк, на которые будет прокручиваться страница за одно нажатие.

Вполне возможно, что по умолчанию *elinks* запустится у вас в монохромном режиме. Это легко исправить: в большинстве дистрибутивов он собран с поддержкой 256 или по крайней мере 88 цветов. Даже если авторы вашего дистрибутива не удосужились собрать *elinks* «достаточно цветным», а вам неохота возиться с его пересборкой, 16 цветов должны быть доступны всегда. Достаточно выбрать цветовую схему в настройках, вызвав их из меню: Настройка > Настройки терминала (рис. 1). После этого можно сохранить настройки кнопкой Save в нижней части окна. Обратите внимание: настройки сохраняются отдельно для каждого типа терминала (то есть каждого значения переменной \$TERM). Это позволяет иметь при необходимости отдельные настройки для каждого эмулятора терминала (если вы их используете несколько) и отдельно для голой консоли (если вы ее используете вообще). Цветной *elinks* будет выглядеть примерно как на рис. 2. Возможно, вы не привыкли к светлому фону в терминале – на этот предмет у *elinks* есть специальная опция: помимо полного отображения цветов, заданных самой web-страницей, доступен вариант

Родословная героя

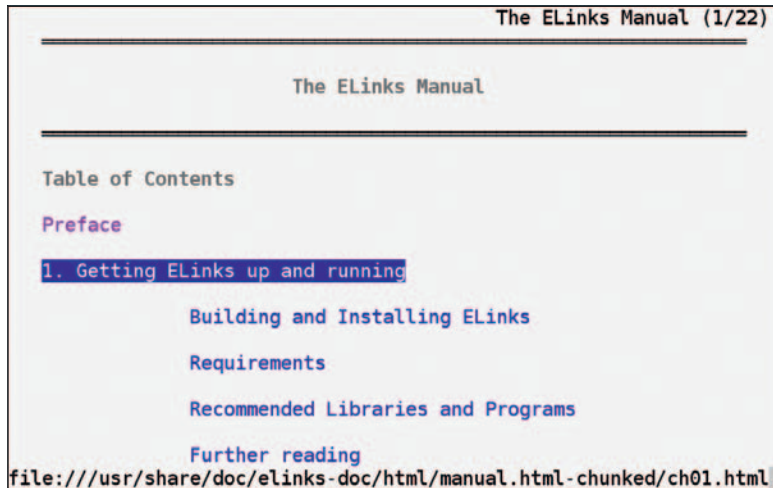
Родоначальником семейства консольных браузеров считается браузер *lynx*. Название его – своеобразный каламбур. С одной стороны, оно созвучно со словом links, то есть ссылки. С другой, «lynx» по-английски означает «рысь», что намекает на его «рысью стремительность», то есть скорость работы. По мере развития функционала графических браузеров эта скорость осталась, пожалуй, единственным весомым преимуществом *lynx*, которое перевешивало все большим и большим числом недостатков.

Со временем идеи *lynx* получили развитие в новом проекте с более незамысловатым названием *links*. Он отличался от родителя более широкими функционалом (в частности, поддержкой таблиц

и фреймов) и возможностями настройки. Однако со временем и этого стало не хватать, и от *links* отпочковались примерно в одно время сразу два проекта.

Links2, впоследствии получивший развитие как «links hacked», приобрел даже способность графического отображения web-страниц в отдельном режиме, равноправном с оригинальным текстовым. *A elinks* пошел по другому пути – создания максимально удобного и функционального чисто текстового браузера. Позже некоторые наработки стали портировать из *links2* в *elinks* и обратно. Но затем развитие *links2/links-hacked* практически прекратилось, а *elinks*, таким образом, стал основной линией развития текстовых браузеров в целом.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)



► Рис. 2. Elinks с умолчательными «цветными» настройками.

«кроме цвета фона», как и полное игнорирование этих установок и отображение всего умолчательными цветами. Установка опций вызывается пунктом меню Настройка > Options manager (увы, интерфейс браузера на русский язык переведен лишь частично), либо клавишей **o** (по умолчанию).

А за вышеописанные варианты отображения цветов отвечает опция Document > Default color settings > Use document-specified color [Документ > Настройка цветов по умолчанию

> Использовать цвета, указанные в документе]. Значение по умолчанию (все цвета из документа, включая цвет фона) – 2; цвета документа, кроме фона – 1; все цвета из настроек браузера – 0.

Но вернемся к навигации по сайтам. Пока мы говорили только о прямых переходах по ссылкам и возврате по истории. Но мало кто в наше время может представить полноценную навигацию без таких вещей, как...

Вкладки и закладки

Конечно же, *elinks*, как и любой современный браузер, умеет работать со вкладками. По умолчанию активную ссылку можно открыть в фоновой вкладке клавишей **t** (от слова “tab” – «вкладка»). А по нажатию **T** (т.е. Shift+t) откроется новая вкладка без перехода по ссылке, по умолчанию сразу с окном для ввода нового адреса. Функция открытия ссылки в новой активной вкладке, конечно же, тоже присутствует, но по умолчанию клавиша

ей не назначена. Видимо, разработчики подразумевают такой стиль работы с браузером, когда эта функция почти не используется: ссылки «на потом» открываются в фоновых вкладках, а «на сейчас» – в этой же, с последующим возвратом назад по истории. В чем-то этот подход оправдан: у *elinks*, в отличие от популярных графических браузеров, есть, помимо кэша, еще и специальная область в оперативной памяти, в которой временно хранятся уже отрисованные страницы. Поэтому возврат на предыдущую страницу происходит, условно говоря, даже быстрее, чем нажатие на клавишу «влево».

Если же умолчательный стиль навигации вам почему-либо неудобен, это как раз тот случай, когда можно перейти к настройке управляющих клавиш – Настройка > Keybinding manager или клавиша **k** (по умолчанию). В этом окне (как, впрочем, и в окне опций) имеется кнопка Поиск. Попробуйте, развернув пункт Main mapping, нажать эту кнопку и ввести в появившейся строке “tab” [вкладка]. Теперь найти нужный пункт будет совсем просто: “Open the current link in a new tab [Открыть текущую ссылку в новой вкладке]”.

Функция создания новой вкладки в *elinks* устроена не совсем обычно. Она добавляет не пустую вкладку, а с копией текущей открытой страницы. Это позволяет использовать вкладки вместо временного маркера на странице. Например, вы читаете некую страницу подряд, и тут вам понадобилось посмотреть что-то в самом ее начале или в конце, или найти нужную фразу. Создайте новую вкладку;

«Можно открывать и по несколько временных вкладок сразу.»

перейдите к нужному месту или воспользуйтесь поиском; а затем закройте вкладку – первоначальная все это время будет оставаться ровно на том же месте, где вы ее покинули.

Можно, разумеется, открывать и по несколько таких временных вкладок сразу.

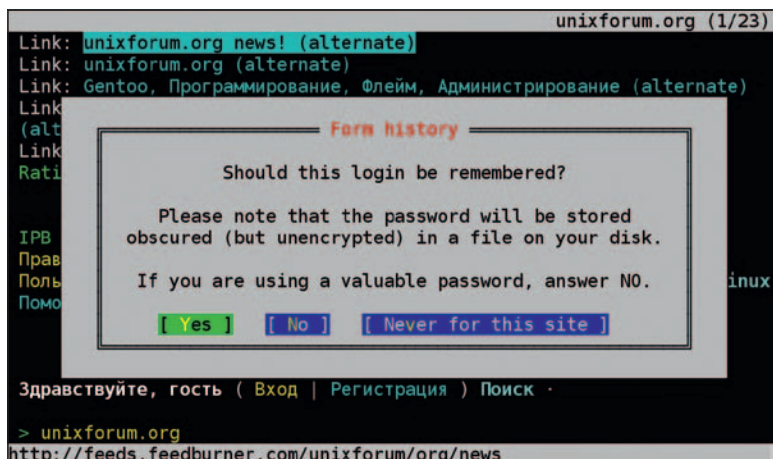
Вообще, не бойтесь создавать много вкладок: *elinks* весьма нетребователен к ресурсам по сравнению с графическими браузерами, и даже с несколькими десятками (а то и сотнями) открытых вкладок не занимает критичного для нынешних машин объема памяти. Перемещаться между открытыми вкладками можно с помощью клавиш **<** и **>** (т.е. Shift+ и Shift+.), а передвигать сами вкладки вправо и влево – клавишами Alt+**<** и Alt+**>** (Alt+Shift+ и Alt+Shift+.). Закрывать текущую вкладку позволит клавиша **c** (от “close” – «закрывать»).

Создав несколько (а тем более много) вкладок, наверняка захочется запомнить, какие именно страницы в них открыты. Чтобы добавить страницу в закладки, нажмите **a** (от “Add bookmark” – «добавить закладку»). По клавише **A** (Shift+a) можно добавить в закладки ссылку, на которой стоит курсор. В обоих случаях будет предложено отредактировать только имя закладки и URL; упорядочение закладок доступно из менеджера закладок, вызываемого клавишей **s** (здесь нет смысловой расшифровки; клавиша используется из-за соседнего расположения с **a** на стандартных клавиатурах).

Еще одна функция, связанная с закладками, не имеет привязанной клавиши по умолчанию, но я вам советую сразу же это исправить, т.к. на практике эта функция очень удобна – она добавляет в закладки сразу все открытые в данный момент вкладки в виде отдельной группы, имя которой можно задать в момент добавления. Открыв несколько вкладок по сходной тематике, вполне резонно добавить их в закладки именно так.

Кроме ручного добавления и редактирования посредством менеджера закладок, закладки используются еще для одной за-

► Рис. 3. Окно запроса на сохранение пароля.



» Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

Дальний родич

Существует параллельная ветвь развития текстовых браузеров. Точнее, даже не ветвь, а одно-единственное звено – браузер *w3m* от японских разработчиков. Его история ненамного короче *lynx/links*: разработка *lynx* началась в 1992 году, а разработка *w3m* – в 1995. Но по возможностям он во многом уступает *elinks*. В частности, у *w3m* вообще нет кэша; нет и возможности фоновой загрузки страниц, то есть пока страница загружается и отрисовывается, переключиться на другую вкладку или создать новую нельзя. Это часто бывает достаточно неприятно, особенно на медленных соединениях. Нет в *w3m* и столь гибкой настройки (например, для переназначения клавиш нужно перекомпилировать браузер), и некоторых из расширенных возможностей *elinks*.

дачи – вместо «менеджера сеансов [session manager]». По умолчанию все эти настройки выключены, а включить их можно в разделе опций User interface > Sessions. Здесь доступно то, что должен уметь любой уважающий себя современный браузер: автоматическое сохранение сеанса при выходе и загрузка при запуске браузера (Auto save session и Auto restore session). Кроме того, есть опция периодического сохранения сеанса на случай некорректного завершения работы браузера или системы (Periodic snapshotting). По умолчанию первая опция сохраняет «сеанс» под именем “Auto saved session”, а вторая – “Session snapshot – <дата>, <время>”. Автосохранение будет перезаписываться, а «снимки [snapshots]» придется время от времени удалять вручную.

На работе с окном закладок подробно останавливаться не будем – там все достаточно очевидно. Добавим только, что во всех подобных окнах (опции, назначение клавиш, cookie manager и т.д.) стрелки вверх-вниз перемещают курсор по пунктам списка, влево-вправо – по управляющим клавишам, а пробел раскрывает и скрывает подпункты в списке.

Настройка и расширенные возможности

Как уже упоминалось, *elinks* поддерживает два способа настройки: в интерфейсе браузера, через пункты Options manager, Keybinding manager и Настройки терминала в меню «Настройка»; и посредством редактирования конфигурационного файла `~/.elinks/elinks.conf`. Советуем воспользоваться обоими способами.

Первый хорош для первоначальной настройки под свои предпочтения: пройтись «от и до» по всем пунктам и подпунктам всех меню и все «оттюнинговать» на свой вкус (не забывая сохранять результаты «тюнинга» кнопкой Save [Сохранить] в тот самый `elinks.conf`). Второй предпочтительнее для внесения точечных правок в процессе работы: поиск по спискам меню в интерфейсе присутствует, но он не столь удобен и гибок, как возможность поиска по текстовому файлу средствами достаточно продвинутого редактора или традиционных утилит командной строки.

Все настройки щедро комментированы, причем комментарии присутствуют как в меню, так и в текстовом конфиге. Увы, на русский они (пока?) не переведены, так что если нет знания английского хотя бы на самом базовом уровне (которого для прочтения комментариев вполне достаточно), Google Translate вам в помощь.

Останавливаться на конкретных пунктах настройки мы не будем, поскольку, как уже говорилось, политика *elinks* – не «быть хорошим для всех» (что пока удавалось, как известно, только червонцу), а «предоставить всем возможности сделать все на свой вкус». А предпочтения у каждого свои.

Первую из еще не упомянутых возможностей, пожалуй, расширенной можно назвать разве что с некоторой натяжкой: в самых популярных браузерах она присутствует уже давно. Речь об автоматическом сохранении логинов и паролей к различным сайтам. При входе по логину и паролю на любой сайт, для которого именно эта пара логин/пароль еще не сохранена, появится окно с вопросом, нужно ли сохранять ее сейчас (рис. 3), с тремя вариантами ответа: сохранить, не сохранять сейчас, не сохранять никогда для этого сайта. При сохранении и логин, и пароль шифруются.

Хранится все это добро в файле `~/.elinks/formhist` (как и все остальные файлы в этом каталоге, он текстовый), а в интерфейсе доступно по пункту меню Tools > Form history. При желании любой сохраненный логин в каждом из этих двух мест можно удалить. Удаляя в интерфейсе, не забывайте опять-таки нажимать кнопку Save.

Одна из самых удобных возможностей, редко встречающихся в других браузерах, в то же время и самая простая – даже, как на первый взгляд может показаться, примитивная: это отключение таблиц. Действительно, зачем оно, казалось бы, в реальной жизни нужно? Но на практике при работе с текстовой информацией такое зачастую оказывается незаменимой штукой, поскольку позволяет «избавиться» от так называемой табличной верстки множества сайтов – к примеру, от боковых колонок с разнообразными ссылками, которые, видимо, по мнению авторов сайтов страсть как нужны при чтении основной информации; а на деле только мешают. В лучшем случае, они отнимают место по краям, оставляя для основного текста узкую полоску. В худшем, если сайтостроители не озаботились подгонкой по ширине окна или реализовали ее неправильно, основной текст, наоборот, растянется шире чем нужно, вынуждая к горизонтальной прокрутке туда-обратно на каждой строчке. Отключите таблицы (по умолчанию клавиша не назначена, но назначить ее можно, как и на любое другое действие) – и левая колонка «уйдет» в самый верх, а правая в самый низ; а основной текст впишется ровно в ширину окна.

О возможностях *elinks* можно говорить долго, и размеров статьи для этого, конечно, недостаточно. Но еще одну возможность хотя бы вкратце упомянуть все же стоит. Называется она “URI rewriting” (дословно: «переписывание ссылок») и настраивается в меню опций или в файле `elinks.conf`.

Раздел, отвечающий за эту возможность, делится на два подраздела: “Dumb prefixes” (дословно: «тупые (или глупые) префиксы») и “Smart prefixes” (дословно, «умные префиксы»).

Первые – это, по сути, текстовый вариант закладок: каждый из них сопоставляет некому буквосочетанию одну конкретную ссылку. Например, можно задать, что при вводе в строке адреса букв “lxf” вы будете автоматически попадать на сайт lxfformat.ru. Со вторыми все намного интереснее (недаром их назвали «умными»). Здесь можно подставлять в адрес страницы «аргументы» префикса. К примеру, заданный в умолчательном конфиге «умный» префикс ‘g’ превратит строку “g набор слов” в `http://www.google.com/search?q=набор+слов`, т.е. попросту найдет заданную фразу в Google.

Аналогично можно задать «умные» префиксы для других поисковых систем, википедии, онлайн-словарей (к примеру, lingvo.yandex.ru), поиска по форумам... да практически для чего угодно. Для этого достаточно один раз выполнить нужный поиск вручную, а затем подставить в полученную ссылку шаблон ‘%s’ вместо конкретного текста. **LXF**

«Можно задать “умные” префиксы практически для чего угодно.»

ОТВЕТЫ

Есть вопрос по открытому ПО? Пишите нам по адресу answers@linuxformat.ru, и мы найдем ответ.

В этом месяце мы ответим на вопросы про...

- 1 Ошибки в скрипте на LXF DVD
- 2 Воспроизведение файлов WMA посредством MPD
- 3 Перезапись винила на компакт-диск
- 4 Утрату соединения картой Ethernet
- 5 Просмотр EXIF в Gwenview
- 6 Banshee и Ubuntu
- 7 Загрузку с USB-брелка

1 DVD2USB.sh

В Я запустил этот скрипт с LXF DVD в терминале и через Alt+F2 после размонтирования своего USB-диска размером 8 ГБ. Ничего не произошло, снова появилась командная строка, без возможности Select a USB drive to copy the DVD to: [Выберите USB-диск, куда нужно скопировать DVD]. Я заглянул в скрипт, надеясь понять, как вручную запустить команду, которая сделала бы то же самое. Поискал информацию на сайте LXF (по строке "dvd2usb.sh") — и все равно ничего.

Моя единственная альтернатива (поскольку я их скачал) — выбрать дистрибутивы для более старого «железа»: LM LXDE, Bodhi, WattOS, Wary, Puppy, Tiny Core с мультизагрузкой. Пожалуй, я создам мультизагрузочный USB-диск с дистрибутивами.

Майк Мур [Mike Moore]

Это ошибка скрипта, точнее, автора скрипта. Он счел, что программа *dialog*, на которой основан скрипт, является стандартной для большинства дистрибутивов Linux. Это не так, хотя она была во всех системах, в которых проверялся скрипт. У Вас есть два варианта: установить *dialog* из менеджера пакетов своего дистрибутива или загрузить один из Live-дистрибутивов с DVD и запустить скрипт в нем. Первый вариант — самый простой. Я отправил Бену модификацию скрипта с проверкой, что команда *dialog* существует, а если ее нет, скрипт выводит совет по решению проблемы и завершается. В будущем DVD такой проблемы быть не должно.

2 MPD и WMA

В Немного отставая от жизни, я успел прочесть только LXF149. Меня привлекли возможности MPD, так как я часто работаю на ноутбуке, слушая музыку, которая воспроизводится с мультимедийного компьютера, подключенного к системе AV — и было бы здорово управлять воспроизведением с ноутбука.

Я установил MPD на мультимедийный компьютер и Aigo на ноутбук, и — вуаля! Все получилось! Потом я заметил, что в базе данных MPD

отсутствует почти вся моя музыкальная коллекция — большая ее часть хранится в формате WMA с варьируемым битрейтом, с тех времен, когда я жил в аду *Windows MCE*. MPD не хочет добавлять файлы этого формата в свою базу данных, только MP3 и Ogg Vorbis.

Запустив MPD --version, я увидел в списке поддерживаемых форматов WMA — так почему же он не добавляется в базу данных? В Интернете мне посоветовали добавить "ffmpeg" в файл *mpd.conf*, но не уточнили куда. В том месте, где я пробыл (под разделом ALSA), это не сработало. Был бы признателен за помощь, потому что сейчас пользы от MPD не больше, чем от козла молока! На мультимедийном ПК установлена Mythbuntu 10.04 LTS, а на ноутбуке — Ubuntu 10.04 LTS.

Эндрю Уильямс [Andrew Williams]

Похоже, способность MPD проигрывать файлы WMA напрямую зависит от параметров, с которыми он был собран. В версии 0.16.8 эти файлы воспроизводятся без дополнительных действий. В Mythbuntu 10.04 еще используется версия MPD 0.15.4, она довольно старая. Возможно, стоит лишь обновиться до более поздней версии, и проблема исчезнет сама собой.

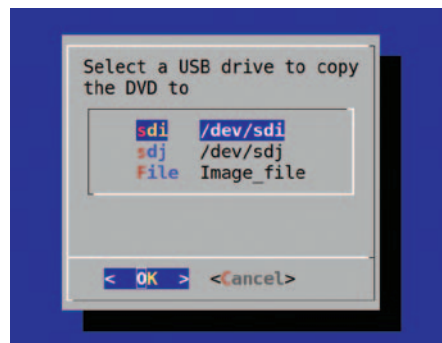
Модуль расширения *ffmpeg* используется для обработки файлов, с которыми MPD не справляется внутренними кодеками — для этого и нужна запись в файле настройки. Ее корректный синтаксис таков:

```
decoder {
  plugin "ffmpeg"
  enabled "yes"
}
```

Добавьте эти строки в */etc/mpd.conf*. Точное положение не играет роли, но нельзя помещать их внутрь других блоков, потому-то вариант с ALSA и не сработал. Я предпочитаю добавлять новые настройки в начало или в конец файла отдельно от других настроек, чтобы их проще было найти. Также стоит добавить комментарий, который поможет вспомнить, почему Вы добавили эти параметры, месяцев через девять, когда Вы в следующий раз откроете этот файл.

```
#### Добавление поддержки воспроизведения
файлов WMA в ffmpeg ####
decoder {
  plugin "ffmpeg"
  enabled "yes"
}
```

И последнее предупреждение относится к *ffmpeg*. Хотя это мощная и быстрая программа,



Если нужно скопировать DVD на USB, убедитесь, что установлен пакет *dialog*.

у ее разработчиков есть привычка постоянно менять API, что выводит из строя программы, ею пользующиеся. Поэтому стоит устанавливать все программы, которые используют *ffmpeg*, и саму *ffmpeg* из одного и того же репозитория.

3 Долгоиграющий Linux

В Недавно я установил Linux Mint и полностью удалил XP — теперь компьютер работает гораздо лучше. Единственное разочарование — теперь я не могу пользоваться *Vinyl Studio* от Alpine Software, программы, с которой очень удобно конвертировать винил в компакт-диски. VLC и некоторые другие программы — не того класса. Можете ли вы подсказать эквивалент?

Терри Бэйл [Terry Bale]

А Вы не пробовали *Audacity* (<http://audacity.sourceforge.net>)? Это — почти самый популярный аудиоредактор для ОС Linux (есть версии и для MacOS и Windows). В *Audacity* можно записать звук с любого источника и затем обработать его — например, удалить щелчки и треск, которые часто слышны при проигрывании виниловых записей. В руководстве подробно описан процесс копирования записей с магнитофонных пленок и пластинок на компакт-диски; это именно то, что Вам нужно. В руководстве освещаются все необходимые вопросы: подключение оборудования, установка уровней, разделение записи на треки, удаление шума и окончательный экспорт записи на компакт-диск.

Проблемы могут возникнуть только с устройствами. Вы не уточнили, чем пользуетесь, но Вам понадобится звуковая карта, которая поддерживается в Linux — таких большинство. К ней можно подключить проигрыватель точно так же, как и в Windows. *Audacity* также поддерживает USB-проигрыватели; с ними Вам вовсе не придется беспокоиться о звуковой карте (хотя без нее

сложнато будет слушать записи). Вы хотели скопировать музыку на компакт-диск, но Audacity может сохранить ее и в файлах MP3 или Ogg Vorbis. Если на диске достаточно места, лучше всего сохранить треки в формате FLAC. FLAC – кодек без сжатия, поэтому качество записи не изменится, но сжатие самих файлов позволяет сэкономить до 50–60% пространства по сравнению с компакт-диск.

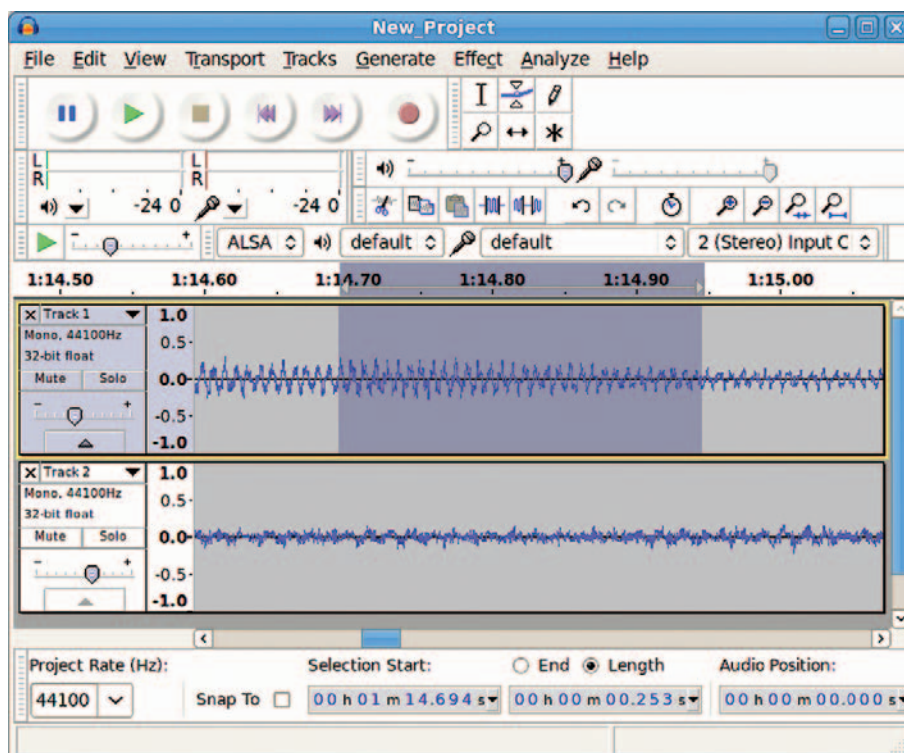
При хранении музыки в формате FLAC ее можно сконвертировать в формат с потерями для портативного плеера, записать на CD или послушать напрямую – это означает, что с одним базовым форматом в Вашем распоряжении будут все возможности.

При установке Audacity через менеджер пакетов дистрибутива будет установлено и руководство. Для доступа к нему воспользуйтесь меню Help [Справка] в Audacity или прочтите его в Интернете перед установкой. Вам нужен раздел http://manual.audacityteam.org/help/manual/man/tutorial_copyping_tapes_lps_or_minidisks_to_cd.html.

4 Склероз у Ethernet

В последних версиях Ubuntu и Mint, начиная с Mint 10, моя сетевая карта страдает электронной версией амнезии и теряет связь, хотя говорит, что все еще подключена. Я связался с производителем контроллера, который сказал, что в Ubuntu и, следовательно, в Mint изменился способ работы с их контроллерами и что я должен использовать статический IP-адрес.

Все это прекрасно, но я пользуюсь Virgin Media Super Hub, который, как мне дали понять, представляет собой замаскированный модуль Netgear и к которому я еще только привыкаю.



» Audacity прекрасно импортирует музыку с винила, магнитофонной пленки и мини-дисков (помните такие?).

Я мог бы смириться с потерей сигнала по проводному соединению и продолжать пользоваться беспроводным Linksys, но у меня большие проблемы и с этим тоже.

Моя система установлена на ноутбуке Toshiba Equium.

Дэвид Киркпатрик [David Kirkpatrick]

Чтобы проверить, в дистрибутиве ли дело, можно загрузиться с Live CD другого дистрибутива – желательно того, с которым Вы работали раньше. Вы не говорите, какая у Вас модель Equium, но во всех, кажется, используются распространенные Ethernet-контроллеры, которые должны просто работать. Описанные Вами симптомы могут говорить об ошибке в работе устройства – либо самого контроллера, либо кабеля. Это также может происходить, когда клиент DHCP пытается синхронизировать свой lease с Вашим маршрутизатором, хотя ни с одним уважающим себя контроллером такого быть не должно.

Так как Virgin Super Hub – полноценный маршрутизатор, а не простой проводной модем, статический адрес задать довольно просто – модемная часть Super Hub все еще получает свой адрес от Virgin через DHCP. Для этого проще всего подключиться по DHCP как обычно, зайти в сетевые настройки (Network Settings) и записать IP-адрес, адрес шлюза и адреса серверов DNS. Теперь выберите использование статических адресов в настройках и снова введите все эти параметры. Можно задать значение IP-адреса, не входящее в диапазон, используемый DHCP-сервером роутера. По умолчанию Super Hub имеет адрес **192.168.0.1** – таким должен быть и адрес шлюза, который Вы записали; поэтому можете воспользоваться **192.168.0.2** – DHCP-серверы обычно используют более высокий диапазон доступных адресов, например, от **192.168.0.100** и выше. Обратите внимание, что менять нужно только последний байт адреса, первые три должны остаться неизменными.

Теперь перезапустите сетевое соединение или перезагрузите систему, и все должно заработать. »



Коротко про...

at

С помощью *Cron* очень удобно запускать команды в заданные моменты времени или через заданные интервалы времени; ну, а если команду нужно запустить однократно, но в заданное время? С помощью команды **sleep** можно дождаться, пока это время наступит –

```
sleep 1h; somecommand
```

но она блокирует терминал, а после перезагрузки ее действие прекращается.

Демон *at* – вариант получше. Он работает как и *cron*, просыпаясь каждую минуту и проверяя, есть ли задачи для запуска; но каждая задача однократна. Вызов **at** задает выполнение некоторой задачи, считав команду для ее запуска из стандартного ввода. Строка

```
echo "somecommand --with-opts" | at 15:00
```

запустит команду *somecommand* в 15:00. Если это время уже прошло, команда будет запущена завтра. Интервалы также допускается задавать

в часах, минутах и днях и даже суммировать их значения:

```
echo "anothercommand --with-opts" | at 5am + 2 days
```

С помощью специального параметра **now** можно указать задержку вместо конкретного времени, и пример со **sleep** переписывается так

```
echo "somecommand" | at now + 1 hour
```

После выполнения этой команды можно закрыть терминал и даже перезагрузить систему, но команда все равно сработает. Для просмотра списка команд в очереди запустите **atq**, а если передумаете запускать команду, то удалить ее из очереди можно командой **atrm**. Более подробное описание параметров для задания времени просмотрите на *man*-странице *at*.

Если нужно запустить несколько команд, запишите их в файл по одной в строке и передайте их **at** так:

```
at midnight <commands.txt
```

Если соединение все еще разрывается, пора подумать о проблеме с «железом». В разных моделях Equium используются различные беспроводные чипсеты, от хорошо поддерживаемого Intel до странного гибрида USB/Mini-PCI, поэтому здесь без дополнительной информации мы ничем помочь не сможем.

5 KDE и EXIF

Вот жизни мне нужно немного — экземпляр LXF, бутылка пива «Курочка Ряба [Old Speckled Hen]» и файловый менеджер, который отображал бы данные EXIF моих фотографий. В старой Windows XP моей жены они отображаются, в KDE3 тоже, а в Dolphin — нет, и даже открытие файлов в Gwenview не помогает. Наверное, дело во мне, или в моей системе, или я пропустил какое-то обновление; команда `kded4 -v` сообщает «KDE Development Platform: 4.8.2 (4.8.2)», и я также приложил вывод команды `hardinfo`.

Кен Уилсон [Ken Wilson]

Gwenview отображает данные EXIF изображений. Если открыть боковую панель из меню View [Просмотр] или клавишей F11, то на средней вкладке будет показана краткая сводка EXIF. Нажатие на ссылку More [Дополнительно] откроет окно, содержащее всю доступную информацию. У каждого элемента в этом окне есть галочка — ею можно пометить те из них, которые нужно показывать в боковой колонке.

6 Ubuntu молчит...

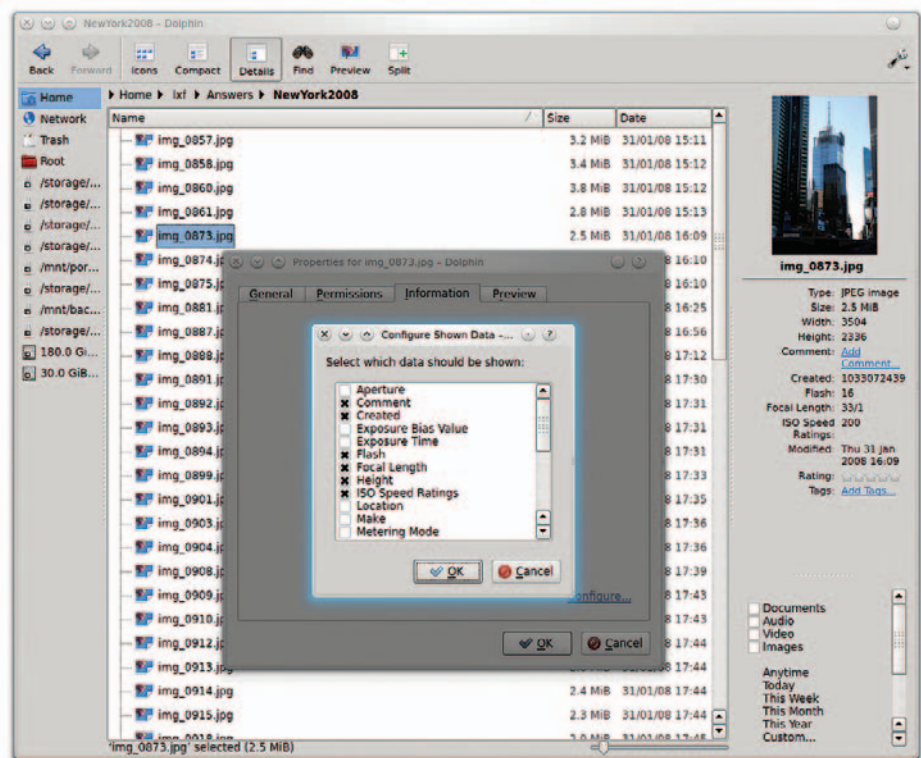
Недавно я обновил Ubuntu до последней версии — 11.10. В меню «Системные настройки [System Settings]» в приложениях говорится, что «в данный момент ни одно из приложений не воспроизводит и не записывает звук [no applications are currently playing or recording audio]», хотя в Центре приложений Ubuntu [Ubuntu Software Centre] в установленных программах есть медиа-проигрыватель Banshee и несколько других программ. Когда я проверяю системные динамики, звука нет — можете помочь?

Кебеде

Сообщение, которое Вы видите на вкладке Приложения [Applications] Настроек звука [Sound Settings], означает только то, что в данный момент ни одно из запущенных приложений не обращается к звуковой системе. Если открыть в Banshee какой-то трек и перейти на эту вкладку, в списке программ появится Banshee.

Проигрыватель по умолчанию и настройки по умолчанию для других типов данных задаются в другом месте.

Нажмите на иконку Информация о системе [System Info] в нижней части главного окна Системных настроек [System Settings], нажмите на Приложения по умолчанию [Default Applications] на левой панели, и Вы сможете выбрать приложения по умолчанию для файлов различных типов.



► KDE будет отображать информацию EXIF для цифровых фотографий, если запущен рабочий стол Semantic. Об отображаемых полях можно узнать в окне Properties [Свойства].

В каждом случае выбирать можно только из установленных приложений. Например, если Вы установите новый музыкальный проигрыватель, его можно будет выбрать как используемый по умолчанию. Если Вы совсем не слышите звука, даже при тестовом воспроизведении, причина в чем-то другом.

Как ни странно, но при установке новой системы в микшере часто отключен звук всех устройств. Кликните по иконке с динамиком в панели меню Экран [Screen] и убедитесь, что Основной источник звука [Master] не отключен, затем вернитесь в звуковые настройки и убедитесь, что уровни выходного сигнала [Output] и оповещений [Alert] установлены в 100% (когда Вы убедитесь, что все работает, их можно будет понизить).

Если ничего не помогает, проверьте настройки каждого устройства в терминале, запустив `alsamixer`. Эта команда позволяет отдельно управлять уровнями каждой части звуковой системы. Выберите выходные устройства с помощью стрелок влево и вправо, измените их уровни стрелками вверх и вниз и выйдите из программы, сохранив настройки, клавишами Alt+q.

Нечто подобное можно сделать и в Dolphin, файловом менеджере KDE. Если рабочий стол KDE Semantic включен в пунктах меню Настройки системы > Внешний вид рабочего стола [System Settings > Workspace Appearance] и Поведение > Поиск по рабочему столу [Behaviour > Desktop Search], теги EXIF должны быть доступными всем приложениям KDE. При подведении мыши к каждому файлу на панели предпросмотра в правой стороне окна Dolphin отображается его миниатюра и базовая информация о нем. Для выбора отображаемой здесь информации щелкните

правой кнопкой мыши на изображении и выберите Свойства [Properties], затем Настройка [Configure] в правом нижнем углу поля Информация [Information]. Здесь можно выбрать отображаемые теги.

Если Вам нужна серьезная программа для работы с коллекцией фотографий, взгляните на Digikam. Это программа для управления фотографиями, которая делает то, что Вам нужно, и гораздо больше. Программа обладает исчерпывающими возможностями по организации коллекции фотографий и поиску по ней, особенно если Вы будете пользоваться тегами и выставлять фотографиям оценки. С помощью системы рейтингов можно быстро избавиться от менее интересных изображений во вновь импортированной коллекции. Теги (пусть на их освоение и требуются некоторые усилия) помогут Вам сэкономить массу времени при поиске изображения.

7 Загрузка с брелка

Недавно я экспериментировал с загрузкой системы с карты CF. С одной и той же картой я попробовал загрузиться с SystemRescueCD 2.0 и затем с Puppy Linux 5.2. Приоритет устройств при загрузке у меня таков: съемные устройства, CD-ROM, жесткий диск. SystemRescueCD определил карту как съемное устройство, и загрузка продолжилась без вмешательства пользователя. Но в Puppy Linux карта была опознана как жесткий диск USB, и дистрибутив не загрузился, после чего контроль перешел к жесткому диску и был запущен Kubuntu.

Если при загрузке нажать F12, BIOS открывает менеджер загрузки, из которого можно выбрать карту CF, отображаемую как второй жесткий диск,

Помогите нам помочь вам

Ежемесячно мы получаем несколько писем, на которые не в состоянии ответить, так как проблема описана в них недостаточно полно. Чтобы дать вам наилучший ответ, нам нужно знать как можно больше.

Если у вас появляется сообщение об ошибке, приведите его точный текст и опишите конкретные условия, когда оно появляется. При возникновении проблемы с устройствами перечислите нам все установленные устройства. Если Linux уже запущен, можете применить для этого отличную программу *Hardinfo* (<http://hardinfo.berlios.de/>) – она сохранит подробную информацию об устройствах и о состоянии системы в HTML-файл, который вы сможете приложить к своему письму. Альтернативный и не менее удобный вариант – *lshw* (<http://ezix.org/project/wiki/HardwareLiSter>). Одна из этих программ должна быть включена в ваш дистрибутив (а то и обе).

Если вы не хотите или не можете их установить, выполните следующие команды в терминале от имени root и приложите файл **system.txt** к письму. Это здорово поможет диагностике.

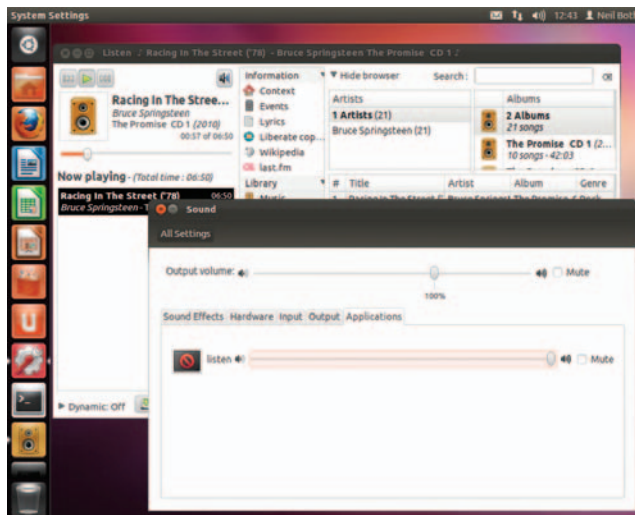
```
uname -a >system.txt
lspci >>system.txt
lspci -vv >>system.txt
```

и после этого Pirru загружается. Приоритет загрузки нельзя установить на конкретный жесткий диск. Почему одна и та же карта определяется по-разному, и можно ли этим управлять?

Питер Уотсон [Peter Watson]

Для загрузки с карты Вы пользуетесь адаптером CF на IDE или SATA? Если да, попробуйте воткнуть его в другой порт. Для дистрибутива здесь не должно быть разницы только потому, что BIOS решил (или ему сказали), с какого устройства загружаться, до того, как загрузка дистрибутива еще не началась. Единственное, что могу предложить – загрузиться с настоящего диска SystemRescue (или Pirru Linux) и затем выбрать из его меню загрузку с жесткого диска.

Если Вы устанавливали Pirru на карту CF как на жесткий диск, поведение будет зависеть от того, в каком порядке жесткие диски распознаются Pirru и где он разместил загрузчик. Я бы попробовал переключить кабели, так что даже если карта CF распознается как жесткий диск, она



На вкладке Applications что-то отображается только в том случае, когда проигрывается звук.

будет первым жестким диском в системе. Перед установкой системы на карту можно даже отключить внутренний жесткий диск – либо физически, либо в BIOS – так загрузчик точно установится на карту.

Раньше я пользовался картами CF в адаптерах IDE для создания тихих клиентов *MythTV* и могу подтвердить, что это темпераментные создания, которые ведут себя ни как полноценные съемные, ни как полноценные внутренние устройства, хотя они должны быть несъемными, поскольку карту нельзя вставить в такой адаптер. **LXF**



Часто задаваемые вопросы

Компиляция ядра

» Ядро у меня есть, зачем мне компилировать новое?

Ядра, поставляемые с дистрибутивами, поддерживают большой набор устройств, но не все. У вас может появиться необходимость добавить поддержку своего устройства. Кроме того, при компиляции исходников с kernel.org вы, как правило, получаете более свежую версию ядра.

» Где находятся исходники ядра?

Стандартные исходники ядра можно загрузить с www.kernel.org или найти на стандартном диске месяца. В большинстве дистрибутивов исходники ядра подлатаны, и некоторые из их утилит основаны на этих заплатках. Пакет исходного кода, соответствующий текущему ядру, можно найти и загрузить

с помощью утилит управления пакетами дистрибутива. Ищите пакет с именем вроде **kernel-source** или **linuxsources**. Совместимые версии обычно можно найти в тестовом репозитории дистрибутива.

» И какие лучше взять?

Обычно лучший выбор – исходники с заплатками для вашего дистрибутива. Если вы планируете применять собственные заплатки, возможно, лучше будет получить исходники с kernel.org.

» Как скомпилировать и установить ядро?

Чтобы установить ядро из архива **kernel.org**, загрузите его и выполните следующие команды от имени суперпользователя-root:

```
cd /usr/src
```

```
tar xjf /path/to/downloaded/linuxx-y-z.bz2
rm -f linux
ln -s linux-x.y.z linux
```

Здесь **x.y.z** – версия ядра, скажем, **2.6.28**. При установке из пакета дистрибутива эти шаги сделают за вас. Затем выполните команды:

```
cd /usr/src/linux
make menuconfig
```

для открытия меню настройки. У большинства параметров вполне разумные значения по умолчанию, но если бы вы хотели остаться с ними, вы бы в это меню не полезли. Пройдитесь по параметрам и установите их в нужные значения. Для поиска конкретного параметра воспользуйтесь поиском в меню ядер серии 2.6. Нажмите / и наберите поисковое слово. Будут показаны соответствующие параметры и способ их установки. Если вы предпочитаете графический интерфейс, воспользуйтесь командой **make xconfig**. Закончив, выйдите из меню

и сохраните настройки. Теперь пора скомпилировать и установить ядро.

» А это что делается?

Для начала, если у вас есть отдельный раздел **/boot**, убедитесь, что он смонтирован. Затем скомандуйте

```
make
make modules_install
make install
```

или объедините эти строки в одну:

```
make modules_install install
```

Первая команда компилирует ядро и модули, вторая устанавливает модули, третья устанавливает само ядро в **/boot**. Ядро также связывается символической ссылкой с **/boot/vmlinuz**, а предыдущее ядро – с **/boot/vmlinuz.old**. Если в меню загрузчика есть два этих пункта, вы всегда сможете выбрать между новым и старым ядром, и это жизненно необходимо, если новое ядро не загрузится. Команда **install** также делает резервную копию конфигурации ядра в **/boot**.

LXF HotPicks



Ник Вейч

Ника однажды укусил радиоактивный менеджер скачиваний, и с тех пор у него болезненная страсть к экспериментам с установкой программ. И дурное пищеварение.

KTorrent » Monit » Gnoduino » YAGF » Pogo » Gnac » Marble » Asteroid » Cube Trains » Fotoxx » Brindump

Клиент Torrent

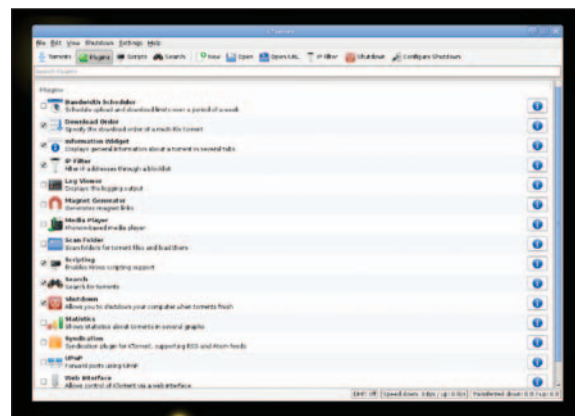
KTorrent

Версия 4.2.0 Сайт <http://ktorrent.org>

Грустно сознавать, что медиа-машина массового потребления, похоже, выставляет перед публикой скачивание через торренты чуть ли не вооруженным нападением с целью отъема ценностей. Но, конечно, на самом деле причин воспользоваться торрент-клиентом полно, и среди разных типов таких клиентов можно выбрать достойный.

Для простой, бездумной работы трудно превзойти *Transmission*, или, возможно, *Miro* для медиа-контента. Но если вы действительно хотите управлять работой с торрентами, стоит оглядеться вокруг попристальнее – возможно, рассмотреть нечто вроде *KTorrent*, который всегда предлагал множество настраиваемых функций и модулей расширения.

Если многочисленные кнопки и опции вызывают у вас немотивированную агрессию, то вам лучше держаться рабочего стола *Unity* и *Transmission*.



» **Всего лишь малая часть массы расширений KTorrent – того, что дает ему превосходство.**

Если вас греет мысль о том, что вы сможете управлять пропускной способностью канала, блокировать неактивные пиры, собирать статистику по соотношению закачек и раздач, автоматически отключаться по завершении скачивания, настраивать целевую раздачу и делать еще массу разных вещей – короче говоря, если вы хотите управлять своим торрент-клиентом, а не позволять ему решать за вас, что вам лучше, то ваш ответ – *KTorrent*.

Стоит также отметить, что обилие дополнительных опций – не такое уж тяжелое осложнение. Все параметры по умолчанию выбраны очень здраво, и *KTorrent* можно просто установить и запустить, не заботясь ни о чем, если у вас нет желания возиться с настройками. Он покажет, на что способен, когда вы решите создать и/или раздавать собственные торренты.

Как вы, возможно, уже смекнули по названию, *KTorrent* – приложение KDE, и оно прекрасно интегрируется в рабочий стол KDE. Оно вполне благополучно работает и с Gnome, если вы вдруг решите сделать нечто безумное – например, выделять и перетаскивать свои торрент-файлы вместо того, чтобы скачивать их у запрашивающего.

KTorrent, наверное, по-прежнему самый полнофункциональный и надежный торрент-клиент для Linux. Вам стоит познакомиться с ним – пока его не запретили или не подвергли остракизму.

«Обилие добавочных опций – не такое уж тяжкое осложнение.»

Исследуем интерфейс KTorrent

Расширения

Ktorrent очень активно использует модули расширения для включения добавочных функций, вроде графика закачек или блокировки IP.

Выключение

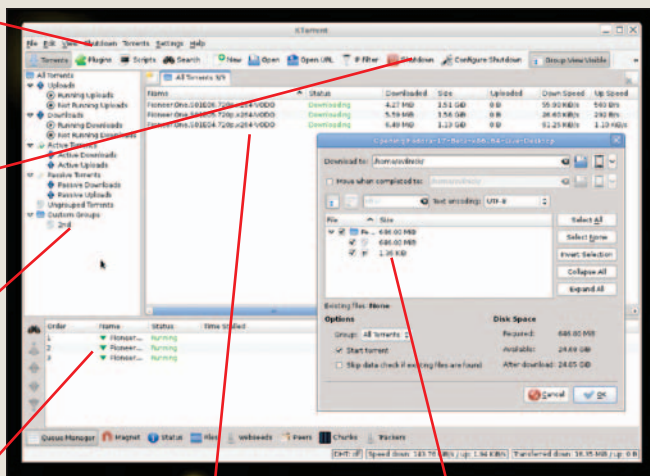
Вы можете настроить программу, чтобы она выключала систему по завершении текущих задач.

Боковая панель

В боковой панели имеются группы и древовидная структура, и даже если у вас масса торрентов, это все равно управляемо.

Нижняя панель

Дополнительная информация, например, списки файлов и соединения для выбранного торрента, отображается здесь.



Активные торренты

Окно Active Torrent отличается информативностью и при этом легко читается.

Опции торрента

При каждом добавлении торрента, вы можете настроить опции именно для этих файлов.

Монитор системы

Monit

Версия 5.4 Сайт <http://bit.ly/JSMghh>

Целую вечность назад, когда я еще был редактором всяких там журналов, у нас возникла проблема: рекламодатель отказался от полосу. Я в тот момент на месте отсутствовал, и по возвращении в офис меня ждал неприятный сюрприз. Однако у моего подручного, поведавшего мне о катастрофе, был туз в рукаве. «Мы тут без вас собрали полосу, и по-моему, ее можно пустить вместо отказной. Все уже проверено и все такое, вам просто нужно ее утвердить». Классная работа. Кризиса не случилось.

Этот рассказ об эффективной работе я привел здесь вовсе не с целью тешиться воспоминаниями, а как метафору того, на что способен *Monit*. Программы для мониторинга все хороши. Есть десятки отличных инструментов, которые выведут на экран предупреждение о переполнении базы данных, или порадуют вас среди ночи сообщением, что хранящиеся на сервере данные превысили квоту. *Monit*

следит и за этим, но его козырная карта — умение автоматически устранять многие проблемы.

Подробности хранятся в его конфигурации (настраиваемой). Вот пример из руководства пользователя (очень подробного):

```
check process resin with pidfile /usr/local/resin/srun.pid
start program = "/usr/local/resin/bin/srun.sh start"
stop program = "/usr/local/resin/bin/srun.sh stop"
```

Этот простенький скриплет находит процесс (**resin**) и проверяет, работает ли он. Если нет, пытается его запустить. Встраивая сюда все больше проверок посредством имеющихся инструментов

«Его козырная карта — умение автоматически устранять проблемы.»

```
trilobite:monit martinpalos$ sudo ./monit -c /etc/monitrc_screenshots -h
Usage: monit [options] [arguments]
Options are as follows:
-c file          Use this control file
-d n            Run as a daemon once per n seconds
-g name        Set group name for start, stop, restart, monitor and unmonitor
-l logfile     Print log information to this file
-p pidfile     Use this lock file in daemon mode
-s statefile    Set the file monit should write state information to
-I            Do not run in background (needed for run from init)
-t            Run syntax check for the control file
-v            Verbose mode, work noisy (diagnostic output)
-vv           Very verbose mode, same as -v plus log stacktrace on error
-H [filename] Print SHA1 and MD5 hashes of the file or of stdin if the
              filename is omitted; monit will exit afterwards
-V            Print version number and patchlevel
-h            Print this text

Optional action arguments for non-daemon mode are as follows:
start all      - Start all services
start name     - Only start the named service
stop all       - Stop all services
stop name      - Only stop the named service
restart all    - Stop and start all services
restart name   - Only restart the named service
monitor all    - Enable monitoring of all services
monitor name   - Only enable monitoring of the named service
```

Из командной строки устанавливается масса параметров, но удобнее предусмотреть настраиваемый файл `config`.

и синтаксиса, его деятельность можно расширить до весьма специфических ситуаций, таких, как перезапуск процесса на сервере при превышении лимита загрузки CPU и расхода памяти, или отключение web-сервера при слишком большом количестве подключений. *Monit* не ставит вас перед фактом наличия проблемы, а предоставляет вам язык для описания определенных проблем и способы их разрешения.

IDE Arduino

Gnoduino

Версия 0.4.0 Сайт <http://bit.ly/Ku9TPN>

Arduino = хорошо. Если вашу душу не влекут таинства тщательного и выверенного создания и разработки собственных электронных гаджетов, то у вас, чего доброго, и души-то нет. И вам стоит прочесть превосходную во всех отношениях серию **LXF**, посвященную Arduino во всей полноте. Arduino ну очень хорош.

Ну, почти хорош. Чуть ли не превосходен. Но есть одна проблема: IDE Arduino. Будем справедливы: со времен первых версий она значительно улучшилась, и текущий релиз 1.0 с современными библиотеками уже почти нормальный.

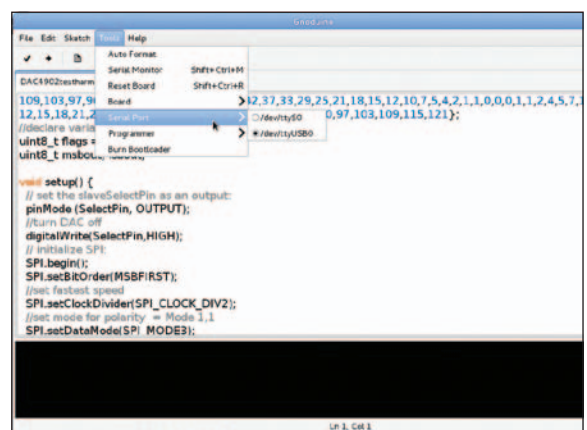
Ее основная проблема, стыдно сказать, в том, что она... написана на Java. Хм. Это само по себе не криминал, но влечет смешение хорошего и дурного. Несовпадения запросов файлов ликвидированы, но по-прежнему есть трудности с употреблением системных шрифтов, и часто возникает проблема переговоров с устройствами Linux — такими, например, как последова-

тельный порт. Или с определением смены устройства USB. Но все это присуще не самой программе, а внутренним библиотекам Java.

Gnoduino являет собой попытку воссоздания IDE Arduino как более эффективной и дружелюбной к Linux. Это бравая попытка, и поскольку релиз Arduino 1.0 серьезно пересмотрел принципы работы всего проекта, уместным оказалось обновить и эту Gnop'ифицированную версию.

С *Gnoduino* вы можете рассчитывать на разумную, аккуратную, читаемую среду, которая сочетает все функции, необходимые в IDE Arduino, с интерфейсом, не являющим собой наждак для глаз, и способностью чаще соединяться с Arduino, чем нет. Кроме того, она написана

«Можете рассчитывать на разумную, аккуратную, читаемую среду.»



Выделение синтаксиса и приятные шрифты — все это часть объекта `GTKSourceView`, вокруг которого построен код.

на Python, и ее можно установить обычным способом:

```
sudo python setup.py install
```

Правда, недостаток ее в том, что вся функциональность Arduino целиком пока не охвачена.

Интерфейс OCR

YAGF

Версия 0.92 Сайт <http://symmetrica.net/cuneiform-linux/yagf-en.html>

Долгие годы, а возможно, даже тысячелетия, люди считали нужным записывать информацию одними предметами на других.

Ныне мы зовем это письменностью, однако, глядя на огамские [древнеирландские, – прим. пер.] палочки для письма, скорее подумаешь, что налицо процесс процарапывания. Тем не менее, мир полон письменных материалов, которые и вправду написаны. Есть даже компании, которые берут слова, созданные разумным, цифровым способом, и превращают их в пятна краски на кусках бумаги. Да еще и продают их! Чудеса! (Читателей данной версии LXF, доступной через Ubuntu Software Centre, это, видимо, удивит больше, чем тех, кто пролистывает этот номер в книжном магазине.)

Многие стараются вернуть эти кляксы в мир полезного цифрового текста, не жалея трудов. Процедура примерно такова: вы берете пятнистое изображение и просите какую-нибудь программу выдать вам

предположение о том, что здесь написано. Смазанные пятна нередко сбивают программу с толку, но в основном она работает хорошо. Фактически, есть два отличных движка для кляксодешифровки – *Tesseract* (который раньше разрабатывался HP и вышел под лицензией Apache License в 2005 г.) и *Cuneiform* (разрабатывался российской фирмой Cognitive Technologies и вышел под лицензией FreeBSD в 2008 г.).

Единственный недостаток этих во всем остальном превосходных движков – то, что они именно движки. При обработке страниц пакетом для них будет, вероятно, быстрее настроить командную строку, но для однократных или особо хитрых случаев желателен удобный интерфейс, способный загрузить сканированные изо-



► Принцип «хлам на входе — хлам на выходе» при конвертировании документов в цифровой текст пока что сохраняется.

«YAGF прост в использовании, а его установка — сущий пустяк.»

бражения или даже вызвать XSane для выяснения, что там сейчас сидит в вашем сканере. Вот это и делает YAGF. Легко.

YAGF прост в использовании, а его установка – сущий пустяк, поскольку он использует систему CMake; ознакомьтесь с инструкциями по установке, включенными в исходник.

Аудиоплеер

Pogo

Версия 0.7 Сайт <https://launchpad.net/pogo>

Как только ни носились с рисунками первобытных людей. Намалеванные на стенах пещер животные и человечки-охотники из палочек вполне могли быть высоким искусством, примитивной формой письменности или списком покупок, вот и все, что мы знаем. Но, по крайней мере, нам хоть что-то перепало из этого рода искусства.

Непонятым остается, как же тогда игралась музыка, ведь примитивные MP3-плееры, вероятно, были сделаны из глины и соломы, или подобных же материалов, почему до нас и не дошли. Аналогично, уцелело лишь пренебрежимо малое количество ноутбуков тех времен, и нам уже не суждено узнать, предпочитали ли они *Banshee* или *Rhythmbox* – или вообще что-то третье. Кое-кто небезосновательно полагает, что любимым плеером мог бы быть *Pogo*.

Почему? А потому, что он прост. *Pogo* не пытается подсунуть вам то, что, по его

личному мнению, следует слушать, или затасчить вас в музыкальный магазин, или подключить к какому-нибудь Facebook, чтобы ваши якобы друзья узнали, какой клевый музон вы сейчас гоняете. С непосредственностью каменного века он играет заказанную музыку.

Эффективные функции сканирования отыщут аудиофайлы и сгруппируют их, чтобы вы могли создать упорядоченный плей-лист (или просто щелкнуть по треку, который хотите послушать). Если вежливо его попросить, в порядке любезности он займется поиском обложки альбома, и ее тоже отобразит. Вот и все. Будучи написан на Python на движке *Gstreamer*, с интерфейсом на *GTK*, *Pogo* бы-



► Продавайся он в консервных банках, он бы делал именно то, что на банке написано.

стро, эффективно, без проблем и без затей воспроизводит музыку для вас.

Зачем он нужен, если можно наслаждаться всеми наворотами *Amarok* или других упомянутых нами программ? Затем, что он выполняет свою работу быстро и просто, и не требует стопки DVD с библиотеками поддержки или постоянного сетевого соединения. Если кому нужен музыкальный проигрыватель для нетбука, киоска или ноутбука, то раз уж *Pogo* (вероятно) выбрали во времена неолита, вам он тоже отлично подойдет.

«Pogo быстро, эффективно и без проблем воспроизводит музыку.»

Конвертирование аудио

Gnac

Версия 0.2.4 Сайт <http://gnac.sourceforge.net>

Подобно большинству полезных приложений, о которых мы часто рассказываем на этих поистине волшебных страницах, *Gnac* не стремится умничать. Он старается сделать что-то одно, зато уж как следует. В данном конкретном случае – конвертировать аудиофайлы из одного формата в другой.

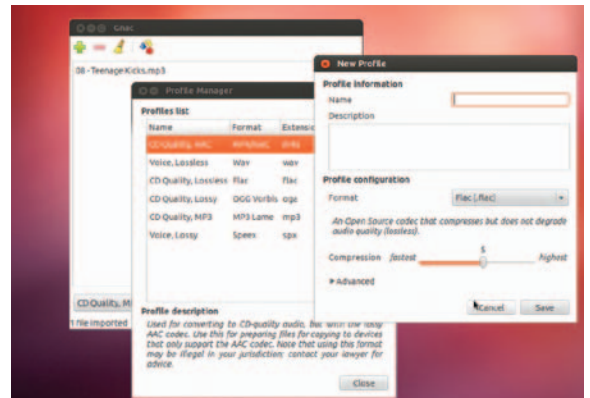
Если вам приходится этим заниматься в промышленном масштабе, вы, вероятно, стали экспертом по обработке файлов из командной строки с помощью утилиты *gststreamer*. Результаты будут практически такими же, потому что *Gnac* для конвертирования пользуется *gststreamer*. Но обеспечивает немного более удобную в использовании оболочку для этих многогранных инструментов терминала.

Самая полезная функция – профили. Эти подборки настроек можно редактировать и сохранять, или создавать с нуля, чтобы они соответствовали именно тому типу файла, который вы хотите создать.

Помимо возни с форматами (*Gnac* справится с чем угодно, лишь бы были кодеки *gststreamer*), можно также изменять особые настройки, например, скорость передачи данных при конверсии.

Ряд полезных функций – например, возможность удалять файлы после того, как они были конвертированы, или выделять сомнительные символы – немного упростит вам деятельность по конвертированию.

Gnac также умеет работать с метаданными ID3, и экспортирует эти данные, если файловый формат на выходе их поддерживает; и вам нечего опасаться, что все ваши файлы при смене формата превратятся в неопределенные отрезки шума без роду и племени.



► Конвертирование форматов файлов – не самая главная забота, но *Gnac* хоть чуть-чуть да облегчает этот процесс.

Данная версия является обновлением к релизу 2.4, вышедшему ранее в этом году и включившему переход на интерфейс GTK3 и множество разных многоязычных опций.

Работа с ним напомнила мне мою поездку в Европу с Майком. Мы рассуждали о том, как изъять с наших телефонов интервью, не имея ничего под рукой для их конвертации. Спустя пару кружек пива мы закачали 'sox' на кракнутый андроидофон Майка, и день был спасен. А будь у нас под рукой *Gnac* и достойная Linux-машина, мы бы провели время намного лучше – правда, пиво бы выпили меньше.

«Ряд полезных функций немного упростит вам конвертирование.»

Бальзам для глаз

Marble

Версия 13 Сайт <http://edu.kde.org/marble>

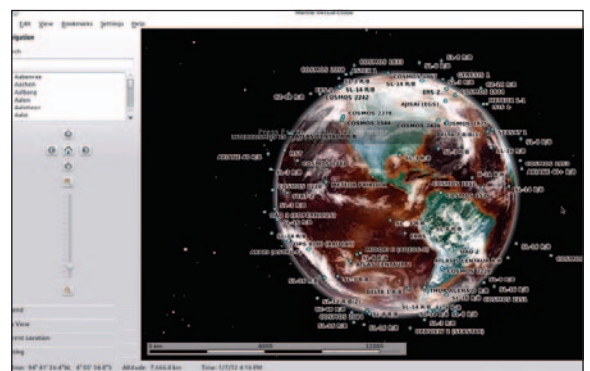
Разве плохо, когда душа просит красоты? Говорят, что красота – в глазах смотрящего, однако немало важной представляется и такая деталь, как местоположение, а применительно к обсуждаемой теме – местоположение этого самого смотрящего. Мошка, увеличенная микроскопом, выглядит явно милее, чем на вашей руке, занятая высасыванием вашей кровушки. Точно так же, трущобы Калькутты, скотобойни на Среднем Западе, жители... (вставьте сюда название любого малонаселенного английского городка, за выпад против которого нас не вздуют) и все человечество в целом очень выигрывают, если смотреть на них с достаточного расстояния.

Но если вы полагаете, что *Marble* – просто диковинка, поставляющая с рабочего стола приятный пространственный корм для вашей сетчатки, значит, вы недостаточно близко его рассматривали. Если ранние версии в основном представляли

собой просто симпатичные текстуры, натянутые на сферу, то затем *Marble* перерос эту стадию. Помимо вида Globe, здесь имеются различные наложения, отображающие Землю в разных видах, и *Marble* обзавелся интеграцией с OpenStreetMap, предоставив опцию прокладывания маршрута.

Но все это было давно, так же как и карты разнообразных типов. А теперь появились интереснейшие новинки, с которыми стоит познакомиться!

Для начала, интеграция с *Marble* добавлена в Plasma – вы можете ввести данные по месту, и автоматически запустится сессия *Marble* с центром именно в этом месте. А раз уж вы там, почему бы не оп-



► По спутниковым данным, на объекты накладываются высоты, и вы сможете понять их соотношение в трехмерном пространстве.

робовать голосовое сопровождение навигатора? Добавлено множество новых голосов, причем на нескольких языках. Теперь *Marble* может более или менее понимать карты в родном формате .osm, применяемые в OpenStreetmap, так что стали доступны слои многих дополнительных подробностей.

Мы тоже малость усомнились в способности превращения орнамента рабочего стола в полезный картографический инструмент, но данный релиз показал, что наши сомнения были напрасны.

«Можете ввести данные по месту, и запустится сессия Marble.»

HotGames Развлекательные приложения

Аркада

Asteroid

Версия 1.2 Сайт <http://chaoslizard.sourceforge.net>

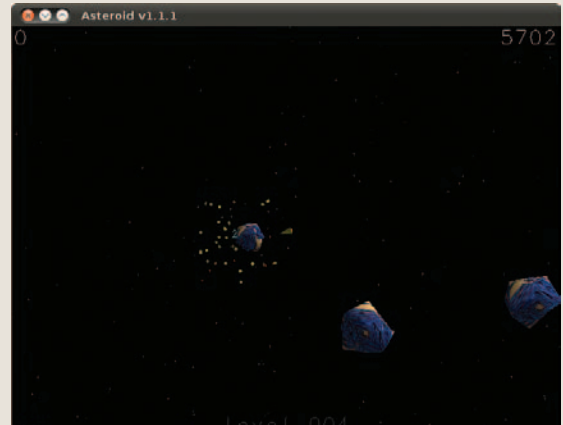
Недавно в новостях вдруг стала муссироваться идея исследования астероидов на предмет их коммерческой ценности. Отчасти это не лишено смысла, поскольку запасы металлов на Земле истощаются быстрее, чем хотелось бы. Например, титана, который, при современном темпе потребления, будет экономически невыгодно добывать на Земле уже к 2050 году. Что весьма грустно, ибо он крайне необходим для производства такой продукции, как... э-э, космические корабли.

Добыча минералов с космических объектов должна осуществляться посредством высадки на объект и тщательного извлечения руды, но куда веселее оснастить космический корабль мощным оружием и вышибить из этих объектов очередную молибден. И если вы строите планы трудоустройства где-то в космосе в период между сегодняшним днем

и 2050 годом, вам явно не помешает этот превосходный симулятор. Возможно, он очень похож на игру, основанную, вероятно, на одной из самых первых векторных классических игр Atari типа «разнеси-всё», но мы вас уверяем, что вы с ее помощью овладеете жизненно необходимыми навыками, стреляя и продираясь сквозь тернии.

Здесь старательно избегают любых намеков на реализм – например, звуковых эффектов и текстур OpenGL. Вообще-то пара-тройка лишних текстур игре явно не помешала бы: мы, например, всегда питали иллюзию, что астероиды отнюдь не идентичны. Но это уже при-

«Здесь старательно избегают любых намеков на реализм.»



» Ладно, мы при-
врали — это
не идеал карьер-
ного роста, но за-
то развеет горечь
от Евро-2012.

дирки. Это увлекательное и вполне приличное воссоздание классики, и хотя оно, может быть, и не станет подлинным испытанием для ваших отточенных игровых навыков, зато даст возможность потешить ностальгию и вспомнить те времена, когда все, что имело значение – это полеты, периодические атаки и много-много пальбы.

Возможно, вы не найдете *Asteroid* в локальном репозитории вашего дистрибутива (хотя для пользователей Ubuntu есть ресурс PPA), но игра легко компилируется из исходника.

Головоломка-пазл

Cube Trains

Версия 0.2.1 Сайт <http://ddr0.github.com>

Никогда не представляли себя главным архитектором? Если честно, этот вариант не так уж часто обсуждают в школах, говоря о выборе профессии, но зато практически каждый полагает, что справился бы с этой работой лучше тех, кто планирует городскую застройку – если ее вообще планируют.

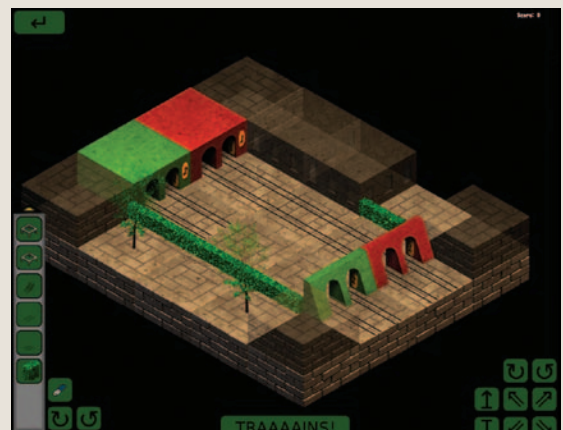
И даже если вы действительно градостроитель, вряд ли вам достался город с железнодорожной сетью из небольших фрагментов, разбросанных по всему городу – именно таков сценарий этой головоломки, непохожей на другие.

По своей сути, это пазл, состоящий из блоков. Используйте разные части железнодорожной инфраструктуры, чтобы соединить линии и пройти уровень. Иногда линий много, и вам нужно убедиться, что они полностью совпадают

(не пересекайте водных объектов!). В игре двадцать уровней, и они очень быстро усложняются, но есть возможность создания собственных. И уж как минимум, радует графика!

Игра пока на стадии бета-версии, так что не огорчайтесь, если случайная ошибка испортит вам песню, и помните, что слоты для сохранения иногда работают нештатно. Фиксированное разрешение экрана тоже слегка напрягает. Но во всем остальном удовольствие гарантируется. Компиляция игры сама по себе уже приключение. Есть ряд требований, удовлетворить которые вы, ви-

«Не огорчайтесь, если случайная ошибка испортит вам песню.»



» Вот бы внутреннее кольцо в Суиндоне было так спроектировано...

димо, не сможете, и она не использует стандартные автоинструменты; вы начнете смутно догадываться об их отсутствии, когда компиляция не удастся. Но на сайте есть весьма полезные инструкции (http://ddr0.github.com/linux_%20compiling_%20guide.html), которые стоит прочитать, если обычная команда `make` должным образом не сработает.

Редактирование изображений

Fotoxx

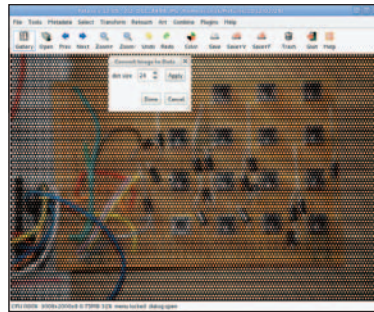
Версия 12.05 Сайт <http://kornelix.squarespace.com/fotoxx>

О *Fotoxx* уже не первый раз говорится в HotPicks. Дело в том, что он – хотя и в престранной манере – весьма хорош, но не похож ни на один из инструментов для работы с фото в Linux. Если вы ищете популярный фоторедактор для продвинутого пользователя, то, вероятно, выберете *DigiKam*. Однако *Fotoxx* ушел с проторенного пути предсказуемости на нехоженые тропы экспериментов, и результаты явно интересны.

Взять, к примеру, эффект Лихтенштейна [Lichtenstein effect]. В *GIMP* вам, пожалуй, удастся создать его за пару часов мороки с цветовыми каналами, а здесь он фигурирует как простой фильтр эффекта, способный на очень достойные результаты. Конечно, каждый день такое не требуется, но в том-то и суть: *Fotoxx* предоставляет инструменты для создания чего-то неординарного. Для обычных действий инструменты в нем тоже есть, но опять-таки с необычной методикой. Некоторые инструменты ретуширования проще в использовании, чем их эквиваленты

в *GIMP* – скажем, возможность «рисовать [paint]» исправления в заданной цветовой гамме и с заданной яркостью. *Fotoxx* поддерживает файлы формата RAW и внутри работает с 24-битной глубиной цвета, так что вы не потеряете ни малейшей частички бесценной информации, содержащейся в вашем изображении.

Компиляция из исходника способна несколько утомить, и порой пакеты репозитория не успевают за темпом разработки, но если вам не терпится найти самые свежие функции, загляните на сайт.



» Сойдите с мейнстрима в нетривиальный мир экспериментов *Fotoxx*.

Программа для заметок

Braindump

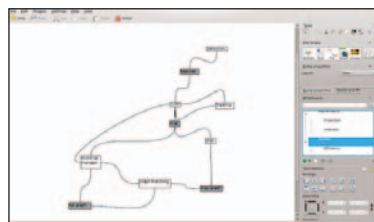
Версия 2.4 Сайт www.calligra.org/braindump

В *raindump* – часть пакета приложений *Calligra*. Если раньше вы о нем не слышали (мы делали обзор в прошлом выпуске!), то, возможно, потому, что это – новое название для комплекта приложений *KOffice*, или, по крайней мере, большинства из них. Но охват был задуман чуть шире, что и вызвало появление данного инструмента графического отображения идей. Он непохож на более структурированные аналоги. Хотя и можно создавать пути и связывать объекты, чтобы получить устрашающую диаграмму, в чем-то являющую собой срез вашей мыслительной деятельности, *Braindump* скорее напоминает альбом вырезок, куда вы стаскиваете элементы или ресурсы с вашего рабочего стола на бесконечный холст, наполненный... ну, чем вы там его наполняете.

Польза переименования пакета как *Calligra* не очень ясна. В общем, бренд паке-

та *KOffice* (который более или менее отмер как единое целое) добился известности, но не всегда в хорошем смысле: были в нем превосходные компоненты, но были и не столь достойные. Интереснее новые компоненты *Calligra* – тот же *Braindump*.

Braindump можно скомпилировать отдельно из кода на DVD, но это слегка проблематично и менее полезно без прочих частей *Calligra*, так что с тем же успехом можно установить все полностью. Однако не забудьте проверить, имеется ли пакет *Calligra* для вашего дистрибутива. **LXF**



» Мы опробовали *Braindump* в редакции LXF – результаты оказались пугающе прискорбными.

Также вышли

Новые и обновленные программы, тоже достойные внимания...

» **gPodder 3.1.2**

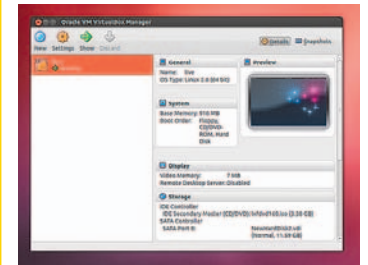
Собирайте в Сети лучшие подкасты и управляйте ими. <http://gpodder.org>

» **SquidClamAV 6.6**

Интеграция антивирусной программы *Clam* с прокси *Squid* для защиты вашей сети от вирусов. <http://squidclamav.darold.net/index.html>

» **VirtualBox 4.1.16**

Запускайте компьютер внутри компьютера при посредстве этой простой в использовании программы виртуализации. www.virtualbox.org



» *VirtualBox* – отличная возможность испытать разные дистрибутивы.

» **Wireshark 1.6.8**

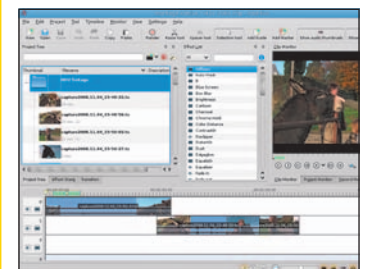
Делает закладки для информации, содержащейся в сети. www.wireshark.org

» **Cura 1.0**

Зачем плестись в душный офис? Выполняйте свои обязанности администратора с пляжа, через Android на вашем мобильном телефоне. www.ohloh.net/p/cura

» **Kdenlive 0.9**

Редактирование видео в очередной раз упростилось. www.kdenlive.org



» Редактируйте свое домашнее видео, или создайте нового Рэмбо.

На диске

Пробуйте новую операционную систему сегодня!



В мире дистрибутивов этот месяц был отмечен большой активностью: вышли основные релизы от Fedora, Mint и Mageia — и все они есть на DVD. Это все тяжеловесы, но мы в LXF любим и малые дистрибутивы. Месяц назад мы представили вам Arch Linux, а на этот раз включили один из его производных, ArchBang. Мы спрашивали людей, нужны ли на диске ISO, и некоторые ответили, что считают их удобными. Поэтому, начиная с данного

выпуска, мы планируем их включение, но для легковесных дистрибутивов, которые с большей вероятностью пригодятся для старых машин, в которых нет DVD-приводов.

Все дистрибутивы на диске этого месяца 64-битные, кроме 32-битного ArchBang, который поэтому должен работать на почти любом оборудовании. Также на DVD вы найдете новый справочник администратора Debian — Administrator's Handbook.

Вкусите Мясистое Чудо

Fedora 17

Даже при миллиардном годовом доходе в Red Hat не забыли о маленьких людях. Fedora, проект компании для сообщества, только что вы-

пустил свою 17-ю версию под кодовым названием Beefy Miracle. Самое главное в этом релизе — Gnome 3.4, улучшенный по сравнению с прежними версиями.

Львиная доля труда была посвящена упрощению работы с приложениями, плюс разработчики Fedora отказались от обновленного Web (ранее — Epiphany) в пользу Firefox как браузера по умолчанию. Еще одна функция, вызвавшая ажиотаж в редакции LXF — способность live CD загру-



«Разработчики Fedora отказались от Web в пользу Firefox.»

жаться на оборудовании Apple, но увы — мы не смогли воспроизвести ее на нашем DVD из-за проблем с мультизагрузкой.

Более близкие к технике пользователи заметят реформы в раскладке файловой системы вековой давности. Ряд папок (типа /bin и /lib) изъят из / и перенесен в /usr. Но остались ссылки на традиционное размещение, так что это не должно вызвать проблем у большинства пользователей.

» Gnome 3.4 содержит множество улучшений... но в нем по-прежнему нет кнопки выключения.

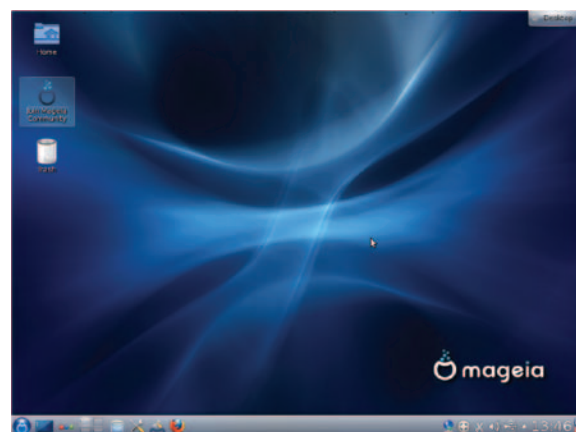
Это — магия Mageia

Mageia

Несмотря на то, что ему всего год, Mageia уже заслужил репутацию качественного дистрибутива KDE. Рабочий стол являет собой больше, чем простой KDE, и разработчиков

явно вдохновлял пример Unity от Canonical (см. обзор на стр. 17). Это порадует одних и разочарует других — как и Unity. Однако, в отличие от Unity, рабочий стол сохранил фирменную способность к настройке KDE, и если кто не в восторге от предложенного, можно переделать его на свой вкус.

Поскольку будущее Mandriva (дистрибутива, производным от которого является Mageia) неопределенно — он пытается превратиться из коммерческого продукта в проект сообщества — создается впечатление, что именно Mageia станет естественным курсом обновления для поклонников этой разновидности Linux.



» Mageia 2, отпрыск Mageia 1 и вероятный наследник трона проблемного Mandriva.

«Разработчиков явно вдохновлял пример Unity.»

Чтобы все было просто

ArchBang

Все дистрибутивы на DVD этого месяца 64-битные, а этот – 32-битный. Как видно из его названия, во многом он вдохновлен CrunchBang – и тоже легковес, но немного потяжелее CrunchBang, и здесь имеется чуть больше графических прелестей, установленных по умолчанию. По его названию также легко догадаться и о том, что он создан поверх Arch, который уделял большое внимание наделению пользователей контролем за своим компьютером. И это превращает его в отличный дистрибутив для тех, кто ищет готовую графическую рабочую станцию (которую родители не покупают) и при этом жаждет получить тонко настроенное управление свойствами системы.

Менеджер окон *OpenBox* отличается быстротой и простотой в использовании. Щелчок правой кнопкой по рабочему столу выведет меню Applications. Следуя вашим

откликам на прошлый месяц, мы включили его в виде ISO, и вам придется перенести его на другой носитель, перед тем как загружать. В качестве альтернативы можно использовать непосредственно сам ISO на виртуальной машине, используя *VirtualBox* или *Qemu*, если хотите познакомиться с ним без установки.



» *OpenBox* и Arch – отлично для старого оборудования и тонкого контроля.

Важно ВНИМАНИЕ!

» Перед тем, как вставить DVD в дисковод, пожалуйста, убедитесь, что вы прочитали, поняли и согласились с нижеследующим.

Диски *Linux Format* DVD тщательно проверяются на предмет отсутствия на них всех известных вирусов. Тем не менее, мы рекомендуем вам всегда проверять любые новые программы надежным и современным антивирусом.

Хотя процесс отбора, тестирования и установки программ на DVD проводится со всем тщанием, редакция *Linux Format* не несет никакой ответственности за повреждение и/или утрату данных или системы, могущее произойти при использовании данного диска, программ или данных на нем. Настоятельно рекомендуем вам создавать своевременные и надежные резервные копии всех важных файлов.

Чтобы узнать об условиях использования, просим вас прочесть лицензии.

Бракованные диски

В маловероятном случае обнаружения бракованного диска *Linux Format*, просим связаться с нашей группой поддержки по адресу disks@linuxformat.ru, для получения содействия.

Теперь с Mate

Mint

Когда Gnome и Ubuntu отказались от традиционного рабочего стола, пользовательская среда поплыла негодованием. Люди отзывались о новых системах как о шаге назад в плане удобства. Mint не последовал за другими рабочими столами в новую парадигму, сохранив более знакомый интерфейс, и поднабрал массу пользователей. Сейчас страсти улеглись, программы стали бо-

лее зрелыми, и мы можем сравнить разные среды.

Это издание Mint выходит с Mate, ответвлением проекта Gnome 2. Здесь сохранен старый подход Gnome 2 к работе, но это не просто реликт, уцелевший от прошлого. Разработчики постоянно обновляют его, и он по-прежнему остается современным.

С помощью DVD этого месяца вы можете провести сравнение двух Gnome, потому что не только Mint вышел с последней версией Mate, но и Fedora



» Mint прославился, предлагая новые рабочие столы в традиционном стиле, и этот релиз тоже не подкачал.

идет с последней версией Gnome 3, что является солидным усовершенствованием по сравнению с первыми попытками магических заклинаний. Была ли оправдана изначальная критика? Попробуйте и вы, и другое – и решите для себя сами. LXF

«Mint, сохранив знакомый интерфейс, набрал пользователей.»

На диске

Помимо четырех отличных дистрибутивов, мы включили весь код из учебников, все новинки и все программы из всех разделов журнала, где вы видели плашку **На диске**. Будучи установлен в ваш компьютер, DVD должен по умолчанию открыть HTML-страницу; но если ваши настройки безопасности этого не позволяют, перейдите в [index.html](#), чтобы открыть для себя все 4,4 ГБ Linux-добра. Чтобы познакомиться с нашими дистрибутивами, установите диск

в дисковод и перезагрузите компьютер. Загрузится экран, где вы сможете выбрать то, что вам нужно. Если ваш компьютер загружается как обычно, а не с диска, вам придется вручную изменить настройки BIOS, чтобы активировать загрузку DVD. Если вы предпочитаете перенести содержимое диска на USB-брелок, мы включили для этого удобный скрипт. Загляните в раздел [dvd2usb.html](#) на диске, чтобы узнать, как он работает. Подписчики цифро-

вой версии, обратите внимание, что вы можете просто использовать **dd** для переноса скачанного ISO на USB, если вам это потребуется. Используйте `sudo dd if=linuxformat.iso of=/dev/sdX` включая путь до скачанного ISO при необходимости, где **X** изменяется соответственно для USB-драйва. **Это уничтожит все данные на диске. Если вы установите не тот диск, он может стереть данные с жесткого диска.**

Пропустили номер?

» Мир свободного ПО богат и разнообразен, а потому далеко не все можно вместить в рамки одной статьи. Linux Format обходит эту проблему, публикуя серии статей по самым актуальным вопросам. Но что делать, если вы поймали интересующий вас материал на середине? Обратитесь в «Линуксцентр» по адресу www.linuxcenter.ru и закажите желаемый номер журнала! Он доставляется как в печатной, так и в электронной форме, поэтому с момента открытия браузера и до получения нужного вам выпуска LXF может пройти не более нескольких минут!

Прямо сейчас для заказа доступны следующие номера:

 <p>LXF157 Май 2012</p> <ul style="list-style-type: none">» Linux — лидер Наша любимая ОС впереди планеты всей» Квартирка для проекта Какой хостинг-провайдер лучше» Node.js Альтернативная среда для JavaScript» Изучаем SQL База данных для телесериалов <p>LXFDVD: Sabayon 8, openSUSE 12.1 Cinnamon Remix и еще 2 загружаемых дистрибутива, remastersys 3.0.0, DiffImage, IP-Link, PedSim и прочее...</p> <p>Печатная версия: shop.linuxformat.ru/lxf_157/ PDF-версия: shop.linuxformat.ru/elxf_157/</p>	 <p>LXF158 Июнь 2012</p> <ul style="list-style-type: none">» Надуем ЦПУ Как не попасться во Всемирную паутину» Братья наши меньшие Альтернативные ОС» The Journal Новый метод ведения системных журналов» FontForge Рисуем собственный шрифт <p>LXFDVD: Tails и еще 3 дистрибутива, Audacity, FontForge, KtechLab, Qemu 1.0.1 и прочее...</p> <p>Печатная версия: shop.linuxformat.ru/lxf_158/ PDF-версия: shop.linuxformat.ru/elxf_158/</p>	 <p>LXF159 Июль 2012</p> <ul style="list-style-type: none">» Linux против Windows Если слон на кита налезет, кто кого соберет?» Рулим обновлениями Менеджеры пакетов – без них никуда» OpenCL Открытый стандарт для параллельного программирования» Презентации Как обаять аудиторию и заразить своей идеей <p>LXFDVD: Ubuntu 12.04 LXF Remix и еще 3 дистрибутива, FWBuilder, LibreOffice 3.5.2 и прочее...</p> <p>Печатная версия: shop.linuxformat.ru/lxf_159/ PDF-версия: shop.linuxformat.ru/elxf_159/</p>
--	--	---

Ну, а если вы хотите быть уверенными, что не пропустите ни один номер журнала – оформите подписку! Помните, что все подписавшиеся на печатную версию журнала через www.linuxcenter.ru или shop.linuxformat.ru получают электронную версию в подарок!

Подписывайтесь на журнал на www.linuxformat.ru/subscribe/

Телефоны отдела подписки: Санкт-Петербург (812) 309-06-86, Москва (499) 271-49-54

Специальное предложение

Купите подборку журнала!

К нам в редакцию периодически приходят письма с вопросами, где можно купить предыдущие выпуски LXF. Если вы тоже этим озадачены, то в интернет-магазине «ГНУ/Линуксцентра» продолжается продажа журналов за 2011 год. Вы можете приобрести как отдельные номера изданий, так и подписки на 6 или 12 месяцев. Спешите – журналов осталось не так уж много!

shop.linuxformat.ru



Информация о диске

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, первым делом следует заглянуть именно туда.

Форматы пакетов

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любых других. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными двоичными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы могли 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 и 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 двоичный 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.tgz** – тот же файл, что и выше этажом 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** – двоичный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;
- » **имя_программы-1.0.1.ppc.Suse9.rpm** – двоичный 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

Внимательно прочтите это перед тем, как использовать LXF DVD!

А ТАКЖЕ: LibreOffice 3.5, GIMP 2.8, Virtual Machine Manager 0.9.1 и другие обновления

» Ядро 3.3.4 » Gnome 3.4 и KDE 4.8 » Open Switch и Sandbox » Обновленная версия OpenStack Site

64-разрядная сборка



fedora 17

А ТАКЖЕ: BCCD, Fotoxx, YAGF и руководство администратора Debian

ПЛЮС ЕЩЕ ТРИ ДИСТРИБУТИВА:
» ArchBang » Fedora 17 » Mint 13 Mate

mandriva 2

Обновленный релиз «Потомка» Mandriva

Август 2012
LXF DVD 160

LINUX
FORMAT

Август 2012
LXF DVD 160

LINUX
FORMAT

Содержание

LINUX FORMAT

ДИСТРИБУТИВЫ

- ArchBang 2012.04** Облегченный дистрибутив на базе Arch с рабочим столом OpenBox, 32-разрядная сборка (ISO-образ)
- BCCD 3.0.3** (Bootable Cluster CD) Live CD для создания вычислительных кластеров, 32-разрядная сборка (ISO-образ)
- Fedora 17** Новая версия популярного дистрибутива, 64-разрядная сборка (загрузка с LXF DVD)
- Fedora 17 DVD** Установочный DVD, 64-разрядная сборка (вторая сторона LXF DVD)

- Mint13** Новый релиз с рабочим столом Mate, 64-разрядная сборка (загрузка с LXF DVD)
- Mageia 2** Вторая версия ответвления Mandriva, разработаемого сообществом, 64-разрядная сборка (загрузка с LXF DVD)

ДОКУМЕНТАЦИЯ

- The Debian Administrator's Handbook** Руководство администратора Debian (на английском языке)

НОТРИСКС

- Asteroid 12** Игра на «классическую» тему спасения Земли от астероидной опасности
- BrainDump** Программа для организации и визуализации заметок любого рода (текста, рисунков, диаграмм и т.д.) из комплекта приложений Sailfish 2.4.1
- CubeTrains 0.2.1** Игра, управление поездами и создание транспортной системы
- Fotox 12.05** Программа для работы с фотографиями в формате RAW
- Glas 0.2.4.1** Приложение Gnome для конвертирования аудиофайлов из одного формата в другой
- Gnollino 0.4.0** Версия графической среды разработки Arduino для Gnome
- KTorrent 4.2.0** Один из наиболее популярных BitTorrent-клиентов
- Marble 1.2.0** Виртуальный глобус и атлас мира
- Monit 5.4** Универсальное средство мониторинга служб, событий файловой системы, сетевых соединений и загрузки аппаратных ресурсов
- Pogo 0.7** Музыкальный проигрыватель с минимальным набором функций и простым интерфейсом
- YAGF 0.9.1** Графическая оболочка для программ распознавания текста Omelet и Tesseract

Пожалуйста, перед использованием данного диска ознакомьтесь с опубликованной в журнале инструкцией!

КОММЕНТАРИЙ Присылайте ваши пожелания и предложения по электронной почте: shoval@lxf.com

Дорогие диски! В маловероятном случае обнаружения дефектов на данной диске, обращайтесь, пожалуйста, по адресу dsk@lxf.com

Настоящий диск тщательно тестировался и проверялся на всех этапах производства, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать аппаратный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. К сожалению, редакция Linux Format не в состоянии принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые могут повлечь за собой использование этого DVD, представляющих собой программы или данные. Прежде чем устанавливать какие-либо ПО на компьютер, пожалуйста, скачайте программу для загрузки с нашего сайта www.linuxformat.ru

Тираж издательства ООО «Уральский электронный завод», 620137, Россия, г. Екатеринбург, Студенческая ул., д. 9. Лицензия МПР ВАР 77-15.

Создание установочных дисков при помощи cdrecord

Самый быстрый способ записать ISO-образ на чистую матрицу – это через *cdrecord*. Для всех перечисленных ниже действий потребуются права root. Сначала определите путь к вашему устройству для записи дисков. Найдите 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 – метка; затем после адреса SCSI-устройства вы должны указать скорость и размер буфера. Теперь 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 и 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 с Windows, Mac OS X, AmigaOS или любой другой ОС.


Нет устройства для записи дисков?

А что если у вас нет устройства, с помощью которого можно было бы записать образ на диск? Вы знаете кого-нибудь с таким устройством? Вам не обязательно использовать Linux для записи дисков: подойдет любая операционная система, способная распознать пишущий привод (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 разработчика дистрибутива.

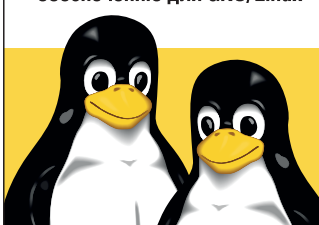
ФОРУМ №1

ДЛЯ ВСЕХ
ПОЛЬЗОВАТЕЛЕЙ
LINUX



LINUXFORUM.RU

LinSoft.Info
Путеводитель по программному
обеспечению для GNU/Linux




WWW.LINSOFT.INFO

Linux по-русски



**Библиотека
книг, статей
и переводов
о Linux**

WWW.RUS-LINUX.NET

Read.ru



Только в Read.ru приобретая любые
журналы Linux Format на сумму от 500 руб.,
вы получаете подарок – DVD архив
журнала 2005-2011

LINUX FORMAT

Главное в мире Linux

Как разместить рекламу в разделе Classifieds?

1/4 полоса (210 × 297 мм)	165 200 руб.
1/2 полосы горизонтально (197 × 144 мм)	88 500 руб.
1/2 полосы вертикально (102 × 278)	88 500 руб.
1/4 полосы вертикально (98 × 138 мм)	53 100 руб.
Фотоблок (44 × 113 мм)	15 000 руб.

Тел.: +7 812 309 06 86

Цены указаны с учетом НДС

ITMOZG.RU

БОЛЕЕ 4000 ИТ-ВАКАНСИЙ

Работа в интереснейших стартапах
или крупнейших ИТ-компаниях.
Большой выбор задач, зарплат и
условий



Unix Education Center

Россия, 190000, Санкт-Петербург
Черноморский переулок, дом 4
Тел.: +7 (812) 611-1575



ГНУ/Линуксцентр приглашает на работу!

Linux center

ВАКАНСИЯ: PHP-программист/web-мастер

ОБЯЗАННОСТИ:

- » Создание сайтов с нуля на базе CMS Drupal: верстка, программирование, разработка баз данных, конвертация данных.
- » Разработка дополнительных модулей для CMS Drupal.
- » Внесение изменений в готовые скрипты.
- » Поддержка сайтов компании.

ПОДРОБНЕЕ: www.linuxcenter.ru/vacancy/

Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия ПИ № ФС77-21973 от 14 сентября 2005 года. Выходит ежемесячно. Тираж 5000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ

Главный редактор
Кирилл Степанов info@linuxformat.ru
Литературный и выпускающий редактор
Елена Толстякова
Переводчики
Елена Ессяк, Светлана Кривошеина,
Валентин Развозжаев, Елена Толстякова
Редактор диска
Кирилл Степанов
Верстка, допечатная подготовка
Сергей Рогожников
Технический директор
Денис Филиппов
Директор по рекламе
Владимир Савельев advert@linuxformat.ru
Генеральный директор
Павел Фролов
Учредители
Частные лица
Издатели
Виктор Федосеев, Павел Фролов

Отпечатано в ООО «Ланинь»

188330, Ленинградская обл., Гатчинский р-н, пос. Сиверский, Вокзальная ул., 4
Заказ 5018

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ

Редактор Грэм Моррисон [Graham Morrison] graham_morrison@futurenet.com
Заместитель редактора Эндрю Грегори [Andrew Gregory] agregory@futurenet.com
Штатный автор Джонатан Робертс [Jonathan Roberts] jonathan.roberts@futurenet.com
Редактор диска Бен Эверард [Ben Everard] ben_everard@futurenet.com
Художественный редактор Эфраин Эрнандес-Мендоса [Efraim Hernandez-Mendoza] efrain.hernandez-mendoza@futurenet.com
Выпускающий редактор Гэри Уокер [Gary Walker] gary.walker@futurenet.com

ПОДГОТОВКА МАТЕРИАЛОВ

Нейл Ботвик [Neil Bothwick], Крис Браун [Chris Brown], Марко Фиоретти [Marco Fioretti], Джульетта Кемп [Juliet Kemp], Бен Эверард [Ben Everard], Дэвид Хейворд [David Hayward], Грэм Моррисон [Graham Morrison], Адам Оксфорд [Adam Oxford], Джонатан Робертс [Jonathan Roberts], Маянк Шарма [Mayank Sharma], Майк Сондерс [Mike Saunders], Ник Вейч [Nick Veitch], Евгений Балдин, Артем Зорин, Тихон Тарнавский, Андрей Ушаков, Алексей Федорчук, Игорь Штомпель, Роман Ярыженко

Художественный ассистент Кейт Мак-Доннелл [Kate McDonnell]

Иллюстрации Шейн Коллиндж [Shane Collinge], Eily Walton Illustrations, iStockPhoto, Саймон Мидлвик [Simon Middleweek]

КОНТАКТНАЯ ИНФОРМАЦИЯ

UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW
Tel. +44 01225 442244 Email: linuxformat@futurenet.com

РОССИЯ:

Санкт-Петербург (редакция):
Лиговский пр., 50, корп. 15
Тел. +7 (812) 309-06-86
Представительство в Москве:
Красноказарменная ул., 17, м. «Авиамоторная» (в помещении АТС МЗИ)
Тел./факс +7 (499) 271-49-54
По вопросам сотрудничества, партнерства, оптовых закупок:
partner@linuxcenter.ru

Авторские права: Статьи, переведенные из английского издания Linux Format, являются собственностью или лицензированы 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.

Ответственность за содержание статьи несет ее автор. Мнение авторов может не совпадать с мнением редакции.

Все присланные материалы могут быть помещены на CD или DVD-диски, 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 ответственности за повреждения или ущерб, нанесенные вашему компьютеру и периферии вследствие использования тех или иных советов.

LINUX – зарегистрированный товарный знак Линуса Торвальдса [Linus Torvalds]. «GNU/Linux» заменяется на «Linux» в целях сокращения. Все остальные товарные знаки являются собственностью их законных владельцев. Весь код, опубликованный в журнале, лицензирован на условиях GPL v3. См. www.gnu.org/copyleft/gpl.html

За информацией о журнале, издаваемом Future plc group company, обращайтесь на сайт <http://www.futureplc.com>



© Linux Format 2005

© Future Publishing Ltd 2005

BATH • LONDON • MILAN • NEW YORK • PARIS • SAN DIEGO • SAN FRANCISCO



В сентябрьском номере

«Железо»

Иногда аппаратура работает превосходно, иногда вообще не работает – отделим балласт от полезного груза и построим идеальную Linux-машину.

Анатомия жесткого диска

Подкачка, root, home, sda4, ext 3 – узнайте, что это значит, и поймете, как действуют ваши магнитные диски.

FreeNAS

Запросто установите домашний сервер, используя NAS (или обветшалый ноутбук), и ваша жизнь улучшится на 12 %.

Виртуализируйтесь на славу

Чтобы реально освоить Linux, надо сперва его взломать. С VirtualBox вы это сделаете, не рискуя помрачить рассудок.

Содержание последующих выпусков может меняться без уведомления – вдруг нам придется дожидаться остывания золы, иначе ее не выбросишь...



Mandriva Linux — один из самых популярных дистрибутивов GNU/Linux в мире. Главные преимущества Mandriva — дружелюбный интерфейс, простота настройки, возможность быстрой адаптации пользователей, ранее не знакомых с этой ОС, совместимость с широким спектром программного и аппаратного обеспечения.

Корпоративные продукты Mandriva Linux

Mandriva 2010.2 Powerpack

Дистрибутив Mandriva 2010.2 Powerpack включает набор офисных и серверных приложений, и подходит для установки на офисной или домашней рабочей станции и на сервере. Дружелюбный интерфейс, простота настройки Mandriva Powerpack, совместимость с широким спектром аппаратного обеспечения и совместимость с «1С:Предприятие» обеспечивают корпоративным пользователям возможность легкого перехода с Windows на GNU/Linux.

Mandriva Enterprise Server 5

Mandriva Enterprise Server 5 (MES 5) — это надежный и производительный дистрибутив GNU/Linux для корпоративного сервера. MES 5 поможет вам снизить текущие расходы и упростить инфраструктуру. В MES 5 интегрированы серверные разработки программистов Mandriva, в том числе сервер каталогов пользователей Mandriva Directory Server, а также ведущие свободные серверные приложения, которые помогут вам с минимумом затрат времени и энергии настроить и поддерживать необходимые вам серверы. Срок поддержки дистрибутива — 5 лет.

Сертифицировано ФСТЭК

Дистрибутивы Mandriva Linux сертифицированы по требованиям ФСТЭК по 5 классу для СВТ и 4 уровню контроля НДВ, что дает возможность использовать их для обработки конфиденциальной информации в автоматизированных системах класса до 1Г включительно и обработки персональных данных в информационных системах класса до К2 включительно.

- **Mandriva 2008 Spring Powerpack** — дистрибутив для рабочей станции или небольшого сервера.
- **Mandriva Corporate Server 4 Update 3** — дистрибутив для создания корпоративного сервера.
- **Mandriva Flash** — дистрибутив GNU/Linux, загружающийся и работающий прямо с USB-носителя.

EduMandriva — свободное ПО для образования

- Создано с участием российских преподавателей и методистов.
- Все ПО, необходимое для преподавания информатики.
- Методические материалы.

Наименование	Стоимость, руб.
Корпоративные продукты Mandriva	
Mandriva Linux 2010.2 Powerpack (DVD-Box)	1 300
Услуга подписки на Mandriva Enterprise Server 5 на 1 год, базовый уровень (с физическим носителем)	13 300
Услуга подписки на Mandriva Enterprise Server 5 на 3 года, базовый уровень (с физическим носителем)	34 800
Продукты Mandriva для образования	
Комплект программного обеспечения Mandriva Linux и EduMandriva для школ	3 500
Сертифицированные ФСТЭК продукты Mandriva	
Сертифицированный ФСТЭК Mandriva 2008 Spring Powerpack на 10 рабочих мест	28 500
Сертифицированный ФСТЭК Mandriva 2008 Spring Powerpack на 5 рабочих мест	14 500
Сертифицированный ФСТЭК Mandriva 2008 Spring Powerpack на 1 рабочее место	4 990
Сертифицированный ФСТЭК Mandriva Corporate Server 4.0 Update 3	10 050

С вопросами по продуктам Mandriva обращайтесь в «ГНУ/Линуксцентр»!

MANDRIVA УЖЕ ИСПОЛЬЗУЮТ:
 МВД РФ, Минздравсоцразвития РФ,
 Минфин р. Саха (Якутия),
 Правительство Московской области,
 администрация Черниговского района,
 Приморского края, ОАО «Морион»,
 сеть магазинов «Компьютер-центр
 «КЕЙ», группа компаний «ИМАГ»,
 компания «Азбука мебели»,
 и многие другие.

Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

Linux-эксперт для вашего бизнеса. www.linuxcenter.ru



ЦЕНЫ В СВОБОДНОМ ПАДЕНИИ

HETZNER DEDICATED
ROOT SERVER **EX 4**

~~5500~~ руб.

1900 руб.

УСТАНОВКА

HETZNER DEDICATED
ROOT SERVER **EX 4S**

~~5500~~ руб.

2200 руб.

УСТАНОВКА

HETZNER DEDICATED
ROOT SERVER **EX 6**

~~5500~~ руб.

2900 руб.

УСТАНОВКА

HETZNER DEDICATED
ROOT SERVER **EX 5**

~~1900~~ руб.

0 руб.

УСТАНОВКА

HETZNER DEDICATED ROOT SERVER **EX 4**

- Intel®Core™ i7-2600 Quadcore с поддержкой технологии Hyper-Threading
- 16 ГБ DDR3 RAM
- 2 x 3 ТБ SATA 6 Гбит/с 7200 об/м! (Software-RAID 1)
- Операционная система Linux
- Неограниченный трафик*
- IPv6 подсеть (/64)
- Установка из образов
- Без минимального контракта
- Установка 1900 рублей

1900
рублей в месяц

HETZNER DEDICATED ROOT SERVER **EX 5**

- Intel®Core™ i7-920 Quadcore с поддержкой технологии Hyper-Threading
- 24 ГБ DDR3 RAM
- 2 x 750 ГБ SATA 3 Гбит/с (Software-RAID 1)
- Операционная система Linux
- Неограниченный трафик*
- IPv6 подсеть (/64)
- Установка из образов
- Без минимального контракта
- Установка 0 рублей

2500
рублей в месяц



GreenIT
Best Practice Award

2011

Hetzner Online активно поддерживает защиту окружающей среды, используя исключительно 100% возобновляемые источники энергии. Выберите более чистое будущее вместе с Hetzner Online!



RU.HETZNER.COM

*Трафик предоставляется бесплатно. При превышении порога 10000 ГБ/месяц скорость соединения будет ограничена до 10 MBit/s. Стоимость аренды постоянного канала с пропускной способностью 100 MBit/s составляет 290 рублей за каждый дополнительный ТВ.